produces an electronic pattern, or biometric, of the physical characteristic. The electronic pattern is transferred through the user system 105 to the trust engine 110 for either enrollment or authentication purposes.

Once the user produces the appropriate authentication data and the trust engine 110 determines a positive match between that authentication data (current authentication data) and the authentication data provided at the time of enrollment (enrollment authentication data), the trust engine 110 provides the user with complete cryptographic functionality. For example, the properly authenticated user may advantageously employ the trust engine 110 to perform hashing, digitally signing, encrypting and decrypting (often referred to only as encrypting), creating or distributing digital certificates, and the like. However, the private cryptographic keys used in the cryptographic functions will not be available outside the trust engine 110, thereby ensuring the integrity of the cryptographic keys.

According to one embodiment, the trust engine 110 generates and stores cryptographic keys. According to another embodiment, at least one cryptographic key is associated with each user. Moreover, when the cryptographic keys include public-key technology, each private key associated with a user is generated within, and not released from, the trust engine 110. Thus, so long as the user has access to the trust engine 110, the user may perform cryptographic functions using his or her private or public key. Such remote access advantageously allows users to remain completely mobile and access cryptographic functionality through practically any Internet connection, such as cellular and satellite phones, kiosks, laptops, hotel rooms and the like.

According to another embodiment, the trust engine 110 performs the cryptographic functionality using a key pair generated for the trust engine 110. According to this embodiment, the trust engine 110 first authenticates the user, and after the user has properly produced authentication data matching the enrollment authentication data, the trust engine 110 uses its own cryptographic key pair to perform cryptographic functions on behalf of the authenticated user.

A skilled artisan will recognize from the disclosure herein that the cryptographic keys may advantageously include some or all of symmetric keys, public keys, and private keys. In addition, a skilled artisan will recognize from the disclosure herein that the foregoing keys may be implemented with a wide number of algorithms available from commercial technologies, such as, for example, RSA, ELGAMAL, or the like.

FIGURE 1 also illustrates the certificate authority 115. According to one embodiment, the certificate authority 115 may advantageously comprise a trusted third-party organization or company that issues digital certificates, such as, for example, VeriSign, Baltimore, Entrust, or the like. The trust engine 110 may advantageously transmit requests for digital certificates, through one or more conventional digital certificate protocols, such as, for example, PKCS10, to the certificate authority 115. In response, the certificate authority 115 will issue a digital certificate in one or more of a number of differing protocols, such as, for example, PKCS7. According to one embodiment of the invention, the trust engine 110 requests digital certificates from several or all of the prominent certificate authorities 115 such that the trust engine 110 has access to a digital certificate corresponding to the certificate standard of any requesting party.

IPR2025-01200
IBM v. Security First Innovations, LLC
IBM's Exhibit 1010, Vol. 2
EX1010, Page 975

According to another embodiment, the trust engine 110 internally performs certificate issuances. In this embodiment, the trust engine 110 may access a certificate system for generating certificates and/or may internally generate certificates when they are requested, such as, for example, at the time of key generation or in the certificate standard requested at the time of the request. The trust engine 110 will be disclosed in greater detail below.

FIGURE 1 also illustrates the vendor system 120. According to one embodiment, the vendor system 120 advantageously comprises a Web server. Typical Web servers generally serve content over the Internet using one of several internet markup languages or document format standards, such as the Hyper-Text Markup Language (HTML) or the Extensible Markup Language (XML). The Web server accepts requests from browsers like Netscape and Internet Explorer and then returns the appropriate electronic documents. A number of server or client-side technologies can be used to increase the power of the Web server beyond its ability to deliver standard electronic documents. For example, these technologies include Common Gateway Interface (CGI) scripts, Secure Sockets Layer (SSL) security, and Active Server Pages (ASPs). The vendor system 120 may advantageously provide electronic content relating to commercial, personal, educational, or other transactions.

Although the vendor system 120 is disclosed with reference to the foregoing embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein that the vendor system 120 may advantageously comprise any of the devices described with reference to the user system 105 or combination thereof.

FIGURE 1 also illustrates the communication link 125 connecting the user system 105, the trust engine 110, the certificate authority 115, and the vendor system 120. According to one embodiment, the communication link 125 preferably comprises the Internet. The Internet, as used throughout this disclosure is a global network of computers. The structure of the Internet, which is well known to those of ordinary skill in the art, includes a network backbone with networks branching from the backbone. These branches, in turn, have networks branching from them, and so on. Routers move information packets between network levels, and then from network to network, until the packet reaches the neighborhood of its destination. From the destination, the destination network's host directs the information packet to the appropriate terminal, or node. In one advantageous embodiment, the Internet routing hubs comprise domain name system (DNS) servers using Transmission Control Protocol/Internet Protocol (TCP/IP) as is well known in the art. The routing hubs connect to one or more other routing hubs via high-speed communication links.

One popular part of the Internet is the World Wide Web. The World Wide Web contains different computers, which store documents capable of displaying graphical and textual information. The computers that provide information on the World Wide Web are typically called "websites." A website is defined by an Internet address that has an associated electronic page. The electronic page can be identified by a Uniform Resource Locator (URL). Generally, an electronic page is a document that organizes the presentation of text, graphical images, audio, video, and so forth.

Although the communication link 125 is disclosed in terms of its preferred embodiment, one of ordinary skill in the art will recognize from the disclosure herein that the communication link 125 may include a wide range of

interactive communications links. For example, the communication link 125 may include interactive television networks, telephone networks, wireless data transmission systems, two-way cable systems, customized private or public computer networks, interactive kiosk networks, automatic teller machine networks, direct links, satellite or cellular networks, and the like.

FIGURE 2 illustrates a block diagram of the trust engine 110 of FIGURE 1 according to aspects of an embodiment of the invention. As shown in FIGURE 2, the trust engine 110 includes a transaction engine 205, a depository 210, an authentication engine 215, and a cryptographic engine 220. According to one embodiment of the invention, the trust engine 110 also includes mass storage 225. As further shown in FIGURE 2, the transaction engine 205 communicates with the depository 210, the authentication engine 215, and the cryptographic engine 220, along with the mass storage 225. In addition, the depository 210 communicates with the authentication engine 215, the cryptographic engine 220, and the mass storage 225. Moreover, the authentication engine 215 communicates with the cryptographic engine 220. According to one embodiment of the invention, some or all of the foregoing communications may advantageously comprise the transmission of XML documents to IP addresses that correspond to the receiving device. As mentioned in the foregoing, XML documents advantageously allow designers to create their own customized document tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations. Moreover, some or all of the foregoing communications may include conventional SSL technologies.

According to one embodiment, the transaction engine 205 comprises a data routing device, such as a conventional Web server available from Netscape, Microsoft, Apache, or the like. For example, the Web server may advantageously receive incoming data from the communication link 125. According to one embodiment of the invention, the incoming data is addressed to a front-end security system for the trust engine 110. For example, the front-end security system may advantageously include a firewall, an intrusion detection system searching for known attack profiles, and/or a virus scanner. After clearing the front-end security system, the data is received by the transaction engine 205 and routed to one of the depository 210, the authentication engine 215, the cryptographic engine 220, and the mass storage 225. In addition, the transaction engine 205 monitors incoming data from the authentication engine 215 and cryptographic engine 220, and routes the data to particular systems through the communication link 125. For example, the transaction engine 205 may advantageously route data to the user system 105, the certificate authority 115, or the vendor system 120.

According to one embodiment, the data is routed using conventional HTTP routing techniques, such as, for example, employing URLs or Uniform Resource Indicators (URIs). URIs are similar to URLs, however, URIs typically indicate the source of files or actions, such as, for example, executables, scripts, and the like. Therefore, according to the one embodiment, the user system 105, the certificate authority 115, the vendor system 120, and the components of the trust engine 210, advantageously include sufficient data within communication URLs or URIs for the transaction engine 205 to properly route data throughout the cryptographic system.

Although the data routing is disclosed with reference to its preferred embodiment, a skilled artisan will recognize a wide number of possible data routing solutions or strategies. For example, XML or other data packets may advantageously be unpacked and recognized by their format, content, or the like, such that the transaction engine 205 may properly route data throughout the trust engine 110. Moreover, a skilled artisan will recognize that the data routing may advantageously be adapted to the data transfer protocols conforming to particular network systems, such as, for example, when the communication link 125 comprises a local network.

According to yet another embodiment of the invention, the transaction engine 205 includes conventional SSL encryption technologies, such that the foregoing systems may authenticate themselves, and vise-versa, with transaction engine 205, during particular communications. As will be used throughout this disclosure, the term "½ SSL" refers to communications where a server but not necessarily the client, is SSL authenticated, and the term "FULL SSL" refers to communications where the client and the server are SSL authenticated. When the instant disclosure uses the term "SSL", the communication may comprise ½ or FULL SSL.

As the transaction engine 205 routes data to the various components of the cryptographic system 100, the transaction engine 205 may advantageously create an audit trail. According to one embodiment, the audit trail includes a record of at least the type and format of data routed by the transaction engine 205 throughout the cryptographic system 100. Such audit data may advantageously be stored in the mass storage 225.

FIGURE 2 also illustrates the depository 210. According to one embodiment, the depository 210 comprises one or more data storage facilities, such as, for example, a directory server, a database server, or the like. As shown in FIGURE 2, the depository 210 stores cryptographic keys and enrollment authentication data. The cryptographic keys may advantageously correspond to the trust engine 110 or to users of the cryptographic system 100, such as the user or vendor. The enrollment authentication data may advantageously include data designed to uniquely identify a user, such as, user ID, passwords, answers to questions, biometric data, or the like. This enrollment authentication data may advantageously be acquired at enrollment of a user or another alternative later time. For example, the trust engine 110 may include periodic or other renewal or reissue of enrollment authentication data.

According to one embodiment, the communication from the transaction engine 205 to and from the authentication engine 215 and the cryptographic engine 220 comprises secure communication, such as, for example conventional SSL technology. In addition, as mentioned in the foregoing, the data of the communications to and from the depository 210 may be transferred using URLs, URIs, HTTP or XML documents, with any of the foregoing advantageously having data requests and formats embedded therein.

As mentioned above, the depository 210 may advantageously comprises a plurality of secure data storage facilities. In such an embodiment, the secure data storage facilities may be configured such that a compromise of the security in one individual data storage facility will not compromise the cryptographic keys or the authentication data stored therein. For example, according to this embodiment, the cryptographic keys and the authentication data are mathematically operated on so as to statistically and substantially randomize the data stored in each data storage facility. According to one embodiment, the randomization of the data of an individual data storage facility renders that

data undecipherable. Thus, compromise of an individual data storage facility produces only a randomized undecipherable number and does not compromise the security of any cryptographic keys or the authentication data as a whole.

FIGURE 2 also illustrates the trust engine 110 including the authentication engine 215. According to one embodiment, the authentication engine 215 comprises a data comparator configured to compare data from the transaction engine 205 with data from the depository 210. For example, during authentication, a user supplies current authentication data to the trust engine 110 such that the transaction engine 205 receives the current authentication data. As mentioned in the foregoing, the transaction engine 205 recognizes the data requests, preferably in the URL or URI, and routes the authentication data to the authentication engine 215. Moreover, upon request, the depository 210 forwards enrollment authentication data corresponding to the user to the authentication engine 215. Thus, the authentication engine 215 has both the current authentication data and the enrollment authentication data for comparison.

According to one embodiment, the communications to the authentication engine comprise secure communications, such as, for example, SSL technology. Additionally, security can be provided within the trust engine 110 components, such as, for example, super-encryption using public key technologies. For example, according to one embodiment, the user encrypts the current authentication data with the public key of the authentication engine 215. In addition, the depository 210 also encrypts the enrollment authentication data with the public key of the authentication engine 215. In this way, only the authentication engine's private key can be used to decrypt the transmissions.

As shown in FIGURE 2, the trust engine 110 also includes the cryptographic engine 220. According to one embodiment, the cryptographic engine comprises a cryptographic handling module, configured to advantageously provide conventional cryptographic functions, such as, for example, public-key infrastructure (PKI) functionality. For example, the cryptographic engine 220 may advantageously issue public and private keys for users of the cryptographic system 100. In this manner, the cryptographic keys are generated at the cryptographic engine 220 and forwarded to the depository 210 such that at least the private cryptographic keys are not available outside of the trust engine 110. According to another embodiment, the cryptographic engine 220 randomizes and splits at least the private cryptographic key data, thereby storing only the randomized split data. Similar to the splitting of the enrollment authentication data, the splitting process ensures the stored keys are not available outside the cryptographic engine 220. According to another embodiment, the functions of the cryptographic engine can be combined with and performed by the authentication engine 215.

According to one embodiment, communications to and from the cryptographic engine include secure communications, such as SSL technology. In addition, XML documents may advantageously be employed to transfer data and/or make cryptographic function requests.

FIGURE 2 also illustrates the trust engine 110 having the mass storage 225. As mentioned in the foregoing, the transaction engine 205 keeps data corresponding to an audit trail and stores such data in the mass storage 225.

Similarly, according to one embodiment of the invention, the depository 210 keeps data corresponding to an audit trail and stores such data in the mass storage device 225. The depository audit trail data is similar to that of the transaction engine 205 in that the audit trail data comprises a record of the requests received by the depository 210 and the response thereof. In addition, the mass storage 225 may be used to store digital certificates having the public key of a user contained therein.

Although the trust engine 110 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize in the disclosure herein, a wide number of alternatives for the trust engine 110. For example, the trust engine 110, may advantageously perform only authentication, or alternatively, only some or all of the cryptographic functions, such as data encryption and decryption. According to such embodiments, one of the authentication engine 215 and the cryptographic engine 220 may advantageously be removed, thereby creating a more straightforward design for the trust engine 110. In addition, the cryptographic engine 220 may also communicate with a certificate authority such that the certificate authority is embodied within the trust engine 110. According to yet another embodiment, the trust engine 110 may advantageously perform authentication and one or more cryptographic functions, such as, for example, digital signing.

FIGURE 3 illustrates a block diagram of the transaction engine 205 of FIGURE 2, according to aspects of an embodiment of the invention. According to this embodiment, the transaction engine 205 comprises an operating system 305 having a handling thread and a listening thread. The operating system 305 may advantageously be similar to those found in conventional high volume servers, such as, for example, Web servers available from Apache. The listening thread monitors the incoming communication from one of the communication link 125, the authentication engine 215, and the cryptographic engine 220 for incoming data flow. The handling thread recognizes particular data structures of the incoming data flow, such as, for example, the foregoing data structures, thereby routing the incoming data to one of the communication link 125, the depository 210, the authentication engine 215, the cryptographic engine 220, or the mass storage 225. As shown in FIGURE 3, the incoming and outgoing data may advantageously be secured through, for example, SSL technology.

FIGURE 4 illustrates a block diagram of the depository 210 of FIGURE 2 according to aspects of an embodiment of the invention. According to this embodiment, the depository 210 comprises one or more lightweight directory access protocol (LDAP) servers. LDAP directory servers are available from a wide variety of manufacturers such as Netscape, ISO, and others. FIGURE 4 also shows that the directory server preferably stores data 405 corresponding to the cryptographic keys and data 410 corresponding to the enrollment authentication data. According to one embodiment, the depository 210 comprises a single logical memory structure indexing authentication data and cryptographic key data to a unique user ID. The single logical memory structure preferably includes mechanisms to ensure a high degree of trust, or security, in the data stored therein. For example, the physical location of the depository 210 may advantageously include a wide number of conventional security measures, such as limited employee access, modern surveillance systems, and the like. In addition to, or in lieu of, the physical securities, the computer system or server may advantageously include software solutions to protect the stored data. For example,

the depository 210 may advantageously create and store data 415 corresponding to an audit trail of actions taken. In addition, the incoming and outgoing communications may advantageously be encrypted with public key encryption coupled with conventional SSL technologies.

According to another embodiment, the depository 210 may comprise distinct and physically separated data storage facilities, as disclosed further with reference to FIGURE 7.

FIGURE 5 illustrates a block diagram of the authentication engine 215 of FIGURE 2 according to aspects of an embodiment of the invention. Similar to the transaction engine 205 of FIGURE 3, the authentication engine 215 comprises an operating system 505 having at least a listening and a handling thread of a modified version of a conventional Web server, such as, for example, Web servers available from Apache. As shown in FIGURE 5, the authentication engine 215 includes access to at least one private key 510. The private key 510 may advantageously be used for example, to decrypt data from the transaction engine 205 or the depository 210, which was encrypted with a corresponding public key of the authentication engine 215.

FIGURE 5 also illustrates the authentication engine 215 comprising a comparator 515, a data splitting module 520, and a data assembling module 525. According to the preferred embodiment of the invention, the comparator 515 includes technology capable of comparing potentially complex patterns related to the foregoing biometric authentication data. The technology may include hardware, software, or combined solutions for pattern comparisons, such as, for example, those representing finger print patterns or voice patterns. In addition, according to one embodiment, the comparator 515 of the authentication engine 215 may advantageously compare conventional hashes of documents in order to render a comparison result. According to one embodiment of the invention, the comparator 515 includes the application of heuristics 530 to the comparison. The heuristics 530 may advantageously address circumstances surrounding an authentication attempt, such as, for example, the time of day, IP address or subnet mask, purchasing profile, email address, processor serial number or ID, or the like.

Moreover, the nature of biometric data comparisons may result in varying degrees of confidence being produced from the matching of current biometric authentication data to enrollment data. For example, unlike a traditional password which may only return a positive or negative match, a fingerprint may be determined to be a partial match, e.g. a 90% match, a 75% match, or a 10% match, rather than simply being correct or incorrect. Other biometric identifiers such as voice print analysis or face recognition may share this property of probabilistic authentication, rather than absolute authentication.

When working with such probabilistic authentication or in other cases where an authentication is considered less than absolutely reliable, it is desirable to apply the heuristics 530 to determine whether the level of confidence in the authentication provided is sufficiently high to authenticate the transaction which is being made.

It will sometimes be the case that the transaction at issue is a relatively low value transaction where it is acceptable to be authenticated to a lower level of confidence. This could include a transaction which has a low dollar value associated with it (e.g., a $10 purchase) or a transaction with low risk (e.g., admission to a members-only web site).

Conversely, for authenticating other transactions, it may be desirable to require a high degree of confidence in the authentication before allowing the transaction to proceed. Such transactions may include transactions of large dollar value (e.g., signing a multi-million dollar supply contract) or transaction with a high risk if an improper authentication occurs (e.g., remotely logging onto a government computer).

The use of the heuristics 530 in combination with confidence levels and transactions values may be used as will be described below to allow the comparator to provide a dynamic context-sensitive authentication system.

According to another embodiment of the invention, the comparator 515 may advantageously track authentication attempts for a particular transaction. For example, when a transaction fails, the trust engine 110 may request the user to re-enter his or her current authentication data. The comparator 515 of the authentication engine 215 may advantageously employ an attempt limiter 535 to limit the number of authentication attempts, thereby prohibiting brute-force attempts to impersonate a user's authentication data. According to one embodiment, the attempt limiter 535 comprises a software module monitoring transactions for repeating authentication attempts and, for example, limiting the authentication attempts for a given transaction to three. Thus, the attempt limiter 535 will limit an automated attempt to impersonate an individual's authentication data to, for example, simply three "guesses." Upon three failures, the attempt limiter 535 may advantageously deny additional authentication attempts. Such denial may advantageously be implemented through, for example, the comparator 515 returning a negative result regardless of the current authentication data being transmitted. On the other hand, the transaction engine 205 may advantageously block any additional authentication attempts pertaining to a transaction in which three attempts have previously failed.

The authentication engine 215 also includes the data splitting module 520 and the data assembling module 525. The data splitting module 520 advantageously comprises a software, hardware, or combination module having the ability to mathematically operate on various data so as to substantially randomize and split the data into portions. According to one embodiment, original data is not recreatable from an individual portion. The data assembling module 525 advantageously comprises a software, hardware, or combination module configured to mathematically operate on the foregoing substantially randomized portions, such that the combination thereof provides the original deciphered data. According to one embodiment, the authentication engine 215 employs the data splitting module 520 to randomize and split enrollment authentication data into portions, and employs the data assembling module 525 to reassemble the portions into usable enrollment authentication data.

FIGURE 6 illustrates a block diagram of the cryptographic engine 220 of the trust engine 200 of FIGURE 2 according to aspects of one embodiment of the invention. Similar to the transaction engine 205 of FIGURE 3, the cryptographic engine 220 comprises an operating system 605 having at least a listening and a handling thread of a modified version of a conventional Web server, such as, for example, Web servers available from Apache. As shown in FIGURE 6, the cryptographic engine 220 comprises a data splitting module 610 and a data assembling module 620 that function similar to those of FIGURE 5. However, according to one embodiment, the data splitting module 610 and the data assembling module 620 process cryptographic key data, as opposed to the foregoing enrollment

authentication data. Although, a skilled artisan will recognize from the disclosure herein that the data splitting module 910 and the data splitting module 620 may be combined with those of the authentication engine 215.

The cryptographic engine 220 also comprises a cryptographic handling module 625 configured to perform some or all of a wide number of cryptographic functions. According to one embodiment, the cryptographic handling module 625 may comprise software modules or programs, hardware, or both. According to another embodiment, the cryptographic handling module 625 may perform data comparisons, data hashing, data encryption or decryption, digital signature verification or creation, digital certificate generation, storage, or requests, cryptographic key generation, or the like. Moreover, a skilled artisan will recognize from the disclosure herein that the cryptographic handling module 625 may advantageously comprises a public-key infrastructure, such as Pretty Good Privacy (PGP), an RSA-based public-key system, or a wide number of alternative key management systems. In addition, the cryptographic handling module 625 may perform public-key encryption, symmetric-key encryption, or both. In addition to the foregoing, the cryptographic handling module 625 may include one or more computer programs or modules, hardware, or both, for implementing seamless, transparent, interoperability functions.

A skilled artisan will also recognize from the disclosure herein that the cryptographic functionality may include a wide number of functions generally surrounding cryptographic key management systems.

FIGURE 7 illustrates a simplified block diagram of a depository system 700 according to aspects of an embodiment of the invention. As shown in FIGURE 7, the depository system 700 advantageously comprises multiple data storage facilities, for example, data storage facilities D1, D2, D3, and D4. According to one embodiment of the invention, each of the data storage facilities D1 through D4 may advantageously comprise some or all of the elements disclosed with reference to the depository 210 of FIGURE 4. Similar to the depository 210, the data storage facilities D1 through D4 communicate with the transaction engine 205, the authentication engine 215, and the cryptographic engine 220, preferably through conventional SSL communication links transferring, for example, XML documents. Communications from the transaction engine 205 may advantageously include requests for data, wherein the request is advantageously broadcast to the IP address of each data storage facility D1 through D4. On the other hand, the transaction engine 205 may broadcast requests to particular data storage facilities based on a wide number of criteria, such as, for example, response time, server loads, maintenance schedules, or the like.

In response to requests for data from the transaction engine 205, the depository system 700 advantageously forwards stored data to the authentication engine 215 and the cryptographic engine 220. The respective data assembling modules receive the forwarded data and assemble the data into useable formats. On the other hand, communications from the authentication engine 215 and the cryptographic engine 220 to the data storage facilities D1 through D4 may include the transmission of sensitive data to be stored. For example, according to one embodiment, the authentication engine 215 and the cryptographic engine 220 may advantageously employ their respective data splitting modules to divide sensitive data into undecipherable portions, and then transmit one or more undecipherable portions of the sensitive data to a particular data storage facility.

According to one embodiment, each data storage facility, D1 through D4, comprises a separate and independent storage system, such as, for example, a directory server. According to another embodiment of the invention, the depository system 700 comprises multiple geographically separated independent data storage systems. By distributing the sensitive data into distinct and independent storage facilities D1 through D4, some or all of which may be advantageously geographically separated, the depository system 700 provides redundancy along with additional security measures. For example, according to one embodiment, only data from two of the multiple data storage facilities, D1 through D4, are needed to decipher and reassemble the sensitive data. Thus, as many as two of the four data storage facilities D1 through D4 may be inoperative due to maintenance, system failure, power failure, or the like, without affecting the functionality of the trust engine 110. In addition, because, according to one embodiment, the data stored in each data storage facility is randomized and undecipherable, compromise of any individual data storage facility does not necessarily compromise the sensitive data. Moreover, in the embodiment having geographically separation of the data storage facilities, a compromise of multiple geographically remote facilities becomes increasingly difficult. In fact, even a rouge employee will be greatly challenged to subvert the needed multiple independent geographically remote data storage facilities.

Although the depository system 700 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the depository system 700. For example, the depository system 700 may comprise two or more data storage facilities. In addition, sensitive data may be mathematically operated such that portions from two or more data storage facilities are needed to reassemble and decipher the sensitive data.

As mentioned in the foregoing, the authentication engine 215 and the cryptographic engine 220 each include a data splitting module 520 and 610, respectively, for splitting sensitive data, such as, for example, the authentication data and the cryptographic key data. FIGURE 8 illustrates a flowchart of a data splitting process 800 performed by the data splitting module according to aspects of an embodiment of the invention. As shown in FIGURE 8, the data splitting process 800 begins at STEP 805 when sensitive data "S" is received by the data splitting module of the authentication engine 215 or the cryptographic engine 220. Preferably, in STEP 810, the data splitting module then generates a substantially random number, value, or string or set of bits, "A." For example, the random number A may be generated in a wide number of varying conventional techniques available to one of ordinary skill in the art, for producing high quality random numbers suitable for use in cryptographic applications. In addition, according to one embodiment, the random number A comprises a bit length equal to the bit length of the sensitive data, S.

In addition, in STEP 820 the data splitting process 800 generates another statistically random number "C." According to the preferred embodiment, the generation of the statistically random numbers A and C may advantageously be done in parallel. The data splitting module then combines the numbers A and C with the sensitive data S such that new numbers "B" and "D" are generated. For example, number B may comprise the binary combination of A XOR S and number D may comprise the binary combination of C XOR S. The foregoing combinations preferably occur in STEPS 825 and 830, respectively, and, according to one embodiment, the foregoing combinations

also occur in parallel. The data splitting process 800 then proceeds to STEP 835 where the random numbers A and C and the numbers B and D are paired such that none of the pairings contain sufficient data, by themselves, to reorganize and decipher the original sensitive data S. For example, the numbers may be paired as follows: AC, AD, BC, and BD. According to one embodiment, each of the foregoing pairings is distributed to one of the depositories D1 through D4 of FIGURE 7. According to another embodiment, each of the foregoing pairings is randomly distributed to one of the depositories D1 through D4. For example, during a first data splitting process 800, the pairing AC may be sent to depository D2, through, for example, a random selection of D2's IP address. Then, during a second data splitting process 800, the pairing AC may be sent to depository D4, through, for example, a random selection of D4's IP address.

Based on the foregoing, the data splitting process 800 advantageously places portions of the sensitive data in each of the four data storage facilities D1 through D4, such that no single data storage facility D1 through D4 includes sufficient encrypted data to recreate the original sensitive data S. As mentioned in the foregoing, such randomization of the data into individually unusable encrypted portions increases security and provides for maintained trust in the data even if one of the data storage facilities, D1 through D4, is compromised.

Although the data splitting process 800 is disclosed with reference to its preferred embodiment, the invention is not intended to be limited thereby. Rather a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the data splitting process 800. For example, the data splitting process may advantageously split the data into two numbers, for example, random number A and number B and, randomly distribute A and B through two data storage facilities. Moreover, the data splitting process 800 may advantageously split the data among a wide number of data storage facilities through generation of an additional random numbers.

As mentioned in the foregoing, in order to recreate the sensitive data S, the data portions need to be derandomized and reorganized. This process may advantageously occur in the data assembling modules, 525 and 620, of the authentication engine 215 and the cryptographic engine 220, respectively. The data assembling module, for example, data assembly module 525, receives data portions from the data storage facilities D1 through D4, and reassembles the data into useable form. For example, according to one embodiment where the data splitting module 520 employed the data splitting process 800 of FIGURE 8, the data assembling module 525 uses data portions from at least two of the data storage facilities D1 through D4 to recreate the sensitive data S. For example, the pairings of AC, AD, BC, and BD, were distributed such that any two provide one of A and B, or, C and D. Noting that S = A XOR B or S = C XOR D indicates that when the data assembling module receives one of A and B, or, C and D, the data assembling module 525 can advantageously reassemble the sensitive data S. Thus, the data assembling module 525 may assemble the sensitive data S, when, for example, it receives data portions from at least the first two of the data storage facilities D1 through D4 to respond to an assemble request by the trust engine 110.

Based on the above data splitting and assembling processes, the sensitive data S exists in usable format only in a limited area of the trust engine 110. For example, when the sensitive data S includes enrollment authentication data, usable, nonrandomized enrollment authentication data is available only in the authentication engine 215.

Likewise, when the sensitive data S includes private cryptographic key data, usable, nonrandomized private cryptographic key data is available only in the cryptographic engine 220.

Although the data splitting and assembling processes are disclosed with reference to their preferred embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for splitting and reassembling the sensitive data S. For example, public-key encryption may be used to further secure the data at the data storage facilities D1 through D4.

FIGURE 9A illustrates a data flow of an enrollment process 900 according to aspects of an embodiment of the invention. As shown in FIGURE 9A, the enrollment process 900 begins at STEP 905 when a user desires to enroll with the trust engine 110 of the cryptographic system 100. According to this embodiment, the user system 105 advantageously includes a client-side applet, such as a Java-based, that queries the user to enter enrollment data, such as demographic data and enrollment authentication data. According to one embodiment, the enrollment authentication data includes user ID, password(s), biometric(s), or the like. According to one embodiment, during the querying process, the client-side applet preferably communicates with the trust engine 110 to ensure that a chosen user ID is unique. When the user ID is nonunique, the trust engine 110 may advantageously suggest a unique user ID. The client-side applet gathers the enrollment data and transmits the enrollment data, for example, through and XML document, to the trust engine 110, and in particular, to the transaction engine 205. According to one embodiment, the transmission is encoded with the public key of the authentication engine 215.

According to one embodiment, the user performs a single enrollment during STEP 905 of the enrollment process 900. For example, the user enrolls himself or herself as a particular person, such as Joe User. When Joe User desires to enroll as Joe User, CEO of Mega Corp., then according to this embodiment, Joe User enrolls a second time, receives a second unique user ID and the trust engine 110 does not associate the two identities. According to another embodiment of the invention, the enrollment process 900 provides for multiple user identities for a single user ID. Thus, in the above example, the trust engine 110 will advantageously associate the two identities of Joe User. As will be understood by a skilled artisan from the disclosure herein, a user may have many identities, for example, Joe User the head of household, Joe User the member of the Charitable Foundations, and the like. Even though the user may have multiple identities, according to this embodiment, the trust engine 110 preferably stores only one set of enrollment data. Moreover, users may advantageously add, edit/update, or delete identities as they are needed.

Although the enrollment process 900 is disclosed with reference to its preferred embodiment, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for gathering of enrollment data, and in particular, enrollment authentication data. For example, the applet may be common object model (COM) based applet or the like.

On the other hand, the enrollment process may include graded enrollment. For example, at a lowest level of enrollment, the user may enroll over the communication link 125 without producing documentation as to his or her identity. According to an increased level of enrollment, the user enrolls using a trusted third party, such as a digital notary. For example, and the user may appear in person to the trusted third party, produce credentials such as a birth

certificate, driver's license, military ID, or the like, and the trusted third party may advantageously include, for example, their digital signature in enrollment submission. The trusted third party may include an actual notary, a government agency, such as the Post Office or Department of Motor Vehicles, a human resources person in a large company enrolling an employee, or the like. A skilled artisan will understand from the disclosure herein that a wide number of varying levels of enrollment may occur during the enrollment process 900.

After receiving the enrollment authentication data, at STEP 915, the transaction engine 205, using conventional FULL SSL technology forwards the enrollment authentication data to the authentication engine 215. In STEP 920, the authentication engine 215 decrypts the enrollment authentication data using the private key of the authentication engine 215. In addition, the authentication engine 215 employs the data splitting module to mathematically operate on the enrollment authentication data so as to split the data into at least two independently undecipherable, randomized, numbers. As mentioned in the foregoing, the at least two numbers may comprise a statistically random number and a binary XORed number. In STEP 925, the authentication engine 215 forwards each portion of the randomized numbers to one of the data storage facilities D1 through D4. As mentioned in the foregoing, the authentication engine 215 may also advantageously randomize which portions are transferred to which depositories.

Often during the enrollment process 900, the user will also desire to have a digital certificate issued such that he or she may receive encrypted documents from others outside the cryptographic system 100. As mentioned in the foregoing, the certificate authority 115 generally issues digital certificates according to one or more of several conventional standards. Generally, the digital certificate includes a public key of the user or system, which is known to everyone.

Whether the user requests a digital certificate at enrollment, or at another time, the request is transferred through the trust engine 110 to the authentication engine 215. According to one embodiment, the request includes an XML document having, for example, the proper name of the user. According to STEP 935, the authentication engine 215 transfers the request to the cryptographic engine 220 instructing the cryptographic engine 220 to generate a cryptographic key or key pair.

Upon request, at STEP 935, the cryptographic engine 220 generates at least one cryptographic key. According to one embodiment, the cryptographic handling module 625 generates a key pair, where one key is used as a private key, and one is used as a public key. The cryptographic engine 220 stores the private key and, according to one embodiment, a copy of the public key. In STEP 945, the cryptographic engine 220 transmits a request for a digital certificate to the transaction engine 205. According to one embodiment, the request advantageously includes a standardized request, such as PKCS10, embedded in, for example, and XML document. The request for a digital certificate may advantageously correspond to one or more certificate authorities and the one or more standard formats the certificate authorities require.

In STEP 950 the transaction engine 205 forwards this request to the certificate authority 115, who, in STEP 955, returns a digital certificate. The return digital certificate may advantageously be in a standardized format, such

as PKCS7, or in a proprietary format of one or more of the certificate authorities 115. In STEP 960, the digital certificate is received by the transaction engine 205, and a copy is forwarded to the user and a copy is stored with the trust engine 110. The trust engine 110 stores a copy of the certificate such that the trust engine 110 will not need to rely on the availability of the certificate authority 115. For example, when the user desires to send a digital certificate, or a third party requests the user's digital certificate, the request for the digital certificate is typically sent to the certificate authority 115. However, if the certificate authority 115 is conducting maintenance or has been victim of a failure or security compromise, the digital certificate may not be available.

At any time after issuing the cryptographic keys, the cryptographic engine 220 may advantageously employ the data splitting process 800 described above such that the cryptographic keys are split into independently undecipherable randomized numbers. Similar to the authentication data, at STEP 965 the cryptographic engine 220 transfers the randomized numbers to the data storage facilities D1 through D4.

A skilled artisan will recognize from the disclosure herein that the user may request a digital certificate anytime after enrollment. Moreover, the communications between systems may advantageously include FULL SSL or public-key encryption technologies. Moreover, the enrollment process may issue multiple digital certificates from multiple certificate authorities, including one or more proprietary certificate authorities internal or external to the trust engine 110.

As disclosed in STEPS 935 through 960, one embodiment of the invention includes the request for a certificate that is eventually stored on the trust engine 110. Because, according to one embodiment, the cryptographic handling module 625 issues the keys used by the trust engine 110, each certificate corresponds to a private key. Therefore, the trust engine 110 may advantageously provide for interoperability through monitoring the certificates owned by, or associated with, a user. For example, when the cryptographic engine 220 receives a request for a cryptographic function, the cryptographic handling module 625 may investigate the certificates owned by the requesting user to determine whether the user owns a private key matching the attributes of the request. When such a certificate exists, the cryptographic handling module 625 may use the certificate or the public or private keys associated therewith, to perform the requested function. When such a certificate does not exist, the cryptographic handling module 625 may advantageously and transparently perform a number of actions to attempt to remedy the lack of an appropriate key. For example, FIGURE 9B illustrates a flowchart of an interoperability process 970, which according to aspects of an embodiment of the invention, discloses the foregoing steps to ensure the cryptographic handling module 625 performs cryptographic functions using appropriate keys.

As shown in FIGURE 9B, the interoperability process 970 begins with STEP 972 where the cryptographic handling module 925 determines the type of certificate desired. According to one embodiment of the invention, the type of certificate may advantageously be specified in the request for cryptographic functions, or other data provided by the requestor. According to another embodiment, the certificate type may be ascertained by the data format of the request. For example, the cryptographic handling module 925 may advantageously recognize the request corresponds to a particular type.

According to one embodiment, the certificate type may include one or more algorithm standards, for example, RSA, ELGAMAL, or the like. In addition, the certificate type may include one or more key types, such as symmetric keys, public keys, strong encryption keys such as 256 bit keys, less secure keys, or the like. Moreover, the certificate type may include upgrades or replacements of one or more of the foregoing algorithm standards or keys, one or more message or data formats, one or more data encapsulation or encoding schemes, such as Base 32 or Base 64. The certificate type may also include compatibility with one or more third-party cryptographic applications or interfaces, one or more communication protocols, or one or more certificate standards or protocols. A skilled artisan will recognize from the disclosure herein that other differences may exist in certificate types, and translations to and from those differences may be implemented as disclosed herein.

Once the cryptographic handling module 625 determines the certificate type, the interoperability process 970 proceeds to STEP 974, and determines whether the user owns a certificate matching the type determined in STEP 974. When the user owns a matching certificate, for example, the trust engine 110 has access to the matching certificate through, for example, prior storage thereof, the cryptographic handling module 625 knows that a matching private key is also stored within the trust engine 110. For example, the matching private key may be stored within the depository 210 or depository system 700. The cryptographic handling module 625 may advantageously request the matching private key be assembled from, for example, the depository 210, and then in STEP 976, use the matching private key to perform cryptographic actions or functions. For example, as mentioned in the foregoing, the cryptographic handling module 625 may advantageously perform hashing, hash comparisons, data encryption or decryption, digital signature verification or creation, or the like.

When the user does not own a matching certificate, the interoperability process 970 proceeds to STEP 978 where the cryptographic handling module 625 determines whether the users owns a cross-certified certificate. According to one embodiment, cross-certification between certificate authorities occurs when a first certificate authority determines to trust certificates from a second certificate authority. In other words, the first certificate authority determines that certificates from the second certificate authority meets certain quality standards, and therefore, may be "certified" as equivalent to the first certificate authority's own certificates. Cross-certification becomes more complex when the certificate authorities issue, for example, certificates having levels of trust . For example, the first certificate authority may provide three levels of trust for a particular certificate, usually based on the degree of reliability in the enrollment process, while the second certificate authority may provide seven levels of trust. Cross-certification may advantageously track which levels and which the certificates from the second certificate authority may be substituted for which levels and which certificates from the first. When the foregoing cross-certification is done officially and publicly between two certification authorities, the mapping of certificates and levels to one another is often called "chaining."

According to another embodiment of the invention, the cryptographic handling module 625 may advantageously develop cross-certifications outside those agreed upon by the certificate authorities. For example, the cryptographic handling module 625 may access a first certificate authority's certificate practice statement (CPS), or

other published policy statement, and using, for example, the authentication tokens required by particular trust levels, match the first certificate authority's certificates to those of another certificate authority.

When, in STEP 978, the cryptographic handling module 625 determines that the users owns a cross-certified certificate, the interoperability process 970 proceeds to STEP 976, and performs the cryptographic action or function using the cross-certified public key, private key, or both. Alternatively, when the cryptographic handling module 625 determines that the users does not own a cross-certified certificate, the interoperability process 970 proceeds to STEP 980, where the cryptographic handling module 625 selects a certificate authority that issues the requested certificate type, or a certificate cross-certified thereto. In STEP 982, the cryptographic handling module 625 determines whether the user enrollment authentication data, discussed in the foregoing, meets the authentication requirements of the chosen certificate authority. For example, if the user enrolled over a network by, for example, answering demographic and other questions, the authentication data provided may establish a lower level of trust than a user providing biometric data and appearing before a third-party, such as, for example, a notary. According to one embodiment, the foregoing authentication requirements may advantageously be provided in the chosen authentication authority's CPS.

When the user has provided the trust engine 110 with enrollment authentication data meeting the requirements of chosen certificate authority, the interoperability process 970 proceeds to STEP 984, where the cryptographic handling module 625 acquires the certificate from the chosen certificate authority. According to one embodiment, the cryptographic handling module 625 acquires the certificate by following STEPS 945 through 960 of the enrollment process 900. For example, the cryptographic handling module 625 may advantageously employ one or more public keys from one or more of the key pairs already available to the cryptographic engine 220, to request the certificate from the certificate authority. According to another embodiment, the cryptographic handling module 625 may advantageously generate one or more new key pairs, and use the public keys corresponding thereto, to request the certificate from the certificate authority.

According to another embodiment, the trust engine 110 may advantageously include one or more certificate issuing modules capable of issuing one or more certificate types. According to this embodiment, the certificate issuing module may provide the foregoing certificate. When the cryptographic handling module 625 acquires the certificate, the interoperability process 970 proceeds to STEP 976, and performs the cryptographic action or function using the public key, private key, or both corresponding to the acquired certificate.

When the user, in STEP 982, has not provided the trust engine 110 with enrollment authentication data meeting the requirements of chosen certificate authority, the cryptographic handling module 625 determines, in STEP 986 whether there are other certificate authorities that have different authentication requirements. For example, the cryptographic handling module 625 may look for certificate authorities having lower authentication requirements, but still issue the chosen certificates, or cross-certifications thereof.

When the foregoing certificate authority having lower requirements exists, the interoperability process 970 proceeds to STEP 980 and chooses that certificate authority. Alternatively, when no such certificate authority exists, in STEP 988, the trust engine 110 may request additional authentication tokens from the user. For example, the trust

engine 110 may request new enrollment authentication data comprising, for example, biometric data. Also, the trust engine 110 may request the user appear before a trusted third party and provide appropriate authenticating credentials, such as, for example, appearing before a notary with a drivers license, social security card, bank card, birth certificate, military ID, or the like. When the trust engine 110 receives updated authentication data, the interoperability process 970 proceeds to Step 984 and acquires the foregoing chosen certificate.

Through the foregoing interoperability process 970, the cryptographic handling module 625 advantageously provides seamless, transparent, translations and conversions between differing cryptographic systems. A skilled artisan will recognize from the disclosure herein, a wide number of advantages and implementations of the foregoing interoperable system. For example, the foregoing STEP 986 of the interoperability process 970 may advantageously include aspects of trust arbitrage, discussed in further detail below, where the certificate authority may under special circumstances accept lower levels of cross-certification. In addition, the interoperability process 970 may include ensuring interoperability between and employment of standard certificate revocations, such as employing certificate revocation lists (CRL), online certificate status protocols (OCSP), or the like.

FIGURE 10 illustrates a data flow of an authentication process 1000 according to aspects of an embodiment of the invention. According to one embodiment, the authentication process 1000 includes gathering current authentication data from a user and comparing that to the enrollment authentication data of the user. For example, the authentication process 1000 begins at STEP 1005 where a user desires to perform a transaction with, for example, a vendor. Such transactions may include, for example, selecting a purchase option, requesting access to a restricted area or device of the vendor system 120, or the like. At STEP 1010, a vendor provides the user with a transaction ID and an authentication request. The transaction ID may advantageously include a 192 bit quantity having a 32 bit timestamp concatenated with a 128 bit random quantity, or a "nonce," concatenated with a 32 bit vendor specific constant. Such a transaction ID uniquely identifies the transaction such that copycat transactions can be refused by the trust engine 110.

The authentication request may advantageously include what level of authentication is needed for a particular transaction. For example, the vendor may specify a particular level of confidence that is required for the transaction at issue. If authentication cannot be made to this level of confidence, as will be discussed below, the transaction will not occur without either further authentication by the user to raise the level of confidence, or a change in the terms of the authentication between the vendor and the server. These issues are discussed more completely below.

According to one embodiment, the transaction ID and the authentication request may advantageously generated by a vendor-side applet or other software program. In addition, the transmission of the transaction ID and authentication data may include one or more XML documents encrypted using conventional SSL technology, such as, for example, ½ SSL, or, in other words vendor-side authenticated SSL.

After the user system 105 receives the transaction ID and authentication request, the user system 105 gathers the current authentication data, potentially including current biometric information, from the user. The user

system 105, at STEP 1015, encrypts at least the current authentication data " B' " and the transaction ID, with the public key of the authentication engine 215, and transfers that data to the trust engine 110. The transmission preferably comprises XML documents encrypted with at least conventional ½ SSL technology. In STEP 1020, the transaction engine 205 receives the transmission, preferably recognizes the data format or request in the URL or URI, and forwards the transmission to the authentication engine 215.

During STEPS 1015 and 1020, the vendor system 120, at STEP 1025, forwards the transaction ID and the authentication request to the trust engine 110, using the preferred FULL SSL technology. This communication may also include a vendor ID, although vendor identification may also be communicated through a non-random portion of the transaction ID. At STEPS 1030 and 1035, the transaction engine 205 receives the communication, creates a record in the audit trail, and generates a request for the user's enrollment authentication data to be assembled from the data storage facilities D1 through D4. At STEP 1040, the depository system 700 transfers the portions of the enrollment authentication data corresponding to the user to the authentication engine 215. At STEP 1045, the authentication engine 215 decrypts the transmission using its private key and compares the enrollment authentication data to the current authentication data provided by the user.

The comparison of STEP 1045 may advantageously apply heuristical context sensitive authentication, as referred to in the forgoing, and discussed in further detail below. For example, if the biometric information received does not match perfectly, a lower confidence match results. In particular embodiments, the level of confidence of the authentication is balanced against the nature of the transaction and the desires of both the user and the vendor. Again, this is discussed in greater detail below.

At STEP 1050, the authentication engine 215 fills in the authentication request with the result of the comparison of STEP 1045. According to one embodiment of the invention, the authentication request is filled with a YES/NO or TRUE/FALSE result of the authentication process 1000. In STEP 1055 the filled-in authentication request is returned to the vendor for the vendor to act upon, for example, allowing the user to complete the transaction that initiated the authentication request. According to one embodiment, a confirmation message is passed to the user.

Based on the foregoing, the authentication process 1000 advantageously keeps sensitive data secure and produces results configured to maintain the integrity of the sensitive data. For example, the sensitive data is assembled only inside the authentication engine 215. For example, the enrollment authentication data is undecipherable until it is assembled in the authentication engine 215 by the data assembling module, and the current authentication data is undecipherable until it is unwrapped by the conventional SSL technology and the private key of the authentication engine 215. Moreover, the authentication result transmitted to the vendor does not include the sensitive data, and the user may not even know whether he or she produced valid authentication data.

Although the authentication process 1000 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the authentication process 1000. For example, the vendor may advantageously be replaced by almost any requesting application, even those residing with the user system 105. For

example, a client application, such as Microsoft Word, may use an application program interface (API) or a cryptographic API (CAPI) to request authentication before unlocking a document. Alternatively, a mail server, a network, a cellular phone, a personal or mobile computing device, a workstation, or the like, may all make authentication requests that can be filled by the authentication process 1000. In fact, after providing the foregoing trusted authentication process 1000, the requesting application or device may provide access to or use of a wide number of electronic or computer devices or systems.

Moreover, the authentication process 1000 may employ a wide number of alternative procedures in the event of authentication failure. For example, authentication failure may maintain the same transaction ID and request that the user reenter his or her current authentication data. As mentioned in the foregoing, use of the same transaction ID allows the comparator of the authentication engine 215 to monitor and limit the number of authentication attempts for a particular transaction, thereby creating a more secure cryptographic system 100.

In addition, the authentication process 1000 may be advantageously be employed to develop elegant single sign-on solutions, such as, unlocking a sensitive data vault. For example, successful or positive authentication may provide the authenticated user the ability to automatically access any number of passwords for an almost limitless number of systems and applications. For example, authentication of a user may provide the user access to password, login, financial credentials, or the like, associated with multiple online vendors, a local area network, various personal computing devices, Internet service providers, auction providers, investment brokerages, or the like. By employing a sensitive data vault, users may choose truly large and random passwords because they no longer need to remember them through association. Rather, the authentication process 1000 provides access thereto. For example, a user may choose a random alphanumeric string that is twenty plus digits in length rather than something associated with a memorable data, name, etc.

According to one embodiment, a sensitive data vault associated with a given user may advantageously be stored in the data storage facilities of the depository 210, or split and stored in the depository system 700. According to this embodiment, after positive user authentication, the trust engine 110 serves the requested sensitive data, such as, for example, to the appropriate password to the requesting application. According to another embodiment, the trust engine 110 may include a separate system for storing the sensitive data vault. For example, the trust engine 110 may include a stand-alone software engine implementing the data vault functionality and figuratively residing "behind" the foregoing front-end security system of the trust engine 110. According to this embodiment, the software engine serves the requested sensitive data after the software engine receives a signal indicating positive user authentication from the trust engine 110.

In yet another embodiment, the data vault may be implemented by a third-party system. Similar to the software engine embodiment, the third-party system may advantageously serve the requested sensitive data after the third-party system receives a signal indicating positive user authentication from the trust engine 110. According to yet another embodiment, the data vault may be implemented on the user system 105. A user-side software engine may

advantageously serve the foregoing data after receiving a signal indicating positive user authentication from the trust engine 110.

Although the foregoing data vaults are disclosed with reference to alternative embodiments, a skilled artisan will recognize from the disclosure herein, a wide number of additional implementations thereof. For example, a particular data vault may include aspects from some or all of the foregoing embodiments. In addition, any of the foregoing data vaults may employ one or more authentication requests at varying times. For example, any of the data vaults may require authentication every one or more transactions, periodically, every one ore more sessions, every access to one or more Webpages or Websites, at one or more other specified intervals, or the like.

FIGURE 11 illustrates a data flow of a signing process 1100 according to aspects of an embodiment of the invention. As shown in FIGURE 11, the signing process 1100 includes steps similar to those of the authentication process 1000 described in the foregoing with reference to FIGURE 10. According to one embodiment of the invention, the signing process 1100 first authenticates the user and then performs one or more of several digital signing functions as will be discussed in further detail below. According to another embodiment, the signing process 1100 may advantageously store data related thereto, such as hashes of messages or documents, or the like. This data may advantageously be used in an audit or any other event, such as for example, when a participating party attempts to repudiate a transaction.

As shown in FIGURE 11, during the authentication steps, the user and vendor may advantageously agree on a message, such as, for example, a contract. During signing, the signing process 1100 advantageously ensures that the contract signed by the user is identical to the contract supplied by the vendor. Therefore, according to one embodiment, during authentication, the vendor and the user include a hash of their respective copies of the message or contract, in the data transmitted to the authentication engine 215. By employing only a hash of a message or contract, the trust engine 110 may advantageously store a significantly reduced amount of data, providing for a more efficient and cost effective cryptographic system. In addition, the stored hash may be advantageously compared to a hash of a document in question to determine whether the document in question matches one signed by any of the parties. The ability to determine whether the document is identical to one relating to a transaction provides for additional evidence that can be used against a claim for repudiation by a party to a transaction.

In STEP 1103, the authentication engine 215 assembles the enrollment authentication data and compares it to the current authentication data provided by the user. When the comparator of the authentication engine 215 indicates that the enrollment authentication data matches the current authentication data, the comparator of the authentication engine 215 also compares the hash of the message supplied by the vendor to the hash of the message supplied by the user. Thus, the authentication engine 215 advantageously ensures that the message agreed to by the user is identical to that agreed to by the vendor.

In STEP 1105, the authentication engine 215 transmits a digital signature request to the cryptographic engine 220. According to one embodiment of the invention, the request includes a hash of the message or contract. However, a skill artisan will recognize from the disclosure herein that the cryptographic engine 220 may encrypt

virtually any text to form the desired digital signature. Returning to STEP 1105, the digital signature request preferably comprises an XML document communicated through conventional SSL technologies.

In STEP 1110, the authentication engine 215 transmits a request to each of the data storage facilities D1 through D4, such that each of the data storage facilities D1 through D4 transmit their respective portion of the cryptographic key or keys corresponding to a signing party. According to another embodiment, the cryptographic engine 220 employs some or all of the steps of the interoperability process 970 discussed in the foregoing, such that the cryptographic engine 220 first determines the appropriate key or keys to request from the depository 210 or the depository system 700 for the signing party, and takes actions to provide appropriate matching keys. According to still another embodiment, the authentication engine 215 or the cryptographic engine 220 may advantageously request one or more of the keys associated with the signing party and stored in the depository 210 or depository system 700.

According to one embodiment, the signing party includes one or both the user and the vendor. In such case, the authentication engine 215 advantageously requests the cryptographic keys corresponding to the user and/or the vendor. According to another embodiment, the signing party includes the trust engine 110. In this embodiment, the trust engine 110 is certifying that the authentication process 1000 properly authenticated the user, vendor, or both. Therefore, the authentication engine 215 requests the cryptographic key of the trust engine 110, such as, for example, the key belonging to the cryptographic engine 220, to perform the digital signature. According to another embodiment, the trust engine 110 performs a digital notary-like function. In this embodiment, the signing party includes the user, vendor, or both, along with the trust engine 110. Thus, the trust engine 110 provides the digital signature of the user and/or vendor, and then indicates with its own digital signature that the user and/or vendor were properly authenticated. In this embodiment, the authentication engine 215 may advantageously request assembly of the cryptographic keys corresponding to the user, the vendor, or both. According to another embodiment, the authentication engine 215 may advantageously request assembly of the cryptographic keys corresponding to the trust engine 110.

According to another embodiment, the trust engine 110 performs power of attorney-like functions. For example, the trust engine 110 may digitally sign the message on behalf of a third party. In such case, the authentication engine 215 requests the cryptographic keys associated with the third party. According to this embodiment, the signing process 1100 may advantageously include authentication of the third party, before allowing power of attorney-like functions. In addition, the authentication process 1000 may include a check for third party constraints, such as, for example, business logic or the like dictating when and in what circumstances a particular third-party's signature may be used.

Based on the foregoing, in STEP 1110, the authentication engine requested the cryptographic keys from the data storage facilities D1 through D4 corresponding to the signing party. In STEP 1115, the data storage facilities D1 through D4 transmit their respective portions of the cryptographic key corresponding to the signing party to the cryptographic engine 220. According to one embodiment, the foregoing transmissions include SSL technologies.

According to another embodiment, the foregoing transmissions may advantageously be super-encrypted with the public key of the cryptographic engine 220.

In STEP 1120, the cryptographic engine 220 assembles the foregoing cryptographic keys of the signing party and encrypts the message therewith, thereby forming the digital signature(s). In STEP 1125 of the signing process 1100, the cryptographic engine 220 transmits the digital signature(s) to the authentication engine 215. In STEP 1130, the authentication engine 215 transmits the filled-in authentication request along with a copy of the hashed message and the digital signature(s) to the transaction engine 205. In STEP 1135, the transaction engine 205 transmits a receipt comprising the transaction ID, an indication of whether the authentication was successful, and the digital signature(s), to the vendor. According to one embodiment, the foregoing transmission may advantageously include the digital signature of the trust engine 110. For example, the trust engine 110 may encrypt the hash of the receipt with its private key, thereby forming a digital signature to be attached to the transmission to the vendor.

According to one embodiment, the transaction engine 205 also transmits a confirmation message to the user.

Although the signing process 1100 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the signing process 1100. For example, the vendor may be replaced with a user application, such as an email application. For example, the user may wish to digitally sign a particular email with his or her digital signature. In such an embodiment, the transmission throughout the signing process 1100 may advantageously include only one copy of a hash of the message. Moreover, a skilled artisan will recognize from the disclosure herein that a wide number of client applications may request digital signatures. For example, the client applications may comprise word processors, spreadsheets, emails, voicemail, access to restricted system areas, or the like.

In addition, a skilled artisan will recognize from the disclosure herein that STEPS 1105 through 1120 of the signing process 1100 may advantageously employ some or all of the steps of the interoperability process 970 of FIGURE 9B, thereby providing interoperability between differing cryptographic systems that may, for example, need to process the digital signature under differing signature types.

FIGURE 12 illustrates a data flow of an encryption/decryption process 1200 according to aspects of an embodiment of the invention. As shown in FIGURE 12, the decryption process 1200 begins by authenticating the user using the authentication process 1000. According to one embodiment, the authentication process 1000 includes in the authentication request, a synchronous session key. For example, in conventional PKI technologies, it is understood by skilled artisans that encrypting or decrypting data using public and private keys is mathematically intensive and may require significant system resources. However, in symmetric key cryptographic systems, or systems where the sender and receiver of a message share a single common key that is used to encrypt and decrypt a message, the mathematical operations are significantly simpler and faster. Thus, in the conventional PKI technologies, the sender of a message will generate synchronous session key, and encrypt the message using the simpler, faster symmetric key system. Then, the sender will encrypt the session key with the public key of the receiver. The encrypted session key will be

attached to the synchronously encrypted message and both data are sent to the receiver. The receiver uses his or her private key to decrypt the session key, and then uses the session key to decrypt the message. Based on the foregoing, the simpler and faster symmetric key system is used for the majority of the encryption/decryption processing. Thus, in the decryption process 1200, the decryption advantageously assumes that a synchronous key has been encrypted with the public key of the user. Thus, as mentioned in the foregoing, the encrypted session key is included in the authentication request.

Returning to the decryption process 1200, after the user has been authenticated in STEP 1205, the authentication engine 215 forwards the encrypted session key to the cryptographic engine 220. In STEP 1210, the authentication engine 215 forwards a request to each of the data storage facilities, D1 through D4, requesting the cryptographic key data of the user. In STEP 1215, each data storage facility, D1 through D4, transmits their respective portion of the cryptographic key to the cryptographic engine 220. According to one embodiment, the foregoing transmission is encrypted with the public key of the cryptographic engine 220.

In STEP 1220 of the decryption process 1200, the cryptographic engine 220 assembles the cryptographic key and decrypts the session key therewith. In STEP 1225, the cryptographic engine forwards the session key to the authentication engine 215. In STEP 1227, the authentication engine 215 fills in the authentication request including the decrypted session key, and transmits the filled-in authentication request to the transaction engine 205. In STEP 1230, the transaction engine 205 forwards the authentication request along with the session key to the requesting application or vendor. Then, according to one embodiment, the requesting application or vendor uses the session key to decrypt the encrypted message.

Although the decryption process 1200 is disclosed with reference to its preferred and alternative embodiments, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the decryption process 1200. For example, the decryption process 1200 may forego synchronous key encryption and rely on full public-key technology. In such an embodiment, the requesting application may transmit the entire message to the cryptographic engine 220, or, may employ some type of compression or reversible hash in order to transmit the message to the cryptographic engine 220. A skilled artisan will also recognize from the disclosure herein that the foregoing communications may advantageously include XML documents wrapped in SSL technology.

The encryption/decryption process 1200 also provides for encryption of documents or other data. Thus, in STEP 1235, a requesting application or vendor may advantageously transmit to the transaction engine 205 of the trust engine 110, a request for the public key of the user. The requesting application or vendor makes this request because the requesting application or vendor uses the public key of the user, for example, to encrypt the session key that will be used to encrypt the document or message. As mentioned in the enrollment process 900, the transaction engine 205 stores a copy of the digital certificate of the user, for example, in the mass storage 225. Thus, in STEP 1240 of the encryption process 1200, the transaction engine 205 requests the digital certificate of the user from the mass storage 225. In STEP 1245, the mass storage 225 transmits the digital certificate corresponding to the user, to the transaction engine 205. In STEP 1250, the transaction engine 205 transmits the digital certificate to the

requesting application or vendor. According to one embodiment, the encryption portion of the encryption process 1200 does not include the authentication of a user. This is because the requesting vendor needs only the public key of the user, and is not requesting any sensitive data.

A skilled artisan will recognize from the disclosure herein that if a particular user does not have a digital certificate, the trust engine 110 may employ some or all of the enrollment process 900 in order to generate a digital certificate for that particular user. Then, the trust engine 110 may initiate the encryption/decryption process 1200 and thereby provide the appropriate digital certificate. In addition, a skilled artisan will recognize from the disclosure herein that STEPS 1220 and 1235 through 1250 of the encryption/decryption process 1200 may advantageously employ some or all of the steps of the interoperability process of FIGURE 9B, thereby providing interoperability between differing cryptographic systems that may, for example, need to process the encryption.

FIGURE 13 illustrates a simplified block diagram of a trust engine system 1300 according to aspects of yet another embodiment of the invention. As shown in FIGURE 13, the trust engine system 1300 comprises a plurality of distinct trust engines 1305, 1310, 1315, and 1320, respectively. To facilitate a more complete understanding of the invention, FIGURE 13 illustrates each trust engine, 1305, 1310, 1315, and 1320 as having a transaction engine, a depository, and an authentication engine. However, a skilled artisan will recognize that each transaction engine may advantageously comprise some or all of the elements and communication channels disclosed with reference to FIGURES 1-8. For example, one embodiment may advantageously include trust engines having transaction engines, depositories, and cryptographic servers.

According to one embodiment of the invention, each of the trust engines 1305, 1310, 1315 and 1320 are geographically separated, such that, for example, the trust engine 1305 may reside in a first location, the trust engine 1310 may reside in a second location, the trust engine 1315 may reside in a third location, and the trust engine 1320 may reside in a fourth location. The foregoing geographic separation advantageously decreases system response time while increasing the security of the overall trust engine system 1300.

For example, when a user logs onto the cryptographic system 100, the user may be nearest the first location and may desire to be authenticated. As described with reference to FIGURE 10, to be authenticated, the user provides current authentication data, such as a biometric or the like, and the current authentication data is compared to that user's enrollment authentication data. Therefore, according to one example, the user advantageously provides current authentication data to the geographically nearest trust engine 1305. The transaction engine 1321 of the trust engine 1305 then forwards the current authentication data to the authentication engine 1322 also residing at the first location. According to another embodiment, the transaction engine 1321 forwards the current authentication data to one or more of the authentication engines of the trust engines 1310, 1315, or 1320.

The transaction engine 1321 also requests the assembly of the enrollment authentication data from the depositories of, for example, each of the trust engines, 1305 through 1320. According to this embodiment, each depository provides its portion of the enrollment authentication data to the authentication engine 1322 of the trust engine 1305. The authentication engine 1322 then employs the encrypted data portions from, for example, the first

two depositories to respond, and assembles the enrollment authentication data into deciphered form. The authentication engine 1322 compares the enrollment authentication data with the current authentication data and returns an authentication result to the transaction engine 1321 of the trust engine 1305.

Based on the above, the trust engine system 1300 employs the nearest one of a plurality of geographically separated trust engines, 1305 through 1320, to perform the authentication process. According to one embodiment of the invention, the routing of information to the nearest transaction engine may advantageously be performed at client-side applets executing on one or more of the user system 105, vendor system 120, or certificate authority 115. According to an alternative embodiment, a more sophisticated decision process may be employed to select from the trust engines 1305 through 1320. For example, the decision may be based on the availability, operability, speed of connections, load, performance, geographic proximity, or a combination thereof, of a given trust engine.

In this way, the trust engine system 1300 lowers its response time while maintaining the security advantages associated with geographically remote data storage facilities, such as those discussed with reference to FIGURE 7 where each data storage facility stores randomized portions of sensitive data. For example, a security compromise at, for example, the depository 1325 of the trust engine 1315 does not necessarily compromise the sensitive data of the trust engine system 1300. This is because the depository 1325 contains only non-decipherable randomized data that, without more, is entirely useless.

According to another embodiment, the trust engine system 1300 may advantageously include multiple cryptographic engines arranged similar to the authentication engines. The cryptographic engines may advantageously perform cryptographic functions such as those disclosed with reference to FIGURES 1-8. According to yet another embodiment, the trust engine system 1300 may advantageously replace the multiple authentication engines with multiple cryptographic engines, thereby performing cryptographic functions such as those disclosed with reference to FIGURES 1-8. According to yet another embodiment of the invention, the trust engine system 1300 may replace each multiple authentication engine with an engine having some or all of the functionality of the authentication engines, cryptographic engines, or both, as disclosed in the foregoing.

Although the trust engine system 1300 is disclosed with reference to its preferred and alternative embodiments, a skilled artisan will recognize that the trust engine system 1300 may comprise portions of trust engines 1305 through 1320. For example, the trust engine system 1300 may include one or more transaction engines, one or more depositories, one or more authentication engines, or one or more cryptographic engines.

FIGURE 14 illustrates a simplified block diagram of a trust engine system 1400 according to aspects of yet another embodiment of the invention. As shown in FIGURE 14, the trust engine system 1400 includes multiple trust engines 1405, 1410, 1415 and 1420. According to one embodiment, each of the trust engines 1405, 1410, 1415 and 1420, comprise some or all of the elements of trust engine 110 disclosed with reference to FIGURES 1-8. According to this embodiment, when the client side applets of the user system 105, the vendor system 120, or the certificate authority 115, communicate with the trust engine system 1400, those communications are sent to the IP address of each of the trust engines 1405 through 1420. Further, each transaction engine of each of the trust engines, 1405,

1410, 1415, and 1420, behaves similar to the transaction engine 1321 of the trust engine 1305 disclosed with reference to FIGURE 13. For example, during an authentication process, each transaction engine of each of the trust engines 1405, 1410, 1415, and 1420 transmits the current authentication data to their respective authentication engines and transmits a request to assemble the randomized data stored in each of the depositories of each of the trust engines 1405 through 1420. FIGURE 14 does not illustrate all of these communications, as such illustration would become overly complex. Continuing with the authentication process, each of the depositories then communicates its portion of the randomized data to each of the authentication engines of the each of the trust engines 1405 through 1420. Each of the authentication engines of the each of the trust engines employs its comparator to determine whether the current authentication data matches the enrollment authentication data provided by the depositories of each of the trust engines 1405 through 1420. According to this embodiment, the result of the comparison by each of the authentication engines is then transmitted to a redundancy module of the other three trust engines. For example, the result of the authentication engine from the trust engine 1405 is transmitted to the redundancy modules of the trust engines 1410, 1415, and 1420. Thus, the redundancy module of the trust engine 1405 likewise receives the result of the authentication engines from the trust engines 1410, 1415, and 1420.

FIGURE 15 illustrates a block diagram of the redundancy module of FIGURE 14. The redundancy module comprises a comparator configured to receive the authentication result from three authentication engines and transmit that result to the transaction engine of the fourth trust engine. The comparator compares the authentication result form the three authentication engines, and if two of the results agree, the comparator concludes that the authentication result should match that of the two agreeing authentication engines. This result is then transmitted back to the transaction engine corresponding to the trust engine not associated with the three authentication engines.

Based on the foregoing, the redundancy module determines an authentication result from data received from authentication engines that are preferably geographically remote from the trust engine of that the redundancy module. By providing such redundancy functionality, the trust engine system 1400 ensures that a compromise of the authentication engine of one of the trust engines 1405 through 1420, is insufficient to compromise the authentication result of the redundancy module of that particular trust engine. A skilled artisan will recognize that redundancy module functionality of the trust engine system 1400 may also be applied to the cryptographic engine of each of the trust engines 1405 through 1420. However, such cryptographic engine communication was not shown in FIGURE 14 to avoid complexity. Moreover, a skilled artisan will recognize a wide number of alternative authentication result conflict resolution algorithms for the comparator of FIGURE 15.

According to yet another embodiment of the invention, the trust engine system 1400 may advantageously employ the redundancy module during cryptographic comparison steps. For example, some or all of the foregoing redundancy module disclosure with reference to FIGURES 14 and 15 may advantageously be implemented during a hash comparison of documents provided by one or more parties during a particular transaction.

Although the foregoing invention has been described in terms of certain preferred and alternative embodiments, other embodiments will be apparent to those of ordinary skill in the art from the disclosure herein. For

example, the trust engine 110 may issue short-term certificates, where the private cryptographic key is released to the user for a predetermined period of time. For example, current certificate standards include a validity field that can be set to expire after a predetermined amount of time. Thus, the trust engine 110 may release a private key to a user where the private key would be valid for, for example, 24 hours. According to such an embodiment, the trust engine 110 may advantageously issue a new cryptographic key pair to be associated with a particular user and then release the private key of the new cryptographic key pair. Then, once the private cryptographic key is released, the trust engine 110 immediately expires any internal valid use of such private key, as it is no longer securable by the trust engine 110.

In addition, a skilled artisan will recognize that the cryptographic system 100 or the trust engine 110 may include the ability to recognize types of devices, such as a laptop, a cell phone, a network or the like. According to one embodiment, such recognition may come from data supplied in the request for a particular service, such as, a request for authentication leading to access or use, a request for cryptographic functionality, or the like. According to one embodiment, the foregoing request may include a unique device identifier, such as, for example, a processor ID. Alternatively, the request may include data in a particular recognizable data format. For example, mobile and satellite phones often do not include the processing power for full X509.v3 heavy encryption certificates, and therefore do not request them. According to this embodiment, the trust engine 110 may recognize the type of data format presented, and respond only in kind.

In an additional aspect of the system described above, context sensitive authentication can be provided using various techniques as will be described below. Context sensitive authentication, for example as shown in FIGURE 16, provides the possibility of evaluating not only the actual data which is sent by the user when attempting to authenticate himself, but also the circumstances surrounding the generation and delivery of that data. Such techniques may also support transaction specific trust arbitrage between the user and trust engine 110 or between the vendor and trust engine 110, as will be described below.

As discussed above, authentication is the process of proving that a user is who he says he is. Generally, authentication requires demonstrating some fact to an authentication authority. The trust engine 110 of the present invention represents the authority to which a user must authenticate himself. The user must demonstrate to the trust engine 110 that he is who he says he is by either: knowing something that only the user should know (knowledge-based authentication), having something that only the user should have (token-based authentication), or by being something that only the user should be (biometric-based authentication).

Examples of knowledge-based authentication include without limitation a password, PIN number, or lock combination. Examples of token-based authentication include without limitation a house key, a physical credit card, a driver's license, or a particular phone number. Examples of biometric-based authentication include without limitation a fingerprint, a voice analysis, or a retinal scan.

Each type of authentication has particular advantages and disadvantages, and each provides a different level of security. For example, it is generally harder to create a false fingerprint that matches someone else's than it is to

overhear someone's password and repeat it. Each type of authentication also requires a different type of data to be known to the authenticating authority in order to verify someone using that form of authentication.

As used herein, "authentication" will refer broadly to the overall process of verifying someone's identity to be who he says he is. An "authentication technique" will refer to a particular type of authentication based upon a particular piece of knowledge, physical token, or biometric reading. "Authentication data" refers to information which is sent to or otherwise demonstrated to an authentication authority in order to establish identity. "Enrollment data" will refer to the data which is initially submitted to an authentication authority in order to establish a baseline for comparison with authentication data. An "authentication instance" will refer to the data associated with an attempt to authenticate by an authentication technique.

The internal protocols and communications involved in the process of authenticating a user is described with reference to FIGURE 10 above. The part of this process within which the context sensitive authentication takes place occurs within the comparison step shown as STEP 1045 of FIGURE 10. This step takes place within the authentication engine 215 and involves assembling the enrollment data 410 retrieved from the depository 210 and comparing the authentication data provided by the user to it. One particular embodiment of this process is shown in FIGURE 16 and described below.

The current authentication data provided by the user and the enrollment data retrieved from the depository 210 are received by the authentication engine 215 in STEP 1600 of FIGURE 16. Both of these sets of data may contain data which is related to separate techniques of authentication. The authentication engine 215 separates the authentication data associated with each individual authentication instance in STEP 1605. This is necessary so that the authentication data is compared with the appropriate subset of the enrollment data for the user (e.g. fingerprint authentication data should be compared with fingerprint enrollment data, rather than password enrollment data).

Generally, authenticating a user involves one or more individual authentication instances, depending on which authentication techniques are available to the user. These methods are limited by the enrollment data which were provided by the user during his enrollment process (if the user did not provide a retinal scan when enrolling, he will not be able to authenticate himself using a retinal scan), as well as the means which may be currently available to the user (e.g. if the user does not have a fingerprint reader at his current location, fingerprint authentication will not be practical). In some cases, a single authentication instance may be sufficient to authenticate a user; however, in certain circumstances a combination of multiple authentication instances may be used in order to more confidently authenticate a user for a particular transaction.

Each authentication instance consists of data related to a particular authentication technique (e.g. fingerprint, password, smart card, etc.) and the circumstances which surround the capture and delivery of the data for that particular technique. For example, a particular instance of attempting to authenticate via password will generate not only the data related to the password itself, but also circumstantial data, known as "metadata", related to that password attempt. This circumstantial data includes information such as: the time at which the particular authentication instance took place, the network address from which the authentication information was delivered, as

well as any other information as is known to those of skill in the art which may be determined about the origin of the authentication data (the type of connection, the processor serial number, etc.).

In many cases, only a small amount of circumstantial metadata will be available. For example, if the user is located on a network which uses proxies or network address translation or another technique which masks the address of the originating computer, only the address of the proxy or router may be determined. Similarly, in many cases information such as the processor serial number will not be available because of either limitations of the hardware or operating system being used, disabling of such features by the operator of the system, or other limitations of the connection between the user's system and the trust engine 110.

As shown in FIGURE 16, once the individual authentication instances represented within the authentication data are extracted and separated in STEP 1605, the authentication engine 215 evaluates each instance for its reliability in indicating that the user is who he claims to be. The reliability for a single authentication instance will generally be determined based on several factors. These may be grouped as factors relating to the reliability associated with the authentication technique, which are evaluated in STEP 1610, and factors relating to the reliability of the particular authentication data provided, which are evaluated in STEP 1615. The first group includes without limitation the inherent reliability of the authentication technique being used, and the reliability of the enrollment data being used with that method. The second group includes without limitation the degree of match between the enrollment data and the data provided with the authentication instance, and the metadata associated with that authentication instance. Each of these factors may vary independently of the others.

The inherent reliability of an authentication technique is based on how hard it is for an imposter to provide someone else's correct data, as well as the overall error rates for the authentication technique. For passwords and knowledge based authentication methods, this reliability is often fairly low because there is nothing that prevents someone from revealing their password to another person and for that second person to use that password. Even a more complex knowledge based system may have only moderate reliability since knowledge may be transferred from person to person fairly easily. Token based authentication, such as having a proper smart card or using a particular terminal to perform the authentication, is similarly of low reliability used by itself, since there is no guarantee that the right person is in possession of the proper token.

However, biometric techniques are more inherently reliable because it is generally difficult to provide someone else with the ability to use your fingerprints in a convenient manner, even intentionally. Because subverting biometric authentication techniques is more difficult, the inherent reliability of biometric methods is generally higher than that of purely knowledge or token based authentication techniques. However, even biometric techniques may have some occasions in which a false acceptance or false rejection is generated. These occurrences may be reflected by differing reliabilities for different implementations of the same biometric technique. For example, a fingerprint matching system provided by one company may provide a higher reliability than one provided by a different company because one uses higher quality optics or a better scanning resolution or some other improvement which reduces the occurrence of false acceptances or false rejections.

Note that this reliability may be expressed in different manners. The reliability is desirably expressed in some metric which can be used by the heuristics 530 and algorithms of the authentication engine 215 to calculate the confidence level of each authentication. One preferred mode of expressing these reliabilities is as a percentage or fraction. For instance, fingerprints might be assigned an inherent reliability of 97%, while passwords might only be assigned an inherent reliability of 50%. Those of skill in the art will recognize that these particular values are merely exemplary and may vary between specific implementations.

The second factor for which reliability must be assessed is the reliability of the enrollment. This is part of the "graded enrollment" process referred to above. This reliability factor reflects the reliability of the identification provided during the initial enrollment process. For instance, if the individual initially enrolls in a manner where they physically produce evidence of their identity to a notary or other public official, and enrollment data is recorded at that time and notarized, the data will be more reliable than data which is provided over a network during enrollment and only vouched for by a digital signature or other information which is not truly tied to the individual.

Other enrollment techniques with varying levels of reliability include without limitation: enrollment at a physical office of the trust engine 110 operator; enrollment at a user's place of employment; enrollment at a post office or passport office; enrollment through an affiliated or trusted party to the trust engine 110 operator; anonymous or pseudonymous enrollment in which the enrolled identity is not yet identified with a particular real individual, as well as such other means as are known in the art.

These factors reflect the trust between the trust engine 110 and the source of identification provided during the enrollment process. For instance, if enrollment is performed in association with an employer during the initial process of providing evidence of identity, this information may be considered extremely reliable for purposes within the company, but may be trusted to a lesser degree by a government agency, or by a competitor. Therefore, trust engines operated by each of these other organizations may assign different levels of reliability to this enrollment.

Similarly, additional data which is submitted across a network, but which is authenticated by other trusted data provided during a previous enrollment with the same trust engine 110 may be considered as reliable as the original enrollment data was, even though the latter data were submitted across an open network. In such circumstances, a subsequent notarization will effectively increase the level of reliability associated with the original enrollment data. In this way for example, an anonymous or pseudonymous enrollment may then be raised to a full enrollment by demonstrating to some enrollment official the identity of the individual matching the enrolled data.

The reliability factors discussed above are generally values which may be determined in advance of any particular authentication instance. This is because they are based upon the enrollment and the technique, rather than the actual authentication. In one embodiment, the step of generating reliability based upon these factors involves looking up previously determined values for this particular authentication technique and the enrollment data of the user. In a further aspect of an advantageous embodiment of the present invention, such reliabilities may be included with the enrollment data itself. In this way, these factors are automatically delivered to the authentication engine 215 along with the enrollment data sent from the depository 210.

While these factors may generally be determined in advance of any individual authentication instance, they still have an effect on each authentication instance which uses that particular technique of authentication for that user. Furthermore, although the values may change over time (e.g. if the user re-enrolls in a more reliable fashion), they are not dependent on the authentication data itself. By contrast, the reliability factors associated with a single specific instance's data may vary on each occasion. These factors, as discussed below, must be evaluated for each new authentication in order to generate reliability scores in STEP 1615.

The reliability of the authentication data reflects the match between the data provided by the user in a particular authentication instance and the data provided during the authentication enrollment. This is the fundamental question of whether the authentication data matches the enrollment data for the individual the user is claiming to be. Normally, when the data do not match, the user is considered to not be successfully authenticated, and the authentication fails. The manner in which this is evaluated may change depending on the authentication technique used. The comparison of such data is performed by the comparator 515 function of the authentication engine 215 as shown in FIGURE 5.

For instance, matches of passwords are generally evaluated in a binary fashion. In other words, a password is either a perfect match, or a failed match. It is usually not desirable to accept as even a partial match a password which is close to the correct password if it is not exactly correct. Therefore, when evaluating a password authentication, the reliability of the authentication returned by the comparator 515 is typically either 100% (correct) or 0% (wrong), with no possibility of intermediate values.

Similar rules to those for passwords are generally applied to token based authentication methods, such as smart cards. This is because having a smart card which has a similar identifier or which is similar to the correct one, is still just as wrong as having any other incorrect token. Therefore tokens tend also to be binary authenticators: a user either has the right token, or he doesn't.

However, certain types of authentication data, such as questionnaires and biometrics, are generally not binary authenticators. For example, a fingerprint may match a reference fingerprint to varying degrees. To some extent, this may be due to variations in the quality of the data captured either during the initial enrollment or in subsequent authentications. (A fingerprint may be smudged or a person may have a still healing scar or burn on a particular finger.) In other instances the data may match less than perfectly because the information itself is somewhat variable and based upon pattern matching. (A voice analysis may seem close but not quite right because of background noise, or the acoustics of the environment in which the voice is recorded, or because the person has a cold.) Finally, in situations where large amounts of data are being compared, it may simply be the case that much of the data matches well, but some doesn't. (A ten-question questionnaire may have resulted in eight correct answers to personal questions, but two incorrect answers.) For any of these reasons, the match between the enrollment data and the data for a particular authentication instance may be desirably assigned a partial match value by the comparator 515. In this way, the fingerprint might be said to be a 85% match, the voice print a 65% match, and the questionnaire an 80% match, for example.

This measure (degree of match) produced by the comparator 515 is the factor representing the basic issue of whether an authentication is correct or not. However, as discussed above, this is only one of the factors which may be used in determining the reliability of a given authentication instance. Note also that even though a match to some partial degree may be determined, that ultimately, it may be desirable to provide a binary result based upon a partial match. In an alternate mode of operation, it is also possible to treat partial matches as binary, i.e. either perfect (100%) or failed (0%) matches, based upon whether or not the degree of match passes a particular threshold level of match. Such a process may be used to provide a simple pass/fail level of matching for systems which would otherwise produce partial matches.

Another factor to be considered in evaluating the reliability of a given authentication instance concerns the circumstances under which the authentication data for this particular instance are provided. As discussed above, the circumstances refer to the metadata associated with a particular authentication instance. This may include without limitation such information as: the network address of the authenticator, to the extent that it can be determined; the time of the authentication; the mode of transmission of the authentication data (phone line, cellular, network, etc.); and the serial number of the system of the authenticator.

These factors can be used to produce a profile of the type of authentication that is normally requested by the user. Then, this information can be used to assess reliability in at least two manners. One manner is to consider whether the user is requesting authentication in a manner which is consistent with the normal profile of authentication by this user. If the user normally makes authentication requests from one network address during business days (when she is at work) and from a different network address during evenings or weekends (when she is at home), an authentication which occurs from the home address during the business day is less reliable because it is outside the normal authentication profile. Similarly, if the user normally authenticates using a fingerprint biometric and in the evenings, an authentication which originates during the day using only a password is less reliable.

An additional way in which the circumstantial metadata can be used to evaluate the reliability of an instance of authentication is to determine how much corroboration the circumstance provides that the authenticator is the individual he claims to be. For instance, if the authentication comes from a system with a serial number known to be associated with the user, this is a good circumstantial indicator that the user is who they claim to be. Conversely, if the authentication is coming from a network address which is known to be in Los Angeles when the user is known to reside in London, this is an indication that this authentication is less reliable based on its circumstances.

It is also possible that a cookie or other electronic data may be placed upon the system being used by a user when they interact with a vendor system or with the trust engine 110. This data is written to the storage of the system of the user and may contain an identification which may be read by a Web browser or other software on the user system. If this data is allowed to reside on the user system between sessions (a "persistent cookie"), it may be sent with the authentication data as further evidence of the past use of this system during authentication of a particular user. In effect, the metadata of a given instance, particularly a persistent cookie, may form a sort of token based authenticator itself.

Once the appropriate reliability factors based on the technique and data of the authentication instance are generated as described above in STEPS 1610 and 1615 respectively, they are used to produce an overall reliability for the authentication instance provided in STEP 1620. One means of doing this is simply to express each reliability as a percentage and then to multiply them together.

For example, suppose the authentication data is being sent in from a network address known to be the user's home computer completely in accordance with the user's past authentication profile (100%), and the technique being used is fingerprint identification (97%), and the initial finger print data was registered through the user's employer with the trust engine 110 (90%), and the match between the authentication data and the original fingerprint template in the enrollment data is very good (99%). The overall reliability of this authentication instance could then be calculated as the product of these reliabilities: 100% * 97% * 90% * 99% = 86.4% reliability.

This calculated reliability represents the reliability of one single instance of authentication. The overall reliability of a single authentication instance may also be calculated using techniques which treat the different reliability factors differently, for example by using formulas where different weights are assigned to each reliability factor. Furthermore, those of skill in the art will recognize that the actual values used may represent values other than percentages and may use non-arithmetic systems. One embodiment may include a module used by an authentication requestor to set the weights for each factor and the algorithms used in establishing the overall reliability of the authentication instance.

The authentication engine 215 may use the above techniques and variations thereof to determine the reliability of a single authentication instance, indicated as STEP 1620. However, it may be useful in many authentication situations for multiple authentication instances to be provided at the same time. For example, while attempting to authenticate himself using the system of the present invention, a user may provide a user identification, fingerprint authentication data, a smart card, and a password. In such a case, three independent authentication instances are being provided to the trust engine 110 for evaluation. Proceeding to STEP 1625, if the authentication engine 215 determines that the data provided by the user includes more than one authentication instance, then each instance in turn will be selected as shown in STEP 1630 and evaluated as described above in STEPS 1610, 1615 and 1620.

Note that many of the reliability factors discussed may vary from one of these instances to another. For instance, the inherent reliability of these techniques is likely to be different, as well as the degree of match provided between the authentication data and the enrollment data. Furthermore, the user may have provided enrollment data at different times and under different circumstances for each of these techniques, providing different enrollment reliabilities for each of these instances as well. Finally, even though the circumstances under which the data for each of these instances is being submitted is the same, the use of such techniques may each fit the profile of the user differently, and so may be assigned different circumstantial reliabilities. (For example, the user may normally use their password and fingerprint, but not their smart card.)

As a result, the final reliability for each of these authentication instances may be different from one another. However, by using multiple instances together, the overall confidence level for the authentication will tend to increase.

Once the authentication engine has performed STEPS 1610 through 1620 for all of the authentication instances provided in the authentication data, the reliability of each instance is used in STEP 1635 to evaluate the overall authentication confidence level. This process of combining the individual authentication instance reliabilities into the authentication confidence level may be modeled by various methods relating the individual reliabilities produced, and may also address the particular interaction between some of these authentication techniques. (For example, multiple knowledge-based systems such as passwords may produce less confidence than a single password and even a fairly weak biometric, such as a basic voice analysis.)

One means in which the authentication engine 215 may combine the reliabilities of multiple concurrent authentication instances to generate a final confidence level is to multiply the unreliability of each instance to arrive at a total unreliability. The unreliability is generally the complementary percentage of the reliability. For example, a technique which is 84% reliable is 16% unreliable. The three authentication instances described above (fingerprint, smart card, password) which produce reliabilities of 86%, 75%, and 72% would have corresponding unreliabilities of (100 - 86)%, (100 - 75)% and (100 - 72)%, or 14%, 25%, and 28%, respectively. By multiplying these unreliabilities, we get a cumulative unreliability of 14% * 25% * 28% = .98% unreliability, which corresponds to a reliability of 99.02%.

In an additional mode of operation, additional factors and heuristics 530 may be applied within the authentication engine 215 to account for the interdependence of various authentication techniques. For example, if someone has unauthorized access to a particular home computer, they probably have access to the phone line at that address as well. Therefore, authenticating based on an originating phone number as well as upon the serial number of the authenticating system does not add much to the overall confidence in the authentication. However, knowledge based authentication is largely independent of token based authentication (i.e. if someone steals your cellular phone or keys, they are no more likely to know your PIN or password than if they hadn't).

Furthermore, different vendors or other authentication requestors may wish to weigh different aspects of the authentication differently. This may include the use of separate weighing factors or algorithms used in calculating the reliability of individual instances as well as the use of different means to evaluate authentication events with multiple instances.

For instance, vendors for certain types of transactions, for instance corporate email systems, may desire to authenticate primarily based upon heuristics and other circumstantial data by default. Therefore, they may apply high weights to factors related to the metadata and other profile related information associated with the circumstances surrounding authentication events. This arrangement could be used to ease the burden on users during normal operating hours, by not requiring more from the user than that he be logged on to the correct machine during business hours. However, another vendor may weigh authentications coming from a particular technique most heavily, for

instance fingerprint matching, because of a policy decision that such a technique is most suited to authentication for the particular vendor's purposes.

Such varying weights may be defined by the authentication requestor in generating the authentication request and sent to the trust engine 110 with the authentication request in one mode of operation. Such options could also be set as preferences during an initial enrollment process for the authentication requestor and stored within the authentication engine in another mode of operation.

Once the authentication engine 215 produces an authentication confidence level for the authentication data provided, this confidence level is used to complete the authentication request in STEP 1640, and this information is forwarded from the authentication engine 215 to the transaction engine 205 for inclusion in a message to the authentication requestor.

The process described above is merely exemplary, and those of skill in the art will recognize that the steps need not be performed in the order shown. Furthermore, certain steps, such as the evaluation of the reliability of each authentication instance provided, may be carried out in parallel with one another if circumstances permit.

In a further aspect of this invention, a method is provided to accommodate conditions when the authentication confidence level produced by the process described above fails to meet the required trust level of the vendor or other party requiring the authentication. In circumstances such as these where a gap exists between the level of confidence provided and the level of trust desired, the operator of the trust engine 110 is in a position to provide opportunities for one or both parties to provide alternate data or requirements in order to close this trust gap. This process will be referred to as "trust arbitrage" herein.

Trust arbitrage may take place within a framework of cryptographic authentication as described above with reference to FIGURES 10 and 11. As shown therein, a vendor or other party will request authentication of a particular user in association with a particular transaction. In one circumstance, the vendor simply requests an authentication, either positive or negative, and after receiving appropriate data from the user, the trust engine 110 will provide such a binary authentication. In circumstances such as these, the degree of confidence required in order to secure a positive authentication is determined based upon preferences set within the trust engine 110.

However, it is also possible that the vendor may request a particular level of trust in order to complete a particular transaction. This required level may be included with the authentication request (e.g. authenticate this user to 96% confidence) or may be determined by the trust engine 110 based on other factors associated with the transaction (i.e. authenticate this user as appropriate for this transaction). One such factor might be the economic value of the transaction. For transactions which have greater economic value, a higher degree of trust may be required. Similarly, for transactions with high degrees of risk a high degree of trust may be required. Conversely, for transactions which are either of low risk or of low value, lower trust levels may be required by the vendor or other authentication requestor.

The process of trust arbitrage occurs between the steps of the trust engine 110 receiving the authentication data in STEP 1050 of FIGURE 10 and the return of an authentication result to the vendor in STEP 1055 of FIGURE 10.

Between these steps, the process which leads to the evaluation of trust levels and the potential trust arbitrage occurs as shown in FIGURE 17. In circumstances where simple binary authentication is performed, the process shown in FIGURE 17 reduces to having the transaction engine 205 directly compare the authentication data provided with the enrollment data for the identified user as discussed above with reference to FIGURE 10, flagging any difference as a negative authentication.

As shown in FIGURE 17, the first step after receiving the data in STEP 1050 is for the transaction engine 205 to determine the trust level which is required for a positive authentication for this particular transaction in STEP 1710. This step may be performed by one of several different methods. The required trust level may be specified to the trust engine 110 by the authentication requestor at the time when the authentication request is made. The authentication requestor may also set a preference in advance which is stored within the depository 210 or other storage which is accessible by the transaction engine 205. This preference may then be read and used each time an authentication request is made by this authentication requestor. The preference may also be associated with a particular user as a security measure such that a particular level of trust is always required in order to authenticate that user, the user preference being stored in the depository 210 or other storage media accessible by the transaction engine 205. The required level may also be derived by the transaction engine 205 or authentication engine 215 based upon information provided in the authentication request, such as the value and risk level of the transaction to be authenticated.

In one mode of operation, a policy management module or other software which is used when generating the authentication request is used to specify the required degree of trust for the authentication of the transaction. This may be used to provide a series of rules to follow when assigning the required level of trust based upon the policies which are specified within the policy management module. One advantageous mode of operation is for such a module to be incorporated with the web server of a vendor in order to appropriately determine required level of trust for transactions initiated with the vendor's web server. In this way, transaction requests from users may be assigned a required trust level in accordance with the policies of the vendor and such information may be forwarded to the trust engine 110 along with the authentication request.

This required trust level correlates with the degree of certainty that the vendor wants to have that the individual authenticating is in fact who he identifies himself as. For example, if the transaction is one where the vendor wants a fair degree of certainty because goods are changing hands, the vendor may require a trust level of 85%. For situation where the vendor is merely authenticating the user to allow him to view members only content or exercise privileges on a chat room, the downside risk may be small enough that the vendor requires only a 60% trust level. However, to enter into a production contract with a value of tens of thousands of dollars, the vendor may require a trust level of 99% or more.

This required trust level represents a metric to which the user must authenticate himself in order to complete the transaction. If the required trust level is 85% for example, the user must provide authentication to the trust engine 110 sufficient for the trust engine 110 to say with 85% confidence that the user is who they say they are. It is the

balance between this required trust level and the authentication confidence level which produces either a positive authentication (to the satisfaction of the vendor) or a possibility of trust arbitrage.

As shown in FIGURE 17, after the transaction engine 205 receives the required trust level, it compares in STEP 1720 the required trust level to the authentication confidence level which the authentication engine 215 calculated for the current authentication (as discussed with reference to FIGURE 16). If the authentication confidence level is higher than the required trust level for the transaction in STEP 1730, then the process moves to STEP 1740 where a positive authentication for this transaction is produced by the transaction engine 205. A message to this effect will then be inserted into the authentication results and returned to the vendor by the transaction engine 205 as shown in STEP 1055 (see FIGURE 10).

However, if the authentication confidence level does not fulfill the required trust level in STEP 1730, then a confidence gap exists for the current authentication, and trust arbitrage is conducted in STEP 1750. Trust arbitrage is described more completely with reference to FIGURE 18 below. This process as described below takes place within the transaction engine 205 of the trust engine 110. Because no authentication or other cryptographic operations are needed to execute trust arbitrage (other than those required for the SSL communication between the transaction engine 205 and other components), the process may be performed outside the authentication engine 215. However, as will be discussed below, any reevaluation of authentication data or other cryptographic or authentication events will require the transaction engine 205 to resubmit the appropriate data to the authentication engine 215. Those of skill in the art will recognize that the trust arbitrage process could alternately be structured to take place partially or entirely within the authentication engine 215 itself.

As mentioned above, trust arbitrage is a process where the trust engine 110 mediates a negotiation between the vendor and user in an attempt to secure a positive authentication where appropriate. As shown in STEP 1805, the transaction engine 205 first determines whether or not the current situation is appropriate for trust arbitrage. This may be determined based upon the circumstances of the authentication, e.g. whether this authentication has already been through multiple cycles of arbitrage, as well as upon the preferences of either the vendor or user, as will be discussed further below.

In such circumstances where arbitrage is not possible, the process proceeds to STEP 1810 where the transaction engine 205 generates a negative authentication and then inserts it into the authentication results which are sent to the vendor in STEP 1055 (see FIGURE 10). One limit which may be advantageously used to prevent authentications from pending indefinitely is to set a time-out period from the initial authentication request. In this way, any transaction which is not positively authenticated within the time limit is denied further arbitrage and negatively authenticated. Those of skill in the art will recognize that such a time limit may vary depending upon the circumstances of the transaction and the desires of the user and vendor. Limitations may also be placed upon the number of attempts that may be made at providing a successful authentication. Such limitations may be handled by an attempt limiter 535 as shown in FIGURE 5.

If arbitrage is not prohibited in STEP 1805, the transaction engine 205 will then engage in negotiation with one or both of the transacting parties. The transaction engine 205 may send a message to the user requesting some form of additional authentication in order to boost the authentication confidence level produced as shown in STEP 1820. In the simplest form, this may simply indicates that authentication was insufficient. A request to produce one or more additional authentication instances to improve the overall confidence level of the authentication may also be sent.

If the user provides some additional authentication instances in STEP 1825, then the transaction engine 205 adds these authentication instances to the authentication data for the transaction and forwards it to the authentication engine 215 as shown in STEP 1015 (see FIGURE 10), and the authentication is reevaluated based upon both the pre-existing authentication instances for this transaction and the newly provided authentication instances.

An additional type of authentication may be a request from the trust engine 110 to make some form of person-to-person contact between the trust engine 110 operator (or a trusted associate) and the user, for example, by phone call. This phone call or other non-computer authentication can be used to provide personal contact with the individual and also to conduct some form of questionnaire based authentication. This also may give the opportunity to verify an originating telephone number and potentially a voice analysis of the user when he calls in. Even if no additional authentication data can be provided, the additional context associated with the user's phone number may improve the reliability of the authentication context. Any revised data or circumstances based upon this phone call are fed into the trust engine 110 for use in consideration of the authentication request.

Additionally, in STEP 1820 the trust engine 110 may provide an opportunity for the user to purchase insurance, effectively buying a more confident authentication. The operator of the trust engine 110 may, at times, only want to make such an option available if the confidence level of the authentication is above a certain threshold to begin with. In effect, this user side insurance is a way for the trust engine 110 to vouch for the user when the authentication meets the normal required trust level of the trust engine 110 for authentication, but does not meet the required trust level of the vendor for this transaction. In this way, the user may still successfully authenticate to a very high level as may be required by the vendor, even though he only has authentication instances which produce confidence sufficient for the trust engine 110.

This function of the trust engine 110 allows the trust engine 110 to vouch for someone who is authenticated to the satisfaction of the trust engine 110, but not of the vendor. This is analogous to the function performed by a notary in adding his signature to a document in order to indicate to someone reading the document at a later time that the person whose signature appears on the document is in fact the person who signed it. The signature of the notary testifies to the act of signing by the user. In the same way, the trust engine is providing an indication that the person transacting is who they say they are.

However, because the trust engine 110 is artificially boosting the level of confidence provided by the user, there is a greater risk to the trust engine 110 operator, since the user is not actually meeting the required trust level of the vendor. The cost of the insurance is designed to offset the risk of a false positive authentication to the trust

engine 110 (who may be effectively notarizing the authentications of the user). The user pays the trust engine 110 operator to take the risk of authenticating to a higher level of confidence than has actually been provided.

Because such an insurance system allows someone to effectively buy a higher confidence rating from the trust engine 110, both vendors and users may wish to prevent the use of user side insurance in certain transactions. Vendors may wish to limit positive authentications to circumstances where they know that actual authentication data supports the degree of confidence which they require and so may indicate to the trust engine 110 that user side insurance is not to be allowed. Similarly, to protect his online identity, a user may wish to prevent the use of user side insurance on his account, or may wish to limit its use to situations where the authentication confidence level without the insurance is higher than a certain limit. This may be used as a security measure to prevent someone from overhearing a password or stealing a smart card and using them to falsely authenticate to a low level of confidence, and then purchasing insurance to produce a very high level of (false) confidence. These factors may be evaluated in determining whether user side insurance is allowed.

If user purchases insurance in STEP 1840, then the authentication confidence level is adjusted based upon the insurance purchased in STEP 1845, and the authentication confidence level and required trust level are again compared in STEP 1730 (see FIGURE 17). The process continues from there, and may lead to either a positive authentication in STEP 1740 (see FIGURE 17), or back into the trust arbitrage process in STEP 1750 for either further arbitrage (if allowed) or a negative authentication in STEP 1810 if further arbitrage is prohibited.

In addition to sending a message to the user in STEP 1820, the transaction engine 205 may also send a message to the vendor in STEP 1830 which indicates that a pending authentication is currently below the required trust level. The message may also offer various options on how to proceed to the vendor. One of these options is to simply inform the vendor of what the current authentication confidence level is and ask if the vendor wishes to maintain their current unfulfilled required trust level. This may be beneficial because in some cases, the vendor may have independent means for authenticating the transaction or may have been using a default set of requirements which generally result in a higher required level being initially specified than is actually needed for the particular transaction at hand.

For instance, it may be standard practice that all incoming purchase order transactions with the vendor are expected to meet a 98% trust level. However, if an order was recently discussed by phone between the vendor and a long-standing customer, and immediately thereafter the transaction is authenticated, but only to a 93% confidence level, the vendor may wish to simply lower the acceptance threshold for this transaction, because the phone call effectively provides additional authentication to the vendor. In certain circumstances, the vendor may be willing to lower their required trust level, but not all the way to the level of the current authentication confidence. For instance, the vendor in the above example might consider that the phone call prior to the order might merit a 4% reduction in the degree of trust needed; however, this is still greater than the 93% confidence produced by the user.

If the vendor does adjust their required trust level in STEP 1835, then the authentication confidence level produced by the authentication and the required trust level are compared in STEP 1730 (see FIGURE 17). If the

confidence level now exceeds the required trust level, a positive authentication may be generated in the transaction engine 205 in STEP 1740 (see FIGURE 17). If not, further arbitrage may be attempted as discussed above if it is permitted.

In addition to requesting an adjustment to the required trust level, the transaction engine 205 may also offer vendor side insurance to the vendor requesting the authentication. This insurance serves a similar purpose to that described above for the user side insurance. Here, however, rather than the cost corresponding to the risk being taken by the trust engine 110 in authenticating above the actual authentication confidence level produced, the cost of the insurance corresponds to the risk being taken by the vendor in accepting a lower trust level in the authentication.

Instead of just lowering their actual required trust level, the vendor has the option of purchasing insurance to protect itself from the additional risk associated with a lower level of trust in the authentication of the user. As described above, it may be advantageous for the vendor to only consider purchasing such insurance to cover the trust gap in conditions where the existing authentication is already above a certain threshold.

The availability of such vendor side insurance allows the vendor the option to either: lower his trust requirement directly at no additional cost to himself, bearing the risk of a false authentication himself (based on the lower trust level required); or, buying insurance for the trust gap between the authentication confidence level and his requirement, with the trust engine 110 operator bearing the risk of the lower confidence level which has been provided. By purchasing the insurance, the vendor effectively keeps his high trust level requirement, because the risk of a false authentication is shifted to the trust engine 110 operator.

If the vendor purchases insurance in STEP 1840, the authentication confidence level and required trust levels are compared in STEP 1730 (see FIGURE 17), and the process continues as described above.

Note that it is also possible that both the user and the vendor respond to messages from the trust engine 110. Those of skill in the art will recognize that there are multiple ways in which such situations can be handled. One advantageous mode of handling the possibility of multiple responses is simply to treat the responses in a first-come, first-served manner. For example, if the vendor responds with a lowered required trust level and immediately thereafter the user also purchases insurance to raise his authentication level, the authentication is first reevaluated based upon the lowered trust requirement from the vendor. If the authentication is now positive, the user's insurance purchase is ignored. In another advantageous mode of operation, the user might only be charged for the level of insurance required to meet the new, lowered trust requirement of the vendor (if a trust gap remained even with the lowered vendor trust requirement).

If no response from either party is received during the trust arbitrage process at STEP 1850 within the time limit set for the authentication, the arbitrage is reevaluated in STEP 1805. This effectively begins the arbitrage process again. If the time limit was final or other circumstances prevent further arbitrage in STEP 1805, a negative authentication is generated by the transaction engine 205 in STEP 1810 and returned to the vendor in STEP 1055 (see FIGURE 10). If not, new messages may be sent to the user and vendor, and the process may be repeated as desired.

Note that for certain types of transactions, for instance, digitally signing documents which are not part of a transaction, there may not necessarily be a vendor or other third party; therefore the transaction is primarily between the user and the trust engine 110. In circumstances such as these, the trust engine 110 will have its own required trust level which must be satisfied in order to generate a positive authentication. However, in such circumstances, it will often not be desirable for the trust engine 110 to offer insurance to the user in order for him to raise the confidence of his own signature.

The process described above and shown in FIGURES 16 - 18 may be carried out using various communications modes as described above with reference to the trust engine 110. For instance, the messages may be web-based and sent using SSL connections between the trust engine 110 and applets downloaded in real time to browsers running on the user or vendor systems. In an alternate mode of operation, certain dedicated applications may be in use by the user and vendor which facilitate such arbitrage and insurance transactions. In another alternate mode of operation, secure email operations may be used to mediate the arbitrage described above, thereby allowing deferred evaluations and batch processing of authentications. Those of skill in the art will recognize that different communications modes may be used as are appropriate for the circumstances and authentication requirements of the vendor.

The following description with reference to FIGURE 19 describes a sample transaction which integrates the various aspects of the present invention as described above. This example illustrates the overall process between a user and a vendor as mediates by the trust engine 110. Although the various steps and components as described in detail above may be used to carry out the following transaction, the process illustrated focuses on the interaction between the trust engine 110, user and vendor.

The transaction begins when the user, while viewing web pages online, fills out an order form on the web site of the vendor in STEP 1900. The user wishes to submit this order form to the vendor, signed with his digital signature. In order to do this, the user submits the order form with his request for a signature to the trust engine 110 in STEP 1905. The user will also provide authentication data which will be used as described above to authenticate his identity.

In STEP 1910 the authentication data is compared to the enrollment data by the trust engine 110 as discussed above, and if a positive authentication is produced, the hash of the order form, signed with the private key of the user, is forwarded to the vendor along with the order form itself.

The vendor receives the signed form in STEP 1915, and then the vendor will generate an invoice or other contract related to the purchase to be made in STEP 1920. This contract is sent back to the user with a request for a signature in STEP 1925. The vendor also sends an authentication request for this contract transaction to the trust engine 110 in STEP 1930 including a hash of the contract which will be signed by both parties. To allow the contract to be digitally signed by both parties, the vendor also includes authentication data for itself so that the vendor's signature upon the contract can later be verified if necessary.

As discussed above, the trust engine 110 then verifies the authentication data provided by the vendor to confirm the vendor's identity, and if the data produces a positive authentication in STEP 1935, continues with STEP 1955 when the data is received from the user. If the vendor's authentication data does not match the enrollment data of the vendor to the desired degree, a message is returned to the vendor requesting further authentication. Trust arbitrage may be performed here if necessary, as described above, in order for the vendor to successfully authenticate itself to the trust engine 110.

When the user receives the contract in STEP 1940, he reviews it, generates authentication data to sign it if it is acceptable in STEP 1945, and then sends a hash of the contract and his authentication data to the trust engine 110 in STEP 1950. The trust engine 110 verifies the authentication data in STEP 1955 and if the authentication is good, proceeds to process the contract as described below. As discussed above with reference to FIGURES 17 and 18, trust arbitrage may be performed as appropriate to close any trust gap which exists between the authentication confidence level and the required authentication level for the transaction.

The trust engine 110 signs the hash of the contract with the user's private key, and sends this signed hash to the vendor in STEP 1960, signing the complete message on its own behalf, i.e. including a hash of the complete message (including the user's signature) encrypted with the private key 510 of the trust engine 110. This message is received by the vendor in STEP 1965. The message represents a signed contract (hash of contract encrypted using user's private key) and a receipt from the trust engine 110 (the hash of the message including the signed contract, encrypted using the trust engine 110's private key).

The trust engine 110 similarly prepares a hash of the contract with the vendor's private key in STEP 1970, and forwards this to the user, signed by the trust engine 110. In this way, the user also receives a copy of the contract, signed by the vendor, as well as a receipt, signed by the trust engine 110, for delivery of the signed contract in STEP 1975.

In addition to the foregoing, an additional aspect of the invention provides a cryptographic Service Provider Module (SPM) which may be available to a client side application as a means to access functions provided by the trust engine 110 described above. One advantageous way to provide such a service is for the cryptographic SPM is to mediate communications between a third party Application Programming Interface (API) and a trust engine 110 which is accessible via a network or other remote connection. A sample cryptographic SPM is described below with reference to FIGURE 20.

For example, on a typical system, a number of API's are available to programmers. Each API provides a set of function calls which may be made by an application 2000 running upon the system. Examples of API's which provide programming interfaces suitable for cryptographic functions, authentication functions, and other security function include the Cryptographic API (CAPI) 2010 provided by Microsoft with its Windows operating systems, and the Common Data Security Architecture (CDSA), sponsored by IBM, Intel and other members of the Open Group. CAPI will be used as an exemplary security API in the discussion that follows. However, the cryptographic SPM described could be used with CDSA or other security API's as are known in the art.

This API is used by a user system 105 or vendor system 120 when a call is made for a cryptographic function. Included among these functions may be requests associated with performing various cryptographic operations, such as encrypting a document with a particular key, signing a document, requesting a digital certificate, verifying a signature upon a signed document, and such other cryptographic functions as are described above or known to those of skill in the art.

Such cryptographic functions are normally performed locally to the system upon which CAPI 2010 is located. This is because generally the functions called require the use of either resources of the local user system 105, such as a fingerprint reader, or software functions which are programmed using libraries which are executed on the local machine. Access to these local resources is normally provided by one or more Service Provider Modules (SPM's) 2015, 2020 as referred to above which provide resources with which the cryptographic functions are carried out. Such SPM's may include software libraries 2015 to perform encrypting or decrypting operations, or drivers and applications 2020 which are capable of accessing specialized hardware 2025, such as biometric scanning devices. In much the way that CAPI 2010 provides functions which may be used by an application 2000 of the system 105, the SPM's 2015, 2020 provide CAPI with access to the lower level functions and resources associated with the available services upon the system.

In accordance with the invention, it is possible to provide a cryptographic SPM 2030 which is capable of accessing the cryptographic functions provided by the trust engine 110 and making these functions available to an application 2000 through CAPI 2010. Unlike embodiments where CAPI 2010 is only able to access resources which are locally available through SPM's 2015, 2020, a cryptographic SPM 2030 as described herein would be able to submit requests for cryptographic operations to a remotely-located, network-accessible trust engine 110 in order to perform the operations desired.

For instance, if an application 2000 has a need for a cryptographic operation, such as signing a document, the application 2000 makes a function call to the appropriate CAPI 2010 function. CAPI 2010 in turn will execute this function, making use of the resources which are made available to it by the SPM's 2015, 2020 and the cryptographic SPM 2030. In the case of a digital signature function, the cryptographic SPM 2030 will generate an appropriate request which will be sent to the trust engine 110 across the communication link 125.

The operations which occur between the cryptographic SPM 2030 and the trust engine 110 are the same operations that would be possible between any other system and the trust engine 110. However, these functions are effectively made available to a user system 105 through CAPI 2010 such that they appear to be locally available upon the user system 105 itself. However, unlike ordinary SPM's 2015, 2020, the functions are being carried out on the remote trust engine 110 and the results relayed to the cryptographic SPM 2030 in response to appropriate requests across the communication link 125.

This cryptographic SPM 2030 makes a number of operations available to the user system 105 or a vendor system 120 which might not otherwise be available. These functions include without limitation: encryption and

decryption of documents; issuance of digital certificates; digital signing of documents; verification of digital signatures; and such other operations as will be apparent to those of skill in the art.

Additionally, other combinations, admissions, substitutions and modifications will be apparent to the skilled artisan in view of the disclosure herein. Accordingly, the present invention is not intended to be limited by the reaction of the preferred embodiments but is to be defined by a reference to the appended claims.

WHAT IS CLAIMED IS:

1. A method of implementing an authentication transaction related to an electronic transaction between a vendor and a user, the method comprising:

receiving a request for an authentication transaction from a vendor, wherein the authentication transaction is related to an electronic transaction involving the vendor and a user, and the authentication transaction includes a transaction identifier (TID);

forwarding enrollment authentication data corresponding to the user and the TID to an authentication engine;

querying the user for current authentication data and the TID;

receiving the current authentication data along with the TID from the user and forwarding the current authentication data and the TID to the authentication engine;

matching the enrollment authentication data to the current authentication data through the TID; and

comparing the enrollment authentication data to the current authentication data in order to generate an authentication result.

2. A method of implementing a cryptographic transaction related to an electronic transaction between a vendor and a user, the method comprising:

associating a user from multiple users with one or more keys from a plurality of private cryptographic keys stored on a secure server, wherein the one or more keys are unknown to the user;

receiving a request for a cryptographic transaction from a vendor, wherein the cryptographic transaction is related to an electronic transaction involving the vendor and the user and the cryptographic transaction includes a transaction identifier (TID);

forwarding enrollment authentication data corresponding to the user and the TID to an authentication engine;

querying the user for current authentication data and the TID;

receiving the current authentication data along with the TID from the user and forwarding the current authentication data and the TID to the authentication engine;

matching the enrollment authentication data to the current authentication data through the TID;

comparing the enrollment authentication data to the current authentication data in order to generate an authentication result; and

when the authentication result uniquely identifies the user, employing the one or more keys to perform one or more cryptographic functions.

3. The method of Claim 2, wherein the one or more cryptographic functions include one of digital signing, encryption, decryption, hash creation, key generation, and key destruction.

4. The method of Claim 2 wherein the electronic transaction represents logging into a portal which provides access to one or more services which require authentication of the user.

5. The method of Claim 4 additionally comprising unlocking a password vault associated with the user, the password vault providing access to additional data which is used to authenticate the user to additional services.

6. The method of Claim 5 wherein the additional data comprises passwords associated with the user.

7. The method of Claim 5 wherein the additional data comprises private cryptographic keys associated with the user.

8. The method of Claim 5 wherein the vendor is a trust engine and the password vault is associated with the trust engine.

9. The method of Claim 5 wherein the vendor is the operator of the portal and the password vault is associated with the vendor.

10. The method of Claim 5 wherein the password vault is stored on a system associated with the user.

11. A method of facilitating a cryptographic function related to an electronic transaction, the method comprising:

receiving at a trust engine, data corresponding to a transaction between at least a user and a vendor;

receiving at the trust engine, a request from the vendor to obtain the digital signature of the user; and

generating the digital signature at the trust engine by using at least one private cryptographic key associated with the user.

12. The method of Claim 11, wherein the step of generating the digital signature further comprises acquiring an appropriate certificate corresponding to the at least one private cryptographic key.

13. A cryptographic system for performing cryptographic functions, the cryptographic system comprising:

data corresponding to a transaction between at least a user and a vendor;

a request from the vendor to obtain a digital signature of the user; and

a cryptographic handling module which receives the data and the request, and which generates the digital signature by using at least one private cryptographic key associated with the user.

14. A method of facilitating a cryptographic function related to an electronic transaction, the method comprising:

receiving at a first entity, data about a transaction between at least a second entity and a third entity;

receiving at the first entity, a request from the third entity to obtain the digital signature of the second entity; and

signing at the first entity the transaction data using at least one private cryptographic key associated with the second entity.

15. A method of facilitating a cryptographic function related to an electronic transaction, the method comprising:

associating in a trust engine, a user from multiple users with one or more keys from a plurality of private cryptographic keys stored on the trust engine;

receiving at a trust engine, transaction data related to a transaction between at least a user and a vendor;

receiving at the trust engine, a request to digitally sign the transaction data; and

signing the transaction data with the one or more keys associated with the user.

16. A cryptographic system for performing cryptographic functions, the cryptographic system comprising:

a computer readable storage medium storing a plurality of private cryptographic keys;

transaction data related to a transaction between at least a user and a vendor; and

a cryptographic handling module which associates the user with one or more keys of the plurality of private cryptographic keys, which receives the request and the transaction data, and which signs the transaction data with the one or more keys associated with the user.

17. A method of facilitating a cryptographic function related to an electronic transaction, the method comprising:

receiving transaction data that comprises a portion of an agreement between at least a first and second entity;

receiving a request for a cryptographic function; and

signing the transaction data with at least one private cryptographic key stored on a secure server.

18. A method of facilitating a cryptographic transaction related to an electronic transaction between first and second users, the method comprising:

receiving authentication data from a first user;

receiving transaction data identifying a cryptographic transaction between the first user and a second user;

comparing the authentication data to enrollment authentication data corresponding to the first user, thereby verifying the identity of the first user;

utilizing at least one private cryptographic key stored on a secure server and unknown to either user to perform cryptographic functions relating to the cryptographic transaction without releasing the at least one private cryptographic key to either user; and

upon authentication of the identity of at least the first user, transmitting data related to the cryptographic transaction to one of the first and second users, without including additional cryptographic keys.

19. The method of Claim 18, wherein the at least one private cryptographic key is associated with the secure server.

20. The method of Claim 18, wherein the at least one private cryptographic key is associated with one of the first or second users.

21. The method of Claim 17, wherein the step of comparing the authentication data includes voiding the cryptographic transaction when a number of authentication attempts related to the cryptographic transaction reaches a predetermined amount.

22. A method as in Claim 18, wherein the authentication data comprises circumstantial data associated with the generation of the authentication data by the first user.

23. A method as in Claim 22, wherein a reliability is associated with the authentication based upon the comparison of the circumstantial data to circumstantial data acquired prior to the transaction on one or more occasions.

24. A method of increasing the speed of an authentication process by performing various authentication steps in parallel, the method comprising:

receiving a request for a cryptographic function related to an electronic transaction involving a vendor and a user;

processing the cryptographic function while awaiting authentication data from the user;

receiving the authentication data from the user; and

transmitting response related to the cryptographic function when the authentication data from the user uniquely identifies the user.

25. The method of Claim 24, wherein the step of processing further comprises retrieving enrollment authentication data.

26. The method of Claim 24, wherein the step of processing further comprises retrieving one or more cryptographic keys associated with one of the user and the vendor.

27. The method of Claim 26, wherein the step of retrieving one or more cryptographic keys further comprises obtaining an appropriate key type.

28. The method of Claim 24, wherein the step of performing the cryptographic function further comprises generating a digital signature for one of the user and the vendor.

29. A method of increasing the speed of an authentication process by performing various authentication steps in parallel, the method comprising:

receiving an authentication request from a vendor wherein the authentication request is related to an electronic transaction involving the vendor and a user;

retrieving enrollment data associated with the user while awaiting authentication data from the user; and

receiving the authentication data from the user and comparing the authentication data to the retrieved enrollment data.

30. The method of Claim 29, wherein the authentication request comprises a user ID identifying the user.

31. The method of Claim 30, wherein the authentication request further comprises a transaction ID identifying the electronic transaction.

32. The method of Claim 24, wherein the authentication request comprises a transaction ID identifying the electronic transaction.

33. The method of Claim 32, wherein the transaction ID comprises a unique number.

34. The method of Claim 29, wherein the authentication request comprises a vendor ID identifying the vendor.

35. The method of Claim 29, wherein the authentication request comprises data indicating a type of authentication data requested from the user.

36. The method of Claim 35, wherein the type of authentication data comprises biometric data.

37. The method of Claim 29, wherein the step of retrieving enrollment data further comprises retrieving substantially randomized portions of enrollment data, each substantially randomized portion being retrieved from a different one of a plurality of data storage facilities.

38. The method of Claim 37, wherein the step of retrieving enrollment data further comprises assembling the substantially randomized portions of enrollment data into a deciphered usable form.

39. The method of Claim 29, wherein the step of comparing the authentication data to the enrollment data includes voiding the authentication request when a number of failed comparison attempts is greater than a predetermined threshold.

40. A method of conducting an authentication transaction to secure an electronic transaction, the method comprising:

receiving an authentication request associated with an electronic transaction from a first user;

receiving data from a second user, the data comprising authentication data generated by the second user and circumstantial data associated with the generation of the authentication data;

determining a level of trust associated with the authentication data, the level of trust based in part upon the circumstantial data; and

providing a response to the authentication request to one of the first user and the second user indicating that the second user has been authenticated.

41. A method as in Claim 40 wherein the first user represents a seller of goods and the electronic transaction represents a sale of goods by the first user.

42. A method as in Claim 40 wherein the first user represents the operator of a server and the electronic transaction represents an access to the server by the first user.

43. A method as in Claim 40 wherein the first user represents a email provider and the electronic transaction represents signing an email message by the first user.

44. A method as in Claim 40 further comprising comparing the level of trust associated with the authentication data with a desired level of trust.

45. A method as in Claim 44 wherein the desired level of trust is determined at least in part by the properties of the transaction associated with the authentication request.

46. A method as in Claim 44 wherein the desired level of trust is specified by the first user in the authentication request.

47. A method for conducting an authentication transaction to provide security for an electronic transaction, the method comprising:

receiving authentication data from a first user, wherein the authentication data is associated with an authentication transaction;

obtaining data corresponding to the circumstances surrounding the process of entering the authentication data;

determining a reliability for the authentication data based upon the obtained data;

comparing the authentication data to previously stored enrollment data;

generating a level of trust for the authentication data associated with the authentication transaction; and

providing a response for use in the transaction.

48. A method as in Claim 47 wherein the authentication data is received over a computer network and the response is sent over the computer network.

49. A method as in Claim 48 wherein the authentication data is received via an email message and the response is sent via an email message.

50. A method as in Claim 48 wherein the authentication data is received from a client-side applet.

51. A method as in Claim 47 wherein the electronic transaction comprises cryptographically signing an email message by the first user.

52. A method as in Claim 47 wherein the electronic transaction comprises cryptographically signing a document by the first user.

53. A method as in Claim 52 further comprising encrypting a hash of the document with a private key associated with the first user.

54. A method as in Claim 47 wherein the electronic transaction comprises encrypting a document by the first user such that it is only readable by a third user.

55. A method as in Claim 47 wherein the electronic transaction comprises decrypting a document with a private key associated with the first user.

56. A system for authenticating an electronic transaction between a first user and a second user, comprising:

authentication data received from a first user and associated with an authentication transaction;

circumstantial data corresponding to the circumstances surrounding the authentication data;

reliability data associated with the authentication data and based upon the circumstantial data;

enrollment data associated with the first user; and

a trust engine which compares the authentication data to the enrollment data and generates a level of trust for the authentication data and provides the level of trust for use authenticating the first user.

57. A system as in Claim 56 wherein the trust engine compares the level of trust to a desired level of trust provided by the second user.

58. A system as in Claim 56 wherein the system further comprises a private key associated with the first user, and wherein the trust engine uses the private key to record the assent of the first user to the electronic transaction.

59. An apparatus that provides cryptographic functions by interconnecting a client-side application programming interface with a network-based trust engine, comprising:

a client-side application programming interface that is configured to operate on a client computer;

a service provider module that is in communication with the application programming interface and in communication with a network-based trust engine, wherein the service provider module is configured to request cryptographic functions from the network-based trust engine.

60. An apparatus as in Claim 59 wherein the cryptographic functions comprise securely authenticating a user to the client-side application programming interface.

61. An apparatus as in Claim 59 wherein the cryptographic functions comprise digitally signing a message for the client-side application programming interface.

62. An apparatus as in Claim 59 wherein the cryptographic functions comprise encrypting a message.

63. A method for performing security functions with a network-based cryptographic server, comprising:

receiving at a user system, a request for a security function from an application programming interface; and

routing the request to a network-based cryptographic server.

64. A method as in Claim 63 wherein the security function comprises an attempt to authenticate a user on the user system.

65. A method as in Claim 63 wherein the security function comprises encrypting a message.

66. A method as in Claim 63 wherein the security function comprises digitally signing a message.

67. An apparatus for making security functions available to a user system from a network based cryptographic server, comprising:

a cryptographic service provider module in communication with an application programming interface of a user system, the cryptographic service provider module receiving requests from the applications programming interface of a user system; and

a cryptographic server in communication with the cryptographic service provider module across a communication link, the cryptographic server providing responses to requests from the cryptographic service provider module across the communication link.

68. An apparatus as in Claim 67 wherein the security functions comprise authenticating a user on the user system.

69. An apparatus as in Claim 67 wherein the cryptographic functions comprise providing encryption of a message.

70. A method of providing a cryptographic service with a network accessible server, comprising:

receiving a request for a cryptographic service from an application programming interface;

preparing a request for cryptographic functions, the cryptographic functions being necessary to provide the requested cryptographic service to the application programming interface;

sending the request for cryptographic functions to a server across a communications link;

receiving a result based upon the request for cryptographic functions from the server;

preparing a response to the request for the cryptographic service based upon the result received from the server; and

sending the response to the request for the cryptographic server available to the application programming interface.

71. A method of performing cryptographic functions with a network-based cryptographic server, comprising:

receiving an authentication request from an application programming interface; and

processing the authentication request with a remotely located server.

72. A method as in Claim 71 further comprising sending authentication data to the remotely located server.

73. A method as in Claim 72 further comprising comparing the authentication data with enrollment data accessible by the remotely located server.

74. A method as in Claim 71 further comprising providing a response from the remotely located server based upon the authentication request.

75. A method of performing authentication functions with a network-based trust engine, comprising:

receiving at a user computer, a request for an authentication function from an application programming interface; and

routing the request to a network-based trust engine.

76. A method as in Claim 75 wherein the authentication function comprises providing digital certificates.

77. A method as in Claim 75 wherein the authentication function comprises verifying the authenticity of a signed document.

78. A method of interconnecting a cryptographic application programming interface with a network accessible resource, comprising:

a cryptographic application programming interface on a client computer; and

a cryptographic service provider module that is in communication with the cryptographic application programming interface wherein the cryptographic service provider is configured to perform cryptographic functions via a network accessible resource.

79. An apparatus for providing cryptographic functions to a user system from a remote trust engine comprising a cryptographic service provider module in communication with an application programming interface of the user system, the cryptographic service provider module receiving a request for cryptographic functions from the

application programming interface, and the cryptographic service provider module connected to a network accessible trust engine via a communication link, the trust engine providing a response to a requests for cryptographic functions from the cryptographic service provider module.

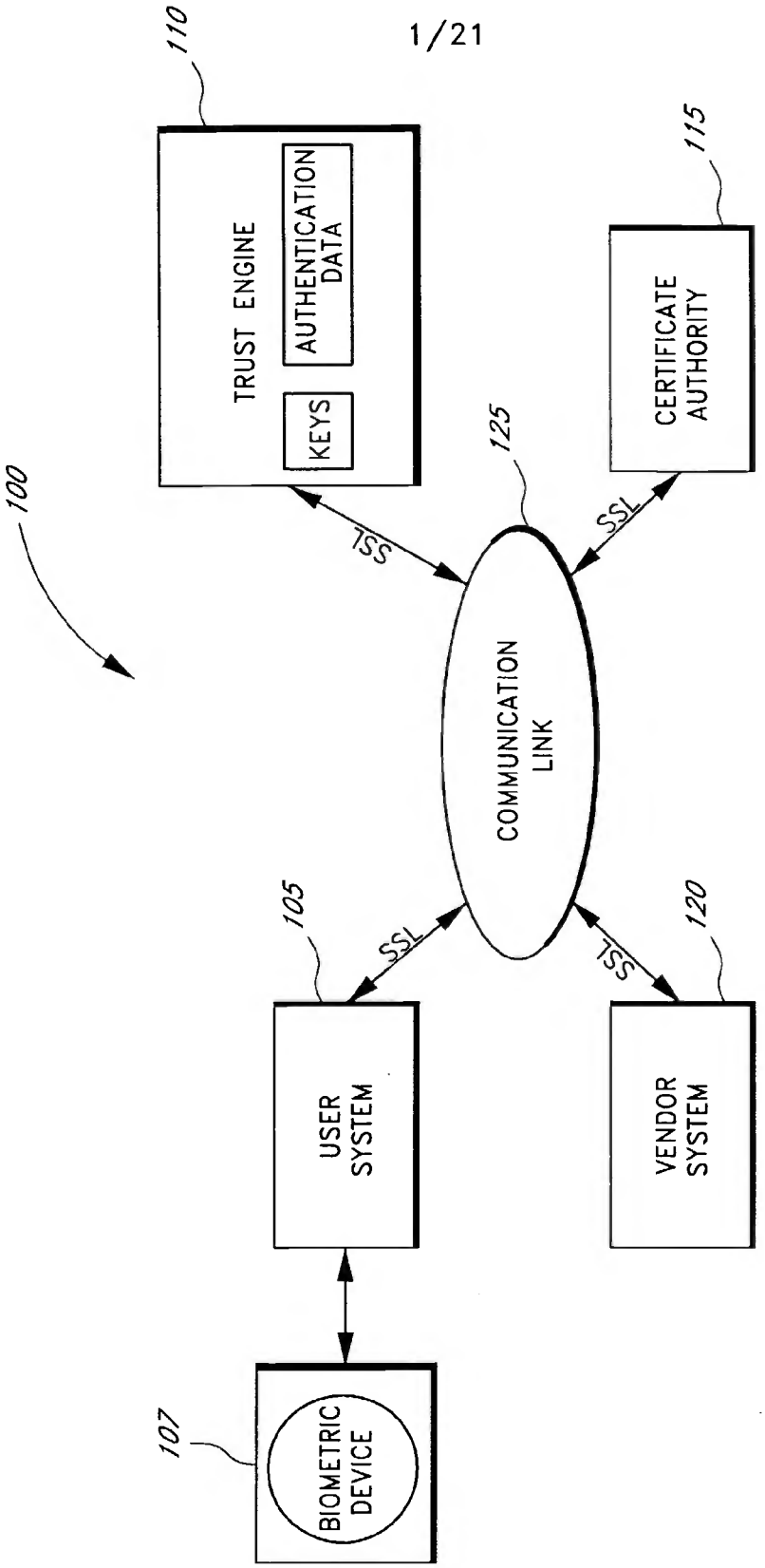

110
TRUST ENGINE
AUTHENTICATION DATA
KEYS
115
CERTIFICATE AUTHORITY
100
125
SSL
SSL
COMMUNICATION LINK
105
SSL
SSL
120
USER SYSTEM
VENDOR SYSTEM
107
BIOMETRIC DEVICE

FIG. 1

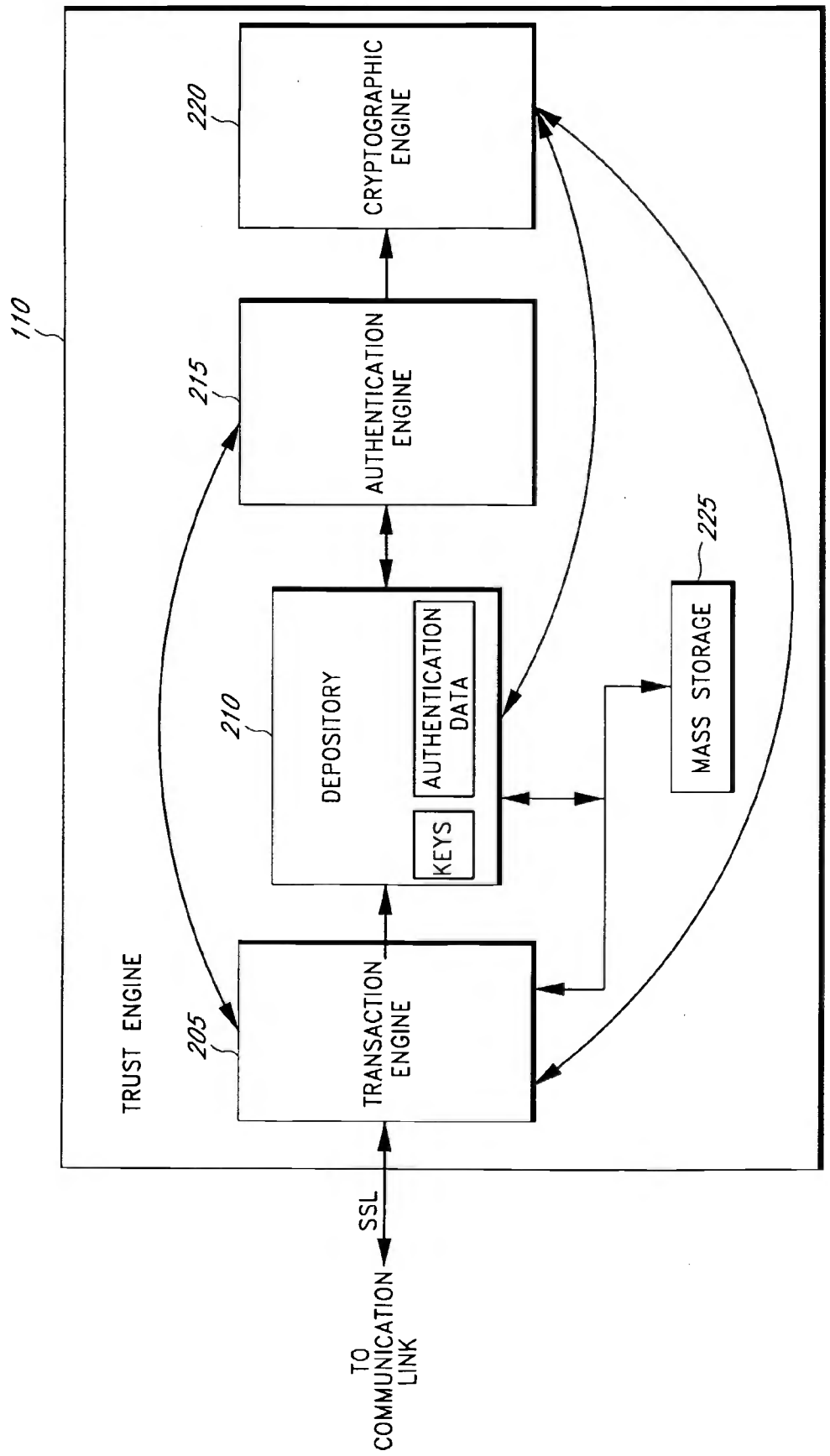

110
TRUST ENGINE
220
CRYPTOGRAPHIC ENGINE
215
AUTHENTICATION ENGINE
210
DEPOSITORY
AUTHENTICATION DATA
KEYS
225
MASS STORAGE
205
TRANSACTION ENGINE
SSL
TO COMMUNICATION LINK

FIG. 2

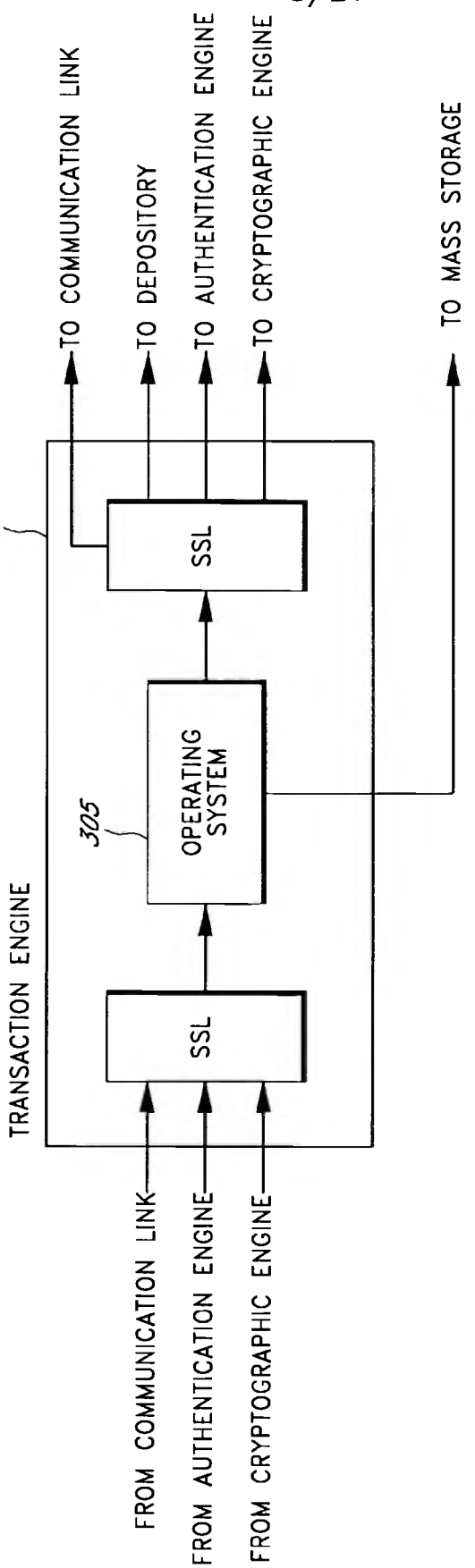

TO COMMUNICATION LINK
TO DEPOSITORY
TO AUTHENTICATION ENGINE
TO CRYPTOGRAPHIC ENGINE
TO MASS STORAGE
205
SSL
OPERATING SYSTEM
305
SSL
TRANSACTION ENGINE
FROM COMMUNICATION LINK
FROM AUTHENTICATION ENGINE
FROM CRYPTOGRAPHIC ENGINE

FIG. 3

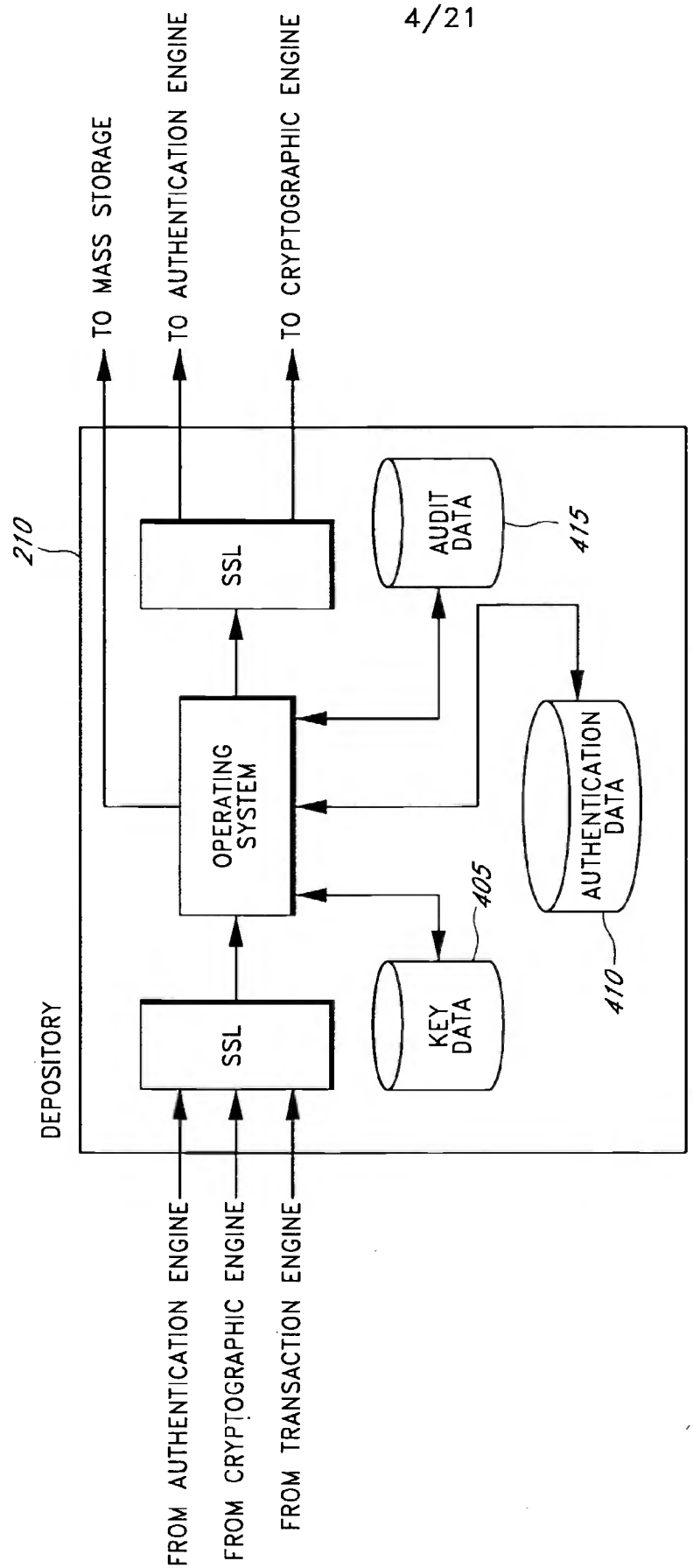

TO MASS STORAGE
TO AUTHENTICATION ENGINE
TO CRYPTOGRAPHIC ENGINE
210
SSL
AUDIT DATA
415
OPERATING SYSTEM
AUTHENTICATION DATA
405
KEY DATA
410
SSL
DEPOSITORY
FROM AUTHENTICATION ENGINE
FROM CRYPTOGRAPHIC ENGINE
FROM TRANSACTION ENGINE

FIG. 4

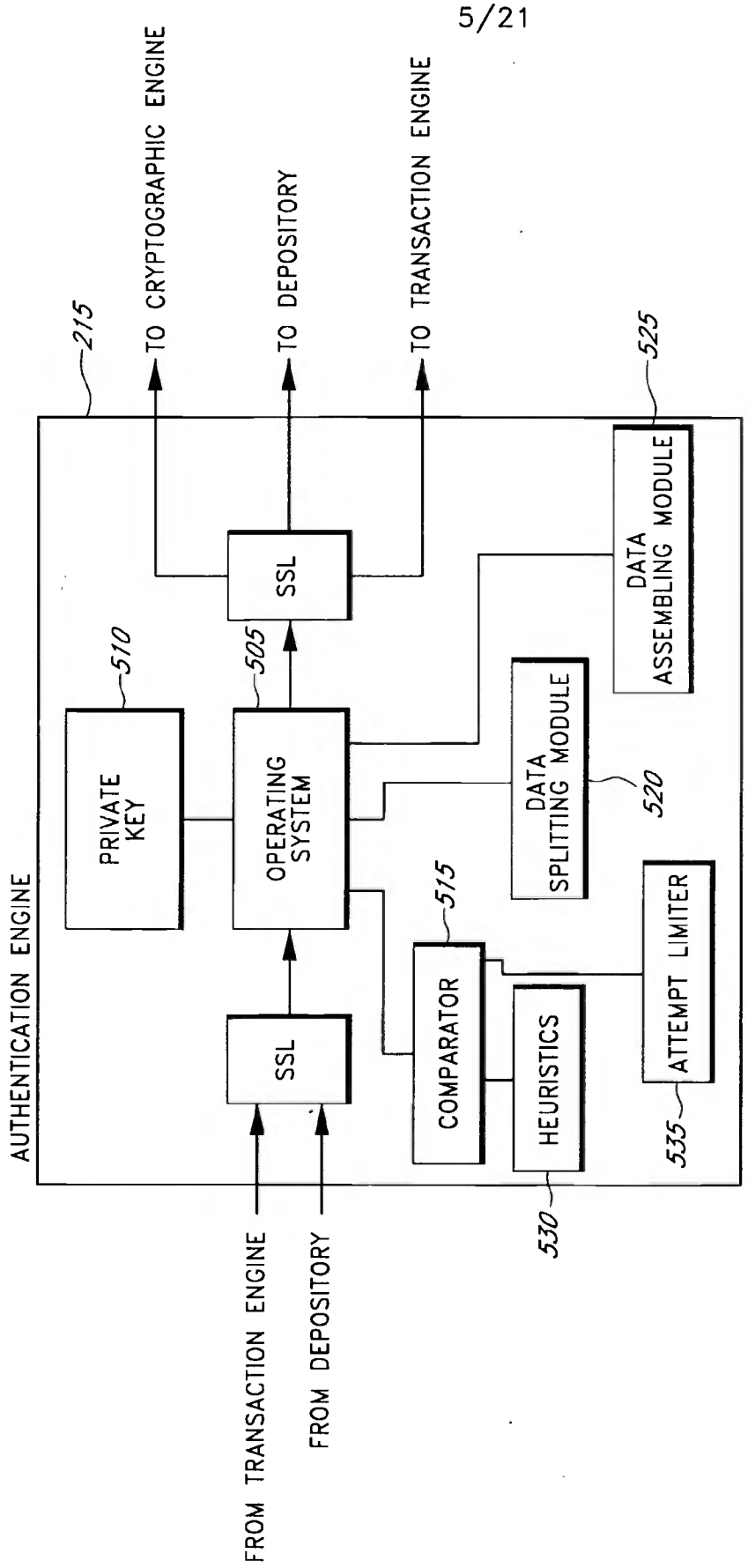

AUTHENTICATION ENGINE
215
TO CRYPTOGRAPHIC ENGINE
TO DEPOSITORY
TO TRANSACTION ENGINE
525
DATA ASSEMBLING MODULE
SSL
505
510
PRIVATE KEY
OPERATING SYSTEM
DATA SPLITTING MODULE
520
515
COMPARATOR
HEURISTICS
ATTEMPT LIMITER
535
530
SSL
FROM TRANSACTION ENGINE
FROM DEPOSITORY

FIG. 5

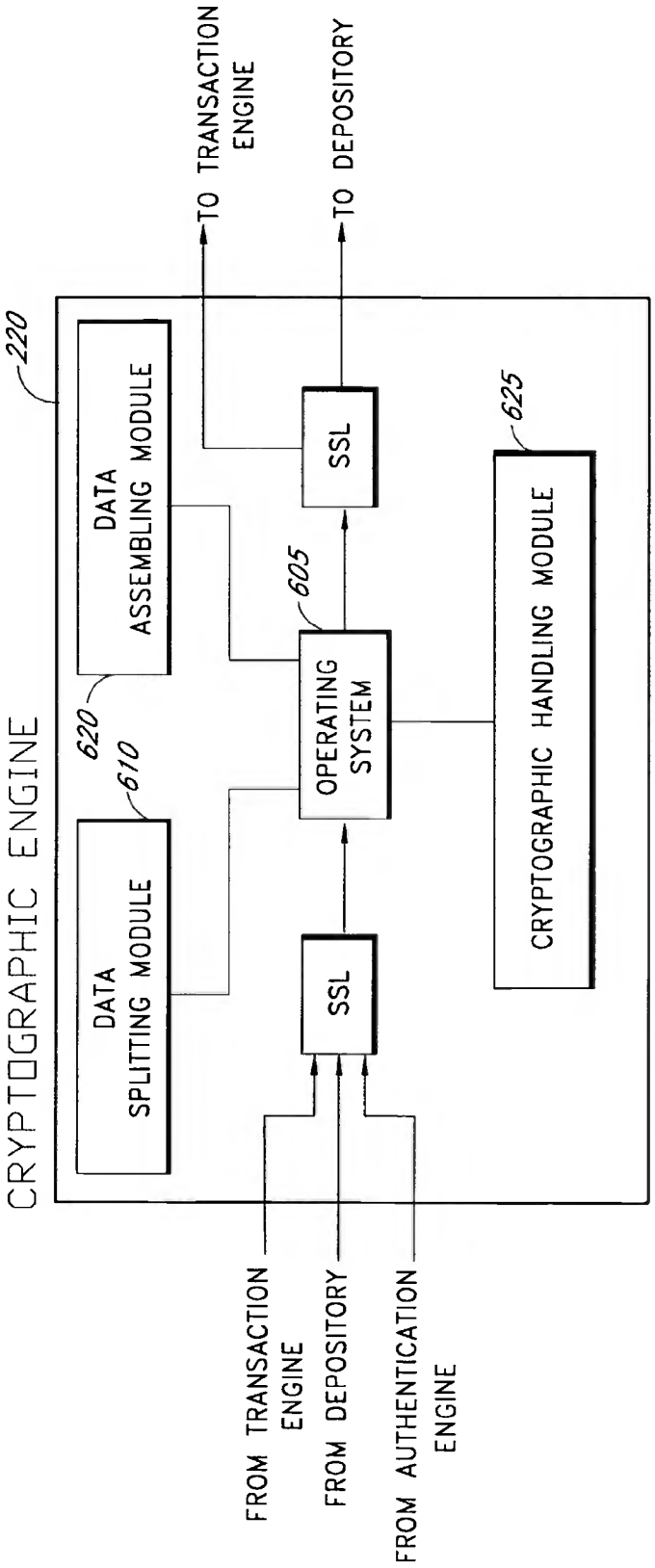

TO TRANSACTION ENGINE
TO DEPOSITORY
220
CRYPTOGRAPHIC ENGINE
DATA ASSEMBLING MODULE
620
610
DATA SPLITTING MODULE
605
OPERATING SYSTEM
SSL
SSL
625
CRYPTOGRAPHIC HANDLING MODULE
FROM TRANSACTION ENGINE
FROM DEPOSITORY
FROM AUTHENTICATION ENGINE

FIG. 6

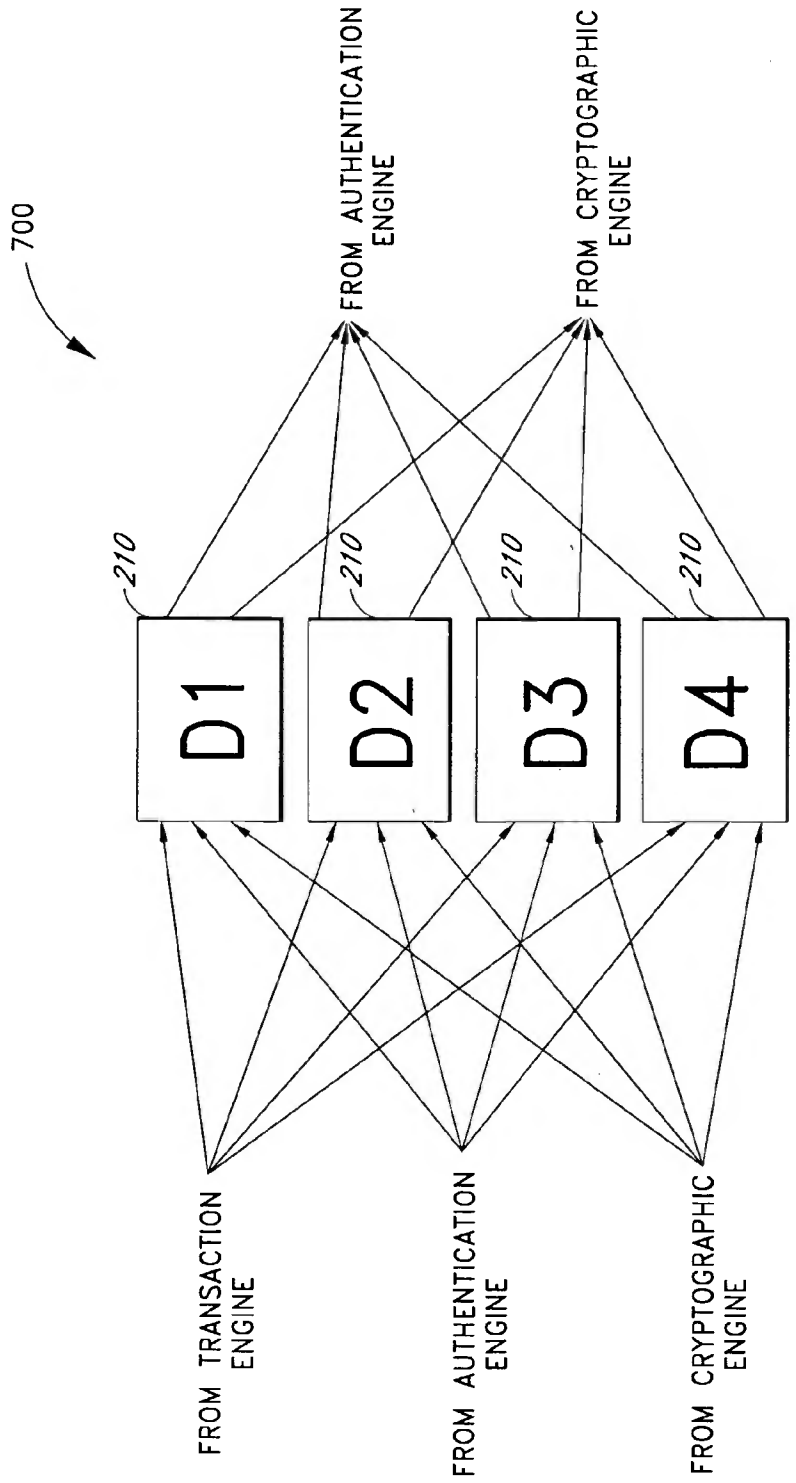

700
FROM AUTHENTICATION ENGINE
FROM CRYPTOGRAPHIC ENGINE
210
210
210
210
D1
D2
D3
D4
FROM TRANSACTION ENGINE
FROM AUTHENTICATION ENGINE
FROM CRYPTOGRAPHIC ENGINE

FIG. 7

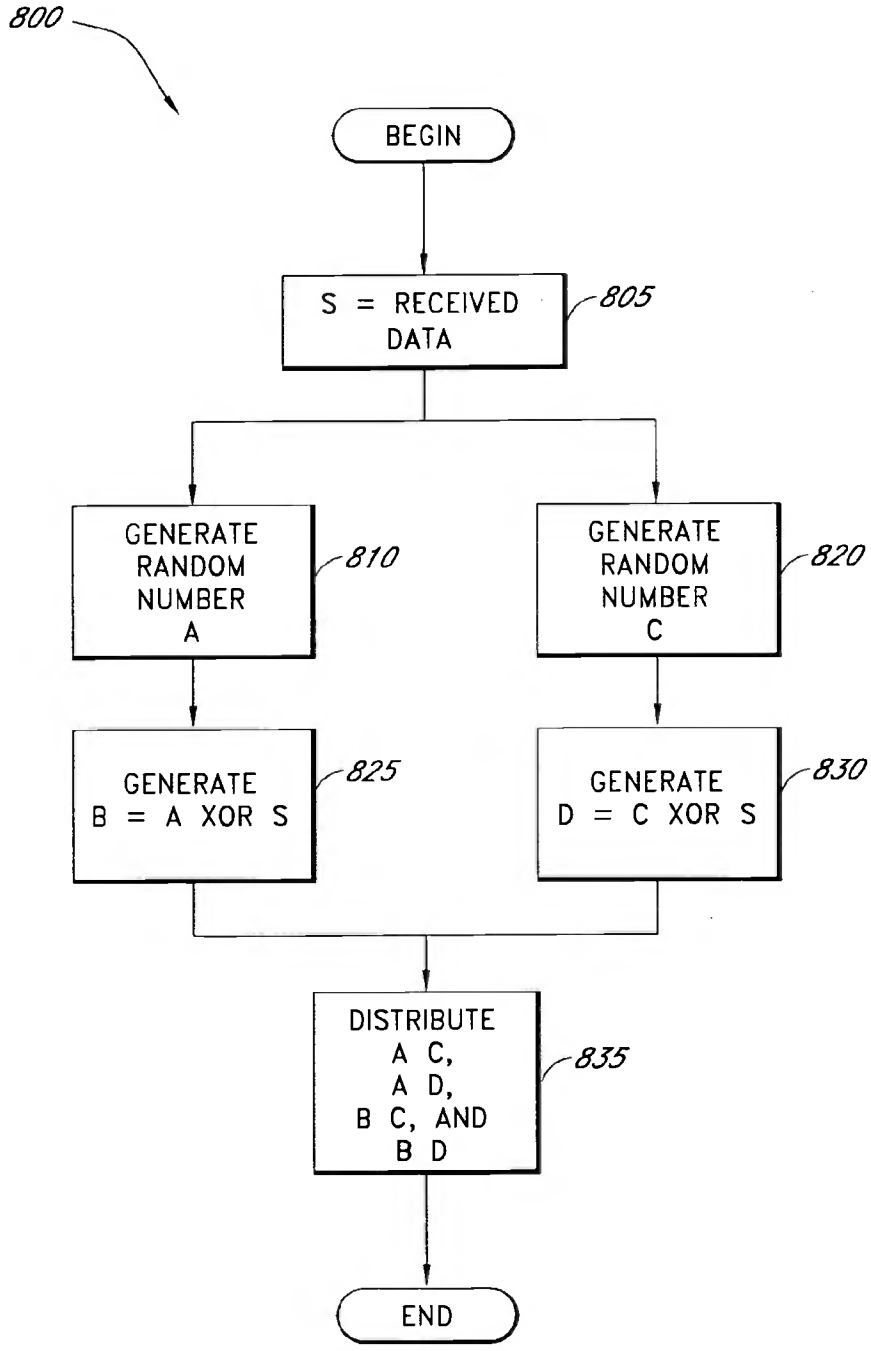

800
BEGIN
S = RECEIVED DATA
805
GENERATE RANDOM NUMBER A
810
GENERATE RANDOM NUMBER C
820
GENERATE B = A XOR S
825
GENERATE D = C XOR S
830
DISTRIBUTE A C, A D, B C, AND B D
835
END

FIG. 8

900

	ENROLLMENT DATA FLOW			
	SEND	RECEIVE	SSL	ACTION
905	USER	TRANSACTION ENGINE(TE)	1/2	TRANSMIT ENROLLMENT AUTHENTICATION DATA (B) AND THE USER ID (UID) ENCRYPTED WITH THE PUBLIC KEY OF THE AUTHENTICATION ENGINE (AE) AS (PUB_AE(UID,B))
915	TE	AE	FULL	FORWARD TRANSMISSION
920				AE DECRYPTS AND SPLITS FORWARDED DATA
925	AE	THE Xth DEPOSITORY (DX)	FULL	STORE RESPECTIVE PORTION OF DATA
	WHEN DIGITAL CERTIFICATE REQUESTED			
930	AE	CRYPTOGRAPHIC ENGINE (CE)	FULL	REQUEST KEY GENERATION
935				CE GENERATES AND SPLITS KEY
945	CE	TE	FULL	TRANSMIT REQUEST FOR DIGITAL CERTIFICATE
950	TE	CERTIFICATION AUTHORITY (CA)	1/2	TRANSMIT REQUEST
955	CA	TE	1/2	TRANSMIT DIGITAL CERTIFICATE
960	TE	USER	1/2	TRANSMIT DIGITAL CERTIFICATE
	TE	MS	FULL	STORE DIGITAL CERTIFICATE
965	TE	DX	FULL	STORE RESPECTIVE PORTION OF KEY

FIG. 9A

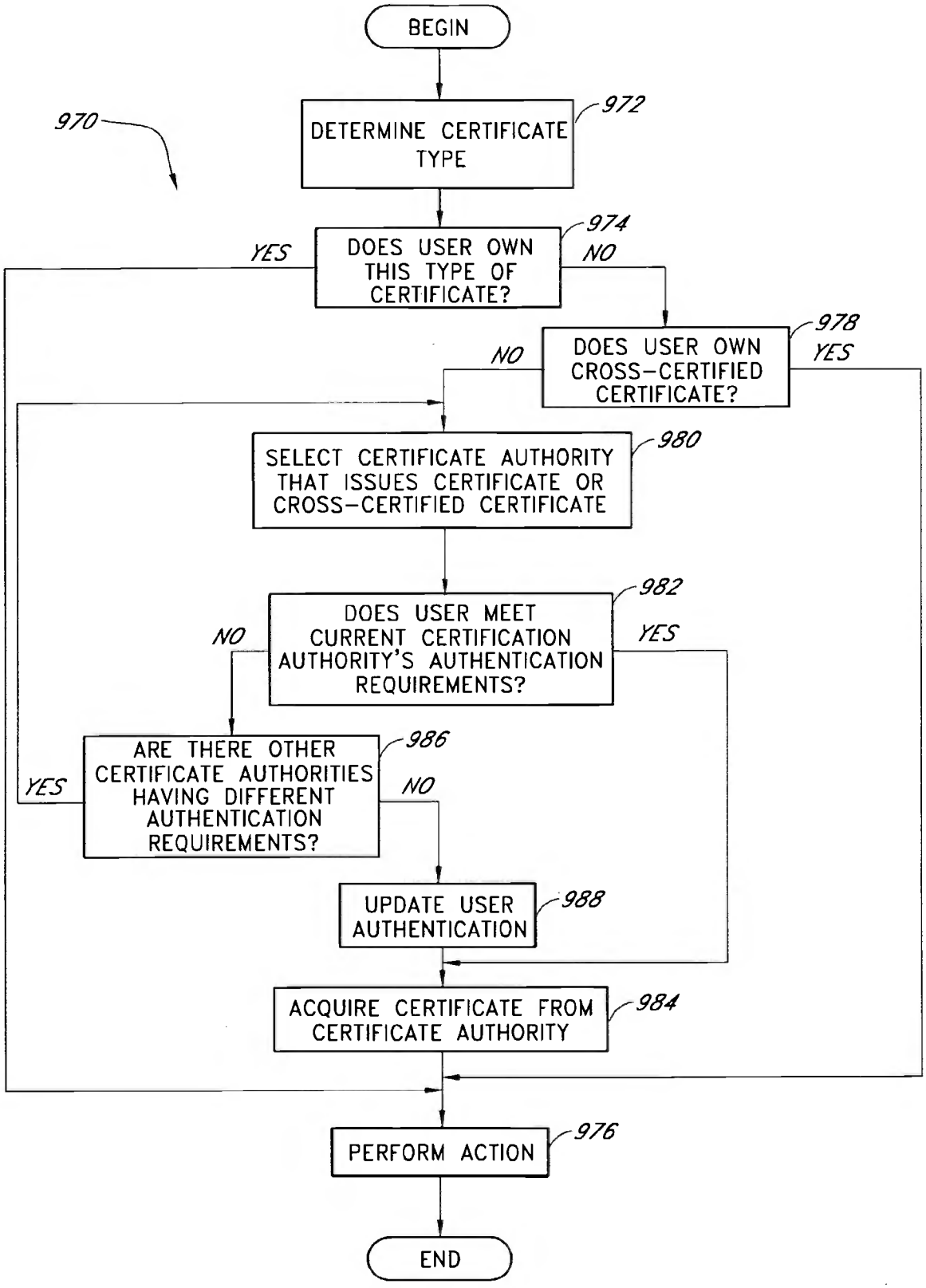
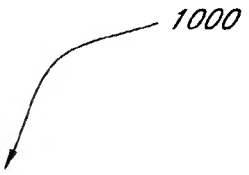

BEGIN
970
DETERMINE CERTIFICATE TYPE
972
DOES USER OWN THIS TYPE OF CERTIFICATE?
974
YES
NO
DOES USER OWN CROSS-CERTIFIED CERTIFICATE?
978
NO
YES
SELECT CERTIFICATE AUTHORITY THAT ISSUES CERTIFICATE OR CROSS-CERTIFIED CERTIFICATE
980
DOES USER MEET CURRENT CERTIFICATION AUTHORITY'S AUTHENTICATION REQUIREMENTS?
982
NO
YES
ARE THERE OTHER CERTIFICATE AUTHORITIES HAVING DIFFERENT AUTHENTICATION REQUIREMENTS?
986
YES
NO
UPDATE USER AUTHENTICATION
988
ACQUIRE CERTIFICATE FROM CERTIFICATE AUTHORITY
984
PERFORM ACTION
976
END

FIG. 9B



AUTHENTICATION DATA FLOW

	SEND	RECEIVE	SSL	ACTION
1005	USER	VENDOR	1/2	TRANSACTION OCCURS, SUCH AS SELECTING PURCHASE
1010	VENDOR	USER	1/2	TRANSMIT TRANSACTION ID (TID) AND AUTHENTICATION REQUEST (AR)
				AUTHENTICATION DATA (B') IS GATHERED FROM USER
1015	USER	TE	1/2	TRANSMIT TID AND B' WRAPPED IN THE PUBLIC KEY OF THE AUTHENTICATION ENGINE (AE), AS (PUB_AE(TID, B'))
1020	TE	AE	FULL	FORWARD TRANSMISSION
				ENROLLMENT AUTHENTICATION DATA (B) IS REQUESTED AND GATHERED
1025	VENDOR	TRANSACTION ENGINE (TE)	FULL	TRANSMITS TID, AR
1030	TE	MASS STORAGE(MS)	FULL	CREATE RECORD IN DATABASE
1035	TE	THE Xth DEPOSITORY(DX)	FULL	UID, TID
1040	DX	AE	FULL	TRANSMIT THE TID AND THE PORTION OF THE AUTHENTICATION DATA STORED AT ENROLLMENT (BX) AS (PUB_AE(TID, BX))
1045				AE ASSEMBLES B AND COMPARES TO B'
1050	AE	TE	FULL	TID, THE FILLED IN AR
	TE	VENDOR	FULL	TID, YES/NO
1055	TE	USER	1/2	TID, CONFIRMATION MESSAGE

FIG. 10

1100

SIGNING DATA FLOW			
SEND	RECEIVE	SSL	ACTION
USER	VENDOR	1/2	TRANSACTION OCCURS, SUCH AS AGREEING ON A DEAL
VENDOR	USER	1/2	TRANSMIT TRANSACTION IDENTIFICATION NUMBER (TID), AUTHENTICATION REQUEST (AR), AND AGREEMENT OR MESSAGE (M)
			CURRENT AUTHENTICATION DATA (B') AND A HASH OF THE MESSAGE RECEIVED BY THE USER (h(M')) IS GATHERED FROM USER
USER	TE	1/2	TRANSMIT TID, B', AR, AND h(M') WRAPPED IN THE PUBLIC KEY OF THE AUTHENTICATION ENGINE (AE) AS (PUB_AE(TID, B', h(M'))
TE	AE	FULL	FORWARD TRANSMISSION
			GATHER ENROLLMENT AUTHENTICATION DATA
VENDOR	TRANSACTION ENGINE (TE)	FULL	TRANSMITS UID, TID, AR, AND A HASH OF THE MESSAGE (h(M)).
TE	MASS STORAGE (MS)	FULL	CREATE RECORD IN DATABASE
TE	THE Xth DEPOSITORY(DX)	FULL	UID, TID
DX	AE	FULL	TRANSMIT THE TID AND THE PORTION OF THE AUTHENTICATION DATA STORED AT ENROLLMENT (BX),AS (PUB_AE(TID, BX))
			THE ORIGINAL VENDOR MESSAGE IS TRANSMITTED TO THE AE
TE	AE	FULL	TRANSMIT h(M)
			AE ASSEMBLES B, COMPARES TO B' AND COMPARES h(M) TO h(M')
AE	CRYPTOGRAPHIC ENGINE (CE)	FULL	REQUEST FOR DIGITAL SIGNATURE AND A MESSAGE TO BE SIGNED, FOR EXAMPLE, THE HASHED MESSAGE
AE	DX	FULL	TID, SIGNING UID
DX	CE	FULL	TRANSMIT THE PORTION OF THE CRYPTOGRAPHIC KEY CORRESPONDING TO THE SIGNING PARTY
			CE ASSEMBLES KEY AND SIGNS
CE	AE	FULL	TRANSMIT THE DIGITAL SIGNATURE (S) OF SIGNING PARTY
AE	TE	FULL	TID, THE FILLED IN AR, h(M), AND S
TE	VENDOR	FULL	TID, A RECEIPT=(TID, YES/NO, AND S), AND THE DIGITAL SIGNATURE OF THE TRUST ENGINE, FOR EXAMPLE, A HASH OF THE RECEIPT ENCRYPTED WITH THE TRUST ENGINE'S PRIVATE KEY (Priv_TE(h(RECEIPT)))
TE	USER	1/2	TID, CONFIRMATION MESSAGE

1103 1105 1110 1115 1120 1125 1130 1135 1140

FIG. 11

1200

ENCRYPTION/DECRYPTION DATA FLOW			
SEND	RECEIVE	SSL	ACTION
DECRYPTION			
			PERFORM AUTHENTICATION DATA PROCESS 1000, INCLUDE THE SESSION KEY (SYNC) IN THE AR, WHERE THE SYNC HAS BEEN ENCRYPTED WITH THE PUBLIC KEY OF THE USER AS PUB_USER(SNYC)
			AUTHENTICATE THE USER
1205 AE	CE	FULL	FORWARD PUB_USER(SYNC) TO CE
1210 AE	DX	FULL	UID, TID
1215 DX	CE	FULL	TRANSMIT THE TID AND THE PORTION OF THE PRIVATE KEY AS (PUB_AE(TID, KEY_USER))
1220			CE ASSEMBLES THE CRYPTIOGRAPHIC KEY AND DECRYPTS THE SYNC
1225 CE	AE	FULL	TID, THE FILLED IN AR INCLUDING DECRYPTED SYNC
1230 AE	TE	FULL	FORWARD TO TE
TE	REQUESTING APP/VENDOR	1/2	TID, YES/NO, SYNC
ENCRYPTION			
1235 REQUESTING APP/VENDOR	TE	1/2	1240 REQUEST FOR PUBLIC KEY OF USER
1245 TE	MS	FULL	REQUEST DIGITAL CERTIFICATE
MS	TE	FULL	TRANSMIT DIGITAL CERTIFICATE
1250 TE	REQUESTING APP/VENDOR	1/2	TRANSMIT DIGITAL CERTIFICATE

FIG. 12

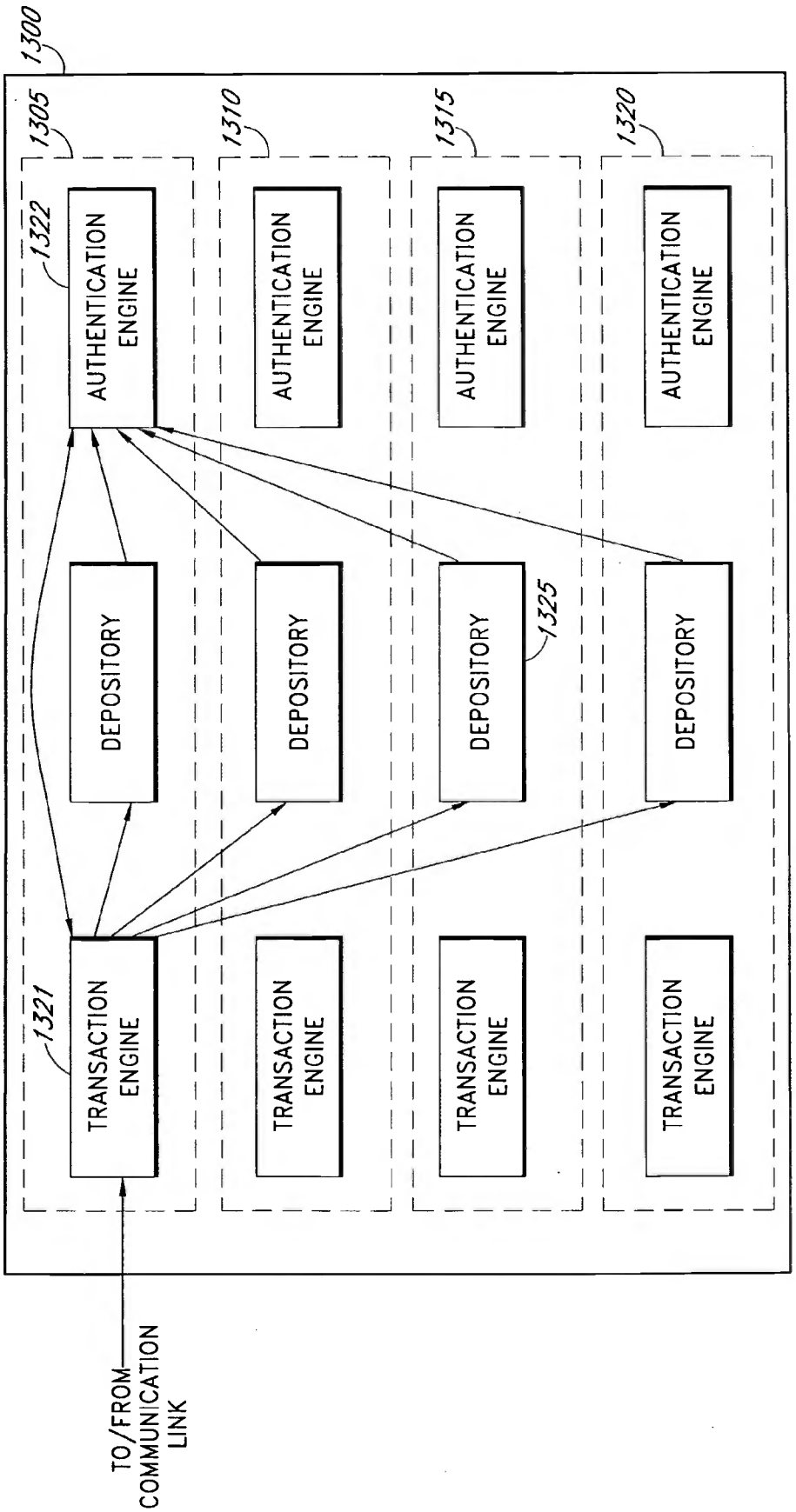

1300
1305
1310
1315
1320
1322
AUTHENTICATION ENGINE
AUTHENTICATION ENGINE
AUTHENTICATION ENGINE
AUTHENTICATION ENGINE
DEPOSITORY
DEPOSITORY
DEPOSITORY
DEPOSITORY
1325
1321
TRANSACTION ENGINE
TRANSACTION ENGINE
TRANSACTION ENGINE
TRANSACTION ENGINE
TO/FROM COMMUNICATION LINK

FIG. 13

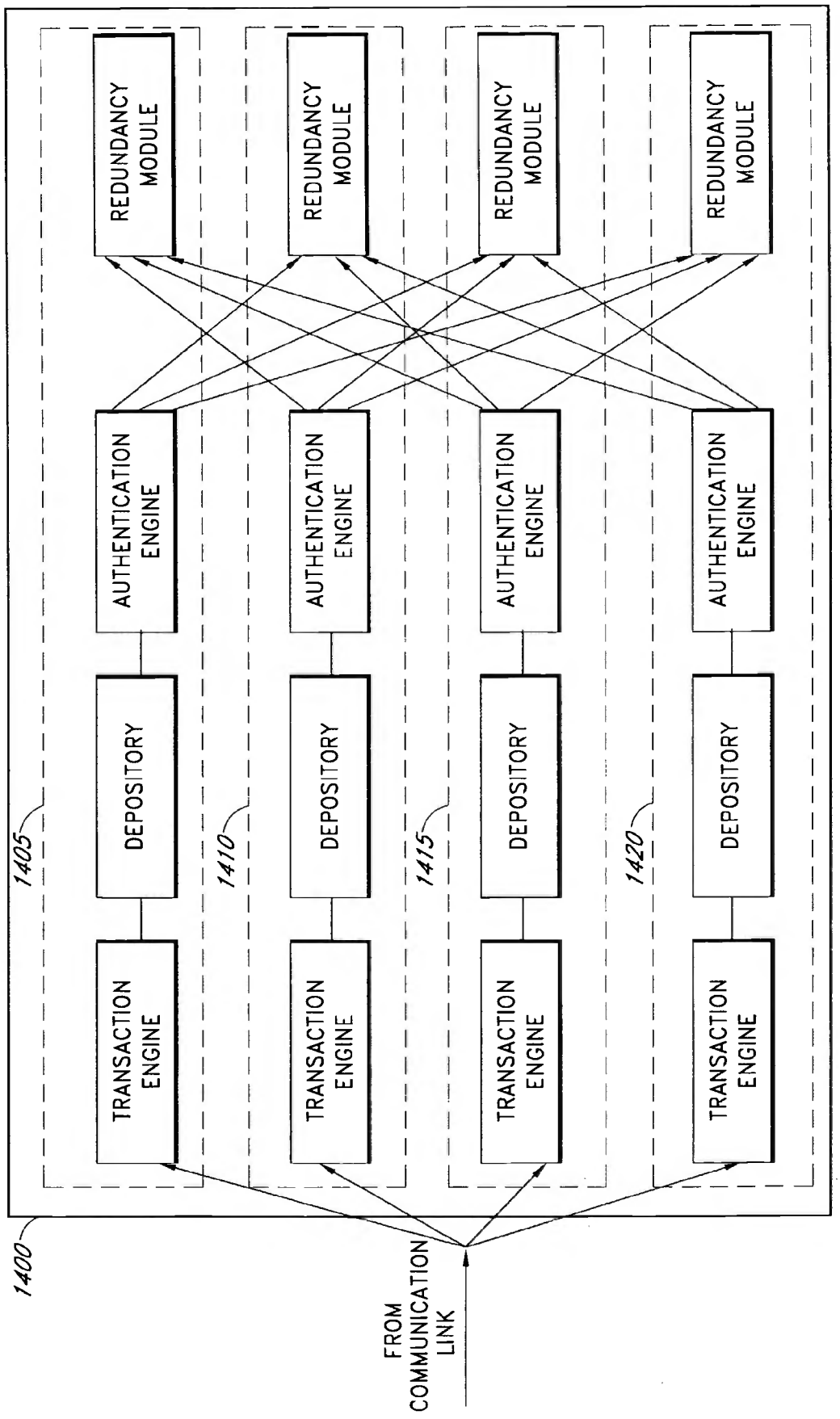

REDUNDANCY MODULE
REDUNDANCY MODULE
REDUNDANCY MODULE
REDUNDANCY MODULE
AUTHENTICATION ENGINE
AUTHENTICATION ENGINE
AUTHENTICATION ENGINE
AUTHENTICATION ENGINE
DEPOSITORY
DEPOSITORY
DEPOSITORY
DEPOSITORY
TRANSACTION ENGINE
TRANSACTION ENGINE
TRANSACTION ENGINE
TRANSACTION ENGINE
1405
1410
1415
1420
1400
FROM COMMUNICATION LINK

FIG. 14

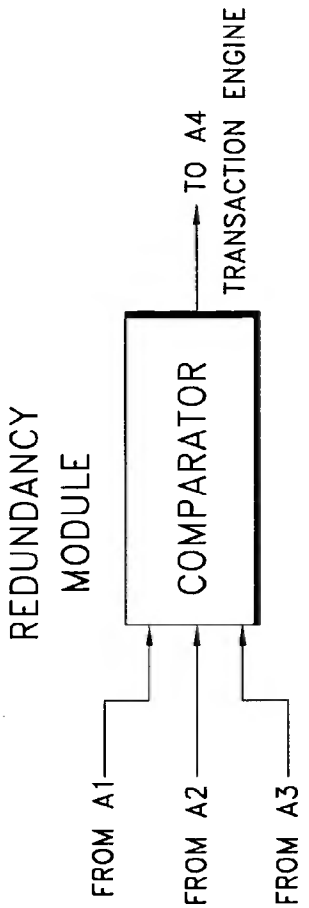

REDUNDANCY
MODULE
COMPARATOR
TO A4
TRANSACTION ENGINE
FROM A1
FROM A2
FROM A3

FIG. 15

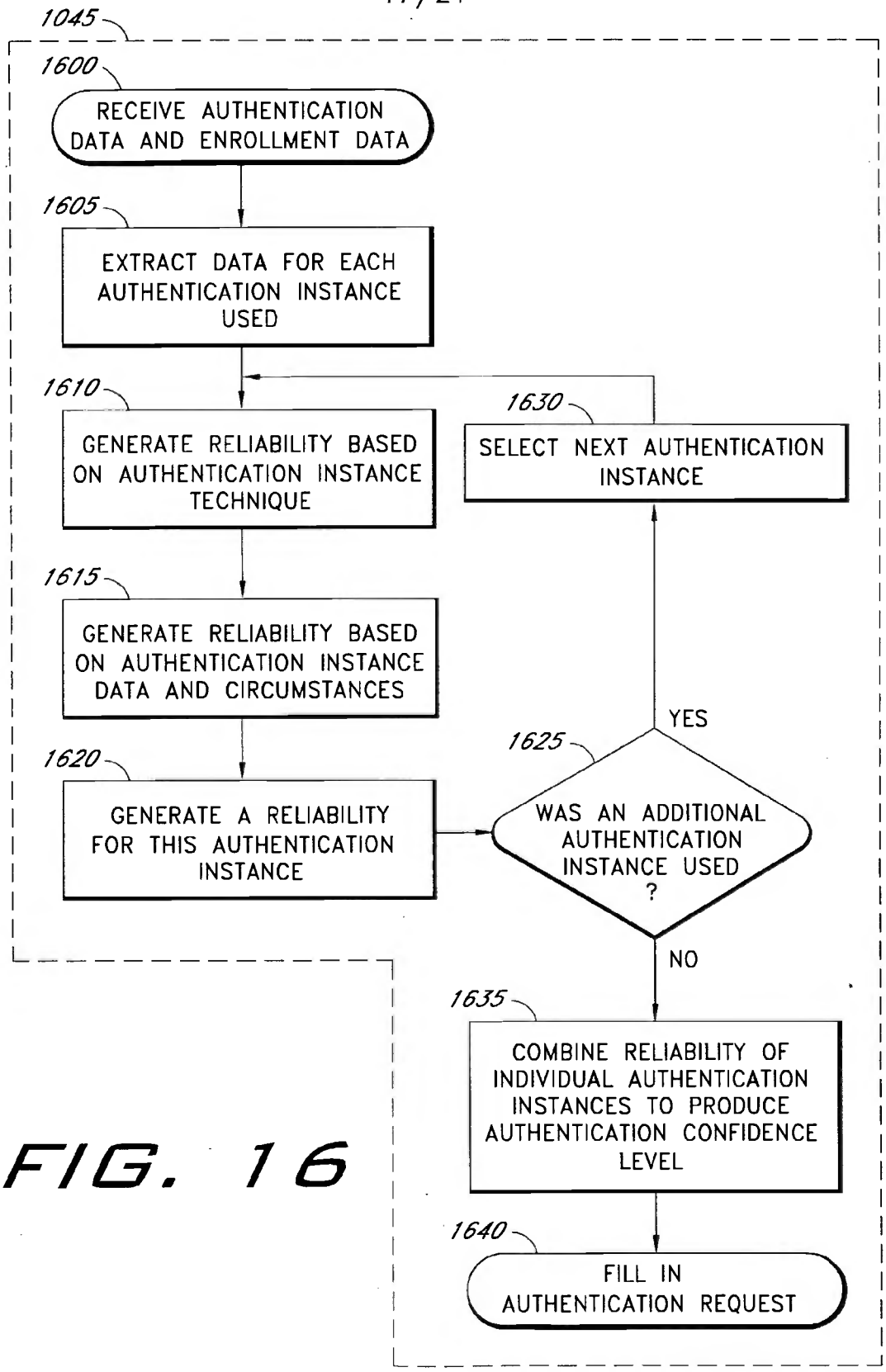

1045
1600
RECEIVE AUTHENTICATION DATA AND ENROLLMENT DATA
1605
EXTRACT DATA FOR EACH AUTHENTICATION INSTANCE USED
1610
GENERATE RELIABILITY BASED ON AUTHENTICATION INSTANCE TECHNIQUE
1615
GENERATE RELIABILITY BASED ON AUTHENTICATION INSTANCE DATA AND CIRCUMSTANCES
1620
GENERATE A RELIABILITY FOR THIS AUTHENTICATION INSTANCE
1625
WAS AN ADDITIONAL AUTHENTICATION INSTANCE USED ?
YES
NO
1630
SELECT NEXT AUTHENTICATION INSTANCE
1635
COMBINE RELIABILITY OF INDIVIDUAL AUTHENTICATION INSTANCES TO PRODUCE AUTHENTICATION CONFIDENCE LEVEL
1640
FILL IN AUTHENTICATION REQUEST

FIG. 16

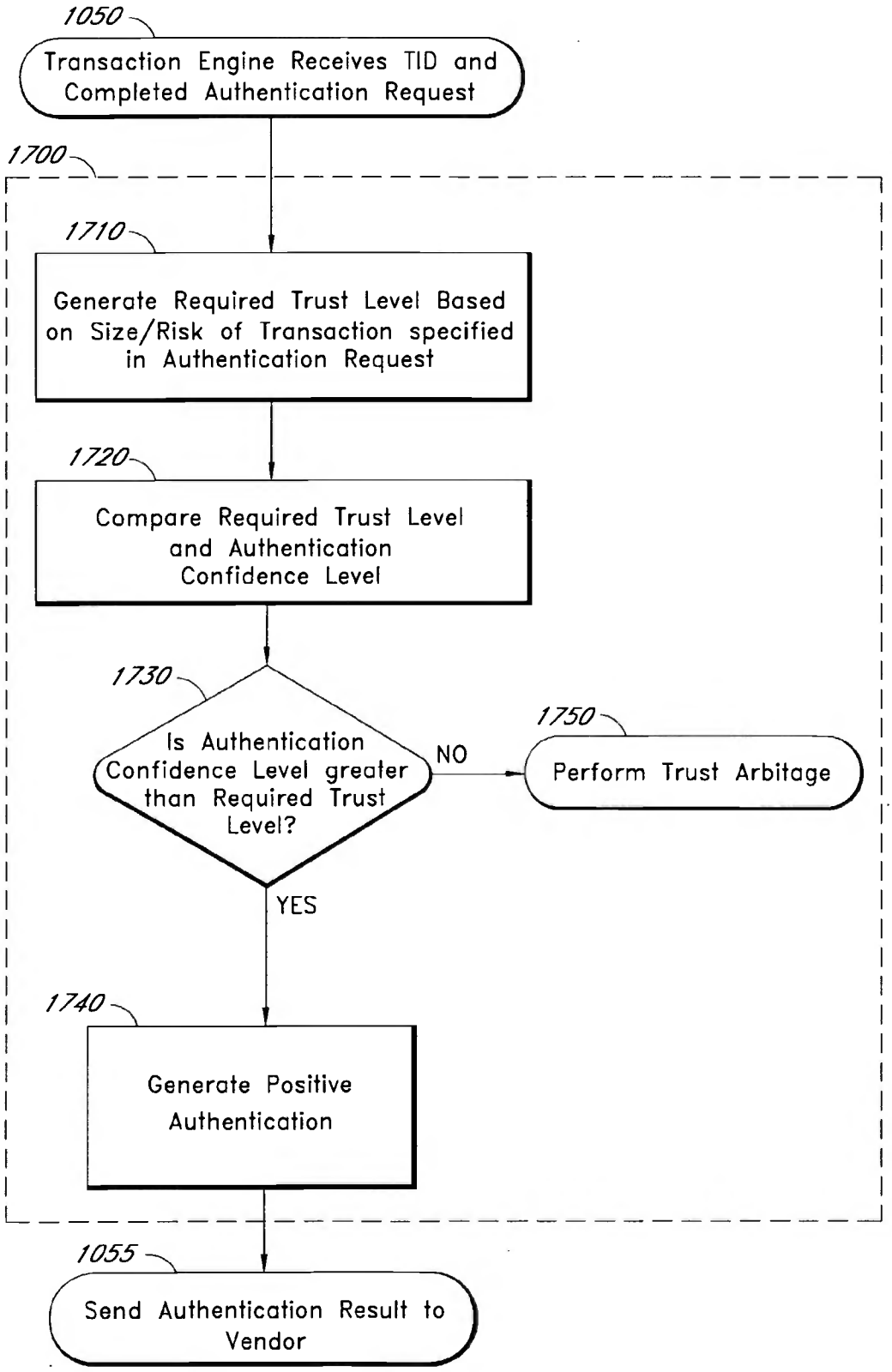

1050
Transaction Engine Receives TID and Completed Authentication Request
1700
1710
Generate Required Trust Level Based on Size/Risk of Transaction specified in Authentication Request
1720
Compare Required Trust Level and Authentication Confidence Level
1730
Is Authentication Confidence Level greater than Required Trust Level?
NO
1750
Perform Trust Arbitage
YES
1740
Generate Positive Authentication
1055
Send Authentication Result to Vendor

FIG. 17

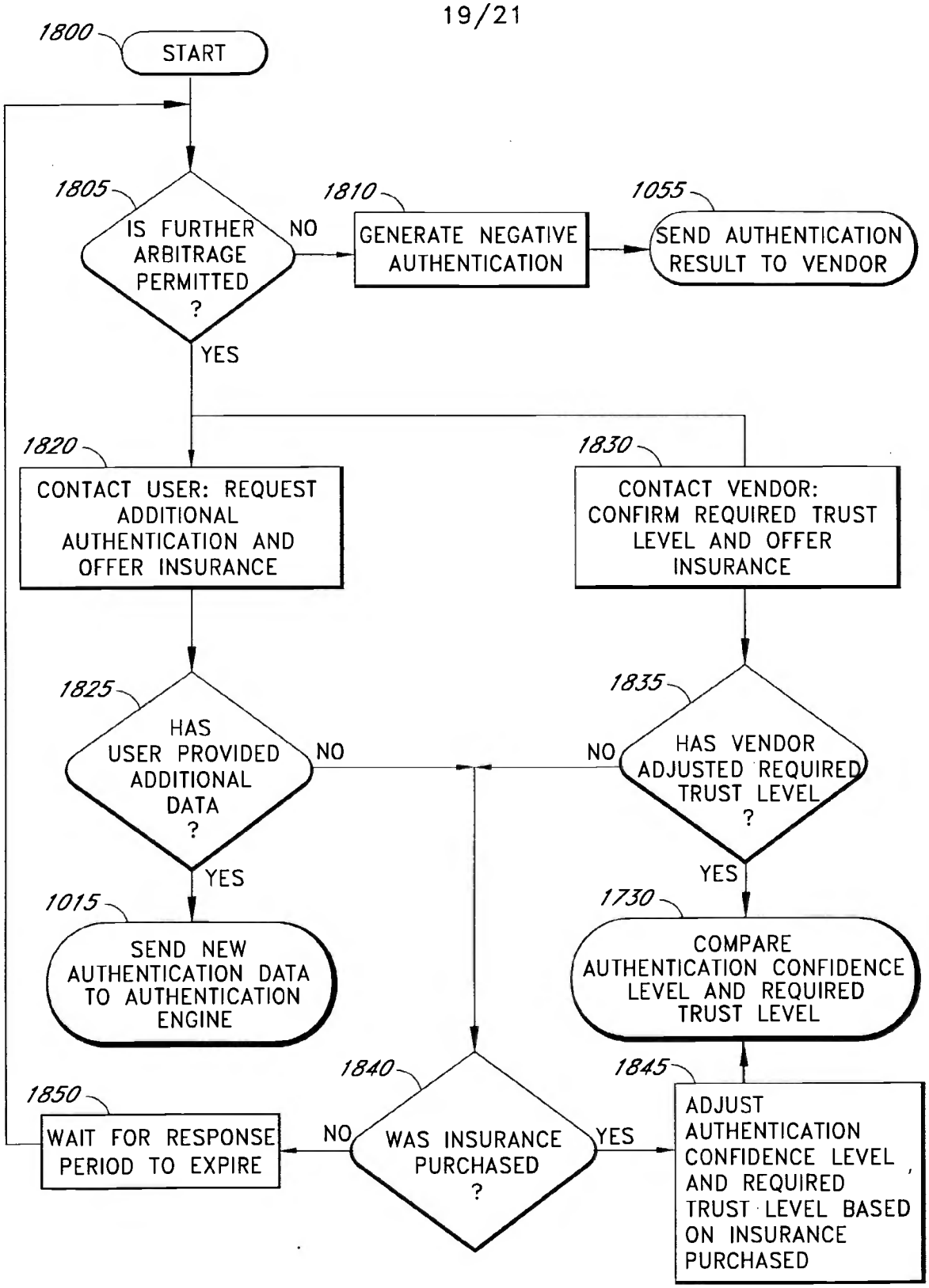

1800
START
1805
IS FURTHER ARBITRAGE PERMITTED ?
NO
1810
GENERATE NEGATIVE AUTHENTICATION
1055
SEND AUTHENTICATION RESULT TO VENDOR
YES
1820
CONTACT USER: REQUEST ADDITIONAL AUTHENTICATION AND OFFER INSURANCE
1830
CONTACT VENDOR: CONFIRM REQUIRED TRUST LEVEL AND OFFER INSURANCE
1825
HAS USER PROVIDED ADDITIONAL DATA ?
NO
1835
HAS VENDOR ADJUSTED REQUIRED TRUST LEVEL ?
NO
YES
YES
1015
SEND NEW AUTHENTICATION DATA TO AUTHENTICATION ENGINE
1730
COMPARE AUTHENTICATION CONFIDENCE LEVEL AND REQUIRED TRUST LEVEL
1840
WAS INSURANCE PURCHASED ?
1845
ADJUST AUTHENTICATION CONFIDENCE LEVEL AND REQUIRED TRUST LEVEL BASED ON INSURANCE PURCHASED
1850
WAIT FOR RESPONSE PERIOD TO EXPIRE
NO
YES

FIG. 18

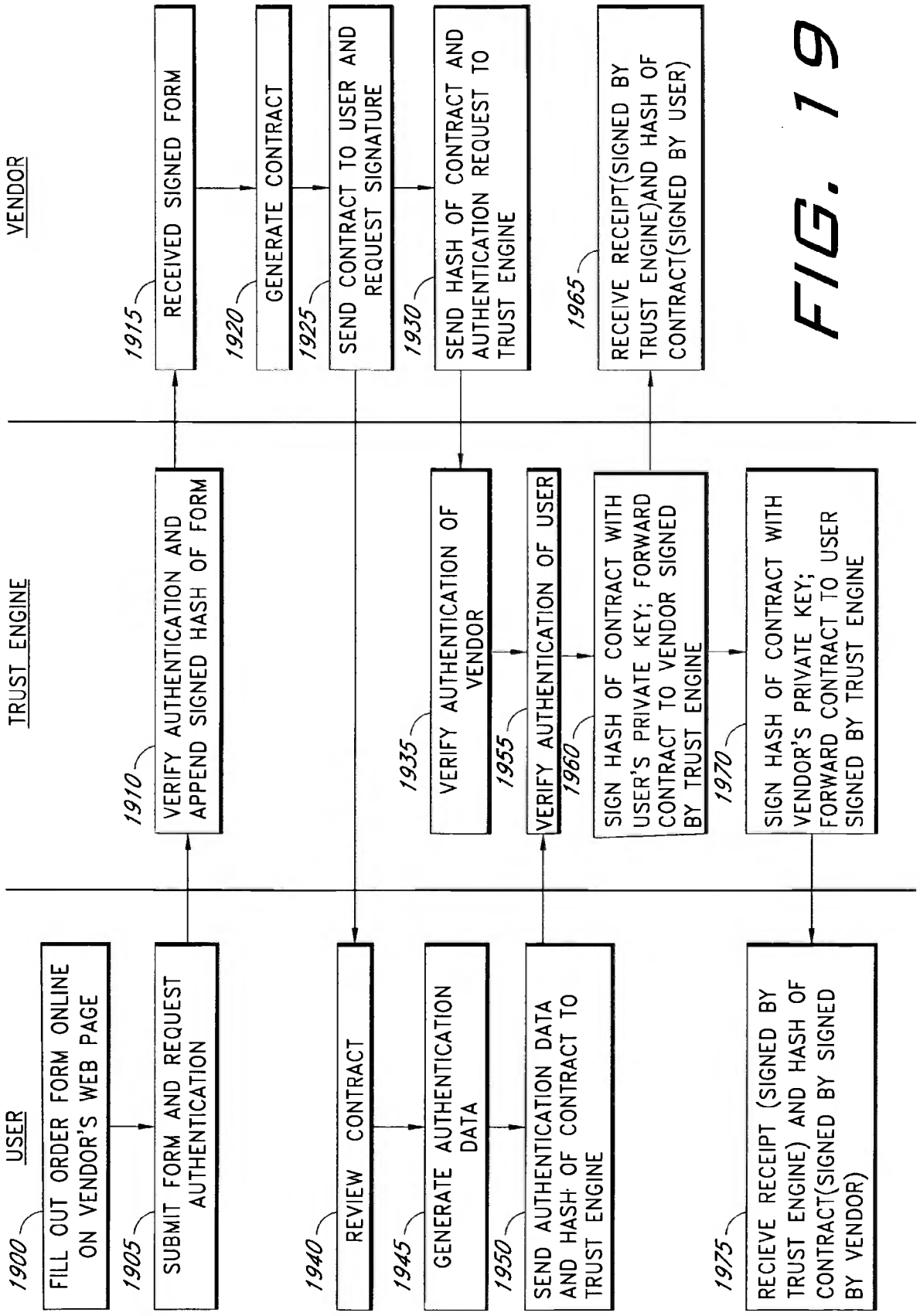

USER
TRUST ENGINE
VENDOR
1900
FILL OUT ORDER FORM ONLINE ON VENDOR'S WEB PAGE
1905
SUBMIT FORM AND REQUEST AUTHENTICATION
1910
VERIFY AUTHENTICATION AND APPEND SIGNED HASH OF FORM
1915
RECEIVED SIGNED FORM
1920
GENERATE CONTRACT
1925
SEND CONTRACT TO USER AND REQUEST SIGNATURE
1930
SEND HASH OF CONTRACT AND AUTHENTICATION REQUEST TO TRUST ENGINE
1935
VERIFY AUTHENTICATION OF VENDOR
1940
REVIEW CONTRACT
1945
GENERATE AUTHENTICATION DATA
1950
SEND AUTHENTICATION DATA AND HASH OF CONTRACT TO TRUST ENGINE
1955
VERIFY AUTHENTICATION OF USER
1960
SIGN HASH OF CONTRACT WITH USER'S PRIVATE KEY; FORWARD CONTRACT TO VENDOR SIGNED BY TRUST ENGINE
1965
RECEIVE RECEIPT(SIGNED BY TRUST ENGINE)AND HASH OF CONTRACT(SIGNED BY USER)
1970
SIGN HASH OF CONTRACT WITH VENDOR'S PRIVATE KEY; FORWARD CONTRACT TO USER SIGNED BY TRUST ENGINE
1975
RECIEVE RECEIPT (SIGNED BY TRUST ENGINE) AND HASH OF CONTRACT(SIGNED BY SIGNED BY VENDOR)

FIG. 19

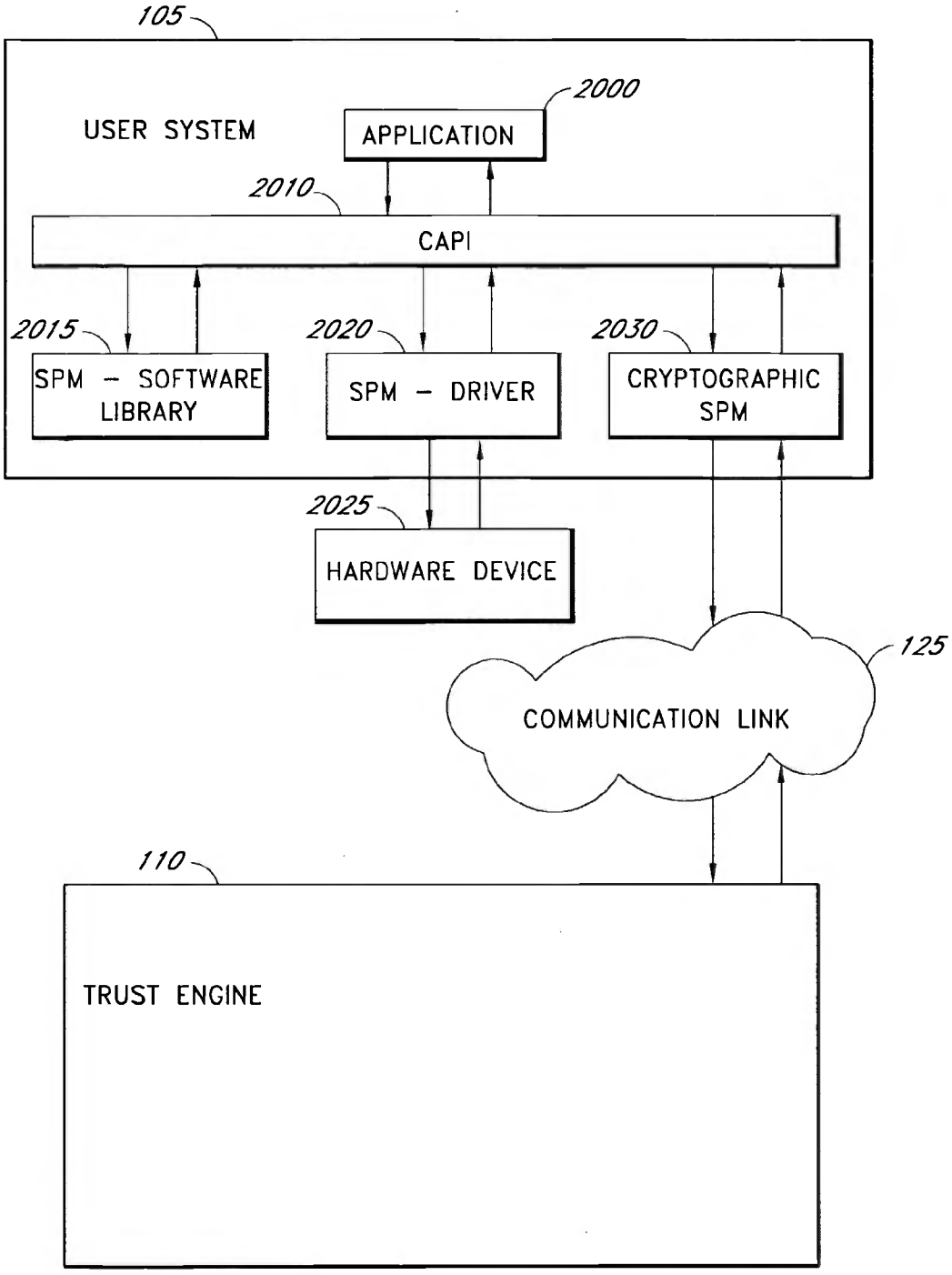

105
USER SYSTEM
2000
APPLICATION
2010
CAPI
2015
SPM – SOFTWARE LIBRARY
2020
SPM – DRIVER
2030
CRYPTOGRAPHIC SPM
2025
HARDWARE DEVICE
125
COMMUNICATION LINK
110
TRUST ENGINE

FIG. 20

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 153/00 — A1

(11) International Publication Number: **WO 99/46720**

(43) International Publication Date: 16 September 1999 (16.09.99)

(21) International Application Number: PCT/US99/05368

(22) International Filing Date: 11 March 1999 (11.03.99)

(30) Priority Data:

60/077,635	11 March 1998 (11.03.98)	US
09/260,874	2 March 1999 (02.03.99)	US

(71) Applicant: CHA! TECHNOLOGIES SERVICES, INC. [US/US]; 2nd floor, 704 Broadway, New York, NY 10003 (US).

(72) Inventors: LEITERSDORF, Yoav, A.; 2nd floor, 704 Broadway, New York, NY 10003 (US). SIXTUS, Timothy; 2nd floor, 704 Broadway, New York, NY 10003 (US).

(74) Agents: KALOW, David, A. et al.; Kalow Springut & Bressler, 19th floor, 488 Madison Avenue, New York, NY 10022 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

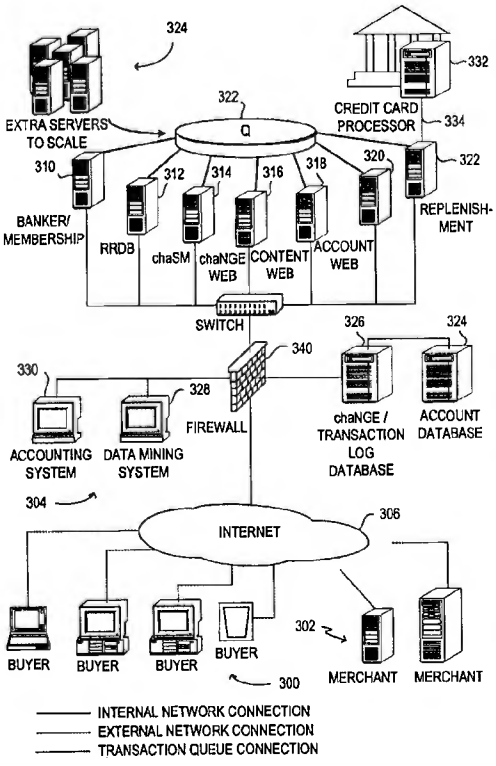
With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: AUTOMATICALLY INVOKED INTERMEDIATION PROCESS FOR NETWORK PURCHASES

(57) Abstract

An automatically invoked intermediation process for network purchases by subscribing purchasers (300) from subscribing merchants (302) includes the step of establishing a resource rules database (312) at the network site of each subscribing purchaser which, one aspect of the intermediation process, includes information encoding resource–locator–data identification criteria corresponding to restricted access ports at subscribing merchant sites (302) and access fee information for purchasing content by way of the corresponding port. Upon receipt of a user communication indicating approval, a purchase request message is transmitted to a transaction processing site (304). Upon verification of purchaser site credentials, the target resource is retrieved from the subscribing merchant site using access–restriction override information being encrypted and forwarded to the subscribing purchaser site.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

AUTOMATICALLY INVOKED INTERMEDIATION PROCESS FOR NETWORK PURCHASES

FIELD OF THE INVENTION

This invention relates to online electronic communications and transactions, and, in particular, to transactions between resource providers and users of such resources that communicate in an internetworked computer system.

BACKGROUND ART

Internetworked computer systems, such as the Internet, allow transactions such as electronic commerce purchases including electronic resource acquisition. In fact, the Internet has quickly become the preferred choice of many users for obtaining information, data, images, sound clips, software programs, etc. (referred to generally as a resource) since a user has virtually unlimited access to any host computer connected to the Internet. Problems have arisen, however, that are somewhat unique to this electronic commerce paradigm.

For instance, one problem has arisen in the area of authentication or confirmation of a user's identity. That is, when a user wishes to access a resource provided by a resource provider, the resource provider must be able to ensure that the user is actually who she says she is. In the real world, the provider (*e.g.*, a shopkeeper) can simply ask for a driver's license or other piece of identification to confirm the shopper's identity prior to accepting payment in the form of a check or credit card. In the online environment, other forms of identification are obviously required.

One unique method of accomplishing such authentication in the online environment is described in detail in co-pending United States patent application serial no. 08/816,410,

assigned to the assignee of the present invention, which is incorporated by reference herein. In that application, a unique triangulation methodology is presented, in which a trust server is utilized to confirm to a requesting online party the identity of another online party that was previously registered with the trust server. This is of course useful in the e-commerce application, wherein the purchaser can pre-register with the trust server and subsequently request an online vendor to execute a purchase transaction. The online vendor requests authentication of the purchaser from the trust server, and a three-way data exchange and confirmation process takes place. After confirmation, the online purchase transaction can be consummated.

Another problem that exists in the online world is the instance wherein a user would like to obtain a resource from a resource provider, such by downloading an image or other piece of information. Many resource providers require users to be registered with their services, for example by paying a monthly fee of $20.00 for unlimited downloading of data. The user may not want have to pay for such subscriptions just to get a single resource, especially since the cost of such registrations can become quite expensive given the number of resource providers that exist on the Internet.

In spite of a number of drawbacks, the Internet digital communications network is widely used for commercial transactions in which a buyer accesses a web site of a merchant using a browser and purchases data, textual material, graphics, or other digital content for immediate delivery from the merchant's site to the purchaser's site over the network. Often the content purchased is relatively inexpensive, say under a dollar in price. Nonetheless, heretofore large programs at both the buyer's site and the merchant's site have been required to handle the financial aspects of the transaction. Payments for purchases made over the network have been conventionally made with a digital currency known as "e-cash" using a

program at the buyer's site known as "wallet" software that is cumbersome to use and makes heavy use of encryption technology, which tends to lengthen the time required to complete the transaction. Transfer of any amount of e-cash, however small, from a buyer to a merchant has had to be cleared by a third-party bank or other financial institution. From the purchaser's perspective, the requirement to use wallet software and e-cash to purchase content over the network makes such purchases a "hassle," particularly in the case of inexpensive purchases, which has tended to discourage potential purchasers from making such purchases. On the merchant's side, difficulties often have been encountered in coordinating the clearance of the buyer's e-cash payment by the merchant's payment processing software and the transmission of the content purchased to the purchaser by web publication software at the merchant's site.

United States patent No. 5,767,917 to Rose *et al.* ("the Rose *et al.* '917 patent") disclosed a computerized payment system for purchasing goods and services on a quasi-public network such as the Internet. As disclosed at column 3, line 51 through column 4, line 14 of the '917 patent, the payment system included an "above-the-line system" and a "below-the-line system" which were separated by a "firewall" which isolated the two systems from one another. The above-the-line system included computer hardware and software directly connected to the Internet and represented the non-secure side of the firewall. The below-the-line system included computer hardware and software and was connected to the above-the-line system via a private network. The private network permitted limited communication between the two systems, but prevented unauthorized access to the below-the-line system through the above-the-line system. The below-the-line system represented the secure side of the sidewall.

According to column 4, lines 46 through 52 of the Rose *et al.* patent, the above-the-line system ran an above-the-line program which provided communication with users on the Internet who were buyers and users on the Internet who were sellers. As disclosed at column 4, line 66 through column 5, line 9 of the '917 patent, a buyer could access the above-the-line program interactively over the Internet by using a user-interface software program that could

be run on the user's computer, or could access the payment system via a conventional e-mail program. A seller could access the above-the-line program over the Internet interactively by running an interface software program on the seller's computer or could access the payment system via a conventional e-mail program.

As disclosed at column 5, lines 17 through 25 of the Rose *et al.* '917 patent, in order for an Internet user to use the payment system of the patent as either a buyer or a seller, the user would have to establish an account with the payment system. According to column 5, lines 35 through 64 of the patent, a buyer's account - termed a "cardholder" account - would include (1) a cardnumber which would identify the account, (2) the cardholder's name, (3) the cardholder's Internet e-mail address, (4) the state of the account; *i.e.*, "active," "suspended," or "invalid;" and (5) a pay-in selection indicating how the cardholder would make payment; *e.g.*,, authorization to charge a credit card. It was stated at column 5, lines 63 and 64 of the '917 patent that the pay-in selection was neither encoded in nor directly derivable from the cardnumber of the account. At column 6, line 51 through column 7, line 12 of the patent, it was disclosed that only a portion of the buyer cardholder account information resided in the above-the-line system of the payment system of the patent. Specifically, the account number, the cardholder's name, the Internet e-mail address of the cardholder, and the state of the account were stored in the above the line computer. The above-the-line computer did not contain pay-in information such as credit card information. A full copy of all the buyers' cardholder account information resided in the below-the-line system.

According to column 6, lines 41 through 50 of the Rose *et al.* '917 patent, a seller's account with the payment system of the patent would include the following data: (1) a seller's account cardnumber, (2) the seller's name, (3) the seller's Internet e-mail address, (4) a state of the account, and (5) a seller's agent number. As disclosed at column 6, lines 15 through 37 of the patent, a seller's agent would be a bank card processor which would interact with credit card bureaus to perform the functions of credit card authorizations and chargebacks. The seller's agent number would be stored in the below-the-line system of the payment system of the patent, and not in the above-the-line system. See column 7, lines 17 through 20 of the Rose *et al.*'917 patent.

As disclosed at column 8, lines 1 through 26 of the Rose *et al.* '917 patent, when a buyer decided to buy goods or services from a seller, the buyer would inform the seller of the buyer's cardnumber by means of an appropriate message over the Internet. Upon receiving the buyer's message, the seller would send a payment-request message to the payment system of the patent over the Internet, either by e-mail or by an interactive protocol on the Internet. The payment-request message from the seller would include the buyer's cardnumber, the seller's cardnumber, a textual description of the transaction, an amount, a merchant's transaction identifier, and any delivery information. According to column 6, lines 27 through 38 of the '917 patent, after receiving the payment-request message, the above-the-line program would ascertain whether the message was from a qualified seller by checking a database file on the above-the-line system. Upon confirmation that the payment-request message was from a qualified seller, the above-the-line program of the payment system of the patent would generate a payment-query message to be sent to the buyer over the Internet to ask the buyer whether the buyer authorized payment for the transaction to the seller. According to column 8, lines 40 through 62 of the Rose *et al.* '917 patent, the payment-query message contained a transaction identifier, the buyer's name, the seller's name, the textual description of the transaction, and an amount. The payment-query message would then be sent to the buyer's e-mail address.

As disclosed at column 8, lines 62 through 64 of the Rose *et al.* '917 patent, the payment-query message would request the buyer to respond with one of three replies: "yes," "no," or "fraud." According to column 10, lines 31 through 66 of the patent, if the buyer responded with a message indicating "yes," the above-the-line program would transfer the transaction information to the below-the-line program. When the transaction information was received, the below-the-line system would associate the buyer's cardnumber with the buyer's payment information and the seller's account number with the seller's agent number. The below-the-line system would then communicate to the seller's agent the identity of the seller, the transaction amount, the buyer's credit card number or other payment information, and any delivery information. The communication to the seller's agent would be carried out on secure communication channels off the Internet. If the seller's agent approved the charge, it would send an authorization code to the below-the-line system. According to column 10, lines 64

through 66 and column 11, lines 8 through 25 of the Rose *et al.* '917 patent, upon receipt of the authorization code, the below-the-line program would generate a cryptographic signature for the authorization code which would then be transferred to the above-the-line program. The above-the-line program would then transmit to the seller a payment-notification message which would include the seller's transaction identifier and the cryptographically signed authorization code. Upon receipt of the payment-notification message, the seller could cryptographically authenticate the authorization code and, upon verification of the authenticity of the message, proceed to deliver the goods or services to the buyer. According to column 11, lines 26 through 32 of the Rose *et al.* '917 patent, further processing of the charges to the buyer's credit card account and credits to the seller's merchant account would be conducted by a conventional settlement system off the Internet, reportedly isolating the buyer-seller activity which occurred on the Internet from the financial and credit activity which occurred off the Internet.

United States patent No. 5,822, 737 to Ogram ("the Ogram '737 patent") concerned an automated payment system for purchases over a distributed computer network such as the Internet. According to column 4, line 57 through column 5, line 4 of the '737 patent, a consumer using a customer computer would connect with a merchant computer over the network. The merchant computer would communicate promotional material to the customer computer via the network. If the customer decided to buy the service or product from the merchant, the link with the merchant computer would be broken - by a mechanism apparently not specified in the '737 patent - and the customer computer would be linked with a payment processing computer. In the change from the merchant computer to the payment processing computer, an indicia of the URL or the product being promoted by the merchant computer would be communicated to the payment processing computer. The payment processing computer could accept the credit card account number which was to be debited the amount of the product. Column 5, lines 22 through 24 of the '737 patent. According to column 2, lines 7 through 10 of the patent, an encrypting software package could be first downloaded to the customer's computer for secure transmission of the credit card number. As disclosed at column 5, lines 25 through 33 of the Ogram '737 patent, the payment processing computer would then establish a link via telephone lines with a credit card server computer while

maintaining linkage with the customer computer. The credit card account number and amount would be communicated to the credit card server computer, which would respond with an authorization indicia.

According to column 2, lines 16 through 37 of the Ogram '737 patent, after receipt of the authorization indicia, the payment processing computer would in certain cases transmit a password to the customer's computer. The password would have been defined by the merchant's computer to pass along to the customer's computer. The password could be used by the customer's computer to gain access to restricted material within the merchant's computer. As disclosed at column 6, lines 25 through 34 of the Ogram '737 patent, the password could be displayed on a window in a screen on the customer's computer for the customer to use with the merchant. The password could be printed or committed to memory, and then entered by the consumer to lift restricted access to material in the merchant's computer.

SUMMARY OF THE INVENTION

The subject invention broadly concerns an automatically invoked intermediation process for purchasing content over a digital communications network by subscribing purchasers from subscribing merchants.

The intermediation process comprises the step of establishing a database of subscriber-purchaser accounts and a database of subscriber-merchant accounts at a central transaction processing site on the digital communications network. The subscriber-merchant accounts database includes information encoding resource locator data identifying at least one restricted-access port at each subscribing merchant site on the network and, for each such restricted-access port, information encoding an access fee schedule for accessing content by way of the restricted-access port and access-restriction-override information such as a password for enabling access to content by way of the restricted-access port. The subscriber-purchaser accounts database includes information encoding purchaser site authentication credentials and a purchaser account balance for each subscribing purchaser.

The intermediation process of the invention further comprises establishing a resource rules database at the network site of each subscribing purchaser. The resource rules database includes information encoding resource-locator-data identification criteria corresponding to each of at least a subset of the restricted-access ports at subscribing merchant sites identified in the subscriber-merchant accounts database and the access fee schedule for accessing content by way of the corresponding restricted-access port.

The intermediation process of the invention also includes the steps of locally monitoring network communication activity information at the network site of each subscribing purchaser with respect to access to a target network resource generated by a browser program at the subscribing purchaser site and comparing such network communication activity information to the resource-locator-data identification criteria corresponding to restricted-access ports at the network sites of subscribing merchants in the resource rules database locally maintained at the site of the subscribing purchaser.

In the event network communication activity information pertaining to the target network resource matches with the resource-locator-data identification criteria corresponding to a restricted-access port in the resource rules database, the intermediation process of the invention also includes the steps of retrieving the access fee schedule corresponding to the matched resource-locator-data identification criteria from the resource rules database, displaying an access fee from the access fee schedule for accessing content by way of the restricted access port corresponding to the resource-locator-data identification criteria, and prompting for approval or disapproval of completing the access for the displayed access fee.

The intermediation process of the invention further comprises, upon receipt at the subscribing purchaser site of a user communication responsive to the prompting indicating approval of completing the access to content by way of the restricted-access port for the access fee, the step of transmitting a purchase-request message from the subscribing purchaser site to the transaction processing site over the network. The purchase-request message includes information encoding purchaser-site authentication credentials and identifying the

restricted-access port at the merchant site corresponding to the matched resource-locator-data identification criteria.

The intermediation process of the invention also comprises, upon receipt of the purchase-request message at the transaction processing site, the step of determining whether the purchaser-site authentication credentials encoded in the purchase-request message matches with any purchaser-site authentication credentials included in the subscriber-purchaser accounts database. If such a match is found, the process of the invention includes the step of transmitting over the network to the restricted-access port of the merchant site a restricted-access-enabling access-request message to download purchased content by way of the restricted-access port to the transaction processing site. The restricted-access-enabling access-request message includes access-restriction-override information corresponding to the restricted-access port retrieved from the subscriber-merchant accounts database.

The intermediation process of the invention further comprises, upon receipt of the downloaded purchased content at the transaction processing site from the subscribing-merchant site by way of the restricted-access port responsive to the restricted-access-enabling access-request message, the steps of encrypting the downloaded purchased content and forwarding the thus encrypted purchased content to the subscribing purchaser site over the network.

The intermediation process of the invention also includes, upon receipt of the encrypted purchased content at the subscribing purchaser site, the step of transmitting a content-received confirmation message to the transaction-processing site by the subscribing purchaser site over the network.

The intermediation process of the invention further includes, upon receipt of the content-received confirmation message from the subscribing purchaser site at the transaction processing site, the step of debiting the purchaser account balance corresponding to the subscribing purchaser in the subscriber-purchaser accounts database by the access fee for accessing purchased content by way of the restricted-access port and forwarding a

cryptographic key for decrypting the encrypted purchased content to the subscribing purchaser site over the network.

Finally, the intermediation process of the invention includes, upon receipt of the cryptographic key at the subscribing purchaser site, the step of decrypting the encrypted purchased content at the purchaser site.

Preferably, the access fee schedule corresponding to a restricted-access port in the subscriber-merchant accounts database consists of a single access fee for accessing content by way of the restricted-access port.

The access-restriction-override information corresponding to a restricted-access port in the subscriber-merchant accounts database preferably includes a password for enabling access to content by way of the restricted-access port.

The subscriber-purchaser accounts database preferably includes, for each subscribing purchaser, information encoding an account-replenishment amount and a financial account number identifying an account at a financial institution upon which the subscribing purchaser may draw. The step of debiting the purchaser account balance corresponding to the subscribing purchaser in the subscriber-purchaser accounts database by the access fee for accessing purchased content by way of the restricted-access port preferably includes the step of comparing the purchaser account balance with an account minimum value. If the purchaser account balance has fallen below the account minimum value, the process of the invention preferably includes the step of sending a funds-transfer request to the financial institution for transferring funds in the amount of the account-replenishment amount from the account identified by the financial account number to an intermediation entity account. Upon receipt at the transaction processing site of an approval of the transfer request, the purchaser account balance is preferably credited by the account replenishment amount to bring the purchaser account balance above the account minimum value.

The funds transfer request is preferably sent from the transaction processing site to the financial institution over a communication channel separate from the digital communications network interconnecting the transaction processing site, the subscribing purchaser sites, and the subscribing merchant sites.

The digital communications network interconnecting the transaction processing site, the subscribing purchaser sites, and the subscribing merchant sites is preferably the Internet.

Content purchased by means of a preferred intermediation process of the invention can include practically any resource which can be transmitted over the Internet. Such content could include text, software, data, visual images, and audio, for example.

Certain types of digital content such as hypertext mark-up language ("HTML") documents frequently contain embedded software objects such as images encoded in a standard format such as the graphic image format ("GIF") or the joint photographic experts group ("JPEG") format. A software object embedded in digital content is conventionally identified by resource locator data such as a URL which encodes the location of the object on the network. In a preferred embodiment of the intermediation process of the invention, digital content downloaded to the transaction processing site from a restricted-access port of a subscribing-merchant site is checked to determine if the content is of a type likely to contain embedded software objects. If the content is found to be an HTML document or other type of content likely to contain embedded software objects, the content is parsed at the transaction processing site and any resource locator data corresponding to a software object embedded in the content extracted. Resource locator data so extracted is then used to retrieve any corresponding software objects over the network. The software objects thus retrieved are combined with the original downloaded content at the transaction processing site to form a composite purchased-content object. The composite purchased-content object is then encrypted and forwarded to the subscribing purchaser site which purchased the content over the network. Decryption of the encrypted composite purchased-content object at the subscribing purchaser site according to such preferred process makes available the original content and any embedded software objects at the purchaser site without the necessity of the

browser running on the purchaser site having to retrieve the embedded software objects over the network.

Optionally, if suitable arrangements are made with the subscribing merchant, a copy of a digital content downloaded to the transaction processing site from a restricted-access port at the network site of the subscribing merchant pursuant to a purchase-request message received from a subscribing purchaser site in accordance with an embodiment of the intermediation process of the invention may be cached at the transaction processing site. If a purchaser-request message is received subsequently from a second subscribing purchaser site which identifies the same digital content as that cached in the transaction processing site, the cached copy of the digital content may be used to provide the purchased content for encryption and downloading to the second subscribing purchaser site without having to access the content a second time by way of the restricted-access port of the subscribing merchant port.

In addition to digital content transmitted over the Internet, "hard goods" may be purchased by subscribing purchasers from subscribing merchants over the Internet for delivery offline by means of a preferred intermediation process of the invention described in the following paragraphs.

The preferred automatically invoked intermediation process for purchasing hard goods over a digital communications network such as the Internet includes the steps of establishing a database of subscriber-purchaser accounts and a database of subscriber-merchant accounts at a central transaction processing site on the network. The subscriber-merchant accounts database includes information encoding resource locator data identifying at least one order-entry port at each subscribing merchant site on the network and, for each such order-entry port, information encoding a schedule of prices and goods purchasable by orders entered by way of the order-entry port. The subscriber-purchaser accounts database includes information for each subscribing purchaser encoding purchaser site authentication credentials and a purchaser account balance.

The intermediation process for purchasing hard goods also includes the step of establishing a resource rules database locally at the network site of each subscribing purchaser. The resource rules database includes information encoding resource-locator-data identification criteria corresponding to each of at least a subset of the order-entry ports at subscribing merchant sites identified in the subscriber-merchant accounts database and the schedule of prices and goods purchasable by orders entered by way of the corresponding order-entry port.

The intermediation process for purchasing hard goods further includes the steps of locally monitoring network communication activity information at the network site of each subscribing purchaser with respect to access to a target network resource generated by a browser program at the subscribing purchaser site and comparing such network communication activity information to the resource-locator-data identification criteria corresponding to order-entry ports at the network sites of subscribing merchants in the resource rules database locally maintained at the site of the subscribing purchaser.

In the event network communication activity information pertaining to the target network resource matches with the resource-locator-data identification criteria corresponding to an order-entry port in the resource rules database, the intermediation process for purchasing hard goods also includes the steps of retrieving the schedule of prices and goods corresponding to the matched resource-locator-data identification criteria from the resource rules database, displaying at least a portion of prices and goods from the schedule, and prompting for identification of goods to be ordered.

The intermediation process of the invention for purchasing hard goods further comprises, upon receipt at the subscribing purchaser site of a user communication responsive to the prompting identifying hard goods to be ordered, the step of transmitting an order-request message from the subscribing purchaser site to the transaction processing site over the network. The order-request message includes information encoding purchaser-site authentication credentials, identifying the order-entry port at the merchant site corresponding to the matched resource-locator-data identification criteria, and specifying the hard goods to be ordered.

The intermediation process of the invention for purchasing hard goods also comprises, upon receipt of the order-request message at the transaction processing site, the step of determining whether the purchaser-site authentication credentials encoded in the order-request message matches with any purchaser-site authentication credentials included in the subscriber-purchaser accounts database. If such a match is found, the process includes the step of transmitting over the network to the order-entry port of the merchant site an order-entry message to enter an order for the hard goods specified in the order-request message by way of the order-entry port and request transmission of an order-confirmation message from the merchant site to the transaction processing site.

The intermediation process of the invention for purchasing hard goods further includes, upon receipt of an order-confirmation message at the transaction processing site from the subscribing merchant site responsive to the order-entry message, the steps of encrypting the order-confirmation message and forwarding the encrypted order confirmation message to the subscribing purchaser site over the digital communications network.

The intermediation process of the invention for purchasing hard goods also includes, upon receipt of the encrypted order-confirmation message at the subscribing purchaser site, the steps of transmitting an order-confirmation received message to the transaction-processing site by the subscribing purchaser site over the network.

The intermediation process of the invention for purchasing hard goods further includes, upon receipt of the order-confirmation message from the subscribing purchaser site at the transaction processing site, the step of debiting the purchaser account balance corresponding to the subscribing purchaser in the subscriber-purchaser accounts database by the total price of the hard goods ordered and forwarding a cryptographic key for decrypting the encrypted order-confirmation message to the subscribing purchaser site over the network.

The intermediation process of the invention for purchasing hard goods also includes, upon receipt of the cryptographic key at the subscribing purchaser site, the step of decrypting the encrypted order-confirmation message at the purchaser site. Delivery of the hard goods to the subscribing purchaser is made offline by the subscribing merchant. The decrypted order-confirmation message may be used as proof to the subscribing merchant that the goods were ordered and paid for.

Advantageously, a subscribing merchant to an intermediation service carrying out a preferred intermediation process of the invention can use conventional Web publishing software at its Internet site to establish publicly accessible ports and restricted-access ports for use in connection with the intermediation process. Commercially available conventional Web publishing software ordinarily has facilities for establishing a password-protected restricted-access port at an Internet site which requires the presentation of a user identification code and a password to obtain access to content by way of the restricted access port. No software in addition to such conventional Web publishing software is required of a subscribing merchant to make use of preferred intermediation processes of the invention, which represents a significant advantage over the prior art.

In one aspect of the present invention, the problem of the user needing subscriptions in order to obtain online resources is addressed through the use of a resource proxy.

The resource proxy acts as a broker, and generally will have its own subscriptions to many resource providers. When a user requests a resource from a provider, a client process on the user's machine intercedes and requests the resource proxy to obtain the resource. In one preferred embodiment of the invention, the client process intercedes when the provider asks for the user's identification and password in response to a request for a resource by the user. In an alternative preferred embodiment of the invention, the client process intercedes when the user generates a request to access a resource from a provider to which access is restricted. The resource proxy, who has an account with the resource provider (or who will get the account if required), then obtains the resource on behalf of the user and provides the resource to the user. Thus, the user can obtain many individual resources from the proxy

without having to subscribe to each service. The resource proxy can serve multiple users, of course, in order that the provision of such individual resources can make economic sense.

In another aspect of the present invention, a secure micropayment system is implemented by the resource proxy and the users to effect payment for the use of the resources.

In another aspect of the present invention, some aspects of the above-described triangulation methodology are used to implement unique and novel applications in the online environment. That is, the resource proxy may use a trust server to ensure that the requesting user is actually who she says she is, and to collect payment (the micropayment system) for the provision of the individual resource.

BRIEF DESCRIPTION OF THE DRAWINGS

Preferred embodiments of the intermediation process for network purchasers of the invention will be described below with reference to the following figures:

Figure 1 is an illustration of a triangulation authentication methodology utilized in a first preferred embodiment of the present invention.

Figure 2 is an illustration of a prior-art resource request made by a user to a resource provider.

Figure 3 is an illustration of the use of a resource proxy in the triangulation authentication schema of Figure 1.

Figures 4-7 and 9 illustrate the interaction between existing financial networks, resource providers, and the triangulation authentication flow.

Figure 8 illustrates resource acquisition in accordance with a preferred embodiment of the present invention.

Figure 10 is an illustration of one aspect of the use of a resource proxy in accordance with a preferred embodiment of the present invention.

Figure 11 illustrates data flow in a back office transaction.

Figures 12A-12C is a sample merchant application.

Figure 13 shows aggregation of micropayment sums.

Figure 14 is a schematic diagram of subscribing purchaser sites, subscribing merchant sites, and an intermediation service transaction processing site interconnected via the Internet for carrying out a preferred embodiment of the intermediation process of the invention.

Figure 15 is an illustration of an intermediation-service subscription application screen generated by a preferred intermediation service.

Figure 16 is an illustration of a subscription-confirmation screen generated by the preferred intermediation service.

Figure 17 illustrates a newspaper contents screen for a hypothetical online financial newspaper.

Figure 18 illustrates a newspaper subscription screen presented by the online financial newspaper when a Web browser attempts to navigate to restricted-access content of the newspaper without making use of a preferred intermediation service of the invention.

Figure 19 illustrates an intermediation-service pop-up window which appears on the screen of a subscribing purchaser's personal computer when a browser running on the computer attempts to access restricted-access content of the online financial newspaper in the case in which the newspaper is a subscribing merchant to the preferred intermediation service.

Figure 20 is an illustration of a screen representing content of the online financial newspaper to which access is restricted.

Figure 21 is an illustration of an account-maintenance screen presented to subscribing purchasers to the preferred intermediation service.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The following name table will facilitate understanding of micropayment system for one preferred intermediation process of the invention as presented herein.

Alice: Registered user running micropayment system client software.

Bob: Registered Server and Resource Proxy.

Ted: Trust Server / Broker.

Reg: Registration Server.

Hal: Resource Vendor, not necessarily associated with the system.

S_A: A unique "secret" Alice shares with Ted.

S_B: A unique "secret" Bob shares with Ted.

R_A: Random number generated by Alice.

R_B: Random number generated by Bob.

NGE Nonce Generation Event in which a session key (or nonce) is created to encrypt/decrypt requested resources.

K Random session key.

K_A,K_B Function of one seed A to seed B, and it's inverse.

F_{A1}-F_{AN} Results 1 through N of a function calculated by A.

F_{B1}-F_{BN} Results 1 through N of a function calculated by B.

Referring to Figure 1, a triangulation authentication data flow over a computer network such as the Internet that will accomplish a successful authentication via a process referred to as a session manager is as follows:

1. User network site Alice initializes and sends an authentication request to user network site Bob.
2. User Bob accepts the authentication request, forms a first compound message comprising the authentication request and information about himself, and forwards the compound message to network site Ted.
3. Ted receives the first compound message (from Alice to Ted through Bob), confirms Alice's intent and sends her a second compound message to Bob (from Ted to Bob through Alice).
4. Alice forwards Ted's message to Bob.

Encryption may optionally be utilized at sensitive message transfers to ensure that no data is viewed or tampered with by any party other than the intended recipient. This includes the compound messages Bob or Alice conveys. At step four, for example, the message Alice sends to Bob from Ted is preferably Ted's confirmation of Alice's status encrypted such that only Bob can decipher the message.

Based on this process, Bob can be a server with resources that Alice seeks or requires. Once the authentication is complete, Bob can serve Alice's requests with impunity.

Under standard HTTP protocol, a resource or resources that are made available via Web subscription are in most cases protected with a simple User ID/Password ("UID/PW") scheme. HTTP defines this type of protection in a straightforward manner. Referring now to Figure 2:

1. Alice requests a World Wide Web resource by entering a Uniform Resource Locator ("URL") into her browser.
2. The HTTP server ("Hal") fetches the resource for her request and returns it to her browser. However, if the resource is protected, Hal sends back a request for a UID/PW.
3. In the latter case, the browser responds to Hal's request by displaying a dialog box with input boxes for a UID/PW pair.
4. Alice's browser makes a second request for the resource, this time providing the user-entered UID/PW data with the request.
5. Hal then fetches the resource for Alice's request and returns it to her browser.

After the initial UID/PW exchange, the entered data is stored during the rest her session to satisfy any of Hal's future requests for them.

Utilizing the triangulation authentication that the session manager facilitates, and the prior-art authentication model of HTTP's UID/PW scheme detailed above, a method for conducting micro-currency transactions is detailed, referred to herein as the micropayment system. Simply stated, in a triangulated system where Bob is a subscriber to an online service or has a previous relationship with a resource vendor, any Alice can obtain password-protected data through Bob's subscription or account without a subscription of her own.

Thus, in the process shown in Figure 3 (again, all data communication is over the Internet):

1. Alice requests "any resource" from Hal.
2. The resource is password protected, so Hal asks Alice for a password (PW).

3. The password request triggers the micropayment system control (the micropayment system control "captures" the password request and initializes the authentication process. Simultaneously, the URL is requested of Bob instead of Hal).
4. Alice asks Bob for "any resource", and sends triangulation authentication data.
5. Bob asks Ted if Alice is who she claims to be.
6. Ted checks, and sends Alice a compound message.
7. Alice forwards Bob's part of the message to Bob.
8. Bob is satisfied that Alice is really Alice, so he uses his account to request "any resource" from Hal (Bob provides his own UID/PW data).
9. Hal delivers the resource to Bob.
10. Bob sends the resource on to Alice.

Finally, since Bob is a trusted member of Ted's network through prior registration, there exists a running dialog between the two. Ted can keep track of Alice's requests and debit her account per delivery.

In Figure 4, all major components are shown and the overall data flow is outlined.

The platform of the present invention provides low cost, easy to use e-Commerce controls. The components are based on the foundation of Session Manager authentication, a process by which any registered user can authenticate himself or herself to any other in the system.

The micropayment system facilitates online purchases of primarily digital resources that may range in value from less than a tenth of a cent to approximately US$15.00. The upper bound may be determined by the abilities of Secure Electronic Transactions ("SET") initiatives.

Referring to Figure 4, prior to accessing any of the services that the system offers, Alice must register with Ted in the following manner:

1. Alice obtains the micropayment system client software via web link, download, CDROM, special offer or some other such means.
2. Alice runs an install program, and she is prompted to fill in her registration information.
3. The client software sends a Registration server (Reg) the appropriate information along with an approved fiscal instrument to charge (*e.g.*, a credit card account number).
4. Reg passes the registration data across a secure gateway to back-office services and generates a unique registration number for Alice.
5. Alice's fiscal instrument is charged a previously negotiated amount.
6. Upon a successful transaction, Alice's funds are credited to an account in the Bank and an "ok" status is returned to Reg from across the secure gateway.
7. The negotiated amount is digitally translated into NGE (Nonce Generation Events, that allow for decryption of resources to be provided in the future, and once used, represent a purchase) which are stored on a Trust Server (Ted) and associated with Alice's registration number.
8. Reg provides Alice with the registration number via a secure channel.
9. Alice activates her micropayment system client software and becomes a registered user.

Reg is not heard from again by this particular Alice, except in rare, well-defined cases discussed below.

Now Alice, as a registered user of the system, is micropayment system enabled and can make micro-currency purchases on the Internet. When Alice comes across a resource that she wants while browsing the Web, she simply has to click on it, or otherwise attempt to retrieve it as is her custom. If the resource vender (Hal) chooses to charge for accessing this resource, he will devise a mechanism to obtain proper credentials from anyone wanting the resource, as known in the art. In one preferred embodiment of the invention, Hal's request for credentials signals the micropayment system client to intercede on Alice's behalf. Alice is asked if she

would like Micropayment System to handle the transaction for her. If she agrees, then the resource request (URL) and authentication information is passed to a registered resource proxy. There exist user-configurable parameters such that Alice does not get bombarded with session manager conversation.

The session manager process is then initialized. Figure 5 shows the session manager authentication triangle (Alice to Bob to Ted to Alice, back to Bob).

The data flow to accomplish a successful authentication via a preferred session manager process is as follows:

1. Alice initializes and sends an authentication request to Bob.
2. Bob accepts this request, forms a first compound message comprising Alice's authentication request and information about himself, and forwards that first compound message to the trust server (Ted).
3. Ted receives the first compound message (from Alice to Ted through Bob), confirms Alice's intent and sends a second compound message (from Ted to Bob through Alice).
4. Alice forwards Ted's message to Bob.

Encryption may be utilized at sensitive message transfers to ensure that no data is viewed or tampered with by any party other than the intended recipient. In particular, the compound message Ted sends in step three is preferably encrypted. A compound message involves all three participants, and during each compound message transfer, the party in the middle adds or reads information unique to itself.

At step four, the message Alice sends to Bob from Ted is Ted's encrypted confirmation of Alice's status that only Bob can decipher. When Bob opens his message from Ted, he can be certain that Alice is who she claims to be.

Extending on this process, Bob can be a server with resources that Alice seeks or requires. Once the authentication is complete, Bob can serve Alice's requests with impunity. This particular Bob is a registered resource proxy. The micropayment system client software on Alice's machine has sent a resource request to Bob, and initiated an authentication session.

In the micropayment system universe and on the network are Ted, Reg and other micropayment system supporting servers (such as directory servers, business rules servers, etc. In the framework of the present invention, Bob is also found in this secured zone - and as such possesses extra abilities and features.

Bob and Ted are preferably locally connected over a fast connection to reduce throughput between them. Additional potential overhead that might be found in negotiating a large number of session manager authentications is reduced for the same reason. Local connection between session manager components increase scalability options as well as reducing the ultimate "cost-per-click."

Resource proxy Bob maintains accounts with multiple resource vendors acting as a universal client.

Figure 6 shows existing resource providers. Bob is a client to the many resource-vendor Hal variants that exist in the online world, including e-Commerce schemes ranging from a simple subscription model to Open Market's CSP (Commerce Service Provider). Preferably Bob is equipped to communicate via diverse front-ends supported or enforced by any particular Hal. Bob is a registered user with respect to each of them. Bob and Hal may have an existing relationship. If Bob should encounter a Hal that has an unknown or new e--Commerce scheme, Bob may opt to subscribe. Thus, Alice's requested resource may be fetched by Bob from a previously unknown Hal.

Referring to Figure 7, Bob has obtained the resource that Alice wants. Bob has effectively encapsulated any vendor requirements for billing and charges, and is now in possession of Alice's requested resource. Bob functions as a "universal client" to every Hal

and as a "universal server" to every Alice. At this point in the data flow, Bob is ready to service Alice's request for a resource.

During the session manager authentication phase, Alice was recorded by Ted as making the request. Ted now creates a nonce generation event which is used in the debit process.

Stepping back to step three of the session manager authentication, Ted sent Alice a compound message where she read the part destined for her and forwarded the rest, which was encrypted for Bob. The second compound message contained a nonce, or one time key (preferably, a pair of keys) generated by Ted, and such keys are what is sent to Bob and Alice.

Since Bob is directly connected to Ted, the second part of the compound message does not need to be forwarded through Alice. The direct connection between Bob and Ted has the two-fold benefit of reducing overall CPU and bandwidth utilization for all three parties, while increasing security - less information, whether it is encrypted or not, is required to be exposed to the public wire.

The data flow at this stage follows the pattern of Figure 8. Bob has a resource for Alice. When Ted is satisfied that Alice is who she claims to be pursuant to the session manager function, Ted sends Alice a decryption key (1) and its mate to Bob (2). Bob encrypts the resource with his encryption key, sends the encrypted resource to Alice (3) where she can decrypt it with her decryption key. Since a nonce was generated, no one but the intended recipient can obtain the resource in this session, and Alice's acceptance and usage of key provides proof of delivery. The nonce generation event has served its purpose and Ted will decrement the available nonce generation events reserved in Alice's account.

Referring now to Figure 9, the back-office processes required to facilitate transaction processing via the present platform are detailed.

There are two types of transactions within this framework: real-time transactions based on nonce generation events, discussed above, and back-end transaction processing, where user credit instruments are charged and resource vendors are remunerated.

The backoffice and accounting services reside apart from the core net-side nonce generation event based transactional activity and across a protected gateway. The backoffice and accounting services layer helps to form the accounting foundation integral to, and found within, any corporate entity.

The backoffice and accounting services layer is where client and vendor accounts are processed and where the purchase orders and invoices get entered and fulfilled. The back office and accounting services layer is where statements and checks are printed and posted. The business model resembling the back office in activity and intent from this vantage point, and the view of the National Automated Clearing House Association ("NACHA"), is often described as mail-order/telephone order ("MOTO"). In a typical MOTO system, customer orders are placed via telephone (for the most part) and accepted without the customer's signature, and then charged with central terminal servers.

It is important to note, however, that the back end of the preferred embodiment bears a distinct kinship to a service provider's revenue model as well. The user base is asked to pay up front for services which the micropayment system promises to provide at a later time. For example, the nonce generation events may be sold in quantities corresponding to twenty United States dollars-(US$20). The subscription agreement between the user and the company providing the micropayment system provides that when the user's supply of nonce generation events gets low enough, the Company is authorized to replenish the user's supply of nonce generation events without the user's needing to intervene.

The replenishing debit accounts function is carried out on one or more terminal servers, running clearing software and linked to an external clearing body by a modem and telephone line (or a dedicated) communications link. The external clearing body may be a NACHA representative, credit card company, bank or a national financial non-bank, such as

ATT Capital, GE Capital Services, etc. The back-end transaction processing may be, in general, carried out in different ways with different parties. Preferably the processing system can select parties and processing methods which are advantageous for a given transaction.

Alice's fiscal instrument is charged, and upon a successful transaction, she is provided a usable online micropayment system account. Under the subscription agreement, the amount charged will not exceed US$20.00 and will not be less than a lower limit based on costs enforced by the external clearing body.

In order to perform micro-currency transactions without incurring the prepayment costs and the associated risks that come from brokering a transaction, the system utilizes debit accounts. A debit account stores a previously negotiated amount charged against a users credit card or other fiscal instrument. This is sometimes called "pay-to-play" and is functionally similar to the familiar automated teller machine (ATM) paradigm.

As mentioned above, the back-end transaction processing can be carried out in a number of ways. Preferably, the various different options for carrying out back-end transaction processing are assembled in a modular layer. One clearing system may prove useful at any single time, but whenever another proves to be more advantageous for any reason (*e.g.*, better rates, partnership arrangements, etc.), the more advantageous system can be adopted without disrupting the rest of the structure.

In a preferred procedure for replenishing debit accounts, Ted will determine when Alice has reached a previously agreed upon lower threshold. At that time, Ted will send a message through the secure gateway to a local terminal server. The terminal software will negotiate via modem or leased line connection an authorized charge against Alice's fiscal instrument. Upon a successful clearing transaction, an "ok" status is sent back through the secure gateway to Ted. The process for replenishing debit accounts is similar to the process for initially charging the debit accounts.

Referring to Figure 10, a basic overview of a micropayment transaction will now be discussed.

1. A user/buyer wants to obtain a resource (article, image, sound, etc.) from an Internet merchant/vendor.
2. The merchant/vendor requires proof of subscription (either to the merchant's services, or to the merchant's choice of e-Commerce vendor) before the resources will be delivered.
3. The user/buyer is not a subscriber and chooses not to subscribe at this time, but would still like to obtain the resource.
4. The micropayment client software residing on the user/buyer's personal computer senses the subscription request and offers to obtain the resource for the user/buyer.
5. If the user/buyer chooses to have the micropayment client handle the transaction, a message is sent to the micropayment servers indicating this.
6. The micropayment server uses it's own subscription to the merchant/vendor to obtain the resource
7. The merchant/vendor recognizes the micropayment server as a subscriber and delivers the resource to the micropayment server.
8. The micropayment server then delivers the resource to the user/buyer.

The following is a description of the back office procedures utilized by a preferred embodiment of the present invention. As with any company that accepts credit cards for payments, the pieces required to process transactions include a merchant account, a terminal server (such as a card reader), a modem link, and access to a clearing company.

Several layers to the credit card payment processing hierarchy are beyond the scope of a simple back-office process. These include:

1. Issuing bank (of the credit card);
2. Interchange access (Visa, MasterCard, AMEX differences, etc.); and

3. Processor (National Automated Clearing House Association representative company).

The complexities of these structures are all masked with traditional terminal server software and hardware issued by an Independent Sales Office (ISO) for a group of national processors.

A number of web vendors offer credit-card payment processing services for conducting a transaction on the Internet. Such companies include WebMall, USWeb, Octagon, and CyberCash, etc. Services such companies provide range from the set-up of a merchant account on an organization's behalf to performing the entire online transaction from the company.

A preferred micropayment process of the invention does not require online Internet transactions. The micropayment back-office activities are akin to a service organization such AOL, MSN, UUNet, etc. in that the buyer is asked periodically to pre-pay for services that will be rendered. No Internet connection is required for the transaction to transpire, the user need not be online at the time, and the transaction does not occur on the Web. This type of transaction process is regarded by NACHA as a mail order/telephone order or MOTO transaction because the credit card itself is never physically present when a transaction is processed.

An Internet service such as the preferred micropayment system does not require an Internet transaction. In fact, it is part of the inherent security and ease of use of the present invention that the transactional activity occurs offline. The CPU and bandwidth requirements of securing an online transaction are completely removed.

Figure 11 illustrates micropayment system backoffice activity.

1. A micropayment terminal server or other point of sale (POS) device receives a charge request for User X.

2. The terminal server looks up User X in an accounts database and gets the information, "Alice User, CC# 1234 4567 8910, expiration 0999."
3. The terminal server then dials into it's credit card processor and requests a US$20.00 charge against Alice User's credit card, passing the information ("Alice User, CC# 1234 4567 8910, expiration 0999") it just received from the accounts database.
4. The credit card processor returns a status (either "OK" or "Unable").
5. The terminal server passes only the status of User X back to the secured gateway and the micropayment online servers beyond.

If the status were "OK," then the credit card processor debit the cardholders account by US$20.00 and credits the micropayment merchant account the same US$20.00. This can be performed in a batch process where many requests for credit card clearing are queued until an appropriate time, or performed at the time of request.

Figures 12A-C comprise an exemplary form for creating a merchant account. Though the costs and charges vary, they are roughly as follows:

Application fee:	US$135.00
Clearing charges:	US$0.30
Clearing rate:	1.75%.
Card not present:	.75%

Note that there is an approximate .75%. premium for MOTO (Card not present) transactions.

Micropayment transactions are conducted at the request of multiple buyers. Nonce generating events are issued to users for purchasing resources from multiple merchant/vendors.

Periodically, a "transaction dump" or raw data from the Internet is processed, where a measure of used nonce generating events is sent through the secured gateway destined for the

accounting services. Accounting services will receive this raw data and calculate an aggregate yield based on the accumulated usage of a merchant/vendor's resources. No individual buyer's information is kept in the online space or found in the raw data -- such information is not required to remunerate merchant/vendors.

As a hypothetical example, suppose 100 users have obtained a news story from vendor "X" over a set period, at an individual cost of US$0.50 per article. Over this same period, suppose 150 users have played "Doom" for US$0.25 per play at the website of vendor "Y." In this example, there were 250 unique transactions. Each merchant/vendor is spared the effort of negotiating each transaction, and tracking each user for the low monetary sum each resource is valued. The data for this period would define US$50.00 associated with vendor "X" and US$37.50 associated with vendor "Y."

Referring to Figure 13, the preferred micropayment process of the invention aggregates small individual purchases into one larger accumulated value. This value is further processed through accounting services to yield a commission for performing these services. The remaining cash value is processed and a check is issued to each respective merchant/vendor at pre-defined intervals (monthly, bi-weekly, etc.) in one lump sum (as opposed to 250 separate checks of less than a dollar apiece). This is true for every vendor.

Benefits for user/buyers in this model come from the fact that none of their purchasing dollars are tied to a single merchant/vendor. Their ability to spend nonce generating events freely allows for more "impulse spending" as well, which also benefits merchant/vendors.

A particularly preferred embodiment of the intermediation process of the invention for the purchase of digital content from a merchant by a purchaser over the Internet is described below.

In the preferred embodiment, both the merchant and the purchaser are subscribers to an automatically-invoked intermediation service which provides facilities for carrying out the process of the invention. In particular, the intermediation service provides a trust server and a

proxy content server. The trust server and the proxy content server can separately access the Internet and can be separately accessed via the Internet. In addition, the trust server and the proxy content server of the intermediation service are interconnected by a private digital communications network separate from the Internet.

A preferred arrangement for inter-connecting subscribing purchasers, subscribing merchants, and components of a transaction processing site for the preferred intermediation service is illustrated schematically in Figure 14. The preferred intermediation service employs a modular and distributed arrangement involving a transaction queue for transaction processing by servers at the transaction processing site, which provides for scalability and work-load balancing by adding additional servers.

Turning now to Figure 14, a plurality of personal computers 300 of buyers who subscribe to the intermediation service, a plurality of web servers 302 of merchants who subscribe to the intermediation service, and a transaction processing site 304 are interconnected by the Internet digital communications network 306. The transaction processing site 304 includes a plurality of servers 310-330 among which various tasks of transaction processing for the intermediation service are distributed. A core group of the servers 310-322 at the transaction processing site 304 are interconnected by a private transaction-queue interconnection network 332 which facilitates distribution of the transaction processing tasks among the servers of the core group 310-322 and permits additional servers 324 be added to the core group for scaling and work-load balancing. As an example of the distribution of the transaction-processing tasks among the servers of the core group, a banker/membership server 310 may manipulate a subscribing purchaser accounts database; a resource rules database server 312 may manipulate a master copy of a resource-rules database for the service, a trust server 314 may perform purchaser account verification; a supervisor server 316 may serve decryption keys for decrypting purchased content and order-confirmation messages; a proxy content server 318 may retrieve purchased content from subscribing merchant sites and, if appropriate, parses the content to identify and retrieve software objects embedded in the purchased context; an account server 320 handles the initial formation of new accounts and account maintenance; and an account replenishment server 322

EX1010, Page 1084

handles replenishment of subscribing purchaser accounts. The account replenishment server 322 can communicate with a credit-card processor 332 of a financial institution via a channel 334 separate from the Internet. A master database server 326 logs transactions processed by the core group and stores the subscribing purchaser accounts database and the master copy of the resource-rules database and, in conjunction with an account database server 324, maintains financial account data such as credit card numbers of subscribing purchasers. An accounting system server 330 may handle disbursements to subscribing merchants and a data mining system 328 may collect statistical data concerning use of the transaction processing site. Communication between the transaction processing site 304 and the Internet is mediated by a firewall system 340 which controls access to the servers of the site.

The preferred intermediation service of the invention maintains a subscriber-purchaser accounts database and a subscriber-merchant accounts database, both of which can be accessed by the trust server and the proxy content server not via the Internet. The subscriber-purchaser accounts database includes for each subscribing purchaser, a purchaser account identification number, cryptographic information for communicating with the subscribing purchaser, a purchaser account balance, and information for replenishing the purchaser's account. Table I below sets forth the information maintained for each subscribing purchaser in the subscriber-purchaser accounts database.

Table I

Subscriber Purchaser Accounts Database

Key	Name	Type	Description
*	UserID	NUM(10)	ID of the purchaser
	PreviousSecret	NUM(10)	Previous shared secret
	Secret	NUM(10)	Secret User shares with the trust server
	NextSecret	NUM(10)	Next secret for encryption
	ReplenishAmount	NUM(10)	Amount for automatic replenishment
	ReplenishInProgress	NUM(10)	If TRUE, additional request for replenish won't occur
	AccountStatus	NUM(10)	Used to determine if account is active, pending delete or withdrawal, etc.)
	ReplenishType	NUM(10)	Describes whether manual/automatic replenishment is set.
	HighReplenish	NUM(10)	The value of balance for sending replenish for this purchaser in $0.0001
	Balance	NUM(10)	Balance of the purchaser account in units of $0.0001

The subscriber-merchant accounts database includes information encoding universal resource locator ("URL") data identifying one or more restricted-access ports at each subscribing merchant site on the network, and, for each such restricted-access port, information encoding an access fee for accessing content by way of the restricted-access port and a password for obtaining access to content by way of the restricted-access port.

A subscribing purchaser will generally communicate over the Internet using a personal computer running an Internet browser program and making use of the services of a commercial Internet service provider ("ISP"). Subscription to the preferred intermediation service of the invention is generally initiated "online" over the Internet, with the new subscriber accessing a Web site of the intermediation service. Referring to Figure 15, the preferred intermediation service downloads a subscription application screen 400 in hypertext markup language ("HTML") format for display on the display screen of the personal computer of the new subscriber. The subscription application screen 400 includes HTML data-entry fields labeled to prompt for entry of the new subscribing purchaser's name 402 and billing address 404, including e-mail address.

The preferred intermediation service maintains a purchaser account for each subscribing purchaser against which charges for purchases made by the purchaser using the intermediation service are applied. The purchaser account of a new purchaser is opened with an initial deposit which is withdrawn from another financial account of the new subscribing purchaser, such as a credit-card account maintained with a bank or other financial institution. The subscription application screen 400 includes HTML data-entry fields labeled to prompt for entry of billing information 406 to be registered with the intermediation service, including card type, card number, and expiration date. The subscription application screen 400 also includes HTML data-entry fields labeled to prompt for entry of purchaser account replenishment information 408, including the amount of an initial deposit in the purchaser account of the new subscribing purchaser. When the balance in a purchaser account maintained by the preferred intermediation service falls below a "replenishment threshold" value, the purchaser account is replenished by making a withdrawal of a replenishment amount

from the credit-card or other financial account of the subscribing purchaser registered with the intermediation service and crediting the purchaser account by the replenishment amount withdrawn. The replenishment information 408 prompted for in the subscription application screen 400 includes a replenishment threshold value, a replenishment amount, and a selection of whether the replenishment is to be carried out automatically or manually by the subscribing purchaser.

Upon transmission of the subscription information prompted for by the subscription application form 400 to a transaction processing site of the preferred intermediation service and verification of the billing information by withdrawal of the initial deposit from the financial account of the new subscribing purchaser, a subscription confirmation screen 410 illustrated in Figure 16 is downloaded in HTML format over the Internet from the transaction processing site to the personal computer of the new subscribing purchaser for display on the personal computer under the control of the browser program. The subscription confirmation screen 410 indicates that a subscribing purchaser account has been successfully established and identifies the purchaser account number. The HTML instructions which specify the subscription confirmation screen 410 painted by the browser on the display screen of the personal computer of the new subscribing purchaser define a labeled control area 412 which carries the label: "click here to download redirection and authentication control program." In accordance with the HTML instructions specifying the subscription confirmation screen 410, the browser can detect when a user executes a "mouse click" when the cursor of the display screen is positioned within the labeled control area. In response to detecting such a mouse click , the browser program transmits a control action message over the Internet to the transaction processing site of the intermediation service which indicates that a mouse click occurred in the labeled control area 412. Upon receipt of the control action message from the subscribing purchaser's computer at the transaction processing site of the intermediation service, a redirection and authentication control program is downloaded over the Internet from the transaction processing site and installed in the personal computer of the new subscribing purchaser.

The redirection and authentication control program runs on the personal computer of each subscribing purchaser of the preferred automatically invoked intermediation service and locally monitors Internet communication activity information generated by the Internet browser program running on the personal computer. The redirection and authentication control program can monitor Internet communication activity generated by the browser program using the application programming interface "API" of the browser program. For example, the redirection and authentication control program can monitor communication activity information generated by a "Netscape Navigator" browser program by way of the "DDE API" interface of the Navigator program and can monitor communication activity information generated by a "Microsoft Internet Explorer" browser program by way of the "COM" interface of the Internet Explorer program.

In order to accomplish URL interception from the Netscape Navigator browser, for example, the following DDE interface calls may be used:

- DdeConnect, DdeDisconnect -- Establishes/closes a DDE connection to the browser. One DDE connection is all that is necessary to monitor all Navigator browser windows.
- WWW_RegisterProtocol, WWW_UnRegisterProtocol (DdeConversation) -- allows the redirection and authentication control program to register the HTTP, FTP, and any other Internet protocols. This allows the control program to take control every time a URL is requested of the browser.
- WWW_ViewDocFile, WWW_WindowChange, DdeFreeStringHandle, DdeClientTransaction, DdeCreateStringHandle, DdeNameService, DdeQueryString, DdeUnaccessData, DdeFreeDataHandle -- maintenance of DDE functions, that altogether allow for the URL interception to take place.

OpenURL, which is not a DDE operation may be used to process a URL request. This call is used for two purposes: (1) to return control to the browser for URL requests for which there are no resource rules database entries; and (2) to instruct the browser to display a cached resource retrieved by the control program via the intermediation service.

The redirection and authentication control program installed in the personal computer of each subscribing purchaser to the preferred intermediation service includes a resource rules database for identifying Internet communication activity information generated by the browser program running on the personal computer with respect to accessing resources available by way of restricted-access ports of network sites of subscribing merchants to the intermediation service. For each of at least a subset of the restricted access ports at subscribing merchant sites identified in the subscriber-merchant accounts database, the resource rules database of the redirection and authentication control program installed in a given subscribing purchaser's personal computer includes a data structure referred to as a "rule" which functions as a scripting language to permit search criteria specified in the rule to be applied to URL's or other communication activity information generated by the Internet browser program running on the computer. For example, a rule could be used to test a URL for the presence of a particular keyword or a hostname to identify a restricted-access port at a subscribing merchant's site. In addition, rules include information for determining how much is to be charged for access to the resource in the restricted access port and for how long access to the resource is to be permitted once permission to access has been purchased.

A master resource rules database is maintained at the transaction processing site of the preferred intermediation service for updating the local resource rules databases in the personal computers of the subscribing purchasers from time to time. Subscribing purchasers may maintain copies of the full master resource rules database or may maintain a copy of the resource rules database for only a subset of the restricted-access ports identified in the master resource rules database. Such a subset might, for example, be limited to sites suitable for access by children, or, for purposes of limiting the size of the research rules database in the personal computer of subscribing purchasers, to sites with content in a particular language.

The master copy of the resource rules database is structured as a structured query language ("SQL") database. An entry in the master copy of the resource rules database takes the form of an SQL table row. Product, fee schedule, and term information used to form the entry in the database is obtained from the subscribing merchant who is to provide the product. For distribution to subscribing purchaser sites, a "flat-file" version of the master copy of the

SUBSTITUTE SHEET (RULE 26)

EX1010, Page 1089

resource rules database is prepared at the transaction processing site, which is then compressed using a conventional data compression algorithm. The compressed flat-file version of the resource rules database is distributed to new subscribing purchaser sites along with the redirection and authentication control program. To maintain the local versions of the resource rules database current at subscribing purchaser sites, the redirection and authentication control program queries the transaction processing site upon initialization of each Internet connection and every twenty four hours for the date of the current version of the master copy of the resource rules database. If the master copy of the resource rules database at the transaction processing site is more recent than the local copy maintained at the subscribing purchaser site, the redirection and authentication control program downloads a copy of the most recent resource rules database in the compressed file format, decompresses the downloaded database file, and updates the copy of the database maintained at the purchaser site. Updated purchase terms are generally distributed throughout the intermediation service within forty-eight hours.

The following Table II gives the structure of the resource rules database.

Table II

Resource Rules Database

Host Table

Key	Name	Type	Description
*	HostName	STR(256)	Hostname of the URL
	HostID	NUM(10)	ID associated with this hostname

RuleLookup Table

Key	Name	Type	Description
*	HostID	NUM(10)	ID associated with hostname
	RuleID	NUM(10)	ID associated with rule belonging to the HostID

Rule Table

Key	Name	Type	Description
*	RuleID	NUM(10)	ID associated with a rule
	Rule	CHAR(512)	Rule itself. A locator string in a logical expression.

RuleDetail Table

Key	Name	Type	Description
*	RuleID	NUM(10)	ID associated with a rule
	Type	NUM(1)	Type of receipt that will be generated for this rule (URL || Rule ID)
	Duration	NUM(10)	Duration of content purchased under this rule in minutes
	RetailPrice	NUM(10)	Price of content purchased under this rule in cents
	CostOfGoods	NUM(10)	Amount due vender for this item
	ExpirationDate	STR(12)	Date when this rule becomes invalid
	Description	STR(255)	Expression that describes the rule to the user
	VendorID	NUM(10)	Identification of the vendor

The redirection and authentication control program running on the personal computer of a subscribing purchaser functions to intercept automatically requests to access a target resource at a restricted-access port of an Internet site of one of the subscribing merchants, to notify the subscribing purchaser that access to such resource may be had under certain conditions including payment of an access fee, and, if the subscribing purchaser chooses to accept the conditions, to redirect the request for access to the transaction processing site of the intermediation service for charging the subscribing purchaser's account the amount of the access fee and fulfilling the request for access to the target resource. The functioning of the redirection and authentication control program of the preferred intermediation service from the perspective of a subscribing purchaser is described below in terms of a hypothetical example involving accessing content at the Internet site of an online financial newspaper.

When a subscribing purchaser desires to browse the Internet, he or she launches a browser program on his or her personal computer and establishes communication with an Internet service provider. The redirection and authentication control program of the preferred intermediation service is launched automatically upon the launching of the browser program. Each time the subscribing purchaser navigates to a new Internet site, the browser program generates a URL which identifies the site on the Internet. Transparently to the subscribing purchaser, the redirection and authentication control program monitors each URL generated by the browser program before the URL is transmitted by the browser program from the personal computer and determines if the URL corresponds to a restricted-access port of a subscribing merchant identified in the resource rules database of the control program.

Consider the cases of one person who has an account with the intermediation service for carrying out the preferred embodiment of the subject invention and another person who does not have such an account, both of whom desire to access an article in the online financial newspaper over the Internet. Both the subscriber and the nonsubscriber to the intermediation service would typically first navigate to a publicly accessible port of the Internet site of the online newspaper to view a newspaper contents screen 414 which would describe the contents of the newspaper and contain links 416 to certain articles and financial data published in newspaper, as illustrated in Figure 17. The articles and financial data, however, would not be accessible to the general public, but would be available only to subscribers of the online newspaper. Thus, if the person who was not a subscriber to the intermediation service attempted to navigate to one of the articles identified in the publicly accessible contents screen 414 of Figure 17, the person would receive a newspaper subscription screen 416 calling for entry of a newspaper subscription user name and password as a condition to access the article, as shown in Figure 18. A subscribing purchaser to the preferred intermediation service, upon attempting to navigate to the same article identified in the publicly accessible newspaper contents screen 414 of Figure 17, would be shown an intermediation-service pop-up window 422 offering the subscribing purchaser the option of accessing the nonpublicly accessible contents of the online newspaper for an access fee and an access time specified in the pop-up window, as illustrated in Figure 19. The intermediation-service pop-up window would generally appear practically instantaneously after the subscribing purchaser attempted to access the non-publicly accessible article, since the redirection and authentication control program of the intermediation service running locally on the subscribing purchase's personal computer would have intercepted the URL generated by the browser program to access the article and would have determined by searching the resource rules database maintained locally on the personal computer that the URL corresponded to a restricted-access port of subscribing merchant to the intermediation service - the online financial newspaper. As a result, the redirection and authentication control program would block transmission of the URL and, using access fee and access time data retrieved locally from the resource rules database, generate the intermediation service pop-up window 422.

EX1010, Page 1092

As shown in Figure 19, the intermediation-service pop-up window 422 generated by the redirection and authentication control program includes two "push-button" control areas 426 and 428 respectively labeled "OK" and "cancel" to prompt for proceeding with the access under the access fee and access time conditions 424 specified in the pop-up window or termination the access process under the intermediation service. If the "cancel" push-button control area 428 is selected, the browser is permitted to transmit the blocked URL and the newspaper subscription screen 418 appears as shown in Figure 18. If the "OK" push-button control area 426 is selected, the request for access to the restricted-access port of the online newspaper is redirected by the redirection and authentication control program over the Internet to the transaction processing site of the preferred intermediation service, as discussed below.

If the subscribing purchaser elects to proceed with the requested access under the conditions specified in the intermediation-service pop-up window 422 by executing a mouse click with the cursor positioned on the push-button control area 426 labeled "OK" on the pop-up window 422, a six-step authentication and fulfillment procedure in accordance with the preferred intermediation process is initiated to authenticate and fulfill the requested access.

In the first step of the authentication and fulfillment procedure of the preferred intermediation process, the redirection and authentication program running on the subscribing purchaser's personal computer forms a URL request data object which is composed of three fields for encoding respectively: (1) the rule identification number - designated "Rule ID" in Table II above - of the rule in the resource rules database which matched the URL generated by the browser program in connection with attempting to access a target resource by way of the restricted-access port of the Internet site of a subscribing merchant and led to the presentation of the intermediation-service pop-up window, (2) the type of the rule, which specifies the type of receipt which will be generated for the rule, and (3) the URL of the target resource accessible by way of the restricted-access port in question.

For authentication purposes, the local redirection and authentication control program maintains a ten digit number denoted "S_A" referred to as the "account secret" which

can be used as an encryption and decryption key in an encryption/decryption procedure. The value of the account secret is updated after each transaction to access a target resource by way of a restricted-access port at an Internet site of a subscribing merchant. As may be seen in Table I, values for a previous account secret, the current account secret, and a next account secret are maintained for the account of each subscribing purchaser in the subscriber-purchaser accounts database maintained at the transaction processing site of the preferred intermediation service.

Also for authentication purposes, a transaction random number is generated by the redirection and authentication control program for each transaction to access a target resource by way of a restricted-access port at an Internet site of a subscribing merchant.

After generation of the transaction random number, the redirection and authentication control program forms a first-step-identifier, random-number augmented data object composed of a first-step identifier code number designated α_1 and the transaction random number. The first-step-identifier, random-number augmented data object is then encrypted by the redirection and authentication control program using a conventional data encryption procedure with the current account secret number as the encryption key. The encryption procedure used permits the encrypted data object to be decrypted with the same key value.

A redirected purchase-request message is then transmitted over the Internet from the personal computer of the subscribing purchaser by the redirection and authentication control program to the proxy content server of the preferred intermediation service. The redirected purchase-request message is composed of the purchaser account number, a version identification number for the intermediation procedure, the first step identifier code number α_1, the URL request data object, and the encrypted first-step-identifier, random-number-augmented data object. Transmission of the redirected purchase-request message to the proxy content server concludes the first step of the authentication and fulfillment procedure of the preferred intermediation process.

In the second step of the authentication and fulfillment procedure of the preferred intermediation process, the proxy content server receives the redirected purchase-request message from the personal computer of the subscribing purchaser and checks the first-step-identifier code number α_1 and the version identification number included in the purchase-request message and generates an appropriate error message and terminates the authentication and fulfillment procedure if either is not recognized. If both the first-step-identifier code number α_1, and the version identification code in the redirected purchase request message are recognized by the proxy content server, the proxy content server retransmits the redirected purchase request message without change to the trust server of the preferred intermediation service over the private network interconnecting the proxy content server and the trust server, thereby concluding the second step of the authentication and fulfillment procedure.

In the third step of the authentication and fulfillment procedure carried out by the preferred intermediation service, the trust server receives the redirected purchase-request message and extracts the rule identification number and the URL for the target resource from the URL request data object included in the purchase-request message. The trust server checks the validity of the redirected purchase request by determining whether the rule identification number and the URL of the target resource correspond to one another in the master resource rules database maintained at the transaction processing site of the preferred intermediation service. If the rule identification number and the URL of the target resource correspond in the master resource rules database, the authentication and fulfillment procedure is permitted to proceed, otherwise the procedure is terminated and an appropriate error message is generated. The first-step-identifier code α_1 is again extracted from the redirected purchase-request message and checked. If the code is not recognized, the authentication and fulfillment procedure is terminated and an appropriate error message issued.

If the authentication and fulfillment procedure is permitted to proceed, a resource encryption key is generated at the trust server by the following procedure. First, two random numbers are generated and then combined to form a double-random-number data object. The double random-number data object is then processed according to a known cryptographic message digest algorithm known as the "SHA1" secure hash algorithm. The SHA1 secure

hash algorithm is described in the following publications: "Secure Hash Standard," National Institute of Standards and Technology, TIPS Publication 180, May 1993 and "SHA-1, Announcement of Weakness in the Secure Hash Standard," National Institute of Standards and Technology, May 1994. The result of applying the SHA1 secure hash algorithm to the double-random-number data object is taken as the resource encryption key.

Next, the purchaser account number and the encrypted first-step-identifier, random-number-augmented data object are extracted from the redirected purchase-request message at the trust server. Using the purchaser account number, the current account secret S_A and the previous account secret S_{Ap} are retrieved from the subscriber-purchaser accounts database maintained at the transaction processing site of the preferred intermediation service. If the purchaser account number is unrecognized, the authentication and fulfillment procedure is terminated and an appropriate error message generated. A decryption algorithm using the account secret as decryption key is then applied to the encrypted first-step-identifier, random-number-augmented data object from the redirected purchase-order message and the result checked to determine if the decryption was successful by determining if the first step identifier code number α_1 was extracted. If the first decryption attempt was not successful, a second decryption attempt is made by the applying the decryption algorithm to the encrypted first-step-identifier, random-number augmented data object using the previous account secret S_{Ap} as decryption key, in case the redirected purchase-request message transmitted by the subscribing purchaser's computer had been delayed. The result of the second decryption attempt is checked as before to determine if the decryption was successful by determining if the first step identifier code number α_1 was extracted. If the second decryption attempt was not successful, the authentication and fulfillment procedure is terminated by the trust server and an appropriate error message issued, in that the identity of the subscribing purchaser whose purchaser account number was included in the redirected purchase-request message was not confirmed.

If either of the two decryption attempts was successful, the authentication and fulfillment procedure is permitted to proceed by the trust server. In the case in which the second decryption attempt using the previous account secret S_{Ap} as decryption key was the

SUBSTITUTE SHEET (RULE 26)

EX1010, Page 1096

successful one, a working value of the current account secret for the account in the trust server is set equal to the value of the previous account secret S_{Ap}. Also in the case in which the second decryption attempt was successful, a working value of the transaction random number in the trust server is set equal to the value of the random number extracted from the decrypted encrypted first-step-identifier, random-number augmented data object, which, in that case, represented the transaction random number of the previous transaction.

Subsequently in the third step of the authentication and fulfillment procedure, a verified-account-identity purchase-request message is transmitted from the trust server to the proxy content server over the private network interconnecting the trust server and the proxy account server. The verified-account-identity purchase-request message includes data encoding the version identification number for the intermediation procedure taken from the redirected purchase-request message previously forwarded to the trust server from the proxy content server, a third step identifier code number α_3, the resource encryption key generated at the trust server, and the URL for the target resource taken from the redirected purchase-request message.

After transmission of the verified-account-identity purchase-request message by the trust server, a composite session key is composed as a sum of the resource encryption key included in the verified-account-identity purchase request message and a locally-computable key augmentation value. The locally-computable key augmentation value is computed by applying the SHA1 secure hash algorithm three times in succession. First, the SHA1 secure hash algorithm is applied to a first composite data object composed of the working values of the transaction random number originally generated by the redirection and authentication control program running on the subscribing purchaser's personal computer and the current account secret S_A in the trust server to obtain a first hash value. The first hash value is combined with the working values of the transaction random number and the current account secret to form a second composite data object. The SHA1 secure hash function is then applied to the second composite data object to obtain a second hash value. The second hash value is then combined with the working values of the transaction random number and the current account secret to form a third composite data object. The SHA1 secure hash function

is then applied to the third composite data object to yield a third hash value which serves as the locally-computable key augmentation value. The locally-computable key augmentation value can also be computed locally by the redirection and authentication control program in the personal computer of the subscribing purchaser, since the values of the transaction random number and the current account secret are available locally to the redirection and authentication control program.

In the fourth step of the authentication and fulfillment procedure, the URL of the target resource is extracted from the verified-account-identity purchase-request message at the proxy content server. The URL of the target resource is used as a pointer to the resource rules data base to extract a vendor identification number. The vendor identification number in turn is used to extract from the subscriber-merchant accounts database a password for accessing the target resource by way of the restricted-access port of the subscribing merchant. Using the URL for the target resource and the password for obtaining access to the target resource by way of the restricted-access port of the Internet site of the subscribing merchant, the target resource is accessed and transmitted over the Internet to the proxy content server at the transaction processing site of the preferred intermediation service.

After the proxy content server receives the target resource, the target resource is combined with a fourth-step-identifier code number α_4 to form a step-labeled target-resource composite data object. The step-labelled target-resource composite data object so formed is then encrypted with an encryption algorithm which employs the resource encryption key extracted from the verified-account-identity purchase-request-message as the encryption key for the algorithm.

To conclude the fourth step of the authentication and fulfillment procedure, an encrypted-target-resource delivery message is transmitted over the Internet from the proxy content server to the personal computer of the subscribing purchaser who originally attempted to access the target resource. The encrypted-target-resource delivery message includes data encoding the version identification number for the intermediation procedure, the fourth step identifier code number α_4, and the encrypted step-labelled target-resource data object.

For the fifth step of the authentication and fulfillment procedure, the locally-computable key augmentation value previously computed at the trust server in the third step of the procedure is recomputed independently by the redirection and authentication control program running on the personal computer of the subscribing purchaser. The final key augmentation value and the intermediate hash values obtained in computing the final key augmentation value should be identical whether computed at the trust server or at the subscribing purchaser's personal computer, since the same procedure involving applying the SHA1 secure hash algorithm three times in succession to successive composite data objects formed from the transaction random number, the current account secret, and prior hash values can be followed in both cases and the values for the transaction random number and the current account secret are the same.

As discussed below, a receipt data object is generated in the sixth step of the authentication and fulfillment procedure for each transaction in which access to a target resource is obtained by means of the preferred intermediation service and transmitted to the personal computer of the subscribing purchaser. The elements of the receipt data object are specified in Table III below:

Table III

Receipt Data Object

Key	Name	Type	Description
*	RuleID	NUM(10)	ID associated with a rule
	URL	STR(1024)	Include this with receipt if the URL is required by the rule type.
	LocalExpirationDate	DATE	Used locally to determine if this content is paid for.
	ServerExpirationDate	DATE	Proves to the server that this content is paid for.
	KeyID	NUM(10)	A pointer to an array of keys maintained in the trust server; Indicates which key was used in the hash.
	ReceiptHash	STR(1024)	Indicates the resultant of the hash.
	User ID	NUM(10)	Purchaser account number

SUBSTITUTE SHEET (RULE 26)

EX1010, Page 1099

The receipt data object may be presented to the transaction processing site of the intermediation service to enable the subscribing purchaser to obtain multiple accesses to the target resource - say, over a fixed period of time established by the rule - without incurring multiple charges. With respect to the present discussion of the fifth step of the authentication and fulfillment procedure, it should be borne in mind that the redirection and authentication control program running on the subscribing purchaser's personal computer may or may not have a receipt data object at the time of a given instance of the fifth step of the authentication and fulfillment procedure, depending on whether the transaction had previously progressed to a stage at which a receipt data object was generated and transmitted to the subscribing purchaser's personal computer.

Whether or not the redirection and control program has a receipt data object in a particular instance of a fifth step of the authentication and fulfillment procedure for a given transaction, the redirection and authentication control program forms a step-labeled receipt indicator composite data object. In the case the redirection and control program has previously received a receipt data object for the transaction, the step-labeled receipt-indicator composite data object is formed by combining the fifth-step identifier code number α_5 with the following data elements from the receipt data object: RuleID, ServerExpirationDate, UserID, KeyID, ReceiptHash, and, if appropriate to the rule type, the URL of the target resource. In the case the redirection and control program has not previously received a receipt data object for the transaction, the step-labeled receipt-indicator data object is formed by combining the fifth-step indicator code number with code indicating that no receipt data object is present. In either case, the step-labeled receipt indicator composite data object is then encrypted with an encryption algorithm using as the encryption key the second hash value obtained as the result of the second of the three successive applications of the SHA1 secure hash algorithm made in connection with the calculation of the locally-computable key augmentation value.

A resource-decryption-key request message is assembled by the redirection and authentication control program running on the personal computer of the subscribing purchaser after receiving the encrypted target-resource delivery message from the proxy content server of the preferred intermediation service. The resource-decryption-key request message is

composed of data encoding the purchaser account number, the version identification number of the intermediation procedure, the fifth-step identifier code number α_5, and the encrypted step-labeled receipt-indicator composite data object. The resource-decryption-key request message is transmitted from the personal computer of the subscribing purchaser to the trust server of the preferred intermediation service over the Internet to conclude the fifth step of the authentication and fulfillment procedure of the preferred intermediation process.

In the sixth and final step of the authentication and fulfillment procedure of the preferred intermediation process of the invention, a determination is made at the trust server of the intermediation service carrying out the preferred intermediation process in response to the resource-decryption-key request message from the subscribing purchaser's personal computer whether or not to transmit a resource decryption key to the subscribing purchaser, and, if so, whether to charge the subscribing purchaser's account for the access fee in connection with transmitting the resource decryption key. As discussed in more detail below, the determination of whether to charge the subscribing purchaser's account for the access fee involves three test criteria applied with respect to the step-labeled receipt-indicator date object included in the resource-decryption-key request message: (1) does the data object include a receipt data object?; (2) if a receipt data object was included, is it genuine?; and (3) if a receipt data object was included, has the expiration date passed?

First, the encrypted step-labeled receipt-indicator composite data object is decrypted at the trust server using as decryption key the second hash value obtained as the result of the second of the three successive applications of the SHA1 secure hash algorithm made in connection with the calculation of the locally-computable key augmentation value. Extraction of the fifth-step identifier code number α_5 from the decrypted step-identified receipt indicator composite data object verifies the identify of the subscribing purchaser specified by the purchaser account number included in the resource-decryption-key request message.

If the identify of the subscribing purchaser specified by the purchaser account number included in the resource-decryption-key request message is verified, the decrypted step-labeled

receipt-indicator composite data object is checked to determine if a receipt data object was included in the data object.

If no receipt data object was found to have been included in the step-labeled receipt-indicator composite data object, the subscribing purchaser's account with the intermediation service is charged the amount of the access fee for accessing the target resource.

If a receipt data object was found to have been included in the step-labeled receipt-data composite data object, the genuineness of the receipt data object is checked before proceeding. Specifically, the SHA1 secure hash algorithm is applied to a composite data object formed of the following data elements included in the receipt data object: KeyID, RuleID, ServerExpirationDate, UserID, and, if called for by the rule type, the URL of the target resource. If the result of the SHA1 hash function matches the ReceiptHash included in the receipt data object, indicating that the receipt data object is genuine, the authentication and fulfillment procedure proceeds to the date verification step without charging the subscribing purchaser's account. If the result of the SHA1 has function does not match the ReceiptHash in the receipt data object, the receipt data object is not genuine and the subscribing purchaser's account is charged the amount of the access fee. If a receipt data object was found to have been included in the step-labeled receipt-indicator composite data object and the receipt data object was found to be genuine, the server expiration date encoded in the receipt data object is compared to the current date provided by the trust server. If the server expiration date has not been exceeded, no charge is made to the subscribing purchaser's account. If the server expiration date has been exceeded, the subscribing purchaser's account is charged the access fee.

If the identity of the subscribing purchaser is verified and either no receipt data object was found to be included in the step-labeled receipt indicator composite data object, or a receipt data object was found to be included in the composite data object and the receipt data object was determined to be genuine, a resource-decryption-key delivery message is assembled at the trust server. The resource-decryption-key delivery message is composed of data-encoding the following data elements: the purchaser account number, the version

identification number of the intermediation procedure, a sixth-step identifier code number α_6, and an encrypted step-labeled key/receipt-carrier composite data object. The step-labeled key/receipt-carrier composite data object is formed of the sixth-step identifier code number α_6, the composite session key computed in the third step of the authentication and fulfillment procedure discussed above, and, a receipt data object incorporating the data elements identified in Table III above. The step-labeled key/receipt-carrier composite data object is then encrypted using an encryption algorithm employing as encryption key the second hash value obtained as the result of the second of the three successive applications of the SHA1 secure hash algorithm made in connection with the calculation of the locally-computable key augmentation value.

The resource decryption-key delivery message is transmitted from the trust server to the personal computer of the subscribing purchaser over the Internet. After the transmission of the resource-decryption-key delivery message, the record of information specific to the transmission may be cleared from the trust server.

After receipt of the resource-decryption-key delivery message at the personal computer of the subscribing purchaser, the redirection and authentication program running on the computer applies a decryption algorithm to the encrypted step-labeled key/receipt carrier composite data object using as decryption key the result of the second of the three successive applications of the SHA1 secure hash algorithm made in connection with the calculations of the locally-computable key augmentation value. As noted above, the calculation for the locally-computable key augmentation value can be carried out identically on the trust server and locally on the subscribing purchaser's personal computer. Decryption of the encrypted step-labeled key-receipt carrier composite data object enables the redirection and authentication control programs to obtain the value of the composite session key included in the data object. The resource decryption key may then be obtained locally from the composite session key by subtracting the locally-computable key augmentation value from the composite session key. The resource decryption key may then be used by the redirection and authentication control program to decrypt the encrypted target resource previously transmitted to the personal computer of the subscribing purchaser from the proxy content

server in the fourth step of the authentication and fulfillment procedure of the intermediation process. For example, the subscribing purchaser can peruse an article from the online financial newspaper 429 on the display screen of his or her personal computer, as illustrated in Figure 20.

In addition, decryption of the encrypted step-labeled key/receipt carrier composite data object enables the redirection and authentication control program to obtain a receipt data object included in the composite data object. Obtaining the receipt data object permits the subscribing purchaser to access additional content in the online newspaper without further charge until the server expiration date included in the receipt data object is exceeded.

Finally, the sixth step of the authentication and fulfillment procedure concludes by updating the account secret at the personal computer of the subscribing purchaser by setting the new account secret equal to the result of the first of the three successive applications of the SHA1 secure hash algorithm made in connection with the calculation of the locally-computable key augmentation value. Additionally, the account secret values in the subscriber-purchaser accounts database at the transaction processing site are updated by setting the new previous account secret value equal to the old current account secret value and setting the new current account secret value equal to the previously determined first hash value, which was the result of applying the SHA1 secure hash algorithm to the first composite data object composed of the working values of the transaction random number and the current account secret S_A.

The subscribing purchaser may monitor the status of his or her account with the preferred intermediation service by downloading an account maintenance screen 430 for the account from the transaction processing site, as illustrated in Figure 21.

It is not intended to limit the present invention to the specific embodiments described above. For example, it is recognized that these and other changes may be made in the intermediation process for network purchases of the invention specifically described herein without departing from the scope and teachings of the instant invention, and it is intended to encompass all other embodiments, alternatives, and modifications consistent with the invention.

CLAIMS

1. An automatically invoked intermediation process for purchasing content over a digital communications network by subscribing purchasers from subscribing merchants, the intermediation process comprising the steps of:

(a) establishing a database of subscriber-purchaser accounts and a database of subscriber-merchant accounts at a central transaction processing site on the digital communications network, the subscriber-merchant accounts database including information encoding resource locator data identifying at least one restricted-access port at each subscribing merchant site on the network and, for each such restricted-access port, information encoding an access fee schedule for accessing content by way of the restricted-access port and access-restriction-override information for enabling access to content by way of the restricted-access port, and the subscriber-purchaser accounts database including information encoding purchaser site authentication credentials and a purchaser account balance for each subscribing purchaser;

(b) at the network site of each subscribing purchaser, establishing a resource rules database including information encoding resource-locator-data identification criteria corresponding to each of at least a subset of the restricted-access ports at subscribing merchant sites identified in the subscriber-merchant accounts database and the access fee schedule for accessing content by way of the corresponding restricted-access port;

(c) at the network site of each subscribing purchaser, locally monitoring network communication activity information with respect to access to a target network resource generated by a browser program at the subscribing purchaser site and comparing such network communication activity information to the resource-locator-data identification criteria corresponding to restricted-access ports at the network sites of subscribing merchants in the resource rules database locally maintained at the site of the subscribing purchaser;

(d) in the event network communication activity information pertaining to the target network resource of step (c) matches with the resource-locator-data identification criteria corresponding to a restricted-access port in the resource rules database, retrieving the access fee schedule corresponding to the matched resource-locator-data identification criteria from the resource rules database, displaying an access fee from the access fee schedule for accessing content by way of the restricted access port corresponding to the resource-locator-data identification criteria, and prompting for approval or disapproval of completing the access for the displayed access fee;

(e) upon receipt at the subscribing purchaser site of a user communication responsive to the prompting of step (d) indicating approval of completing the access to content by way of the restricted-access port for the access fee, transmitting a purchase-request message from the subscribing purchaser site to the transaction processing site over the network, the purchase-request message including information encoding purchaser-site authentication credentials and identifying the restricted-access port at the merchant site corresponding to the matched resource-locator-data identification criteria;

(f) upon receipt of the purchase-request message of step (e) at the transaction processing site, determining whether the purchaser-site authentication credentials encoded in the purchase-request message matches with any purchaser-site authentication credentials included in the subscriber-purchaser accounts database and, if such a match is found, transmitting over the network to the restricted-access port of the merchant site a restricted-access-enabling access-request message to download purchased content by way of the restricted-access port to the transaction processing site, the restricted-access-enabling access-request message including access-restriction-override information corresponding to the restricted-access port retrieved from the subscriber-merchant accounts database;

(g) upon receipt of the downloaded purchased content at the transaction processing site from the subscribing-merchant site by way of the restricted-access port responsive to the restricted-access-enabling access-request message of step (f),

encrypting the downloaded purchased content and forwarding the thus encrypted purchased content to the subscribing purchaser site over the network;

(h) upon receipt of the encrypted purchased content of step (g) at the subscribing purchaser site, transmitting a content-received confirmation message to the transaction-processing site by the subscribing purchaser site over the network;

(i) upon receipt of the content-received confirmation message of step (h) from the subscribing purchaser site at the transaction processing site, debiting the purchaser account balance corresponding to the subscribing purchaser in the subscriber-purchaser accounts database by the access fee for accessing purchased content by way of the restricted-access port and forwarding a cryptographic key for decrypting the encrypted purchased content to the subscribing purchaser site over the network; and

(j) upon receipt of the cryptographic key at the subscribing purchaser site, decrypting the encrypted purchased content at the purchaser site.

2. The automatically invoked intermediation process according to claim 1 in which the access fee schedule corresponding to a restricted-access port in the subscriber-merchant accounts database consists of a single access fee for accessing content by way of the restricted-access port.

3. The automatically invoked intermediation process according to claim 1 further including the steps of parsing the downloaded purchased content at the transaction processing site prior to encrypting the purchased content to extract any resource locator data corresponding to software objects embedded in the content, retrieving the software objects over the digital communications network using the resource locator data, and combining the software objects with the purchased content to form a composite purchased-content data object, and in which the step of encrypting the downloaded purchased content is carried out by the encrypting the composite purchased-content data object.

4. The automatically invoked intermediation process according to claim 3 in which the downloaded purchased content is a hypertext mark-up language document and the software objects embedded in the document includes graphics image objects identified with universal resource locator data.

5. The automatically invoked intermediation process according to claim 1 further including the step of caching a copy of the downloaded purchased content at the transaction processing site for a time extending subsequent to the forwarding of the encrypted purchased content to the subscribing purchaser site.

6. The automatically invoked intermediation process according to claim 1 in which the access-restriction-override information corresponding to a restricted-access port in the subscriber-merchant accounts database includes a password for enabling access to content by way of the restricted-access port.

7. The automatically invoked intermediation process according to claim 1 in which the subscriber-purchaser accounts database includes for each subscribing purchaser information encoding an account-replenishment amount and a financial account number identifying an account at a financial institution upon which the subscribing purchaser may draw, the step (i) of debiting the purchaser account balance corresponding to the subscribing purchaser in the subscriber-purchaser accounts database by the access fee for accessing purchased content by way of the restricted-access port includes the step of comparing the purchaser account balance with an account minimum value and, if the purchaser account balance has fallen below the account minimum value, sending a funds-transfer request to the financial institution for transferring funds in the amount of the account-replenishment amount from the account identified by the financial account number to an intermediation entity account and, upon receipt at the transaction processing site of an approval of the transfer request, crediting the purchaser account balance by the account replenishment amount to bring the purchaser account balance above the account minimum value.

8. The automatically invoked intermediation process according to claim 7 in which the funds transfer request is sent from the transaction processing site to the financial institution over a communication channel separate from the digital communications network interconnecting the transaction processing site, the subscribing purchaser notes, and the subscribing merchant sites.

9. The automatically invoked intermediation process according to claim 1 in which the digital communications network interconnecting the transaction processing site, the subscribing purchaser sites, and the subscribing merchant sites is the Internet.

10. An automatically invoked intermediation process for purchasing content over a digital communications network by subscribing purchasers from subscribing merchants, the intermediation process comprising the steps of:

(a) establishing a database of subscriber-purchaser accounts and a database of subscriber-merchant accounts at a central transaction processing site on the digital communications network, the subscriber-merchant accounts database including information encoding resource locator data identifying at least one restricted-access port at each subscribing merchant site on the network and, for each such restricted-access port, information encoding an access fee schedule for accessing content by way of the restricted-access port and access-restriction-override information for enabling access to content by way of the restricted-access port, and the subscriber-purchaser accounts database including information encoding purchaser site authentication credentials and a purchaser account balance for each subscribing purchaser;

(b) at the network site of each subscribing purchaser, establishing a resource rules database including information encoding resource-locator-data identification criteria corresponding to each of at least a subset of the restricted-access ports at subscribing merchant sites identified in the subscriber-merchant accounts database and the access fee schedule for accessing content by way of the corresponding restricted-access port;

(c) at the network site of each subscribing purchaser, locally monitoring network communication activity information with respect to access to a target network resource generated by a browser program at the subscribing purchaser site and comparing such network communication activity information to the resource-locator-data identification criteria corresponding to restricted-access ports at the network sites of subscribing merchants in the resource rules database locally maintained at the site of the subscribing purchaser;

(d) in the event network communication activity information pertaining to the target network resource of step (c) matches with the resource-locator-data identification criteria corresponding to a restricted-access port in the resource rules database, retrieving the access fee schedule corresponding to the matched resource-locator-data identification criteria from the resource rules database, displaying an access fee from the access fee schedule for accessing content by way of the restricted access port corresponding to the resource-locator-data identification criteria, and prompting for approval or disapproval of completing the access for the displayed access fee;

(e) upon receipt at the subscribing purchaser site of a user communication responsive to the prompting of step (d) indicating approval of completing the access to content by way of the restricted-access port for the access fee, transmitting a purchase-request message from the subscribing purchaser site to the transaction processing site over the network, the purchase-request message including information encoding purchaser-site authentication credentials and identifying the restricted-access port at the merchant site corresponding to the matched resource-locator-data identification criteria;

(f) upon receipt of the purchase-request message of step (e) at the transaction processing site, determining whether the purchaser-site authentication credentials encoded in the purchase-request message matches with any purchaser-site authentication credentials included in the subscriber-purchaser accounts database and, if such a match is found, transmitting over the network to the restricted-access port of the merchant site a restricted-access-enabling access-request message to download

purchased content by way of the restricted-access port to the transaction processing site, the restricted-access-enabling access-request message including access-restriction-override information corresponding to the restricted-access port retrieved from the subscriber-merchant accounts database;

(g) upon receipt of the downloaded purchased content at the transaction processing site from the subscribing-merchant site by way of the restricted-access port responsive to the restricted-access-enabling access-request message of step (f), encrypting the downloaded purchased content and forwarding the thus encrypted purchased content to the subscribing purchaser site over the network;

(h) upon receipt of the encrypted purchased content of step (g) at the subscribing purchaser site, transmitting a content-received confirmation message to the transaction-processing site by the subscribing purchaser site over the network, the content received confirmation message including receipt-indicator the receipt-indicator data encoding whether or not a receipt data object with respect to purchased content had previously been received at the subscribing purchaser site;

(i) upon receipt of the content-received confirmation message of step (h) from the subscribing purchaser site at the transaction processing site, checking the receipt-indicator data to determine whether or not a receipt data object had previously been received with respect to the purchased content at the subscribing purchaser site, and, in the event no such receipt data object had been received, debiting the purchaser account balance corresponding to the subscribing purchaser in the subscriber-purchaser accounts database by the access fee for accessing purchased content by way of the restricted-access port, and forwarding a receipt-data object and a cryptographic key for decrypting the encrypted purchased content to the subscribing purchaser site over the network; and

(j) upon receipt of the cryptographic key at the subscribing purchaser site, decrypting the encrypted purchased content at the purchaser site.

11. The automatically invoked intermediation process according to claim 10 in which the receipt-indicator data included in the content received confirmation message of step (h) includes expiration date data encoding an expiration date for a receipt data object previously received at the subscribing purchaser site, and the step of checking the receipt indicator data of step (i) includes the step of comparing the expiration date data included in the receipt indicator data to a current date to determine if the expiration date encoded in the expiration date data has expired, and, in the event the expiration date has expired, debiting the purchaser account balance corresponding to the subscribing purchaser in the subscriber purchaser accounts database by the access fee for accessing purchased content by way of the restricted-access port.

12. The automatically invoked intermediation process according to claim 11 in which the receipt-indicator data included in the content received confirmation message of step (h) includes receipt verification check data for testing whether a receipt data object previously received at the subscribing purchaser site is genuine, and the step of checking the receipt indicator data of step (i) includes the step of testing the verification check data to determine if the receipt data object is genuine, and, in the event the receipt data object is not genuine, debiting the purchaser account balance corresponding to the subscribing purchaser in the subscribing purchaser accounts database by the access fee for accessing purchased content by way of the restricted-access port.

13. An automatically invoked intermediation process for purchasing hard goods over a digital communications network by subscribing purchasers from subscribing merchants, the intermediation process comprising the steps of:

(a) establishing a database of subscriber-purchaser accounts and a database of subscriber-merchant accounts at a central transaction processing site on the digital communications network, the subscriber-merchant accounts database including information encoding resource locator data identifying at least one order-entry port at each subscribing merchant site on the network and, for each such order-entry port, information encoding a schedule of prices and goods purchasable by orders entered by way of the order-entry port, and the subscriber-purchaser accounts database

including information encoding purchaser site authentication credentials and a purchaser account balance for each subscribing purchaser;

(b) at the network site of each subscribing purchaser, establishing a resource rules database including information encoding resource-locator-data identification criteria corresponding to each of at least a subset of the order-entry ports at subscribing merchant sites identified in the subscriber-merchant accounts database and the schedule of prices and goods purchasable by orders entered by way of the corresponding order-entry port;

(c) at the network site of each subscribing purchaser, locally monitoring network communication activity information with respect to access to a target network resource generated by a browser program at the subscribing purchaser site and comparing such network communication activity information to the resource-locator-data identification criteria corresponding to order-entry ports at the network sites of subscribing merchants in the resource rules database locally maintained at the site of the subscribing purchaser;

(d) in the event network communication activity information pertaining to the target network resource of step (c) matches with the resource-locator-data identification criteria corresponding to a order-entry port in the resource rules database, retrieving the schedule of prices and goods corresponding to the matched resource-locator-data identification criteria from the resource rules database, displaying at least a portion of prices and goods from the schedule, and prompting for identification of goods to be ordered;

(e) upon receipt at the subscribing purchaser site of a user communication responsive to the prompting of step (d) identifying hard goods to be ordered, transmitting an order-request message from the subscribing purchaser site to the transaction processing site over the network, the order-request message including information encoding purchaser-site authentication credentials, identifying the order-entry port at the merchant site corresponding to the matched resource-locator-data identification criteria, and specifying the hard goods to be ordered;

(f) upon receipt of the order-request message of step (e) at the transaction processing site, determining whether the purchaser-site authentication credentials encoded in the order-request message matches with any purchaser-site authentication credentials included in the subscriber-purchaser accounts database and, if such a match is found, transmitting over the network to the order-entry port of the merchant site an order-entry message to enter an order for the hard goods specified in the order-request message by way of the order-entry port and request transmission of an order-confirmation message from the merchant site to the transaction processing site;

(g) upon receipt of an order-confirmation message at the transaction processing site from the subscribing merchant site responsive to the order-entry message of step (f), encrypting the order-confirmation message and forwarding the encrypted order confirmation message to the subscribing purchaser site over the digital communications network;

(h) upon receipt of the encrypted order-confirmation message of step (g) at the subscribing purchaser site, transmitting an order-confirmation received message to the transaction-processing site by the subscribing purchaser site over the network;

(i) upon receipt of the order-confirmation message of step (h) from the subscribing purchaser site at the transaction processing site, debiting the purchaser account balance corresponding to the subscribing purchaser in the subscriber-purchaser accounts database by the total price of the hard goods ordered and forwarding a cryptographic key for decrypting the encrypted order-confirmation message to the subscribing purchaser site over the network; and

(j) upon receipt of the cryptographic key at the subscribing purchaser site, decrypting the encrypted order-confirmation message at the purchaser site.

14. The automatically invoked intermediation process according to claim 13 in which the subscriber-purchaser accounts database includes for each subscribing purchaser information encoding an account-replenishment amount and a financial account number identifying an

account at a financial institution upon which the subscribing purchaser may draw, the step (i) of debiting the purchaser account balance corresponding to the subscribing purchaser in the subscriber-purchaser accounts database by the total price of the hard goods ordered includes the step of comparing the purchaser account balance with an account minimum value and, if the purchaser account balance has fallen below the account minimum value, sending a funds-transfer request to the financial institution for transferring funds in the amount of the account-replenishment amount from the account identified by the financial account number to an intermediation entity account and, upon receipt at the transaction processing site of an approval of the transfer request, crediting the purchaser account balance by the account replenishment amount to bring the purchaser account balance above the account minimum value.

15. The automatically invoked intermediation process according to claim 14 in which the funds transfer request is sent from the transaction processing site to the financial institution over a communication channel separate from the digital communications network interconnecting the transaction processing site, the subscribing purchaser notes, and the subscribing merchant sites.

16. The automatically invoked intermediation process according to claim 13 in which the digital communications network interconnecting the transaction processing site, the subscribing purchaser sites, and the subscribing merchant sites is the Internet.

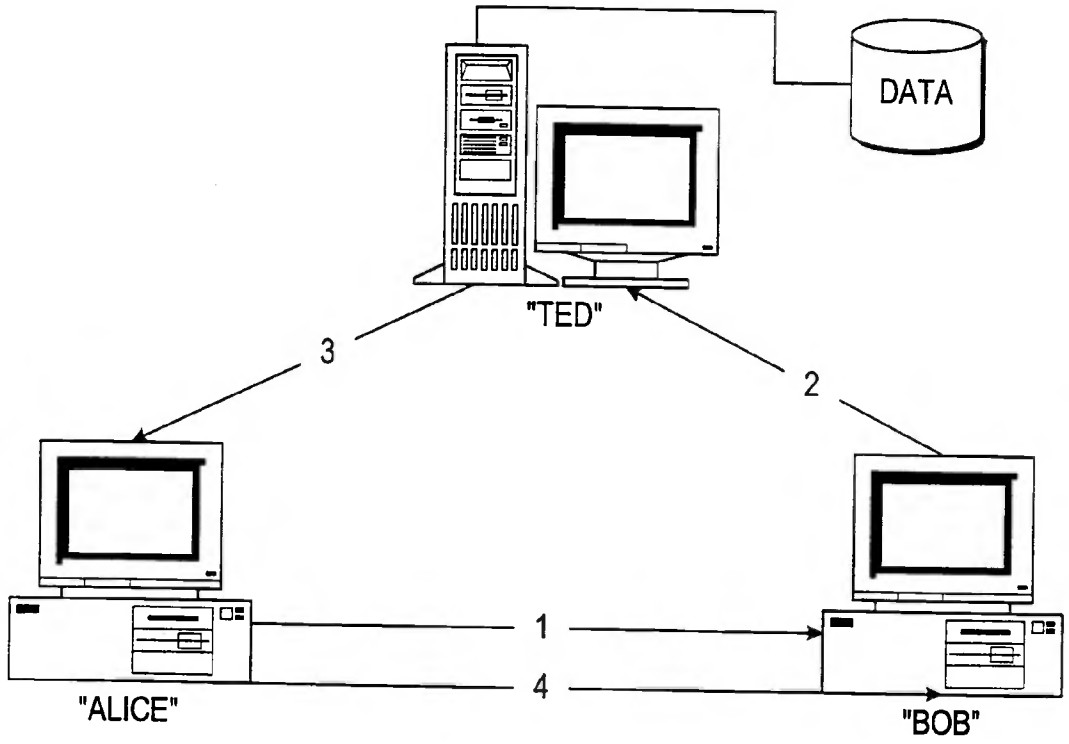


FIG. 1

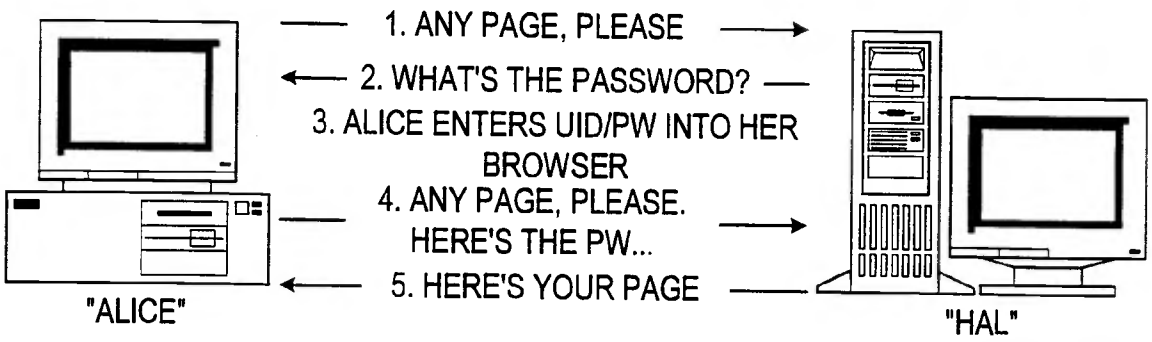


FIG. 2
PRIOR ART

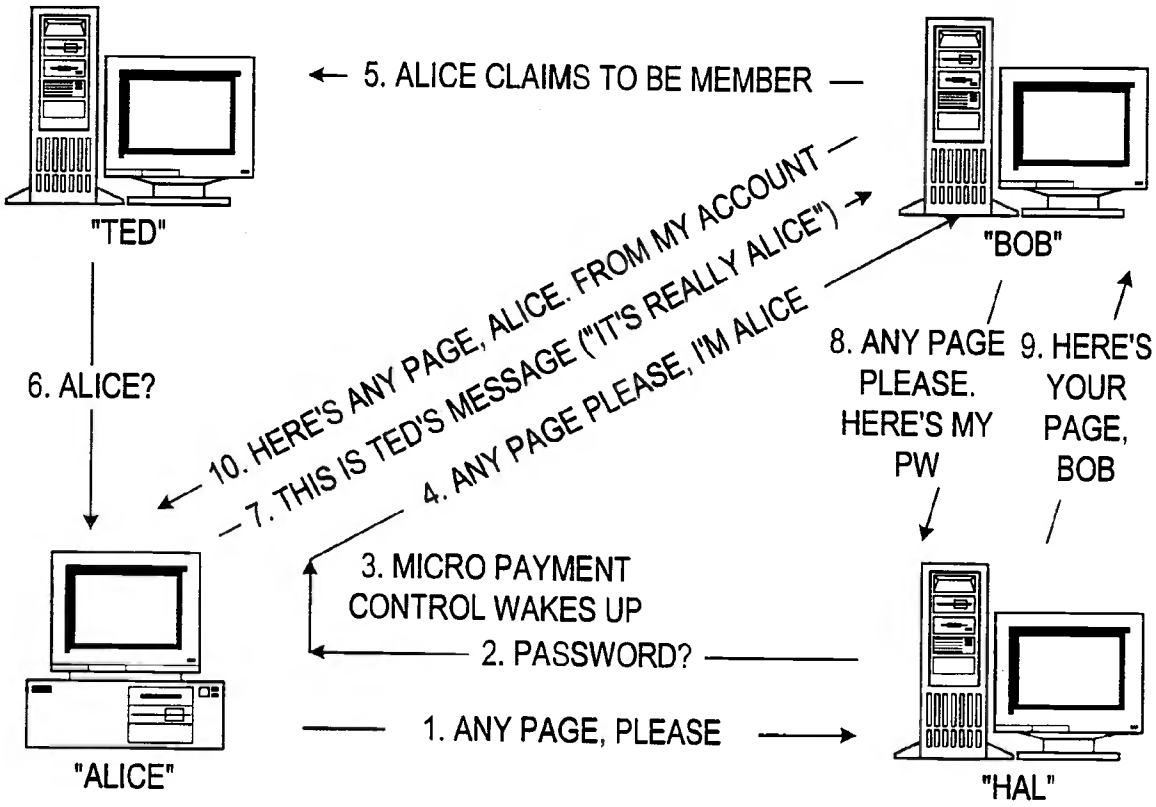

5. ALICE CLAIMS TO BE MEMBER
"TED"
"BOB"
6. ALICE?
10. HERE'S ANY PAGE, ALICE. FROM MY ACCOUNT
7. THIS IS TED'S MESSAGE ("IT'S REALLY ALICE")
4. ANY PAGE PLEASE, I'M ALICE
8. ANY PAGE PLEASE. HERE'S MY PW
9. HERE'S YOUR PAGE, BOB
3. MICRO PAYMENT CONTROL WAKES UP
2. PASSWORD?
1. ANY PAGE, PLEASE
"ALICE"
"HAL"

FIG. 3

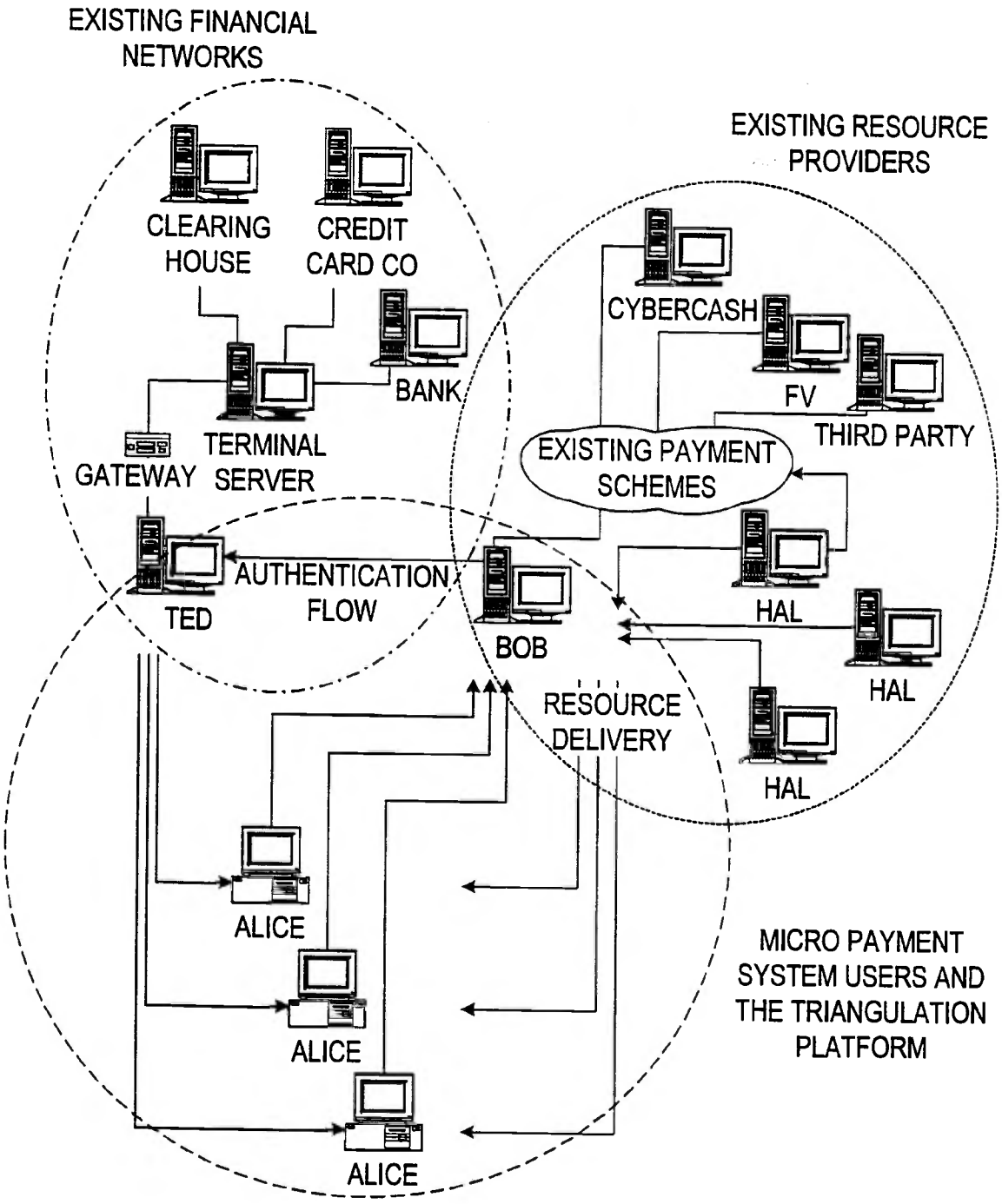

EXISTING FINANCIAL NETWORKS
CLEARING HOUSE
CREDIT CARD CO
BANK
TERMINAL SERVER
GATEWAY
TED
AUTHENTICATION FLOW
EXISTING RESOURCE PROVIDERS
CYBERCASH
FV
THIRD PARTY
EXISTING PAYMENT SCHEMES
BOB
HAL
HAL
HAL
RESOURCE DELIVERY
ALICE
ALICE
ALICE
MICRO PAYMENT SYSTEM USERS AND THE TRIANGULATION PLATFORM

FIG. 4

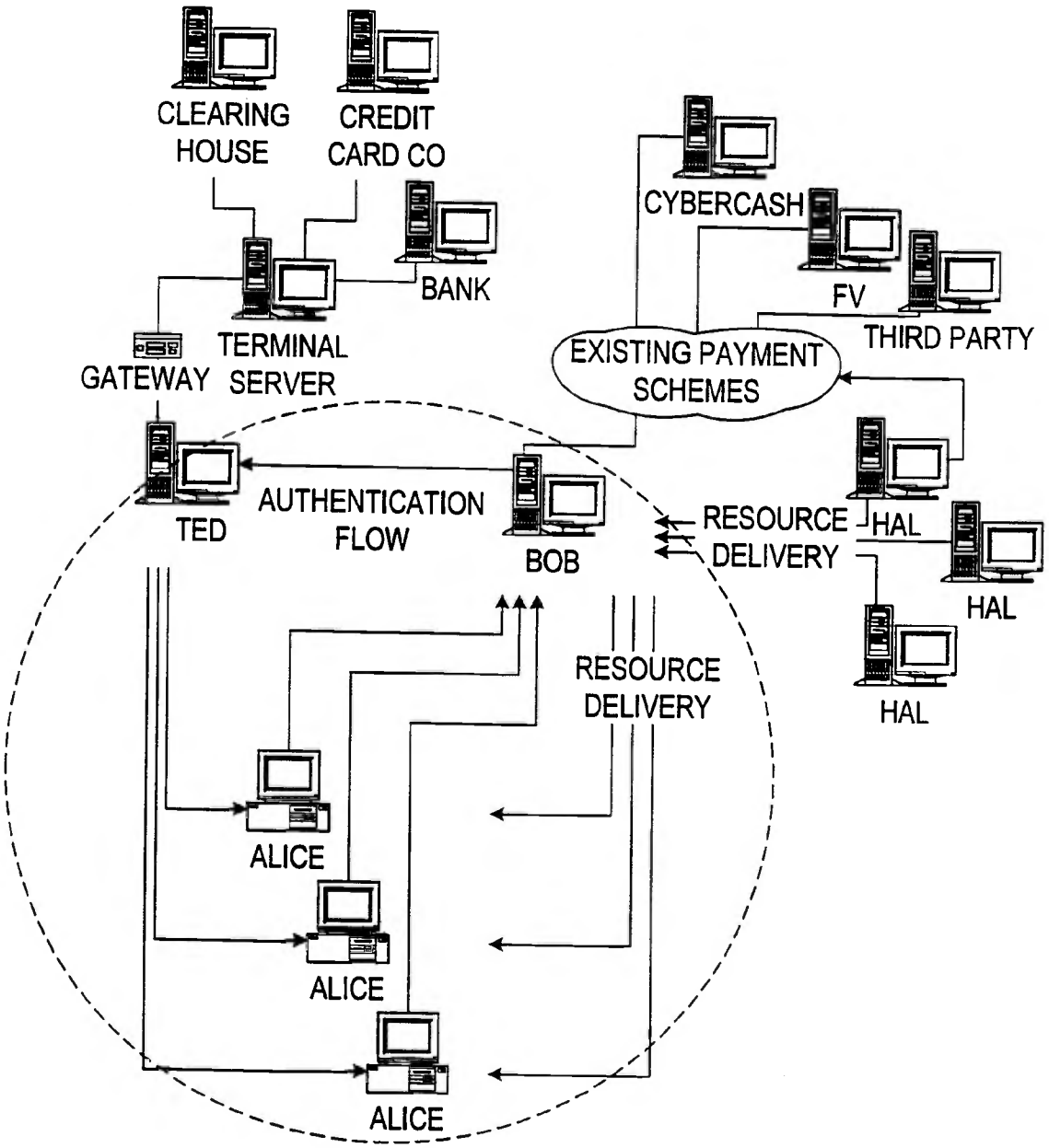

CLEARING HOUSE
CREDIT CARD CO
BANK
TERMINAL SERVER
GATEWAY
CYBERCASH
FV
THIRD PARTY
EXISTING PAYMENT SCHEMES
TED
AUTHENTICATION FLOW
BOB
RESOURCE DELIVERY
HAL
HAL
HAL
RESOURCE DELIVERY
ALICE
ALICE
ALICE

FIG. 5

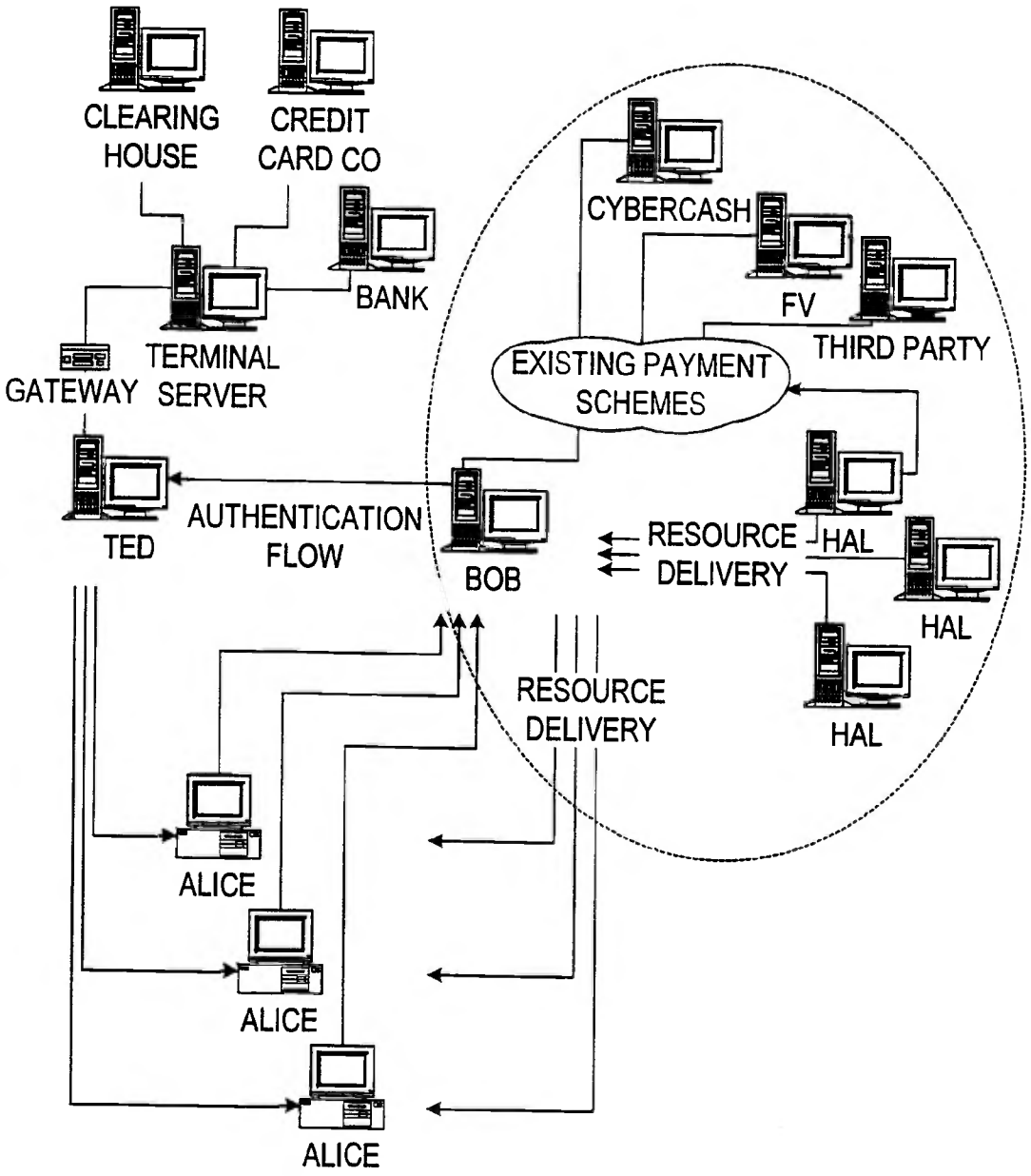

CLEARING HOUSE
CREDIT CARD CO
BANK
TERMINAL SERVER
GATEWAY
TED
AUTHENTICATION FLOW
BOB
CYBERCASH
FV
THIRD PARTY
EXISTING PAYMENT SCHEMES
RESOURCE DELIVERY
HAL
HAL
HAL
RESOURCE DELIVERY
ALICE
ALICE
ALICE

FIG. 6

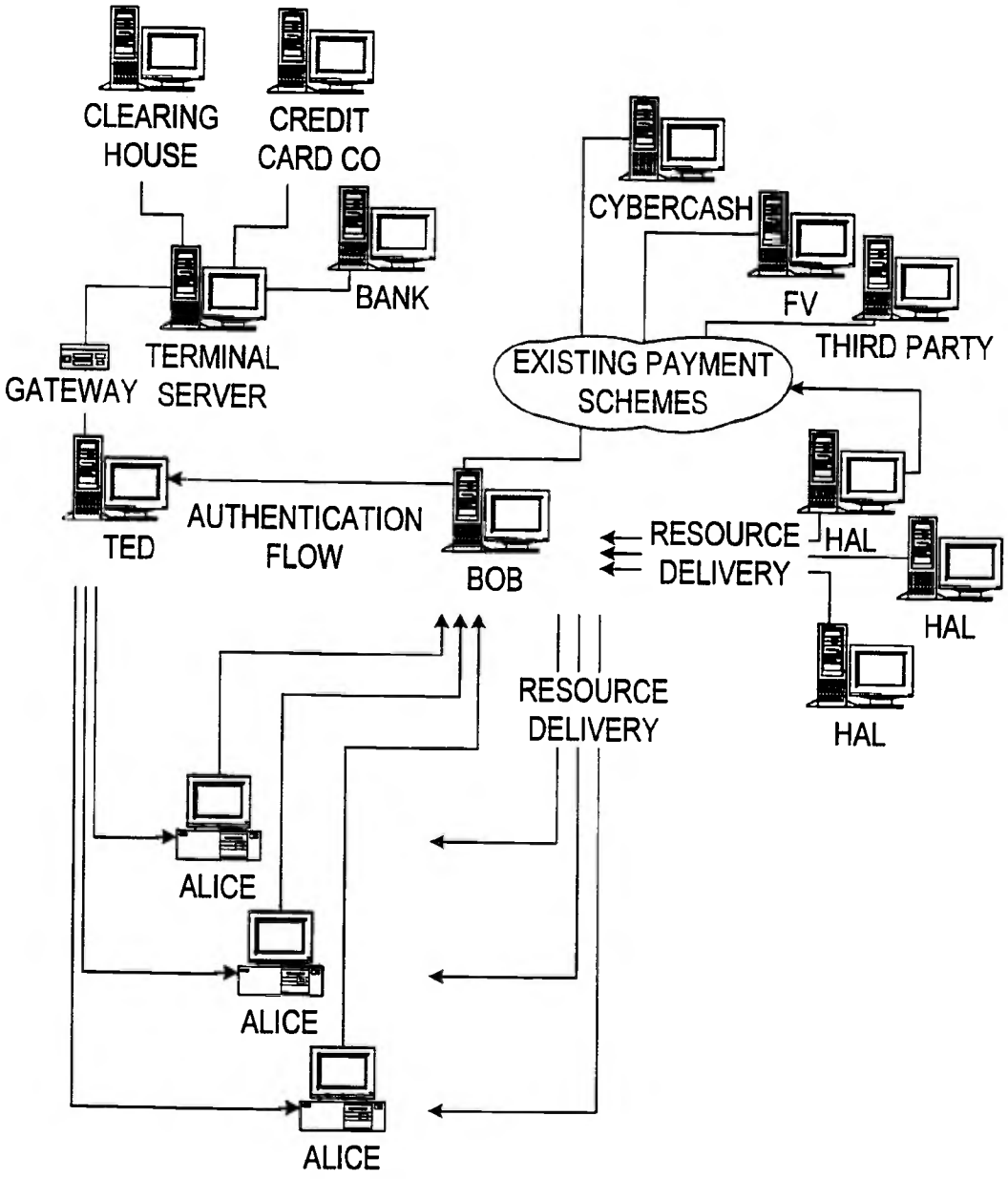

CLEARING HOUSE
CREDIT CARD CO
BANK
TERMINAL SERVER
GATEWAY
CYBERCASH
FV
THIRD PARTY
EXISTING PAYMENT SCHEMES
TED
AUTHENTICATION FLOW
BOB
RESOURCE DELIVERY
HAL
HAL
HAL
RESOURCE DELIVERY
ALICE
ALICE
ALICE

FIG. 7

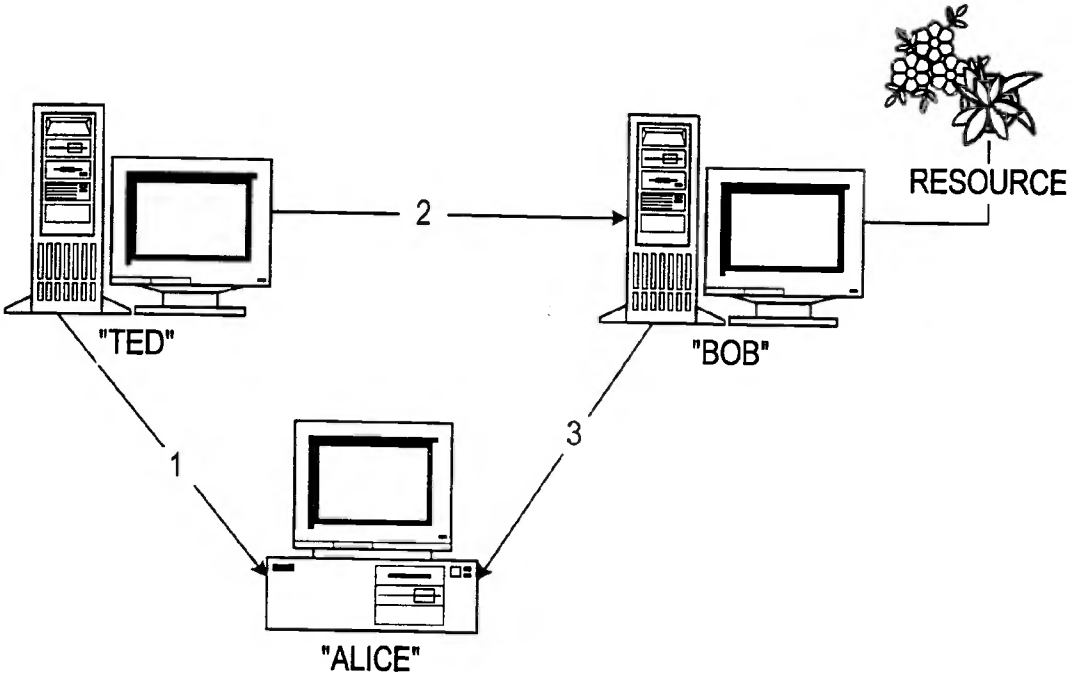


FIG. 8

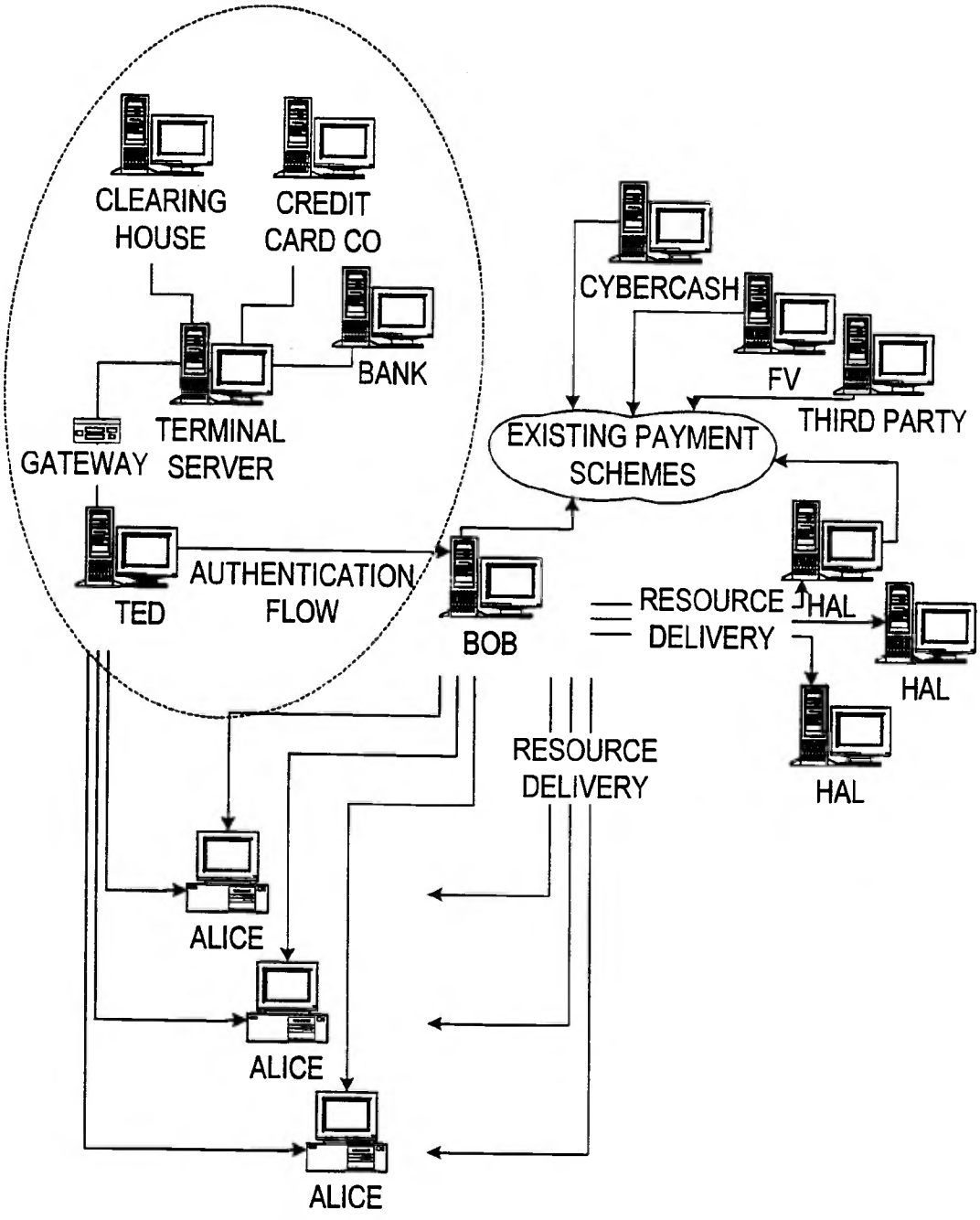

CLEARING HOUSE
CREDIT CARD CO
BANK
TERMINAL SERVER
GATEWAY
TED
AUTHENTICATION FLOW
CYBERCASH
FV
THIRD PARTY
EXISTING PAYMENT SCHEMES
BOB
RESOURCE DELIVERY
HAL
HAL
HAL
RESOURCE DELIVERY
ALICE
ALICE
ALICE

FIG. 9

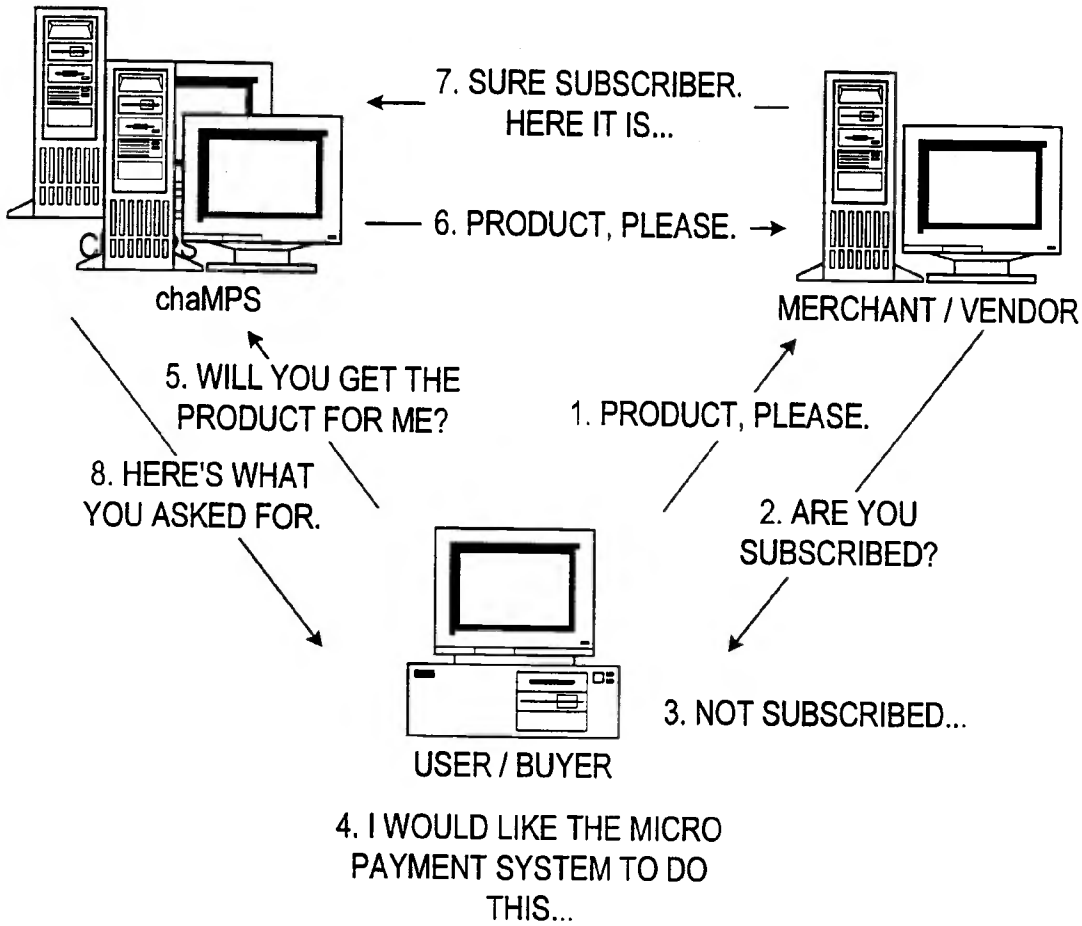

7. SURE SUBSCRIBER. HERE IT IS...
6. PRODUCT, PLEASE.
chaMPS
MERCHANT / VENDOR
5. WILL YOU GET THE PRODUCT FOR ME?
1. PRODUCT, PLEASE.
8. HERE'S WHAT YOU ASKED FOR.
2. ARE YOU SUBSCRIBED?
3. NOT SUBSCRIBED...
USER / BUYER
4. I WOULD LIKE THE MICRO PAYMENT SYSTEM TO DO THIS...

FIG. 10

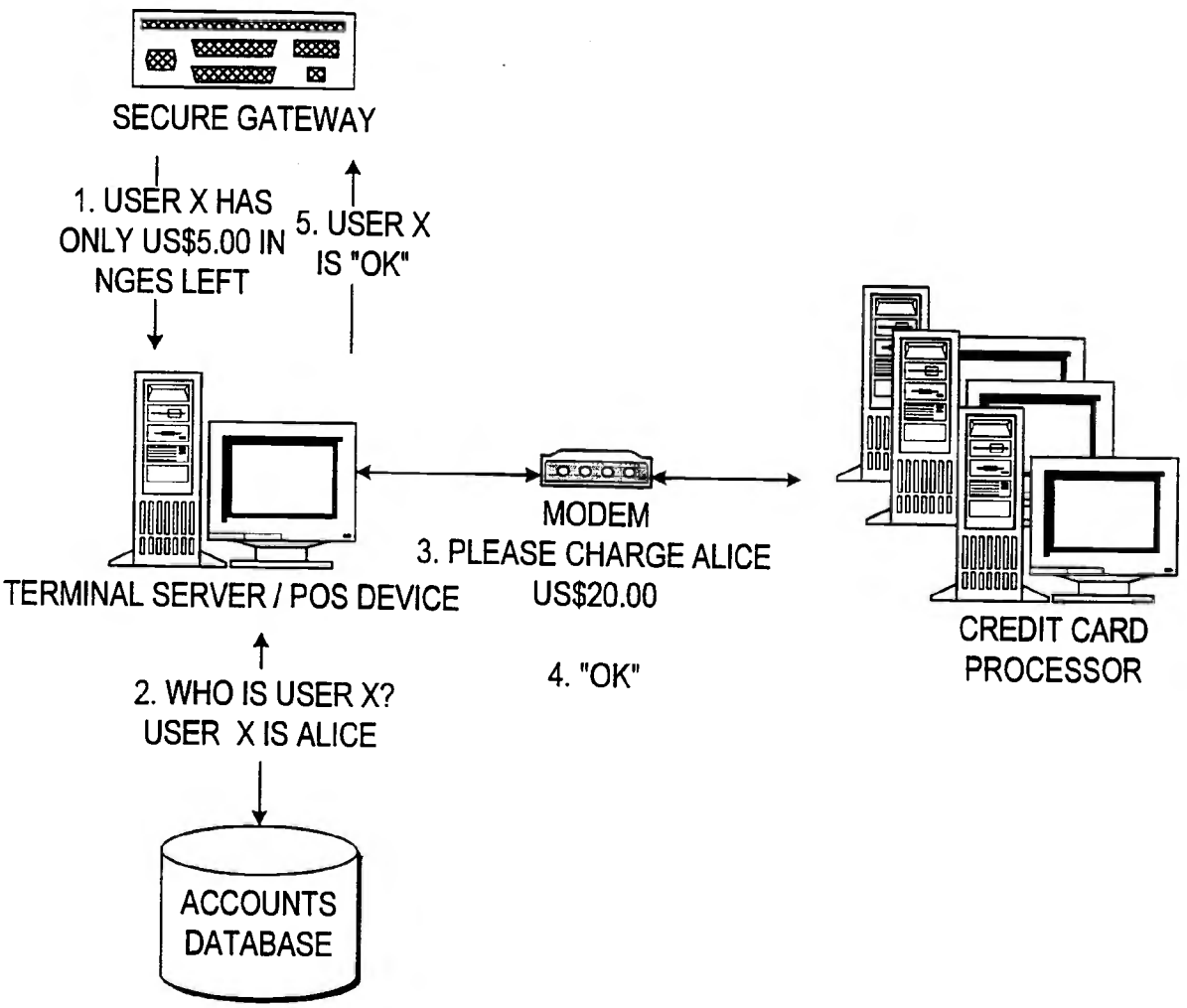

SECURE GATEWAY
1. USER X HAS ONLY US$5.00 IN NGES LEFT
5. USER X IS "OK"
MODEM
3. PLEASE CHARGE ALICE US$20.00
4. "OK"
TERMINAL SERVER / POS DEVICE
2. WHO IS USER X? USER X IS ALICE
ACCOUNTS DATABASE
CREDIT CARD PROCESSOR

FIG. 11

FIG. 12A

MERCHANT ACCOUNT PRE-APPLICATION

TO GET STARTED, FILL OUT THIS PRE-APPLICATION AND FAX IT OR SEND IT TO POINT OF SALE SYSTEMS. ALONG WITH A CHECK FOR $75, WHICH WILL BE CREDITED TOWARDS THE APPLICATION FEE. IF YOU ARE GOING WITH OUR $20 DOWN PROGRAM, YOU WOULD SIMPLY SEND US THE $20. **THIS FEE IS REFUNDED TO YOU IF YOU ARE DECLINED FOR ANY REASON EXCEPT FOR DEROGATORY CREDIT OR LACK OF CREDIT.** IF YOU FAX A CHECK, WE WILL REPRODUCE IT AND DEPOSIT IT AT OUR BANK, PNC BANK IN BENSALEM, PA. YOU WILL RECEIVE IT BACK AS A CANCELED CHECK WITH YOUR OTHER CHECKS. WE REQUEST THIS FEE IN ADVANCE BECAUSE WE DO NOT SIMPLY SEND OUT APPLICATIONS TO PEOPLE WHO ARE STILL SEARCHING FOR A BETTER DEAL OR COMPARING OFFERS OR WHO ARE NOT READY TO GET SET UP YET. APPLICATIONS AND POSTAGE COST US ABOUT $3 EACH. SINCE ONLY 1 OUT OF 10 PEOPLE SEND APPLICATIONS BACK, IT COSTS US ABOUT $30 IN WASTED EXPENSE FOR EACH MERCHANT WHO APPLIES. IF YOU SIMPLY WANT AN APPLICATION AND ARE NOT COMMITED TO SIGNING UP, SEND US $30 AND WE WILL SEND YOU A BLANK APPLICATION. IF YOU WANT YOUR $30 BACK, RETURN THE BLANK APPLICATION TO US. WHEN WE RECEIVE THIS APPLICATION, WE WILL FILL OUT A HARD COPY VERSION AND RETURN IT TO YOU FOR YOUR SIGNATURE. WHEN YOU SEND IT BACK IN, YOU WOULD THEN SEND IN THE REQUIRED SUPPORTING MATERIAL, SUCH AS COPIES OF ANY ADS, BROCHURES OR FLIERS, OR A PRINT OUT OF YOUR EXISTING OR PROPOSED WEB SITE, ALONG WITH A PHOTO OF THE OUTSIDE OF YOUR HOME OR BUSINESS, SHOWING THE PHYSICAL ADDRESS. AN INSIDE PHOTO SHOWING YOUR PRODUCT, INVENTORY OR OFFICE SET UP IS ALSO REQUIRED. DEPENDING ON THE PROGRAM YOU HAVE SELECTED, A COPY OF A BUSINESS LICENSE OR BUSINESS NAME REGISTRATION MAY BE REQUIRED.

BUSINESS INFORMATION

LEGAL NAME OF BUSINESS:	BUSINESS START DATE:____# LOCATIONS:____
DBA NAME:	MONTHLY VOLUME:____AVERAGE TICKET:____
PHYSICAL ADDRESS:	TYPE OF BUSINESS:__RETAIL,__MAIL ORDER,__SERVICE
CITY, SATE, ZIP:	PRODUCT OR SERVICE SOLD:
CONTACT NAME:	CARD PRESENT:___% CARD NOT PRESENT:___%
PHONE NUMBER:	FAX NUMBER:

OWNER INFORMATION

__SOLE PROP	__PARTNERSHIP	__CORP	__OTHER:___	TAX ID#(CAN USE SSN):
OWNER 1/PARTNER/OFFICER NAME:		TITLE IN BUSINESS:		SOCIAL SECURITY #:
HOME ADDRESS:		CITY/ST/ZIP:		PHONE NUMBER:
OWNER 2/PARTNER/OFFICER NAME:		TITLE IN BUSINESS:		SOCIAL SECURITY #:
HOME ADDRESS:		CITY/ST/ZIP:		PHONE NUMBER:

FIG. 12B

BANK & TRADE REFERENCE INFORMATION

BANK NAME:	CONTACT:	PHONE NUMBER:	ACCOUNT #:
TRADE REFERENCE:	CONTACT:	PHONE NUMBER:	ACCOUNT #:
TRADE REFERENCE:	CONTACT:	PHONE NUMBER:	ACCOUNT #:
TRADE REFERENCE:	CONTACT:	PHONE NUMBER:	ACCOUNT #:

FEE SCHEDULE INFORMATION

SELECT THE APPROPRIATE PROGRAM BASED ON THE RATES & FEES SECTION OF OUR INFORMATION.

PROGRAM 1 (PROCESS BY TELEPHONE)

DISCOUNT RATE: 2.49%	TRANSACTION CHARGE: 50 CENTS	MONTHLY MIN. FEE: $20	STATEMENT. FEE: $20

SET UP FEE: $295, INCLUDES THE $75 APPLICATION FEE PAID WITH THIS PRE-APPLICATION. UPON APPROVAL, YOU WILL RECEIVE INSTRUCTIONS ON HOW TO USE THE PHONE PROCESS METHOD. NO EQUIPMENT IS NEEDED. ADDRESS VERIFICATION IS SUPPORTED ON THIS SYSTEM. THIS PROGRAM IS FOR TANGIBLE PRODUCTS ONLY, NOT SERVICES. THIS SERVICE IS PROVIDED BY MCCS, AN AGENT FOR WOODFOREST BANK, HOUSTON TX. MEMBER FDIC

PROGRAM 2 (NO MONTHLY STATEMENT FEE, $20 MONTHLY MINIMUM FEE)

CARD SWIPED DISCOUNT RATE (NON-COMPUTERS): 1.49%	TRANSACTION CHARGE: 20 CENTS
CARD SWIPED DISCOUNT RATE (COMPUTERS): 1.69%	TRANSACTION CHARGE: 20 CENTS

A 1% KEYED-IN SURCHARGE APPLIES TO THE ABOVE DISCOUNT RATES.

MAIL/PHONE ORDER DISCOUNT RATE (NON-COMPUTERS): 2.29%	TRANSACTION CHARGE: 30 CENTS
MAIL/PHONE ORDER DISCOUNT RATE (COMPUTERS): 2.69%	TRANSACTION CHARGE: 30 CENTS

THIS PROGRAM REQUIRES GOOD PERSONAL CREDIT (NO EXCESSIVE CHARGEOFFS OF COLLECTION ACCOUNTS). CREDIT PROBLEMS DUE TO MEDICAL BILLS OR OTHER UNEXPECTED EMERGENCIES ARE OVERLOOKED. DEPENDING ON CREDIT AND TYPE OF BUSINESS, PROCESSING BANK MAY REQUIRE COPIES OF LAST YEAR'S TAX RETURN AND THREE MONTH'S MOST RECENT CHECKING ACCOUNT STATEMENTS, TO VERIFY THAT YOU HAVE AN INCOME OUTSIDE OF CREDIT CARD SALES.

PROGRAM 3 (NO MONTHLY MINIMUM FEE, $10 MONTHLY STATEMENT FEE)

CARD SWIPED DISCOUNT RATE (NON-COMPUTERS): 1.49%	TRANSACTION CHARGE: 20 CENTS
CARD SWIPED DISCOUNT RATE (COMPUTERS): 1.69%	TRANSACTION CHARGE: 20 CENTS

A 1% KEYED-IN SURCHARGE APPLIES TO THE ABOVE DISCOUNT RATES. A7 DAY RESERVE ACCOUNT MAY BE NEEDED ON NEW OR MAIL ORDER BUSINESSES. *A $50 SITE INSPECTION FEE IS ALSO REQUIRED. PROPERTY RESOURCE NETWORK WILL CALL YOU TO SET UP AN APPOINTMENT TO VISIT YOU TO VERIFY YOUR BUSINESS AND TAKE THE NEEDED PHOTOS.*

MAIL/PHONE ORDER DISCOUNT RATE (NON-COMPUTERS): 1.99%	TRANSACTION CHARGE: 30 CENTS
MAIL/PHONE ORDER DISCOUNT RATE (COMPUTERS): 2.69%	TRANSACTION CHARGE: 30 CENTS

THIS PROGRAM REQUIRES A COPY OF A BUSINESS LICENSE, A FICTITIOUS NAME REGISTRATION OR A COPY OF YOUR CERTIFICATE OF INCORPORATION. A BUSINESS CHECKING ACCOUNT IS ALSO NEEDED. A COPY OF A DRIVER'S LICENSE IS ALSO NEEDED

PROGRAM 4 (FOR MERCHANTS WITH LIMITED OR BAD PERSONAL CREDIT - $95 APP FEE)		
CARD SWIPED DISCOUNT RATE (NON-COMPUTERS): 1.69%	TRANSACTION CHARGE: 20 CENTS	
CARD SWIPED DISCOUNT RATE (COMPUTERS): 1.89%	TRANSACTION CHARGE: 20 CENTS	
A 1% KEYED-IN SURCHARGE APPLIES TO THE ABOVE DISCOUNT RATES. THERE IS A *$25* MONTHLY MINIMUM FEE AND A *$10* MONTHLY STATEMENT FEE. THE APPLICATION FEE IS $95.		
MAIL/PHONE ORDER DISCOUNT RATE (NON-COMPUTER):2.69%	TRANSACTION CHARGE: 30 CENTS	
MAIL/PHONE ORDER DISCOUNT RATE (COMPUTERS): 2.99%	TRANSACTION CHARGE: 30 CENTS	

BASED ON MY BUSINESS TYPE AND NEEDS, I HAVE SELECTED PROGRAM ___.
MY TYPE OF BUSINESS IS__CARD SWIPED, __NON-CARD SWIPED(**SELECT ONE**).

EQUIPMENT SELECTION			
EQUIPMENT TYPE	PURCHASE PRICE	24 MONTH LEASE	SUPPORTS AVS ?
VERIFONE ZON XL	$395	$27/MONTH	NO
VERIFONE PRINTER	$295	$18/MONTH	N/A
VERIFONE TRANZ 330	$495	$29/MONTH	PROGRAMS 3 & 4
VERIFONE TRANZ 380	$595	$34/MONTH	PROGRAMS 2,3 & 4
VERIFONE TRANZ 460	$695	$39/MONTH	PROGRAMS 3 & 4
VERIFONE TRANZ 420	$795	$44/MONTH	PROGRAMS 3 & 4
HYPERCOM T7P	$695	$39/MONTH	PROGRAMS 2,3 & 4
POS CHARGE	$395	$27/MONTH	PROGRAMS 2,3 & 4
MAC AUTHORIZE	$495	$29/MONTH	PROGRAMS 2,3 & 4
AN OPTIONAL 24 MONTH LEASE IS AVAILABLE INSTEAD OF PURCHASING THE EQUIPMENT. THE LEASING COMPANY ALSO CHARGES YOUR STATE SALES TAX RATE PLUS A $2.50 PER MONTH LOSS AND DESTRUCTION WAIVER. LEASE APPROVAL IS BASED ON YOUR PERSONAL CREDIT.			

I HAVE SELECTED THE FOLLOWING EQUIPMENT:_____. I WISH TO __PURCHASE. __LEASE IT (**SELECT ONE**).
I ALREADY HAVE THE FOLLOWING EQUIPMENT:_____. I WISH TO HAVE IT REPROGRAMMED FOR FEE OF $150, IN ADDITION TO THE APPLICATION FEE.

FAX OR MAIL THIS PRE-APPLICATION TODAY!

I WISH TO APPLY FOR A MERCHANT ACCOUNT BASED ON THE ABOVE INFORMATION. I AM ENCLOSING A CHECK MADE PAYABLE TO POINT OF SALE SYSTEMS FOR $75, WHICH IS REFUNDED IF MY ACCOUNT IS DECLINED FOR ANY REASON EXCEPT FOR DEROGATORY OR LACK OF CREDIT.
SIGNED: _____
DATE: _____
FAX THIS PAPERWORK AND A CHECK TO 215-245-5770. POINT OF SALE SYSTEMS MAINTAINS A CHECKING ACCOUNT AT PNC BANK, BENSALEM, PA. YOUR CHECK WILL BE PRINTED ON OUR CHECK PRINTING SYSTEM AND DEPOSITED. YOU WILL GET IT BACK AS A CANCELED CHECK. YOU CAN ALSO MAIL THIS PACKAGE TO:

POINT OF SALE SYSTEMS
1234 BAYBERRY RD
BENSALEM, PA 19020

I WISH TO HAVE MY WEB PAGE REGISTERED WITH OVER 200 SEARCH ENGINES AND DIRECTORIES AT NO COST. I HAVE PROVIDED THE FOLLOWING INFORMATION:
WEB PAGE URL: _____
EMAIL ADDRESS:

FIG. 12C

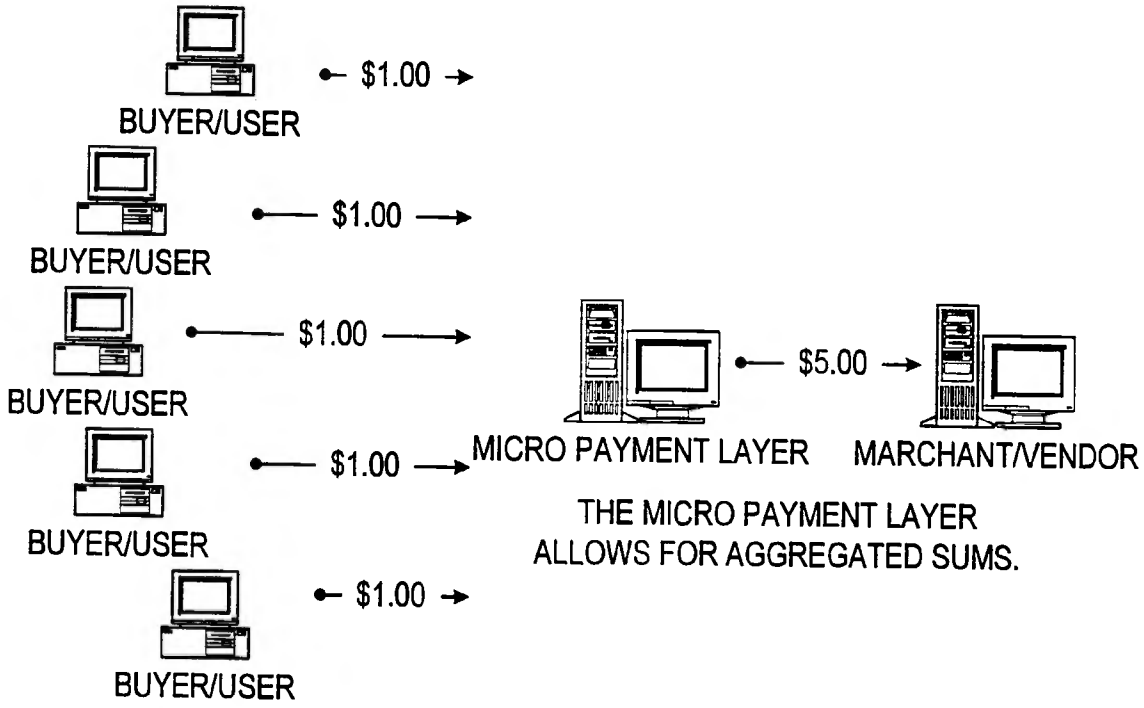

$1.00
BUYER/USER
$1.00
BUYER/USER
$1.00
BUYER/USER
$1.00
BUYER/USER
$1.00
BUYER/USER
MICRO PAYMENT LAYER
$5.00
MARCHANT/VENDOR
THE MICRO PAYMENT LAYER
ALLOWS FOR AGGREGATED SUMS.

FIG. 13

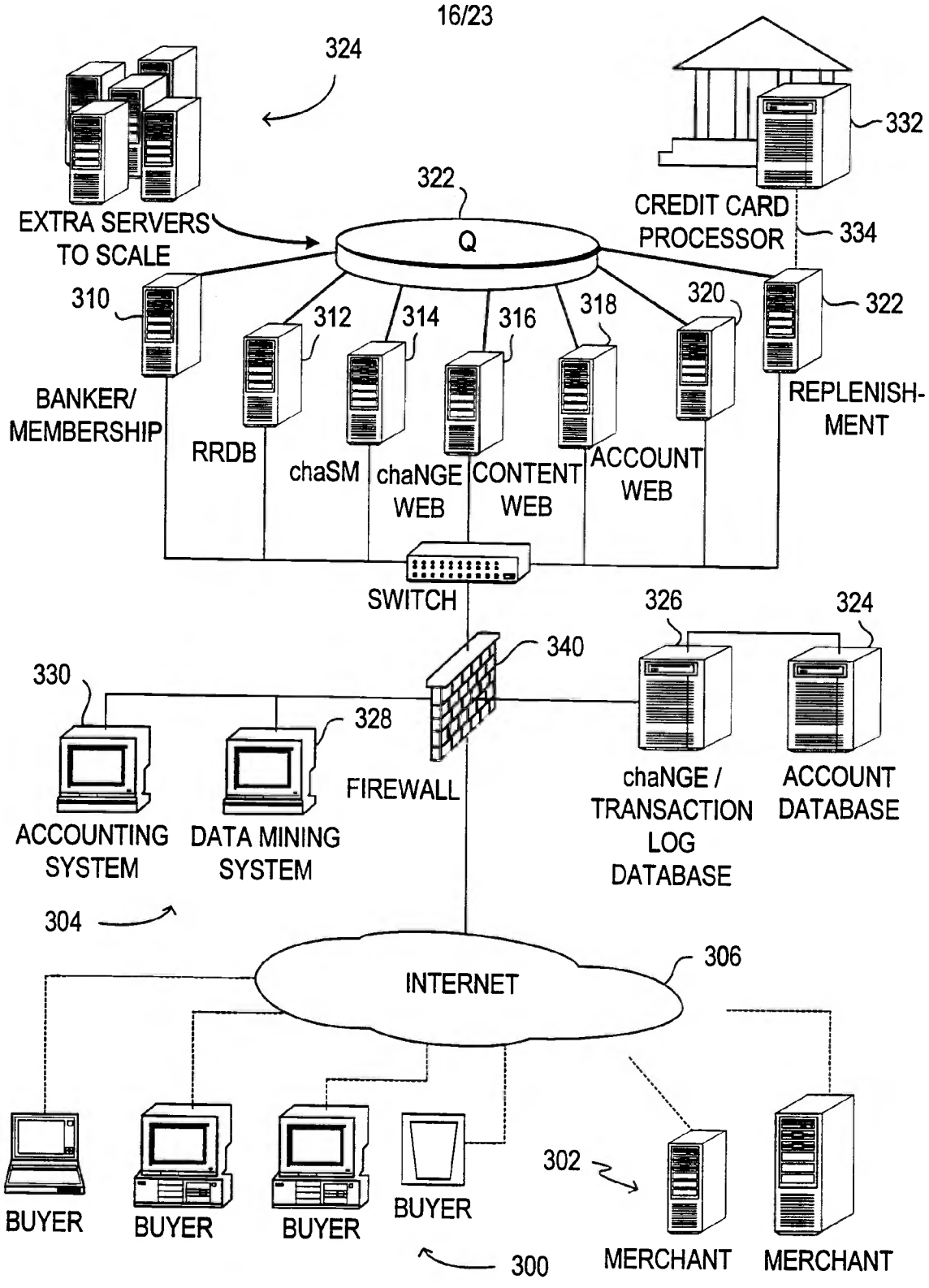

324
EXTRA SERVERS TO SCALE
322
Q
332
CREDIT CARD PROCESSOR
334
310
312
314
316
318
320
322
BANKER/ MEMBERSHIP
RRDB
chaSM
chaNGE WEB
CONTENT WEB
ACCOUNT WEB
REPLENISH-MENT
SWITCH
326
324
340
330
328
FIREWALL
chaNGE / TRANSACTION LOG DATABASE
ACCOUNT DATABASE
ACCOUNTING SYSTEM
DATA MINING SYSTEM
304
306
INTERNET
BUYER
BUYER
BUYER
BUYER
302
MERCHANT
MERCHANT
300
INTERNAL NETWORK CONNECTION
EXTERNAL NETWORK CONNECTION
TRANSACTION QUEUE CONNECTION

FIG. 14

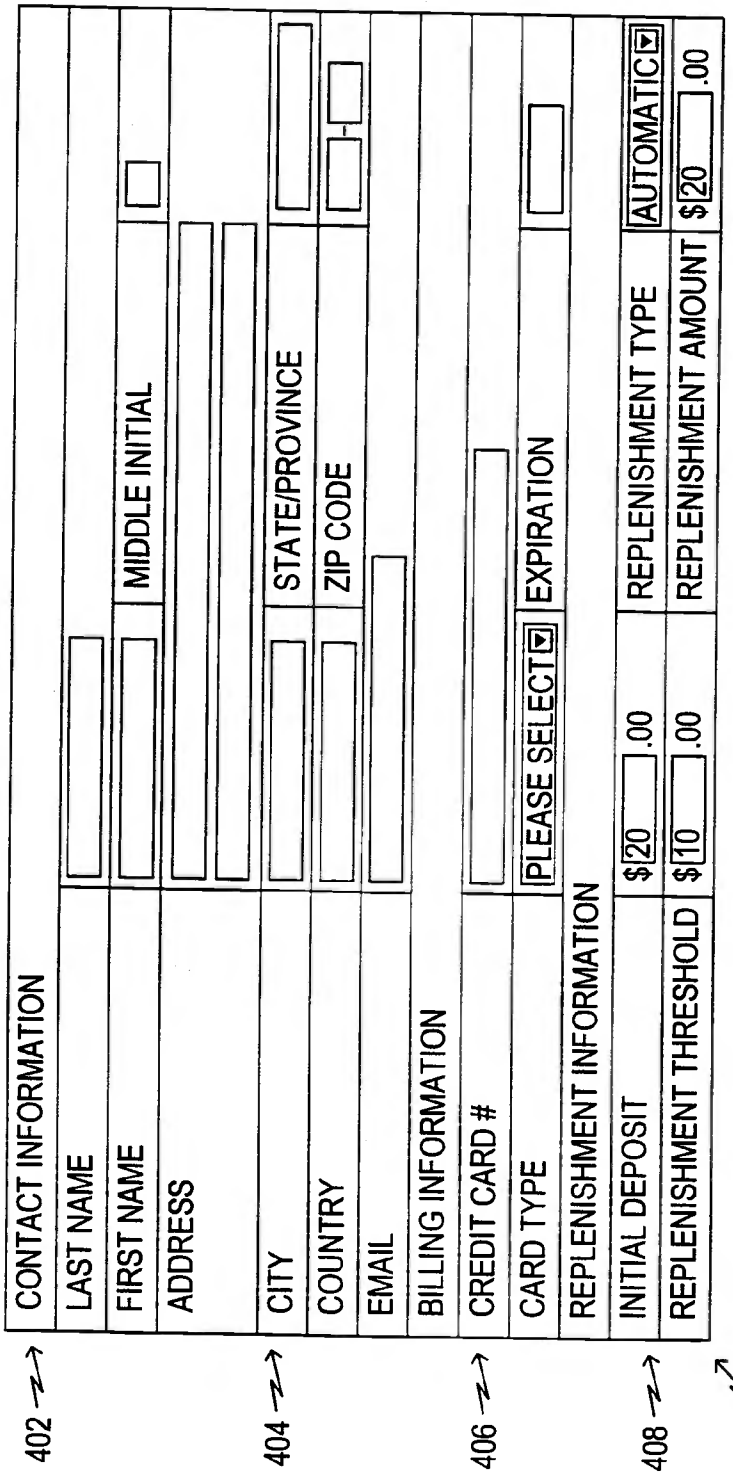

CONTACT INFORMATION
LAST NAME
FIRST NAME
MIDDLE INITIAL
ADDRESS
CITY
STATE/PROVINCE
COUNTRY
ZIP CODE
EMAIL
BILLING INFORMATION
CREDIT CARD #
CARD TYPE
PLEASE SELECT
EXPIRATION
REPLENISHMENT INFORMATION
INITIAL DEPOSIT
$20 .00
REPLENISHMENT TYPE
AUTOMATIC
REPLENISHMENT THRESHOLD
$10 .00
REPLENISHMENT AMOUNT
$20 .00
402
404
406
408
400

FIG. 15

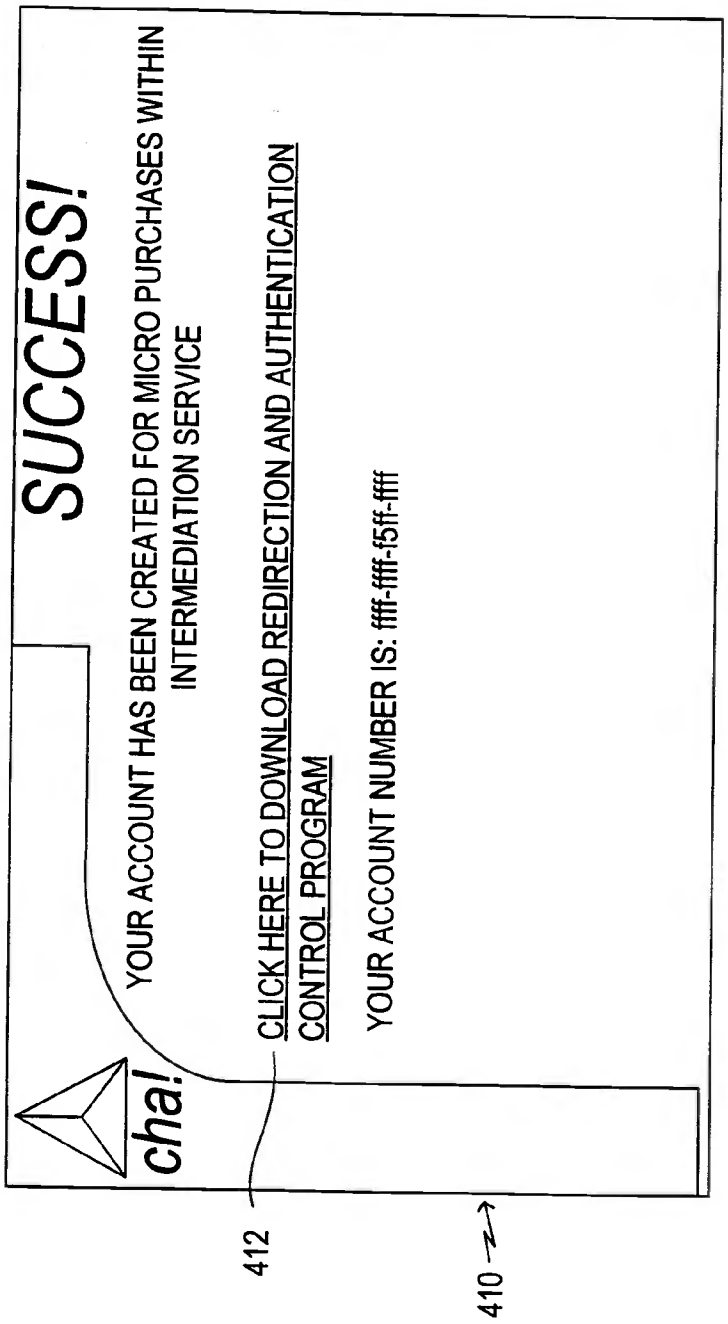

SUCCESS!
YOUR ACCOUNT HAS BEEN CREATED FOR MICRO PURCHASES WITHIN INTERMEDIATION SERVICE
CLICK HERE TO DOWNLOAD REDIRECTION AND AUTHENTICATION CONTROL PROGRAM
YOUR ACCOUNT NUMBER IS: ffff-ffff-f5ff-ffff
cha!
412
410

FIG. 16

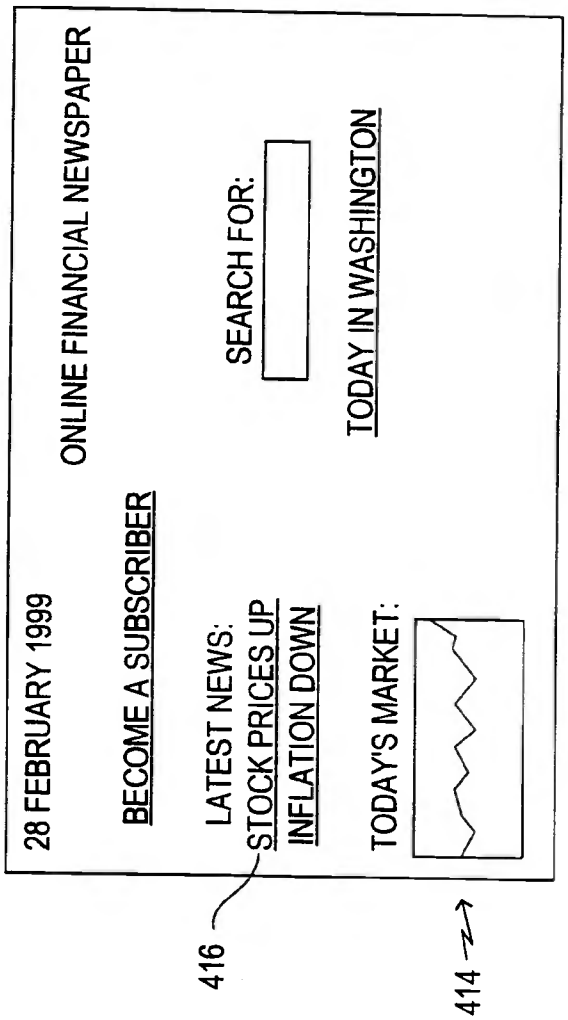

28 FEBRUARY 1999
ONLINE FINANCIAL NEWSPAPER
BECOME A SUBSCRIBER
LATEST NEWS:
STOCK PRICES UP
INFLATION DOWN
SEARCH FOR:
TODAY IN WASHINGTON
TODAY'S MARKET:
416
414

FIG. 17

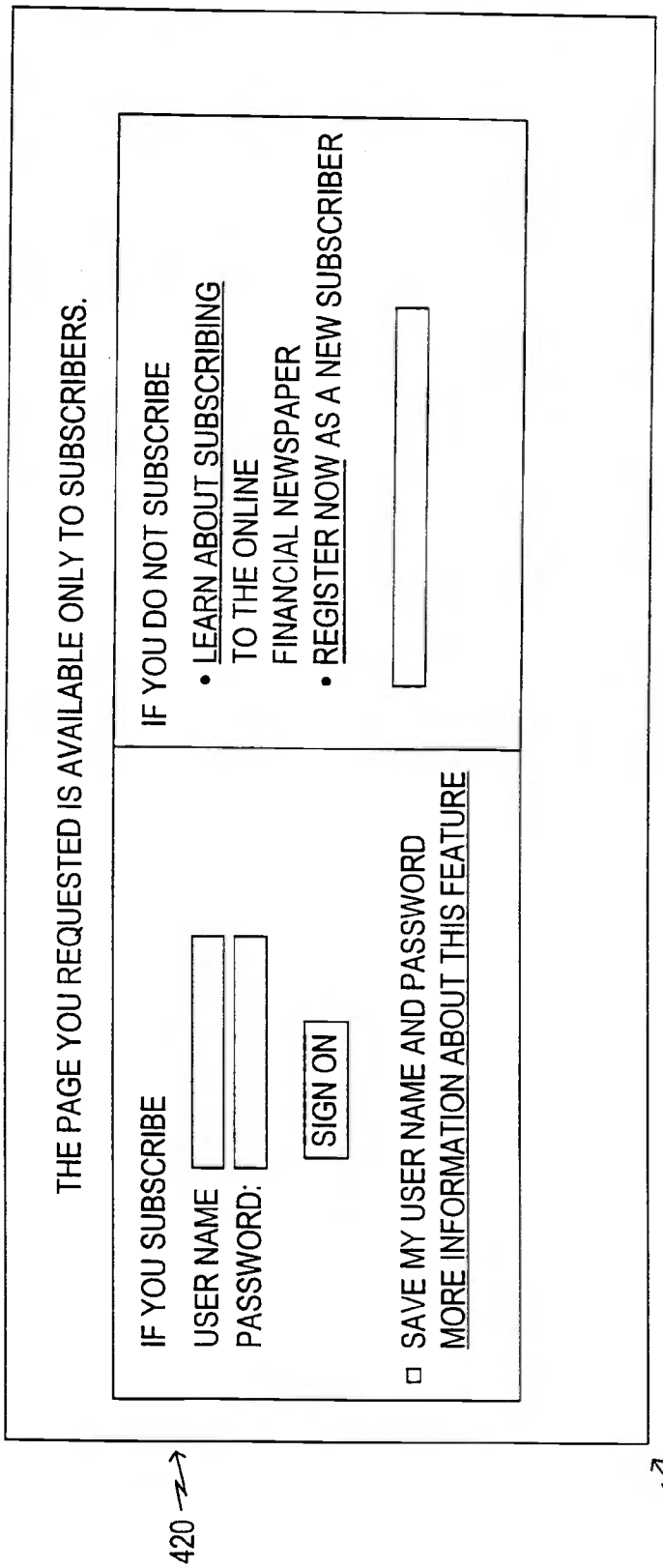

THE PAGE YOU REQUESTED IS AVAILABLE ONLY TO SUBSCRIBERS.
IF YOU SUBSCRIBE
USER NAME
PASSWORD:
SIGN ON
SAVE MY USER NAME AND PASSWORD
MORE INFORMATION ABOUT THIS FEATURE
IF YOU DO NOT SUBSCRIBE
• LEARN ABOUT SUBSCRIBING TO THE ONLINE FINANCIAL NEWSPAPER
• REGISTER NOW AS A NEW SUBSCRIBER
420
418

FIG. 18

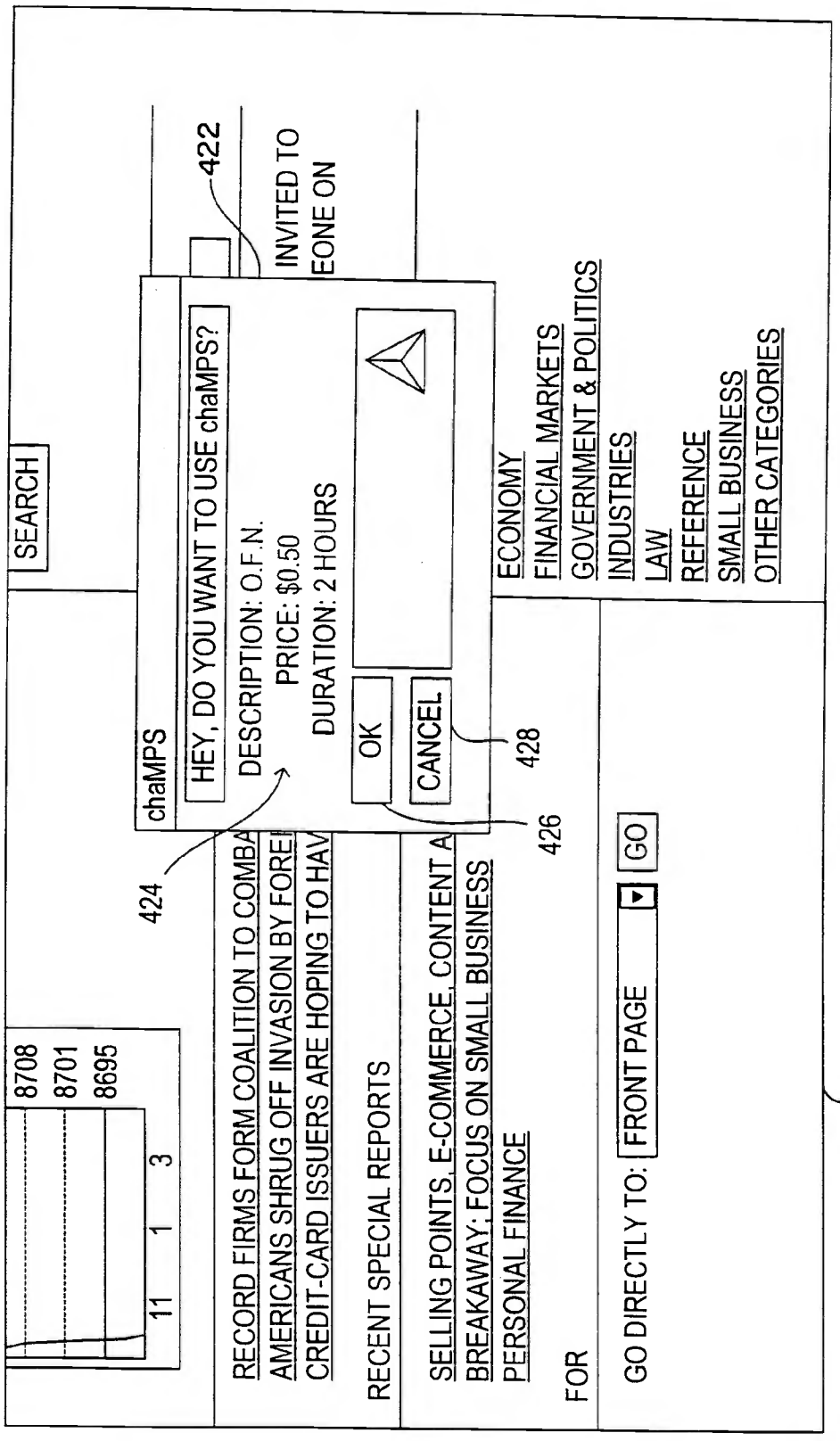

SEARCH
8708
8701
8695
11
1
3
chaMPS
HEY, DO YOU WANT TO USE chaMPS?
DESCRIPTION: O.F.N.
PRICE: $0.50
DURATION: 2 HOURS
OK
CANCEL
422
INVITED TO
EONE ON
424
426
428
RECORD FIRMS FORM COALITION TO COMBA
AMERICANS SHRUG OFF INVASION BY FOREI
CREDIT-CARD ISSUERS ARE HOPING TO HAV
RECENT SPECIAL REPORTS
SELLING POINTS, E-COMMERCE, CONTENT A
BREAKAWAY; FOCUS ON SMALL BUSINESS
PERSONAL FINANCE
FOR
ECONOMY
FINANCIAL MARKETS
GOVERNMENT & POLITICS
INDUSTRIES
LAW
REFERENCE
SMALL BUSINESS
OTHER CATEGORIES
GO DIRECTLY TO: FRONT PAGE
GO
414

FIG. 19

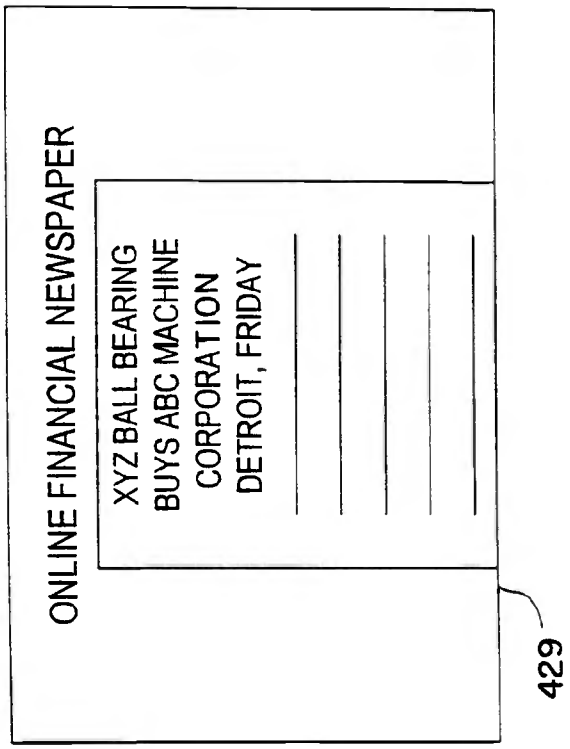

ONLINE FINANCIAL NEWSPAPER
XYZ BALL BEARING
BUYS ABC MACHINE
CORPORATION
DETROIT, FRIDAY
429

FIG. 20

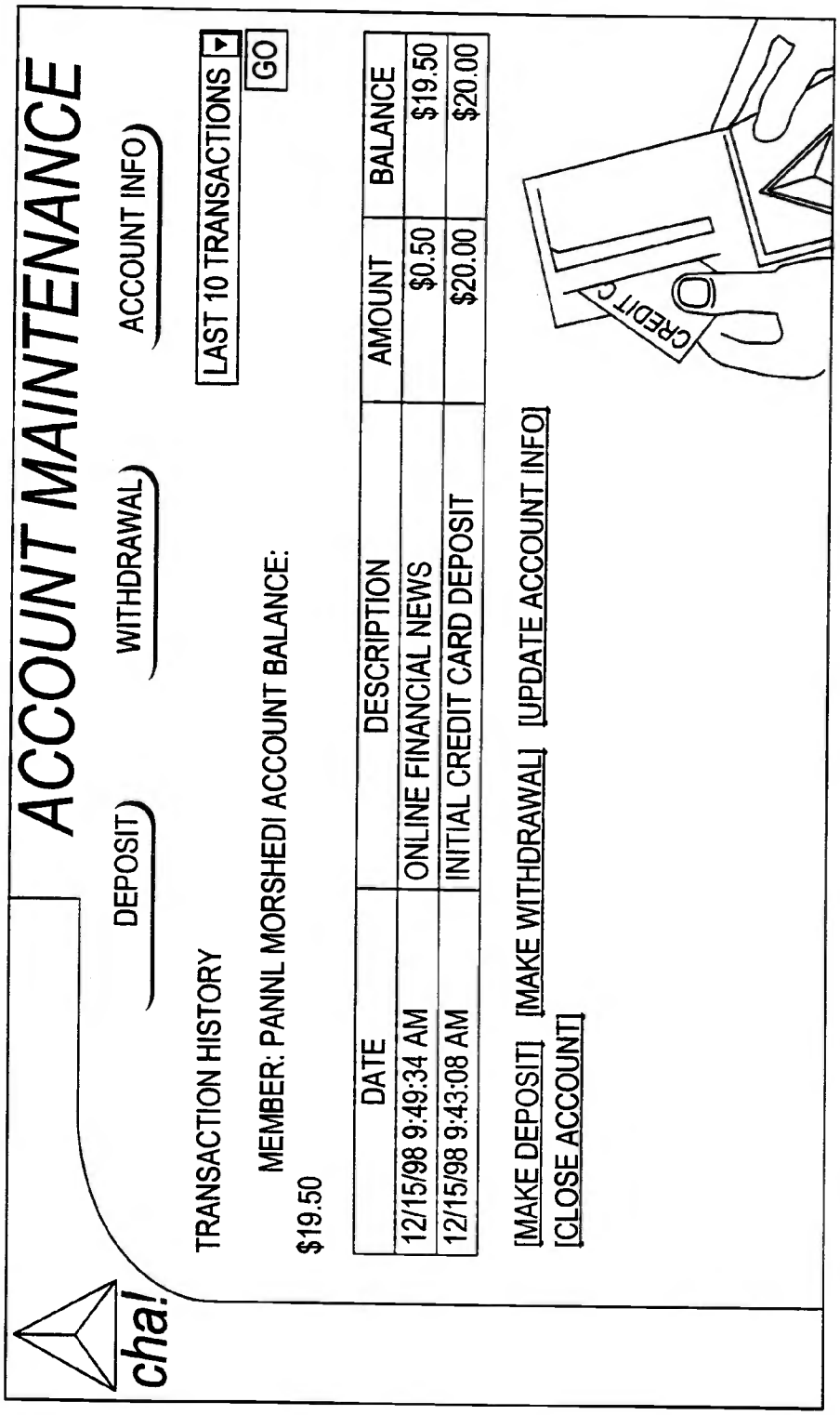

ACCOUNT MAINTENANCE
cha!
DEPOSIT
WITHDRAWAL
ACCOUNT INFO
TRANSACTION HISTORY
LAST 10 TRANSACTIONS
GO
MEMBER: PANNL MORSHEDI ACCOUNT BALANCE:
$19.50
DATE
DESCRIPTION
AMOUNT
BALANCE
12/15/98 9:49:34 AM
ONLINE FINANCIAL NEWS
$0.50
$19.50
12/15/98 9:43:08 AM
INITIAL CREDIT CARD DEPOSIT
$20.00
$20.00
[MAKE DEPOSIT] [MAKE WITHDRAWAL] [UPDATE ACCOUNT INFO]
[CLOSE ACCOUNT]
CREDIT
430

FIG. 21

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/05368

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : G06F 15300
US CL : 705/26

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 705/26, 27

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,870,473 A (BOESCH ET AL.) 09 FEBRUARY 1999, COL 61, LINE 55 TO COL 64, LINE 43.	1-16
A	US 5,864,667 A (BARKAN) 26 JANUARY 1999, COLUMN 3, LINE 2 TO COLUMN 4, LINE 52.	1-16
A,P	US 5,848,400 A (CHANG) 08 DECEMBER 1998, COLUMN 3, LINE 1 TO COLUMN 7, LINE 17.	1-16
A,P	US 5,815,657 A (WILLIAMS ET AL.) 29 SEPTEMBER 1998, COLUMN 12, LINE 11 TO COLUMN 18, LINE 30.	1-16

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 28 JUNE 1999	Date of mailing of the international search report 02 AUG 1999
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer FRANTZY POINVIL Telephone No. (703) 305-9779

Form PCT/ISA/210 (second sheet)(July 1992) ★

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 March 2001 (29.03.2001)

PCT

(10) International Publication Number
WO 01/22319 A1

(51) International Patent Classification[7]: G06F 17/60, G07F 7/10

(21) International Application Number: PCT/US00/25754

(22) International Filing Date:
20 September 2000 (20.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/154,734 20 September 1999 (20.09.1999) US
60/200,396 27 April 2000 (27.04.2000) US

(71) Applicant: ETHENTICA, INC. [US/US]; 100 North Pointe Drive, Lake Forest. CA 92630 (US).

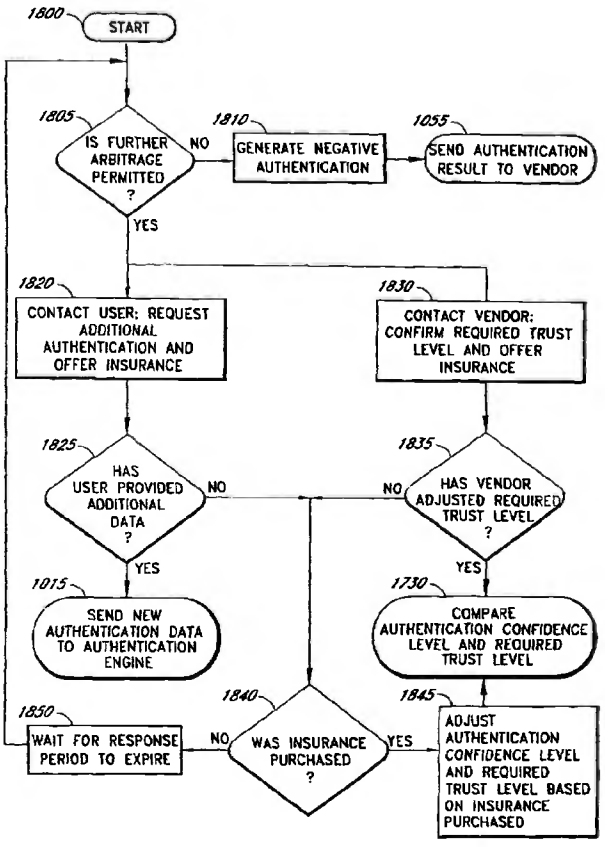
(72) Inventors: DICKINSON, Alexander, G.; 466 Monterey Drive. Laguna Beach, CA 92651 (US). OHARE, Mark, S.; 22 Mountain Laurel, Dove Canyon, CA 92679 (US). ROHRBACH, Mark, D.; 14685 Deerwood Street, Poway, CA 92064 (US). BERGER, Brian; 23518 Sandstone, Mission Viejo, CA 92692 (US). DOBSON, Robert, T., Jr.; 15 Inverary, Dove Canyon, CA 92679 (US).

(74) Agent: ALTMAN, Daniel, E.; Knobbe, Martens, Olson and Bear, LLP, 16th Floor, 620 Newport Center Drive, Newport Beach, CA 92660 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AT (utility model), AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, CZ (utility model), DE, DE (utility model), DK, DK (utility model), DM, DZ, EE, EE (utility model), ES, FI, FI (utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KR (utility model), KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (utility model), SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

[Continued on next page]

(54) Title: TRUST ARBITRAGE IN CRYPTOGRAPHIC AUTHENTICATION



(57) Abstract: A method of allocating the risk associated with a false authentication during an electronic transaction is provided when a confidence gap exists between an authentication confidence level and a desired confidence level. The confidence gap represents the difference between the level of security required for the electronic transaction and that which is currently available. Insurance is offered to at least one of the parties to the transaction. The insurance may be used to close the confidence gap by either raising the authentication confidence level or lowering the desired confidence level. The insurance seller bears the increased risk of a false authentication.

WO 01/22319 A1



(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— *With international search report.*

— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

TRUST ARBITRAGE IN CRYPTOGRAPHIC AUTHENTICATION

Background of the Invention

Field of the Invention

The present invention relates to cryptographic authentication. More specifically, the present invention relates to the shifting and balancing of risk during an authentication event.

Description of the Related Art

With the continued increase in commercial and other transactions taking place across linked computer systems, it has become desirable to secure these transactions and the information related to these transactions. One form of security is to prevent access to systems which perform certain functions, for instance by requiring a password or PIN number in order to use an ATM. Another form of security is to protect data from being intercepted and used by those other than the intended recipients, for instance, when sending a credit card number electronically. Another form of security involves allowing someone to undeniably sign a document or otherwise assent to a transaction electronically.

All of these functions are related to the concept of authentication, or proof of identity. Authentication of electronic systems, particularly digital systems, is generally carried out using cryptographic techniques and protocols. Cryptography is the scrambling of information in such a specific way that it can only be unscrambled by someone who holds the appropriate unscrambling key. By exchanging messages which can only be decrypted by those with access to the proper key, cryptographic protocols can be used as a means to authenticate individuals.

Messages may be sent from the user to the authenticating authority, secured by cryptographic protocols to prevent tampering, and these messages can include user-specific data which is then compared to other user-specific data which has been previously registered with the authenticating authority. The messages sent generally correspond to data which represents some authenticating aspect of the user, such as a password, fingerprint, or data read from a smart card.

However, in some cases, the level of reliability in the authentication of a user provided may not be sufficient for a successful authentication. For instance, a user may be at a public terminal or away from their desk and therefore unable to access a smart card reader, fingerprint reader, or other identifying matter. The authentication may be sufficient for some purposes, but not others. In such cases, it may still be desirable to allow the transaction to occur.

Therefore, there is a continued need for means of allocating the risk of a false authentication in circumstances where at least one party will vouch for the identity of an user.

Summary of the Invention

Based on the foregoing, a need exists for a means to distribute the risk associated with a false authentication to the party most willing and able to bear such risk during the authentication event. Accordingly, one aspect of the invention is to provide a secure server, or trust engine, having server-centric keys, or in other words, storing cryptographic keys and user authentication data on a server. According to this embodiment, a user accesses the trust engine in order to perform authentication and cryptographic functions, such as, for example, authentication, authorization, digital signing and generation, storage, and retrieval of certificates, encryption, notary-like and power-of-attorney-like actions, and the like.

Another aspect of the invention is to provide a reliable, or trusted, authentication process. Moreover, subsequent to a trustworthy positive authentication, a wide number of differing actions may be taken, from providing cryptographic technology, to system or device authorization and access, to permitting use or control of a wide number of electronic devices.

Another aspect of the invention is to provide cryptographic keys and authentication data in an environment where they are not lost, stolen, or compromised, thereby advantageously avoiding a need to continually reissue and manage new keys and authentication data. According to another aspect of the invention, the trust engine allows a user to use one key pair for multiple activities, vendors, and/or authentication requests. According to yet another aspect of the invention, the trust engine performs the majority of cryptographic processing, such as encrypting, authenticating, or signing, on the server side, thereby allowing clients to possess only minimal computing resources.

According to yet another aspect of the invention, the trust engine includes multiple depositories for storing portions of each cryptographic key and authentication data. The portions are created through a data splitting process that prohibits reconstruction without a predetermined portion from more than one depository. According to another embodiment, the multiple depositories are geographically remote such that a rogue employee or otherwise compromised system at one depository will not provide access to a user's key or authentication data.

According to yet another embodiment, the authentication process advantageously allows the trust engine to process vendor and user authentication activities in parallel. According to yet another embodiment, the trust engine may advantageously track failed access attempts and thereby limit the number of times malicious intruders may attempt to subvert the system.

According to yet another embodiment, the trust engine may include multiple instantiations where each trust engine may predict and share processing loads with the others. According to yet another embodiment, the trust engine may include a redundancy module for polling a plurality of authentication results to ensure that more than one system authenticates the user.

Therefore, one aspect of the invention provides a method of allocating the risk during an electronic transaction when a confidence gap exists between an authentication confidence level and a desired confidence level. The method comprises that determining that an authentication confidence level fails to meet a desired confidence level and offering insurance to at least one party of an electronic transaction.

An additional aspect of the invention provides a method of facilitating an electronic transaction when a confidence gap exists between an authentication confidence level and a desired confidence level. Authentication data associated with an electronic transaction is received. The vendor specifies a desired confidence level and a user submits authentication data to produce an authentication confidence level. If the authentication confidence level fails to meet the desired confidence level, the vendor is inviting to reduce the desired confidence level if appropriate. If the vendor reduces the requirement the use of the user's private key is allowed.

Another aspect of the invention provides a method of facilitating an electronic transaction when a confidence gap exists between an authentication confidence level and a desired confidence level. Authentication data associated with an electronic transaction is received from a vendor who specifies a desired confidence level. A user submits authentication data to produce an authentication confidence level for the transaction. If the authentication confidence level fails to meet the desired confidence level, the user is invited to provide additional authentication data. Any additional data provided is used to improve the authentication confidence level. If the additional authentication data increases the authentication confidence level to meet the desired confidence level, use of the user's private key is allowed.

In an additional aspect of the invention, an apparatus for authenticating an electronic transaction between a first party and a second party is provided. The apparatus comprises a trust engine which receives authentication data from a first party to the electronic transaction and a desired confidence level from a second party to the electronic transaction. The trust engine determines an authentication confidence level based on the authentication data. If the desired confidence level is higher than the authentication confidence level, the trust engine invites the second party to reduce the desired confidence level for authenticating the electronic transaction.

Yet another additional aspect of the invention provides a system for facilitating an electronic transaction between a merchant and a user. The system comprises a trust engine which receives authentication data from a user and a desired confidence level from a merchant. The trust engine determines an authentication confidence level based on the authentication data, and invites the user to provide additional authentication data if the desired confidence level is higher than the authentication confidence level. If authentication confidence level is raised above the desired confidence level for the transaction, the trust engine uses a private cryptographic key of the user to cryptographically sign a record of the transaction.

In yet another aspect of the invention, a method of facilitating an electronic transaction when a confidence gap exists between an authentication confidence level and a desired confidence level is provided. It is first determined that an authentication confidence level fails to meet a desired confidence level for a transaction involving a first entity that specifies the desired confidence level and a second entity that submits authentication data to produce an authentication confidence level. At this point, the first entity is invited to reduce the desired confidence level in order to allow a successful authentication.

Brief Description of the Drawings

The present invention is described in more detail below in connection with the attached drawings, which are meant to illustrate and not to limit the invention, and in which:

FIGURE 1 illustrates a block diagram of a cryptographic system, according to aspects of an embodiment of the invention;

FIGURE 2 illustrates a block diagram of the trust engine of FIGURE 1, according to aspects of an embodiment of the invention;

FIGURE 3 illustrates a block diagram of the transaction engine of FIGURE 2, according to aspects of an embodiment of the invention;

FIGURE 4 illustrates a block diagram of the depository of FIGURE 2, according to aspects of an embodiment of the invention;

FIGURE 5 illustrates a block diagram of the authentication engine of FIGURE 2, according to aspects of an embodiment of the invention;

FIGURE 6 illustrates a block diagram of the cryptographic engine of FIGURE 2, according to aspects of an embodiment of the invention;

FIGURE 7 illustrates a block diagram of a depository system, according to aspects of another embodiment of the invention;

FIGURE 8 illustrates a flow chart of a data splitting process according to aspects of an embodiment of the invention;

FIGURE 9A illustrates a data flow of an enrollment process according to aspects of an embodiment of the invention;

FIGURE 9B illustrates a flow chart of an interoperability process according to aspects of an embodiment of the invention;

FIGURE 10 illustrates a data flow of an authentication process according to aspects of an embodiment of the invention;

FIGURE 11 illustrates a data flow of a signing process according to aspects of an embodiment of the invention.

FIGURE 12 illustrates a data flow and an encryption/decryption process according to aspects and yet another embodiment of the invention;

FIGURE 13 illustrates a simplified block diagram of a trust engine system according to aspects of another embodiment of the invention;

FIGURE 14 illustrates a simplified block diagram of a trust engine system according to aspects of another embodiment of the invention;

FIGURE 15 illustrates a block diagram of the redundancy module of FIGURE 14, according to aspects of an embodiment of the invention;

FIGURE 16 illustrates a process for evaluating authentications according to one aspect of the invention;

FIGURE 17 illustrates a process for assigning a value to an authentication according to one aspect as shown in FIGURE 16 of the invention;

FIGURE 18 illustrates a process for performing trust arbitrage in an aspect of the invention as shown in FIGURE 17; and

FIGURE 19 illustrates a sample transaction between a user and a vendor according to aspects of an embodiment of the invention where an initial web based contact leads to a sales contract signed by both parties.

FIGURE 20 illustrates a sample user system with a cryptographic service provider module which provides security functions to a user system.

Detailed Description of the Preferred Embodiment

One aspect of the present invention is to provide a cryptographic system where one or more secure servers, or a trust engine, stores cryptographic keys and user authentication data. Users access the functionality of conventional cryptographic systems through network access to the trust engine, however, the trust engine does not release actual keys and other authentication data and therefore, the keys and data remain secure. This server-centric storage of keys and authentication data provides for user-independent security, portability, availability, and straightforwardness.

Because users can be confident in, or trust, the cryptographic system to perform user and document authentication and other cryptographic functions, a wide variety of functionality may be incorporated into the system. For example, the trust engine provider can ensure against agreement repudiation by, for example, authenticating the agreement participants, digitally signing the agreement on behalf of or for the participants, and storing a record of the agreement digitally signed by each participant. In addition, the cryptographic system may monitor agreements and determine to apply varying degrees of authentication, based on, for example, price, user, vendor, geographic location, place of use, or the like.

To facilitate a complete understanding of the invention, the remainder of the detailed description describes the invention with reference to the figures, wherein like elements are referenced with like numerals throughout.

FIGURE 1 illustrates a block diagram of a cryptographic system 100, according to aspects of an embodiment of the invention. As shown in FIGURE 1, the cryptographic system 100 includes a user system 105, a trust engine 110, a certificate authority 115, and a vendor system 120, communicating through a communication link 125.

According to one embodiment of the invention, the user system 105 comprises a conventional general-purpose computer having one or more microprocessors, such as, for example, an Intel-based processor. Moreover, the user system 105 includes an appropriate operating system, such as, for example, an operating system capable of including graphics or windows, such as Windows, Unix, Linux, or the like. As shown in FIGURE 1, the user system 105 may include a biometric device 107. The biometric device 107 may advantageously capture a user's biometric and transfer the captured biometric to the trust engine 110. According to one embodiment of the invention, the biometric device may advantageously comprise a device having attributes and features similar to those disclosed in U.S. Patent

Application No. 08/926,277, filed on September 5, 1997, entitled "RELIEF OBJECT IMAGE GENERATOR," U.S. Patent Application No. 09/558,634, filed on April 26, 2000, entitled "IMAGING DEVICE FOR A RELIEF OBJECT AND SYSTEM AND METHOD OF USING THE IMAGE DEVICE," U.S. Patent Application No. 09/435,011, filed on November 5, 1999, entitled "RELIEF OBJECT SENSOR ADAPTOR," and U.S. Patent Application No. 09/477,943, filed on January 5, 2000, entitled "PLANAR OPTICAL IMAGE SENSOR AND SYSTEM FOR GENERATING AN ELECTRONIC IMAGE OF A RELIEF OBJECT FOR FINGERPRINT READING," all of which are owned by the instant assignee, and all of which are hereby incorporated by reference herein.

In addition, the user system 105 may connect to the communication link 125 through a conventional service provider, such as, for example, a dial up, digital subscriber line (DSL), cable modem, fiber connection, or the like. According to another embodiment, the user system 105 connects the communication link 125 through network connectivity such as, for example, a local or wide area network. According to one embodiment, the operating system includes a TCP/IP stack that handles all incoming and outgoing message traffic passed over the communication link 125.

Although the user system 105 is disclosed with reference to the foregoing embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives embodiments of the user system 105, including almost any computing device capable of sending or receiving information from another computer system. For example, the user system 105 may include a computer workstation, an interactive television, an interactive kiosk, a personal mobile computing device, such as a digital assistant, mobile phone, laptop, or the like, a wireless communications device, a smartcard, an embedded computing device, or the like, which can interact with the communication link 125. In such alternative systems, the operating systems will likely differ and be adapted for the particular device. However, according to one embodiment, the operating systems advantageously continue to provide the appropriate communications protocols needed to establish communication with the communication link 125.

FIGURE 1 illustrates the trust engine 110. According to one embodiment, the trust engine 110 comprises one or more secure servers for accessing and storing sensitive information, such as user authentication data and public and private cryptographic keys. According to one embodiment, the authentication data includes data designed to uniquely identify a user of the cryptographic system 100. For example, the authentication data may include a user identification number, one or more biometrics, and a series of questions and answers generated by the trust engine 110 or the user, but answered initially by the user at enrollment. The foregoing questions may include demographic data, such as place of birth, address, anniversary, or the like, personal data, such as mother's maiden name, favorite ice cream, or the like, or other data designed to uniquely identify the user. The trust engine 110 compares a user's authentication data associated with a current transaction, to the authentication data provided at an earlier time, such as, for example, during enrollment. The trust engine 110 may advantageously require the user to produce the authentication data at the time of each transaction, or, the trust engine 110 may advantageously allow the user to

periodically produce authentication data, such as at the beginning of a string of transactions or the logging onto a particular vendor website.

According to the embodiment where the user produces biometric data, the user provides a physical characteristic, such as a fingerprint or speech, to the biometric device 107. The biometric device advantageously produces an electronic pattern, or biometric, of the physical characteristic. The electronic pattern is transferred through the user system 105 to the trust engine 110 for either enrollment or authentication purposes.

Once the user produces the appropriate authentication data and the trust engine 110 determines a positive match between that authentication data (current authentication data) and the authentication data provided at the time of enrollment (enrollment authentication data), the trust engine 110 provides the user with complete cryptographic functionality. For example, the properly authenticated user may advantageously employ the trust engine 110 to perform hashing, digitally signing, encrypting and decrypting (often referred to only as encrypting), creating or distributing digital certificates, and the like. However, the private cryptographic keys used in the cryptographic functions will not be available outside the trust engine 110, thereby ensuring the integrity of the cryptographic keys.

According to one embodiment, the trust engine 110 generates and stores cryptographic keys. According to another embodiment, at least one cryptographic key is associated with each user. Moreover, when the cryptographic keys include public-key technology, each private key associated with a user is generated within, and not released from, the trust engine 110. Thus, so long as the user has access to the trust engine 110, the user may perform cryptographic functions using his or her private or public key. Such remote access advantageously allows users to remain completely mobile and access cryptographic functionality through practically any Internet connection, such as cellular and satellite phones, kiosks, laptops, hotel rooms and the like.

According to another embodiment, the trust engine 110 performs the cryptographic functionality using a key pair generated for the trust engine 110. According to this embodiment, the trust engine 110 first authenticates the user, and after the user has properly produced authentication data matching the enrollment authentication data, the trust engine 110 uses its own cryptographic key pair to perform cryptographic functions on behalf of the authenticated user.

A skilled artisan will recognize from the disclosure herein that the cryptographic keys may advantageously include some or all of symmetric keys, public keys, and private keys. In addition, a skilled artisan will recognize from the disclosure herein that the foregoing keys may be implemented with a wide number of algorithms available from commercial technologies, such as, for example, RSA, ELGAMAL, or the like.

FIGURE 1 also illustrates the certificate authority 115. According to one embodiment, the certificate authority 115 may advantageously comprise a trusted third-party organization or company that issues digital certificates, such as, for example, VeriSign, Baltimore, Entrust, or the like. The trust engine 110 may advantageously transmit requests for digital certificates, through one or more conventional digital certificate protocols, such as, for example, PKCS10, to the certificate authority 115. In response, the certificate authority 115 will issue a digital certificate in one or more of a number of differing protocols, such as, for example, PKCS7. According to one

embodiment of the invention, the trust engine 110 requests digital certificates from several or all of the prominent certificate authorities 115 such that the trust engine 110 has access to a digital certificate corresponding to the certificate standard of any requesting party.

According to another embodiment, the trust engine 110 internally performs certificate issuances. In this embodiment, the trust engine 110 may access a certificate system for generating certificates and/or may internally generate certificates when they are requested, such as, for example, at the time of key generation or in the certificate standard requested at the time of the request. The trust engine 110 will be disclosed in greater detail below.

FIGURE 1 also illustrates the vendor system 120. According to one embodiment, the vendor system 120 advantageously comprises a Web server. Typical Web servers generally serve content over the Internet using one of several internet markup languages or document format standards, such as the Hyper-Text Markup Language (HTML) or the Extensible Markup Language (XML). The Web server accepts requests from browsers like Netscape and Internet Explorer and then returns the appropriate electronic documents. A number of server or client-side technologies can be used to increase the power of the Web server beyond its ability to deliver standard electronic documents. For example, these technologies include Common Gateway Interface (CGI) scripts, Secure Sockets Layer (SSL) security, and Active Server Pages (ASPs). The vendor system 120 may advantageously provide electronic content relating to commercial, personal, educational, or other transactions.

Although the vendor system 120 is disclosed with reference to the foregoing embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein that the vendor system 120 may advantageously comprise any of the devices described with reference to the user system 105 or combination thereof.

FIGURE 1 also illustrates the communication link 125 connecting the user system 105, the trust engine 110, the certificate authority 115, and the vendor system 120. According to one embodiment, the communication link 125 preferably comprises the Internet. The Internet, as used throughout this disclosure is a global network of computers. The structure of the Internet, which is well known to those of ordinary skill in the art, includes a network backbone with networks branching from the backbone. These branches, in turn, have networks branching from them, and so on. Routers move information packets between network levels, and then from network to network, until the packet reaches the neighborhood of its destination. From the destination, the destination network's host directs the information packet to the appropriate terminal, or node. In one advantageous embodiment, the Internet routing hubs comprise domain name system (DNS) servers using Transmission Control Protocol/Internet Protocol (TCP/IP) as is well known in the art. The routing hubs connect to one or more other routing hubs via high-speed communication links.

One popular part of the Internet is the World Wide Web. The World Wide Web contains different computers, which store documents capable of displaying graphical and textual information. The computers that provide information on the World Wide Web are typically called "websites." A website is defined by an Internet address that has an associated electronic page. The electronic page can be identified by a Uniform Resource Locator (URL).

Generally, an electronic page is a document that organizes the presentation of text, graphical images, audio, video, and so forth.

Although the communication link 125 is disclosed in terms of its preferred embodiment, one of ordinary skill in the art will recognize from the disclosure herein that the communication link 125 may include a wide range of interactive communications links. For example, the communication link 125 may include interactive television networks, telephone networks, wireless data transmission systems, two-way cable systems, customized private or public computer networks, interactive kiosk networks, automatic teller machine networks, direct links, satellite or cellular networks, and the like.

FIGURE 2 illustrates a block diagram of the trust engine 110 of FIGURE 1 according to aspects of an embodiment of the invention. As shown in FIGURE 2, the trust engine 110 includes a transaction engine 205, a depository 210, an authentication engine 215, and a cryptographic engine 220. According to one embodiment of the invention, the trust engine 110 also includes mass storage 225. As further shown in FIGURE 2, the transaction engine 205 communicates with the depository 210, the authentication engine 215, and the cryptographic engine 220, along with the mass storage 225. In addition, the depository 210 communicates with the authentication engine 215, the cryptographic engine 220, and the mass storage 225. Moreover, the authentication engine 215 communicates with the cryptographic engine 220. According to one embodiment of the invention, some or all of the foregoing communications may advantageously comprise the transmission of XML documents to IP addresses that correspond to the receiving device. As mentioned in the foregoing, XML documents advantageously allow designers to create their own customized document tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations. Moreover, some or all of the foregoing communications may include conventional SSL technologies.

According to one embodiment, the transaction engine 205 comprises a data routing device, such as a conventional Web server available from Netscape, Microsoft, Apache, or the like. For example, the Web server may advantageously receive incoming data from the communication link 125. According to one embodiment of the invention, the incoming data is addressed to a front-end security system for the trust engine 110. For example, the front-end security system may advantageously include a firewall, an intrusion detection system searching for known attack profiles, and/or a virus scanner. After clearing the front-end security system, the data is received by the transaction engine 205 and routed to one of the depository 210, the authentication engine 215, the cryptographic engine 220, and the mass storage 225. In addition, the transaction engine 205 monitors incoming data from the authentication engine 215 and cryptographic engine 220, and routes the data to particular systems through the communication link 125. For example, the transaction engine 205 may advantageously route data to the user system 105, the certificate authority 115, or the vendor system 120.

According to one embodiment, the data is routed using conventional HTTP routing techniques, such as, for example, employing URLs or Uniform Resource Indicators (URIs). URIs are similar to URLs, however, URIs typically indicate the source of files or actions, such as, for example, executables, scripts, and the like. Therefore, according to

the one embodiment, the user system 105, the certificate authority 115, the vendor system 120, and the components of the trust engine 210, advantageously include sufficient data within communication URLs or URIs for the transaction engine 205 to properly route data throughout the cryptographic system.

Although the data routing is disclosed with reference to its preferred embodiment, a skilled artisan will recognize a wide number of possible data routing solutions or strategies. For example, XML or other data packets may advantageously be unpacked and recognized by their format, content, or the like, such that the transaction engine 205 may properly route data throughout the trust engine 110. Moreover, a skilled artisan will recognize that the data routing may advantageously be adapted to the data transfer protocols conforming to particular network systems, such as, for example, when the communication link 125 comprises a local network.

According to yet another embodiment of the invention, the transaction engine 205 includes conventional SSL encryption technologies, such that the foregoing systems may authenticate themselves, and vise-versa, with transaction engine 205, during particular communications. As will be used throughout this disclosure, the term "½ SSL" refers to communications where a server but not necessarily the client, is SSL authenticated, and the term "FULL SSL" refers to communications where the client and the server are SSL authenticated. When the instant disclosure uses the term "SSL", the communication may comprise ½ or FULL SSL.

As the transaction engine 205 routes data to the various components of the cryptographic system 100, the transaction engine 205 may advantageously create an audit trail. According to one embodiment, the audit trail includes a record of at least the type and format of data routed by the transaction engine 205 throughout the cryptographic system 100. Such audit data may advantageously be stored in the mass storage 225.

FIGURE 2 also illustrates the depository 210. According to one embodiment, the depository 210 comprises one or more data storage facilities, such as, for example, a directory server, a database server, or the like. As shown in FIGURE 2, the depository 210 stores cryptographic keys and enrollment authentication data. The cryptographic keys may advantageously correspond to the trust engine 110 or to users of the cryptographic system 100, such as the user or vendor. The enrollment authentication data may advantageously include data designed to uniquely identify a user, such as, user ID, passwords, answers to questions, biometric data, or the like. This enrollment authentication data may advantageously be acquired at enrollment of a user or another alternative later time. For example, the trust engine 110 may include periodic or other renewal or reissue of enrollment authentication data.

According to one embodiment, the communication from the transaction engine 205 to and from the authentication engine 215 and the cryptographic engine 220 comprises secure communication, such as, for example conventional SSL technology. In addition, as mentioned in the foregoing, the data of the communications to and from the depository 210 may be transferred using URLs, URIs, HTTP or XML documents, with any of the foregoing advantageously having data requests and formats embedded therein.

As mentioned above, the depository 210 may advantageously comprises a plurality of secure data storage facilities. In such an embodiment, the secure data storage facilities may be configured such that a compromise of the security in one individual data storage facility will not compromise the cryptographic keys or the authentication data

stored therein. For example, according to this embodiment, the cryptographic keys and the authentication data are mathematically operated on so as to statistically and substantially randomize the data stored in each data storage facility. According to one embodiment, the randomization of the data of an individual data storage facility renders that data undecipherable. Thus, compromise of an individual data storage facility produces only a randomized undecipherable number and does not compromise the security of any cryptographic keys or the authentication data as a whole.

FIGURE 2 also illustrates the trust engine 110 including the authentication engine 215. According to one embodiment, the authentication engine 215 comprises a data comparator configured to compare data from the transaction engine 205 with data from the depository 210. For example, during authentication, a user supplies current authentication data to the trust engine 110 such that the transaction engine 205 receives the current authentication data. As mentioned in the foregoing, the transaction engine 205 recognizes the data requests, preferably in the URL or URI, and routes the authentication data to the authentication engine 215. Moreover, upon request, the depository 210 forwards enrollment authentication data corresponding to the user to the authentication engine 215. Thus, the authentication engine 215 has both the current authentication data and the enrollment authentication data for comparison.

According to one embodiment, the communications to the authentication engine comprise secure communications, such as, for example, SSL technology. Additionally, security can be provided within the trust engine 110 components, such as, for example, super-encryption using public key technologies. For example, according to one embodiment, the user encrypts the current authentication data with the public key of the authentication engine 215. In addition, the depository 210 also encrypts the enrollment authentication data with the public key of the authentication engine 215. In this way, only the authentication engine's private key can be used to decrypt the transmissions.

As shown in FIGURE 2, the trust engine 110 also includes the cryptographic engine 220. According to one embodiment, the cryptographic engine comprises a cryptographic handling module, configured to advantageously provide conventional cryptographic functions, such as, for example, public-key infrastructure (PKI) functionality. For example, the cryptographic engine 220 may advantageously issue public and private keys for users of the cryptographic system 100. In this manner, the cryptographic keys are generated at the cryptographic engine 220 and forwarded to the depository 210 such that at least the private cryptographic keys are not available outside of the trust engine 110. According to another embodiment, the cryptographic engine 220 randomizes and splits at least the private cryptographic key data, thereby storing only the randomized split data. Similar to the splitting of the enrollment authentication data, the splitting process ensures the stored keys are not available outside the cryptographic engine 220. According to another embodiment, the functions of the cryptographic engine can be combined with and performed by the authentication engine 215.

According to one embodiment, communications to and from the cryptographic engine include secure communications, such as SSL technology. In addition, XML documents may advantageously be employed to transfer data and/or make cryptographic function requests.

FIGURE 2 also illustrates the trust engine 110 having the mass storage 225. As mentioned in the foregoing, the transaction engine 205 keeps data corresponding to an audit trail and stores such data in the mass storage 225. Similarly, according to one embodiment of the invention, the depository 210 keeps data corresponding to an audit trail and stores such data in the mass storage device 225. The depository audit trail data is similar to that of the transaction engine 205 in that the audit trail data comprises a record of the requests received by the depository 210 and the response thereof. In addition, the mass storage 225 may be used to store digital certificates having the public key of a user contained therein.

Although the trust engine 110 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize in the disclosure herein, a wide number of alternatives for the trust engine 110. For example, the trust engine 110, may advantageously perform only authentication, or alternatively, only some or all of the cryptographic functions, such as data encryption and decryption. According to such embodiments, one of the authentication engine 215 and the cryptographic engine 220 may advantageously be removed, thereby creating a more straightforward design for the trust engine 110. In addition, the cryptographic engine 220 may also communicate with a certificate authority such that the certificate authority is embodied within the trust engine 110. According to yet another embodiment, the trust engine 110 may advantageously perform authentication and one or more cryptographic functions, such as, for example, digital signing.

FIGURE 3 illustrates a block diagram of the transaction engine 205 of FIGURE 2, according to aspects of an embodiment of the invention. According to this embodiment, the transaction engine 205 comprises an operating system 305 having a handling thread and a listening thread. The operating system 305 may advantageously be similar to those found in conventional high volume servers, such as, for example, Web servers available from Apache. The listening thread monitors the incoming communication from one of the communication link 125, the authentication engine 215, and the cryptographic engine 220 for incoming data flow. The handling thread recognizes particular data structures of the incoming data flow, such as, for example, the foregoing data structures, thereby routing the incoming data to one of the communication link 125, the depository 210, the authentication engine 215, the cryptographic engine 220, or the mass storage 225. As shown in FIGURE 3, the incoming and outgoing data may advantageously be secured through, for example, SSL technology.

FIGURE 4 illustrates a block diagram of the depository 210 of FIGURE 2 according to aspects of an embodiment of the invention. According to this embodiment, the depository 210 comprises one or more lightweight directory access protocol (LDAP) servers. LDAP directory servers are available from a wide variety of manufacturers such as Netscape, ISO, and others. FIGURE 4 also shows that the directory server preferably stores data 405 corresponding to the cryptographic keys and data 410 corresponding to the enrollment authentication data. According to one embodiment, the depository 210 comprises a single logical memory structure indexing authentication data and

cryptographic key data to a unique user ID. The single logical memory structure preferably includes mechanisms to ensure a high degree of trust, or security, in the data stored therein. For example, the physical location of the depository 210 may advantageously include a wide number of conventional security measures, such as limited employee access, modern surveillance systems, and the like. In addition to, or in lieu of, the physical securities, the computer system or server may advantageously include software solutions to protect the stored data. For example, the depository 210 may advantageously create and store data 415 corresponding to an audit trail of actions taken. In addition, the incoming and outgoing communications may advantageously be encrypted with public key encryption coupled with conventional SSL technologies.

According to another embodiment, the depository 210 may comprise distinct and physically separated data storage facilities, as disclosed further with reference to FIGURE 7.

FIGURE 5 illustrates a block diagram of the authentication engine 215 of FIGURE 2 according to aspects of an embodiment of the invention. Similar to the transaction engine 205 of FIGURE 3, the authentication engine 215 comprises an operating system 505 having at least a listening and a handling thread of a modified version of a conventional Web server, such as, for example, Web servers available from Apache. As shown in FIGURE 5, the authentication engine 215 includes access to at least one private key 510. The private key 510 may advantageously be used for example, to decrypt data from the transaction engine 205 or the depository 210, which was encrypted with a corresponding public key of the authentication engine 215.

FIGURE 5 also illustrates the authentication engine 215 comprising a comparator 515, a data splitting module 520, and a data assembling module 525. According to the preferred embodiment of the invention, the comparator 515 includes technology capable of comparing potentially complex patterns related to the foregoing biometric authentication data. The technology may include hardware, software, or combined solutions for pattern comparisons, such as, for example, those representing finger print patterns or voice patterns. In addition, according to one embodiment, the comparator 515 of the authentication engine 215 may advantageously compare conventional hashes of documents in order to render a comparison result. According to one embodiment of the invention, the comparator 515 includes the application of heuristics 530 to the comparison. The heuristics 530 may advantageously address circumstances surrounding an authentication attempt, such as, for example, the time of day, IP address or subnet mask, purchasing profile, email address, processor serial number or ID, or the like.

Moreover, the nature of biometric data comparisons may result in varying degrees of confidence being produced from the matching of current biometric authentication data to enrollment data. For example, unlike a traditional password which may only return a positive or negative match, a fingerprint may be determined to be a partial match, e.g. a 90% match, a 75% match, or a 10% match, rather than simply being correct or incorrect. Other biometric identifiers such as voice print analysis or face recognition may share this property of probabilistic authentication, rather than absolute authentication.

When working with such probabilistic authentication or in other cases where an authentication is considered less than absolutely reliable, it is desirable to apply the heuristics 530 to determine whether the level of confidence in the authentication provided is sufficiently high to authenticate the transaction which is being made.

It will sometimes be the case that the transaction at issue is a relatively low value transaction where it is acceptable to be authenticated to a lower level of confidence. This could include a transaction which has a low dollar value associated with it (e.g., a $10 purchase) or a transaction with low risk (e.g., admission to a members-only web site).

Conversely, for authenticating other transactions, it may be desirable to require a high degree of confidence in the authentication before allowing the transaction to proceed. Such transactions may include transactions of large dollar value (e.g., signing a multi-million dollar supply contract) or transaction with a high risk if an improper authentication occurs (e.g., remotely logging onto a government computer).

The use of the heuristics 530 in combination with confidence levels and transactions values may be used as will be described below to allow the comparator to provide a dynamic context-sensitive authentication system.

According to another embodiment of the invention, the comparator 515 may advantageously track authentication attempts for a particular transaction. For example, when a transaction fails, the trust engine 110 may request the user to re-enter his or her current authentication data. The comparator 515 of the authentication engine 215 may advantageously employ an attempt limiter 535 to limit the number of authentication attempts, thereby prohibiting brute-force attempts to impersonate a user's authentication data. According to one embodiment, the attempt limiter 535 comprises a software module monitoring transactions for repeating authentication attempts and, for example, limiting the authentication attempts for a given transaction to three. Thus, the attempt limiter 535 will limit an automated attempt to impersonate an individual's authentication data to, for example, simply three "guesses." Upon three failures, the attempt limiter 535 may advantageously deny additional authentication attempts. Such denial may advantageously be implemented through, for example, the comparator 515 returning a negative result regardless of the current authentication data being transmitted. On the other hand, the transaction engine 205 may advantageously block any additional authentication attempts pertaining to a transaction in which three attempts have previously failed.

The authentication engine 215 also includes the data splitting module 520 and the data assembling module 525. The data splitting module 520 advantageously comprises a software, hardware, or combination module having the ability to mathematically operate on various data so as to substantially randomize and split the data into portions. According to one embodiment, original data is not recreatable from an individual portion. The data assembling module 525 advantageously comprises a software, hardware, or combination module configured to mathematically operate on the foregoing substantially randomized portions, such that the combination thereof provides the original deciphered data. According to one embodiment, the authentication engine 215 employs the data splitting module 520 to randomize and split enrollment authentication data into portions, and employs the data assembling module 525 to reassemble the portions into usable enrollment authentication data.

FIGURE 6 illustrates a block diagram of the cryptographic engine 220 of the trust engine 200 of FIGURE 2 according to aspects of one embodiment of the invention. Similar to the transaction engine 205 of FIGURE 3, the cryptographic engine 220 comprises an operating system 605 having at least a listening and a handling thread of a modified version of a conventional Web server, such as, for example, Web servers available from Apache. As shown in FIGURE 6, the cryptographic engine 220 comprises a data splitting module 610 and a data assembling module 620 that function similar to those of FIGURE 5. However, according to one embodiment, the data splitting module 610 and the data assembling module 620 process cryptographic key data, as opposed to the foregoing enrollment authentication data. Although, a skilled artisan will recognize from the disclosure herein that the data splitting module 910 and the data splitting module 620 may be combined with those of the authentication engine 215.

The cryptographic engine 220 also comprises a cryptographic handling module 625 configured to perform some or all of a wide number of cryptographic functions. According to one embodiment, the cryptographic handling module 625 may comprise software modules or programs, hardware, or both. According to another embodiment, the cryptographic handling module 625 may perform data comparisons, data hashing, data encryption or decryption, digital signature verification or creation, digital certificate generation, storage, or requests, cryptographic key generation, or the like. Moreover, a skilled artisan will recognize from the disclosure herein that the cryptographic handling module 625 may advantageously comprises a public-key infrastructure, such as Pretty Good Privacy (PGP), an RSA-based public-key system, or a wide number of alternative key management systems. In addition, the cryptographic handling module 625 may perform public-key encryption, symmetric-key encryption, or both. In addition to the foregoing, the cryptographic handling module 625 may include one or more computer programs or modules, hardware, or both, for implementing seamless, transparent, interoperability functions.

A skilled artisan will also recognize from the disclosure herein that the cryptographic functionality may include a wide number of functions generally surrounding cryptographic key management systems.

FIGURE 7 illustrates a simplified block diagram of a depository system 700 according to aspects of an embodiment of the invention. As shown in FIGURE 7, the depository system 700 advantageously comprises multiple data storage facilities, for example, data storage facilities D1, D2, D3, and D4. According to one embodiment of the invention, each of the data storage facilities D1 through D4 may advantageously comprise some or all of the elements disclosed with reference to the depository 210 of FIGURE 4. Similar to the depository 210, the data storage facilities D1 through D4 communicate with the transaction engine 205, the authentication engine 215, and the cryptographic engine 220, preferably through conventional SSL communication links transferring, for example, XML documents. Communications from the transaction engine 205 may advantageously include requests for data, wherein the request is advantageously broadcast to the IP address of each data storage facility D1 through D4. On the other hand, the transaction engine 205 may broadcast requests to particular data storage facilities based on a wide number of criteria, such as, for example, response time, server loads, maintenance schedules, or the like.

In response to requests for data from the transaction engine 205, the depository system 700 advantageously forwards stored data to the authentication engine 215 and the cryptographic engine 220. The respective data

assembling modules receive the forwarded data and assemble the data into useable formats. On the other hand, communications from the authentication engine 215 and the cryptographic engine 220 to the data storage facilities D1 through D4 may include the transmission of sensitive data to be stored. For example, according to one embodiment, the authentication engine 215 and the cryptographic engine 220 may advantageously employ their respective data splitting modules to divide sensitive data into undecipherable portions, and then transmit one or more undecipherable portions of the sensitive data to a particular data storage facility.

According to one embodiment, each data storage facility, D1 through D4, comprises a separate and independent storage system, such as, for example, a directory server. According to another embodiment of the invention, the depository system 700 comprises multiple geographically separated independent data storage systems. By distributing the sensitive data into distinct and independent storage facilities D1 through D4, some or all of which may be advantageously geographically separated, the depository system 700 provides redundancy along with additional security measures. For example, according to one embodiment, only data from two of the multiple data storage facilities, D1 through D4, are needed to decipher and reassemble the sensitive data. Thus, as many as two of the four data storage facilities D1 through D4 may be inoperative due to maintenance, system failure, power failure, or the like, without affecting the functionality of the trust engine 110. In addition, because, according to one embodiment, the data stored in each data storage facility is randomized and undecipherable, compromise of any individual data storage facility does not necessarily compromise the sensitive data. Moreover, in the embodiment having geographically separation of the data storage facilities, a compromise of multiple geographically remote facilities becomes increasingly difficult. In fact, even a rouge employee will be greatly challenged to subvert the needed multiple independent geographically remote data storage facilities.

Although the depository system 700 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the depository system 700. For example, the depository system 700 may comprise two or more data storage facilities. In addition, sensitive data may be mathematically operated such that portions from two or more data storage facilities are needed to reassemble and decipher the sensitive data.

As mentioned in the foregoing, the authentication engine 215 and the cryptographic engine 220 each include a data splitting module 520 and 610, respectively, for splitting sensitive data, such as, for example, the authentication data and the cryptographic key data. FIGURE 8 illustrates a flowchart of a data splitting process 800 performed by the data splitting module according to aspects of an embodiment of the invention. As shown in FIGURE 8, the data splitting process 800 begins at STEP 805 when sensitive data "S" is received by the data splitting module of the authentication engine 215 or the cryptographic engine 220. Preferably, in STEP 810, the data splitting module then generates a substantially random number, value, or string or set of bits, "A." For example, the random number A may be generated in a wide number of varying conventional techniques available to one of ordinary skill in the art, for producing high quality random numbers suitable for use in cryptographic applications. In addition, according to one embodiment, the random number A comprises a bit length equal to the bit length of the sensitive data, S.

In addition, in STEP 820 the data splitting process 800 generates another statistically random number "C." According to the preferred embodiment, the generation of the statistically random numbers A and C may advantageously be done in parallel. The data splitting module then combines the numbers A and C with the sensitive data S such that new numbers "B" and "D" are generated. For example, number B may comprise the binary combination of A XOR S and number D may comprise the binary combination of C XOR S. The foregoing combinations preferably occur in STEPS 825 and 830, respectively, and, according to one embodiment, the foregoing combinations also occur in parallel. The data splitting process 800 then proceeds to STEP 835 where the random numbers A and C and the numbers B and D are paired such that none of the pairings contain sufficient data, by themselves, to reorganize and decipher the original sensitive data S. For example, the numbers may be paired as follows: AC, AD, BC, and BD. According to one embodiment, each of the foregoing pairings is distributed to one of the depositories D1 through D4 of FIGURE 7. According to another embodiment, each of the foregoing pairings is randomly distributed to one of the depositories D1 through D4. For example, during a first data splitting process 800, the pairing AC may be sent to depository D2, through, for example, a random selection of D2's IP address. Then, during a second data splitting process 800, the pairing AC may be sent to depository D4, through, for example, a random selection of D4's IP address.

Based on the foregoing, the data splitting process 800 advantageously places portions of the sensitive data in each of the four data storage facilities D1 through D4, such that no single data storage facility D1 through D4 includes sufficient encrypted data to recreate the original sensitive data S. As mentioned in the foregoing, such randomization of the data into individually unusable encrypted portions increases security and provides for maintained trust in the data even if one of the data storage facilities, D1 through D4, is compromised.

Although the data splitting process 800 is disclosed with reference to its preferred embodiment, the invention is not intended to be limited thereby. Rather a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the data splitting process 800. For example, the data splitting process may advantageously split the data into two numbers, for example, random number A and number B and, randomly distribute A and B through two data storage facilities. Moreover, the data splitting process 800 may advantageously split the data among a wide number of data storage facilities through generation of an additional random numbers.

As mentioned in the foregoing, in order to recreate the sensitive data S, the data portions need to be derandomized and reorganized. This process may advantageously occur in the data assembling modules, 525 and 620, of the authentication engine 215 and the cryptographic engine 220, respectively. The data assembling module, for example, data assembly module 525, receives data portions from the data storage facilities D1 through D4, and reassembles the data into useable form. For example, according to one embodiment where the data splitting module 520 employed the data splitting process 800 of FIGURE 8, the data assembling module 525 uses data portions from at least two of the data storage facilities D1 through D4 to recreate the sensitive data S. For example, the pairings of AC, AD, BC, and BD, were distributed such that any two provide one of A and B, or, C and D. Noting that S = A XOR B or S = C XOR D indicates that when the data assembling module receives one of A and B, or, C and D,

the data assembling module 525 can advantageously reassemble the sensitive data S. Thus, the data assembling module 525 may assemble the sensitive data S, when, for example, it receives data portions from at least the first two of the data storage facilities D1 through D4 to respond to an assemble request by the trust engine 110.

Based on the above data splitting and assembling processes, the sensitive data S exists in usable format only in a limited area of the trust engine 110. For example, when the sensitive data S includes enrollment authentication data, usable, nonrandomized enrollment authentication data is available only in the authentication engine 215. Likewise, when the sensitive data S includes private cryptographic key data, usable, nonrandomized private cryptographic key data is available only in the cryptographic engine 220.

Although the data splitting and assembling processes are disclosed with reference to their preferred embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for splitting and reassembling the sensitive data S. For example, public-key encryption may be used to further secure the data at the data storage facilities D1 through D4.

FIGURE 9A illustrates a data flow of an enrollment process 900 according to aspects of an embodiment of the invention. As shown in FIGURE 9A, the enrollment process 900 begins at STEP 905 when a user desires to enroll with the trust engine 110 of the cryptographic system 100. According to this embodiment, the user system 105 advantageously includes a client-side applet, such as a Java-based, that queries the user to enter enrollment data, such as demographic data and enrollment authentication data. According to one embodiment, the enrollment authentication data includes user ID, password(s), biometric(s), or the like. According to one embodiment, during the querying process, the client-side applet preferably communicates with the trust engine 110 to ensure that a chosen user ID is unique. When the user ID is nonunique, the trust engine 110 may advantageously suggest a unique user ID. The client-side applet gathers the enrollment data and transmits the enrollment data, for example, through and XML document, to the trust engine 110, and in particular, to the transaction engine 205. According to one embodiment, the transmission is encoded with the public key of the authentication engine 215.

According to one embodiment, the user performs a single enrollment during STEP 905 of the enrollment process 900. For example, the user enrolls himself or herself as a particular person, such as Joe User. When Joe User desires to enroll as Joe User, CEO of Mega Corp., then according to this embodiment, Joe User enrolls a second time, receives a second unique user ID and the trust engine 110 does not associate the two identities. According to another embodiment of the invention, the enrollment process 900 provides for multiple user identities for a single user ID. Thus, in the above example, the trust engine 110 will advantageously associate the two identities of Joe User. As will be understood by a skilled artisan from the disclosure herein, a user may have many identities, for example, Joe User the head of household, Joe User the member of the Charitable Foundations, and the like. Even though the user may have multiple identities, according to this embodiment, the trust engine 110 preferably stores only one set of enrollment data. Moreover, users may advantageously add, edit/update, or delete identities as they are needed.

Although the enrollment process 900 is disclosed with reference to its preferred embodiment, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of

alternatives for gathering of enrollment data, and in particular, enrollment authentication data. For example, the applet may be common object model (COM) based applet or the like.

On the other hand, the enrollment process may include graded enrollment. For example, at a lowest level of enrollment, the user may enroll over the communication link 125 without producing documentation as to his or her identity. According to an increased level of enrollment, the user enrolls using a trusted third party, such as a digital notary. For example, and the user may appear in person to the trusted third party, produce credentials such as a birth certificate, driver's license, military ID, or the like, and the trusted third party may advantageously include, for example, their digital signature in enrollment submission. The trusted third party may include an actual notary, a government agency, such as the Post Office or Department of Motor Vehicles, a human resources person in a large company enrolling an employee, or the like. A skilled artisan will understand from the disclosure herein that a wide number of varying levels of enrollment may occur during the enrollment process 900.

After receiving the enrollment authentication data, at STEP 915, the transaction engine 205, using conventional FULL SSL technology forwards the enrollment authentication data to the authentication engine 215. In STEP 920, the authentication engine 215 decrypts the enrollment authentication data using the private key of the authentication engine 215. In addition, the authentication engine 215 employs the data splitting module to mathematically operate on the enrollment authentication data so as to split the data into at least two independently undecipherable, randomized, numbers. As mentioned in the foregoing, the at least two numbers may comprise a statistically random number and a binary XORed number. In STEP 925, the authentication engine 215 forwards each portion of the randomized numbers to one of the data storage facilities D1 through D4. As mentioned in the foregoing, the authentication engine 215 may also advantageously randomize which portions are transferred to which depositories.

Often during the enrollment process 900, the user will also desire to have a digital certificate issued such that he or she may receive encrypted documents from others outside the cryptographic system 100. As mentioned in the foregoing, the certificate authority 115 generally issues digital certificates according to one or more of several conventional standards. Generally, the digital certificate includes a public key of the user or system, which is known to everyone.

Whether the user requests a digital certificate at enrollment, or at another time, the request is transferred through the trust engine 110 to the authentication engine 215. According to one embodiment, the request includes an XML document having, for example, the proper name of the user. According to STEP 935, the authentication engine 215 transfers the request to the cryptographic engine 220 instructing the cryptographic engine 220 to generate a cryptographic key or key pair.

Upon request, at STEP 935, the cryptographic engine 220 generates at least one cryptographic key. According to one embodiment, the cryptographic handling module 625 generates a key pair, where one key is used as a private key, and one is used as a public key. The cryptographic engine 220 stores the private key and, according to one embodiment, a copy of the public key. In STEP 945, the cryptographic engine 220 transmits a request for a digital

certificate to the transaction engine 205. According to one embodiment, the request advantageously includes a standardized request, such as PKCS10, embedded in, for example, and XML document. The request for a digital certificate may advantageously correspond to one or more certificate authorities and the one or more standard formats the certificate authorities require.

In STEP 950 the transaction engine 205 forwards this request to the certificate authority 115, who, in STEP 955, returns a digital certificate. The return digital certificate may advantageously be in a standardized format, such as PKCS7, or in a proprietary format of one or more of the certificate authorities 115. In STEP 960, the digital certificate is received by the transaction engine 205, and a copy is forwarded to the user and a copy is stored with the trust engine 110. The trust engine 110 stores a copy of the certificate such that the trust engine 110 will not need to rely on the availability of the certificate authority 115. For example, when the user desires to send a digital certificate, or a third party requests the user's digital certificate, the request for the digital certificate is typically sent to the certificate authority 115. However, if the certificate authority 115 is conducting maintenance or has been victim of a failure or security compromise, the digital certificate may not be available.

At any time after issuing the cryptographic keys, the cryptographic engine 220 may advantageously employ the data splitting process 800 described above such that the cryptographic keys are split into independently undecipherable randomized numbers. Similar to the authentication data, at STEP 965 the cryptographic engine 220 transfers the randomized numbers to the data storage facilities D1 through D4.

A skilled artisan will recognize from the disclosure herein that the user may request a digital certificate anytime after enrollment. Moreover, the communications between systems may advantageously include FULL SSL or public-key encryption technologies. Moreover, the enrollment process may issue multiple digital certificates from multiple certificate authorities, including one or more proprietary certificate authorities internal or external to the trust engine 110.

As disclosed in STEPS 935 through 960, one embodiment of the invention includes the request for a certificate that is eventually stored on the trust engine 110. Because, according to one embodiment, the cryptographic handling module 625 issues the keys used by the trust engine 110, each certificate corresponds to a private key. Therefore, the trust engine 110 may advantageously provide for interoperability through monitoring the certificates owned by, or associated with, a user. For example, when the cryptographic engine 220 receives a request for a cryptographic function, the cryptographic handling module 625 may investigate the certificates owned by the requesting user to determine whether the user owns a private key matching the attributes of the request. When such a certificate exists, the cryptographic handling module 625 may use the certificate or the public or private keys associated therewith, to perform the requested function. When such a certificate does not exist, the cryptographic handling module 625 may advantageously and transparently perform a number of actions to attempt to remedy the lack of an appropriate key. For example, FIGURE 9B illustrates a flowchart of an interoperability process 970, which according to aspects of an embodiment of the invention, discloses the foregoing steps to ensure the cryptographic handling module 625 performs cryptographic functions using appropriate keys.

As shown in FIGURE 9B, the interoperability process 970 begins with STEP 972 where the cryptographic handling module 925 determines the type of certificate desired. According to one embodiment of the invention, the type of certificate may advantageously be specified in the request for cryptographic functions, or other data provided by the requestor. According to another embodiment, the certificate type may be ascertained by the data format of the request. For example, the cryptographic handling module 925 may advantageously recognize the request corresponds to a particular type.

According to one embodiment, the certificate type may include one or more algorithm standards, for example, RSA, ELGAMAL, or the like. In addition, the certificate type may include one or more key types, such as symmetric keys, public keys, strong encryption keys such as 256 bit keys, less secure keys, or the like. Moreover, the certificate type may include upgrades or replacements of one or more of the foregoing algorithm standards or keys, one or more message or data formats, one or more data encapsulation or encoding schemes, such as Base 32 or Base 64. The certificate type may also include compatibility with one or more third-party cryptographic applications or interfaces, one or more communication protocols, or one or more certificate standards or protocols. A skilled artisan will recognize from the disclosure herein that other differences may exist in certificate types, and translations to and from those differences may be implemented as disclosed herein.

Once the cryptographic handling module 625 determines the certificate type, the interoperability process 970 proceeds to STEP 974, and determines whether the user owns a certificate matching the type determined in STEP 974. When the user owns a matching certificate, for example, the trust engine 110 has access to the matching certificate through, for example, prior storage thereof, the cryptographic handling module 625 knows that a matching private key is also stored within the trust engine 110. For example, the matching private key may be stored within the depository 210 or depository system 700. The cryptographic handling module 625 may advantageously request the matching private key be assembled from, for example, the depository 210, and then in STEP 976, use the matching private key to perform cryptographic actions or functions. For example, as mentioned in the foregoing, the cryptographic handling module 625 may advantageously perform hashing, hash comparisons, data encryption or decryption, digital signature verification or creation, or the like.

When the user does not own a matching certificate, the interoperability process 970 proceeds to STEP 978 where the cryptographic handling module 625 determines whether the users owns a cross-certified certificate. According to one embodiment, cross-certification between certificate authorities occurs when a first certificate authority determines to trust certificates from a second certificate authority. In other words, the first certificate authority determines that certificates from the second certificate authority meets certain quality standards, and therefore, may be "certified" as equivalent to the first certificate authority's own certificates. Cross-certification becomes more complex when the certificate authorities issue, for example, certificates having levels of trust . For example, the first certificate authority may provide three levels of trust for a particular certificate, usually based on the degree of reliability in the enrollment process, while the second certificate authority may provide seven levels of trust. Cross-certification may advantageously track which levels and which the certificates from the second

certificate authority may be substituted for which levels and which certificates from the first. When the foregoing cross-certification is done officially and publicly between two certification authorities, the mapping of certificates and levels to one another is often called "chaining."

According to another embodiment of the invention, the cryptographic handling module 625 may advantageously develop cross-certifications outside those agreed upon by the certificate authorities. For example, the cryptographic handling module 625 may access a first certificate authority's certificate practice statement (CPS), or other published policy statement, and using, for example, the authentication tokens required by particular trust levels, match the first certificate authority's certificates to those of another certificate authority.

When, in STEP 978, the cryptographic handling module 625 determines that the users owns a cross-certified certificate, the interoperability process 970 proceeds to STEP 976, and performs the cryptographic action or function using the cross-certified public key, private key, or both. Alternatively, when the cryptographic handling module 625 determines that the users does not own a cross-certified certificate, the interoperability process 970 proceeds to STEP 980, where the cryptographic handling module 625 selects a certificate authority that issues the requested certificate type, or a certificate cross-certified thereto. In STEP 982, the cryptographic handling module 625 determines whether the user enrollment authentication data, discussed in the foregoing, meets the authentication requirements of the chosen certificate authority. For example, if the user enrolled over a network by, for example, answering demographic and other questions, the authentication data provided may establish a lower level of trust than a user providing biometric data and appearing before a third-party, such as, for example, a notary. According to one embodiment, the foregoing authentication requirements may advantageously be provided in the chosen authentication authority's CPS.

When the user has provided the trust engine 110 with enrollment authentication data meeting the requirements of chosen certificate authority, the interoperability process 970 proceeds to STEP 984, where the cryptographic handling module 625 acquires the certificate from the chosen certificate authority. According to one embodiment, the cryptographic handling module 625 acquires the certificate by following STEPS 945 through 960 of the enrollment process 900. For example, the cryptographic handling module 625 may advantageously employ one or more public keys from one or more of the key pairs already available to the cryptographic engine 220, to request the certificate from the certificate authority. According to another embodiment, the cryptographic handling module 625 may advantageously generate one or more new key pairs, and use the public keys corresponding thereto, to request the certificate from the certificate authority.

According to another embodiment, the trust engine 110 may advantageously include one or more certificate issuing modules capable of issuing one or more certificate types. According to this embodiment, the certificate issuing module may provide the foregoing certificate. When the cryptographic handling module 625 acquires the certificate, the interoperability process 970 proceeds to STEP 976, and performs the cryptographic action or function using the public key, private key, or both corresponding to the acquired certificate.

When the user, in STEP 982, has not provided the trust engine 110 with enrollment authentication data meeting the requirements of chosen certificate authority, the cryptographic handling module 625 determines, in STEP

986 whether there are other certificate authorities that have different authentication requirements. For example, the cryptographic handling module 625 may look for certificate authorities having lower authentication requirements, but still issue the chosen certificates, or cross-certifications thereof.

When the foregoing certificate authority having lower requirements exists, the interoperability process 970 proceeds to STEP 980 and chooses that certificate authority. Alternatively, when no such certificate authority exists, in STEP 988, the trust engine 110 may request additional authentication tokens from the user. For example, the trust engine 110 may request new enrollment authentication data comprising, for example, biometric data. Also, the trust engine 110 may request the user appear before a trusted third party and provide appropriate authenticating credentials, such as, for example, appearing before a notary with a drivers license, social security card, bank card, birth certificate, military ID, or the like. When the trust engine 110 receives updated authentication data, the interoperability process 970 proceeds to Step 984 and acquires the foregoing chosen certificate.

Through the foregoing interoperability process 970, the cryptographic handling module 625 advantageously provides seamless, transparent, translations and conversions between differing cryptographic systems. A skilled artisan will recognize from the disclosure herein, a wide number of advantages and implementations of the foregoing interoperable system. For example, the foregoing STEP 986 of the interoperability process 970 may advantageously include aspects of trust arbitrage, discussed in further detail below, where the certificate authority may under special circumstances accept lower levels of cross-certification. In addition, the interoperability process 970 may include ensuring interoperability between and employment of standard certificate revocations, such as employing certificate revocation lists (CRL), online certificate status protocols (OCSP), or the like.

FIGURE 10 illustrates a data flow of an authentication process 1000 according to aspects of an embodiment of the invention. According to one embodiment, the authentication process 1000 includes gathering current authentication data from a user and comparing that to the enrollment authentication data of the user. For example, the authentication process 1000 begins at STEP 1005 where a user desires to perform a transaction with, for example, a vendor. Such transactions may include, for example, selecting a purchase option, requesting access to a restricted area or device of the vendor system 120, or the like. At STEP 1010, a vendor provides the user with a transaction ID and an authentication request. The transaction ID may advantageously include a 192 bit quantity having a 32 bit timestamp concatenated with a 128 bit random quantity, or a "nonce," concatenated with a 32 bit vendor specific constant. Such a transaction ID uniquely identifies the transaction such that copycat transactions can be refused by the trust engine 110.

The authentication request may advantageously include what level of authentication is needed for a particular transaction. For example, the vendor may specify a particular level of confidence that is required for the transaction at issue. If authentication cannot be made to this level of confidence, as will be discussed below, the transaction will not occur without either further authentication by the user to raise the level of confidence, or a change in the terms of the authentication between the vendor and the server. These issues are discussed more completely below.

According to one embodiment, the transaction ID and the authentication request may advantageously generated by a vendor-side applet or other software program. In addition, the transmission of the transaction ID and authentication data may include one or more XML documents encrypted using conventional SSL technology, such as, for example, ½ SSL, or, in other words vendor-side authenticated SSL.

After the user system 105 receives the transaction ID and authentication request, the user system 105 gathers the current authentication data, potentially including current biometric information, from the user. The user system 105, at STEP 1015, encrypts at least the current authentication data " B' " and the transaction ID, with the public key of the authentication engine 215, and transfers that data to the trust engine 110. The transmission preferably comprises XML documents encrypted with at least conventional ½ SSL technology. In STEP 1020, the transaction engine 205 receives the transmission, preferably recognizes the data format or request in the URL or URI, and forwards the transmission to the authentication engine 215.

During STEPS 1015 and 1020, the vendor system 120, at STEP 1025, forwards the transaction ID and the authentication request to the trust engine 110, using the preferred FULL SSL technology. This communication may also include a vendor ID, although vendor identification may also be communicated through a non-random portion of the transaction ID. At STEPS 1030 and 1035, the transaction engine 205 receives the communication, creates a record in the audit trail, and generates a request for the user's enrollment authentication data to be assembled from the data storage facilities D1 through D4. At STEP 1040, the depository system 700 transfers the portions of the enrollment authentication data corresponding to the user to the authentication engine 215. At STEP 1045, the authentication engine 215 decrypts the transmission using its private key and compares the enrollment authentication data to the current authentication data provided by the user.

The comparison of STEP 1045 may advantageously apply heuristical context sensitive authentication, as referred to in the forgoing, and discussed in further detail below. For example, if the biometric information received does not match perfectly, a lower confidence match results. In particular embodiments, the level of confidence of the authentication is balanced against the nature of the transaction and the desires of both the user and the vendor. Again, this is discussed in greater detail below.

At STEP 1050, the authentication engine 215 fills in the authentication request with the result of the comparison of STEP 1045. According to one embodiment of the invention, the authentication request is filled with a YES/NO or TRUE/FALSE result of the authentication process 1000. In STEP 1055 the filled-in authentication request is returned to the vendor for the vendor to act upon, for example, allowing the user to complete the transaction that initiated the authentication request. According to one embodiment, a confirmation message is passed to the user.

Based on the foregoing, the authentication process 1000 advantageously keeps sensitive data secure and produces results configured to maintain the integrity of the sensitive data. For example, the sensitive data is assembled only inside the authentication engine 215. For example, the enrollment authentication data is undecipherable until it is assembled in the authentication engine 215 by the data assembling module, and the current authentication data is undecipherable until it is unwrapped by the conventional SSL technology and the private key of

the authentication engine 215. Moreover, the authentication result transmitted to the vendor does not include the sensitive data, and the user may not even know whether he or she produced valid authentication data.

Although the authentication process 1000 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the authentication process 1000. For example, the vendor may advantageously be replaced by almost any requesting application, even those residing with the user system 105. For example, a client application, such as Microsoft Word, may use an application program interface (API) or a cryptographic API (CAPI) to request authentication before unlocking a document. Alternatively, a mail server, a network, a cellular phone, a personal or mobile computing device, a workstation, or the like, may all make authentication requests that can be filled by the authentication process 1000. In fact, after providing the foregoing trusted authentication process 1000, the requesting application or device may provide access to or use of a wide number of electronic or computer devices or systems.

Moreover, the authentication process 1000 may employ a wide number of alternative procedures in the event of authentication failure. For example, authentication failure may maintain the same transaction ID and request that the user reenter his or her current authentication data. As mentioned in the foregoing, use of the same transaction ID allows the comparator of the authentication engine 215 to monitor and limit the number of authentication attempts for a particular transaction, thereby creating a more secure cryptographic system 100.

In addition, the authentication process 1000 may be advantageously be employed to develop elegant single sign-on solutions, such as, unlocking a sensitive data vault. For example, successful or positive authentication may provide the authenticated user the ability to automatically access any number of passwords for an almost limitless number of systems and applications. For example, authentication of a user may provide the user access to password, login, financial credentials, or the like, associated with multiple online vendors, a local area network, various personal computing devices, Internet service providers, auction providers, investment brokerages, or the like. By employing a sensitive data vault, users may choose truly large and random passwords because they no longer need to remember them through association. Rather, the authentication process 1000 provides access thereto. For example, a user may choose a random alphanumeric string that is twenty plus digits in length rather than something associated with a memorable data, name, etc.

According to one embodiment, a sensitive data vault associated with a given user may advantageously be stored in the data storage facilities of the depository 210, or split and stored in the depository system 700. According to this embodiment, after positive user authentication, the trust engine 110 serves the requested sensitive data, such as, for example, to the appropriate password to the requesting application. According to another embodiment, the trust engine 110 may include a separate system for storing the sensitive data vault. For example, the trust engine 110 may include a stand-alone software engine implementing the data vault functionality and figuratively residing "behind" the foregoing front-end security system of the trust engine 110. According to this embodiment, the software engine

serves the requested sensitive data after the software engine receives a signal indicating positive user authentication from the trust engine 110.

In yet another embodiment, the data vault may be implemented by a third-party system. Similar to the software engine embodiment, the third-party system may advantageously serve the requested sensitive data after the third-party system receives a signal indicating positive user authentication from the trust engine 110. According to yet another embodiment, the data vault may be implemented on the user system 105. A user-side software engine may advantageously serve the foregoing data after receiving a signal indicating positive user authentication from the trust engine 110.

Although the foregoing data vaults are disclosed with reference to alternative embodiments, a skilled artisan will recognize from the disclosure herein, a wide number of additional implementations thereof. For example, a particular data vault may include aspects from some or all of the foregoing embodiments. In addition, any of the foregoing data vaults may employ one or more authentication requests at varying times. For example, any of the data vaults may require authentication every one or more transactions, periodically, every one ore more sessions, every access to one or more Webpages or Websites, at one or more other specified intervals, or the like.

FIGURE 11 illustrates a data flow of a signing process 1100 according to aspects of an embodiment of the invention. As shown in FIGURE 11, the signing process 1100 includes steps similar to those of the authentication process 1000 described in the foregoing with reference to FIGURE 10. According to one embodiment of the invention, the signing process 1100 first authenticates the user and then performs one or more of several digital signing functions as will be discussed in further detail below. According to another embodiment, the signing process 1100 may advantageously store data related thereto, such as hashes of messages or documents, or the like. This data may advantageously be used in an audit or any other event, such as for example, when a participating party attempts to repudiate a transaction.

As shown in FIGURE 11, during the authentication steps, the user and vendor may advantageously agree on a message, such as, for example, a contract. During signing, the signing process 1100 advantageously ensures that the contract signed by the user is identical to the contract supplied by the vendor. Therefore, according to one embodiment, during authentication, the vendor and the user include a hash of their respective copies of the message or contract, in the data transmitted to the authentication engine 215. By employing only a hash of a message or contract, the trust engine 110 may advantageously store a significantly reduced amount of data, providing for a more efficient and cost effective cryptographic system. In addition, the stored hash may be advantageously compared to a hash of a document in question to determine whether the document in question matches one signed by any of the parties. The ability to determine whether the document is identical to one relating to a transaction provides for additional evidence that can be used against a claim for repudiation by a party to a transaction.

In STEP 1103, the authentication engine 215 assembles the enrollment authentication data and compares it to the current authentication data provided by the user. When the comparator of the authentication engine 215 indicates that the enrollment authentication data matches the current authentication data, the comparator of the

authentication engine 215 also compares the hash of the message supplied by the vendor to the hash of the message supplied by the user. Thus, the authentication engine 215 advantageously ensures that the message agreed to by the user is identical to that agreed to by the vendor.

In STEP 1105, the authentication engine 215 transmits a digital signature request to the cryptographic engine 220. According to one embodiment of the invention, the request includes a hash of the message or contract. However, a skill artisan will recognize from the disclosure herein that the cryptographic engine 220 may encrypt virtually any text to form the desired digital signature. Returning to STEP 1105, the digital signature request preferably comprises an XML document communicated through conventional SSL technologies.

In STEP 1110, the authentication engine 215 transmits a request to each of the data storage facilities D1 through D4, such that each of the data storage facilities D1 through D4 transmit their respective portion of the cryptographic key or keys corresponding to a signing party. According to another embodiment, the cryptographic engine 220 employs some or all of the steps of the interoperability process 970 discussed in the foregoing, such that the cryptographic engine 220 first determines the appropriate key or keys to request from the depository 210 or the depository system 700 for the signing party, and takes actions to provide appropriate matching keys. According to still another embodiment, the authentication engine 215 or the cryptographic engine 220 may advantageously request one or more of the keys associated with the signing party and stored in the depository 210 or depository system 700.

According to one embodiment, the signing party includes one or both the user and the vendor. In such case, the authentication engine 215 advantageously requests the cryptographic keys corresponding to the user and/or the vendor. According to another embodiment, the signing party includes the trust engine 110. In this embodiment, the trust engine 110 is certifying that the authentication process 1000 properly authenticated the user, vendor, or both. Therefore, the authentication engine 215 requests the cryptographic key of the trust engine 110, such as, for example, the key belonging to the cryptographic engine 220, to perform the digital signature. According to another embodiment, the trust engine 110 performs a digital notary-like function. In this embodiment, the signing party includes the user, vendor, or both, along with the trust engine 110. Thus, the trust engine 110 provides the digital signature of the user and/or vendor, and then indicates with its own digital signature that the user and/or vendor were properly authenticated. In this embodiment, the authentication engine 215 may advantageously request assembly of the cryptographic keys corresponding to the user, the vendor, or both. According to another embodiment, the authentication engine 215 may advantageously request assembly of the cryptographic keys corresponding to the trust engine 110.

According to another embodiment, the trust engine 110 performs power of attorney-like functions. For example, the trust engine 110 may digitally sign the message on behalf of a third party. In such case, the authentication engine 215 requests the cryptographic keys associated with the third party. According to this embodiment, the signing process 1100 may advantageously include authentication of the third party, before allowing power of attorney-like functions. In addition, the authentication process 1000 may include a check for third party

constraints, such as, for example, business logic or the like dictating when and in what circumstances a particular third-party's signature may be used.

Based on the foregoing, in STEP 1110, the authentication engine requested the cryptographic keys from the data storage facilities D1 through D4 corresponding to the signing party. In STEP 1115, the data storage facilities D1 through D4 transmit their respective portions of the cryptographic key corresponding to the signing party to the cryptographic engine 220. According to one embodiment, the foregoing transmissions include SSL technologies. According to another embodiment, the foregoing transmissions may advantageously be super-encrypted with the public key of the cryptographic engine 220.

In STEP 1120, the cryptographic engine 220 assembles the foregoing cryptographic keys of the signing party and encrypts the message therewith, thereby forming the digital signature(s). In STEP 1125 of the signing process 1100, the cryptographic engine 220 transmits the digital signature(s) to the authentication engine 215. In STEP 1130, the authentication engine 215 transmits the filled-in authentication request along with a copy of the hashed message and the digital signature(s) to the transaction engine 205. In STEP 1135, the transaction engine 205 transmits a receipt comprising the transaction ID, an indication of whether the authentication was successful, and the digital signature(s), to the vendor. According to one embodiment, the foregoing transmission may advantageously include the digital signature of the trust engine 110. For example, the trust engine 110 may encrypt the hash of the receipt with its private key, thereby forming a digital signature to be attached to the transmission to the vendor.

According to one embodiment, the transaction engine 205 also transmits a confirmation message to the user.

Although the signing process 1100 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the signing process 1100. For example, the vendor may be replaced with a user application, such as an email application. For example, the user may wish to digitally sign a particular email with his or her digital signature. In such an embodiment, the transmission throughout the signing process 1100 may advantageously include only one copy of a hash of the message. Moreover, a skilled artisan will recognize from the disclosure herein that a wide number of client applications may request digital signatures. For example, the client applications may comprise word processors, spreadsheets, emails, voicemail, access to restricted system areas, or the like.

In addition, a skilled artisan will recognize from the disclosure herein that STEPS 1105 through 1120 of the signing process 1100 may advantageously employ some or all of the steps of the interoperability process 970 of FIGURE 9B, thereby providing interoperability between differing cryptographic systems that may, for example, need to process the digital signature under differing signature types.

FIGURE 12 illustrates a data flow of an encryption/decryption process 1200 according to aspects of an embodiment of the invention. As shown in FIGURE 12, the decryption process 1200 begins by authenticating the user using the authentication process 1000. According to one embodiment, the authentication process 1000 includes in the authentication request, a synchronous session key. For example, in conventional PKI technologies, it is understood by

skilled artisans that encrypting or decrypting data using public and private keys is mathematically intensive and may require significant system resources. However, in symmetric key cryptographic systems, or systems where the sender and receiver of a message share a single common key that is used to encrypt and decrypt a message, the mathematical operations are significantly simpler and faster. Thus, in the conventional PKI technologies, the sender of a message will generate synchronous session key, and encrypt the message using the simpler, faster symmetric key system. Then, the sender will encrypt the session key with the public key of the receiver. The encrypted session key will be attached to the synchronously encrypted message and both data are sent to the receiver. The receiver uses his or her private key to decrypt the session key, and then uses the session key to decrypt the message. Based on the foregoing, the simpler and faster symmetric key system is used for the majority of the encryption/decryption processing. Thus, in the decryption process 1200, the decryption advantageously assumes that a synchronous key has been encrypted with the public key of the user. Thus, as mentioned in the foregoing, the encrypted session key is included in the authentication request.

Returning to the decryption process 1200, after the user has been authenticated in STEP 1205, the authentication engine 215 forwards the encrypted session key to the cryptographic engine 220. In STEP 1210, the authentication engine 215 forwards a request to each of the data storage facilities, D1 through D4, requesting the cryptographic key data of the user. In STEP 1215, each data storage facility, D1 through D4, transmits their respective portion of the cryptographic key to the cryptographic engine 220. According to one embodiment, the foregoing transmission is encrypted with the public key of the cryptographic engine 220.

In STEP 1220 of the decryption process 1200, the cryptographic engine 220 assembles the cryptographic key and decrypts the session key therewith. In STEP 1225, the cryptographic engine forwards the session key to the authentication engine 215. In STEP 1227, the authentication engine 215 fills in the authentication request including the decrypted session key, and transmits the filled-in authentication request to the transaction engine 205. In STEP 1230, the transaction engine 205 forwards the authentication request along with the session key to the requesting application or vendor. Then, according to one embodiment, the requesting application or vendor uses the session key to decrypt the encrypted message.

Although the decryption process 1200 is disclosed with reference to its preferred and alternative embodiments, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the decryption process 1200. For example, the decryption process 1200 may forego synchronous key encryption and rely on full public-key technology. In such an embodiment, the requesting application may transmit the entire message to the cryptographic engine 220, or, may employ some type of compression or reversible hash in order to transmit the message to the cryptographic engine 220. A skilled artisan will also recognize from the disclosure herein that the foregoing communications may advantageously include XML documents wrapped in SSL technology.

The encryption/decryption process 1200 also provides for encryption of documents or other data. Thus, in STEP 1235, a requesting application or vendor may advantageously transmit to the transaction engine 205 of the trust engine 110, a request for the public key of the user. The requesting application or vendor makes this request

because the requesting application or vendor uses the public key of the user, for example, to encrypt the session key that will be used to encrypt the document or message. As mentioned in the enrollment process 900, the transaction engine 205 stores a copy of the digital certificate of the user, for example, in the mass storage 225. Thus, in STEP 1240 of the encryption process 1200, the transaction engine 205 requests the digital certificate of the user from the mass storage 225. In STEP 1245, the mass storage 225 transmits the digital certificate corresponding to the user, to the transaction engine 205. In STEP 1250, the transaction engine 205 transmits the digital certificate to the requesting application or vendor. According to one embodiment, the encryption portion of the encryption process 1200 does not include the authentication of a user. This is because the requesting vendor needs only the public key of the user, and is not requesting any sensitive data.

A skilled artisan will recognize from the disclosure herein that if a particular user does not have a digital certificate, the trust engine 110 may employ some or all of the enrollment process 900 in order to generate a digital certificate for that particular user. Then, the trust engine 110 may initiate the encryption/decryption process 1200 and thereby provide the appropriate digital certificate. In addition, a skilled artisan will recognize from the disclosure herein that STEPS 1220 and 1235 through 1250 of the encryption/decryption process 1200 may advantageously employ some or all of the steps of the interoperability process of FIGURE 9B, thereby providing interoperability between differing cryptographic systems that may, for example, need to process the encryption.

FIGURE 13 illustrates a simplified block diagram of a trust engine system 1300 according to aspects of yet another embodiment of the invention. As shown in FIGURE 13, the trust engine system 1300 comprises a plurality of distinct trust engines 1305, 1310, 1315, and 1320, respectively. To facilitate a more complete understanding of the invention, FIGURE 13 illustrates each trust engine, 1305, 1310, 1315, and 1320 as having a transaction engine, a depository, and an authentication engine. However, a skilled artisan will recognize that each transaction engine may advantageously comprise some or all of the elements and communication channels disclosed with reference to FIGURES 1-8. For example, one embodiment may advantageously include trust engines having transaction engines, depositories, and cryptographic servers.

According to one embodiment of the invention, each of the trust engines 1305, 1310, 1315 and 1320 are geographically separated, such that, for example, the trust engine 1305 may reside in a first location, the trust engine 1310 may reside in a second location, the trust engine 1315 may reside in a third location, and the trust engine 1320 may reside in a fourth location. The foregoing geographic separation advantageously decreases system response time while increasing the security of the overall trust engine system 1300.

For example, when a user logs onto the cryptographic system 100, the user may be nearest the first location and may desire to be authenticated. As described with reference to FIGURE 10, to be authenticated, the user provides current authentication data, such as a biometric or the like, and the current authentication data is compared to that user's enrollment authentication data. Therefore, according to one example, the user advantageously provides current authentication data to the geographically nearest trust engine 1305. The transaction engine 1321 of the trust engine 1305 then forwards the current authentication data to the authentication engine 1322 also residing at the first

location. According to another embodiment, the transaction engine 1321 forwards the current authentication data to one or more of the authentication engines of the trust engines 1310, 1315, or 1320.

The transaction engine 1321 also requests the assembly of the enrollment authentication data from the depositories of, for example, each of the trust engines, 1305 through 1320. According to this embodiment, each depository provides its portion of the enrollment authentication data to the authentication engine 1322 of the trust engine 1305. The authentication engine 1322 then employs the encrypted data portions from, for example, the first two depositories to respond, and assembles the enrollment authentication data into deciphered form. The authentication engine 1322 compares the enrollment authentication data with the current authentication data and returns an authentication result to the transaction engine 1321 of the trust engine 1305.

Based on the above, the trust engine system 1300 employs the nearest one of a plurality of geographically separated trust engines, 1305 through 1320, to perform the authentication process. According to one embodiment of the invention, the routing of information to the nearest transaction engine may advantageously be performed at client-side applets executing on one or more of the user system 105, vendor system 120, or certificate authority 115. According to an alternative embodiment, a more sophisticated decision process may be employed to select from the trust engines 1305 through 1320. For example, the decision may be based on the availability, operability, speed of connections, load, performance, geographic proximity, or a combination thereof, of a given trust engine.

In this way, the trust engine system 1300 lowers its response time while maintaining the security advantages associated with geographically remote data storage facilities, such as those discussed with reference to FIGURE 7 where each data storage facility stores randomized portions of sensitive data. For example, a security compromise at, for example, the depository 1325 of the trust engine 1315 does not necessarily compromise the sensitive data of the trust engine system 1300. This is because the depository 1325 contains only non-decipherable randomized data that, without more, is entirely useless.

According to another embodiment, the trust engine system 1300 may advantageously include multiple cryptographic engines arranged similar to the authentication engines. The cryptographic engines may advantageously perform cryptographic functions such as those disclosed with reference to FIGURES 1-8. According to yet another embodiment, the trust engine system 1300 may advantageously replace the multiple authentication engines with multiple cryptographic engines, thereby performing cryptographic functions such as those disclosed with reference to FIGURES 1-8. According to yet another embodiment of the invention, the trust engine system 1300 may replace each multiple authentication engine with an engine having some or all of the functionality of the authentication engines, cryptographic engines, or both, as disclosed in the foregoing.

Although the trust engine system 1300 is disclosed with reference to its preferred and alternative embodiments, a skilled artisan will recognize that the trust engine system 1300 may comprise portions of trust engines 1305 through 1320. For example, the trust engine system 1300 may include one or more transaction engines, one or more depositories, one or more authentication engines, or one or more cryptographic engines.

FIGURE 14 illustrates a simplified block diagram of a trust engine system 1400 according to aspects of yet another embodiment of the invention. As shown in FIGURE 14, the trust engine system 1400 includes multiple trust engines 1405, 1410, 1415 and 1420. According to one embodiment, each of the trust engines 1405, 1410, 1415 and 1420, comprise some or all of the elements of trust engine 110 disclosed with reference to FIGURES 1-8. According to this embodiment, when the client side applets of the user system 105, the vendor system 120, or the certificate authority 115, communicate with the trust engine system 1400, those communications are sent to the IP address of each of the trust engines 1405 through 1420. Further, each transaction engine of each of the trust engines, 1405, 1410, 1415, and 1420, behaves similar to the transaction engine 1321 of the trust engine 1305 disclosed with reference to FIGURE 13. For example, during an authentication process, each transaction engine of each of the trust engines 1405, 1410, 1415, and 1420 transmits the current authentication data to their respective authentication engines and transmits a request to assemble the randomized data stored in each of the depositories of each of the trust engines 1405 through 1420. FIGURE 14 does not illustrate all of these communications, as such illustration would become overly complex. Continuing with the authentication process, each of the depositories then communicates its portion of the randomized data to each of the authentication engines of the each of the trust engines 1405 through 1420. Each of the authentication engines of the each of the trust engines employs its comparator to determine whether the current authentication data matches the enrollment authentication data provided by the depositories of each of the trust engines 1405 through 1420. According to this embodiment, the result of the comparison by each of the authentication engines is then transmitted to a redundancy module of the other three trust engines. For example, the result of the authentication engine from the trust engine 1405 is transmitted to the redundancy modules of the trust engines 1410, 1415, and 1420. Thus, the redundancy module of the trust engine 1405 likewise receives the result of the authentication engines from the trust engines 1410, 1415, and 1420.

FIGURE 15 illustrates a block diagram of the redundancy module of FIGURE 14. The redundancy module comprises a comparator configured to receive the authentication result from three authentication engines and transmit that result to the transaction engine of the fourth trust engine. The comparator compares the authentication result form the three authentication engines, and if two of the results agree, the comparator concludes that the authentication result should match that of the two agreeing authentication engines. This result is then transmitted back to the transaction engine corresponding to the trust engine not associated with the three authentication engines.

Based on the foregoing, the redundancy module determines an authentication result from data received from authentication engines that are preferably geographically remote from the trust engine of that the redundancy module. By providing such redundancy functionality, the trust engine system 1400 ensures that a compromise of the authentication engine of one of the trust engines 1405 through 1420, is insufficient to compromise the authentication result of the redundancy module of that particular trust engine. A skilled artisan will recognize that redundancy module functionality of the trust engine system 1400 may also be applied to the cryptographic engine of each of the trust engines 1405 through 1420. However, such cryptographic engine communication was not shown in FIGURE 14

to avoid complexity. Moreover, a skilled artisan will recognize a wide number of alternative authentication result conflict resolution algorithms for the comparator of FIGURE 15.

According to yet another embodiment of the invention, the trust engine system 1400 may advantageously employ the redundancy module during cryptographic comparison steps. For example, some or all of the foregoing redundancy module disclosure with reference to FIGURES 14 and 15 may advantageously be implemented during a hash comparison of documents provided by one or more parties during a particular transaction.

Although the foregoing invention has been described in terms of certain preferred and alternative embodiments, other embodiments will be apparent to those of ordinary skill in the art from the disclosure herein. For example, the trust engine 110 may issue short-term certificates, where the private cryptographic key is released to the user for a predetermined period of time. For example, current certificate standards include a validity field that can be set to expire after a predetermined amount of time. Thus, the trust engine 110 may release a private key to a user where the private key would be valid for, for example, 24 hours. According to such an embodiment, the trust engine 110 may advantageously issue a new cryptographic key pair to be associated with a particular user and then release the private key of the new cryptographic key pair. Then, once the private cryptographic key is released, the trust engine 110 immediately expires any internal valid use of such private key, as it is no longer securable by the trust engine 110.

In addition, a skilled artisan will recognize that the cryptographic system 100 or the trust engine 110 may include the ability to recognize types of devices, such as a laptop, a cell phone, a network or the like. According to one embodiment, such recognition may come from data supplied in the request for a particular service, such as, a request for authentication leading to access or use, a request for cryptographic functionality, or the like. According to one embodiment, the foregoing request may include a unique device identifier, such as, for example, a processor ID. Alternatively, the request may include data in a particular recognizable data format. For example, mobile and satellite phones often do not include the processing power for full X509.v3 heavy encryption certificates, and therefore do not request them. According to this embodiment, the trust engine 110 may recognize the type of data format presented, and respond only in kind.

In an additional aspect of the system described above, context sensitive authentication can be provided using various techniques as will be described below. Context sensitive authentication, for example as shown in FIGURE 16, provides the possibility of evaluating not only the actual data which is sent by the user when attempting to authenticate himself, but also the circumstances surrounding the generation and delivery of that data. Such techniques may also support transaction specific trust arbitrage between the user and trust engine 110 or between the vendor and trust engine 110, as will be described below.

As discussed above, authentication is the process of proving that a user is who he says he is. Generally, authentication requires demonstrating some fact to an authentication authority. The trust engine 110 of the present invention represents the authority to which a user must authenticate himself. The user must demonstrate to the trust engine 110 that he is who he says he is by either: knowing something that only the user should know (knowledge-

based authentication), having something that only the user should have (token-based authentication), or by being something that only the user should be (biometric-based authentication).

Examples of knowledge-based authentication include without limitation a password, PIN number, or lock combination. Examples of token-based authentication include without limitation a house key, a physical credit card, a driver's license, or a particular phone number. Examples of biometric-based authentication include without limitation a fingerprint, a voice analysis, or a retinal scan.

Each type of authentication has particular advantages and disadvantages, and each provides a different level of security. For example, it is generally harder to create a false fingerprint that matches someone else's than it is to overhear someone's password and repeat it. Each type of authentication also requires a different type of data to be known to the authenticating authority in order to verify someone using that form of authentication.

As used herein, "authentication" will refer broadly to the overall process of verifying someone's identity to be who he says he is. An "authentication technique" will refer to a particular type of authentication based upon a particular piece of knowledge, physical token, or biometric reading. "Authentication data" refers to information which is sent to or otherwise demonstrated to an authentication authority in order to establish identity. "Enrollment data" will refer to the data which is initially submitted to an authentication authority in order to establish a baseline for comparison with authentication data. An "authentication instance" will refer to the data associated with an attempt to authenticate by an authentication technique.

The internal protocols and communications involved in the process of authenticating a user is described with reference to FIGURE 10 above. The part of this process within which the context sensitive authentication takes place occurs within the comparison step shown as STEP 1045 of FIGURE 10. This step takes place within the authentication engine 215 and involves assembling the enrollment data 410 retrieved from the depository 210 and comparing the authentication data provided by the user to it. One particular embodiment of this process is shown in FIGURE 16 and described below.

The current authentication data provided by the user and the enrollment data retrieved from the depository 210 are received by the authentication engine 215 in STEP 1600 of FIGURE 16. Both of these sets of data may contain data which is related to separate techniques of authentication. The authentication engine 215 separates the authentication data associated with each individual authentication instance in STEP 1605. This is necessary so that the authentication data is compared with the appropriate subset of the enrollment data for the user (e.g. fingerprint authentication data should be compared with fingerprint enrollment data, rather than password enrollment data).

Generally, authenticating a user involves one or more individual authentication instances, depending on which authentication techniques are available to the user. These methods are limited by the enrollment data which were provided by the user during his enrollment process (if the user did not provide a retinal scan when enrolling, he will not be able to authenticate himself using a retinal scan), as well as the means which may be currently available to the user (e.g. if the user does not have a fingerprint reader at his current location, fingerprint authentication will not be practical). In some cases, a single authentication instance may be sufficient to authenticate a user; however, in

certain circumstances a combination of multiple authentication instances may be used in order to more confidently authenticate a user for a particular transaction.

Each authentication instance consists of data related to a particular authentication technique (e.g. fingerprint, password, smart card, etc.) and the circumstances which surround the capture and delivery of the data for that particular technique. For example, a particular instance of attempting to authenticate via password will generate not only the data related to the password itself, but also circumstantial data, known as "metadata", related to that password attempt. This circumstantial data includes information such as: the time at which the particular authentication instance took place, the network address from which the authentication information was delivered, as well as any other information as is known to those of skill in the art which may be determined about the origin of the authentication data (the type of connection, the processor serial number, etc.).

In many cases, only a small amount of circumstantial metadata will be available. For example, if the user is located on a network which uses proxies or network address translation or another technique which masks the address of the originating computer, only the address of the proxy or router may be determined. Similarly, in many cases information such as the processor serial number will not be available because of either limitations of the hardware or operating system being used, disabling of such features by the operator of the system, or other limitations of the connection between the user's system and the trust engine 110.

As shown in FIGURE 16, once the individual authentication instances represented within the authentication data are extracted and separated in STEP 1605, the authentication engine 215 evaluates each instance for its reliability in indicating that the user is who he claims to be. The reliability for a single authentication instance will generally be determined based on several factors. These may be grouped as factors relating to the reliability associated with the authentication technique, which are evaluated in STEP 1610, and factors relating to the reliability of the particular authentication data provided, which are evaluated in STEP 1615. The first group includes without limitation the inherent reliability of the authentication technique being used, and the reliability of the enrollment data being used with that method. The second group includes without limitation the degree of match between the enrollment data and the data provided with the authentication instance, and the metadata associated with that authentication instance. Each of these factors may vary independently of the others.

The inherent reliability of an authentication technique is based on how hard it is for an imposter to provide someone else's correct data, as well as the overall error rates for the authentication technique. For passwords and knowledge based authentication methods, this reliability is often fairly low because there is nothing that prevents someone from revealing their password to another person and for that second person to use that password. Even a more complex knowledge based system may have only moderate reliability since knowledge may be transferred from person to person fairly easily. Token based authentication, such as having a proper smart card or using a particular terminal to perform the authentication, is similarly of low reliability used by itself, since there is no guarantee that the right person is in possession of the proper token.

However, biometric techniques are more inherently reliable because it is generally difficult to provide someone else with the ability to use your fingerprints in a convenient manner, even intentionally. Because subverting biometric authentication techniques is more difficult, the inherent reliability of biometric methods is generally higher than that of purely knowledge or token based authentication techniques. However, even biometric techniques may have some occasions in which a false acceptance or false rejection is generated. These occurrences may be reflected by differing reliabilities for different implementations of the same biometric technique. For example, a fingerprint matching system provided by one company may provide a higher reliability than one provided by a different company because one uses higher quality optics or a better scanning resolution or some other improvement which reduces the occurrence of false acceptances or false rejections.

Note that this reliability may be expressed in different manners. The reliability is desirably expressed in some metric which can be used by the heuristics 530 and algorithms of the authentication engine 215 to calculate the confidence level of each authentication. One preferred mode of expressing these reliabilities is as a percentage or fraction. For instance, fingerprints might be assigned an inherent reliability of 97%, while passwords might only be assigned an inherent reliability of 50%. Those of skill in the art will recognize that these particular values are merely exemplary and may vary between specific implementations.

The second factor for which reliability must be assessed is the reliability of the enrollment. This is part of the "graded enrollment" process referred to above. This reliability factor reflects the reliability of the identification provided during the initial enrollment process. For instance, if the individual initially enrolls in a manner where they physically produce evidence of their identity to a notary or other public official, and enrollment data is recorded at that time and notarized, the data will be more reliable than data which is provided over a network during enrollment and only vouched for by a digital signature or other information which is not truly tied to the individual.

Other enrollment techniques with varying levels of reliability include without limitation: enrollment at a physical office of the trust engine 110 operator; enrollment at a user's place of employment; enrollment at a post office or passport office; enrollment through an affiliated or trusted party to the trust engine 110 operator; anonymous or pseudonymous enrollment in which the enrolled identity is not yet identified with a particular real individual, as well as such other means as are known in the art.

These factors reflect the trust between the trust engine 110 and the source of identification provided during the enrollment process. For instance, if enrollment is performed in association with an employer during the initial process of providing evidence of identity, this information may be considered extremely reliable for purposes within the company, but may be trusted to a lesser degree by a government agency, or by a competitor. Therefore, trust engines operated by each of these other organizations may assign different levels of reliability to this enrollment.

Similarly, additional data which is submitted across a network, but which is authenticated by other trusted data provided during a previous enrollment with the same trust engine 110 may be considered as reliable as the original enrollment data was, even though the latter data were submitted across an open network. In such circumstances, a subsequent notarization will effectively increase the level of reliability associated with the original enrollment data. In

this way for example, an anonymous or pseudonymous enrollment may then be raised to a full enrollment by demonstrating to some enrollment official the identity of the individual matching the enrolled data.

The reliability factors discussed above are generally values which may be determined in advance of any particular authentication instance. This is because they are based upon the enrollment and the technique, rather than the actual authentication. In one embodiment, the step of generating reliability based upon these factors involves looking up previously determined values for this particular authentication technique and the enrollment data of the user. In a further aspect of an advantageous embodiment of the present invention, such reliabilities may be included with the enrollment data itself. In this way, these factors are automatically delivered to the authentication engine 215 along with the enrollment data sent from the depository 210.

While these factors may generally be determined in advance of any individual authentication instance, they still have an effect on each authentication instance which uses that particular technique of authentication for that user. Furthermore, although the values may change over time (e.g. if the user re-enrolls in a more reliable fashion), they are not dependent on the authentication data itself. By contrast, the reliability factors associated with a single specific instance's data may vary on each occasion. These factors, as discussed below, must be evaluated for each new authentication in order to generate reliability scores in STEP 1615.

The reliability of the authentication data reflects the match between the data provided by the user in a particular authentication instance and the data provided during the authentication enrollment. This is the fundamental question of whether the authentication data matches the enrollment data for the individual the user is claiming to be. Normally, when the data do not match, the user is considered to not be successfully authenticated, and the authentication fails. The manner in which this is evaluated may change depending on the authentication technique used. The comparison of such data is performed by the comparator 515 function of the authentication engine 215 as shown in FIGURE 5.

For instance, matches of passwords are generally evaluated in a binary fashion. In other words, a password is either a perfect match, or a failed match. It is usually not desirable to accept as even a partial match a password which is close to the correct password if it is not exactly correct. Therefore, when evaluating a password authentication, the reliability of the authentication returned by the comparator 515 is typically either 100% (correct) or 0% (wrong), with no possibility of intermediate values.

Similar rules to those for passwords are generally applied to token based authentication methods, such as smart cards. This is because having a smart card which has a similar identifier or which is similar to the correct one, is still just as wrong as having any other incorrect token. Therefore tokens tend also to be binary authenticators: a user either has the right token, or he doesn't.

However, certain types of authentication data, such as questionnaires and biometrics, are generally not binary authenticators. For example, a fingerprint may match a reference fingerprint to varying degrees. To some extent, this may be due to variations in the quality of the data captured either during the initial enrollment or in subsequent authentications. (A fingerprint may be smudged or a person may have a still healing scar or burn on a

particular finger.) In other instances the data may match less than perfectly because the information itself is somewhat variable and based upon pattern matching. (A voice analysis may seem close but not quite right because of background noise, or the acoustics of the environment in which the voice is recorded, or because the person has a cold.) Finally, in situations where large amounts of data are being compared, it may simply be the case that much of the data matches well, but some doesn't. (A ten-question questionnaire may have resulted in eight correct answers to personal questions, but two incorrect answers.) For any of these reasons, the match between the enrollment data and the data for a particular authentication instance may be desirably assigned a partial match value by the comparator 515. In this way, the fingerprint might be said to be a 85% match, the voice print a 65% match, and the questionnaire an 80% match, for example.

This measure (degree of match) produced by the comparator 515 is the factor representing the basic issue of whether an authentication is correct or not. However, as discussed above, this is only one of the factors which may be used in determining the reliability of a given authentication instance. Note also that even though a match to some partial degree may be determined, that ultimately, it may be desirable to provide a binary result based upon a partial match. In an alternate mode of operation, it is also possible to treat partial matches as binary, i.e. either perfect (100%) or failed (0%) matches, based upon whether or not the degree of match passes a particular threshold level of match. Such a process may be used to provide a simple pass/fail level of matching for systems which would otherwise produce partial matches.

Another factor to be considered in evaluating the reliability of a given authentication instance concerns the circumstances under which the authentication data for this particular instance are provided. As discussed above, the circumstances refer to the metadata associated with a particular authentication instance. This may include without limitation such information as: the network address of the authenticator, to the extent that it can be determined; the time of the authentication; the mode of transmission of the authentication data (phone line, cellular, network, etc.); and the serial number of the system of the authenticator.

These factors can be used to produce a profile of the type of authentication that is normally requested by the user. Then, this information can be used to assess reliability in at least two manners. One manner is to consider whether the user is requesting authentication in a manner which is consistent with the normal profile of authentication by this user. If the user normally makes authentication requests from one network address during business days (when she is at work) and from a different network address during evenings or weekends (when she is at home), an authentication which occurs from the home address during the business day is less reliable because it is outside the normal authentication profile. Similarly, if the user normally authenticates using a fingerprint biometric and in the evenings, an authentication which originates during the day using only a password is less reliable.

An additional way in which the circumstantial metadata can be used to evaluate the reliability of an instance of authentication is to determine how much corroboration the circumstance provides that the authenticator is the individual he claims to be. For instance, if the authentication comes from a system with a serial number known to be associated with the user, this is a good circumstantial indicator that the user is who they claim to be. Conversely, if

the authentication is coming from a network address which is known to be in Los Angeles when the user is known to reside in London, this is an indication that this authentication is less reliable based on its circumstances.

It is also possible that a cookie or other electronic data may be placed upon the system being used by a user when they interact with a vendor system or with the trust engine 110. This data is written to the storage of the system of the user and may contain an identification which may be read by a Web browser or other software on the user system. If this data is allowed to reside on the user system between sessions (a "persistent cookie"), it may be sent with the authentication data as further evidence of the past use of this system during authentication of a particular user. In effect, the metadata of a given instance, particularly a persistent cookie, may form a sort of token based authenticator itself.

Once the appropriate reliability factors based on the technique and data of the authentication instance are generated as described above in STEPS 1610 and 1615 respectively, they are used to produce an overall reliability for the authentication instance provided in STEP 1620. One means of doing this is simply to express each reliability as a percentage and then to multiply them together.

For example, suppose the authentication data is being sent in from a network address known to be the user's home computer completely in accordance with the user's past authentication profile (100%), and the technique being used is fingerprint identification (97%), and the initial finger print data was registered through the user's employer with the trust engine 110 (90%), and the match between the authentication data and the original fingerprint template in the enrollment data is very good (99%). The overall reliability of this authentication instance could then be calculated as the product of these reliabilities: 100% * 97% * 90% * 99% = 86.4% reliability.

This calculated reliability represents the reliability of one single instance of authentication. The overall reliability of a single authentication instance may also be calculated using techniques which treat the different reliability factors differently, for example by using formulas where different weights are assigned to each reliability factor. Furthermore, those of skill in the art will recognize that the actual values used may represent values other than percentages and may use non-arithmetic systems. One embodiment may include a module used by an authentication requestor to set the weights for each factor and the algorithms used in establishing the overall reliability of the authentication instance.

The authentication engine 215 may use the above techniques and variations thereof to determine the reliability of a single authentication instance, indicated as STEP 1620. However, it may be useful in many authentication situations for multiple authentication instances to be provided at the same time. For example, while attempting to authenticate himself using the system of the present invention, a user may provide a user identification, fingerprint authentication data, a smart card, and a password. In such a case, three independent authentication instances are being provided to the trust engine 110 for evaluation. Proceeding to STEP 1625, if the authentication engine 215 determines that the data provided by the user includes more than one authentication instance, then each instance in turn will be selected as shown in STEP 1630 and evaluated as described above in STEPS 1610, 1615 and 1620.

Note that many of the reliability factors discussed may vary from one of these instances to another. For instance, the inherent reliability of these techniques is likely to be different, as well as the degree of match provided between the authentication data and the enrollment data. Furthermore, the user may have provided enrollment data at different times and under different circumstances for each of these techniques, providing different enrollment reliabilities for each of these instances as well. Finally, even though the circumstances under which the data for each of these instances is being submitted is the same, the use of such techniques may each fit the profile of the user differently, and so may be assigned different circumstantial reliabilities. (For example, the user may normally use their password and fingerprint, but not their smart card.)

As a result, the final reliability for each of these authentication instances may be different from one another. However, by using multiple instances together, the overall confidence level for the authentication will tend to increase.

Once the authentication engine has performed STEPS 1610 through 1620 for all of the authentication instances provided in the authentication data, the reliability of each instance is used in STEP 1635 to evaluate the overall authentication confidence level. This process of combining the individual authentication instance reliabilities into the authentication confidence level may be modeled by various methods relating the individual reliabilities produced, and may also address the particular interaction between some of these authentication techniques. (For example, multiple knowledge-based systems such as passwords may produce less confidence than a single password and even a fairly weak biometric, such as a basic voice analysis.)

One means in which the authentication engine 215 may combine the reliabilities of multiple concurrent authentication instances to generate a final confidence level is to multiply the unreliability of each instance to arrive at a total unreliability. The unreliability is generally the complementary percentage of the reliability. For example, a technique which is 84% reliable is 16% unreliable. The three authentication instances described above (fingerprint, smart card, password) which produce reliabilities of 86%, 75%, and 72% would have corresponding unreliabilities of (100 - 86)%, (100 - 75)% and (100 - 72)%, or 14%, 25%, and 28%, respectively. By multiplying these unreliabilities, we get a cumulative unreliability of 14% * 25% * 28% = .98% unreliability, which corresponds to a reliability of 99.02%.

In an additional mode of operation, additional factors and heuristics 530 may be applied within the authentication engine 215 to account for the interdependence of various authentication techniques. For example, if someone has unauthorized access to a particular home computer, they probably have access to the phone line at that address as well. Therefore, authenticating based on an originating phone number as well as upon the serial number of the authenticating system does not add much to the overall confidence in the authentication. However, knowledge based authentication is largely independent of token based authentication (i.e. if someone steals your cellular phone or keys, they are no more likely to know your PIN or password than if they hadn't).

Furthermore, different vendors or other authentication requestors may wish to weigh different aspects of the authentication differently. This may include the use of separate weighing factors or algorithms used in calculating the

reliability of individual instances as well as the use of different means to evaluate authentication events with multiple instances.

For instance, vendors for certain types of transactions, for instance corporate email systems, may desire to authenticate primarily based upon heuristics and other circumstantial data by default. Therefore, they may apply high weights to factors related to the metadata and other profile related information associated with the circumstances surrounding authentication events. This arrangement could be used to ease the burden on users during normal operating hours, by not requiring more from the user than that he be logged on to the correct machine during business hours. However, another vendor may weigh authentications coming from a particular technique most heavily, for instance fingerprint matching, because of a policy decision that such a technique is most suited to authentication for the particular vendor's purposes.

Such varying weights may be defined by the authentication requestor in generating the authentication request and sent to the trust engine 110 with the authentication request in one mode of operation. Such options could also be set as preferences during an initial enrollment process for the authentication requestor and stored within the authentication engine in another mode of operation.

Once the authentication engine 215 produces an authentication confidence level for the authentication data provided, this confidence level is used to complete the authentication request in STEP 1640, and this information is forwarded from the authentication engine 215 to the transaction engine 205 for inclusion in a message to the authentication requestor.

The process described above is merely exemplary, and those of skill in the art will recognize that the steps need not be performed in the order shown. Furthermore, certain steps, such as the evaluation of the reliability of each authentication instance provided, may be carried out in parallel with one another if circumstances permit.

In a further aspect of this invention, a method is provided to accommodate conditions when the authentication confidence level produced by the process described above fails to meet the required trust level of the vendor or other party requiring the authentication. In circumstances such as these where a gap exists between the level of confidence provided and the level of trust desired, the operator of the trust engine 110 is in a position to provide opportunities for one or both parties to provide alternate data or requirements in order to close this trust gap. This process will be referred to as "trust arbitrage" herein.

Trust arbitrage may take place within a framework of cryptographic authentication as described above with reference to FIGURES 10 and 11. As shown therein, a vendor or other party will request authentication of a particular user in association with a particular transaction. In one circumstance, the vendor simply requests an authentication, either positive or negative, and after receiving appropriate data from the user, the trust engine 110 will provide such a binary authentication. In circumstances such as these, the degree of confidence required in order to secure a positive authentication is determined based upon preferences set within the trust engine 110.

However, it is also possible that the vendor may request a particular level of trust in order to complete a particular transaction. This required level may be included with the authentication request (e.g. authenticate this user

to 96% confidence) or may be determined by the trust engine 110 based on other factors associated with the transaction (i.e. authenticate this user as appropriate for this transaction). One such factor might be the economic value of the transaction. For transactions which have greater economic value, a higher degree of trust may be required. Similarly, for transactions with high degrees of risk a high degree of trust may be required. Conversely, for transactions which are either of low risk or of low value, lower trust levels may be required by the vendor or other authentication requestor.

The process of trust arbitrage occurs between the steps of the trust engine 110 receiving the authentication data in STEP 1050 of FIGURE 10 and the return of an authentication result to the vendor in STEP 1055 of FIGURE 10. Between these steps, the process which leads to the evaluation of trust levels and the potential trust arbitrage occurs as shown in FIGURE 17. In circumstances where simple binary authentication is performed, the process shown in FIGURE 17 reduces to having the transaction engine 205 directly compare the authentication data provided with the enrollment data for the identified user as discussed above with reference to FIGURE 10, flagging any difference as a negative authentication.

As shown in FIGURE 17, the first step after receiving the data in STEP 1050 is for the transaction engine 205 to determine the trust level which is required for a positive authentication for this particular transaction in STEP 1710. This step may be performed by one of several different methods. The required trust level may be specified to the trust engine 110 by the authentication requestor at the time when the authentication request is made. The authentication requestor may also set a preference in advance which is stored within the depository 210 or other storage which is accessible by the transaction engine 205. This preference may then be read and used each time an authentication request is made by this authentication requestor. The preference may also be associated with a particular user as a security measure such that a particular level of trust is always required in order to authenticate that user, the user preference being stored in the depository 210 or other storage media accessible by the transaction engine 205. The required level may also be derived by the transaction engine 205 or authentication engine 215 based upon information provided in the authentication request, such as the value and risk level of the transaction to be authenticated.

In one mode of operation, a policy management module or other software which is used when generating the authentication request is used to specify the required degree of trust for the authentication of the transaction. This may be used to provide a series of rules to follow when assigning the required level of trust based upon the policies which are specified within the policy management module. One advantageous mode of operation is for such a module to be incorporated with the web server of a vendor in order to appropriately determine required level of trust for transactions initiated with the vendor's web server. In this way, transaction requests from users may be assigned a required trust level in accordance with the policies of the vendor and such information may be forwarded to the trust engine 110 along with the authentication request.

This required trust level correlates with the degree of certainty that the vendor wants to have that the individual authenticating is in fact who he identifies himself as. For example, if the transaction is one where the

vendor wants a fair degree of certainty because goods are changing hands, the vendor may require a trust level of 85%. For situation where the vendor is merely authenticating the user to allow him to view members only content or exercise privileges on a chat room, the downside risk may be small enough that the vendor requires only a 60% trust level. However, to enter into a production contract with a value of tens of thousands of dollars, the vendor may require a trust level of 99% or more.

This required trust level represents a metric to which the user must authenticate himself in order to complete the transaction. If the required trust level is 85% for example, the user must provide authentication to the trust engine 110 sufficient for the trust engine 110 to say with 85% confidence that the user is who they say they are. It is the balance between this required trust level and the authentication confidence level which produces either a positive authentication (to the satisfaction of the vendor) or a possibility of trust arbitrage.

As shown in FIGURE 17, after the transaction engine 205 receives the required trust level, it compares in STEP 1720 the required trust level to the authentication confidence level which the authentication engine 215 calculated for the current authentication (as discussed with reference to FIGURE 16). If the authentication confidence level is higher than the required trust level for the transaction in STEP 1730, then the process moves to STEP 1740 where a positive authentication for this transaction is produced by the transaction engine 205. A message to this effect will then be inserted into the authentication results and returned to the vendor by the transaction engine 205 as shown in STEP 1055 (see FIGURE 10).

However, if the authentication confidence level does not fulfill the required trust level in STEP 1730, then a confidence gap exists for the current authentication, and trust arbitrage is conducted in STEP 1750. Trust arbitrage is described more completely with reference to FIGURE 18 below. This process as described below takes place within the transaction engine 205 of the trust engine 110. Because no authentication or other cryptographic operations are needed to execute trust arbitrage (other than those required for the SSL communication between the transaction engine 205 and other components), the process may be performed outside the authentication engine 215. However, as will be discussed below, any reevaluation of authentication data or other cryptographic or authentication events will require the transaction engine 205 to resubmit the appropriate data to the authentication engine 215. Those of skill in the art will recognize that the trust arbitrage process could alternately be structured to take place partially or entirely within the authentication engine 215 itself.

As mentioned above, trust arbitrage is a process where the trust engine 110 mediates a negotiation between the vendor and user in an attempt to secure a positive authentication where appropriate. As shown in STEP 1805, the transaction engine 205 first determines whether or not the current situation is appropriate for trust arbitrage. This may be determined based upon the circumstances of the authentication, e.g. whether this authentication has already been through multiple cycles of arbitrage, as well as upon the preferences of either the vendor or user, as will be discussed further below.

In such circumstances where arbitrage is not possible, the process proceeds to STEP 1810 where the transaction engine 205 generates a negative authentication and then inserts it into the authentication results which

are sent to the vendor in STEP 1055 (see FIGURE 10). One limit which may be advantageously used to prevent authentications from pending indefinitely is to set a time-out period from the initial authentication request. In this way, any transaction which is not positively authenticated within the time limit is denied further arbitrage and negatively authenticated. Those of skill in the art will recognize that such a time limit may vary depending upon the circumstances of the transaction and the desires of the user and vendor. Limitations may also be placed upon the number of attempts that may be made at providing a successful authentication. Such limitations may be handled by an attempt limiter 535 as shown in FIGURE 5.

If arbitrage is not prohibited in STEP 1805, the transaction engine 205 will then engage in negotiation with one or both of the transacting parties. The transaction engine 205 may send a message to the user requesting some form of additional authentication in order to boost the authentication confidence level produced as shown in STEP 1820. In the simplest form, this may simply indicates that authentication was insufficient. A request to produce one or more additional authentication instances to improve the overall confidence level of the authentication may also be sent.

If the user provides some additional authentication instances in STEP 1825, then the transaction engine 205 adds these authentication instances to the authentication data for the transaction and forwards it to the authentication engine 215 as shown in STEP 1015 (see FIGURE 10), and the authentication is reevaluated based upon both the pre-existing authentication instances for this transaction and the newly provided authentication instances.

An additional type of authentication may be a request from the trust engine 110 to make some form of person-to-person contact between the trust engine 110 operator (or a trusted associate) and the user, for example, by phone call. This phone call or other non-computer authentication can be used to provide personal contact with the individual and also to conduct some form of questionnaire based authentication. This also may give the opportunity to verify an originating telephone number and potentially a voice analysis of the user when he calls in. Even if no additional authentication data can be provided, the additional context associated with the user's phone number may improve the reliability of the authentication context. Any revised data or circumstances based upon this phone call are fed into the trust engine 110 for use in consideration of the authentication request.

Additionally, in STEP 1820 the trust engine 110 may provide an opportunity for the user to purchase insurance, effectively buying a more confident authentication. The operator of the trust engine 110 may, at times, only want to make such an option available if the confidence level of the authentication is above a certain threshold to begin with. In effect, this user side insurance is a way for the trust engine 110 to vouch for the user when the authentication meets the normal required trust level of the trust engine 110 for authentication, but does not meet the required trust level of the vendor for this transaction. In this way, the user may still successfully authenticate to a very high level as may be required by the vendor, even though he only has authentication instances which produce confidence sufficient for the trust engine 110.

This function of the trust engine 110 allows the trust engine 110 to vouch for someone who is authenticated to the satisfaction of the trust engine 110, but not of the vendor. This is analogous to the function performed by a

notary in adding his signature to a document in order to indicate to someone reading the document at a later time that the person whose signature appears on the document is in fact the person who signed it. The signature of the notary testifies to the act of signing by the user. In the same way, the trust engine is providing an indication that the person transacting is who they say they are.

However, because the trust engine 110 is artificially boosting the level of confidence provided by the user, there is a greater risk to the trust engine 110 operator, since the user is not actually meeting the required trust level of the vendor. The cost of the insurance is designed to offset the risk of a false positive authentication to the trust engine 110 (who may be effectively notarizing the authentications of the user). The user pays the trust engine 110 operator to take the risk of authenticating to a higher level of confidence than has actually been provided.

Because such an insurance system allows someone to effectively buy a higher confidence rating from the trust engine 110, both vendors and users may wish to prevent the use of user side insurance in certain transactions. Vendors may wish to limit positive authentications to circumstances where they know that actual authentication data supports the degree of confidence which they require and so may indicate to the trust engine 110 that user side insurance is not to be allowed. Similarly, to protect his online identity, a user may wish to prevent the use of user side insurance on his account, or may wish to limit its use to situations where the authentication confidence level without the insurance is higher than a certain limit. This may be used as a security measure to prevent someone from overhearing a password or stealing a smart card and using them to falsely authenticate to a low level of confidence, and then purchasing insurance to produce a very high level of (false) confidence. These factors may be evaluated in determining whether user side insurance is allowed.

If user purchases insurance in STEP 1840, then the authentication confidence level is adjusted based upon the insurance purchased in STEP 1845, and the authentication confidence level and required trust level are again compared in STEP 1730 (see FIGURE 17). The process continues from there, and may lead to either a positive authentication in STEP 1740 (see FIGURE 17), or back into the trust arbitrage process in STEP 1750 for either further arbitrage (if allowed) or a negative authentication in STEP 1810 if further arbitrage is prohibited.

In addition to sending a message to the user in STEP 1820, the transaction engine 205 may also send a message to the vendor in STEP 1830 which indicates that a pending authentication is currently below the required trust level. The message may also offer various options on how to proceed to the vendor. One of these options is to simply inform the vendor of what the current authentication confidence level is and ask if the vendor wishes to maintain their current unfulfilled required trust level. This may be beneficial because in some cases, the vendor may have independent means for authenticating the transaction or may have been using a default set of requirements which generally result in a higher required level being initially specified than is actually needed for the particular transaction at hand.

For instance, it may be standard practice that all incoming purchase order transactions with the vendor are expected to meet a 98% trust level. However, if an order was recently discussed by phone between the vendor and a long-standing customer, and immediately thereafter the transaction is authenticated, but only to a 93% confidence

level, the vendor may wish to simply lower the acceptance threshold for this transaction, because the phone call effectively provides additional authentication to the vendor. In certain circumstances, the vendor may be willing to lower their required trust level, but not all the way to the level of the current authentication confidence. For instance, the vendor in the above example might consider that the phone call prior to the order might merit a 4% reduction in the degree of trust needed; however, this is still greater than the 93% confidence produced by the user.

If the vendor does adjust their required trust level in STEP 1835, then the authentication confidence level produced by the authentication and the required trust level are compared in STEP 1730 (see FIGURE 17). If the confidence level now exceeds the required trust level, a positive authentication may be generated in the transaction engine 205 in STEP 1740 (see FIGURE 17). If not, further arbitrage may be attempted as discussed above if it is permitted.

In addition to requesting an adjustment to the required trust level, the transaction engine 205 may also offer vendor side insurance to the vendor requesting the authentication. This insurance serves a similar purpose to that described above for the user side insurance. Here, however, rather than the cost corresponding to the risk being taken by the trust engine 110 in authenticating above the actual authentication confidence level produced, the cost of the insurance corresponds to the risk being taken by the vendor in accepting a lower trust level in the authentication.

Instead of just lowering their actual required trust level, the vendor has the option of purchasing insurance to protect itself from the additional risk associated with a lower level of trust in the authentication of the user. As described above, it may be advantageous for the vendor to only consider purchasing such insurance to cover the trust gap in conditions where the existing authentication is already above a certain threshold.

The availability of such vendor side insurance allows the vendor the option to either: lower his trust requirement directly at no additional cost to himself, bearing the risk of a false authentication himself (based on the lower trust level required); or, buying insurance for the trust gap between the authentication confidence level and his requirement, with the trust engine 110 operator bearing the risk of the lower confidence level which has been provided. By purchasing the insurance, the vendor effectively keeps his high trust level requirement, because the risk of a false authentication is shifted to the trust engine 110 operator.

If the vendor purchases insurance in STEP 1840, the authentication confidence level and required trust levels are compared in STEP 1730 (see FIGURE 17), and the process continues as described above.

Note that it is also possible that both the user and the vendor respond to messages from the trust engine 110. Those of skill in the art will recognize that there are multiple ways in which such situations can be handled. One advantageous mode of handling the possibility of multiple responses is simply to treat the responses in a first-come, first-served manner. For example, if the vendor responds with a lowered required trust level and immediately thereafter the user also purchases insurance to raise his authentication level, the authentication is first reevaluated based upon the lowered trust requirement from the vendor. If the authentication is now positive, the user's insurance purchase is ignored. In another advantageous mode of operation, the user might only be charged for the level of

insurance required to meet the new, lowered trust requirement of the vendor (if a trust gap remained even with the lowered vendor trust requirement).

If no response from either party is received during the trust arbitrage process at STEP 1850 within the time limit set for the authentication, the arbitrage is reevaluated in STEP 1805. This effectively begins the arbitrage process again. If the time limit was final or other circumstances prevent further arbitrage in STEP 1805, a negative authentication is generated by the transaction engine 205 in STEP 1810 and returned to the vendor in STEP 1055 (see FIGURE 10). If not, new messages may be sent to the user and vendor, and the process may be repeated as desired.

Note that for certain types of transactions, for instance, digitally signing documents which are not part of a transaction, there may not necessarily be a vendor or other third party; therefore the transaction is primarily between the user and the trust engine 110. In circumstances such as these, the trust engine 110 will have its own required trust level which must be satisfied in order to generate a positive authentication. However, in such circumstances, it will often not be desirable for the trust engine 110 to offer insurance to the user in order for him to raise the confidence of his own signature.

The process described above and shown in FIGURES 16 - 18 may be carried out using various communications modes as described above with reference to the trust engine 110. For instance, the messages may be web-based and sent using SSL connections between the trust engine 110 and applets downloaded in real time to browsers running on the user or vendor systems. In an alternate mode of operation, certain dedicated applications may be in use by the user and vendor which facilitate such arbitrage and insurance transactions. In another alternate mode of operation, secure email operations may be used to mediate the arbitrage described above, thereby allowing deferred evaluations and batch processing of authentications. Those of skill in the art will recognize that different communications modes may be used as are appropriate for the circumstances and authentication requirements of the vendor.

The following description with reference to FIGURE 19 describes a sample transaction which integrates the various aspects of the present invention as described above. This example illustrates the overall process between a user and a vendor as mediates by the trust engine 110. Although the various steps and components as described in detail above may be used to carry out the following transaction, the process illustrated focuses on the interaction between the trust engine 110, user and vendor.

The transaction begins when the user, while viewing web pages online, fills out an order form on the web site of the vendor in STEP 1900. The user wishes to submit this order form to the vendor, signed with his digital signature. In order to do this, the user submits the order form with his request for a signature to the trust engine 110 in STEP 1905. The user will also provide authentication data which will be used as described above to authenticate his identity.

In STEP 1910 the authentication data is compared to the enrollment data by the trust engine 110 as discussed above, and if a positive authentication is produced, the hash of the order form, signed with the private key of the user, is forwarded to the vendor along with the order form itself.

The vendor receives the signed form in STEP 1915, and then the vendor will generate an invoice or other contract related to the purchase to be made in STEP 1920. This contract is sent back to the user with a request for a signature in STEP 1925. The vendor also sends an authentication request for this contract transaction to the trust engine 110 in STEP 1930 including a hash of the contract which will be signed by both parties. To allow the contract to be digitally signed by both parties, the vendor also includes authentication data for itself so that the vendor's signature upon the contract can later be verified if necessary.

As discussed above, the trust engine 110 then verifies the authentication data provided by the vendor to confirm the vendor's identity, and if the data produces a positive authentication in STEP 1935, continues with STEP 1955 when the data is received from the user. If the vendor's authentication data does not match the enrollment data of the vendor to the desired degree, a message is returned to the vendor requesting further authentication. Trust arbitrage may be performed here if necessary, as described above, in order for the vendor to successfully authenticate itself to the trust engine 110.

When the user receives the contract in STEP 1940, he reviews it, generates authentication data to sign it if it is acceptable in STEP 1945, and then sends a hash of the contract and his authentication data to the trust engine 110 in STEP 1950. The trust engine 110 verifies the authentication data in STEP 1955 and if the authentication is good, proceeds to process the contract as described below. As discussed above with reference to FIGURES 17 and 18, trust arbitrage may be performed as appropriate to close any trust gap which exists between the authentication confidence level and the required authentication level for the transaction.

The trust engine 110 signs the hash of the contract with the user's private key, and sends this signed hash to the vendor in STEP 1960, signing the complete message on its own behalf, i.e. including a hash of the complete message (including the user's signature) encrypted with the private key 510 of the trust engine 110. This message is received by the vendor in STEP 1965. The message represents a signed contract (hash of contract encrypted using user's private key) and a receipt from the trust engine 110 (the hash of the message including the signed contract, encrypted using the trust engine 110's private key).

The trust engine 110 similarly prepares a hash of the contract with the vendor's private key in STEP 1970, and forwards this to the user, signed by the trust engine 110. In this way, the user also receives a copy of the contract, signed by the vendor, as well as a receipt, signed by the trust engine 110, for delivery of the signed contract in STEP 1975.

In addition to the foregoing, an additional aspect of the invention provides a cryptographic Service Provider Module (SPM) which may be available to a client side application as a means to access functions provided by the trust engine 110 described above. One advantageous way to provide such a service is for the cryptographic SPM is to mediate communications between a third party Application Programming Interface (API) and a trust engine 110 which is accessible via a network or other remote connection. A sample cryptographic SPM is described below with reference to FIGURE 20.

For example, on a typical system, a number of API's are available to programmers. Each API provides a set of function calls which may be made by an application 2000 running upon the system. Examples of API's which provide programming interfaces suitable for cryptographic functions, authentication functions, and other security function include the Cryptographic API (CAPI) 2010 provided by Microsoft with its Windows operating systems, and the Common Data Security Architecture (CDSA), sponsored by IBM, Intel and other members of the Open Group. CAPI will be used as an exemplary security API in the discussion that follows. However, the cryptographic SPM described could be used with CDSA or other security API's as are known in the art.

This API is used by a user system 105 or vendor system 120 when a call is made for a cryptographic function. Included among these functions may be requests associated with performing various cryptographic operations, such as encrypting a document with a particular key, signing a document, requesting a digital certificate, verifying a signature upon a signed document, and such other cryptographic functions as are described above or known to those of skill in the art.

Such cryptographic functions are normally performed locally to the system upon which CAPI 2010 is located. This is because generally the functions called require the use of either resources of the local user system 105, such as a fingerprint reader, or software functions which are programmed using libraries which are executed on the local machine. Access to these local resources is normally provided by one or more Service Provider Modules (SPM's) 2015, 2020 as referred to above which provide resources with which the cryptographic functions are carried out. Such SPM's may include software libraries 2015 to perform encrypting or decrypting operations, or drivers and applications 2020 which are capable of accessing specialized hardware 2025, such as biometric scanning devices. In much the way that CAPI 2010 provides functions which may be used by an application 2000 of the system 105, the SPM's 2015, 2020 provide CAPI with access to the lower level functions and resources associated with the available services upon the system.

In accordance with the invention, it is possible to provide a cryptographic SPM 2030 which is capable of accessing the cryptographic functions provided by the trust engine 110 and making these functions available to an application 2000 through CAPI 2010. Unlike embodiments where CAPI 2010 is only able to access resources which are locally available through SPM's 2015, 2020, a cryptographic SPM 2030 as described herein would be able to submit requests for cryptographic operations to a remotely-located, network-accessible trust engine 110 in order to perform the operations desired.

For instance, if an application 2000 has a need for a cryptographic operation, such as signing a document, the application 2000 makes a function call to the appropriate CAPI 2010 function. CAPI 2010 in turn will execute this function, making use of the resources which are made available to it by the SPM's 2015, 2020 and the cryptographic SPM 2030. In the case of a digital signature function, the cryptographic SPM 2030 will generate an appropriate request which will be sent to the trust engine 110 across the communication link 125.

The operations which occur between the cryptographic SPM 2030 and the trust engine 110 are the same operations that would be possible between any other system and the trust engine 110. However, these functions are

effectively made available to a user system 105 through CAPI 2010 such that they appear to be locally available upon the user system 105 itself. However, unlike ordinary SPM's 2015, 2020, the functions are being carried out on the remote trust engine 110 and the results relayed to the cryptographic SPM 2030 in response to appropriate requests across the communication link 125.

This cryptographic SPM 2030 makes a number of operations available to the user system 105 or a vendor system 120 which might not otherwise be available. These functions include without limitation: encryption and decryption of documents; issuance of digital certificates; digital signing of documents; verification of digital signatures; and such other operations as will be apparent to those of skill in the art.

Additionally, other combinations, admissions, substitutions and modifications will be apparent to the skilled artisan in view of the disclosure herein. Accordingly, the present invention is not intended to be limited by the reaction of the preferred embodiments but is to be defined by a reference to the appended claims.

WHAT IS CLAIMED IS:

1. A method of allocating the risk during an electronic transaction when a confidence gap exists between an authentication confidence level and a desired confidence level, the method comprising:

determining that an authentication confidence level fails to meet a desired confidence level; and

offering insurance to at least one party of an electronic transaction when the authentication confidence level associated with the electronic transaction fails to meet a desired confidence level.

2. A method as in Claim 1 wherein the authentication confidence level is stored by a third party to the transaction.

3. A method as in Claim 2 wherein the offer of insurance is made by the third party.

4. A method as in Claim 1 wherein the insurance is purchased by one party of the electronic transaction to raise the authentication confidence level associated with the electronic transaction.

5. A method as in Claim 1 wherein the insurance is purchased by one party of the electronic transaction to lower the desired confidence level.

6. A method as in Claim 1 wherein responses are handled in the order in which they were received when more than one party of the electronic transaction responds to the offer of insurance.

7. A method as in Claim 1 wherein the authentication confidence level is determined based upon user data received from a first party to the electronic transaction, and the desired confidence level is determined based upon transaction data received from a second party to the electronic transaction.

8. A method as in Claim 7 wherein insurance is offered to the first party in order to increase the authentication confidence level.

9. A method as in Claim 8 wherein the cost of the insurance is based upon the additional risk in providing a higher authentication confidence level for this transaction.

10. A method as in Claim 7, wherein insurance is offered to the second party in order to decrease the desired confidence level.

11. A method as in Claim 10 wherein the cost of the insurance is based upon the additional risk in accepting a lower desired confidence level for this transaction.

12. An apparatus for authenticating an electronic transaction between a first party and a second party, the apparatus comprising a trust engine which receives authentication data from a first party to the electronic transaction and a desired confidence level from a second party to the electronic transaction, and wherein the trust engine determines an authentication confidence level based on the authentication data, and invites the second party to reduce the desired confidence level if the desired confidence level is higher than the authentication confidence level.

13. An apparatus as in Claim 12 wherein the trust engine is connected to the first party via a communication link.

14. An apparatus as in Claim 13 wherein the communication link comprises a computer network.

15. An apparatus as in Claim 12 wherein the trust engine also offers insurance to the second party if the second party does not reduce the desired confidence level.

16. An apparatus as in Claim 12 wherein the trust engine provides an assurance of the first user's identity to the second user.

17. An apparatus as in Claim 12 wherein the electronic transaction represents the sale of goods from a vendor to a purchaser and wherein the first party represents the purchaser and wherein the second party represents the vendor.

18. A method of facilitating an electronic transaction when a confidence gap exists between an authentication confidence level and a desired confidence level, the method comprising:

receiving authentication data at a third party to the electronic transaction, the authentication data associated with an electronic transaction wherein a vendor specifies a desired confidence level and a user submits the authentication data to produce an authentication confidence level;

determining at the third party that the authentication confidence level fails to meet the desired confidence level; and

inviting the vendor to reduce the desired confidence level with a third party.

19. A method as in Claim 18 further comprising allowing the use of the user's private key in response to the vendor's reduction of the desired confidence level.

20. A method as in Claim 19 wherein the authentication data received by the third party is not made available to the vendor.

21. A method as in Claim 18 wherein the electronic transaction involves the acceptance of an offer for sale of goods or services.

22. A method as in Claim 18 wherein the electronic transaction involves the vendor providing access to a service.

23. A method as in Claim 18 wherein the electronic transaction involves providing an electronic record of an agreement between the vendor and the user.

24. A method as in Claim 18 wherein the vendor comprises an authentication authority.

25. A method as in Claim 18 further comprising offering the vendor insurance in order to decrease the desired confidence level.

26. A method as in Claim 18 wherein an authentication authority provides an assurance to the vendor of the user's identity.

27. A method as in Claim 26 wherein the authentication authority contacts the vendor personally in order to establish the desired level of trust.

28. A method as in Claim 26 wherein the authentication authority signs a transaction record using its private key in assuring the vendor of the user's identity.

29. A method as in Claim 26 wherein the authentication authority signs a transaction record using the user's private key in assuring the vendor of the user's identity.

30. A method as in Claim 26 wherein the authentication authority signs a transaction record using both its own private key and the user's private key in assuring the vendor of the user's identity.

31. A system for facilitating an electronic transaction between a merchant and a user, the system comprising a trust engine which receives authentication data from a user and a desired confidence level from a merchant, and wherein the trust engine determines an authentication confidence level based on the authentication

data, and invites the user to provide additional authentication data if the desired confidence level is higher than the authentication confidence level, the trust engine using a private cryptographic key of the user to cryptographically sign a record of the transaction.

32. A system as in Claim 31 wherein the trust engine offers insurance to the user in order to raise the authentication confidence level.

33. A system as in Claim 32 wherein the trust engine bears the risk associated with the gap between the authentication confidence level and the desired confidence level.

34. A system as in Claim 31 wherein the electronic transaction represents access to an electronic service provided by the merchant.

35. A method of facilitating an electronic transaction when a confidence gap exists between an authentication confidence level and a desired confidence level, the method comprising:

receiving authentication data associated with an electronic transaction wherein a vendor specifies a desired confidence level and a user submits the authentication data to produce an authentication confidence level;

determining that the authentication confidence level fails to meet the desired confidence level; and

inviting the user to provide additional authentication data.

36. A method as in Claim 35 further comprising allowing the use of the user's private key if the additional authentication data increases the authentication confidence level to meet the desired confidence level.

37. A method as in Claim 35 wherein the authentication data is received by a third party to the transaction in order to maintain security of the authentication data.

38. A method as in Claim 37 wherein the third party comprises an authentication authority who provides an assurance of the user's identity to the vendor.

39. A method as in Claim 38 wherein the authentication authority contacts the user personally in order to generate additional authentication data.

40. A method as in Claim 39 wherein the additional authentication data comprises additional circumstantial data regarding the generation of the authentication data.

41. A method as in Claim 39 wherein the additional authentication data comprises data generated using authentication techniques involving person-to-person contact.

42. A method as in Claim 35 wherein the additional authentication data comprises data generated using authentication techniques other than those used in the initially submitted authentication data.

43. A method as in Claim 35 further comprising offering the user insurance in order to increase the authentication confidence level.

44. A method as in Claim 43 wherein the user may be prevented from buying insurance at the discretion of the vendor.

45. A method of facilitating an electronic transaction when a confidence gap exists between an authentication confidence level and a desired confidence level, the method comprising:

determining that an authentication confidence level fails to meet a desired confidence level for a transaction involving a first entity that specifies the desired confidence level and a second entity that submits authentication data to produce an authentication confidence level; and

inviting the first entity to reduce the desired confidence level.

46. A method as in Claim 45 wherein the desired confidence level is specified in a stored preference set by the first entity.

47. A method as in Claim 45 wherein the desired confidence level is associated with the identity of the second entity.

48. A method as in Claim 45 wherein the first entity bears the risk associated with the confidence gap.

49. A method as in Claim 45 wherein an insurer bears the risk associated with the confidence gap.

50. A method as in Claim 49 wherein the insurer comprises an authentication authority that provides an assurance of the second party's identity to the first party.

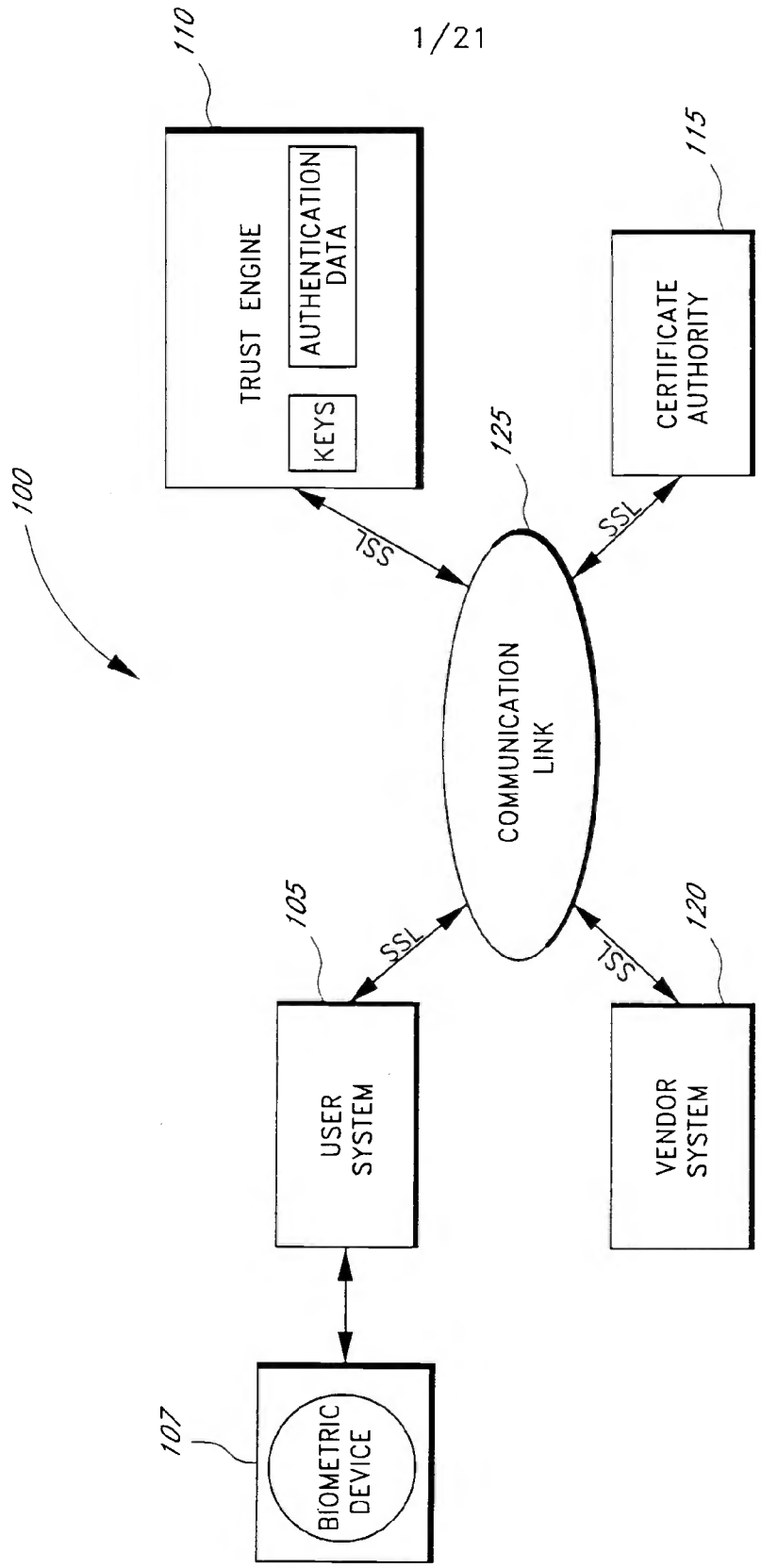

110
TRUST ENGINE
AUTHENTICATION DATA
KEYS
115
CERTIFICATE AUTHORITY
100
125
SSL
SSL
COMMUNICATION LINK
105
SSL
SSL
120
USER SYSTEM
VENDOR SYSTEM
107
BIOMETRIC DEVICE

FIG. 1

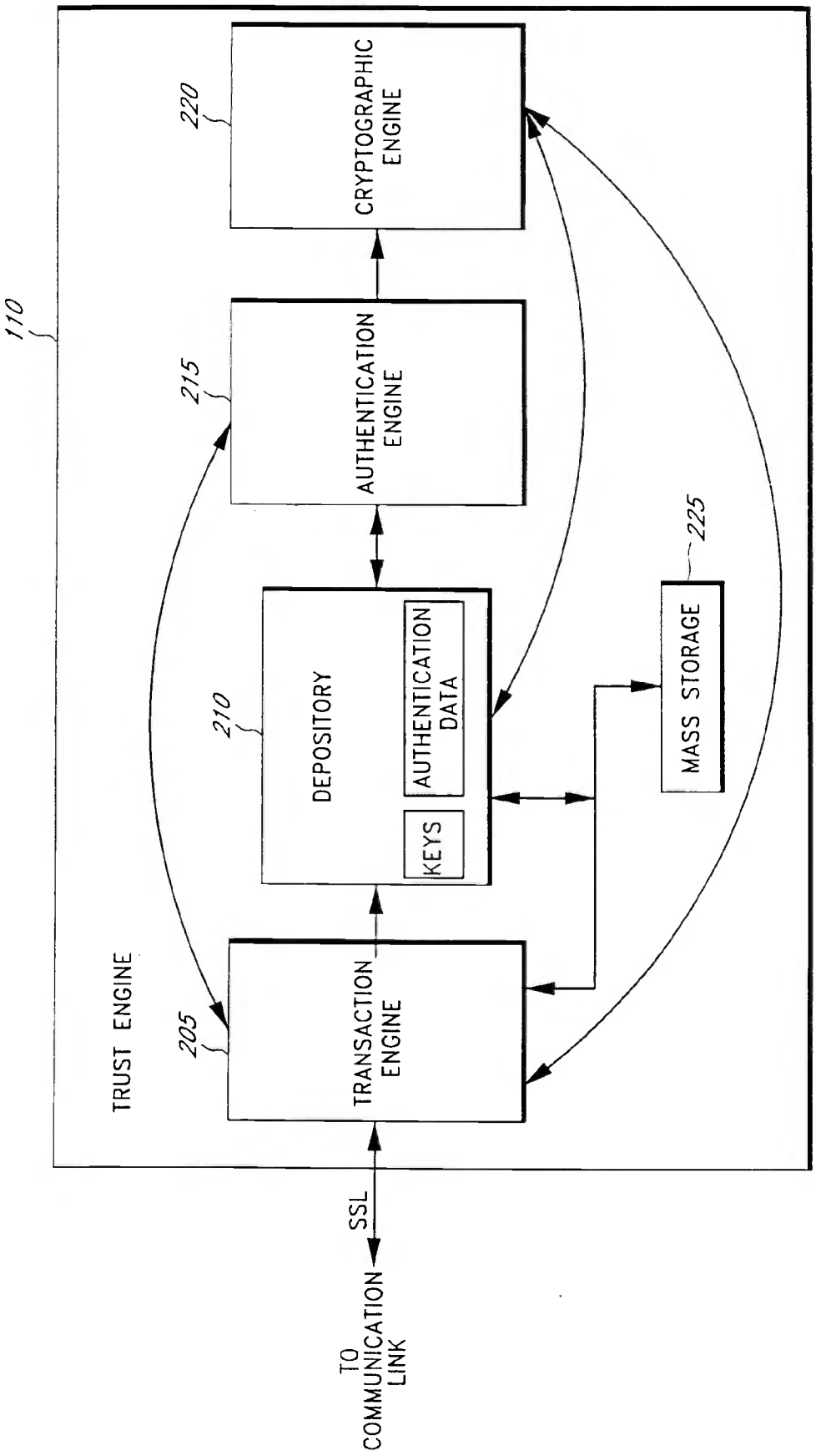

110
TRUST ENGINE
220
CRYPTOGRAPHIC ENGINE
215
AUTHENTICATION ENGINE
210
DEPOSITORY
AUTHENTICATION DATA
KEYS
225
MASS STORAGE
205
TRANSACTION ENGINE
SSL
TO COMMUNICATION LINK

FIG. 2

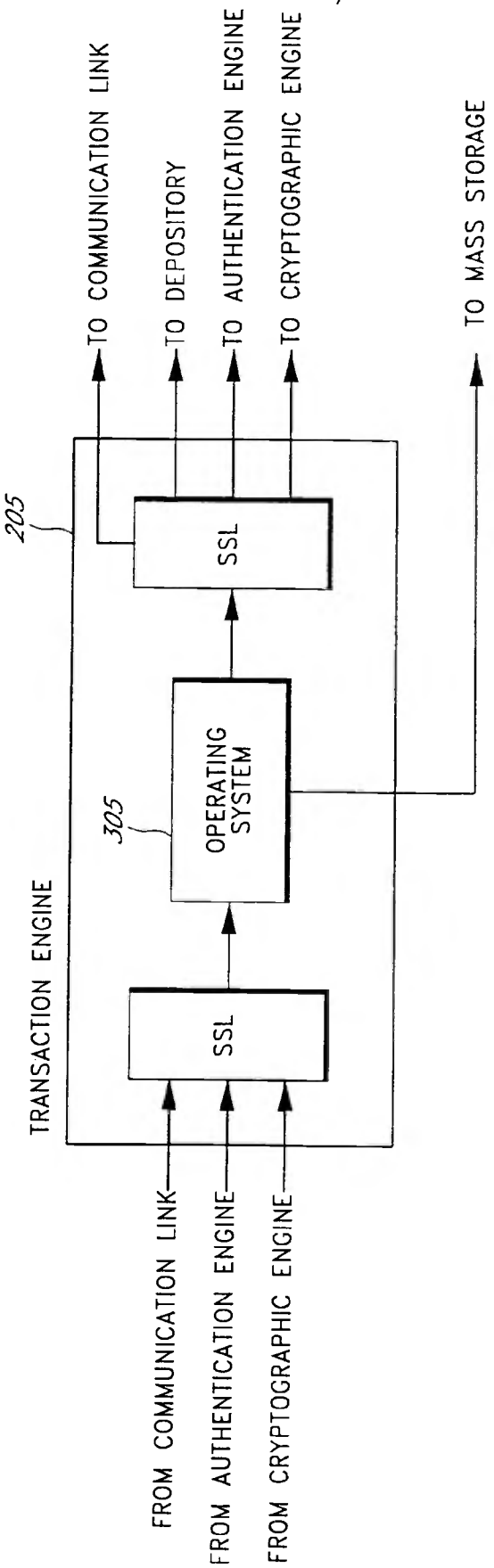

TO COMMUNICATION LINK
TO DEPOSITORY
TO AUTHENTICATION ENGINE
TO CRYPTOGRAPHIC ENGINE
TO MASS STORAGE
205
SSL
OPERATING SYSTEM
305
SSL
TRANSACTION ENGINE
FROM COMMUNICATION LINK
FROM AUTHENTICATION ENGINE
FROM CRYPTOGRAPHIC ENGINE

FIG. 3

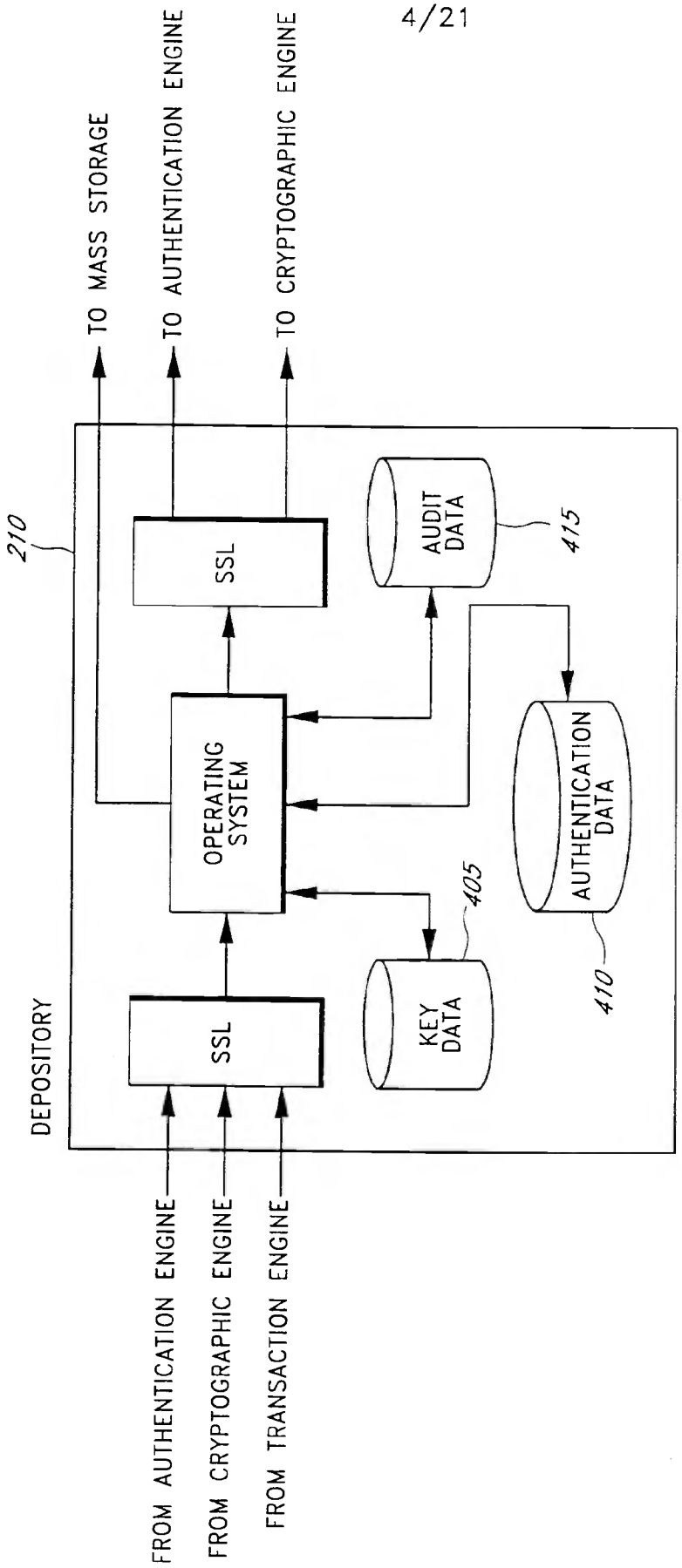

TO MASS STORAGE
TO AUTHENTICATION ENGINE
TO CRYPTOGRAPHIC ENGINE
210
SSL
AUDIT DATA
415
OPERATING SYSTEM
AUTHENTICATION DATA
405
KEY DATA
410
SSL
DEPOSITORY
FROM AUTHENTICATION ENGINE
FROM CRYPTOGRAPHIC ENGINE
FROM TRANSACTION ENGINE

FIG. 4

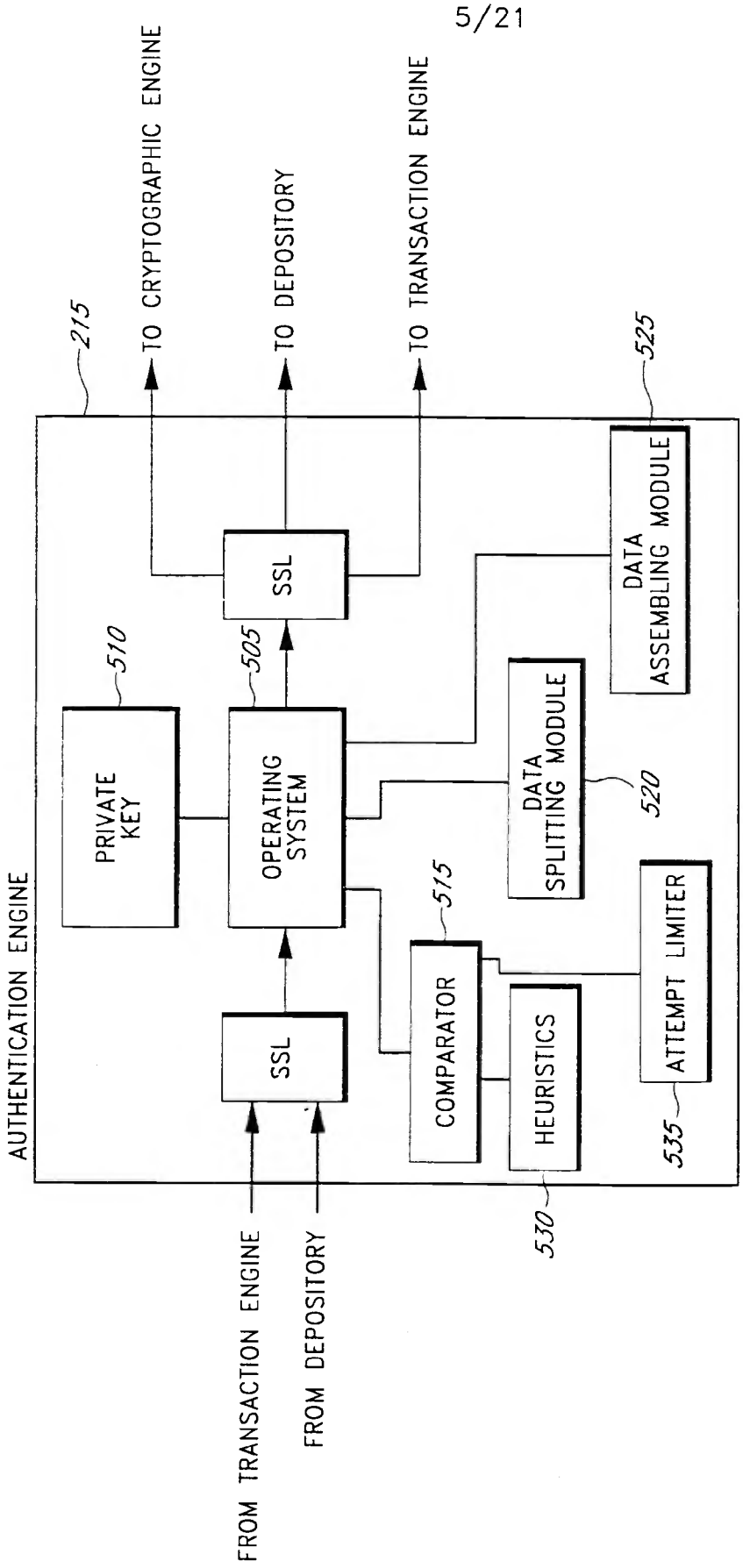

AUTHENTICATION ENGINE
215
TO CRYPTOGRAPHIC ENGINE
TO DEPOSITORY
TO TRANSACTION ENGINE
525
DATA ASSEMBLING MODULE
SSL
505
510
PRIVATE KEY
OPERATING SYSTEM
DATA SPLITTING MODULE
520
515
COMPARATOR
HEURISTICS
ATTEMPT LIMITER
535
530
SSL
FROM TRANSACTION ENGINE
FROM DEPOSITORY

FIG. 5

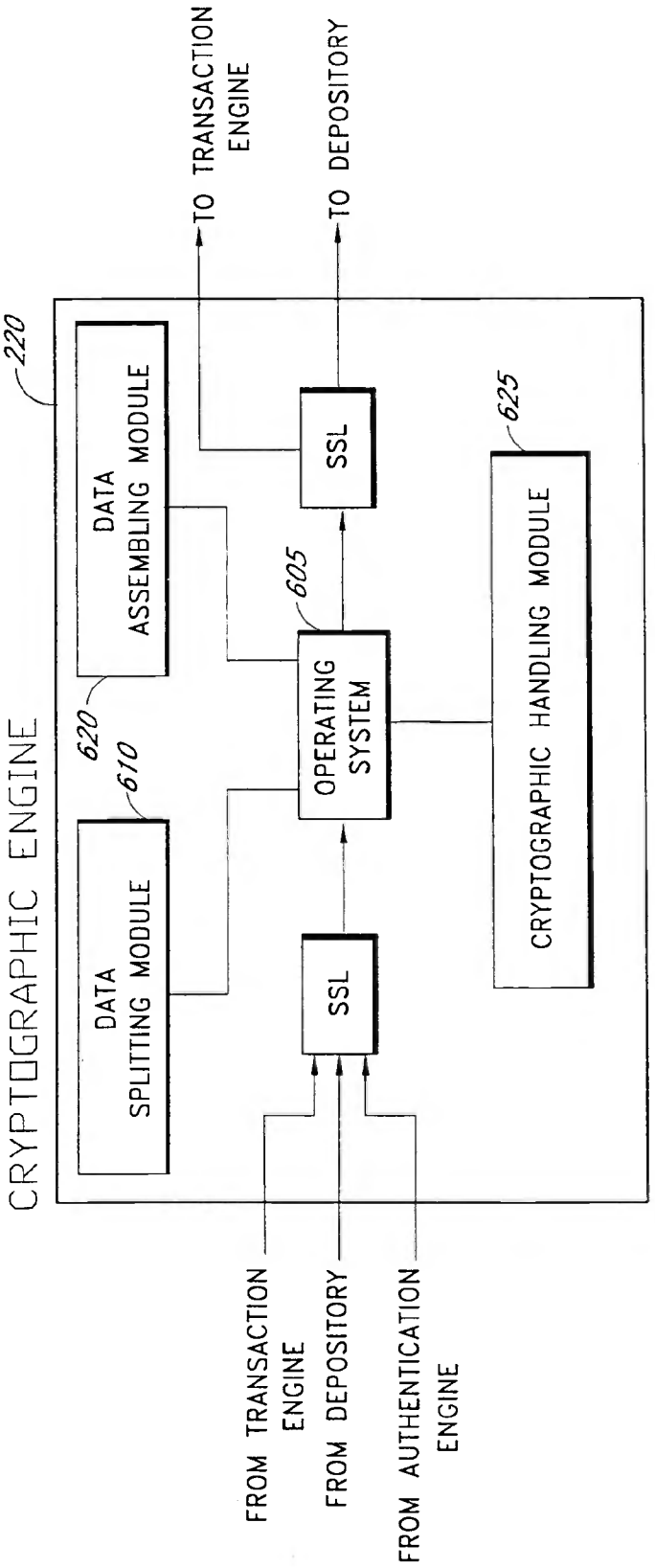

TO TRANSACTION ENGINE
TO DEPOSITORY
220
CRYPTOGRAPHIC ENGINE
DATA ASSEMBLING MODULE
620
610
DATA SPLITTING MODULE
605
OPERATING SYSTEM
SSL
SSL
625
CRYPTOGRAPHIC HANDLING MODULE
FROM TRANSACTION ENGINE
FROM DEPOSITORY
FROM AUTHENTICATION ENGINE

FIG. 6

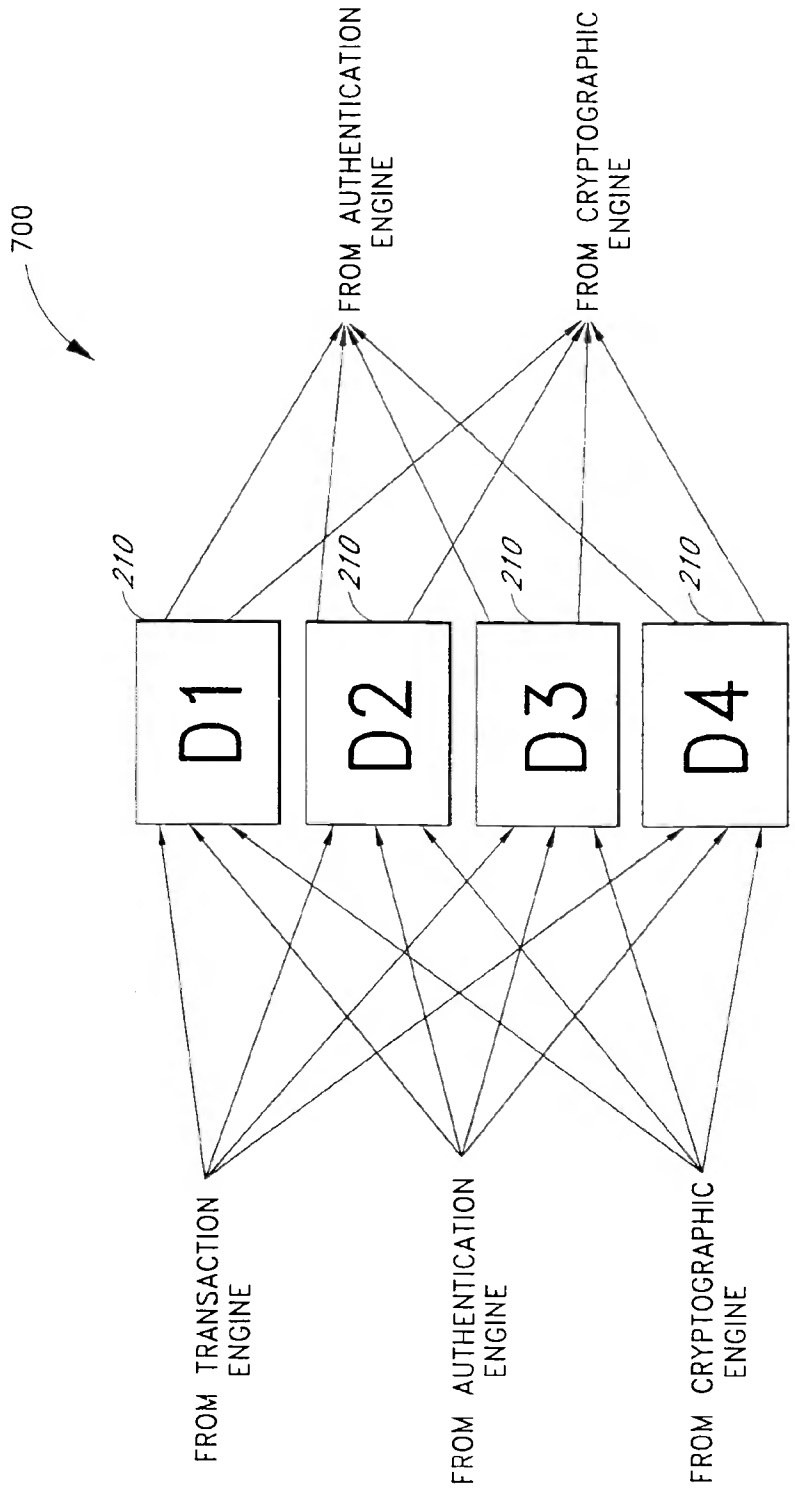

700
FROM AUTHENTICATION ENGINE
FROM CRYPTOGRAPHIC ENGINE
210
210
210
210
D1
D2
D3
D4
FROM TRANSACTION ENGINE
FROM AUTHENTICATION ENGINE
FROM CRYPTOGRAPHIC ENGINE

FIG. 7

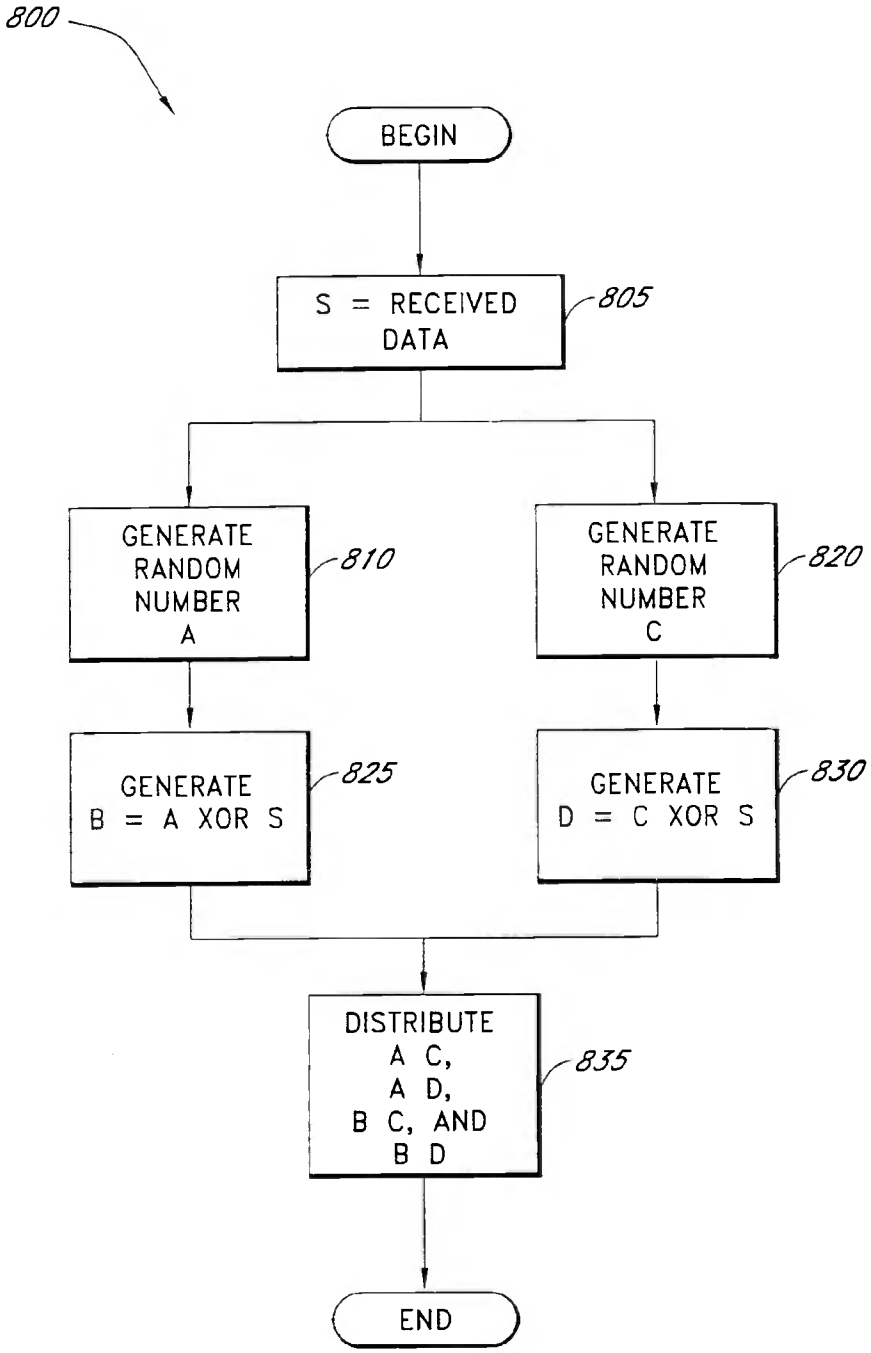
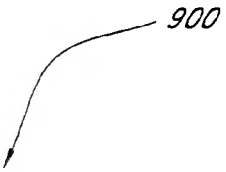

800
BEGIN
S = RECEIVED DATA
805
GENERATE RANDOM NUMBER A
810
GENERATE RANDOM NUMBER C
820
GENERATE B = A XOR S
825
GENERATE D = C XOR S
830
DISTRIBUTE A C, A D, B C, AND B D
835
END

FIG. 8



	ENROLLMENT DATA FLOW			
	SEND	RECEIVE	SSL	ACTION
905	USER	TRANSACTION ENGINE(TE)	1/2	TRANSMIT ENROLLMENT AUTHENTICATION DATA (B) AND THE USER ID (UID) ENCRYPTED WITH THE PUBLIC KEY OF THE AUTHENTICATION ENGINE (AE) AS (PUB_AE(UID,B))
915	TE	AE	FULL	FORWARD TRANSMISSION
920				AE DECRYPTS AND SPLITS FORWARDED DATA
925	AE	THE Xth DEPOSITORY (DX)	FULL	STORE RESPECTIVE PORTION OF DATA
	WHEN DIGITAL CERTIFICATE REQUESTED			
930	AE	CRYPTOGRAPHIC ENGINE (CE)	FULL	REQUEST KEY GENERATION
935				CE GENERATES AND SPLITS KEY
945	CE	TE	FULL	TRANSMIT REQUEST FOR DIGITAL CERTIFICATE
950	TE	CERTIFICATION AUTHORITY (CA)	1/2	TRANSMIT REQUEST
955	CA	TE	1/2	TRANSMIT DIGITAL CERTIFICATE
960	TE	USER	1/2	TRANSMIT DIGITAL CERTIFICATE
	TE	MS	FULL	STORE DIGITAL CERTIFICATE
965	TE	DX	FULL	STORE RESPECTIVE PORTION OF KEY

FIG. 9A

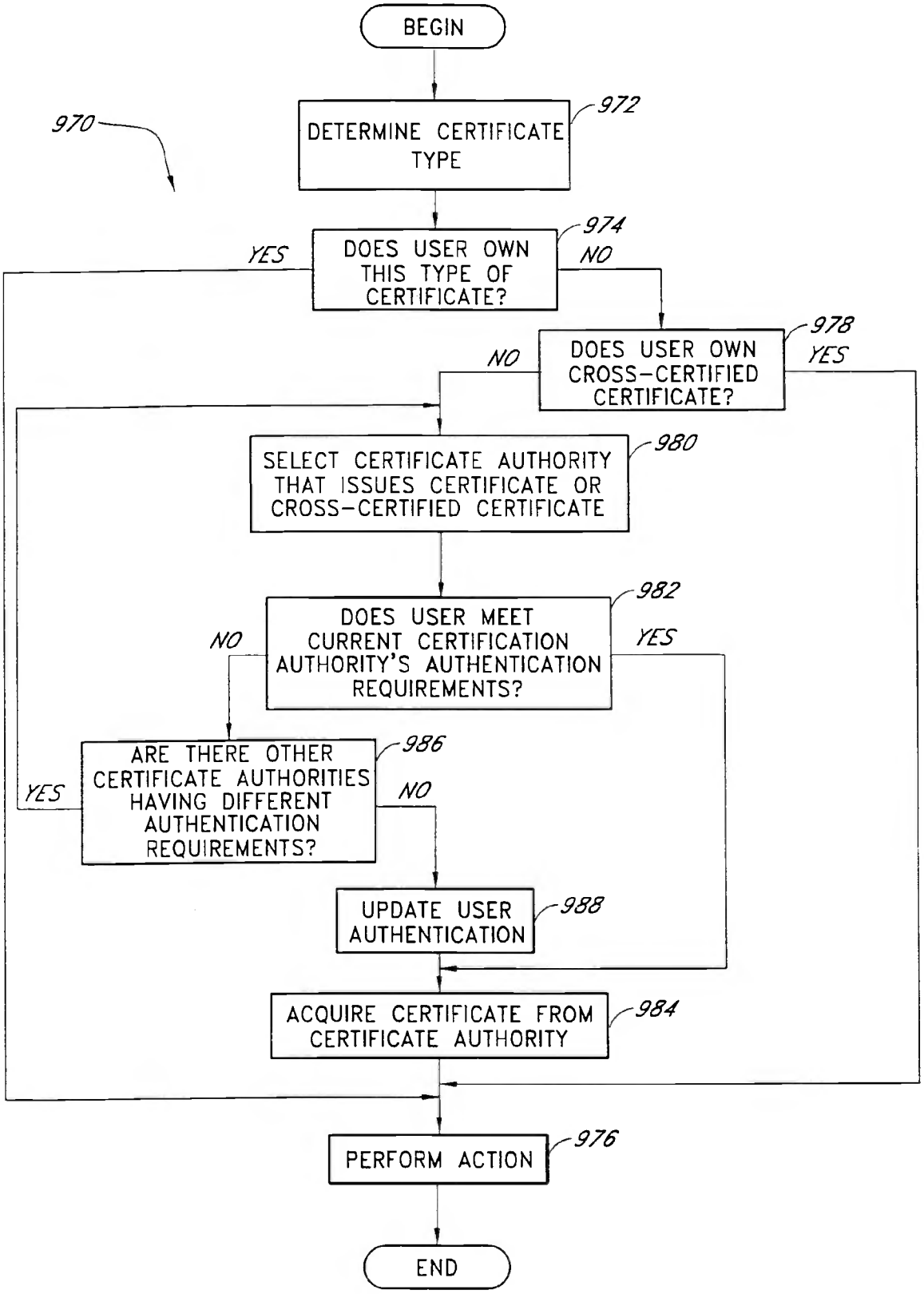
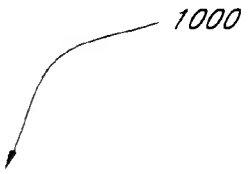

BEGIN
970
DETERMINE CERTIFICATE TYPE
972
DOES USER OWN THIS TYPE OF CERTIFICATE?
974
YES
NO
DOES USER OWN CROSS-CERTIFIED CERTIFICATE?
978
NO
YES
SELECT CERTIFICATE AUTHORITY THAT ISSUES CERTIFICATE OR CROSS-CERTIFIED CERTIFICATE
980
DOES USER MEET CURRENT CERTIFICATION AUTHORITY'S AUTHENTICATION REQUIREMENTS?
982
NO
YES
ARE THERE OTHER CERTIFICATE AUTHORITIES HAVING DIFFERENT AUTHENTICATION REQUIREMENTS?
986
YES
NO
UPDATE USER AUTHENTICATION
988
ACQUIRE CERTIFICATE FROM CERTIFICATE AUTHORITY
984
PERFORM ACTION
976
END

FIG. 9B



AUTHENTICATION DATA FLOW

	SEND	RECEIVE	SSL	ACTION
1005	USER	VENDOR	1/2	TRANSACTION OCCURS, SUCH AS SELECTING PURCHASE
1010	VENDOR	USER	1/2	TRANSMIT TRANSACTION ID (TID) AND AUTHENTICATION REQUEST (AR)
				AUTHENTICATION DATA (B') IS GATHERED FROM USER
1015	USER	TE	1/2	TRANSMIT TID AND B' WRAPPED IN THE PUBLIC KEY OF THE AUTHENTICATION ENGINE (AE), AS (PUB_AE(TID, B'))
1020	TE	AE	FULL	FORWARD TRANSMISSION
				ENROLLMENT AUTHENTICATION DATA (B) IS REQUESTED AND GATHERED
1025	VENDOR	TRANSACTION ENGINE (TE)	FULL	TRANSMITS TID, AR
1030	TE	MASS STORAGE(MS)	FULL	CREATE RECORD IN DATABASE
1035	TE	THE Xth DEPOSITORY(DX)	FULL	UID, TID
1040	DX	AE	FULL	TRANSMIT THE TID AND THE PORTION OF THE AUTHENTICATION DATA STORED AT ENROLLMENT (BX) AS (PUB_AE(TID, BX))
1045				AE ASSEMBLES B AND COMPARES TO B'
1050	AE	TE	FULL	TID, THE FILLED IN AR
	TE	VENDOR	FULL	TID, YES/NO
1055	TE	USER	1/2	TID, CONFIRMATION MESSAGE

FIG. 10

1100

SIGNING DATA FLOW			
SEND	RECEIVE	SSL	ACTION
USER	VENDOR	1/2	TRANSACTION OCCURS, SUCH AS AGREEING ON A DEAL
VENDOR	USER	1/2	TRANSMIT TRANSACTION IDENTIFICATION NUMBER (TID), AUTHENTICATION REQUEST (AR), AND AGREEMENT OR MESSAGE (M)
			CURRENT AUTHENTICATION DATA (B') AND A HASH OF THE MESSAGE RECEIVED BY THE USER (h(M')) IS GATHERED FROM USER
USER	TE	1/2	TRANSMIT TID, B', AR, AND h(M') WRAPPED IN THE PUBLIC KEY OF THE AUTHENTICATION ENGINE (AE) AS (PUB_AE(TID, B', h(M'))
TE	AE	FULL	FORWARD TRANSMISSION
			GATHER ENROLLMENT AUTHENTICATION DATA
VENDOR	TRANSACTION ENGINE (TE)	FULL	TRANSMITS UID, TID, AR, AND A HASH OF THE MESSAGE (h(M)).
TE	MASS STORAGE (MS)	FULL	CREATE RECORD IN DATABASE
TE	THE Xth DEPOSITORY(DX)	FULL	UID, TID
DX	AE	FULL	TRANSMIT THE TID AND THE PORTION OF THE AUTHENTICATION DATA STORED AT ENROLLMENT (BX),AS (PUB_AE(TID, BX))
			THE ORIGINAL VENDOR MESSAGE IS TRANSMITTED TO THE AE
TE	AE	FULL	TRANSMIT h(M)
			AE ASSEMBLES B, COMPARES TO B' AND COMPARES h(M) TO h(M')
AE	CRYPTOGRAPHIC ENGINE (CE)	FULL	REQUEST FOR DIGITAL SIGNATURE AND A MESSAGE TO BE SIGNED, FOR EXAMPLE, THE HASHED MESSAGE
AE	DX	FULL	TID, SIGNING UID
DX	CE	FULL	TRANSMIT THE PORTION OF THE CRYPTOGRAPHIC KEY CORRESPONDING TO THE SIGNING PARTY
			CE ASSEMBLES KEY AND SIGNS
CE	AE	FULL	TRANSMIT THE DIGITAL SIGNATURE (S) OF SIGNING PARTY
AE	TE	FULL	TID, THE FILLED IN AR, h(M), AND S
TE	VENDOR	FULL	TID, A RECEIPT=(TID, YES/NO, AND S), AND THE DIGITAL SIGNATURE OF THE TRUST ENGINE, FOR EXAMPLE, A HASH OF THE RECEIPT ENCRYPTED WITH THE TRUST ENGINE'S PRIVATE KEY (Priv_TE(h(RECEIPT)))
TE	USER	1/2	TID, CONFIRMATION MESSAGE

1103 1105 1110 1115 1120 1125 1130 1135 1140

FIG. 11



ENCRYPTION/DECRYPTION DATA FLOW			
SEND	RECEIVE	SSL	ACTION
DECRYPTION			
			PERFORM AUTHENTICATION DATA PROCESS 1000, INCLUDE THE SESSION KEY (SYNC) IN THE AR, WHERE THE SYNC HAS BEEN ENCRYPTED WITH THE PUBLIC KEY OF THE USER AS PUB_USER(SNYC)
			AUTHENTICATE THE USER
AE	CE	FULL	FORWARD PUB_USER(SYNC) TO CE
AE	DX	FULL	UID, TID
DX	CE	FULL	TRANSMIT THE TID AND THE PORTION OF THE PRIVATE KEY AS (PUB_AE(TID, KEY_USER))
			CE ASSEMBLES THE CRYPTIOGRAPHIC KEY AND DECRYPTS THE SYNC
CE	AE	FULL	TID, THE FILLED IN AR INCLUDING DECRYPTED SYNC
AE	TE	FULL	FORWARD TO TE
TE	REQUESTING APP/VENDOR	1/2	TID, YES/NO, SYNC
ENCRYPTION			
REQUESTING APP/VENDOR	TE	1/2	REQUEST FOR PUBLIC KEY OF USER
TE	MS	FULL	REQUEST DIGITAL CERTIFICATE
MS	TE	FULL	TRANSMIT DIGITAL CERTIFICATE
TE	REQUESTING APP/VENDOR	1/2	TRANSMIT DIGITAL CERTIFICATE

1205 1210 1215 1220 1225 1230 1235 1240 1245 1250

FIG. 12

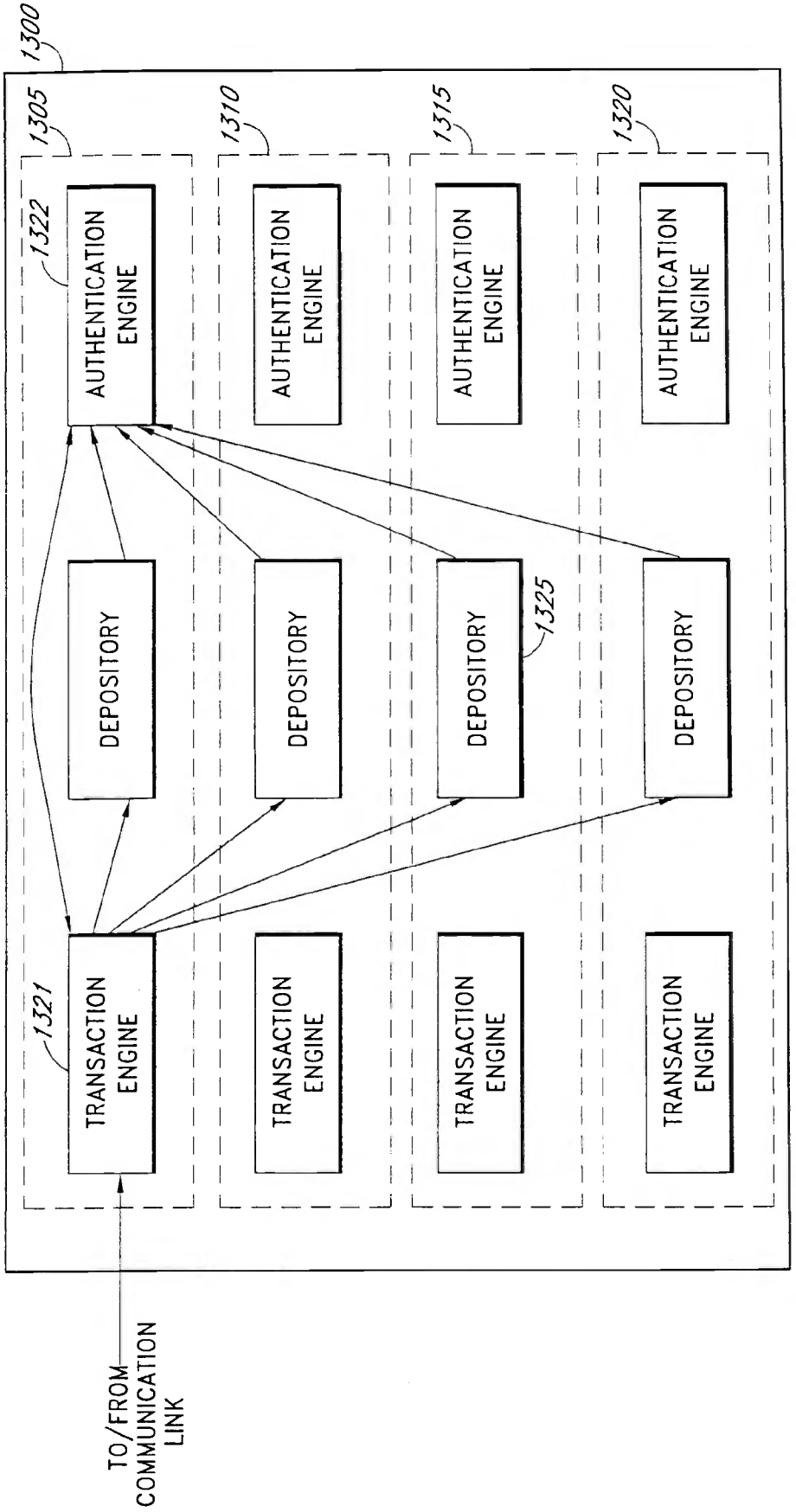

1300
1305
1310
1315
1320
1322
AUTHENTICATION ENGINE
AUTHENTICATION ENGINE
AUTHENTICATION ENGINE
AUTHENTICATION ENGINE
DEPOSITORY
DEPOSITORY
DEPOSITORY
1325
DEPOSITORY
1321
TRANSACTION ENGINE
TRANSACTION ENGINE
TRANSACTION ENGINE
TRANSACTION ENGINE
TO/FROM COMMUNICATION LINK

FIG. 13

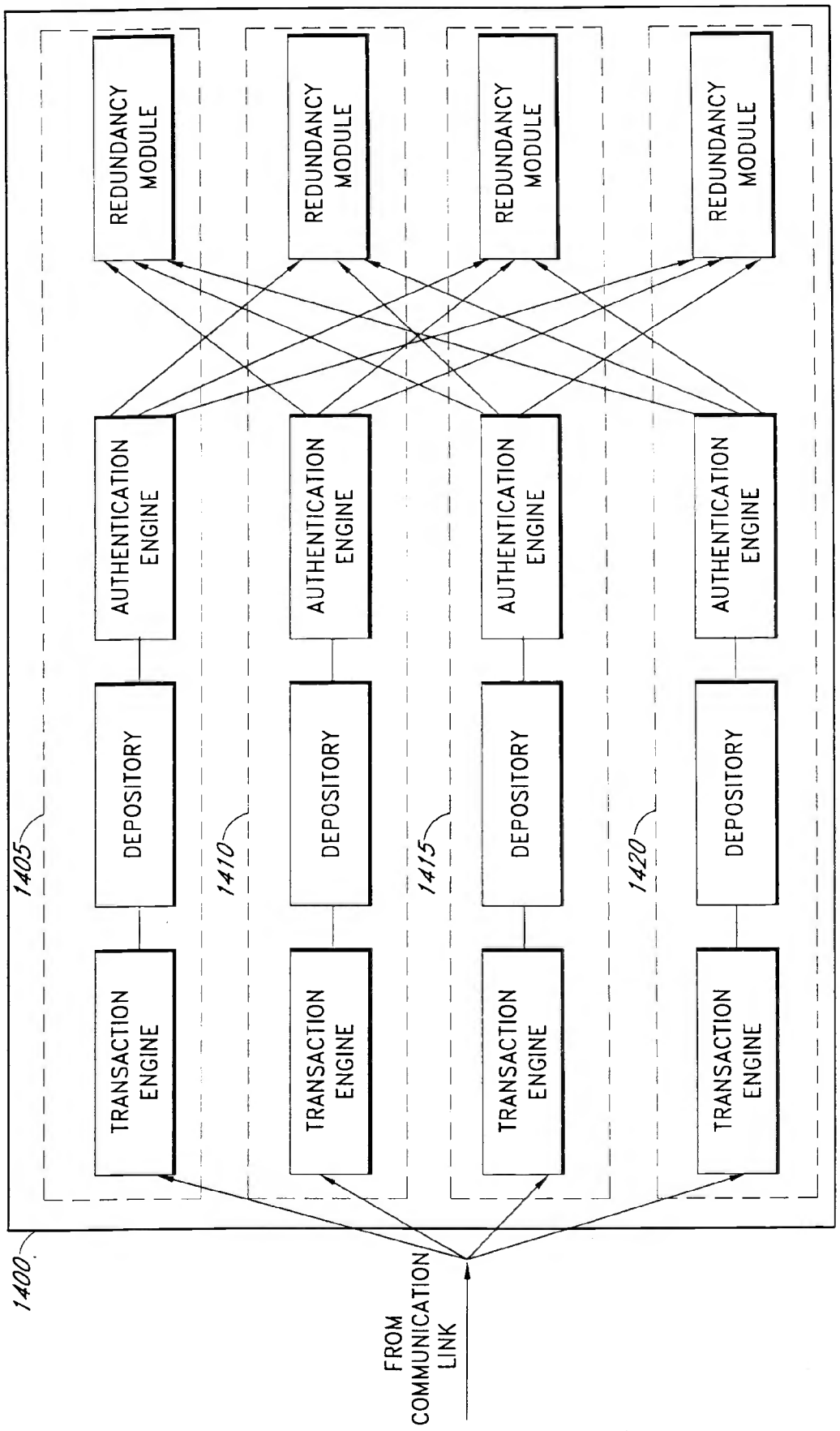

REDUNDANCY MODULE
REDUNDANCY MODULE
REDUNDANCY MODULE
REDUNDANCY MODULE
AUTHENTICATION ENGINE
AUTHENTICATION ENGINE
AUTHENTICATION ENGINE
AUTHENTICATION ENGINE
DEPOSITORY
DEPOSITORY
DEPOSITORY
DEPOSITORY
TRANSACTION ENGINE
TRANSACTION ENGINE
TRANSACTION ENGINE
TRANSACTION ENGINE
1405
1410
1415
1420
1400
FROM COMMUNICATION LINK

FIG. 14

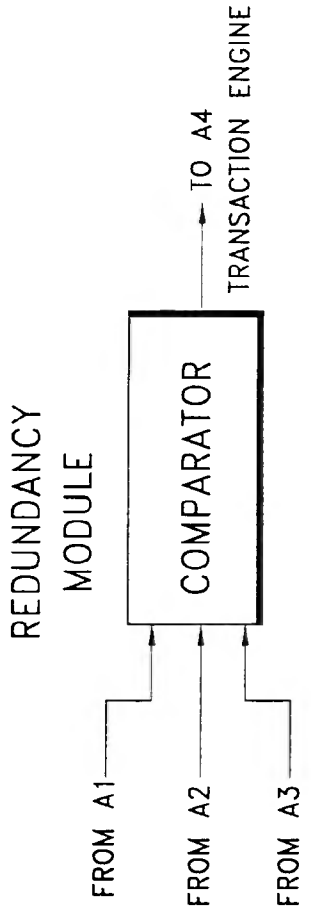

REDUNDANCY MODULE
COMPARATOR
TO A4
TRANSACTION ENGINE
FROM A1
FROM A2
FROM A3

FIG. 15

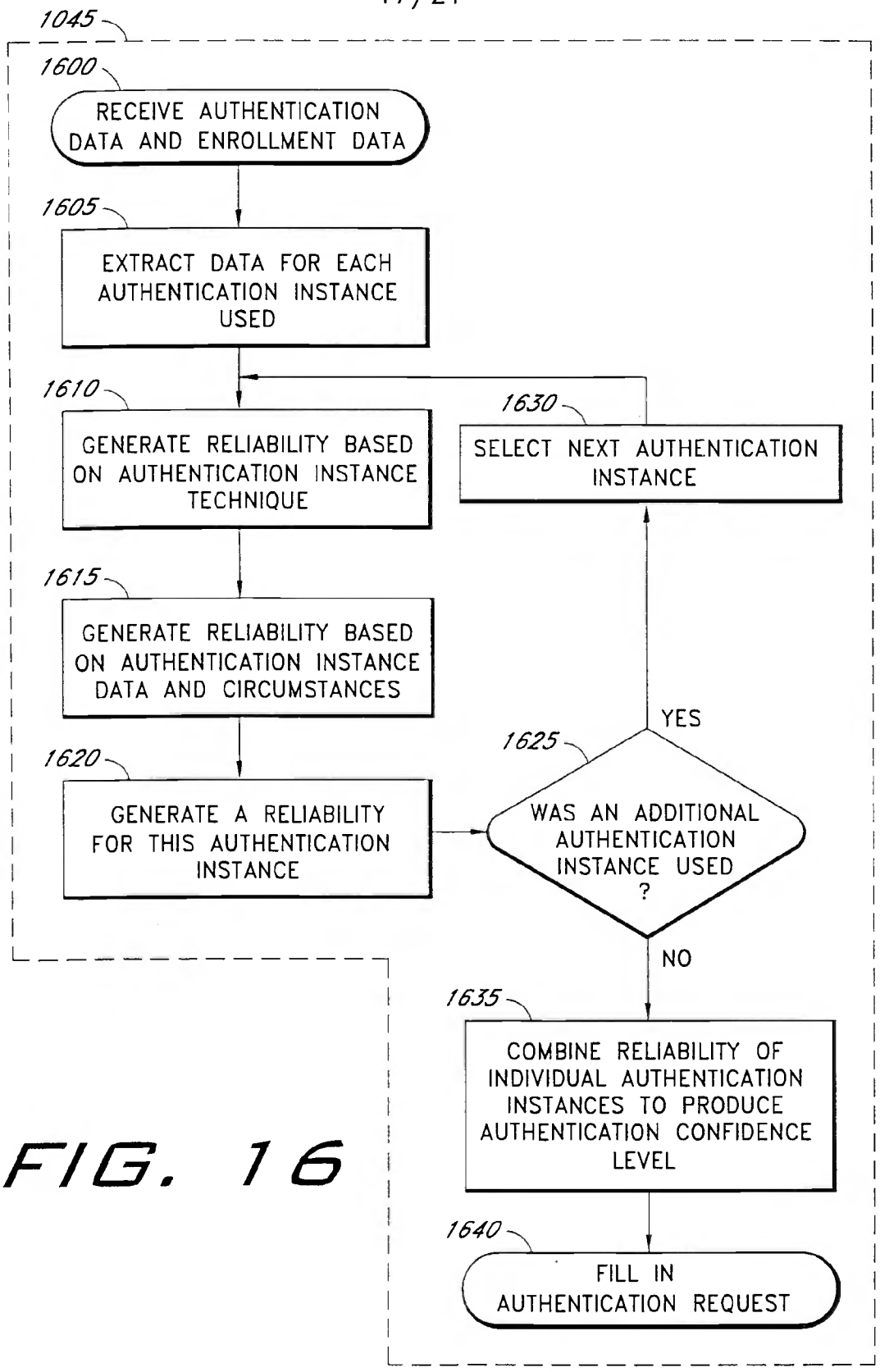

1045
1600
RECEIVE AUTHENTICATION DATA AND ENROLLMENT DATA
1605
EXTRACT DATA FOR EACH AUTHENTICATION INSTANCE USED
1610
GENERATE RELIABILITY BASED ON AUTHENTICATION INSTANCE TECHNIQUE
1615
GENERATE RELIABILITY BASED ON AUTHENTICATION INSTANCE DATA AND CIRCUMSTANCES
1620
GENERATE A RELIABILITY FOR THIS AUTHENTICATION INSTANCE
1625
WAS AN ADDITIONAL AUTHENTICATION INSTANCE USED ?
YES
NO
1630
SELECT NEXT AUTHENTICATION INSTANCE
1635
COMBINE RELIABILITY OF INDIVIDUAL AUTHENTICATION INSTANCES TO PRODUCE AUTHENTICATION CONFIDENCE LEVEL
1640
FILL IN AUTHENTICATION REQUEST

FIG. 16

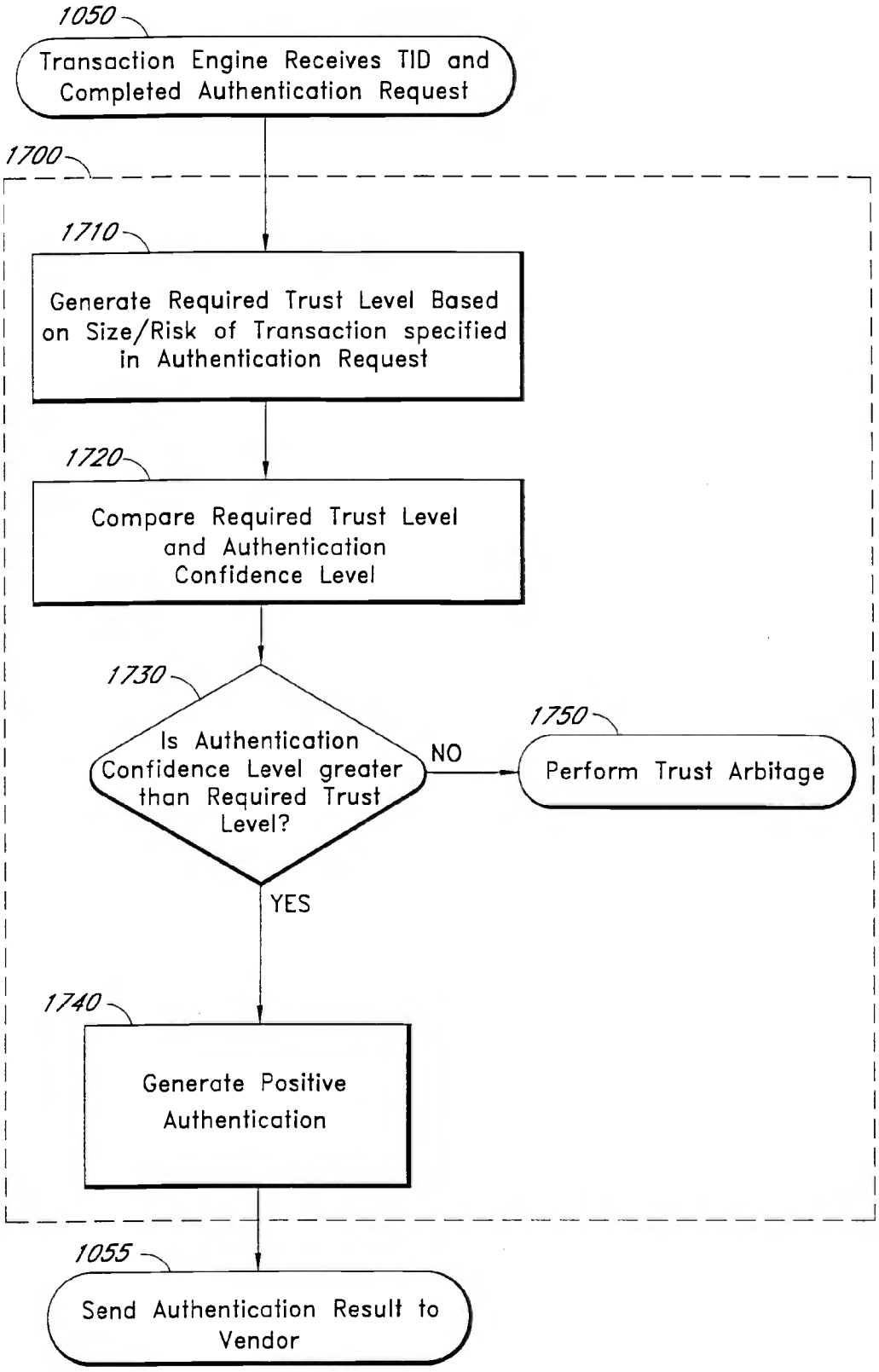

1050
Transaction Engine Receives TID and Completed Authentication Request
1700
1710
Generate Required Trust Level Based on Size/Risk of Transaction specified in Authentication Request
1720
Compare Required Trust Level and Authentication Confidence Level
1730
Is Authentication Confidence Level greater than Required Trust Level?
NO
1750
Perform Trust Arbitage
YES
1740
Generate Positive Authentication
1055
Send Authentication Result to Vendor

FIG. 17

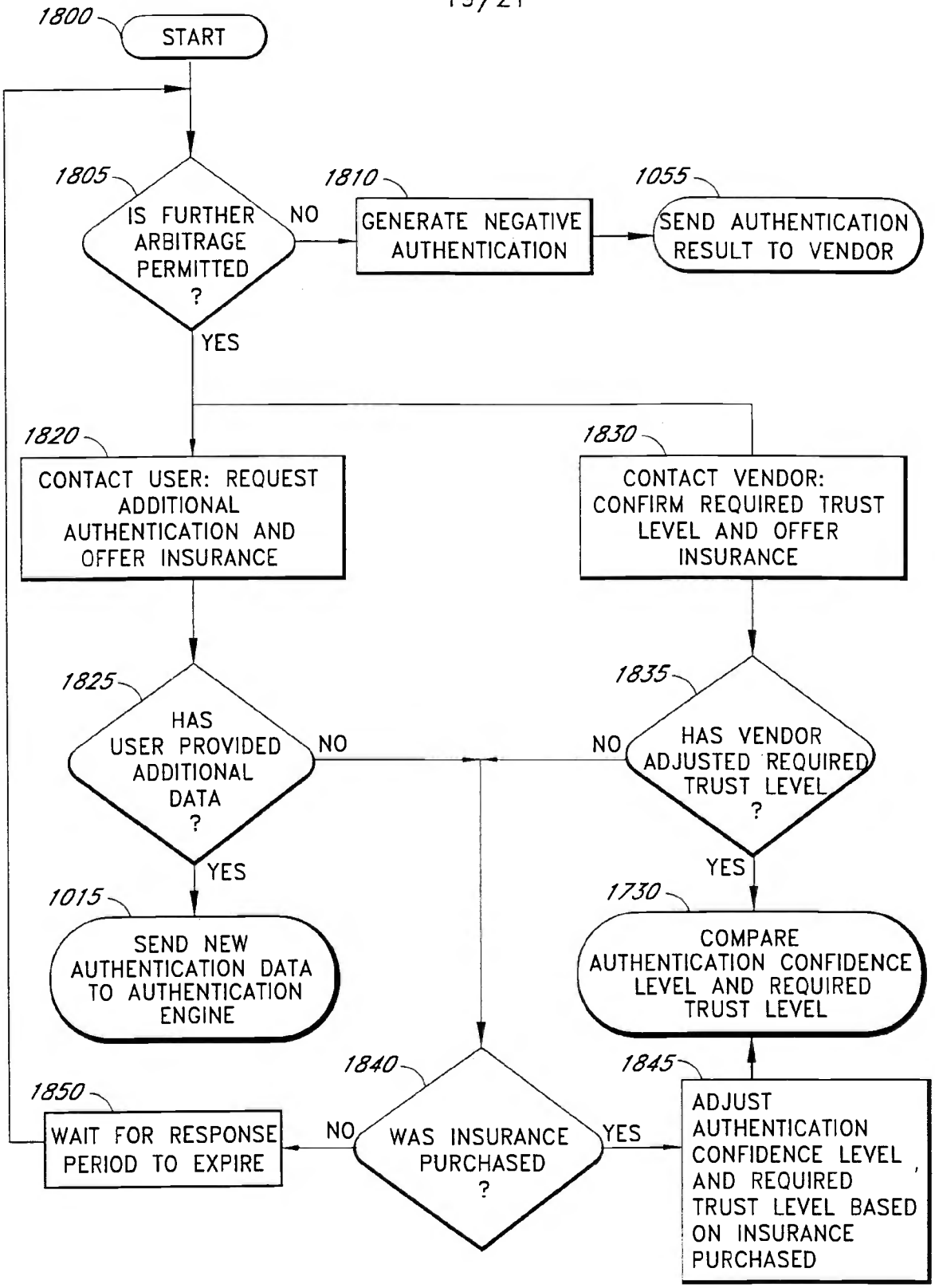

1800
START
1805
IS FURTHER ARBITRAGE PERMITTED ?
NO
1810
GENERATE NEGATIVE AUTHENTICATION
1055
SEND AUTHENTICATION RESULT TO VENDOR
YES
1820
CONTACT USER: REQUEST ADDITIONAL AUTHENTICATION AND OFFER INSURANCE
1830
CONTACT VENDOR: CONFIRM REQUIRED TRUST LEVEL AND OFFER INSURANCE
1825
HAS USER PROVIDED ADDITIONAL DATA ?
NO
1835
HAS VENDOR ADJUSTED REQUIRED TRUST LEVEL ?
NO
YES
YES
1015
SEND NEW AUTHENTICATION DATA TO AUTHENTICATION ENGINE
1730
COMPARE AUTHENTICATION CONFIDENCE LEVEL AND REQUIRED TRUST LEVEL
1840
WAS INSURANCE PURCHASED ?
1845
ADJUST AUTHENTICATION CONFIDENCE LEVEL AND REQUIRED TRUST LEVEL BASED ON INSURANCE PURCHASED
1850
WAIT FOR RESPONSE PERIOD TO EXPIRE
NO
YES

FIG. 18

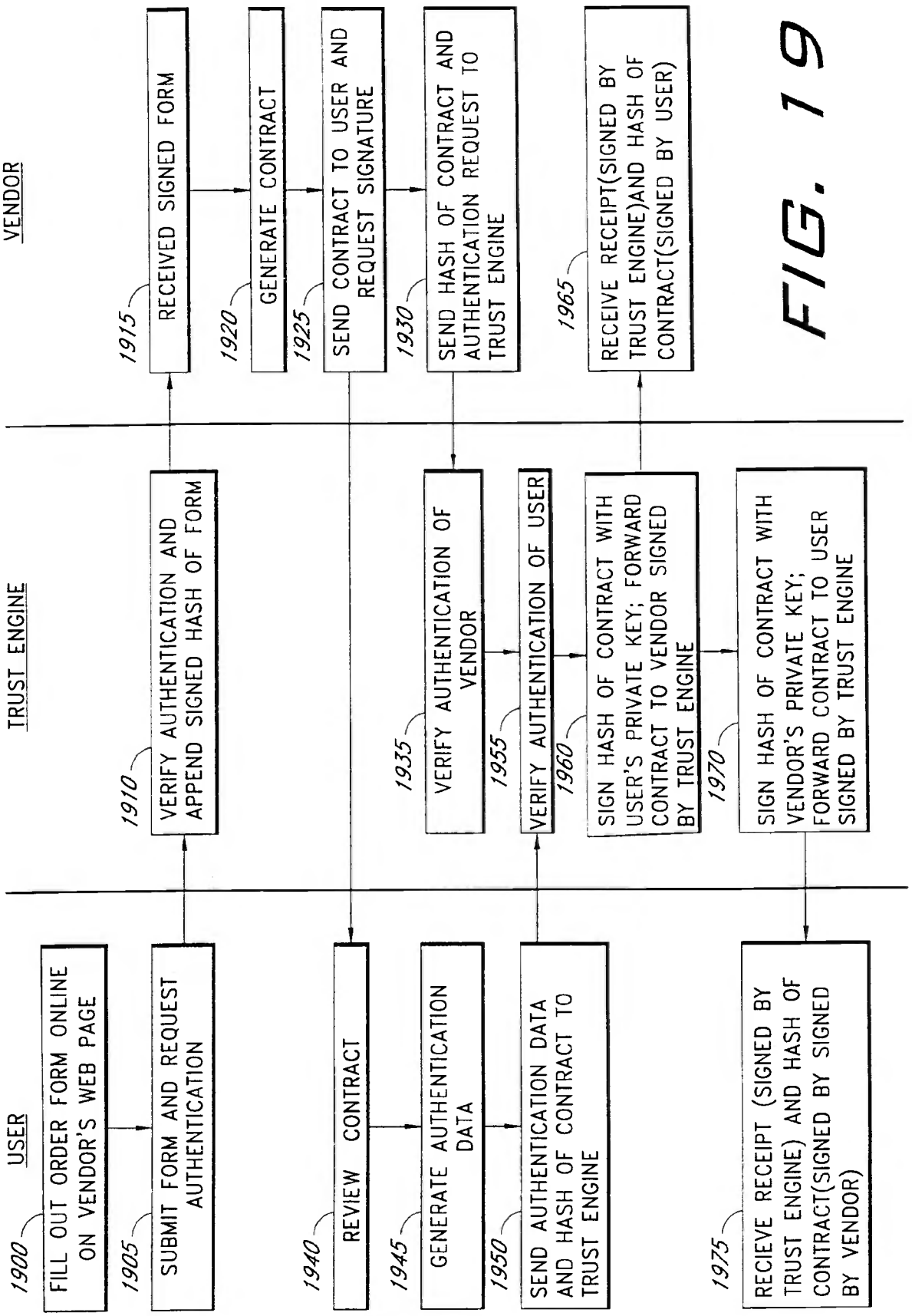

USER
TRUST ENGINE
VENDOR
1900
FILL OUT ORDER FORM ONLINE ON VENDOR'S WEB PAGE
1905
SUBMIT FORM AND REQUEST AUTHENTICATION
1910
VERIFY AUTHENTICATION AND APPEND SIGNED HASH OF FORM
1915
RECEIVED SIGNED FORM
1920
GENERATE CONTRACT
1925
SEND CONTRACT TO USER AND REQUEST SIGNATURE
1930
SEND HASH OF CONTRACT AND AUTHENTICATION REQUEST TO TRUST ENGINE
1935
VERIFY AUTHENTICATION OF VENDOR
1940
REVIEW CONTRACT
1945
GENERATE AUTHENTICATION DATA
1950
SEND AUTHENTICATION DATA AND HASH OF CONTRACT TO TRUST ENGINE
1955
VERIFY AUTHENTICATION OF USER
1960
SIGN HASH OF CONTRACT WITH USER'S PRIVATE KEY; FORWARD CONTRACT TO VENDOR SIGNED BY TRUST ENGINE
1965
RECEIVE RECEIPT(SIGNED BY TRUST ENGINE)AND HASH OF CONTRACT(SIGNED BY USER)
1970
SIGN HASH OF CONTRACT WITH VENDOR'S PRIVATE KEY; FORWARD CONTRACT TO USER SIGNED BY TRUST ENGINE
1975
RECIEVE RECEIPT (SIGNED BY TRUST ENGINE) AND HASH OF CONTRACT(SIGNED BY SIGNED BY VENDOR)

FIG. 19

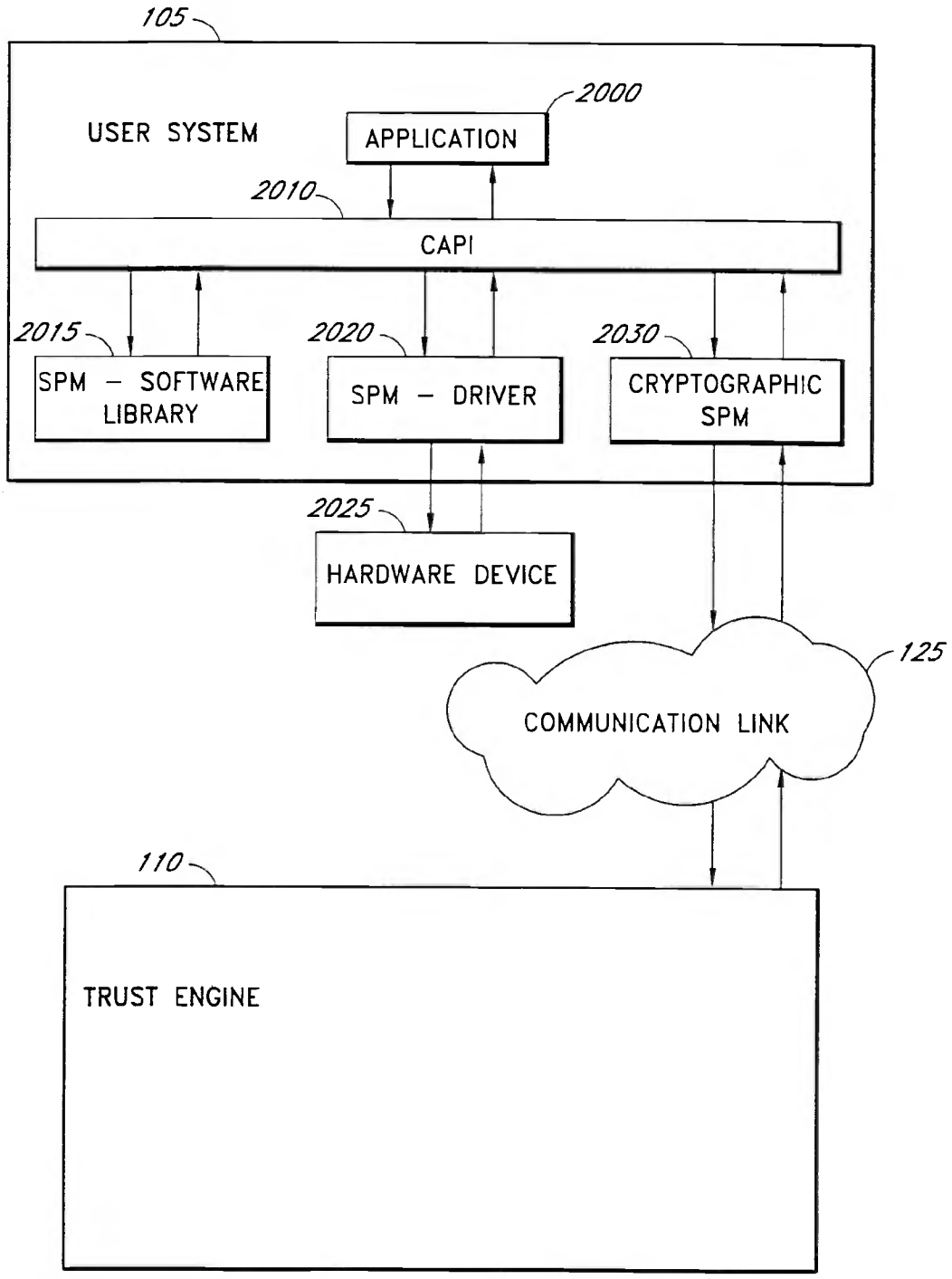

105
USER SYSTEM
2000
APPLICATION
2010
CAPI
2015
SPM – SOFTWARE LIBRARY
2020
SPM – DRIVER
2030
CRYPTOGRAPHIC SPM
2025
HARDWARE DEVICE
125
COMMUNICATION LINK
110
TRUST ENGINE

FIG. 20

INTERNATIONAL SEARCH REPORT

Int tional Application No
PCT/US 00/25754

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F17/60 G07F7/10

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F G07F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropnate, of the relevant passages	Relevant to claim No.
X A	US 5 386 104 A (SIME IAIN R F) 31 January 1995 (1995-01-31) abstract column 5, line 45 - line 65 figure 3 ---	31, 34-40,42 1-30,32, 33,41, 43-50
A	US 5 903 882 A (ASAY ALAN ET AL) 11 May 1999 (1999-05-11) abstract column 5, line 7 - line 65 figures 3,6 claims 1-56 --- -/--	1-50

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16 February 2001	27/02/2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Van Dop, E

1

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Int .tional Application No
PCT/US 00/25754

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	WO 99 19845 A (AT & T CORP) 22 April 1999 (1999-04-22) abstract page 7, line 13 -page 8, line 7 page 12, line 1 - line 16 ---	1-50
A	EP 0 485 090 A (VISA INT SERVICE ASS) 13 May 1992 (1992-05-13) abstract -----	1-50

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int ational Application No
PCT/US 00/25754

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5386104 A	31-01-1995	DE 69425944 D	26-10-2000
		EP 0652540 A	10-05-1995
		ES 2150474 T	01-12-2000
		JP 7192164 A	28-07-1995
US 5903882 A	11-05-1999	AU 5515398 A	03-07-1998
		BR 9714400 A	18-04-2000
		CN 1244936 A	16-02-2000
		EP 0965111 A	22-12-1999
		WO 9826385 A	18-06-1998
WO 9919845 A	22-04-1999	EP 0941526 A	15-09-1999
		US 6125349 A	26-09-2000
EP 0485090 A	13-05-1992	US 5177342 A	05-01-1993
		AT 165927 T	15-05-1998
		AU 657060 B	23-02-1995
		AU 4863693 A	09-12-1993
		AU 644015 B	02-12-1993
		AU 8479191 A	14-05-1992
		CA 2052033 A,C	10-05-1992
		DE 69129357 D	10-06-1998
		DE 69129357 T	10-09-1998
		EP 0709811 A	01-05-1996
		HK 1013876 A	05-05-2000
		JP 2907609 B	21-06-1999
		JP 4267495 A	24-09-1992

Form PCT/ISA/210 (patent family annex) (July 1992)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 636 259 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
04.06.1997 Bulletin 1997/23

(21) Application number: **93912226.3**

(22) Date of filing: **15.04.1993**

(51) Int Cl.[6]: **G06F 12/14**, G06F 1/00

(86) International application number:
PCT/US93/03472

(87) International publication number:
WO 93/21581 (28.10.1993 Gazette 1993/26)

(54) CRYPTOGRAPHIC DATA SECURITY IN A SECURED COMPUTER SYSTEM

KRYPTOGRAPHISCHE DATENSICHERHEIT IN EINEM GESICHERTEN COMPUTERSYSTEM.

SECURISATION DE DONNEES CRYPTOGRAPHIQUES DANS UN SYSTEME INFORMATIQUE SECURISE

(84) Designated Contracting States:
AT BE CH DE DK ES FR GB GR IE IT LI LU MC NL PT SE

(30) Priority: **17.04.1992 US 870556**

(43) Date of publication of application:
01.02.1995 Bulletin 1995/05

(60) Divisional application: **96201432.0**

(73) Proprietor: **SECURE COMPUTING CORPORATION**
Wilmington, New Castle (US)

(72) Inventors:
- **BOEBERT, William, E.**
  Minneapolis, MN 55409 (US)
- **MARKHAM, Thomas, R.**
  Anoka, MN 55303 (US)
- **OLMSTED, Robert, A.**
  Minnetonka, MN (US)

(74) Representative:
Beresford, Keith Denis Lewis et al
BERESFORD & Co.
2-5 Warwick Court
High Holborn
London WC1R 5DJ (GB)

(56) References cited:
EP-A- 0 421 409
EP-A- 0 471 538
US-A- 5 052 040

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Printed by Jouve, 75001 PARIS (FR)

EP 0 636 259 B1

Description

Field of the Invention

This invention relates generally to data communication systems, and more specifically to secure data processing on a data communication system.

Background of the Invention

Data Enclave

Individuals working in a departmental computing environment typically have a substantial amount of computing power on their desks in the form of personal computers and workstations. A workstation has a computational subsystem, keyboard, and display for user interaction, and typically substantial amounts of local data storage in the form of fixed and removable media.

In order for the individual in the departmental computing environment to interact and share data, their workstations are typically attached to a local area network (LAN) which permits the transfer of data files and electronic mail between the workstations. In addition, "servers" may be attached to the LAN to provide specialized services, such as the management of centralized databases, which are not practical for individual workstations.

Departmental computing environments are typically members of a larger organization or have other reasons to communicate with computing facilities outside themselves. They therefore make use of a special kind of server, called a "gateway", to gain access to a wide area network (WAN). WANs are often interconnected (called "internetting") to provide world-wide data transmission paths.

Departmental Computing Environment

A typical overall departmental computing environment is shown in Figure 1. In the departmental computer environment 1, large amounts of valuable data is stored on magnetic or other electronic Media 2, 4 for processing in the Workstations 10 and file servers (not shown). This media offers the benefits of compact storage, easy retrieval, and in the case of removable Media 4 (e.g., "diskettes"), convenient sharing and distribution.

In addition, data is transmitted freely around the Local Area Network 12 and occasionally through a Gateway 14 to the Wide Area Network 16 and Remote Sites 18. This transmission is necessary in order for the organization performing departmental computing to perform its internal work and interact with the outside world.

There is also a requirement that certain operations, including but not limited to the transmission of data to the outside world, be restricted to individuals who possess special privileges. Examples of such operations are messages (electronic mail) which are directive in nature, such as users to transfer funds, and operations such as the adding of new orders or the granting of limited access to departmental data to users on the Wide Area Network 16 (remote login and file transfer).

Threats Against Department Computing Environment

The threats against the departmental computing environment are shown in Figure 2.

The data in this environment is vulnerable to theft and tampering. Removable media can be stolen, copied, and returned with no sign that loss has occurred. The fruits of thousands of hours of labor can be stolen in a package that fits easily in a coat pocket. Crucial data can be modified or destroyed, either directly or through the agency of technical entities such as "viruses", which are introduced into the Workstations 10 and servers through the agency of corrupted media or through the wide area network connection.

There are also threats to the privileged operations. Unauthorized individuals, masquerading as someone else, can cause disruptive or erroneous directives to be issued and thereby perpetrate sabotage and fraud. Malicious "hackers" with access to the wide area network can use that network to "reach in" to the departmental computing environment and masquerade as authorized users or otherwise obtain access to data, which they can then transfer worldwide, again with no sign that compromise has occurred.

Accordingly, there is a need for techniques whereby a departmental computing system 1 can be converted into a "data enclave." Within such an enclave:

(1) Data can be restricted to a single organization, such as a government agency or a corporation.

(2) Sharing of data between organizational elements (directorates, departments, projects, etc.) can be controlled. For example, it may be required that data such as a telephone directory be accessible by every employee, but data such as engineering drawings should not be allowed to circulate throughout the whole corporation.

(3) Sharing of data between individuals in organizational elements can be controlled. For example, even though an individual is a member of the engineering department, that individual may not have a "need to know" for all of the drawings in the department.

(4) Data is protected from technical attacks such as "viruses" and "worms."

(5) Intellectual property is protected irrespective of whether it is on electronic media, being processed in a Workstation, or being transferred around the local area network.

(6) The protections are achieved with minimum cost and disruption of operations, such as would occur if access to the wide area network were forbidden.

EX1010, Page 1222

(7) Privileged operations are restricted to those users possessing the requisite privileges and cannot be invoked, through masquerading or other technical means, by unauthorized users.

As shown in overview form in Figure 3, and as will be described more fully in the Detailed Description of the Invention, the facilities provided by the present invention convert a departmental computing environment into a "data enclave" 20 with a well-defined perimeter 22. Sharing of data within the Enclave 20 is controlled, and movement of data within and outside the enclave can only be effected by authorized individuals with suitable privilege. There are no "sneak paths" or "holes" that exist.

The present invention also minimizes the damage that can be done by privileged individuals who become subverted. Cryptographic keys are transmitted and stored entirely in enciphered form, and well-known techniques (called "antitamper" technology) can be used to protect an enclave key when it is in use inside a cryptographic device. Theft of elements of the present invention does not compromise any part of the operation of the invention.

Individuals desiring access to Media 2,4 have to deal with a Secure Computer 24, in this case a security server, only when Media 2,4 is initialized. "Unlocking" a unit of Media 2,4 requires an operation no more complicated than using a television remote control. Overhead and delay is concentrated at the time a Media 2,4 is "unlocked", and no delays or incompatibilities are introduced during operations using the Media 2 or 4.

Remotely invoked privileged operations at the security server 24 are under the positive control of the user. That control is cryptographically protected and mutually authenticated.

Identification and authentication of users to the security server 24 is both simpler and more robust than former implementations such as passwords. The same basic steps are used for security operations dealing with Media 2,4 and dealing with the security server 24.

In the data protection area, the system associates Media 2 or 4 primarily with users and secondarily with machines or Workstations 10. This is a more natural structure than one where media is only useable on a single machine or Workstation 10.

Control logic computes allowed access at the last possible moment using the combination of an "access vector" assigned to an individual and the "device attributes" assigned to a particular Workstation 10, which can be used to enforce a variety of security policies. For example, an individual's access to data may be restricted not only on the basis of the individual's attributes but also to protected physical locations. Thus an individual's access vector may grant "read" access to a unit of media which contains proprietary engineering data, but the comparison against the device attributes making the access, may restrict display of the contents of the unit of media to those machines inside a particular facility or office. Physical security measures can then be used to restrict who may be in the vicinity when the data is displayed. Previous implementations in this area have permitted only an "all or nothing" approach to access.

European Patent Specification No. EP-A-421409 relates to data security for networks converting a host computer to several work stations and personal data carriers such as IC-cards by means of different keys or data elements.

US Patent Specification US-A-5052040 also discloses a security system using different keys for communication and data processing.

In accordance with a first aspect of the invention there is provided a data enclave as set out in claim 1.

In accordance with a second aspect of the invention there is provided a data enclave method as set out in claim 2.

A media key is provided for each unit of media, and used to encrypt and protect data carried on the media, with the media keys stored in the personal keying devices. A media unique identifier (media UID) is provided for each unit of media, stored on the media, and used to identify the corresponding media key for the unit of media stored in a personal keying device, and to identify media attributes assigned to the unit of media. Media attributes are associated with each unit of media to which a media UID has been assigned, and used to represent the sensitivity or other security related information that may pertain to the data carried on that unit of media.

An access vector is associated with each media key to form media key/access vector pairs, stored in the personal keying devices, and used to represent the possible conditions of access to the data encrypted on the media for the user assigned to the personal keying device holding the media key/access vector pair or pairs with each access vector formed using the corresponding media attributes and user attributes, and a set of access rules. The media key/access vector pairs are stored in the personal keying devices enciphered with a combined key including the user's UID, the user's PIN and the enclave key. Device attributes are assigned to each workstation, stored in that device's crypto media controller, and used to represent the security attributes of the workstations.

Each crypto media controller includes access control logic for restricting access to the data on the media based on the user's PIN, the access vector and the device attributes for the workstation from which access is attempted.

Brief Description of the Drawings

The operational enhancements and features of the present invention become more apparent from a consideration of the drawings and following detailed description.

EX1010, Page 1223

Figure 1 is a diagram illustrating a typical departmental computing environment incorporating a local area network with a wide area network.

Figure 2 is a diagram illustrating possible threats against the departmental computing environment.

Figure 3 is an overall simplified block diagram of a secure data processing system illustrating the Data Enclave implementation.

Figure 4 is a simplified block diagram of the main data processing elements in the apparatus implementing the present invention.

Figure 5 is a simplified block diagram of the Workstation data processing elements using a Workstation configuration supporting coprocessor cryptography.

Figure 6 is a simplified block diagram of the Workstation data processing elements using a Workstation configuration supporting inline cryptography.

Figure 6a is a pictorial diagram of a personal keying device illustrating the appearance, features, and functions.

Figure 6b is a schematic diagram of the data elements created and utilized for the protection of data in the present invention.

Figure 7 is a simplified block diagram illustrating the steps for the extraction of user data at the Workstation, implemented in the Media Initialization and Key Generation phase of Data Enclave operation.

Figure 8 is a simplified block diagram illustrating the step for preparation and sending of a "Request Packet", implemented in the Media Initialization and Key Generation phase of Data Enclave operation.

Figure 9 is a simplified block diagram illustrating the step for receipt of a "Request Packet" at the Security Server, implemented in the Media Initialization and Key Generation phase of Data Enclave operation.

Figure 10 is a simplified block diagram illustrating the steps for the checking of user identity and the generation of a Media UID, implemented in the Media Initialization and Key Generation phase of Data Enclave operation.

Figure 11 is a simplified block diagram illustrating the steps for Access Vector generation, implemented in the Media Initialization and Key Generation phase of Data Enclave operation.

Figure 12 is a simplified block diagram illustrating the steps for "Key Packet" generation and storage, implemented in the Media Initialization and Key Generation phase of Data Enclave operation.

Figure 13 is a simplified block diagram illustrating the steps for Media UID and "Key Packet" assignment, implemented in the Media Initialization and Key Generation phase of Data Enclave operation.

Figure 14 is a simplified block diagram illustrating the steps for extracting identification data and forming a Request, implemented in the Key Assignment phase of Data Enclave operation.

Figure 15 is a simplified block diagram illustrating the step for the encryption and transmission of a "Request Packet", implemented in the Key Assignment phase of Data Enclave operation.

Figure 16 is a simplified block diagram illustrating the steps for the computation of an Access Vector, implemented in the Key Assignment phase of Data Enclave operation.

Figure 17 is a simplified block diagram illustrating the steps for key generation, storage, and transmission, implemented in the Key Assignment phase of Data Enclave operation.

Figure 18 is a simplified block diagram illustrating the step for the transfer of the key to the personal keying device, implemented in the Key Assignment phase of Data Enclave operation.

Figure 19 is a simplified block diagram illustrating the steps for Media Key and Access Vector extraction, implemented in the Keying of Devices phase of Data Enclave operation.

Figure 20 is a simplified block diagram illustrating the steps for Media Key and Access Vector use, implemented in the Keying of Devices phase of Data Enclave operation.

Figure 21 is a simplified block diagram illustrating the steps for the initialization of the authentication process, implemented in the Identification and Authentication phase of Trusted Path operation.

Figure 22 is a simplified block diagram illustrating the step for the authentication of identity and the establishment of privileges, implemented in the Identification and Authentication phase of Trusted Path operation.

Figure 23 is a simplified block diagram illustrating the step for the preparation and transmission of the "Response Packet", implemented in the Identification and Authentication phase of Trusted Path operation.

Figure 24 is a simplified block diagram illustrating the step for the completion of the authentication sequence, implemented in the Identification and Authentication phase of Trusted Path operation.

Figure 25 is a simplified block diagram illustrating the steps for the initiation of a privileged operation, implemented in the Privileged Services phase of Trusted Path operation.

Figure 26 is a simplified block diagram illustrating the steps for the determination of privileges, implemented in the Privileged Services phase of Trusted Path operation.

Figure 27 is a simplified block diagram illustrating the step for the acknowledgment of privileges, implemented in the Privileged Services phase of Trusted Path operation.

Figure 28 is a simplified block diagram illustrating the step for the display of the acknowledgment, implemented in the Privileged Services phase of Trusted Path operation.

Figure 29 shows an alternate embodiment of the Data Enclave system.

Figure 30 shows the configuration for initializing fixed media according to the alternate embodiment of

EX1010, Page 1224

Figure 29.

Figure 31 shows the configuration for initializing removable media according to the alternate embodiment of Figure 29.

Detailed Description of the Invention

In the following detailed description of the preferred embodiments, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration, specific embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

The term "logic" is used throughout the ensuing description with reference to the structure of various electronic components of the invention. The term is intended to have a broad meaning, and to encompass hardware implemenations, software implementations, and combinations thereof.

Processing Elements

The present invention consists of processing elements and data elements. The interrelation of the processing elements is shown generally in Figures 3 and 4 (in part described above) and in more detail in Figures 5 and 6. The descriptions given below show cryptographic protection provided only to those distinguished transmissions required in the operation of the invention. In such a case, the elements of the invention are preferably arranged with regard to the Workstation 10 as shown in Figure 5.

If it is desired to protect all transmissions over the Local Area Network 12, e.g., to prevent wiretapping or other monitoring by unauthorized personnel, then the Crypto Media Controller 26 could be used to encipher and decipher all data going out over the Network 12. In this case, the elements of the invention could be arranged with regard to the Workstation 10 as shown in Figure 6.

Security Server

The Security Server 24, a secure computer, is a distinguished server that performs gateway and security functions at the interface between the Local Area Network 12 and the Wide Area Network 16. It also performs the key management and backup functions for the cryptography in the Enclave 20. The Security Server 24 can be implemented in the form of a secure computer for example, as disclosed in U.S. Patent No. 4,621,321 to Boebert et al, entitled "Secure Data Processing System Architecture", 4,713,753 to Boebert et al, entitled "Secure Data Processing System Architecture with Format Control", and 4,701,840 to Boebert et al, entitled "Secure Data Processing System Architecture".

Personal Keying Device

Each user 5 is issued a Personal Keying Device 30. Personal Keying Devices 30 are used for key insertion and individual authentication. A Personal Keying Device 30 (shown in more detail in Figure 6a) preferably contains fixed or removable electronic storage and processor 32, a keypad 34, a display 36, and a data transfer interface 38 that can be either wired or wireless (e.g., radio, infrared) and is compatible with an interface 31 on a Crypto Media Controller 26. The Personal Keying Device 30 can be highly portable, e.g., pocket calculator size. Personal Keying Devices 30 may also be equipped with theft detection circuitry to prevent them from being physically removed from the enclave working area.

Crypto Media Controller

The standard media controller on each Workstation 10 is replaced with a Crypto Media Controller 26. Crypto Media Controllers 26 perform key management, media encryption and decryption, and authentication functions. A Crypto Media Controller 26 has the same interfaces as the standard media controllers, as well as a data transfer interface that is compatible with the one on the Personal Keying Device 30. The Crypto Media Controllers 26 can be the same size as the standard media controllers they replace.

Data Elements

The present invention also includes a variety of data elements, as described below and schematically represented in Figure 6b.

Enclave Key

There is one Enclave Key 40 per organization. It is held in protected storage in the Security Server 24 and the Crypto Media Controllers 26, and is used to protect Media Keys 42 when they are being transmitted along the LAN 12.

Media Key

There is one Media Key 42 assigned to each physical unit of the media, whether that unit is fixed 2 or removable 4. Assignment is done when the media is initialized at the Workstation 10. This key is used to protect the data on the Media 2 or 4.

Combined Keys

Combined Keys 44 are generated in the operation of the present invention from other data elements and keys.

EX1010, Page 1225

Media Unique Identifier (Media UID)

Each physical unit of media, whether fixed 2 or removable 4, is assigned a Media Unique Identifier 46 (Media UID). This number is generated by the Security Server 24, and stored in whatever field the Media 2 or 4 software uses to identify physical units (e.g., Volume Label). The Media UID 46 is used to find the appropriate Media Key 42 in the Personal Keying Device 30, and to locate that data pertaining to the unit of media which is stored in the Security Server 24 (e.g., Media Attributes).

User Unique Identifier (User UID)

Each individual who has potential access to encrypted media is assigned a User Unique Identifier 48 (User UID) which is stored in that user's Personal Keying Device 30, encrypted with the Enclave Key 40. The User UID 48 forms part of the key used to protect Media Keys 42 in the Personal Keying Device 30, and is used to extract that data pertaining to the user 5 which is stored in the Security Server 24 (e.g., User Attributes).

Personal Identification Number (PIN)

Each user 5 is assigned a Personal Identification Number 50 (PIN), which is used to form part of the key that protects Media Keys 42 in the Personal Keying Device 30.

Access Vector

An Access Vector 52 is associated with each Media Key 42 stored in a Personal Keying Device 30. The Access Vector 52 is used to represent those possible conditions of access to the data enciphered with that Media Key 42 that may apply to the individual assigned to that Personal Keying Device 30.

Media Attributes

Media Attributes 54 are associated with each element of Media 2 or 4 to which a Media UID 46 has been assigned. Media Attributes 54 are used to represent the sensitivity or other security related information that may pertain to the data on that element of media.

User Attributes

A set of "User Attributes" 56 are associated with each user to which a User UID 48 has been assigned. User Attributes 56 are used to represent the privileges and other security related information which pertains to that user.

Device Attributes

Device Attributes 58 are assigned to each Crypto Media Controller 26, and reflects the Security Attributes 57 of the machine in which the Crypto Media Controller 26 is installed. Device Attributes 58 are combined with Access Vectors 52 to set limits on media access (e.g., read only). Device Attributes 58 are typically defined by the physical security measures which surround the Workstation 10 in which the Crypto Media Controller 26 is installed. For example, a Workstation 10 installed in an open environment may have Device Attributes 58 set to "Authorized to Process Public Data Only", whereas one in a closed engineering facility may have Device Attributes 58 set to "Authorized to Process Proprietary Engineering Data."

Requests

Requests 60 are transmitted back and forth between the Crypto Media Controller 26 and Security Server 24 in the course of operations which require cooperation between the two devices. Requests 60 contain a variety of information depending on the nature of the operation being performed as well as optional integrity fields such as cyclic redundancy checks or check sums.

Countersigns

The purpose of the Countersign 62 logic is to prevent malicious code in the Workstations 10 from masquerading as the Security Server 24, and thereby duping users 5 into taking inappropriate actions. Each time a user 5 is identified to the Security Server 24 (e.g., each new session), the Security Server 24 generates a "fresh" Countersign 62. Countersigns 62 are words, symbols, or phrases which are easy to remember and which are generated by some process which makes it computationally infeasible to guess from one Countersign 62 what the value of the next one will be. The Countersign 62 for a session is presented by the Security Server 24 as a header to each message it sends to the user 5 when communicating over a Trusted Path. The present invention also provides a "Trusted Path." A Trusted Path is a logical communications path between a human user 5 and the Secure Computer 24 (Figure 3). A Trusted Path differs from other modes of communication in that there is a high degree of assurance on the part of both parties that the communication is authentic; that is, the human user is truly seeing what the secure computer intends the human user to see, and the secure computer is making decisions on the basis of precisely what the human user has transmitted to it.

The Countersign 62 is displayed to the user 5 on the Personal Keying Device 30 when the Trusted Path is in effect, and is protected from the Workstations 10 and the communications media by cryptography and is computationally infeasible to guess. It's presence on the display of the Personal Keying Device 30 is a positive indication to a user that the communication in which the

EX1010, Page 1226

user is engaged, is taking place over a Trusted Path to the Security Server 24.

Countersigns 62 are arranged so that the logic in the Security Server 24 can, for any given Countersign 62, determine what the previous Countersign 62 in the sequence was. That is, given a Countersign 62, the Security Server 24 can compute or retrieve a correct value of the previous one, which is called the "last countersign" 62'.

OPERATION OF DATA ENCLAVE 20

The present invention makes use of cryptography to protect the data on Media 2 or 4 and uses an innovative method to distribute and protect the cryptographic keys in order to achieve security, flexibility, and ease of use. The same cryptographic services are used to prevent unauthorized access through the Wide Area Network 16, or the unauthorized use of privileged services.

As described in more detail below, protection of the data on Media 2 or 4 takes place in three broad phases. The first phase, which is done very infrequently, is media initialization and key assignment to the individual user 5 requesting the initialization. The second phase, which is also infrequently done, is the assignment of a key for already-initialized Media 2 or 4 to additional individuals. The third phase, which is done more frequently, is the keying of devices, so access to the data may be made.

Media Initialization and Key Generation

The media initialization and key generation phase generates a Media Key 42 and an Access Vector 52 for a unit of Media 2 or 4 and places them in enciphered form in the Personal Keying Device 30 assigned to the individual requesting the initialization. This data is also archived in the Security Server 24 so that it may be restored at a later time.

Key Assignment

The key assignment phase assigns a Media Key/Access Vector pair, or combination, for an already-initialized unit of media to a new individual. The Media Key 42 will be a copy of the one generated when the unit of Media 2 or 4 was initialized. The Access Vector 52, since it depends on User Attributes 56 as well as Media Attributes 54, will be newly computed.

Keying of Devices

The keying of devices phase automatically extracts the proper Media Key/Access Vector combination from the Personal Keying Device 30, decrypts them and uses them to allow controlled access to the unit of Media 2 or 4. The Media Key/Access Vector combination are enciphered with a Combined Key 44 which includes the user's PIN 50. This restricts a particular Media Key/Access Vector combination to the individual to whom it was assigned.

Media Initialization and Key Generation

The operations in the Media Initialization and Key Generation Phase occur when a blank unit of Media 2 or 4 is to be prepared for safe use in the Enclave 20. This preparation involves initializing the Media 2 or 4, assigning a Media UID 46 to it, generating a Media Key 42 which is unique to that unit of media, and assigning a Media Key/Access Vector pair to the user 5, initializing the media.

The operations in this phase are keyed to the diagrams in Figure 7 through Figure 13. The logic used to implement the Trusted Path facilities is omitted from these diagrams.

Step 1 (Figure 7)

An individual brings together a blank unit of physical Media 2 or 4 and his or her Personal Keying Device 30 to a Workstation 10 which is equipped with a Crypto Media Controller 26 and attached to a Local Area Network 12. If the Media 4 is removable, this is done by carrying Media 4 and Personal Keying Device 30 to an appropriate Workstation 10. If Media 4 is permanently installed (Fixed Media 2), Personal Keying Device 30 is brought to the Workstation containing the fixed media controlled by Crypto Media Controller 26, and the Workstation 10 is temporarily attached to the Local Area Network 12.

Step 2 (Figure 7)

The individual user 5 desiring access to Media 2 or 4 then enters his or her PIN 50 into Personal Keying Device 30 which transmits it to Crypto Media Controller 26, where it is stored for use in later steps.

Step 3 (Figure 7)

Crypto Media Controller 26 then extracts the encrypted User UID 48' from their Personal Keying Device 30, decrypts the User UID 48 using the Enclave Key 40, and stores it for use in later steps.

Step 4 (Figure 8)

Crypto Media Controller 26 forms a packet consisting of the PIN 50, the User UID 48, and a Request 60 for media initialization. The request field will include the nature of the request and appropriate supporting data such as the Security Attributes 57 to be assigned to Media 2 or 4. Key Management Crypto 70 in Crypto Media Controller 26 enciphers it using the Enclave Key 40, and transmits it across the Local Area Network 12 to Security Server 24.

EX1010, Page 1227

Step 5 (Figure 9)

Security Server 24 receives the encrypted packet 90, decrypts it using its copy of the Enclave Key 40, and stores the PIN 50, User UID 40, and Request 60 for use in later steps.

Step 6 (Figure 10)

Storage Search Logic 72 in Security Server 24 uses the User UID 48 to index User Attribute Data Base 80, which returns a pass value if the PIN 50 entered by the user 5 in Step 1 is the same as that stored in the data base, i.e., a valid PIN 50. User Attribute Data Base 80 returns a fail value if the PIN 50 entered by the user is invalid. A fail value will cause the initialization process to abort and a notification to be sent back to Crypto Media Controller 26, which will display it to the user 5 in an appropriate fashion. The abort sequence is not diagrammed in the figures.

Step 7 (Figure 10)

Storage Search Logic 72 extracts the Media Attributes 54 from the Request and commands Media Attribute Data Base 82 to make an entry for the new element of Media 2 or 4. Since Media Attribute Data Base 82 is indexed by the Media UID 46, this has the effect of creating a new Media UID 46 which is sent to Crypto Media Controller 26 and saved for use in later steps.

Step 8 (Figure 11)

Storage Search Logic 72 uses the User UID 48 to index User Attribute Data Base 80 and extract the set of Security Attributes 57 pertaining to this user, and passes these attributes to Security Policy Logic 86.

Step 9 (Figure 11)

Security Policy Logic 86 accepts the Media Attributes 54 and User Attributes 56, and, using a set of rules defined by the administrators of the facility, computes an Access Vector 52 which defines limits on the access this user 5 may have to this unit of Media 2 or 4. This computation may involve the intervention of administrative personnel to authorize or deny the granting of certain privileges.

Step 10 (Figure 12)

Key Management Crypto 70, with the optional aid of authorized individuals, then generates a Media Key 42 for this unit of Media 2 or 4. The manner of generation can involve computation, access to stored tables, requests for inputs from authorized individuals, or any combination thereof. Other methods of key generation may also be used. The Media Key 42 and Access Vector 52 pair 91 are enciphered with a combined key 44 consisting of the User UID 48, the user's PIN 50 and the Enclave Key 40.

Step 11 (Figure 12)

The enciphered packet is sent to Storage Search Logic 72 where the User UID 48 and Media UID 46 are used to store the enciphered packet 92 in Crypto Key Data Base 84. The Media UID and the enciphered packet 92 are transmitted along the LAN 12 to Crypto Media Controller 26.

Step 12 (Figure 13)

The Media UID 46 arrives at Crypto Media Controller 26 and is written to the appropriate location on Media 2 or 4 (e.g., Volume Label).

Step 13 (Figure 13)

The enciphered Media Key/Access Vector pair packet 92 arrives at Crypto Media Controller 26 and the Media UID 46 is used as an index to store the enciphered pair packet 92 in Personal Keying Device 30.

At this point the initialization process is complete. The media can be identified and the individual Personal Keying Device 30 contains a Media Key 42 which can only be used by someone who has physical possession of that Personal Keying Device 30, knows that individual's PIN 50, and has the Media 2 or 4 controlled by a Crypto Media Controller 26 containing the Enclave Key 40. The individual's Personal Keying Device 30 also contains an Access Vector 52 which defines further restrictions on access in a manner that is specific to the individual who has physical possession of that Personal Keying Device 30 and knows that individual's PIN 50.

Key Assignment

The operations in the Key Assignment Phase of the invention occur when an already-initialized unit of Media 2 or 4 is to be shared with a user 5 other than the one who initialized it. In this case, the unit of Media 2 or 4 has a Media Key 42 generated for it, and a Media Key/Access Vector pair 91 has been assigned to the initial user of the unit Media 2 or 4. The necessary steps are to copy the Media Key/Access Vector pair 91 to the new user 5.

The operations in this description are keyed to the diagrams in Figure 14 through Figure 18. The logic used to implement the Trusted Path facilities is omitted from these diagrams.

Step 1 (Figure 14)

An individual brings together a unit of physical Media 2 or 4 and his or her Personal Keying Device 30 to

EX1010, Page 1228

a Workstation 10 which is equipped with Crypto Media Controller 26, and which is attached to the Local Area Network 12. If Media 2 or 4 is removable, this is done by carrying Media 4 and their Personal Keying Device 30 to an appropriate Workstation 10. If Media 2 or 4 is permanently installed (fixed media), Personal Keying Device 30 is brought to the computer containing the fixed Media 2 controlled by Crypto Media Controller 26.

Step 2 (Figure 14)

The individual desiring access to Media 2 or 4 then enters his or her PIN 50 into Personal Keying Device 30 which transmits it to Crypto Media Controller 26, where it is stored for use in later steps.

Step 3 (Figure 14)

Crypto Media Controller 26 then extracts the encrypted User UID 48 from Personal Keying Device 30, decrypts the User UID 48 using the Enclave Key 40 and stores it for use in later steps.

Step 4 (Figure 14)

Storage Search Logic 72 in Crypto Media Controller 26 then reads the Media UID 46 off Media 2 or 4 and searches Personal Keying Device 30 for a Media Key/ Access Vector pair 91 for this unit of Media 2 or 4 for this user 5. Finding none, it generates a Request 60 for key assignment.

Step 5 (Figure 15)

Key Management Crypto 70 forms a request packet 94 consisting of the PIN 50, User UID 48, Media UID 46 and Request 60, encrypts it with the Enclave Key 40, and transmits it over the Local Area Network 12 to Security Server 24.

Step 6 (Figure 16)

Security Server 24 receives the encrypted packet 94, decrypts it using its copy of the Enclave Key 40, and stores the PIN 50, User UID 48, Media UID 46 and Request 60 for use in later steps.

Step 7 (Figure 16)

Storage Search Logic 72 in Security Server 24 uses the User UID 48 to index User Attribute Data Base 80. User Attribute Data Base 80 returns a pass value if the PIN 50 entered by the user 5 was the same as that stored in the data base (i.e. valid). User Attribute Data Base 80 returns a fail value if the PIN 50 entered by the user is invalid. A fail value will cause the assignment process to abort and a notification to be sent back to Crypto Media Controller 26, which will display it to the user in an appropriate fashion. The abort sequence is not diagrammed in the figures.

Step 8 (Figure 16)

The User UID 48 is used as an index into User Attribute Data Base 80 by Storage Search Logic 72, and the Security Attributes 57 of the user 5 requesting key assignment are extracted and passed to Security Policy Logic 86.

Step 9 (Figure 16)

The Media UID 46 is used as an index into Media Attribute Data Base 82 by Storage Search Logic 72, and the Security Attributes 57 of the denoted item of Media 2 or 4 are extracted and passed to the Security Policy Logic 86.

Step 10 (Figure 16)

Security Policy Logic 86 accepts these Attributes 57, and, using a set of rules defined by the administrators of the facility, computes an Access Vector 52 which defines limits on the access this user 5 may have to this unit of Media 2 or 4. This computation may involve the intervention of administrative personnel to authorize the granting or denying of certain privileges. This Access Vector 52 is saved for use in later steps.

Step 11 (Figure 17)

The Media UID 46 is used by Storage Search Logic 72 to find an enciphered key packet in Crypto Key Data Base 84 which has been previously stored and which contains a Media Key 42 for this unit of media. Since the Media 2 or 4 has been initialized and assigned a Media UID 46, then at least one such packet must exist. Any such packet will suffice, since all packets pertaining to a given unit of Media 2 or 4 will contain the same Media Key 42. When such a packet is found, the Media Key 42 is extracted from it for use in later steps.

Step 12 (Figure 17)

A new Key Packet 93 is formed consisting of the Media Key 42, Access Vector 52, User UID 48, and Media UID 46 and placed in Crypto Key Data Base 84 for archival storage and retrieval.

Step 13 (Figure 17)

The Media Key and Access Vector pair 91 are enciphered with a Combined Key 44 consisting of the User UID 48, the user's PIN 50, and the Enclave Key 40, and the enciphered packet 92 is transmitted along the LAN 12 to Crypto Media Controller 26.

EX1010, Page 1229

Step 14 (Figure 18)

The Media UID 46 is used as an index to store the enciphered Media Key/Access Vector pair 91 in Personal Keying Device 30.

At this point the new individual's Personal Keying Device 80 contains a Media Key 42 which can only be used by someone who has physical possession of that Personal Keying Device 30, knows that individual's PIN 50, and has the Media 2 or 4 controlled by a Crypto Media Controller 26 containing the Enclave Key 40. The individual's Personal Keying Device 30 also contains an Access Vector 52, which defines further restrictions on access in a manner that is specific to the individual who has physical possession of that Personal Keying Device 30 and knows that individual's PIN 50.

Keying of Devices

The operations in the Keying of Devices Phase occur when a Media Key/Access Vector pair 91 for a unit of Media 2 or 4 has been assigned to a user 5, and that user 5 wants to exercise the assigned accesses. The steps in this description are keyed to the diagrams in Figures 19 and 20. The logic used to implement the Trusted Path facilities is omitted from these diagrams.

Step 1 (Figure 19)

An individual user 5 establishes a data transfer interface between his or her Personal Keying Device 30 and any Crypto Media Controller 26 containing the Enclave Key 40, and between that Crypto Media Controller 26 and Media 2 or 4 the individual user 5 desires to access. In the latter case, this will involve placing the unit of Media 4 into the appropriate device (e.g., diskette drive).

Step 2 (Figure 19)

The individual user 5 desiring access to Media 2 or 4 then enters his or her PIN 50 into Personal Keying Device 30 which transmits it to Crypto Media Controller 26, where it is stored for use in later steps.

Step 3 (Figure 19)

Storage Search Logic 72 in Crypto Media Controller 26 reads the Media 2 or 4 and extracts the Media UID 46.

Step 4 (Figure 19)

Using the Media UID 46, Storage Search Logic 72 searches Storage 78 in Personal Keying Device 30 and extracts the enciphered Media Key/Access Vector pair packet 92 and passes it to Key Management Crypto 70.

Step 5 (Figure 19)

The enciphered User UID 48' is fetched from Personal Keying Device 30 and deciphered using the Enclave Key 40.

Step 6 (Figure 19)

The User UID 48, PIN 50, and Enclave Key 40 are then combined to form the Combined Key 44 to decrypt the Media Key/Access Vector packet 92. The Media Key 42 is passed to Data Crypto 74, and the Access Vector 52 is passed to Access Control Logic 76.

Step 7 (Figure 20)

Workstation's 10 internal logic makes a request for data. That logic need not be aware the data is protected by cryptography. The request illustrated in the figure is a "read" request, but the handling of "write" requests are symmetric.

Step 8 (Figure 20)

Enciphered data 3' is then fetched from Media 2 or 4.

Step 9 (Figure 20)

Data Crypto 74 deciphers the data using the Media Key 42 and passes data 3 to the Access Control Logic 76.

Step 10 (Figure 20)

Access Control Logic 76 consults the Access Vector 52 and the Device Attributes 58 contained within itself and decides whether the desired mode of access ("read," "write," etc.) shall be permitted. If not, the data transfer is aborted and an error indication is sent to the Workstation 10.

At this point the data has been transferred to the Workstation 10 for processing. Removal of the Media 2 or 4 or the Personal Keying Device 30 from the Crypto Media Controller 26 will cause the complete reset of the Crypto Media Controller 26 and require the keying process be started from the beginning.

Trusted Path

Identification and Authorization

This phase of the operation involves the steps whereby a user 5 presents his or her identity to the Security Server 24 and has that identity authenticated and a set of privileges associated with the user 5 at the Security Server 24.

This operation is protected against forged identities

EX1010, Page 1230

and authentications, and so-called "replay" attacks in which malicious software in other Workstations 10 masquerades as the authentications mechanism, accepts identification and authorization data (such as passwords) from an unwitting user 5, and then passes that data to an unauthorized individual.

The operation is also protected against compromise of the authentication data in the Personal Keying Device 30. The invention uses the Countersign logic to effect this protection. It will be recalled that Countersigns 62 come in a sequence which is generated by the Security Server 24, but which is computationally infeasible for an outsider to guess. Thus, for each Countersign 62, the Security Server 24 (but no one else) can determine the value of Last Countersign 62'.

The Last Countersign 62' for a given is stored in a distinguished location in that user's Personal Keying Device 30. At each identification and authentication interaction the Last Countersign 62' is extracted from the Personal Keying Device 30 and compared with the Last Countersign 62' independently generated or retrieved by the Security Server 24. If the two values are unequal then it is known that the identification and authentication process has been compromised and suitable alarms are raised.

The manner in which this mechanism operates can be made clear from example. Assume that the sequence of Countersigns 62 is "A," "B," "C," etc. Further assume that a given user's Personal Keying Device 30 contains the Last Countersign 62' value "A". Since it is computationally infeasible for an attacker to guess this value, the attacker's recourse is to either steal the Personal Keying Device 30 or copy the data from it.

If the attacker steals the Personal Keying Device 30, then its absence will be noted and alarms will be raised. If the attacker copies the Last Countersign 62' and by some subterfuge succeeds in being authenticated as the legitimate user 5, then the identification and authentication process will update the Last Countersign 62' value in the spurious Personal Keying Device 30 to "B." When the legitimate user 5 attempts identification and authentication, the Last Counterside 62' in his or her Personal Keying Device 30 will still be at "A"; the difference will be noted by the Security Server 24 and alarms raised.

Thus, the copying and successful use of data from a Personal Keying Device 30 will enable a false identity to be presented to the Security Server 24 only until the time at which the legitimate user 5 attempts identification and authentication.

The steps involved in this phase of the operation are keyed to the diagrams given in Figure 21 through Figure 24. The logic used in data protection is omitted from these diagrams.

Step 1 (Figure 21)

The User UID 48, encrypted with the Enclave Key (48') is extracted from the user's Personal Keying Device 30.

Step 2 (Figure 21)

The Last Countersign 62' (denoted "Old C/S" in Figure 21), encrypted with the Enclave Key 40, is extracted from the user's Personal Keying Device 30.

Step 3 (Figure 21)

The user 5 desiring access to operations on the Security Server 24 then enters his or her PIN 50 through the keyboard on the Personal Keying Device 30.

Step 4 (Figure 21)

The User UID 48' and Last Countersign 62' are decrypted, combined with the PIN 50, and re-encrypted with the Enclave Key 40 for transmission to the Security Server 24.

Step 5 (Figure 22)

The combined Last Countersign 62', PIN 50, and User UID 48 are decrypted using the Enclave Key 40 and passed to the storage search logic 72. That logic searches the User Attributes Data Base 80 for the authentication record belonging to this user 5, compares the User UID/PIN combination 92 that was entered against the stored value, and checks the Last Countersign 62' from the Personal Keying Device 30 against the stored value from the previous identification and authentication interaction. Based on these checks the logic computes a Result 94 (e.g., "Login Successful," "Login Failed") and in the case of successful identification, a set of privileges which that user may exercise in future interactions with the Security Server 24. Also in the case of successful identification, the next Countersign 62 in the sequence is generated, stored in the User Attribute Data Base 80 as the new Last Countersign 62' and saved for use in the next step. This value is denoted "New C/S" in the figures.

Step 6 (Figure 23)

The Result 94 and the updated Countersign 62 value is encrypted with the Enclave Key 40 and transmitted to the Crypto Media Controller 26.

Step 7 (Figure 24)

The combined Result and updated Countersign 62 is decrypted. The updated Countersign 62 is encrypted with the Enclave Key 40 and stored in the user's Personal Keying Device 30 as the new value of Last Countersign 62'. The Countersign and result are displayed on the display portion of the Personal Keying Device 30.

EX1010, Page 1231

At this point, the user has been authenticated to the Security Server 24 and assigned a set of Privileges 95, which may be invoked at a later time. The Security Server 24 has also displayed to the user 5 the Countersign 62 that it will use in the session to authenticate itself to the user.

Privileged Services

This phase of the operation involves a user 5, whose identity has already been presented to and authenticated by the Security Server 24, invoking a privileged operation by that Server 24. The user is identified to the Security Server 24 by the User UID 48. The Security Server 24 is authenticated to the user by the Countersign 62.

The steps involved in this phase of the operation are keyed to the diagrams given in Figure 25 to Figure 28. The logic used in data protection is omitted from these diagrams.

Step 1 (Figure 25)

The user 5 signals his or her desire to invoke a privileged operation by an appropriate entry in the keyboard 34 of the Personal Keying Device 30. This entry is shown as "ATTN" in the Figures. The User UID 48 is then extracted from the Personal Keying Device 30.

Step 2 (Figure 25)

The combination of the "ATTN" signal and the User UID 48 is encrypted with the Enclave Key 40 and transmitted to the Security Server 24.

Step 3 (Figure 26)

The combination of the "ATTN" signal and the User UID 48 is decrypted using the Enclave Key 40.

Step 4 (Figure 26)

The User UID 48 is transferred to the Storage Search Logic 72 and the "ATTN" signal is transferred to the Privileged Operation Logic 73.

Step 5 (Figure 26)

The Storage Search Logic 72 then extracts the user's Privileges 95 from the User Attribute Data Base 80 and passes them to the Privileged Operation Logic 73.

Step 6 (Figure 27)

The Storage Search Logic 72 extracts the Countersign 62 from the User Attribute Data Base 80 and passes it to the Key Management Crypto 70, which encrypts it with the Enclave Key 40 and transmits it to the Crypto Media Controller 26, which initiated the request.

Step 7 (Figure 28)

The Crypto Media Controller 26 decrypts the Countersign 62 and causes it to be displayed on the Personal Keying Device 30.

At this point, both the user and the Security Server 24 are aware, in authenticated fashion, that a privileged operation is to be invoked. The invocation of the operation, which may involve multiple interactions, can then proceed. The operation is terminated by a series of steps which is symmetric to those presented above.

An alternate, preferred embodiment of the Trusted Path is described further below, with reference to Figures 29 - 34. The Trusted Path phase of the Data Enclave process is preferably implemented using the relevant aspects of this alternate embodiment. These aspects include Identification and Authentication, Trusted Command Initiation (Privileged Services) and Key Management.

ADVANTAGES OVER PRIOR ART

The Data Enclave System of the present invention provides a number of advantages over the prior art, as outlined below.

Security

The data enclave invention offers comprehensive security to the data within the Enclave 20; there are no "sneak paths" or "holes" that exist in approaches where the data is protected on media but the Wide Area Network 16 connections are open, or vice versa.

The invention minimizes the damage that can be done by privileged individuals who become subverted. Cryptographic keys are transmitted and stored entirely in enciphered form. Well-known techniques (so-called "antitamper" technology) can be used to protect the Enclave Key when it is stored in the Crypto Media Controllers 26 and the Security Server 24. Theft of elements of the invention such as the Personal Keying Device 30 and the Crypto Media Controllers 26 does not compromise any part of the operation of the invention.

Low Cost

The invention uses a small number of special elements in a wide variety of ways. Maximum use is made of the cryptographic devices, which are typically the most expensive parts of a data security device. The same devices are used for media protection and authenticated interactions with the Security Server.

Ease of Use

Individuals desiring access to media have to deal

EX1010, Page 1232

with the Security Server only when media is initialized. "Unlocking" a unit of media requires an operation no more complicated than using a TV remote control. Overhead and delay is concentrated at the time a media is "unlocked" and no delays or incompatibilities are introduced during operations using the media.

Identification and authentication of users to the Security Server 24 is both simpler and more robust than prior art such as passwords. The same basic steps are used for security operations dealing with media and dealing with the Security Server 24.

Exceptional or emergency situations can be accommodated. A trusted command initiation can override a security policy enforced by the Security Server 24 and release data to persons who would normally be unauthorized to access it.

Flexible Control of Media

In the data protection area, the system associates Media 2 or 4 primarily with users and secondarily with machines. This is a more natural structure than one where Media 2 or 4 is only useable on a single machine.

The access control logic, which computes allowed access at the last possible moment using the combination of an individual's Access Vector 52 and the Device Attributes 58 assigned to a particular Workstation, can be used to enforce a variety of security policies. For example, an individual's access to data may be restricted not only on the basis of the individual's attributes, but also to protected physical locations. Thus, an individual's Access Vector 52 may grant "read" access to a unit of media which contains proprietary engineering data, but the comparison against the Device Attributes 58 of the Crypto Media Controller 26 making the access may restrict display of the contents of the unit of media to those machines inside a particular facility or office. Physical security measures can then be used to restrict who may be in the vicinity when the data is displayed. Prior art in this area permits only an "all or nothing" approach to access.

Sharing and Backup of Media

An individual's access to an initialized media can be restored, or a second individual granted access, by bringing together the media, the requisite Personal Keying Device 30, and a Workstation 10 equipped with a Crypto Media Controller 26 that is keyed with the appropriate Enclave Key.

Positive Control of Privileged Operations

Remotely invoked privileged operations at the Security Server 24 are under the positive control of the user 5. That control is cryptographically protected and mutually authenticated.

ALTERNATE EMBODIMENT OF DATA ENCLAVE SYSTEM

An alternate embodiment of the Data Enclave System 20 is shown in Figs. 29, 30 and 31. Alternate embodiment 300 provides for operation of the Data Enclave System in a non-networked environment.

Data Elements

The data elements of the alternate embodiment 300 correspond to those described with reference to embodiment 20.

Processing Elements

Crypto Support Center

A Crypto Support Center 310 is provided for each organization or set of organizations. The Crypto Support Center 310 is used for archival storage and distribution of cryptographic keys. Crypto Support Center 310 is permanently installed in a secure area, and includes a Secure Computer 311 and a Communications Security Device 312. Secure Computer 311 may be of generally the same design as Security Server 24 as described and illustrated with reference to embodiment 20. However, there is no requirement that the Secure Computer 311 be networked to the work stations 340 within the organization.

Local Crypto Support Device

There is at least one local Crypto Support Device 320 for each organization. Each local Crypto Support Device 320 is portable, for example, lap-top computer size. Preferably, local Crypto Support Devices 320 are equipped with theft detection circuitry, such as that used to deter shoplifting. Local Crypto Support Devices 320 are used in key distribution and are equipped with a Communications Security Device 322 that is compatible with the Communication Device 312 in Crypto Support Center 310. Local Crypto Support Device 320 includes a Key Management Crypto 324 which functions substantially the same as the Key Management Crypto 70 described with reference to the embodiment 20 of the data enclave system, insofar as media initialization, key generation and key assignment are concerned. Crypto Support Devices 320 further include a disk drive 326, which may be used to read and write removable media 302, and a data interface 328, which may be coupled to a Crypto Media Controller in a Workstation 340. The interface can either be wired or wireless (for example, radio infra-red).

Personal Keying Device

Each user is issued a Personal Keying Device 330

EX1010, Page 1233

of the same design as Personal Keying Device 30 described above with reference to embodiment 20 of the Data Enclave System. Personal Keying Device 330 is used for key insertion and individual authentication. Personal Keying Device 330 includes electronic storage 331, a key pad, a display and a Data Transfer Interface 332, which is compatible with the Data Transfer Interface in the local Crypto support device 320. Personal Keying Devices 330 may also be equipped with theft detection circuitry.

Crypto Media Controller

Each work station 340 operating within the enclave 300 includes a Crypto Media Controller 342 of the same design as Crypto Media Controller 26, with the exception that Crypto Media Controller 342 does not include logic and functions for media initialization and key generation, or key assignment for already initialized media. Crypto Media Controller 342 further includes a Data Interface 344 compatible with Data Interface 328 in the Local Crypto Support Device 320.

OPERATION OF ALTERNATE EMBODIMENT 300

Alternate embodiment 300 is similar in many respects to embodiment 20, except that Local Crypto Support Device 320 and Crypto Support Center 310 perform certain functions performed by Crypto Media Controller 26 and Security Server 24, respectively, embodiment 20. Namely, those functions described in Steps 1 - 13 of the Media Initialization and Key Generation and the Key Assignment process (for initialized media) Steps 1 - 14 of embodiment 20. In addition, the Local Area Network 12 link used in embodiment 20 is replaced with the secure connection established between Communications Security Devices 312 and 322 in the Local Crypto Support Device 320 and Crypto Support Center 310.

Media Initialization and Key Assignment

The following description of the media initialization and key assignment operation refers to Figs. 30 and 31.

An individual brings together a blank unit of physical media 302, his or her Personal Keying Device 330, and the appropriate Local Crypto Support Device 320. If the media is fixed, Personal Keying Device 330 and local Crypto support device 320 are brought to the Workstation 340 containing the fixed media 302. As shown in Fig. 30, data interfaces are then established between Personal Keying Device 330 and Local Crypto Support Device 320 on the one hand and in between Local Crypto Support Device 320 and the Crypto Media Controller 342 for the fixed media on the other. Once these interfaces are established, a secure link is made between Local Crypto Support Device 320 and Crypto Support Center 310 using the Communication Security Devices 312 and 322. The Trusted Path Protocol of the present invention may be used to establish a secure link.

If the media 302 is removable, the media 302 is brought to the Local Crypto Support Device 320, where it can be read and written using Disk Drive 326. This configuration is shown in Figure 31.

The individual desiring access to Media 302 then enters his or her PIN 58 into Personal Keying Device 330 which transmits it to Local Crypto Support Device 320. Local Crypto Support Device 320 extracts the encrypted User UID 56 from Personal Keying Device 330 and decrypts it using the Enclave Key 50.

Local Crypto Support Device 320 then initiates a secure connection to the Crypto Support Center 310 and transmits the User UID 56 to it.

Local Crypto Support Device 320 and the Crypto Support Center 310, with the optional aid of authorized individuals, generate a Media UID 54, Media Key 52, and Access Vector 60 for use of the media 302. At the end of this process, the Media UID 54, Media Key 52, User UID 56, and Access Vector 60 are archived together at the Crypto Support Center 310 and stored temporarily in Local Crypto Support Device 320.

Local Crypto Support Device 320 then writes the Media UID 54 to an appropriate location on Media 302 (e.g., Volume Label). It combines the User UID 56, Enclave Key 50, and PIN 58 to form a key with which it enciphers the Media Key/Access Vector pair 62. It uses the Media UID 54 to index storage 332 of Personal Keying Device 330 and stores the enciphered pair 62 in the appropriate location.

At this point, the initialization is complete. Media 302 can be identified and the individual's Personal Keying Device 330 contains a Media Key 52 which can only be used by an individual who has physical possession of that Personal Keying Device 330, knows that individual's PIN 50, and has Media 302 controlled by a Crypto Media Controller 342, containing the Enclave Key.

Keying of Devices

An individual establishes a data transfer interface between his or her Personal Keying Device 330 and any Crypto Media Controller 342 containing the Enclave Key, and between that Crypto Media Controller 342 and the Media 302 the individual desires to access. If the media 302 is removable, this will involve placing the unit of media 302 into the appropriate device (e.g. diskette drive) or the Workstation 340. From this point on, the alternate embodiment 300 operates in the same manner as the first described Data Enclave embodiment 20, as set forth in Steps 1-10 under the heading "Keying of Devices."

Key Assignment for Already Initialized Media

Key assignment is performed in substantially the same fashion as Media Initialization and Key Generation, insofar as the configuration and interaction of the

EX1010, Page 1234

Personal Keying Device 330, Workstation 340, Local Crypto Support Device 330 and Support Center 310 interact to generate a Media Key/Access Vector pair 91 for the already initialized media 302 by reference to the archived Media Key 42 for the media.

The present invention is to be limited only in accordance with the scope of the appended claims, since others skilled in the art may devise other embodiments still within the limits of the claims. The above-described detailed architectures are not meant to be limiting, and other equivalent forms may be substituted if desired.

Claims

1. A data enclave (20) for securing data carried on physical units of fixed (2) and removable (4) media, the data enclave (20) including a security server (24) connected over a network (42) to one or more workstations (10), wherein each workstation (10) includes a crypto media controller (26) used to read one of said physical units of media (2,4), the data enclave further comprising:

   an enclave key (40) used to encrypt data transmitted within the data enclave (20), wherein a copy of the enclave key (40) is stored in the security server (24) and the workstations (10);
   a personal keying device (30) for each user in the data enclave (20);
   a personal identification number (PIN) (50) and a user unique identifier (user UID) (48) assigned to each user in the enclave (20), wherein each user UID (48) is encrypted with the enclave key and stored in the personal keying device (30) of the user associated with the user UID;
   a set of user attributes (56) provided for each user, wherein each set of user attributes (56) represents user privileges and other security related information pertaining to a particular user and wherein each set of user attributes (56) is associated with the user UID (48) of its respective user;
   a media key (42) for each physical unit of media (2,4), wherein the media key (42) is used to encrypt and protect data carried on the media;
   a media unique identifier (media UID) (46) for each physical unit of media (2,4); and
   a set of media attributes (54) provided for each physical unit of media (2,4), wherein each set of media attributes (54) represents sensitivity or other security related information pertaining to data carried on a particular unit of media and wherein each set of media attributes (54) is associated with the media UID (46) of its respective physical unit of media (2,4); characterised in that

   the security server (24) comprises:

   security policy logic (86) for computing, from the set of user attributes assigned to a particular user (5) and the set of media attributes assigned to a particular unit of media (2,4), an access vector (52) which defines limits on access by the particular user (5) to the particular unit of media (2,4); and
   a key management crypto (70) for combining the access vector (52) and the media key (42) assigned to the particular unit of media (2,4) to form a media key/access vector pair (91) and for enciphering the media key/access vector pair (91) with a combined key formed from the enclave key (40) and the user UID (48) and PIN (50) of the particular user (5);
   wherein the personal keying device (30) comprises means (78) for storing the enciphered media key/access vector pair (91); and
   wherein the crypto media controller (26) comprises means (70, 72, 76) for controlling access to data on the particular unit of media (2,4) as a function of the PIN (50) of the particular user (5), the media UID (46) of the particular physical unit of media (2,4) and the media key/access vector pair (91) retrieved from the personal keying device (30) of the particular user (5).

2. A data enclave method for securing data carried on physical units of fixed (2) and removable (4) media in a data enclave (20) including a security server (24) connected over a network (12) to one or more workstations (10), wherein each workstation (10) includes a crypto media controller (26) used to read one of said physical units of media (2,4), the method comprising the steps of:

   providing an enclave key (40) used to encrypt data transmitted within the data enclave (20);
   storing a copy of the enclave key (40) in the security server (24) and the workstations (10);
   providing a personal keying device (30) for each user in the data enclave (20);
   assigning a personal identification number (PIN) (50) and a user unique identifier (user UID) (48) to each user in the enclave (20);
   assigning a set of user attributes (56) for each user, wherein each set of user attributes (56) represents user privileges and other security related information pertaining to a particular user;
   associating each set of user attributes (56) with the user UID (48) of its respective user;

EX1010, Page 1235

encrypting each user UID (48) with the enclave key and storing each encrypted user UID (48') in the personal keying device (30) of the user associated with the user UID (48);
assigning a media key (42) and a media unique identifier (media UID) (46) for each physical unit of media (2,4), wherein the media key (42) is used to encrypt and protect data carried on the media;
assigning a set of media attributes (54) for each physical unit of media (2,4), wherein each set of media attributes (54) represents sensitivity or other security related information pertaining to data carried on a particular unit of media; and
associating each set of media attributes (54) with the media UID (46) of its respective physical unit of media (2,4); and characterised by
computing, from the set of user attributes assigned to a particular user (5) and the set of media attributes assigned to a particular unit of media (2,4), an access vector (52) which defines limits on access by the particular user (5) to the particular unit of media (2,4);
combining the access vector (52) and the media key (42) assigned to the particular unit of media (2,4) to form a media key/access vector pair (91);
enciphering the media key/access vector pair (91) with a combined key formed from the enclave key (40) and the user UID (48) and PIN (50) of the particular user (5); and
storing the enciphered media key/access vector pair (91) in the personal keying device (30) of the particular user (5); and
controlling access to data on the particular unit of media (2,4) as a function of the PIN (50) of the particular user (5), the media UID (46) of the particular physical unit of media (2,4) and the media key/access vector pair (91) retrieved from the personal keying device (30) of the particular user (5).

3. A method according to claim 2 wherein the method further comprises the step of providing device attributes for each workstation (10), the device attributes representing security attributes of the workstations (10), and wherein the step of controlling access comprises the steps of:

determining the workstation (10) being used by the particular user (5);
retrieving the device attributes (58) associated with the workstation (10) being used by the particular user (5);
extracting the access vector (52) from the encrypted media key/access vector pair (91) retrieved from the personal keying device (30) of the particular user (5); and
combining the retrieved device attributes (58) with the extracted access vector (52) to determine access rights by the particular user (5) on the particular workstation (10).

4. A method according to claim 2 wherein the method further comprises the steps of:

(a) providing key management crypto logic in each crypto media controller for (i) receiving a requesting user's PIN from a personal keying device (ii) receiving an encrypted user UID from the personal keying device and decrypting the user UID using the enclave key, and (iii) forming a first packet including the requesting user's PIN, the user UID and a request for initialization of a new unit of media, the request including the media attributes for the new unit of media;
(b) providing key management crypto logic in the server for decrypting the first packet using the enclave key stored in the server,
(c) providing storage search logic in the server for (i) reading a user attribute data base stored in the server using the user UID as an index, (ii) returning a pass value if the requesting user's PIN received in the first packet matches a valid PIN stored in the user attribute data base, (iii) aborting the request for initialization if the requesting user's PIN is not valid, (iv) extracting the media attributes from the request and commanding a media attribute data base stored in the server to make an entry for the new unit of media, and to create a new media UID for the new unit of media, and (v) indexing the user attribute data base with the user UID to extract the set of security attributes pertaining to the requesting user and passing the security attributes to security policy logic in the server;
(d) the security policy logic accepting the media attributes and the requesting user's security attributes and, using a set of rules and/or under the direction of a system administrator, computing a new access vector which defines limits on the access the requesting user will have to the new unit of media;
(e) the key management crypto in the server also (i) generating, with the optional aid of a system administrator, a new media key for the new unit of media, and (ii) enciphering the new media key/access vector pair formed with the new media key and the new access vector with a combined key including the user UID, the user PIN and the enclave key, to form a second packet;
(f) the storage search logic also storing the enciphered second packet in a crypto key data base stored in the server, the second packet indexed according to the requesting user's user

EX1010, Page 1236

UID and the new media UID;
(g) providing further logic for sending the new media UID and the second packet to the Workstation from which the first packet was received; and
(h) providing storage search logic in the crypto media controller for (i) receiving the new media UID and writing it to an appropriate location on the new unit of media and (ii) storing the second packet containing the new media key/ access vector pair in the personal keying device attached to the Workstation using the new media UID as an index

5. A method according to claim 2 further comprising the steps of:

(a) providing key management crypto logic in each crypto media controller for (i) receiving a requesting user's PIN from a personal keying device, (ii) receiving an encrypted user UID from the personal keying device and decrypting the user UID using the enclave key, and (iii) reading the media UID off an initialized unit of media and searching the personal keying device for a media key/access vector pair for the initialized unit of media for the requesting user using the user's PIN as an index, and (iv) if no pair is found generating a request for a key assignment;
(b) the key management crypto logic in the workstations further (i) forming the first packet including the requesting user's PIN and user UID, the media UID for the initialized unit of media, and the request for key assignment, (ii) encrypting the first packet with the enclave key, and (iii) sending the packet to the security server over the network;
(c) providing key management crypto logic in the server for decrypting the first packet using the enclave key stored in the server to obtain the requesting user's PIN and user UID, and the media UID and the request;
(d) providing storage search logic in the security server for (i) reading a user attribute data base stored in the server using the user UID as an index, (ii) returning a pass value if the requesting user's PIN received in the first packet matches a valid PIN stored in the user attribute data base, (iii) aborting the request for initialization set forth in the first packet if the requesting user's PIN is not valid, (iv) reading the user attribute data base using the user's PIN as an index and extracting the security attributes of the requesting user, and (v) passing the security attributes to security policy logic in the server;
(e) the security policy logic receiving the security attributes and computing a new access vector which defines limits on the access the user may have to the initialized unit of media, the new access vector computed using a set of rules and/or with the intervention of a system administrator;
(f) the storage search logic also (i) finding an enciphered key packet in a crypto key data base held in the security server which has been previously stored and which contains the media key for the initialized unit of media, (ii) when a packet is found extracting the media key from it, and (iii) forming a new media key/access vector pair with the extracted media key and the new access vector, and a new key packet including the new media key/access vector pair, the user UID, and the media UID, and placing the new key packet in the crypto key data base for archival purposes;
(g) the crypto key logic also enciphering the new media key/access vector pair with a combined key including the user UID, the user's PIN, and the enclave key, and transmitting the enciphered packet along the network to the crypto media controller; and
(h) the crypto media controller using the media UID as an index to store the new media key/ access vector pair in the personal keying device from which the user's PIN was entered whereby the personal keying device contains a media key which can only be used by someone who has physical possession of that personal keying device, knows the user PIN associated with the media key, and has physical possession of the unit of media controlled by a crypto media controller containing the enclave key, the access of the user further being restricted by the access vector paired with the media key.

6. A method according to claim 2, further comprising the steps of:

(a) the crypto media controller also (i) receiving a user PIN from a personal keying device from a user seeking access to an initialized unit of media under control of the crypto media controller;
(b) providing storage search logic in the crypto media controller for (i) reading the initialized unit of media and extracting the media UID, (ii) searching the storage in the personal keying device and extracting the enciphered media key/access vector pair for the media UID and passing it to a key management crypto in the crypto media controller;
(c) the key management crypto (i) fetching the user UID from the personal keying device and deciphering it using the enclave key, (ii) com-

EX1010, Page 1237

bining the user UID, the user PIN, and the enclave key to form a combined key to decrypt the media key/access vector pair, and passing the extracted media key to a data crypto and the access vector to the access control logic;
(d) the data crypto deciphering data on a unit of media using the media key and passing it to the access control logic, the data deciphered in response to a read or write request for the data by the Workstation;
(e) the access control logic controlling whether the desired mode of access is permitted based on the access vector and the device attributes contained within the crypto media controller, and aborting the attempted access to the data if the access is not permitted and otherwise permitting the access whereby data is transferred to a Workstation for procession; and
(f) providing logic in the crypto media controller for causing a complete reset of the crypto media controller and requiring the keying process to be started from the beginning in the event that the personal keying device is uncoupled or the unit of media is removed from the Workstation.

Patentansprüche

1. Datenenklave (20) zum Absichern von Daten, die sich auf physikalischen Einheiten, bestehend aus nicht entfernbaren (2) und entfernbaren (4) Medien, befinden, wobei die Datenenklave (20) eine Sicherheitsdiensteinrichtung (Server) (24) umfaßt, die mittels eines Netzwerkes (12) mit einem oder mehreren Arbeitsplatzrechnern (10) verbunden ist, wobei jeder Arbeitsplatzrechner (10) eine Verschlüsselungs-Mediensteuereinheit (26) umfaßt, die eingesetzt ist, um eine der physikalischen Medieneinheiten (2, 4) zu lesen, wobei die Datenenklave (20) weiterhin umfaßt:

einen Enklavenschlüssel (40), der zum Verschlüsseln von Daten eingesetzt ist, die innerhalb der Datenenklave (20) übertragen werden, wobei eine Kopie des Enklavenschlüssels (40) in der Sicherheitsdiensteinrichtung (24) und in den Arbeitsplatzrechnern (10) gespeichert ist;
einen persönlichen Schlüssel (30) für jeden Benutzer innerhalb der Datenenklave (20);
eine persönliche Identifikationsnummer (PIN) (50) und eine für jeden Benutzer einmalige Identifizierung (Benutzer-UID) (48), die jedem Benutzer innerhalb der Datenenklave (20) zugewiesen ist, wobei jede Benutzer-UID (48) mittels des Enklavenschlüssels (40) verschlüsselt ist und in dem persönlichen Schlüssel (30) des Benutzers gespeichert ist, der zu dieser Benutzer-UID (48) zugehörig ist;
eine Reihe von Benutzer-Attributen (56), die für jeden Benutzer vorgesehen sind, wobei jede Reihe von Benutzer-Attributen (56) Benutzer-Privilegien und andere sicherheitsrelevante Informationen darstellt, die einem einzelnen Benutzer zuzuordnen sind und wobei jede Reihe von Benutzer-Attributen (56) der Benutzer-UID (48) des jeweiligen Benutzers zugeordnet ist;
einen Medienschlüssel (42) für jede physikalische Einheit (2, 4) von Medien, wobei der Medienschlüssel (42) eingesetzt ist, um die Daten zu verschlüsseln und zu schützen, die sich auf den Medien befinden;
eine für jedes Medium einmalige Identifizierung (Medien-UID) (46) für jede physikalische Medien-Einheit (2, 4); und
eine Reihe von Medien-Attributen (54), die für jede physikalische Medien-Einheit (2, 4) vorgesehen ist, wobei jede Reihe von Medien-Attributen (54) die Sensitivität oder andere sicherheitsrelevante Informationen darstellt, die Daten zuzuordnen sind, die sich auf einer einzelnen Medieneinheit befinden und wobei jede Reihe von Medien-Attributen (54) der Medien-UID (46) der zugehörigen physikalischen Medieneinheit (2, 4) zugeordnet ist, **dadurch gekennzeichnet**, **daß** die Sicherheitsdiensteinrichtung (24) umfaßt:
eine Sicherheitsleitlogik (86) zur Berechnung eines Zugangsvektors (52), der die Beschränkungen eines Zugriffes des einzelnen Benutzers (5) auf eine bestimmte Medieneinheit (2, 4) definiert, wobei die Berechnung mittels der Reihe von Benutzer-Attributen, die einem bestimmten Benutzer (5) zugeordnet sind sowie der Reihe von Medien-Attributen erfolgt, die einer bestimmten Medieneinheit (2, 4) zugeordnet sind; und
eine Schlüsselverarbeitungs-Verschlüsselungseinrichtung (70) zum Kombinieren des Zugangsvektors (52) und des Medienschlüssels (42), der einer bestimmten Medieneinheit (2, 4) zugeordnet ist, um ein Medienschlüssel/Zugangsvektor-Paar (91) zu erhalten sowie zur Verschlüsselung des Medienschlüssel/Zugangsvektor-Paares (91) mit einem zusammengesetzten Schlüssel, der aus dem Enklavenschlüssel (40), der Benutzer-UID (48) und der PIN (50) des bestimmten Benutzers (5) gebildet ist;
wobei der persönliche Schlüssel (30) Mittel (78) zum Speichern des verschlüsselten Medienschlüssel/Zugangsvektor-Paares (91) aufweist; und
wobei die Verschlüsselungs-Mediensteuereinheit (26) Mittel (70, 72, 76) zur Steuerung des

EX1010, Page 1238

Zugriffes auf Daten umfaßt, die sich auf der bestimmten Medieneinheit (2, 4) befinden, wobei diese Steuerung als Funktion der PIN (50) des bestimmten Benutzers (5), der Medien-UID (46) der bestimmten physikalischen Medieneinheit (2, 4) und des Medienschlüssel/Zugangsvektor-Paares (91), der aus dem persönlichen Schlüssel (30) des bestimmten Benutzers (5) erhalten wird, ausgeführt ist.

2. Verfahren zur Verschlüsselung von Daten in einer Datenenklave, wobei sich die Daten auf physikalischen Einheiten, bestehend aus nicht entfernbaren (2) und entfernbaren (4) Medien innerhalb der Datenenklave (20), befinden, wobei die Datenenklave (20) eine Sicherheitsdiensteinrichtung (Server) (24) umfaßt, die mittels eines Netzwerkes (12) mit einem oder mehreren Arbeitsplatzrechnern (10) verbunden ist, wobei jeder Arbeitsplatzrechner (10) eine Verschlüsselungs-Mediensteuereinheit (26) umfaßt, die eingesetzt ist, um eine der physikalischen Medieneinheiten (2, 4) zu lesen, wobei das Verfahren die folgenden Schritte umfaßt:

Vorsehen eines Enklavenschlüssels (40), der zum Verschlüsseln von Daten eingesetzt ist, die innerhalb der Datenenklave (20) übertragen werden;
Speichern einer Kopie des Enklavenschlüssels (40) in der Sicherheitsdiensteinrichtung (24) und in den Arbeitsplatzrechnern (10);
Vorsehen eines persönlichen Schlüssels (30) für jeden Benutzer innerhalb der Datenenklave (20);
Zuordnen einer persönlichen Identifikationsnummer (PIN) (50) und einer für jeden Benutzer einmaligen Identifizierung (Benutzer-UID) (48), an jeden Benutzer innerhalb der Datenenklave (20);
Zuordnen einer Reihe von Benutzer-Attributen (56) zu jedem Benutzer, wobei jede Reihe von Benutzer-Attributen (56) Benutzer-Privilegien und andere sicherheitsrelevante Informationen darstellt, die einem einzelnen Benutzer zugeordnet sind;
Zuordnen jeder Reihe von Benutzer-Attributen (56) zu der Benutzer-UID (48) des jeweiligen Benutzers;
Verschlüsseln jeder Benutzer-UID (48) mittels des Enklavenschlüssels (40) und Speichern jeder verschlüsselten Benutzer-UID (48') in dem persönlichen Schlüssel (30) des Benutzers, der zu dieser Benutzer-UID (48) zugehörig ist;
Zuordnen eines Medienschlüssels (42) und einer für jedes Medium einmaligen Identifizierung (Medien-UID) (46) zu jeder physikalische Medieneinheit (2, 4), wobei der Medienschlüssel (42) eingesetzt ist, um die Daten zu verschlüsseln und zu schützen, die sich auf den Medien befinden;
Zuordnen einer Reihe von Medien-Attributen (54) zu jeder physikalischen Medieneinheit (2, 4), wobei jede Reihe von Medien-Attributen (54) die Sensitivität oder andere sicherheitsrelevante Informationen darstellt, die Daten zuzuordnen sind, die sich auf einer bestimmten Medieneinheit (2, 4) befinden; und
Zuordnen jeder Reihe von Medien-Attributen (54) zu der Medien-UID (46) der jeweils zugehörigen physikalischen Medieneinheit (2, 4), **gekennzeichnet durch**
das Berechnen eines Zugangsvektors (52), der die Beschränkungen eines Zugriffes des einzelnen, bestimmten Benutzers (5) auf eine bestimmte Medieneinheit (2, 4) definiert, wobei die Berechnung mittels der Reihe von Benutzer-Attributen, die einem bestimmten Benutzer (5) zugeordnet sind sowie der Reihe von Medien-Attributen erfolgt, die einer bestimmten Medieneinheit (2, 4) zugeordnet sind;
das Kombinieren des Zugangsvektors (52) und des Medienschlüssels (42), der einer bestimmten Medieneinheit (2, 4) zugeordnet ist, um ein Medienschlüssel/Zugangsvektor-Paar (91) zu erhalten;
das Verschlüsseln des Medienschlüssel/Zugangsvektor-Paares (91) mit einem kombinierten Schlüssel, der aus dem Enklavenschlüssel (40), der Benutzer-UID (48) und der PIN (50) des bestimmten Benutzers (5) gebildet ist; und
das Speichern des verschlüsselten Medienschlüssel/Zugangsvektor-Paares (91) in dem persönlichen Schlüssel (30) des bestimmten Benutzers (5); und
das Steuern des Zugriffes auf Daten, die sich auf der bestimmten Medieneinheit (2, 4) befinden, wobei diese Steuerung als Funktion der PIN (50) des bestimmten Benutzers (5), der Medien-UID (46) der bestimmten physikalischen Medieneinheit (2, 4) und des Medienschlüssel/Zugangsvektor-Paares (91), der aus dem persönlichen Schlüssel (30) des bestimmten Benutzers (5) erhalten wird, ausgeführt ist.

3. Verfahren nach Anspruch 2, wobei das Verfahren darüber hinaus die folgenden Schritte aufweist:

Vorsehen von Geräte-Attributen für jeden Arbeitsplatzrechner (10), wobei die Geräte-Attribute Sicherheitsattribute der Arbeitsplatzrechner (10) darstellen, und wobei der Schritt des Steuerns des Zugriffes die folgenden Schritte umfaßt:
Feststellen des Arbeitsplatzrechners (10), der von dem bestimmten Benutzer (5) benutzt wird;
Laden der Geräte-Attribute (58), die dem Ar-

EX1010, Page 1239

beitsplatzrechner (10) zugeordnet sind, der von dem bestimmten Benutzer (5) benutzt wird;
Extrahieren des Zugangsvektors (52) aus dem verschlüsselten Medienschlüssel/Zugangsvektor-Paar (91), das aus dem persönlichen Schlüssel (30) des bestimmten Benutzers (5) erhalten wird; und
Kombinieren der geladenen Geräte-Attribute (58) mit dem extrahierten Zugangsvektor (52), um die Zugriffsrechte des bestimmten Benutzers (5) an einem bestimmten Arbeitsplatzrechner (10) festzulegen.

4. Verfahren nach Anspruch 2, wobei das Verfahren weiterhin die folgenden Schritte umfaßt:

(a) Vorsehen einer Schlüsselverarbeitungs-Verschlüsselungslogik in jeder Verschlüsselungs-Mediensteuereinheit zum (i) Erhalten einer anfordernden Benutzer-PIN aus dem persönlichen Schlüssel, (ii) zum Erhalten einer verschlüsselten Benutzer-UID aus dem persönlichen Schlüssel und zum Entschlüsseln der Benutzer-UID unter Einsatz des Enklavenschlüssels, und (iii) zum Erzeugen eines ersten Pakets, welches die anfordernde Benutzer-PIN, die Benutzer-UID und eine Anweisung zur Initialisierung einer neuen Medieneinheit umfaßt, wobei die Anweisung die Medien-Attribute für die neue Medieneinheit aufweist;
(b) Vorsehen der Schlüsselverarbeitungs-Verschlüsselungslogik in dem Server, zur Entschlüsselung des ersten Pakets unter Einsatz des Enklavenschlüssels, der in dem Server gespeichert ist;
(c) Vorsehen einer Speicherdurchsuchlogik in dem Server, zum (i) Lesen einer Benutzer-Attribut-Datenbank, die in dem Server abgespeichert ist, unter Verwendung der Benutzer-UID als Index, (ii) zum Zurückgeben eines Weitergabewertes, falls die anfordernde Benutzer-PIN, die mit dem ersten Paket erhalten wird, mit einer gültigen PIN übereinstimmt, die in der Benutzer-Attribut-Datenbank gespeichert ist, (iii) zum Abbrechen der Anforderung zur Initialisierung, falls die anfordernde Benutzer-PIN nicht gültig ist, (iv) zum Extrahieren der Medien-Attribute aus der Anforderung und zum Anweisen einer Medien-Attribut-Datenbank, die in dem Server abgespeichert ist, einen neuen Eintrag für die neue Medieneinheit vorzunehmen, sowie zum Erzeugen einer neuen Medien-UID für die neue Medieneinheit, und (v) zum Indizieren der Benutzer-Attribut-Datenbank mittels der Benutzer-UID, um die Reihe von Sicherheits-Attributen zu extrahieren, die dem anfordernden Benutzer zuzuordnen sind sowie zur Weitergabe der Sicherheits-Attribute zu der Sicherheitsleitlogik in dem Server;
(d) wobei die Sicherheitsleitlogik, die die Medien-Attribute und die Sicherheits-Attribute des anfordernden Benutzers empfängt sowie einen Satz von Regeln verwendet und/oder unter der Leitung eines Systemverwalters arbeitet, einen neuen Zugangsvektor errechnet, der die Beschränkungen beim Zugriff des anfordernden Benutzers festlegt, die dieser auf die neue Medieneinheit hat;
(e) wobei die Schlüsselverarbeitungs-Verschlüsselungslogik in dem Server ebenso (i) einen neuen Medienschlüssel, eventuell mit Hilfe eines Systemverwalters, für die neue Medieneinheit erzeugt, und (ii) das neue Medienschlüssel/Zugangsvektor-Paar verschlüsselt, welches mit dem neuen Medienschlüssel und dem neuen Zugangsvektor mittels eines kombinierten Schlüssels erzeugt ist, der die Benutzer-UID, die Benutzer-PIN und den Enklavenschlüssel umfaßt, um ein zweites Paket zu erzeugen;
(f) wobei die Speicherdurchsuchlogik ebenso das verschlüsselte zweite Paket in einer Verschlüsselungs-Schlüssel-Datenbank speichert, die in dem Serve abgespeichert ist, wobei das zweite Paket in Übereinstimmung mit der Benutzer-UID des anfordernden Benutzers und der neuen Medien-UID indiziert ist;
(g) wobei weiterhin eine Logik vorgesehen ist, um die neue Medien-UID und das zweite Paket zu dem Arbeitsplatzrechner zu schicken, von dem das erste Paket erhalten worden ist; und
(h) wobei eine Speicherdurchsuchlogik in der Verschlüsselungs-Mediensteuereinheit vorgesehen ist, um (i) die neue Medien-UID zu empfangen und um diese an einer geeigneten Position auf die neue Medieneinheit zu schreiben, sowie (ii) zum Abspeichern des zweiten Pakets, welches das neue Medienschlüssel/Zugangsvektor-Paar enthält, in dem persönlichen Schlüssel, der an dem Arbeitsplatzrechner angebracht ist, indem die neue Medien-UID als ein Index verwendet wird.

5. Verfahren nach Anspruch 2, weiterhin mit den Schritten:

(a) Vorsehen einer Schlüsselverarbeitungs-Verschlüsselungslogik in jeder Verschlüsselungs-Mediensteuereinheit zum (i) Erhalten einer anfordernden Benutzer-PIN aus dem persönlichen Schlüssel, (ii) zum Erhalten einer verschlüsselten Benutzer-UID aus dem persönlichen Schlüssel und zum Entschlüsseln der Benutzer-UID unter Einsatz des Enklavenschlüssels, und (iii) zum Lesen der Medien-UID aus einer initialisierten Medieneinheit und zum

EX1010, Page 1240

Durchsuchen des persönlichen Schlüssels nach einem Medienschlüssel/Zugangsvektor-Paar für die initialisierte Medieneinheit des anfordernden Benutzers, indem die Benutzer-PIN als ein Index eingesetzt wird, und (iv), falls kein Paar gefunden wird, zur Erzeugung einer Anforderung für eine Zuordnung eines Schlüssels;

(b) wobei die Schlüsselverarbeitungs-Verschlüsselungslogik in den Arbeitsplatzrechnern weiterhin (i) ein erstes Paket erzeugt, welches die anfordernde Benutzer-PIN und die Benutzer-UID, die Medien-UID zur Initialisierung der Medien-einheit, und eine Anweisung zur Zuordnung eines Schlüssels umfaßt, sowie (ii) zur Verschlüsselung des ersten Pakets mittels des Enklavenschlüssels eingesetzt ist, und (iii) zum Versenden des Pakets zu dem Sicherheits-Server, über das Netzwerk, dient;

(c) Vorsehen der Schlüsselverarbeitungs-Verschlüsselungslogik in dem Server, zur Entschlüsselung des ersten Pakets unter Einsatz des Enklavenschlüssels, der in dem Server gespeichert ist, um die anfordernde Benutzer-PIN und Benutzer-UID, die Medien-UID und die Anforderung zu erhalten;

(d) Vorsehen einer Speicherdurchsuchlogik in dem Sicherheits-Server, zum (i) Lesen einer Benutzer-Attribut-Datenbank, die in dem Server abgespeichert ist, unter Verwendung der Benutzer-UID als Index, (ii) zum Zurückgeben eines Weitergabewertes, falls die angeforderte Benutzer-PIN, die mit dem ersten Paket erhalten wird, mit einer gültigen PIN übereinstimmt, die in der Benutzer-Attribut-Datenbank gespeichert ist, (iii) zum Abbrechen der Anforderung zur Initialisierung, die in dem ersten Paket enthalten ist, falls die anfordernde Benutzer-PIN nicht gültig ist, (iv) Auslesen der Benutzer-Attribute-Datenbank, unter Verwendung der Benutzer-PIN als Index und Extrahieren der Sicherheits-Attribute des anfordernden Benutzers, und (v) Weitergabe der Sicherheits-Attribute zu der Sicherheitsleitlogik in dem Server;

(e) wobei die Sicherheitsleitlogik, die die Sicherheits-Attribute empfängt und die einen neuen Zugangsvektor errechnet, der die Beschränkungen beim Zugriff des anfordernden Benutzers festlegt, die dieser auf die neue Medieneinheit hat, den neuen Zugangsvektor errechnet, indem eine Reihe von Regeln verwendet wird und/oder unter Eingriff eines Systemverwalters erfolgt;

(f) wobei die Speicherdurchsuchlogik ebenso (i) ein verschlüsseltes Schlüsselpaket in einer Verschlüsselungs-Schlüssel-Datenbank findet, die in dem Sicherheits-Server vorhanden ist, und die vorher abgespeichert worden ist und die den Medienschlüssel für die initialisierte Medieneinheit enthält, und (ii), falls ein Paket gefunden ist, daraus den Medienschlüssel extrahiert, und (iii) ein neues Medienschlüssel/Zugangsvektor-Paar erzeugt, nämlich mittels des extrahierten Medienschlüssels und des neuen Zugangsvektors, sowie ein neues Schlüsselpaket erzeugt, welches das neue Medienschlüssel/Zugangsvektor-Paar, die Benutzer-UID und die Medien-UID umfaßt sowie sie das neue Schlüsselpaket in der Verschlüsselungs-Schlüssel-Datenbank zu Zwecken einer Archivierung ablegt;

(g) wobei die Verschlüsselungslogik ebenso das neue Medienschlüssel/Zugangsvektor-Paar mit einem kombinierten Schlüssel verschlüsselt, der die Benutzer-UID, die Benutzer-PIN und den Enklavenschlüssel umfaßt, sowie sie das verschlüsselte Paket über das Netzwerk zu der Verschlüsselungs-Mediensteuereinheit versendet; und

(h) wobei die Verschlüsselungs-Mediensteuereinheit, die die Medien-UID als Index verwendet, um das neue Medienschlüssel/Zugangsvektor-Paar in dem persönlichen Schlüssel zu speichern, von dem die Benutzer-PIN eingegeben worden ist, wobei der persönliche Schlüssel einen Medienschlüssel enthält, der nur von einer Person eingesetzt werden kann, die in tatsächlichem Besitz des persönlichen Schlüssels ist, die Benutzer-PIN kennt, die dem Medienschlüssel zugeordnet ist, und wobei die Person die Medieneinheit tatsächlich besitzt, die durch eine Verschlüsselungs-Mediensteuereinheit gesteuert ist, die den Enklavenschlüssel enthält, wobei der Zugriff des Benutzers weiter eingeschränkt wird, nämlich durch eine Kombination des Zugangsvektors und des Medienschlüssels.

6. Verfahren nach Anspruch 2, weiterhin mit den Schritten:

(a) Empfangen einer Benutzer-PIN durch die Verschlüsselungs-Mediensteuereinheit, von einem persönlichen Schlüssel eines Benutzers, der den Zugang zu einem initialisierten Medium sucht, und zwar unter der Kontrolle der Verschlüsselungs-Mediensteuereinheit;

(b) Vorsehen einer Speicherdurchsuchlogik in der Verschlüsselungs-Mediensteuereinheit, zum (i) Lesen des initialisierten Mediums und zum Extrahieren der Medien-UID, (ii) zum Durchsuchen des Speichers in dem persönlichen Schlüssel und zum Extrahieren des verschlüsselten Medienschlüssel/Zugangsvektor-Paares der Medien-UID und zum Weitergeben an eine Schlüsselverarbeitungs-Verschlüsse-

EX1010, Page 1241

lungseinrichtung in der Verschlüsselungs-Mediensteuereinheit;
(c) wobei die Schlüsselverarbeitungs-Verschlüsselungseinrichtung (i) die Benutzer-UID dem persönlichen Schlüssel entnimmt und diesen entschlüsselt, indem sie den Enklavenschlüssel verwendet, (ii) die Benutzer-UID, die Benutzer-PIN und den Enklavenschlüssel kombiniert, um einen zusammengesetzten Schlüssel zu bilden, um das Medienschlüssel/Zugangsvektor-Paar zu entschlüsseln, sowie um den extrahierten Medienschlüssel an die Datenverschlüsselungseinrichtung zurück zu geben und den Zugangsvektor an die Zugriffssteuerlogik weiterzugeben;
(d) wobei die Datenverschlüsselungseinrichtung die Daten auf einer Medieneinheit entschlüsselt, indem sie den Medienschlüssel einsetzt und an die Zugriffssteuerlogik weitergibt, wobei die Daten entschlüsselt werden, und zwar in Antwort auf eine Lese- oder Schreibanweisung der Daten durch den Arbeitsplatzrechner;
(e) wobei die Zugriffssteuerlogik kontrolliert, ob der erwünschte Modus des Zugriffes erlaubt ist, basierend auf dem Zugangsvektor und den Geräte-Attributen, die in der Verschlüsselungs-Mediensteuereinheit enthalten sind, und wobei die Logik den Versuch des Zugriffs auf die Daten abbricht, falls der Zugriff nicht autorisiert ist und andernfalls den Zugriff ermöglicht, wodurch die Daten auf den Arbeitsplatzrechner übertragen werden, um dort weiter verarbeitet zu werden; und
(f) wobei eine Logik in der Verschlüsselungs-Mediensteuereinheit vorgesehen ist, um ein vollständiges Zurücksetzen der Verschlüsselungs-Mediensteuereinheit zu bewirken, wodurch es erforderlich ist, das die Schlüsselverarbeitung von vorne beginnt, falls der persönliche Schlüssel entfernt wird oder falls die Medieneinheit aus dem Arbeitsplatzrechner entfernt wird.

Revendications

1. Enclave de données (20) pour sécuriser des données portées sur des unités physiques de supports fixes (2) et amovibles (4), l'enclave de données (20) comprenant un serveur de sécurité (24) connecté, via un réseau (12), à un ou plusieurs postes de travail (10), enclave dans laquelle chaque poste de travail (10) comprend un contrôleur de support cryptographique (26) utilisé pour lire l'une desdites unités physiques de supports (2, 4), l'enclave de données comprenant en outre :

une clé d'enclave (40) utilisée pour coder des données transmises à l'intérieur de l'enclave de données (20), une copie de la clé d'enclave (40) étant stockée dans le serveur de sécurité (24) et dans les postes de travail (10),
un dispositif de saisie personnel (30) pour chaque utilisateur de l'enclave de données (20),
un numéro d'identification personnel (PIN) (50) et un identificateur unique d'utilisateur (UID d'utilisateur) (48) affectés à chaque utilisateur de l'enclave (20), chaque UID d'utilisateur (48) étant codé avec la clé d'enclave et stocké dans le dispositif de saisie personnel (30) de l'utilisateur associé à l'UID d'utilisateur,
un ensemble d'attributs d'utilisateur (56) prévu pour chaque utilisateur, chaque ensemble d'attributs d'utilisateur (56) représentant des privilèges d'utilisateur et d'autres informations concernant la sécurité à propos d'un utilisateur particulier et chaque ensemble d'attributs d'utilisateur (56) étant associé à l'UID d'utilisateur (48) de son utilisateur respectif,
une clé de support (42) pour chaque unité de support physique (2, 4), la clé de support (42) étant utilisée pour coder et protéger les données portées sur le support,
un identificateur unique de support (UID de support) (46) pour chaque unité physique de support (2, 4), et
un ensemble d'attributs de support (54) prévu pour chaque unité physique de support (2, 4), chaque ensemble d'attributs de support (54) représentant une sensibilité ou d'autres informations concernant la sécurité à propos des données portées sur une unité particulière de support et chaque ensemble d'attributs de support (54) étant associé à l'UID de support (46) de son unité physique respective de support (2, 4), caractérisée en ce que :

le serveur de sécurité (24) comprend :

une logique de police de sécurité (86) pour calculer à partir de l'ensemble d'attributs d'utilisateur affecté à un utilisateur particulier (5) et de l'ensemble d'attributs de support affectés à une unité de support particulière (2, 4), un vecteur d'accès (52) qui définit des limites à l'accès par l'utilisateur particulier (5) à l'unité particulière de support (2, 4), et
un système cryptographique de gestion de clés (70) pour combiner le vecteur d'accès (52) et la clé de support (42) affectée à l'unité particulière de support (2, 4) pour former une paire de clés de support/vecteur d'accès (91) et pour chiffrer la paire de clés de support/vecteur d'accès (91) avec une clé combinée formée de la clé d'encla-

EX1010, Page 1242

ve (40) et de l'UID d'utilisateur (48) et du PIN (50) de l'utilisateur particulier (5),
le dispositif de saisie personnel (30) comprenant des moyens (78) pour stocker la paire chiffrée de clés de support/vecteur d'accès (91), et
le contrôleur de support cryptographique (26) comprenant des moyens (70, 72, 76) pour contrôler l'accès à des données sur l'unité particulière de support (2, 4) en fonction du PIN (50) de l'utilisateur particulier (5), de l'UID de support (46) de l'unité physique particulière de support (2, 4) et de la paire de clés de support/vecteur d'accès (91) retrouvée dans le dispositif de saisie personnel (30) de l'utilisateur particulier (5).

2. Procédé à enclave de données pour sécuriser des données portées sur des unités physiques de support fixes (2) et amovibles (4) dans une enclave de données (20) comprenant un serveur de sécurité (24) connecté, via un réseau (12), à un ou plusieurs postes de travail (10), chaque poste de travail (10) comprenant un contrôleur de support cryptographique (26) utilisé pour lire une desdites unités physiques de support (2, 4), ledit procédé comprenant les étapes suivantes :

on met en oeuvre une clé d'enclave (40) utilisée pour coder des données transmises à l'intérieur de l'enclave de données (20),
on stocke une copie de la clé d'enclave (40) dans le serveur de sécurité (24) et dans les postes de travail (10),
on utilise un dispositif de saisie personnel (30) pour chaque utilisateur dans l'enclave de données (20),
on affecte un numéro d'identification personnel (PIN) (50) et un identificateur unique d'utilisateur (UID d'utilisateur) (48) pour chaque utilisateur dans l'enclave (20),
on affecte un ensemble d'attributs d'utilisateur (56) pour chaque utilisateur, chaque ensemble d'attributs d'utilisateur (56) représentant des privilèges de l'utilisateur et d'autres informations concernant la sécurité à propos d'un utilisateur particulier,
on associe chaque ensemble d'attributs d'utilisateur (56) à l'UID d'utilisateur (48) de son utilisateur respectif,
on code chaque UID d'utilisateur (48) avec la clé d'enclave et on stocke chaque UID d'utilisateur codé (48') dans le dispositif de saisie personnel (30) de l'utilisateur associé à l'UID d'utilisateur (48),
on affecte une clé de support (42) et un identificateur unique de support (UID de support) (46) à chaque unité physique de support (2, 4), la clé de support (42) étant utilisée pour coder et protéger les données portées sur le support,
on affecte un ensemble d'attributs de support (54) pour chaque unité physique de support (2, 4), chaque ensemble d'attributs de support (54) représentant une sensibilité ou d'autres informations concernant la sécurité à propos des données portées sur une unité particulière de support, et
on associe chaque ensemble d'attributs de support (54) à l'UID de support (48) de son unité physique respective de support (2, 4), caractérisé en ce que :
on calcule, à partir de l'ensemble d'attributs d'utilisateur affecté à un utilisateur particulier (5) et de l'ensemble d'attributs de support affecté à une unité particulière de support (2, 4), un vecteur d'accès (52) qui définit des limites à l'accès par l'utilisateur particulier (5) à l'unité particulière de support (2, 4),
on combine le vecteur d'accès (52) et la clé de support (42) affectée à l'unité particulière de support (2, 4) pour former une paire de clés de support/vecteur d'accès (91),
on chiffre la paire de clés de support/vecteur d'accès (91) avec une clé combinée formée de la clé d'enclave (40) et de l'UID d'utilisateur (48) et du PIN (50) de l'utilisateur particulier (5), et
on stocke la paire chiffrée de clés de support/vecteur d'accès (91) dans le dispositif de saisie personnel (30) de l'utilisateur particulier (5), et
on contrôle l'accès aux données sur l'unité particulière de support (2, 4) en fonction du PIN (50) de l'utilisateur particulier (5), de l'UID de support (46) de l'unité physique particulière de support (2, 4) et de la paire de clés de support/vecteur d'accès (91) retrouvée dans le dispositif de saisie personnel (30) de l'utilisateur particulier (5).

3. Procédé selon la revendication 2, dans lequel le procédé comprend en outre la mise en oeuvre d'attributs de dispositif pour chaque poste de travail (10), les attributs de dispositif représentant des attributs de sécurité des postes de travail (10), et dans lequel l'étape de commande d'accès comprend les étapes suivantes :

on détermine le poste de travail (10) utilisé par l'utilisateur particulier (5),
on recherche les attributs de dispositif (58) associés au poste de travail (10) utilisé par l'utilisateur particulier (5),
on extrait le vecteur d'accès (52) de la paire codée de clés de support/vecteur d'accès (91) retrouvée dans le dispositif de saisie personnel (30) de l'utilisateur particulier (5), et

EX1010, Page 1243

on combine les attributs de dispositif retrouvés (58) avec le vecteur d'accès extrait (52) pour déterminer les droits d'accès par l'utilisateur particulier (5) au poste de travail particulier (10).

4. Procédé selon la revendication 2, dans lequel le procédé comprend en outre les étapes suivantes :

(a) on met en oeuvre une logique cryptographique de gestion de clés dans chaque contrôleur de support cryptographique (i) pour recevoir un PIN d'utilisateur demandeur d'un dispositif de saisie personnel, (ii) pour recevoir un UID d'utilisateur codé du dispositif de saisie personnel et décoder l'UID d'utilisateur en utilisant la clé d'enclave, et (iii) pour former un premier paquet comprenant le PIN de l'utilisateur demandeur, l'UID d'utilisateur et une demande d'initialisation d'une nouvelle unité de support, la demande comprenant les attributs de support pour la nouvelle unité de support,
(b) on met en oeuvre une logique cryptographique de gestion de clés dans le serveur pour décoder le premier paquet en utilisant la clé d'enclave stockée dans le serveur,
(c) on met en oeuvre une logique de recherche de stockage dans le serveur (i) pour lire une base de données d'attributs d'utilisateur dans le serveur en utilisant l'UID d'utilisateur à titre d'index, (ii) pour renvoyer une valeur de validation si le PIN de l'utilisateur demandeur reçu dans le premier paquet correspond à un PIN valide stocké dans la base de données d'attributs d'utilisateur, (iii) pour avorter la demande d'initialisation si le PIN de l'utilisateur demandeur n'est pas valide, (iv) pour extraire les attributs de support de la demande et commander à une base de données d'attributs de support stockée dans le serveur d'effectuer une entrée pour la nouvelle unité de support et de créer un nouvel UID de support pour la nouvelle unité de support, et (v) pour indexer la base de données d'attributs d'utilisateur avec l'UID d'utilisateur afin d'extraire la série d'attributs de sécurité appartenant à l'utilisateur demandeur et pour faire passer les attributs de sécurité à une logique de police de sécurité dans le serveur,
(d) la logique de police de sécurité acceptant les attributs de support et les attributs de sécurité de l'utilisateur demandeur et, en utilisant une série de règles et/ou sous la direction d'un administrateur du système, on calcule un nouveau vecteur d'accès qui définit des limites à l'accès que l'utilisateur demandeur aura sur la nouvelle unité de support,
(e) le circuit logique cryptographique de gestion de clés dans le serveur (i) générant également, avec l'aide facultative de l'administrateur du système, une nouvelle clé de support pour la nouvelle unité de support et (ii) chiffrant la nouvelle paire de clés de support/vecteur d'accès formée avec la nouvelle clé de support et le nouveau vecteur d'accès avec une clé combinée comprenant l'UID d'utilisateur, le PIN d'utilisateur et la clé d'enclave, pour former un deuxième paquet,
(f) la logique de recherche de stockage stockant également le deuxième paquet chiffré dans une base de données de clés cryptographiques stockée dans le serveur, le deuxième paquet indexé selon l'UID d'utilisateur de l'utilisateur demandeur et le nouveau UID de support,
(g) on met en oeuvre une autre logique pour envoyer le nouvel UID de support et le deuxième paquet au poste de travail d'où le premier paquet a été reçu, et
(h) on met en oeuvre une logique de recherche de stockage dans le contrôleur de support cryptographique (i) pour recevoir le nouvel UID de support et le transcrire dans un emplacement approprié sur la nouvelle unité de support et (ii) pour stocker le deuxième paquet comprenant la nouvelle paire de clés de support/vecteur d'accès dans le dispositif de saisie personnel lié au poste de travail en utilisant le nouvel UID de support comme index.

5. Procédé selon la revendication 2, comprenant en outre les étapes suivantes :

(a) on met en oeuvre une logique cryptographique de gestion de clés dans chaque contrôleur de support cryptographique (i) pour recevoir un PIN de l'utilisateur demandeur d'un dispositif de saisie personnel, (ii) pour recevoir un UID d'utilisateur codé du dispositif de saisie personnel et décoder l'UID d'utilisateur en utilisant la clé d'enclave et (iii) pour lire l'UID de support sur l'unité de support initialisée et rechercher sur le dispositif de saisie personnel une paire de clés de support/vecteur d'accès pour l'unité de support initialisée pour l'utilisateur demandeur en utilisant le PIN d'utilisateur comme index, et (iv) si aucune paire n'est trouvée, pour générer une demande d'affectation de clé,
(b) la logique cryptographique de gestion de clés dans les postes de travail formant en outre (i) le premier paquet comprenant le PIN de l'utilisateur demandeur et l'UID d'utilisateur, l'UID de support pour l'unité de support initialisée et la demande d'affectation de clé, (ii) codant le premier paquet avec la clé d'enclave et (iii) envoyant le paquet au serveur de sécurité via le réseau,

EX1010, Page 1244

(c) on met en oeuvre une logique cryptographique de gestion de clés dans le serveur pour décoder le premier paquet en utilisant la clé d'enclave stockée dans le serveur afin d'obtenir le PIN de l'utilisateur demandeur, et l'UID d'utilisateur, l'UID de support et la demande,

(d) on met en oeuvre une logique de recherche de stockage dans le serveur de sécurité (i) pour lire une base de données d'attributs d'utilisateur stockée dans le serveur en utilisant l'UID d'utilisateur comme index, (ii) pour renvoyer une valeur de validation si le PIN de l'utilisateur demandeur reçu dans le premier paquet correspond à un PIN valide stocké dans la base de données d'attributs d'utilisateur, (iii) pour avorter la demande d'initialisation établie dans le premier paquet si le PIN de l'utilisateur demandeur n'est pas valide, (iv) pour lire la base de données d'attributs d'utilisateur en utilisant le PIN d'utilisateur comme index et extraire les attributs de sécurité de l'utilisateur demandeur, et (v) pour faire passer les attributs de sécurité à la logique de police de sécurité dans le serveur,

(e) la logique de police de sécurité recevant les attributs de sécurité et calculant un nouveau vecteur d'accès qui définit des limites à l'accès que l'utilisateur peut avoir sur l'unité de support initialisée, le nouveau vecteur d'accès calculé utilisant une série de règles et/ou avec l'intervention d'un administrateur du système,

(f) la logique de recherche de stockage définissant également (i) un paquet de clé chiffré dans une base de données de clés cryptographiques conservée dans le serveur de sécurité, qui a été précédemment stocké et qui contient la clé de support pour l'unité de support initialisée, (ii) lorsqu'un paquet est trouvé, extrayant la clé de support de celui-ci, et (iii) formant une nouvelle paire de clés de support/vecteur d'accès avec la clé de support extraite et le nouveau vecteur d'accès, et un nouveau paquet de clé comprenant la nouvelle paire de clés de support/vecteur d'accès, l'UID d'utilisateur et l'UID de support, et en plaçant le nouveau paquet de clé dans la base de données de clés cryptographiques à des fins d'archivage,

(g) la logique de clés cryptographiques chiffrant également la nouvelle paire de clés de support/vecteur d'accès avec une clé combinée en utilisant l'UID d'utilisateur, le PIN d'utilisateur et la clé d'enclave, et en transmettant le paquet chiffré par le réseau au contrôleur de support cryptographique, et

(h) le contrôleur de support cryptographique utilisant l'UID de support comme index pour stocker la nouvelle paire de clés de support/vecteur d'accès dans le dispositif de saisie personnel à partir duquel le PIN d'utilisateur a été entré, de telle sorte que le dispositif de saisie personnel contienne une clé de support qui ne peut être utilisée que par quelqu'un qui est en possession physique de ce dispositif de saisie personnel, connaît le PIN d'utilisateur associé à la clé de support et est en possession physique de l'unité de support commandée par le contrôleur de support cryptographique contenant la clé d'enclave, l'accès de l'utilisateur étant en outre restreint par le vecteur d'accès jumelé à la clé de support.

6. Procédé selon la revendication 2, comprenant en outre les étapes suivantes :

(a) le contrôleur de support cryptographique (i) recevant également un PIN d'utilisateur d'un dispositif de saisie personnel d'un utilisateur cherchant un accès à une unité de support initialisée sous le contrôle du contrôleur de support cryptographique,

(b) on met en oeuvre le circuit logique de recherche de stockage dans le contrôleur de support cryptographique (i) pour lire l'unité de support initialisée et extraire l'UID de support, (ii) pour rechercher le stockage dans le dispositif de saisie personnel et extraire la paire chiffrée de clés de support/vecteur d'accès pour l'UID de support et l'envoyer à un système cryptographique de gestion de clés dans le contrôleur de support cryptographique,

(c) le système cryptographique de gestion de clés (i) recherchant l'UID d'utilisateur dans le dispositif de saisie personnel et le déchiffrant en utilisant la clé d'enclave, (ii) combinant l'UID d'utilisateur, le PIN d'utilisateur et la clé d'enclave pour former une clé combinée afin de décoder la paire de clés de support/vecteur d'accès, et faisant passer la clé de support extraite à un système cryptographique de données et le vecteur d'accès à la logique de commande d'accès,

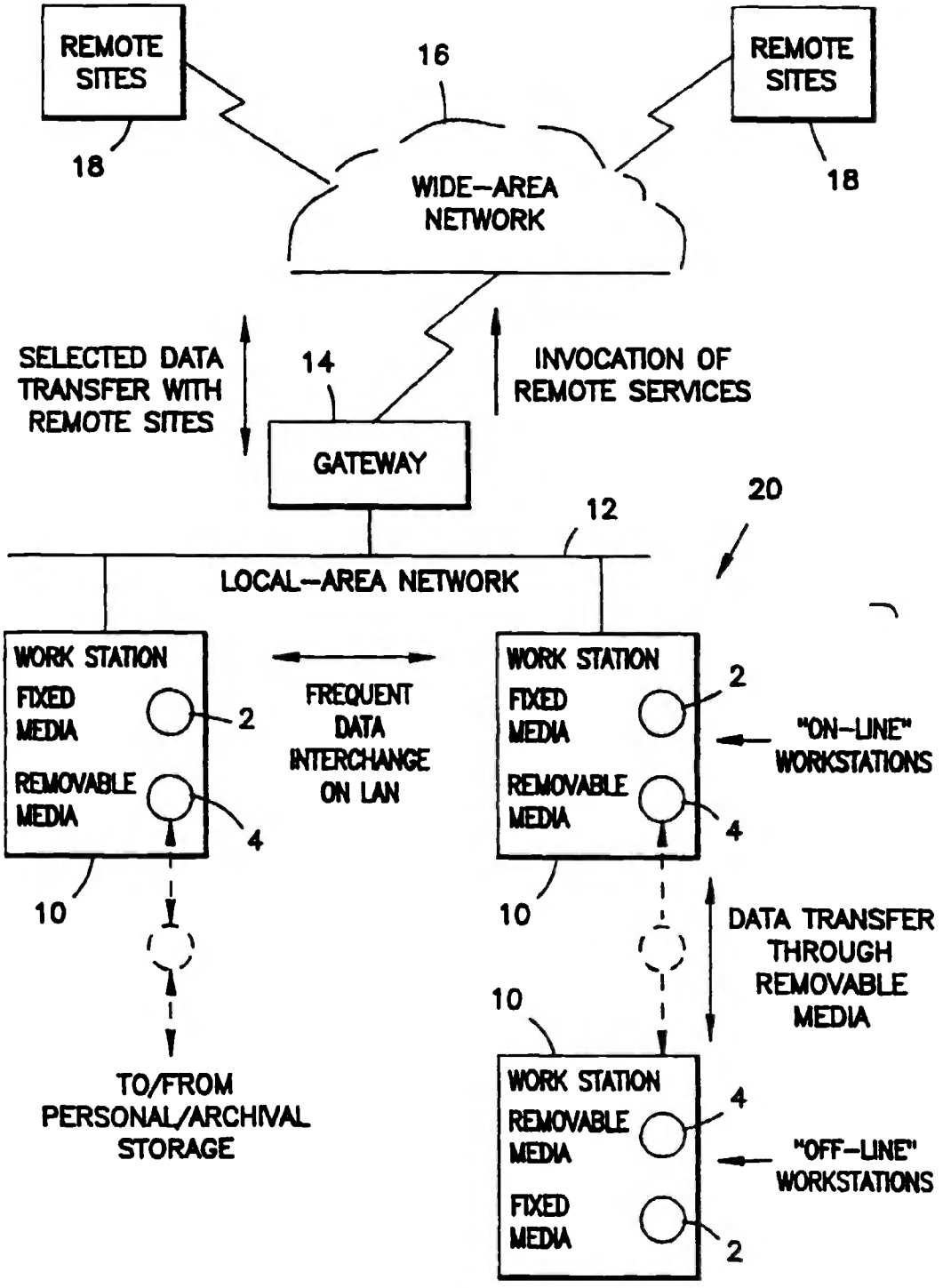
(d) le système cryptographique de données déchiffrant les données d'une unité de support en utilisant la clé de support et faisant passer dans la logique de commande d'accès les données déchiffrées en réponse à une demande de lecture ou d'écriture des données par le poste de travail,

(e) la logique de commande d'accès contrôlant si le mode souhaité d'accès est autorisé sur la base du vecteur d'accès et des attributs de dispositif contenus dans le contrôleur de support cryptographique et avortant la tentative d'accès aux données si l'accès n'est pas autorisé et permettant autrement l'accès de telle sorte que les données soient transférées à un poste de tra-

EX1010, Page 1245

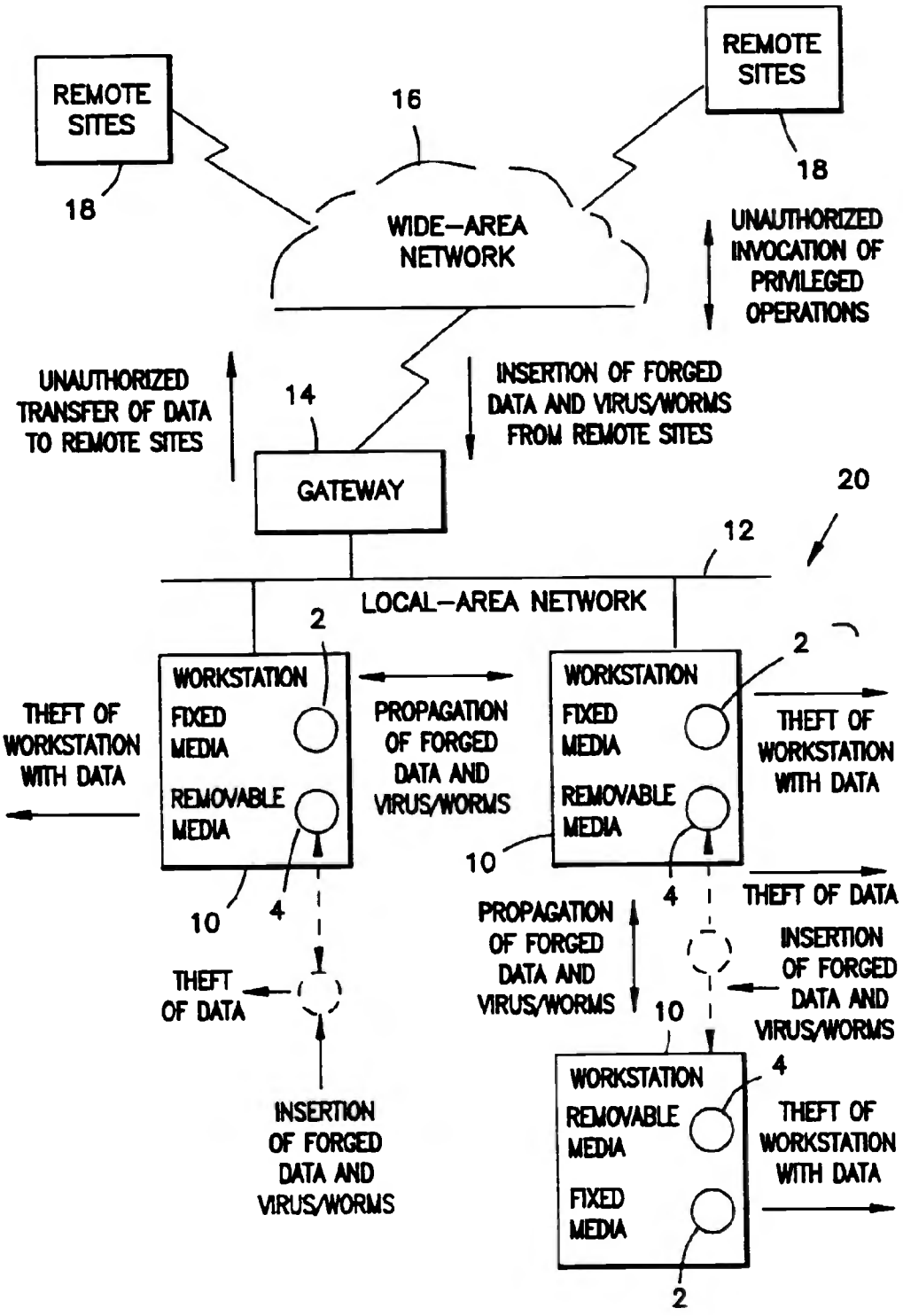
vail pour traitement, et
(f) on met en oeuvre une logique dans le contrôleur de support cryptographique pour provoquer une remise à zéro complète du contrôleur de support cryptographique et on recommence le processus de saisie depuis le début dans le cas où le dispositif de saisie personnel est désaccouplé ou que l'unité de support est retirée du poste de travail.

EX1010, Page 1246



PRIOR ART

FIG. 1



PRIOR ART

FIG. 2

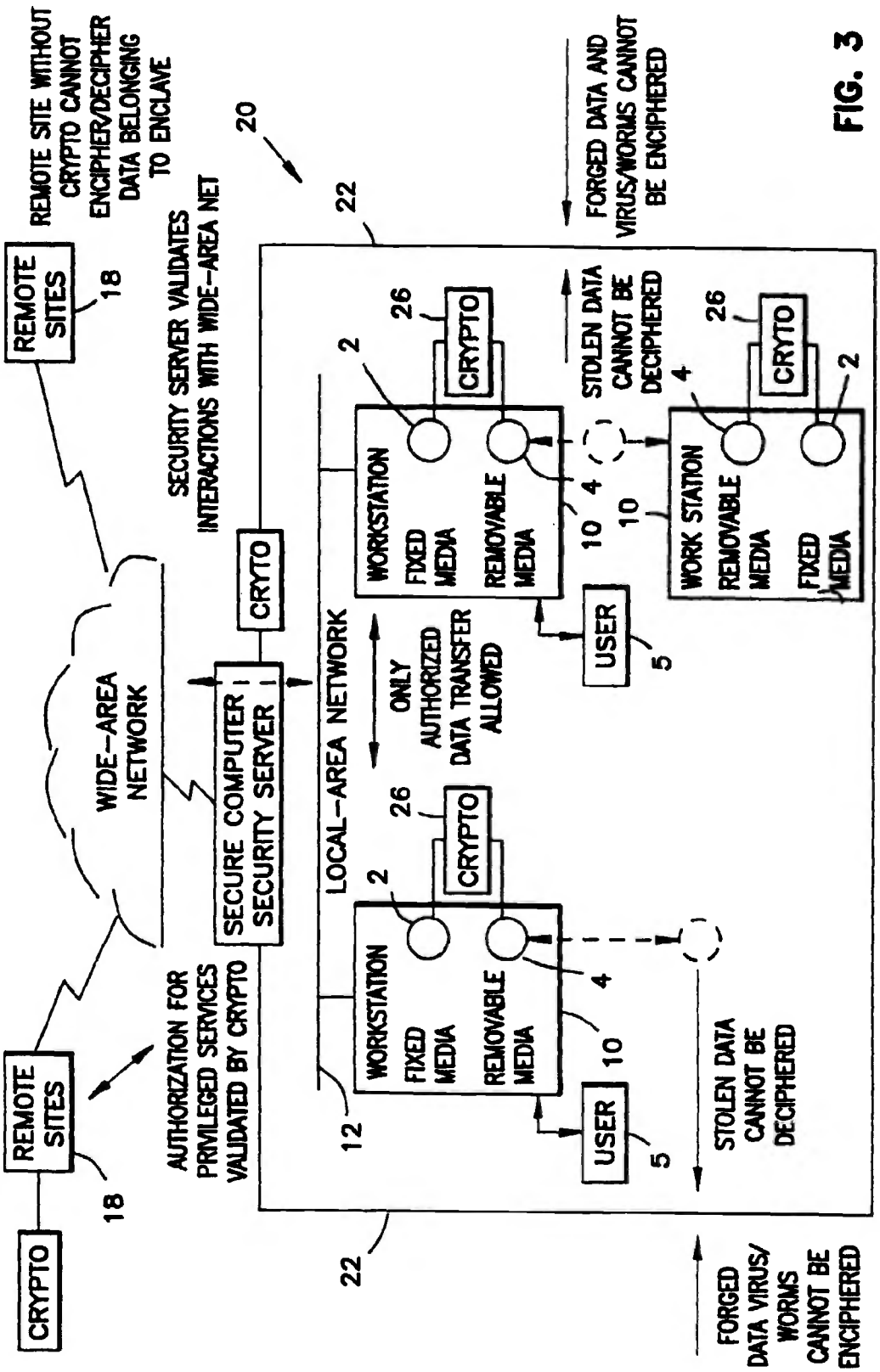

REMOTE SITES
18
REMOTE SITE WITHOUT CRYPTO CANNOT ENCIPHER/DECIPHER DATA BELONGING TO ENCLAVE
SECURITY SERVER VALIDATES INTERACTIONS WITH WIDE-AREA NET
20
22
FORGED DATA AND VIRUS/WORMS CANNOT BE ENCIPHERED
CRYPTO
CRYTO
WIDE-AREA NETWORK
SECURE COMPUTER SECURITY SERVER
LOCAL-AREA NETWORK
ONLY AUTHORIZED DATA TRANSFER ALLOWED
WORKSTATION
FIXED MEDIA
REMOVABLE MEDIA
WORK STATION
USER
STOLEN DATA CANNOT BE DECIPHERED
AUTHORIZATION FOR PRIVILEGED SERVICES VALIDATED BY CRYPTO
FORGED DATA VIRUS/ WORMS CANNOT BE ENCIPHERED
2
4
5
10
12
26

FIG. 3

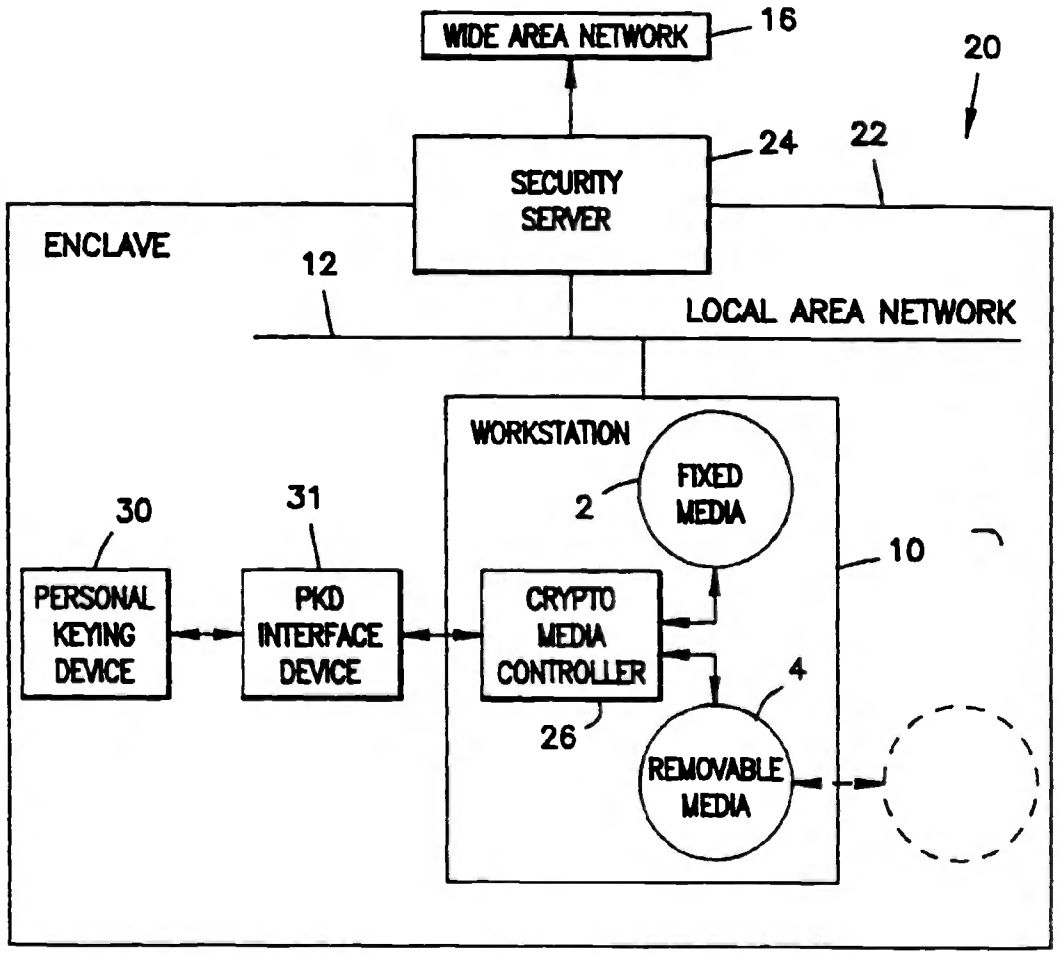

WIDE AREA NETWORK
16
20
24
22
SECURITY SERVER
ENCLAVE
12
LOCAL AREA NETWORK
WORKSTATION
FIXED MEDIA
2
30
31
10
PERSONAL KEYING DEVICE
PKD INTERFACE DEVICE
CRYPTO MEDIA CONTROLLER
4
26
REMOVABLE MEDIA

FIG. 4

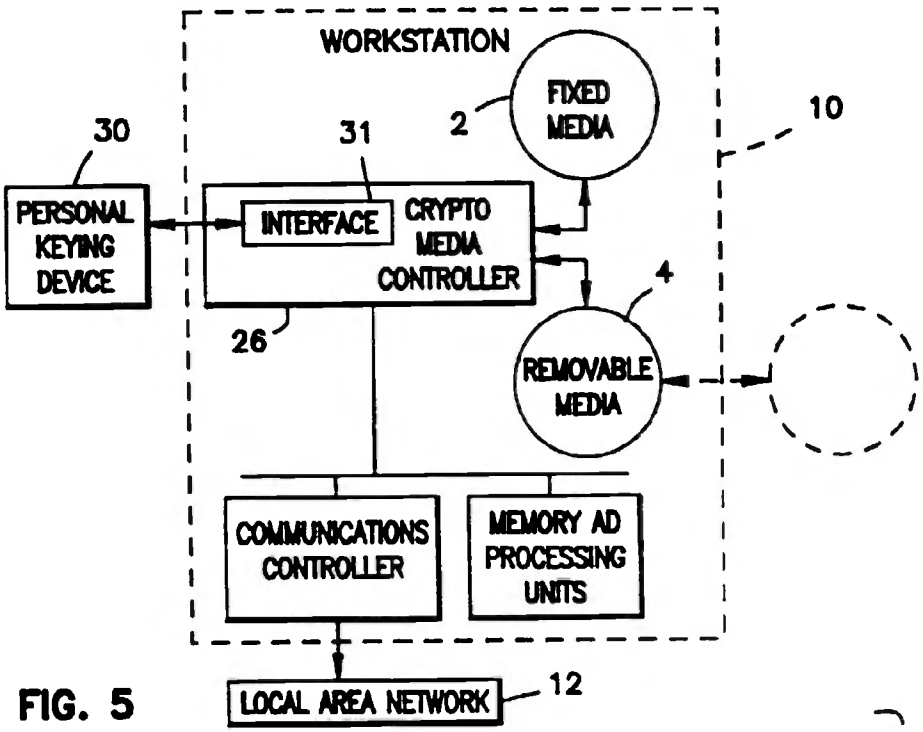

WORKSTATION
FIXED MEDIA
2
10
30
31
PERSONAL KEYING DEVICE
INTERFACE
CRYPTO MEDIA CONTROLLER
4
26
REMOVABLE MEDIA
COMMUNICATIONS CONTROLLER
MEMORY AD PROCESSING UNITS
LOCAL AREA NETWORK
12

FIG. 5

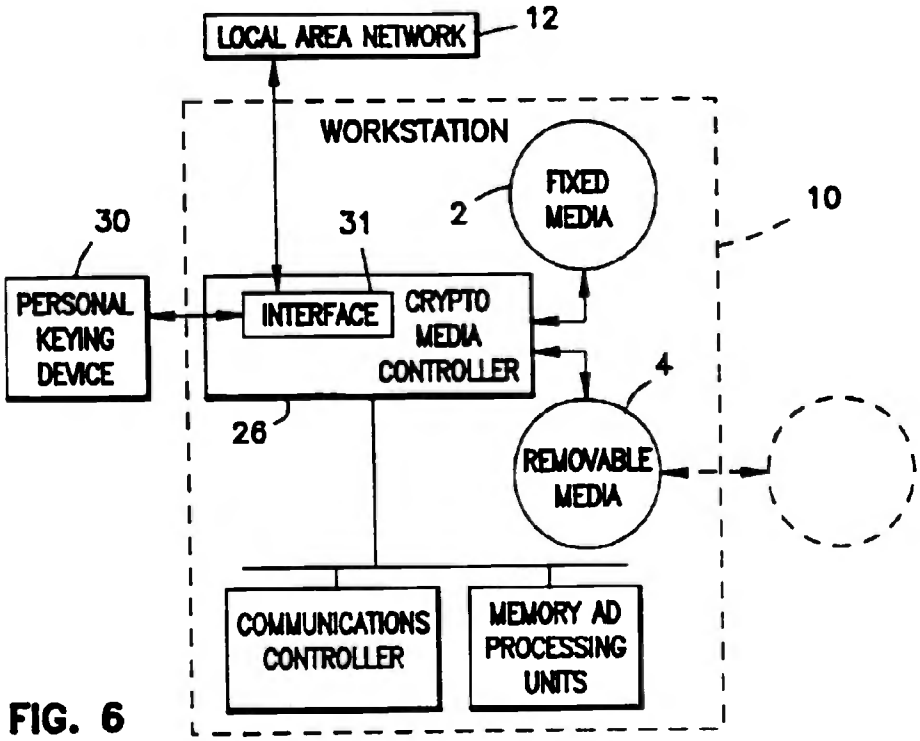

LOCAL AREA NETWORK
12
WORKSTATION
FIXED MEDIA
2
10
30
31
PERSONAL KEYING DEVICE
INTERFACE
CRYPTO MEDIA CONTROLLER
4
26
REMOVABLE MEDIA
COMMUNICATIONS CONTROLLER
MEMORY AD PROCESSING UNITS

FIG. 6

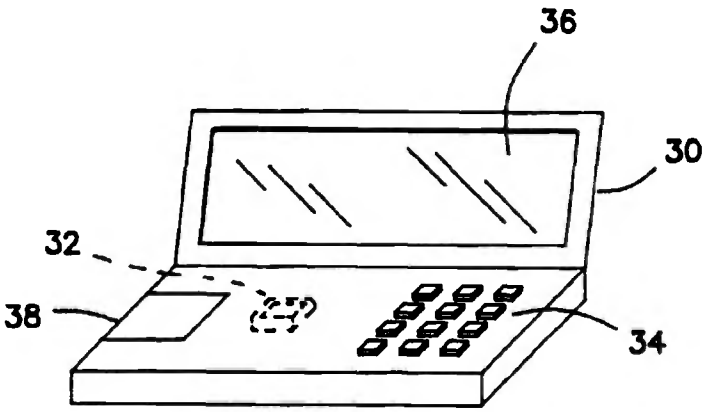


FIG. 6A

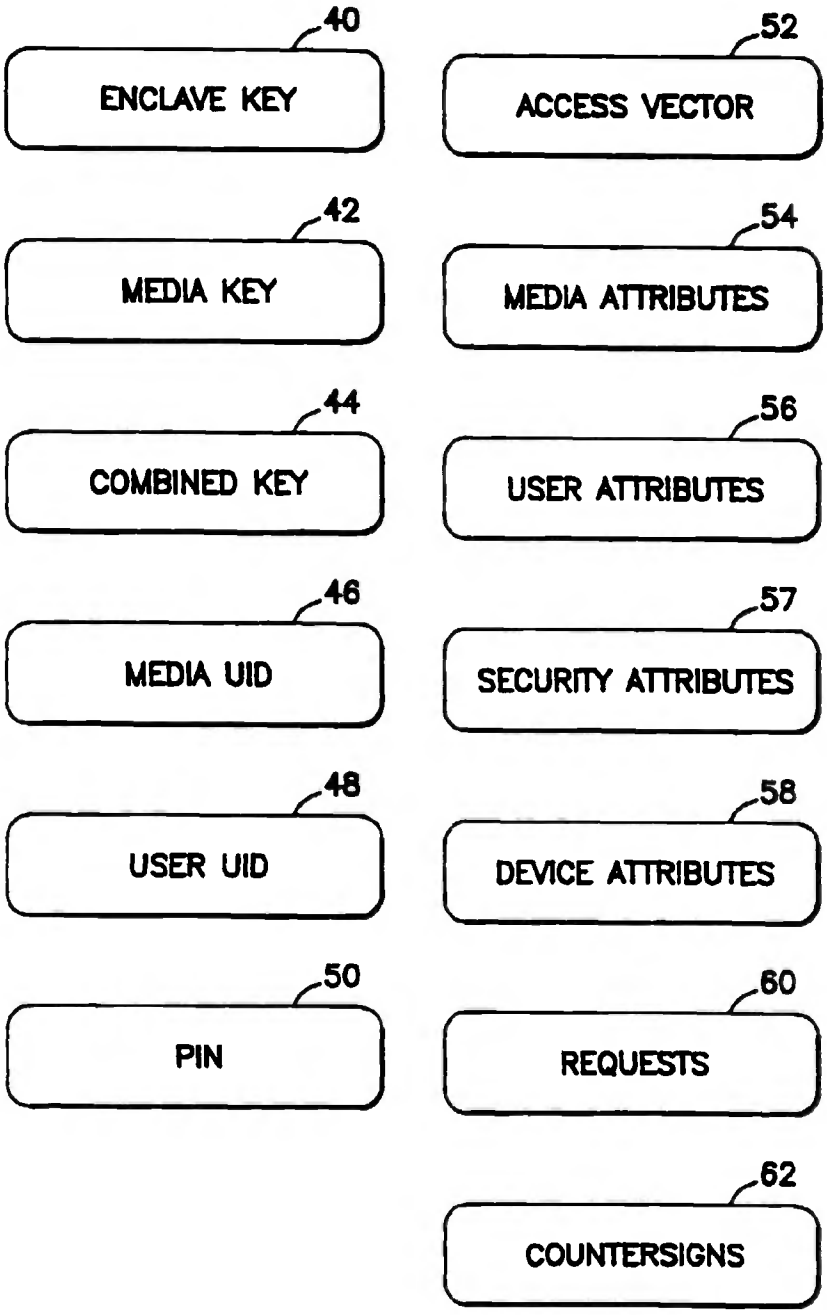

40
ENCLAVE KEY
42
MEDIA KEY
44
COMBINED KEY
46
MEDIA UID
48
USER UID
50
PIN
52
ACCESS VECTOR
54
MEDIA ATTRIBUTES
56
USER ATTRIBUTES
57
SECURITY ATTRIBUTES
58
DEVICE ATTRIBUTES
60
REQUESTS
62
COUNTERSIGNS

FIG. 6B

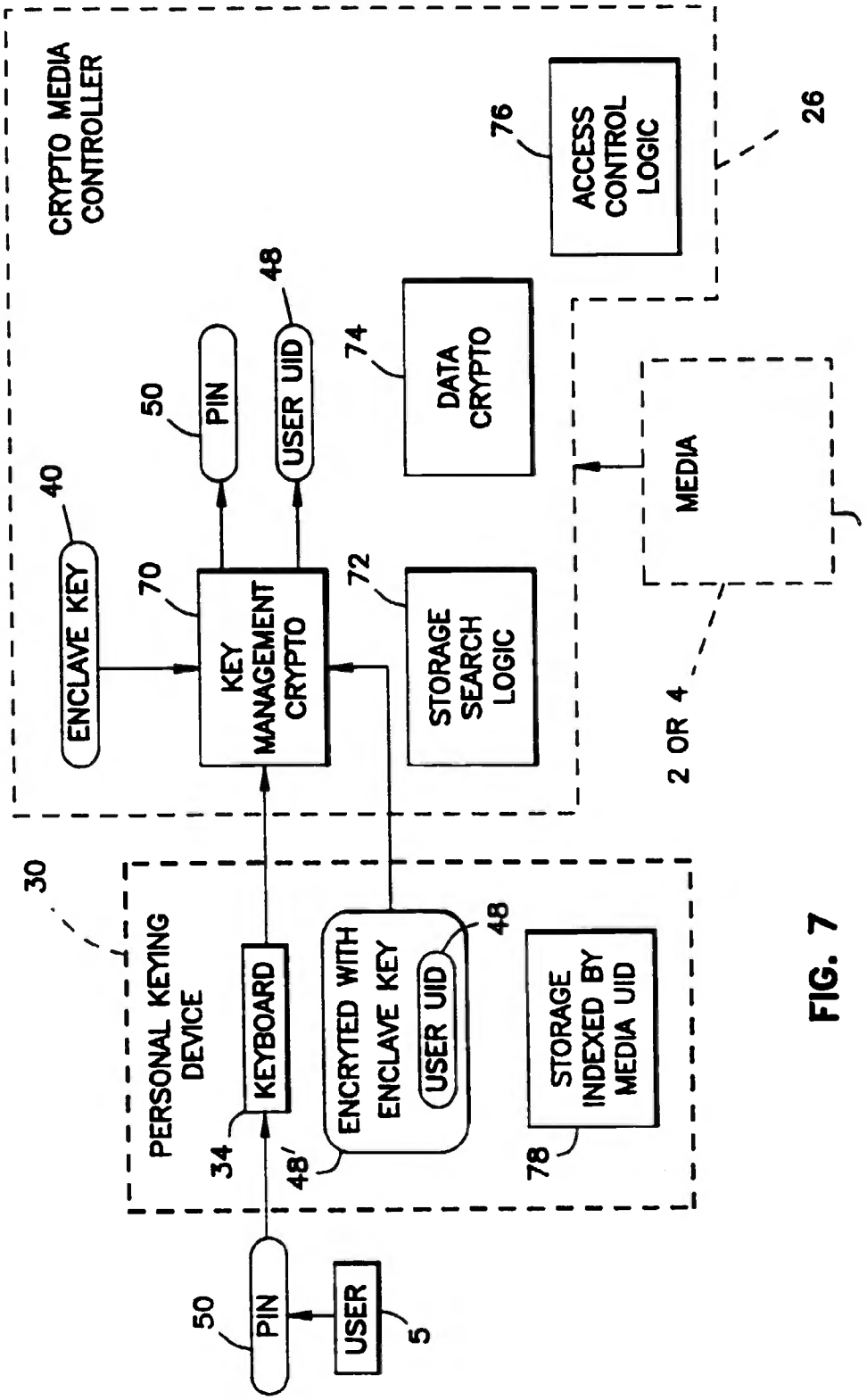

CRYPTO MEDIA CONTROLLER
26
76
ACCESS CONTROL LOGIC
48
50
PIN
USER UID
74
DATA CRYPTO
40
ENCLAVE KEY
70
KEY MANAGEMENT CRYPTO
72
STORAGE SEARCH LOGIC
MEDIA
2 OR 4
30
PERSONAL KEYING DEVICE
KEYBOARD
34
ENCRYTED WITH ENCLAVE KEY
USER UID
48
48'
STORAGE INDEXED BY MEDIA UID
78
50
PIN
USER
5

FIG. 7

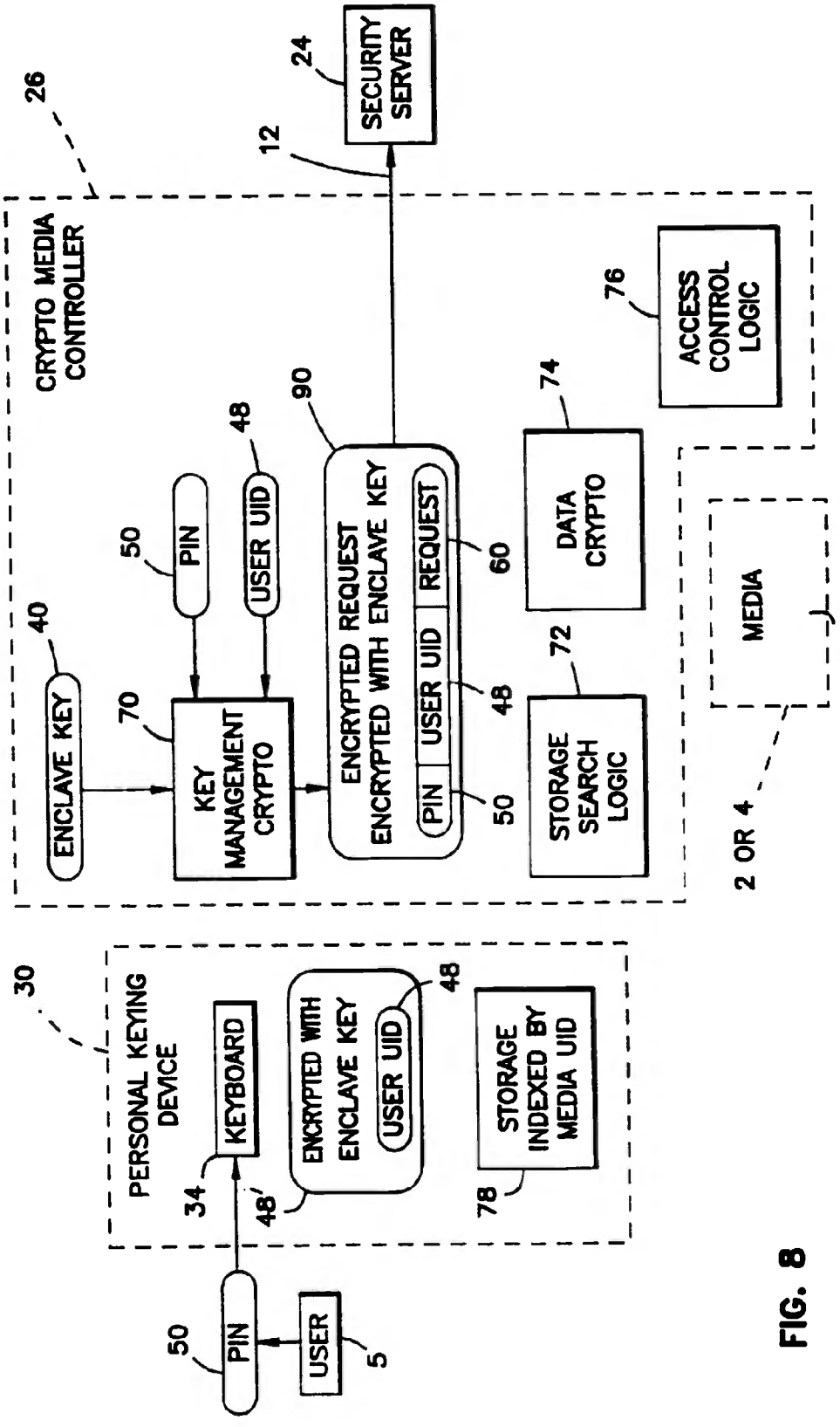

SECURITY SERVER
24
12
26
CRYPTO MEDIA CONTROLLER
ACCESS CONTROL LOGIC
76
DATA CRYPTO
74
MEDIA
2 OR 4
ENCRYPTED REQUEST ENCRYPTED WITH ENCLAVE KEY
90
PIN
USER UID
REQUEST
60
48
50
PIN
USER UID
48
50
ENCLAVE KEY
40
KEY MANAGEMENT CRYPTO
70
STORAGE SEARCH LOGIC
72
30
PERSONAL KEYING DEVICE
KEYBOARD
34
ENCRYPTED WITH ENCLAVE KEY
USER UID
48
48
STORAGE INDEXED BY MEDIA UID
78
50
PIN
USER
5

FIG. 8

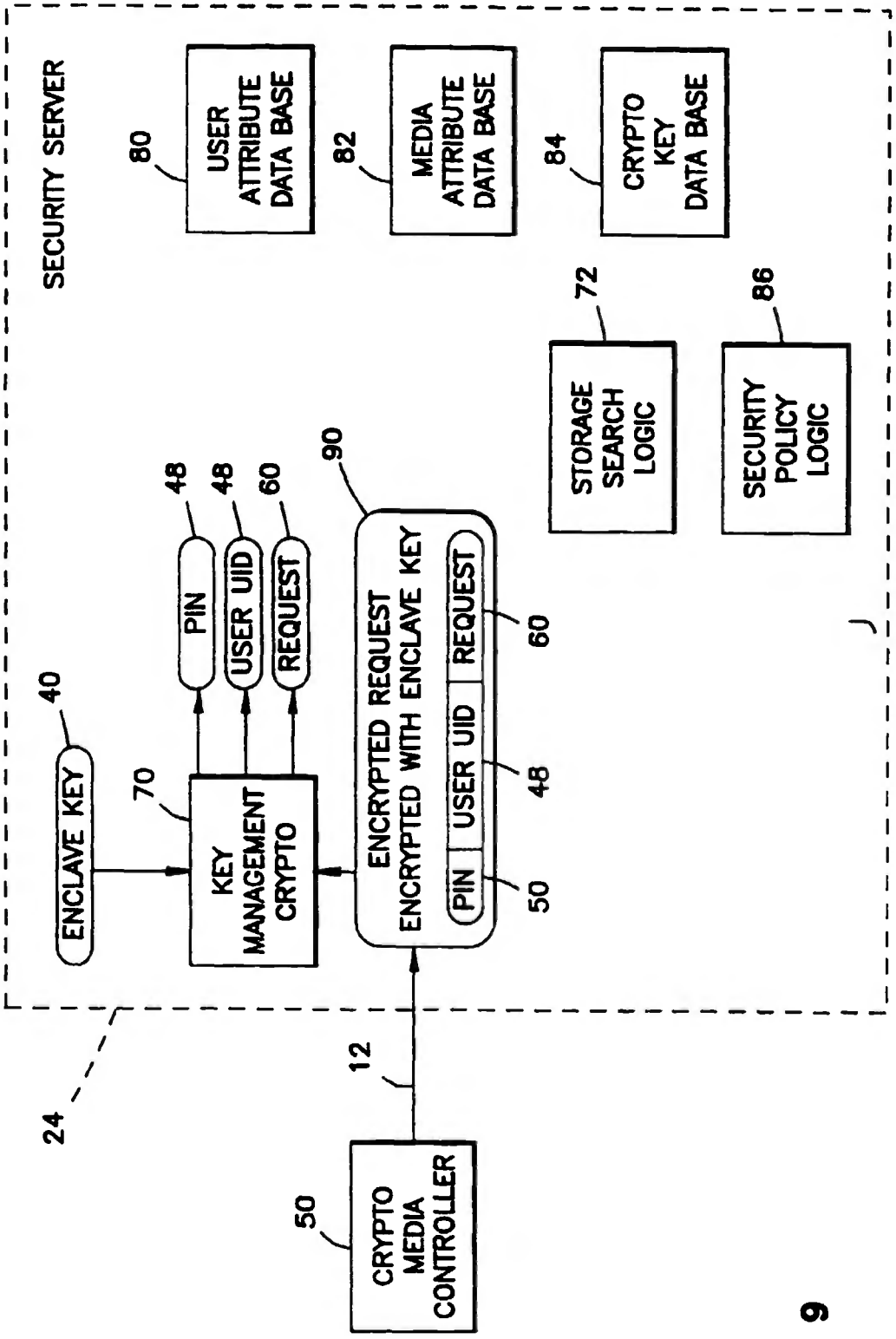

SECURITY SERVER
80
USER ATTRIBUTE DATA BASE
82
MEDIA ATTRIBUTE DATA BASE
84
CRYPTO KEY DATA BASE
72
STORAGE SEARCH LOGIC
86
SECURITY POLICY LOGIC
48
PIN
48
USER UID
60
REQUEST
90
ENCRYPTED REQUEST ENCRYPTED WITH ENCLAVE KEY
PIN
USER UID
REQUEST
60
48
50
40
ENCLAVE KEY
70
KEY MANAGEMENT CRYPTO
24
12
50
CRYPTO MEDIA CONTROLLER

FIG. 9

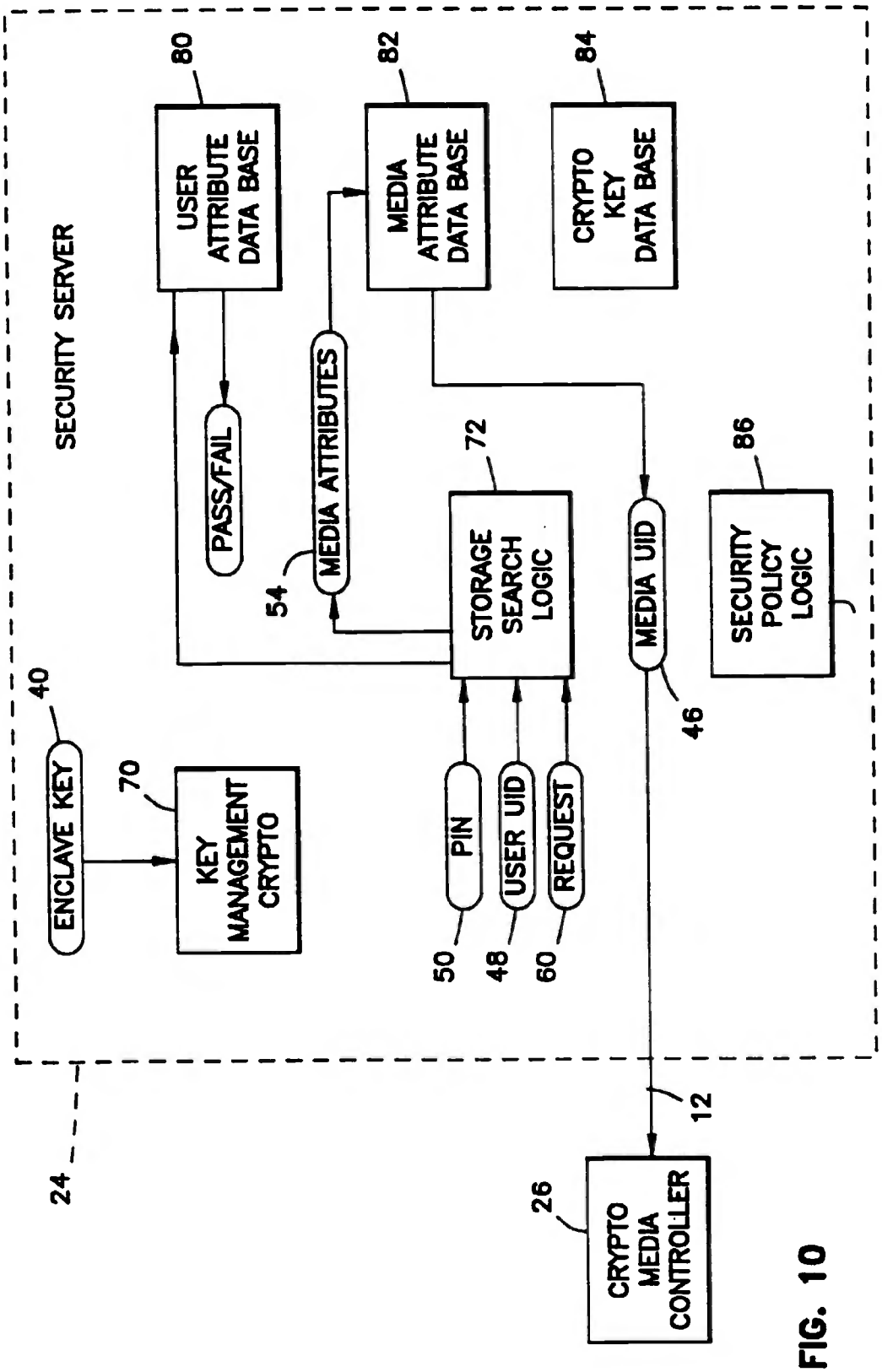

SECURITY SERVER
USER ATTRIBUTE DATA BASE
80
MEDIA ATTRIBUTE DATA BASE
82
CRYPTO KEY DATA BASE
84
PASS/FAIL
MEDIA ATTRIBUTES
54
STORAGE SEARCH LOGIC
72
MEDIA UID
46
SECURITY POLICY LOGIC
86
ENCLAVE KEY
40
KEY MANAGEMENT CRYPTO
70
PIN
50
USER UID
48
REQUEST
60
24
12
CRYPTO MEDIA CONTROLLER
26

FIG. 10

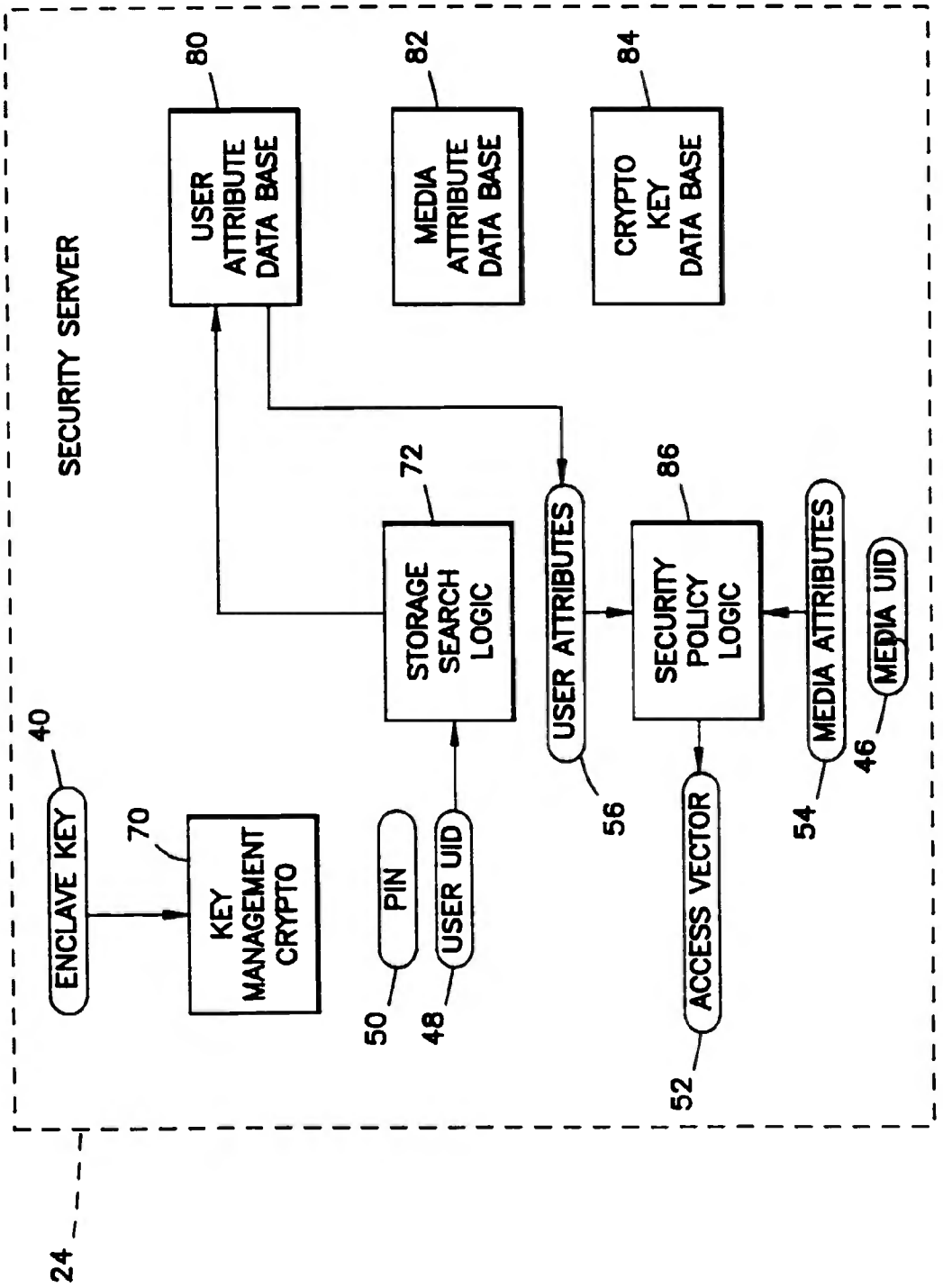

SECURITY SERVER
80
USER ATTRIBUTE DATA BASE
82
MEDIA ATTRIBUTE DATA BASE
84
CRYPTO KEY DATA BASE
72
STORAGE SEARCH LOGIC
86
SECURITY POLICY LOGIC
USER ATTRIBUTES
56
MEDIA ATTRIBUTES
54
MEDIA UID
46
40
ENCLAVE KEY
70
KEY MANAGEMENT CRYPTO
PIN
50
USER UID
48
ACCESS VECTOR
52
24

FIG. 11

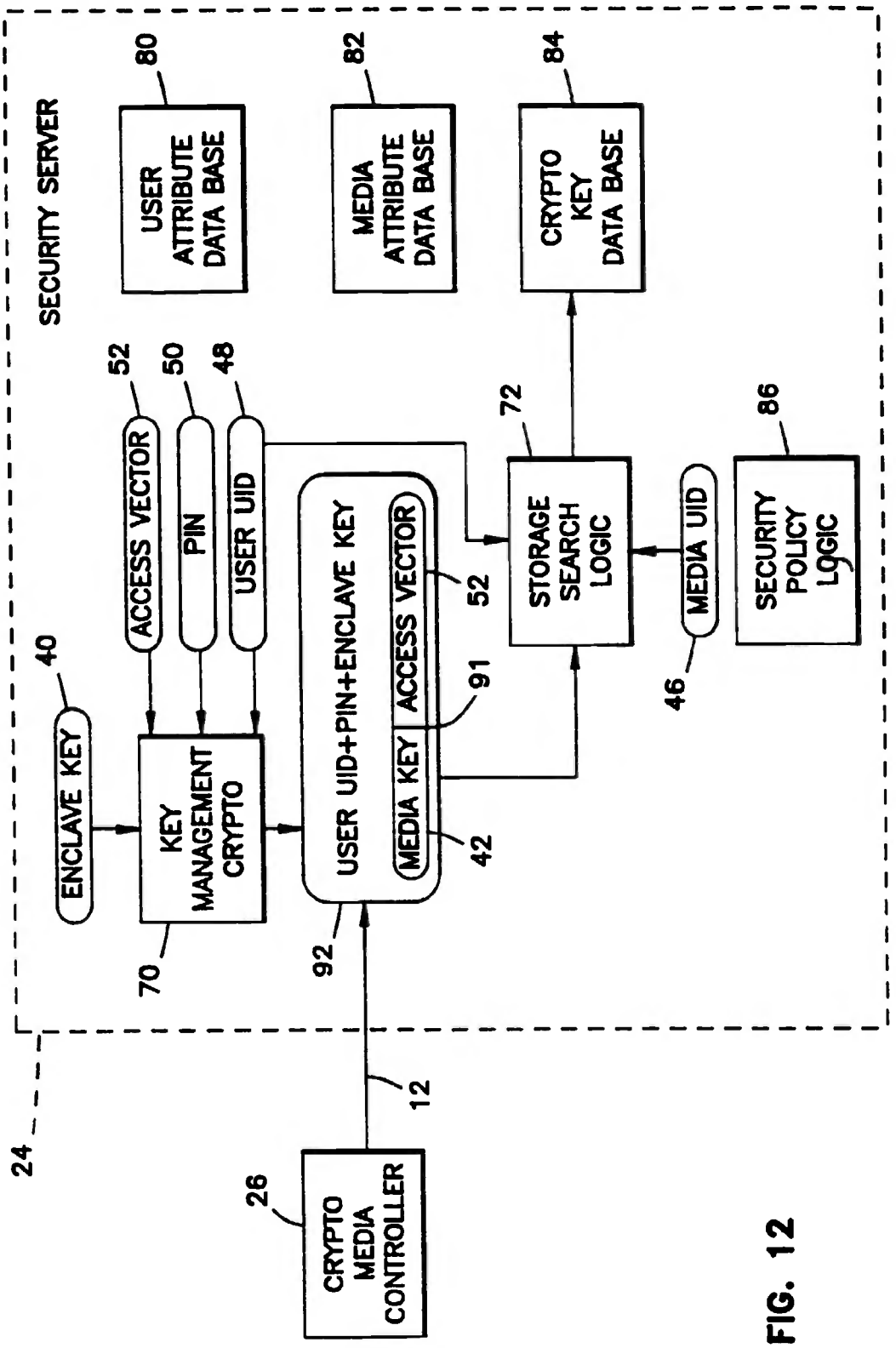

SECURITY SERVER
80
USER ATTRIBUTE DATA BASE
82
MEDIA ATTRIBUTE DATA BASE
84
CRYPTO KEY DATA BASE
52
50
48
ACCESS VECTOR
PIN
USER UID
72
STORAGE SEARCH LOGIC
86
SECURITY POLICY LOGIC
40
ENCLAVE KEY
KEY MANAGEMENT CRYPTO
USER UID+PIN+ENCLAVE KEY
ACCESS VECTOR
MEDIA KEY
52
91
42
MEDIA UID
46
70
92
12
24
26
CRYPTO MEDIA CONTROLLER

FIG. 12

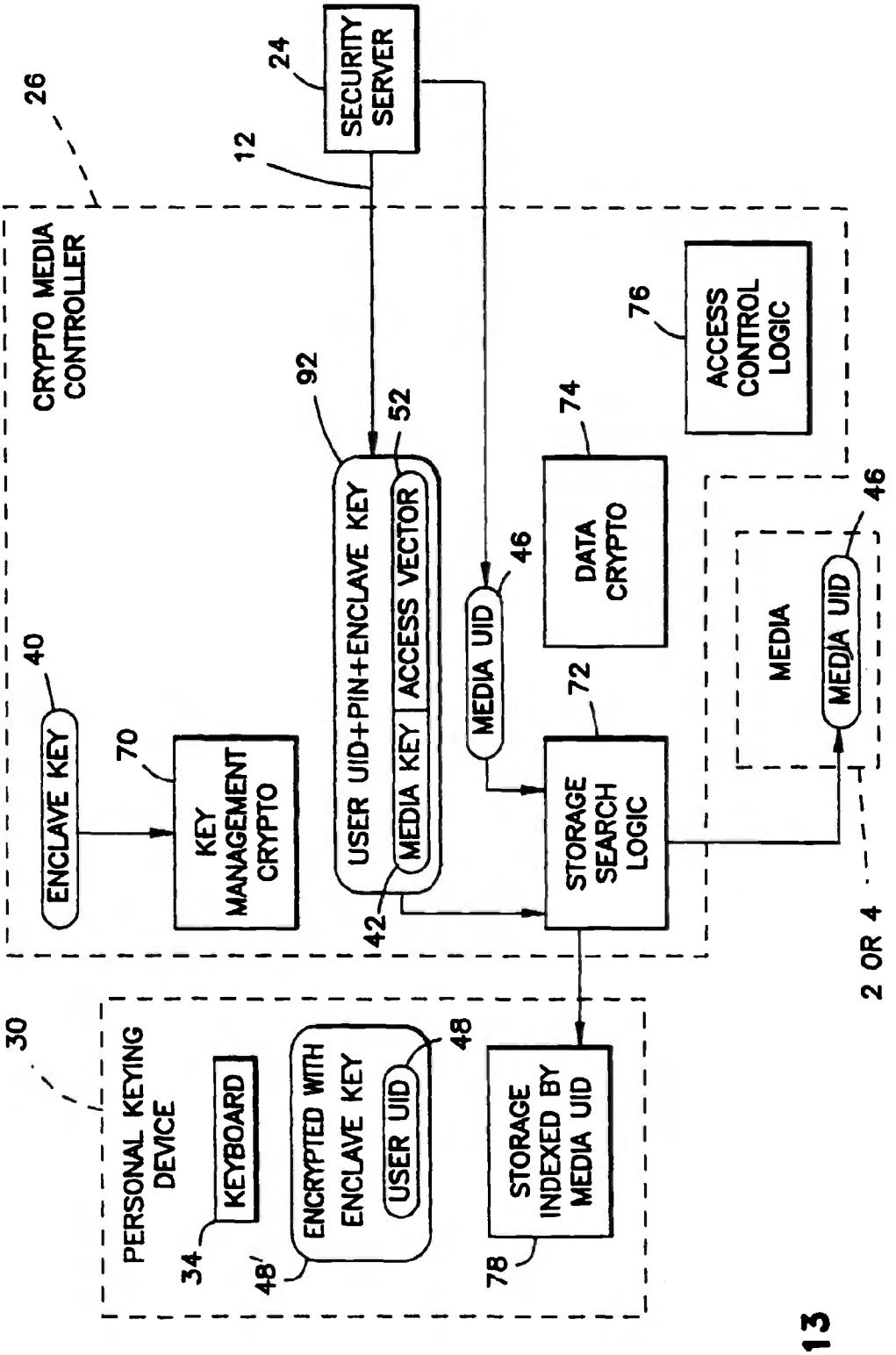

SECURITY SERVER
24
26
12
CRYPTO MEDIA CONTROLLER
ACCESS CONTROL LOGIC
76
92
52
74
USER UID+PIN+ENCLAVE KEY
ACCESS VECTOR
DATA CRYPTO
MEDIA KEY
MEDIA UID
46
MEDIA
MEDIA UID
46
40
ENCLAVE KEY
70
KEY MANAGEMENT CRYPTO
72
STORAGE SEARCH LOGIC
42
2 OR 4
30
PERSONAL KEYING DEVICE
KEYBOARD
34
ENCRYPTED WITH ENCLAVE KEY
USER UID
48
48'
STORAGE INDEXED BY MEDIA UID
78

FIG. 13

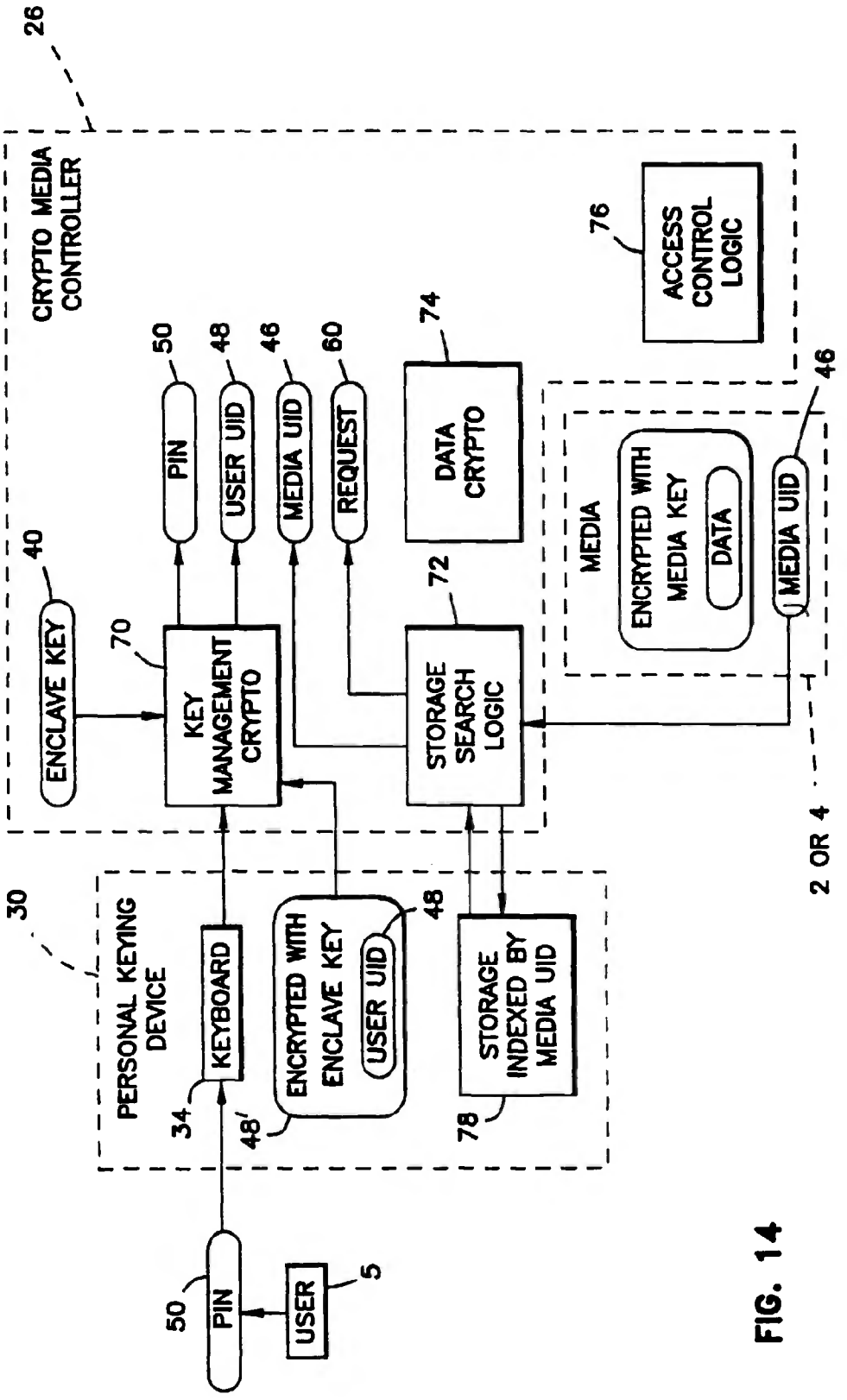

26
CRYPTO MEDIA CONTROLLER
ACCESS CONTROL LOGIC
76
PIN
50
USER UID
48
MEDIA UID
46
REQUEST
60
DATA CRYPTO
74
MEDIA
ENCRYPTED WITH MEDIA KEY
DATA
MEDIA UID
46
ENCLAVE KEY
40
KEY MANAGEMENT CRYPTO
70
STORAGE SEARCH LOGIC
72
2 OR 4
30
PERSONAL KEYING DEVICE
KEYBOARD
34
ENCRYPTED WITH ENCLAVE KEY
USER UID
48
48'
STORAGE INDEXED BY MEDIA UID
78
PIN
50
USER
5

FIG. 14

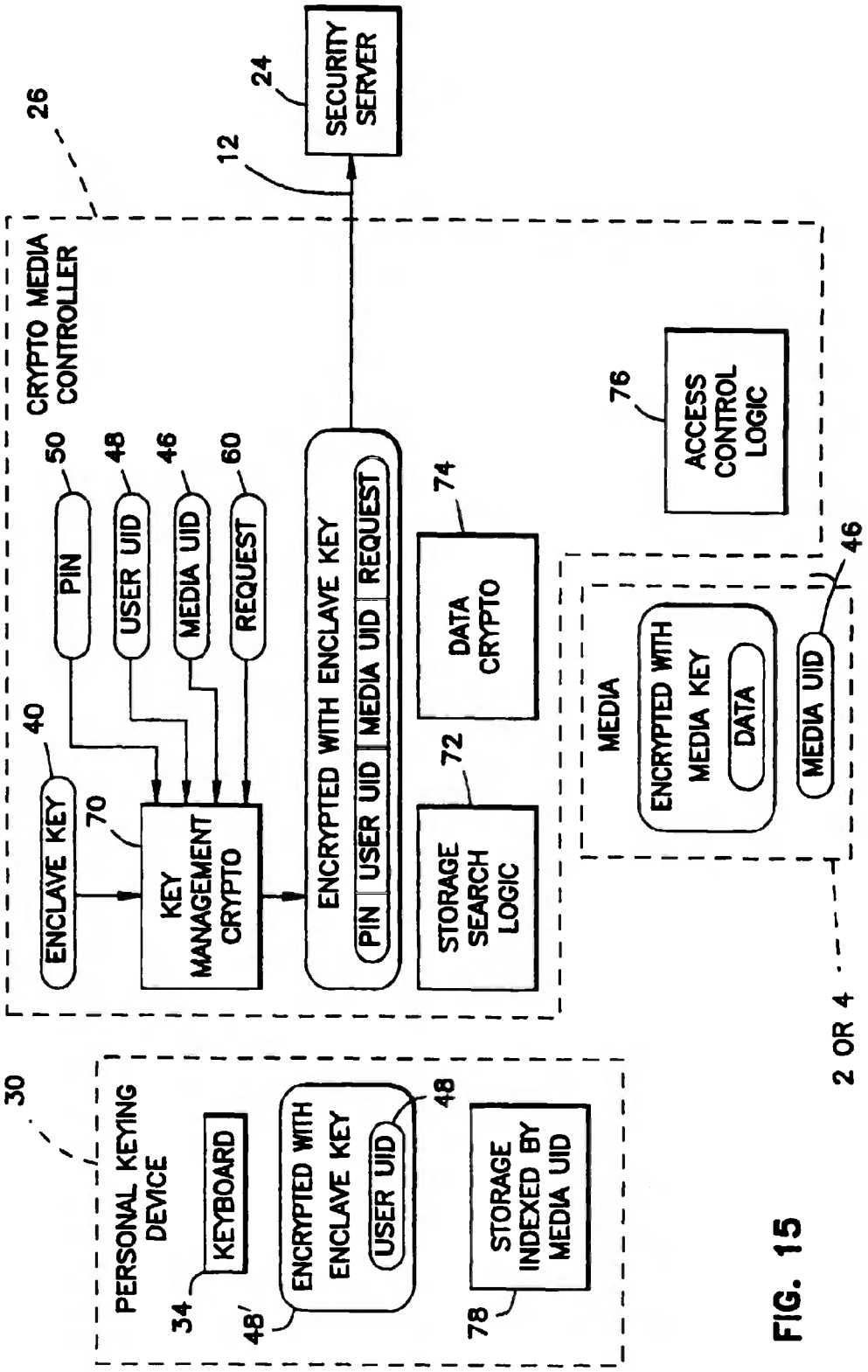

SECURITY SERVER
24
12
26
CRYPTO MEDIA CONTROLLER
PIN
50
USER UID
48
MEDIA UID
46
REQUEST
60
ENCLAVE KEY
40
70
KEY MANAGEMENT CRYPTO
ENCRYPTED WITH ENCLAVE KEY
PIN
USER UID
MEDIA UID
REQUEST
74
DATA CRYPTO
72
STORAGE SEARCH LOGIC
76
ACCESS CONTROL LOGIC
MEDIA
ENCRYPTED WITH MEDIA KEY
DATA
MEDIA UID
46
2 OR 4
30
PERSONAL KEYING DEVICE
KEYBOARD
34
ENCRYPTED WITH ENCLAVE KEY
USER UID
48
48'
STORAGE INDEXED BY MEDIA UID
78

FIG. 15

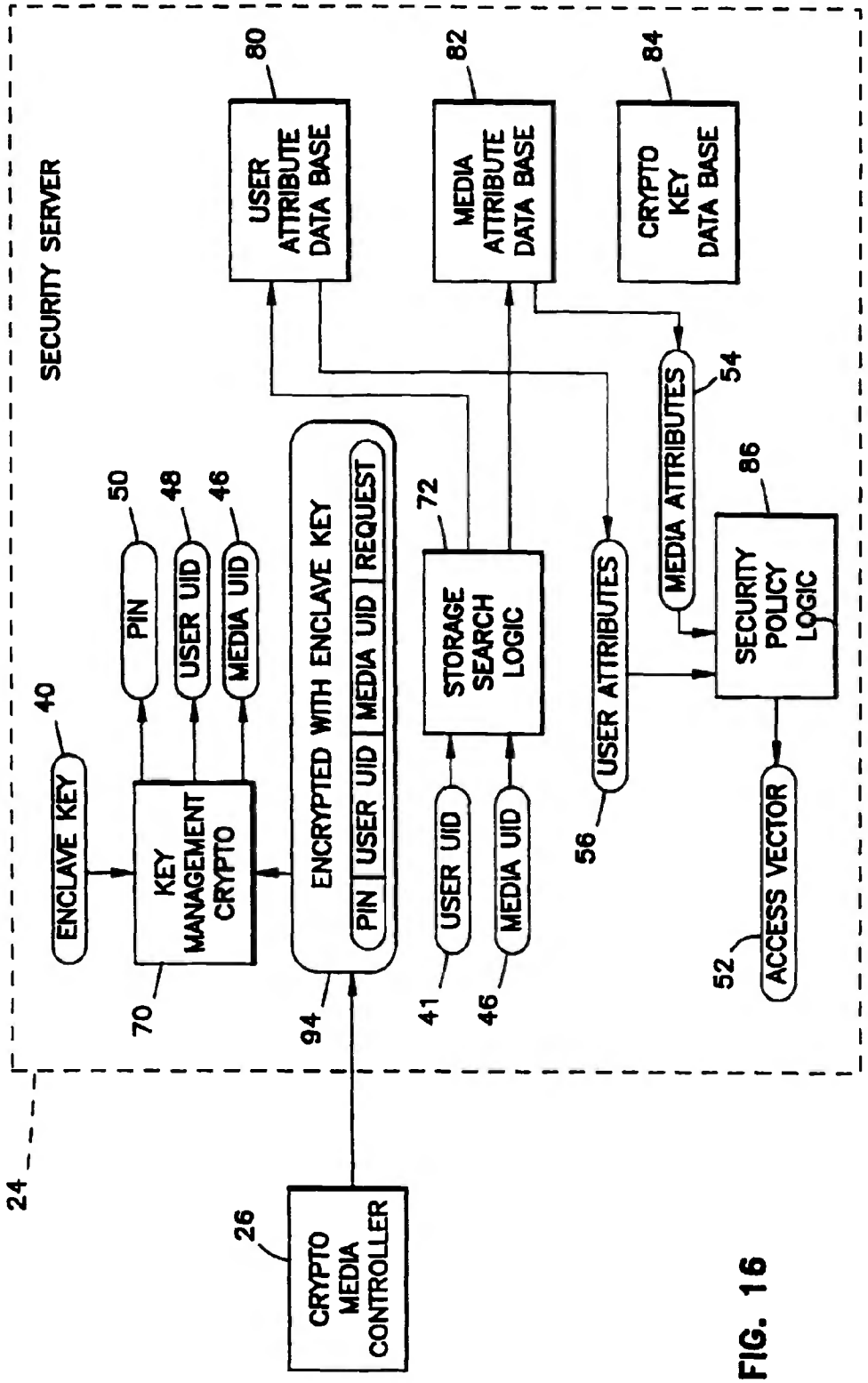

SECURITY SERVER
24
80
USER ATTRIBUTE DATA BASE
82
MEDIA ATTRIBUTE DATA BASE
84
CRYPTO KEY DATA BASE
54
MEDIA ATTRIBUTES
86
SECURITY POLICY LOGIC
50
PIN
48
USER UID
46
MEDIA UID
40
ENCLAVE KEY
KEY MANAGEMENT CRYPTO
70
ENCRYPTED WITH ENCLAVE KEY
PIN
USER UID
MEDIA UID
REQUEST
94
72
STORAGE SEARCH LOGIC
41
USER UID
46
MEDIA UID
56
USER ATTRIBUTES
52
ACCESS VECTOR
26
CRYPTO MEDIA CONTROLLER

FIG. 16

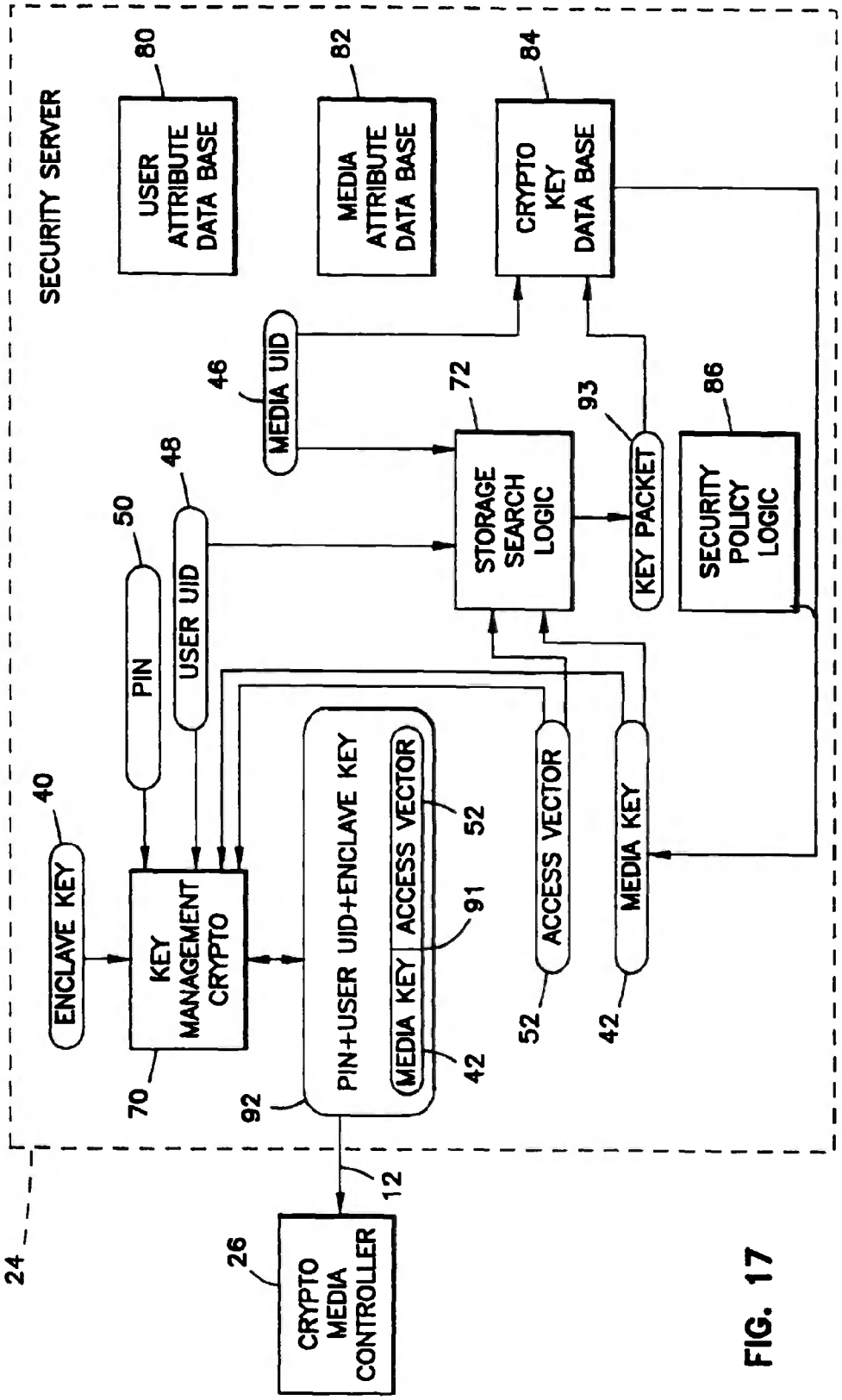

SECURITY SERVER
80
USER ATTRIBUTE DATA BASE
82
MEDIA ATTRIBUTE DATA BASE
84
CRYPTO KEY DATA BASE
46
MEDIA UID
72
93
86
48
50
STORAGE SEARCH LOGIC
KEY PACKET
SECURITY POLICY LOGIC
PIN
USER UID
40
ENCLAVE KEY
KEY MANAGEMENT CRYPTO
PIN+USER UID+ENCLAVE KEY
ACCESS VECTOR
52
91
MEDIA KEY
42
ACCESS VECTOR
MEDIA KEY
52
42
70
92
12
24
26
CRYPTO MEDIA CONTROLLER

FIG. 17

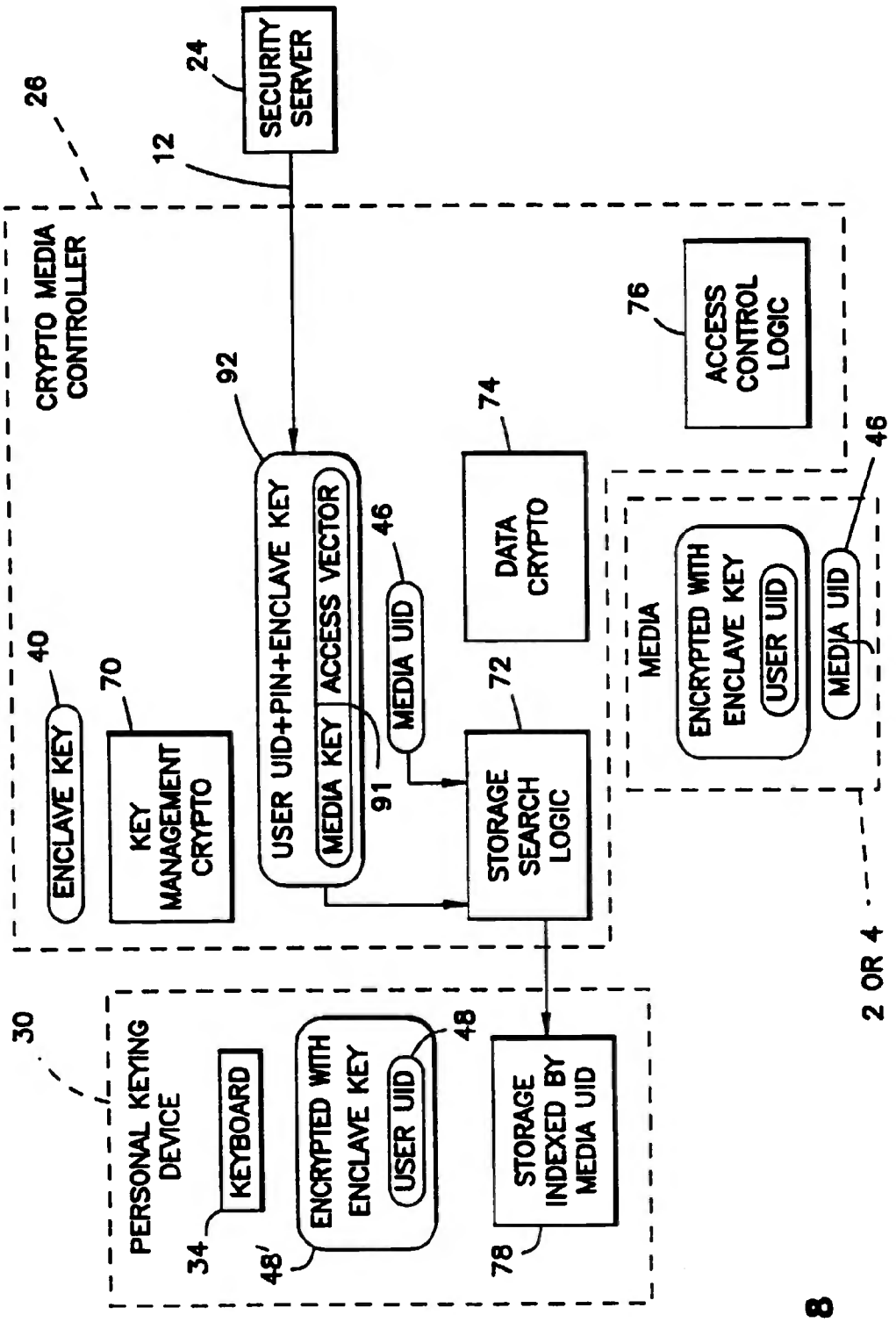

SECURITY SERVER
24
26
12
CRYPTO MEDIA CONTROLLER
92
USER UID+PIN+ENCLAVE KEY
MEDIA KEY
ACCESS VECTOR
91
46
MEDIA UID
74
DATA CRYPTO
76
ACCESS CONTROL LOGIC
40
ENCLAVE KEY
70
KEY MANAGEMENT CRYPTO
72
STORAGE SEARCH LOGIC
MEDIA
ENCRYPTED WITH ENCLAVE KEY
USER UID
MEDIA UID
46
2 OR 4
30
PERSONAL KEYING DEVICE
KEYBOARD
34
ENCRYPTED WITH ENCLAVE KEY
USER UID
48
48'
STORAGE INDEXED BY MEDIA UID
78

FIG. 18

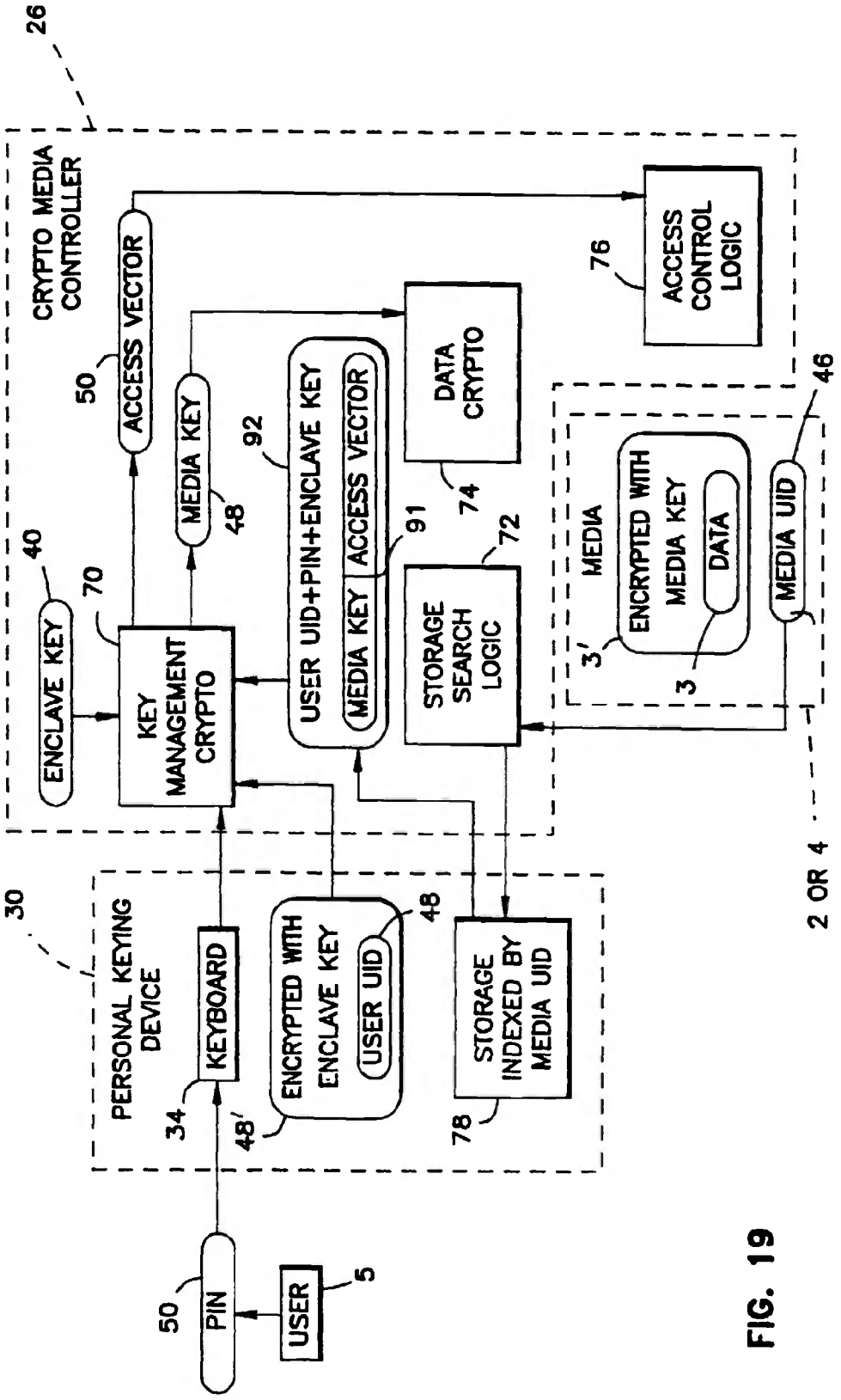

26
CRYPTO MEDIA CONTROLLER
ACCESS VECTOR
50
MEDIA KEY
48
92
USER UID+PIN+ENCLAVE KEY
MEDIA KEY
ACCESS VECTOR
91
DATA CRYPTO
74
ACCESS CONTROL LOGIC
76
ENCLAVE KEY
40
70
KEY MANAGEMENT CRYPTO
STORAGE SEARCH LOGIC
72
MEDIA
3'
ENCRYPTED WITH MEDIA KEY
DATA
3
MEDIA UID
46
2 OR 4
30
PERSONAL KEYING DEVICE
KEYBOARD
34
ENCRYPTED WITH ENCLAVE KEY
USER UID
48
48'
STORAGE INDEXED BY MEDIA UID
78
PIN
50
USER
5

FIG. 19

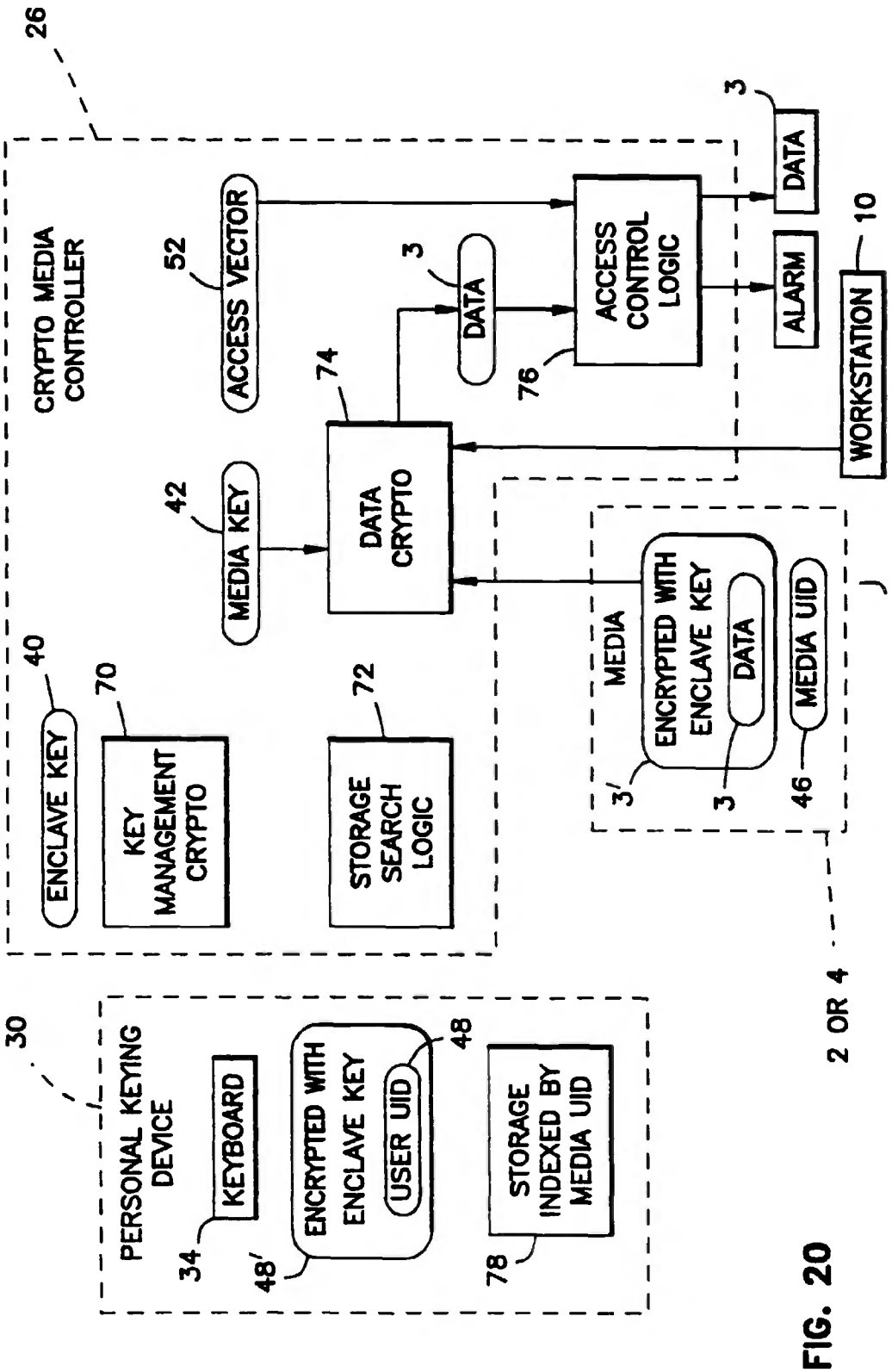

26
CRYPTO MEDIA CONTROLLER
40
ENCLAVE KEY
70
KEY MANAGEMENT CRYPTO
72
STORAGE SEARCH LOGIC
42
MEDIA KEY
74
DATA CRYPTO
52
ACCESS VECTOR
3
DATA
76
ACCESS CONTROL LOGIC
3
DATA
ALARM
10
WORKSTATION
MEDIA
3'
ENCRYPTED WITH ENCLAVE KEY
3
DATA
46
MEDIA UID
2 OR 4
30
PERSONAL KEYING DEVICE
34
KEYBOARD
48'
ENCRYPTED WITH ENCLAVE KEY
48
USER UID
78
STORAGE INDEXED BY MEDIA UID

FIG. 20

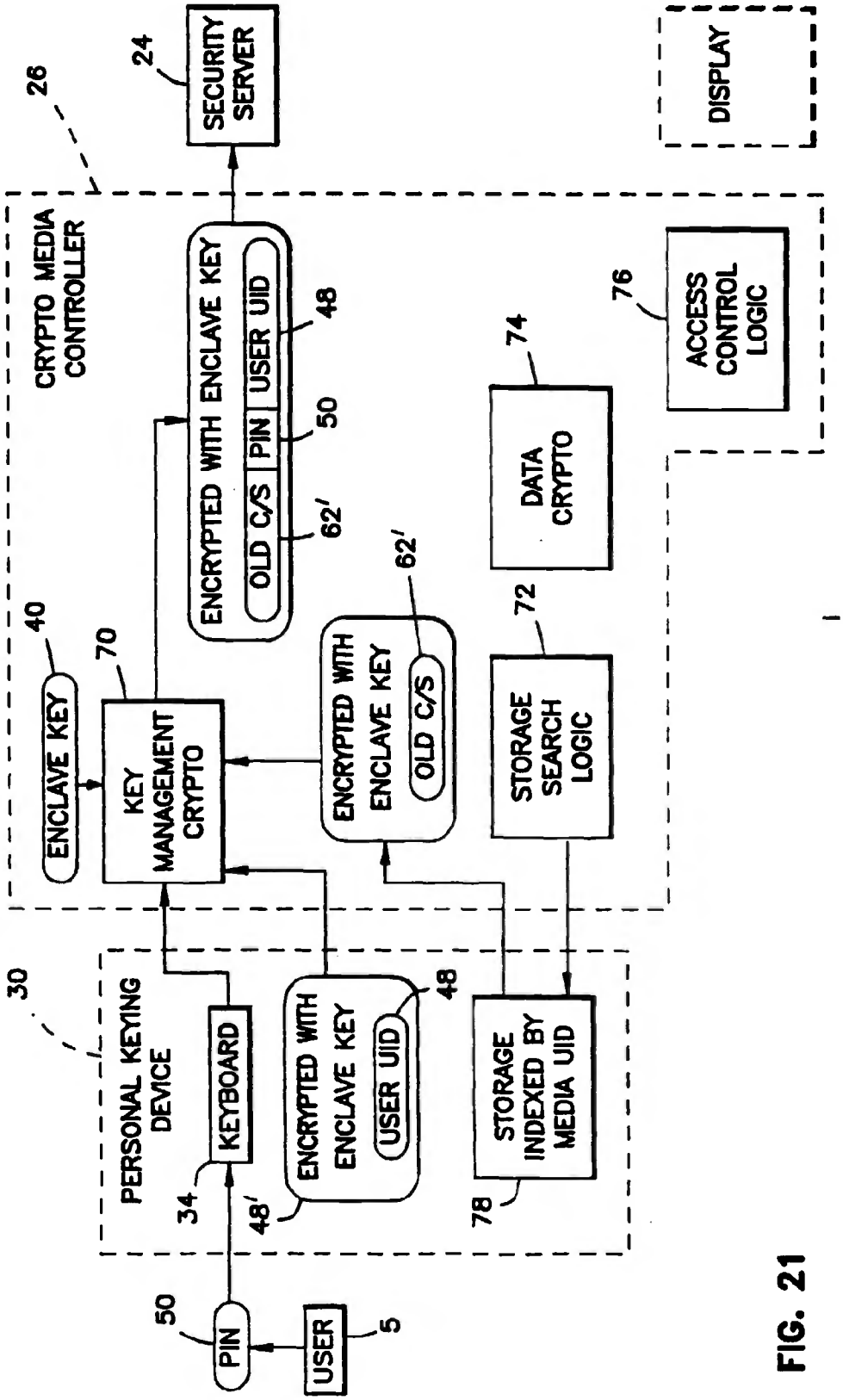

SECURITY SERVER
24
26
DISPLAY
CRYPTO MEDIA CONTROLLER
ENCRYPTED WITH ENCLAVE KEY
USER UID
PIN
OLD C/S
48
50
62'
ACCESS CONTROL LOGIC
76
74
DATA CRYPTO
40
ENCLAVE KEY
70
KEY MANAGEMENT CRYPTO
ENCRYPTED WITH ENCLAVE KEY
OLD C/S
62'
72
STORAGE SEARCH LOGIC
30
PERSONAL KEYING DEVICE
KEYBOARD
34
ENCRYPTED WITH ENCLAVE KEY
USER UID
48'
48
STORAGE INDEXED BY MEDIA UID
78
50
PIN
USER
5

FIG. 21

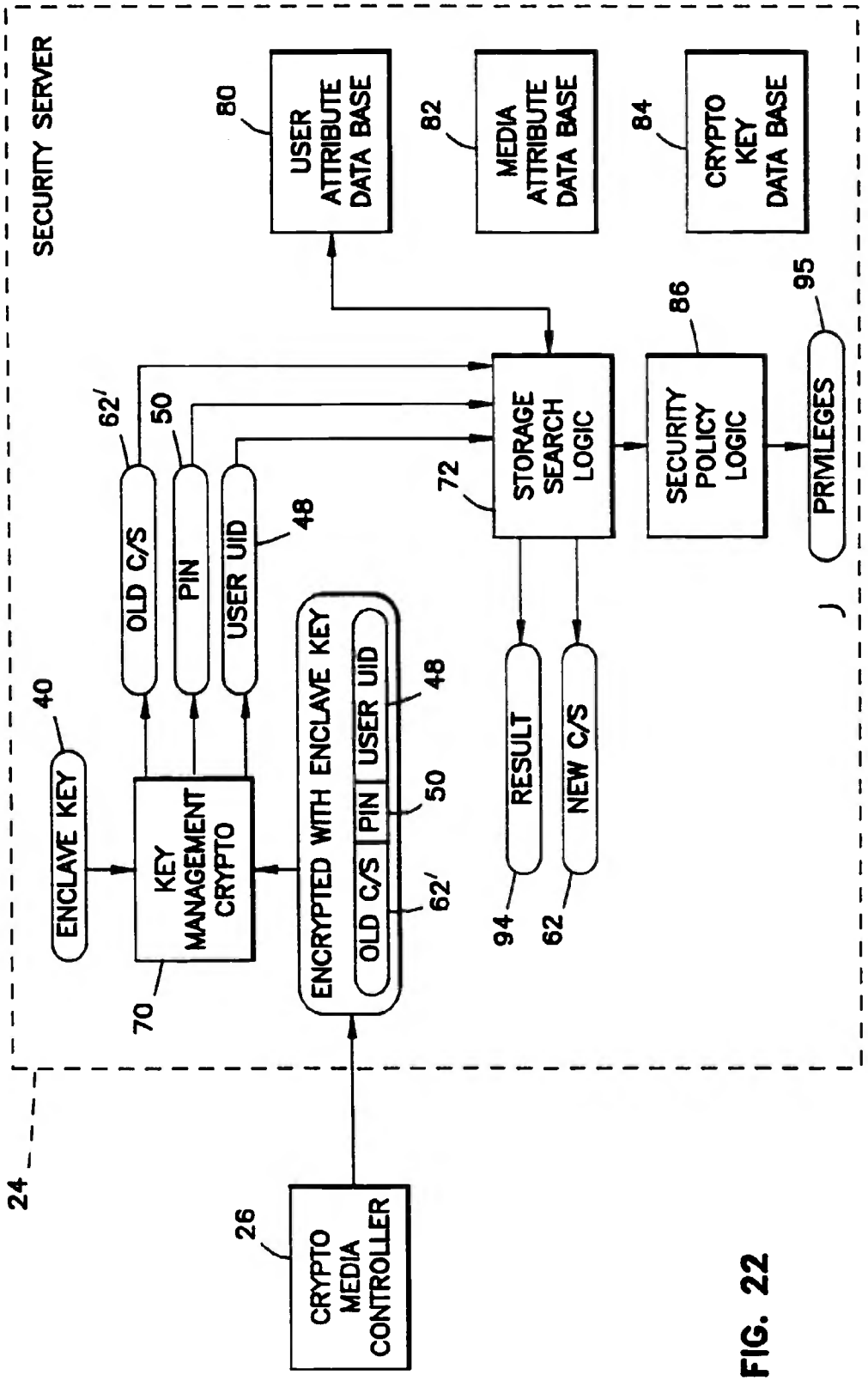

SECURITY SERVER
80
USER ATTRIBUTE DATA BASE
82
MEDIA ATTRIBUTE DATA BASE
84
CRYPTO KEY DATA BASE
86
SECURITY POLICY LOGIC
95
PRIVILEGES
72
STORAGE SEARCH LOGIC
62'
OLD C/S
50
PIN
48
USER UID
40
ENCLAVE KEY
KEY MANAGEMENT CRYPTO
70
ENCRYPTED WITH ENCLAVE KEY
OLD C/S
PIN
USER UID
48
50
62'
RESULT
94
NEW C/S
62
24
26
CRYPTO MEDIA CONTROLLER

FIG. 22

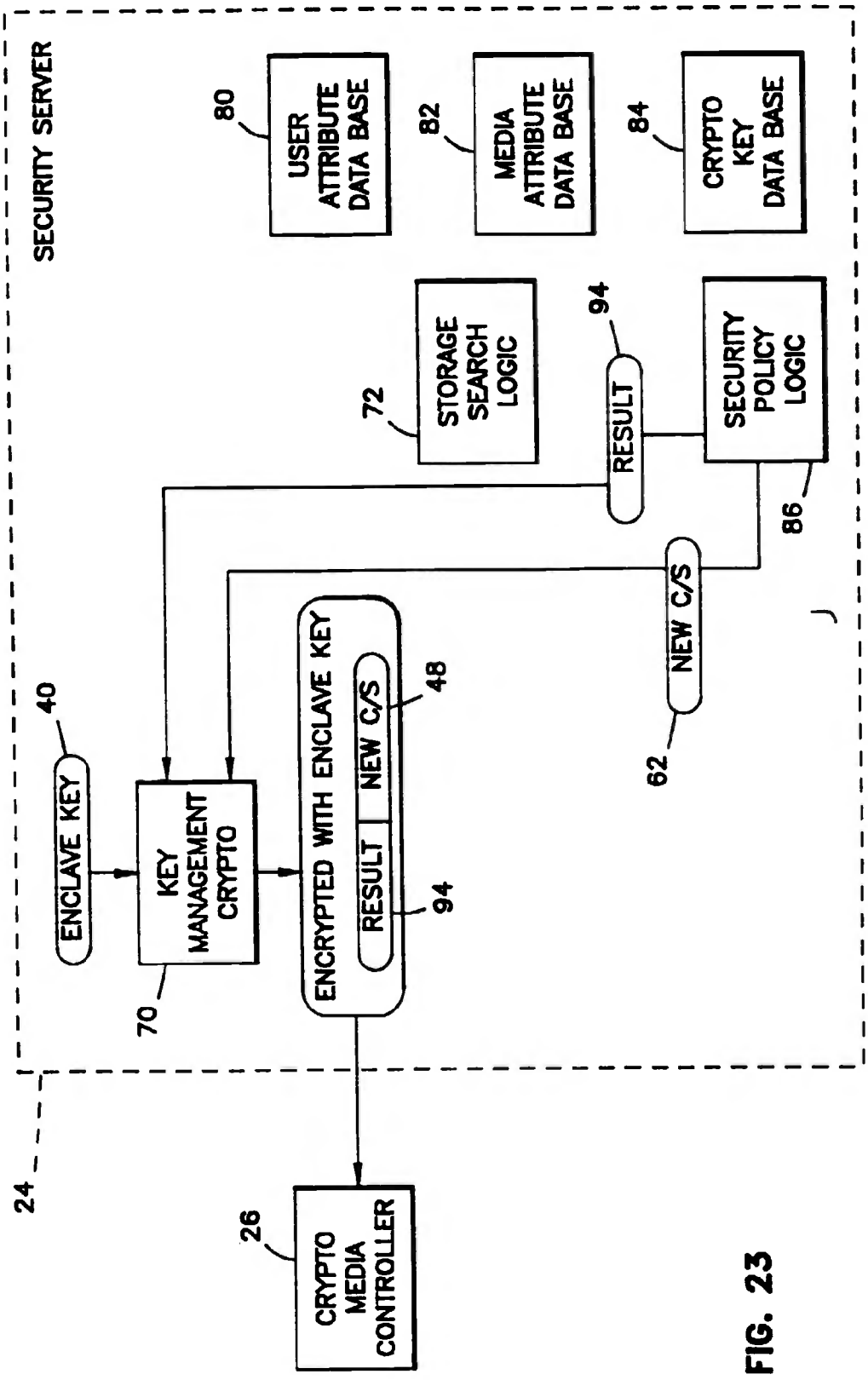

SECURITY SERVER
80
USER ATTRIBUTE DATA BASE
82
MEDIA ATTRIBUTE DATA BASE
84
CRYPTO KEY DATA BASE
72
STORAGE SEARCH LOGIC
94
RESULT
SECURITY POLICY LOGIC
86
NEW C/S
62
40
ENCLAVE KEY
KEY MANAGEMENT CRYPTO
70
ENCRYPTED WITH ENCLAVE KEY
RESULT
NEW C/S
48
94
24
26
CRYPTO MEDIA CONTROLLER

FIG. 23

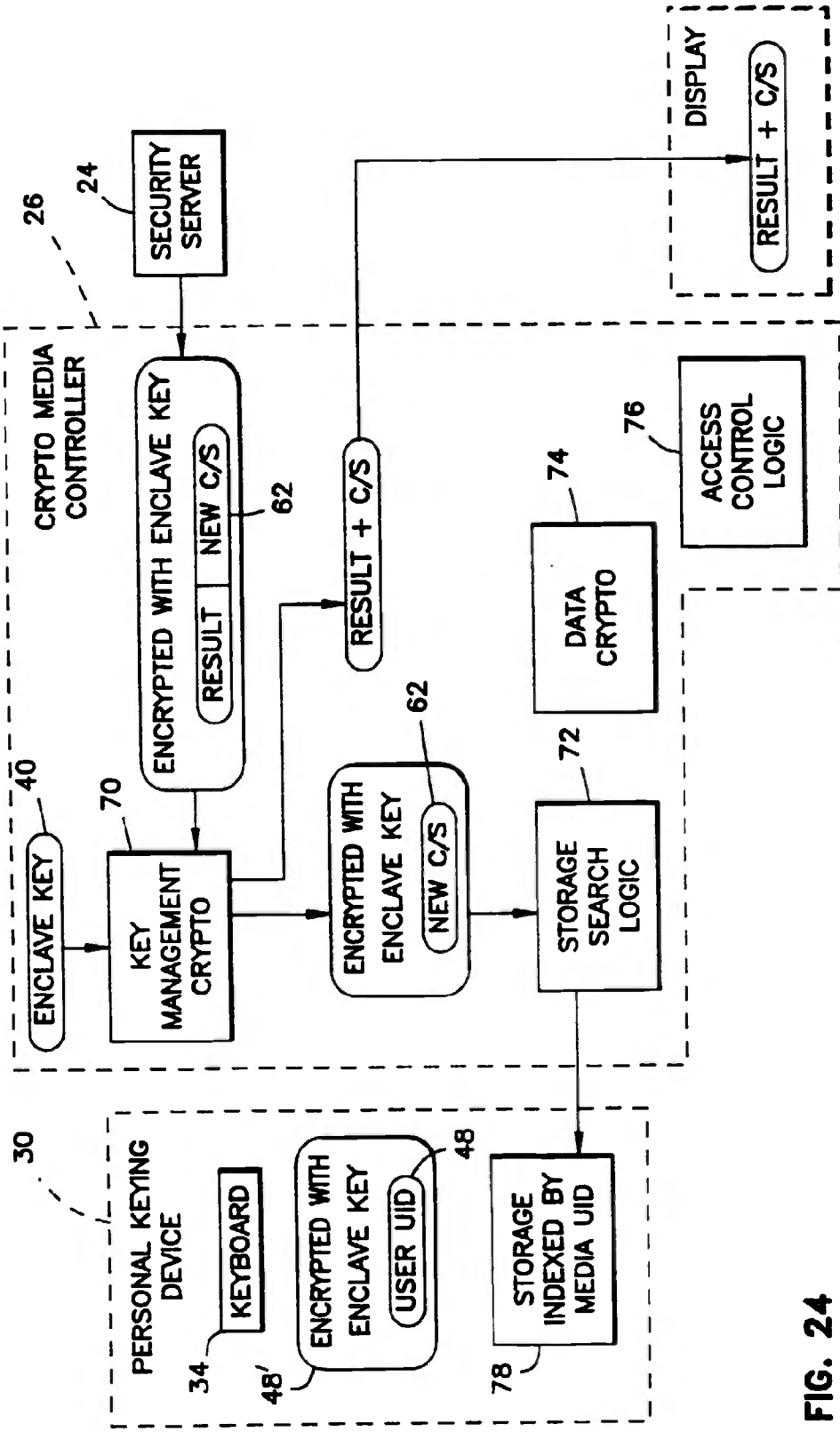

DISPLAY
RESULT + C/S
SECURITY SERVER
24
26
CRYPTO MEDIA CONTROLLER
ENCRYPTED WITH ENCLAVE KEY
RESULT
NEW C/S
62
RESULT + C/S
ACCESS CONTROL LOGIC
76
74
DATA CRYPTO
ENCLAVE KEY
40
70
KEY MANAGEMENT CRYPTO
ENCRYPTED WITH ENCLAVE KEY
NEW C/S
62
72
STORAGE SEARCH LOGIC
30
PERSONAL KEYING DEVICE
KEYBOARD
34
ENCRYPTED WITH ENCLAVE KEY
USER UID
48
48'
STORAGE INDEXED BY MEDIA UID
78

FIG. 24

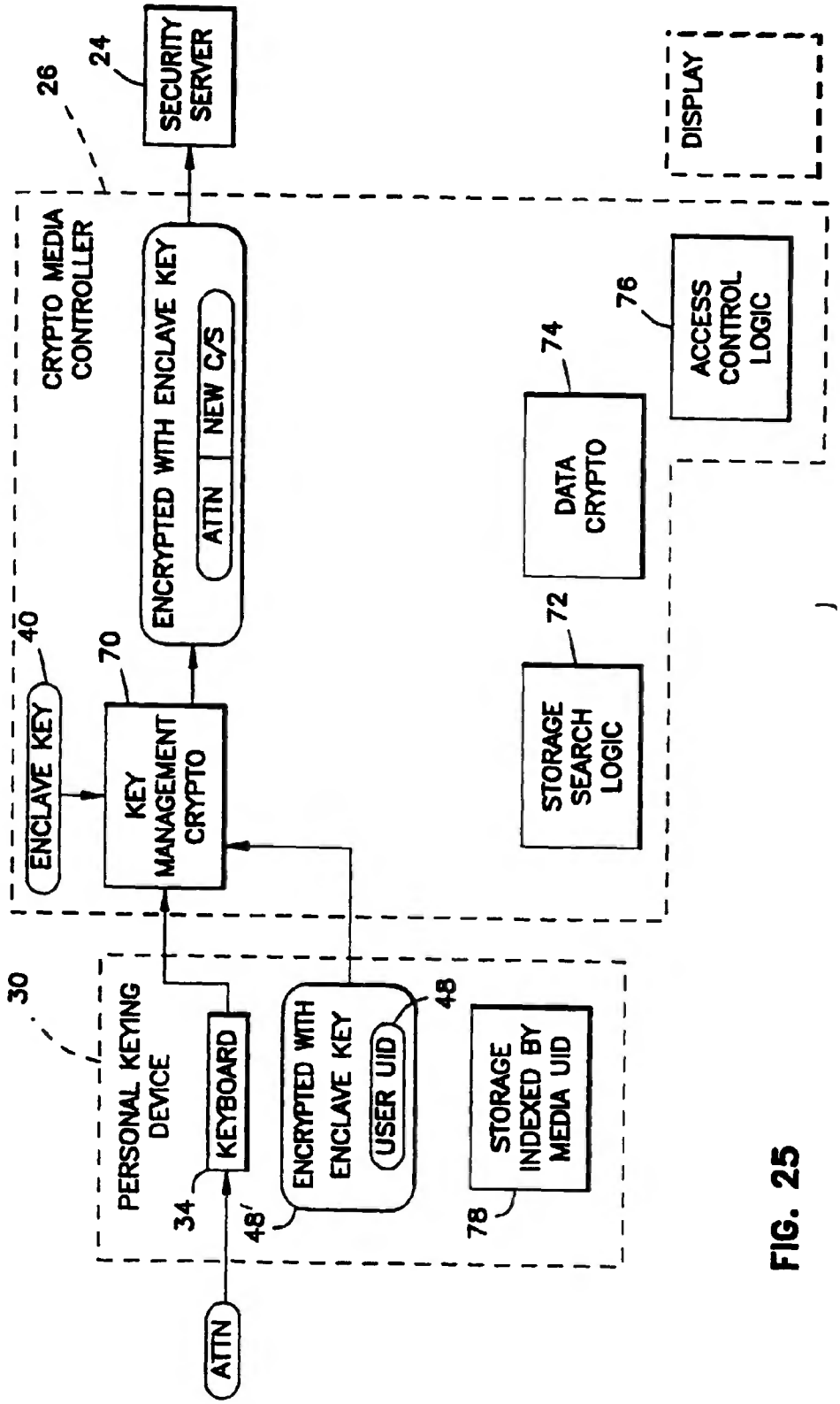

SECURITY SERVER
24
26
CRYPTO MEDIA CONTROLLER
DISPLAY
ENCRYPTED WITH ENCLAVE KEY
ATTN
NEW C/S
ACCESS CONTROL LOGIC
76
74
DATA CRYPTO
72
STORAGE SEARCH LOGIC
40
ENCLAVE KEY
70
KEY MANAGEMENT CRYPTO
30
PERSONAL KEYING DEVICE
KEYBOARD
34
48'
ENCRYPTED WITH ENCLAVE KEY
USER UID
48
STORAGE INDEXED BY MEDIA UID
78
ATTN

FIG. 25

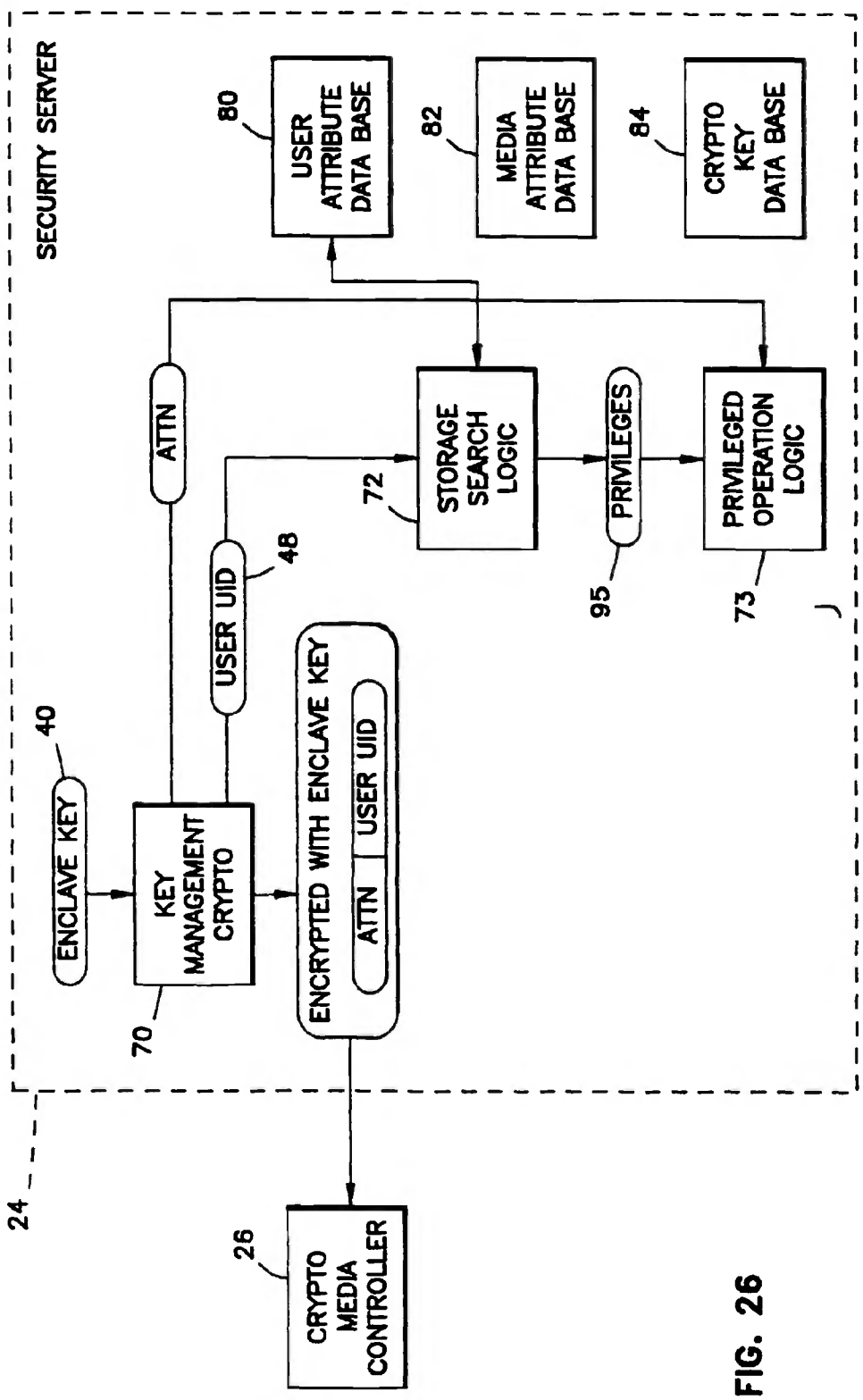

SECURITY SERVER
80
USER ATTRIBUTE DATA BASE
82
MEDIA ATTRIBUTE DATA BASE
84
CRYPTO KEY DATA BASE
ATTN
72
STORAGE SEARCH LOGIC
PRIVILEGES
PRIVILEGED OPERATION LOGIC
95
73
USER UID
48
40
ENCLAVE KEY
KEY MANAGEMENT CRYPTO
70
ENCRYPTED WITH ENCLAVE KEY
ATTN
USER UID
24
26
CRYPTO MEDIA CONTROLLER

FIG. 26

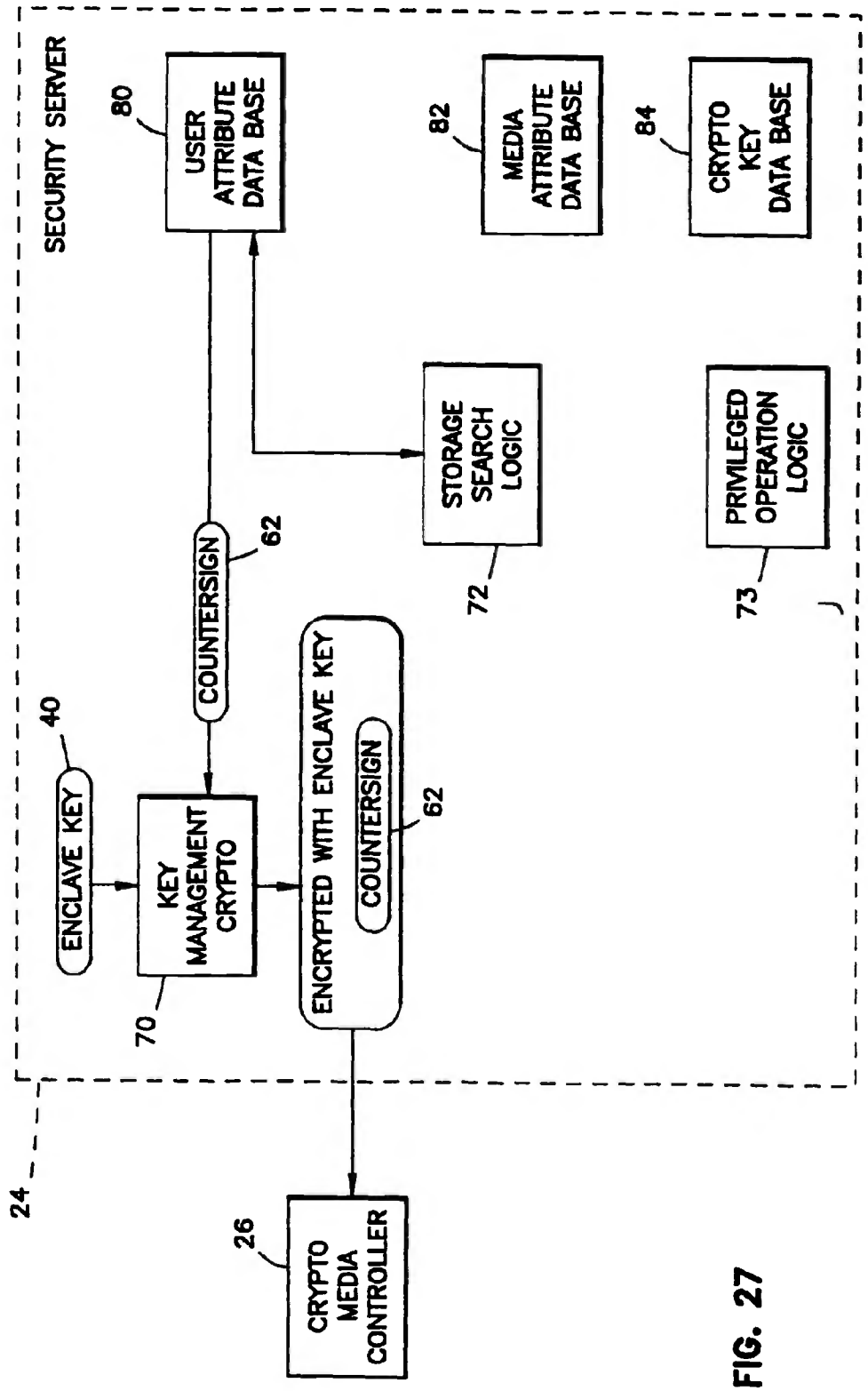

SECURITY SERVER
80
USER ATTRIBUTE DATA BASE
82
MEDIA ATTRIBUTE DATA BASE
84
CRYPTO KEY DATA BASE
STORAGE SEARCH LOGIC
72
PRIVILEGED OPERATION LOGIC
73
COUNTERSIGN
62
40
ENCLAVE KEY
KEY MANAGEMENT CRYPTO
70
ENCRYPTED WITH ENCLAVE KEY
COUNTERSIGN
62
24
26
CRYPTO MEDIA CONTROLLER

FIG. 27

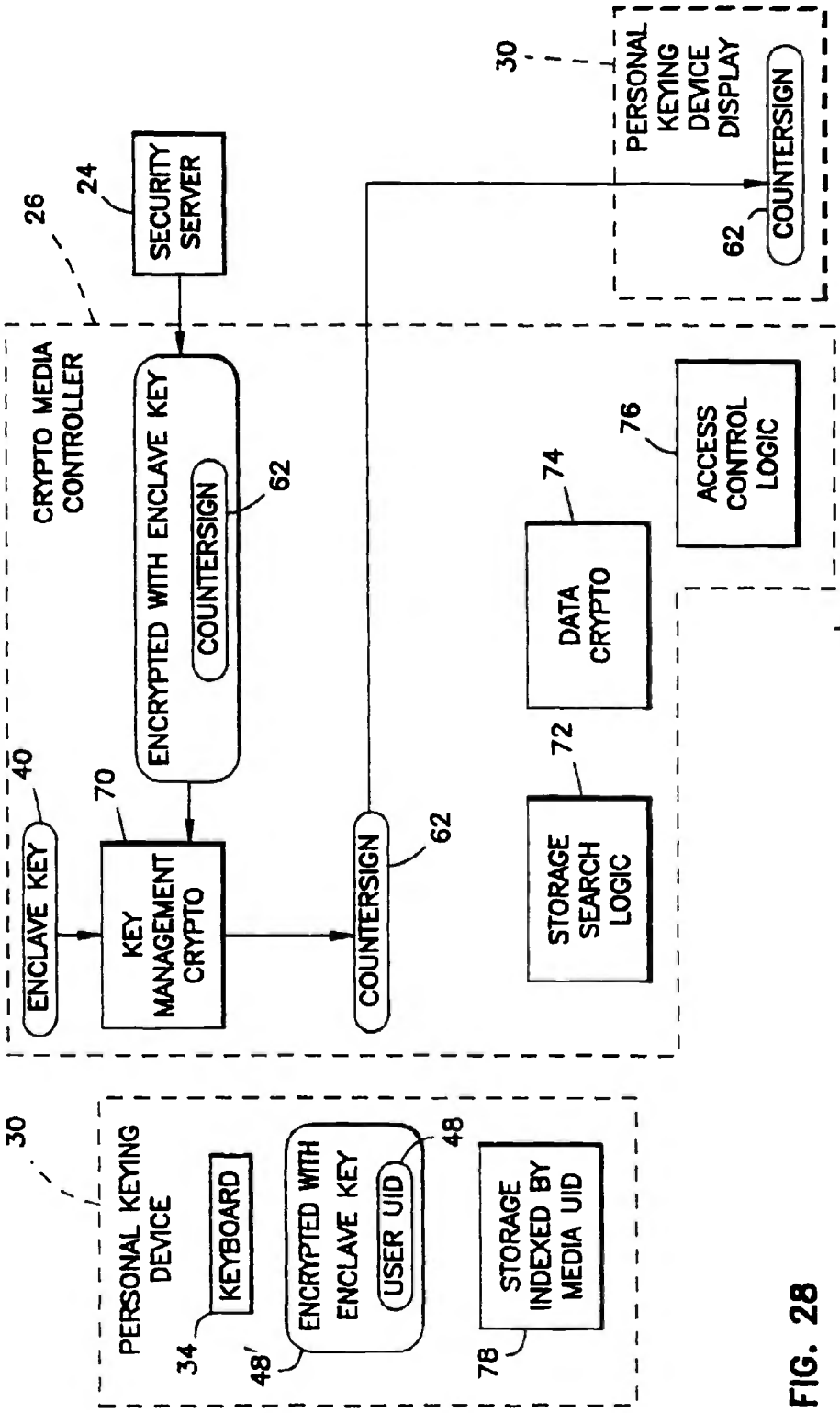

30
PERSONAL KEYING DEVICE DISPLAY
COUNTERSIGN
62
30
PERSONAL KEYING DEVICE
KEYBOARD
34
48'
ENCRYPTED WITH ENCLAVE KEY
USER UID
48
78
STORAGE INDEXED BY MEDIA UID
26
CRYPTO MEDIA CONTROLLER
24
SECURITY SERVER
ENCRYPTED WITH ENCLAVE KEY
COUNTERSIGN
62
40
ENCLAVE KEY
70
KEY MANAGEMENT CRYPTO
COUNTERSIGN
62
72
STORAGE SEARCH LOGIC
74
DATA CRYPTO
76
ACCESS CONTROL LOGIC

FIG. 28

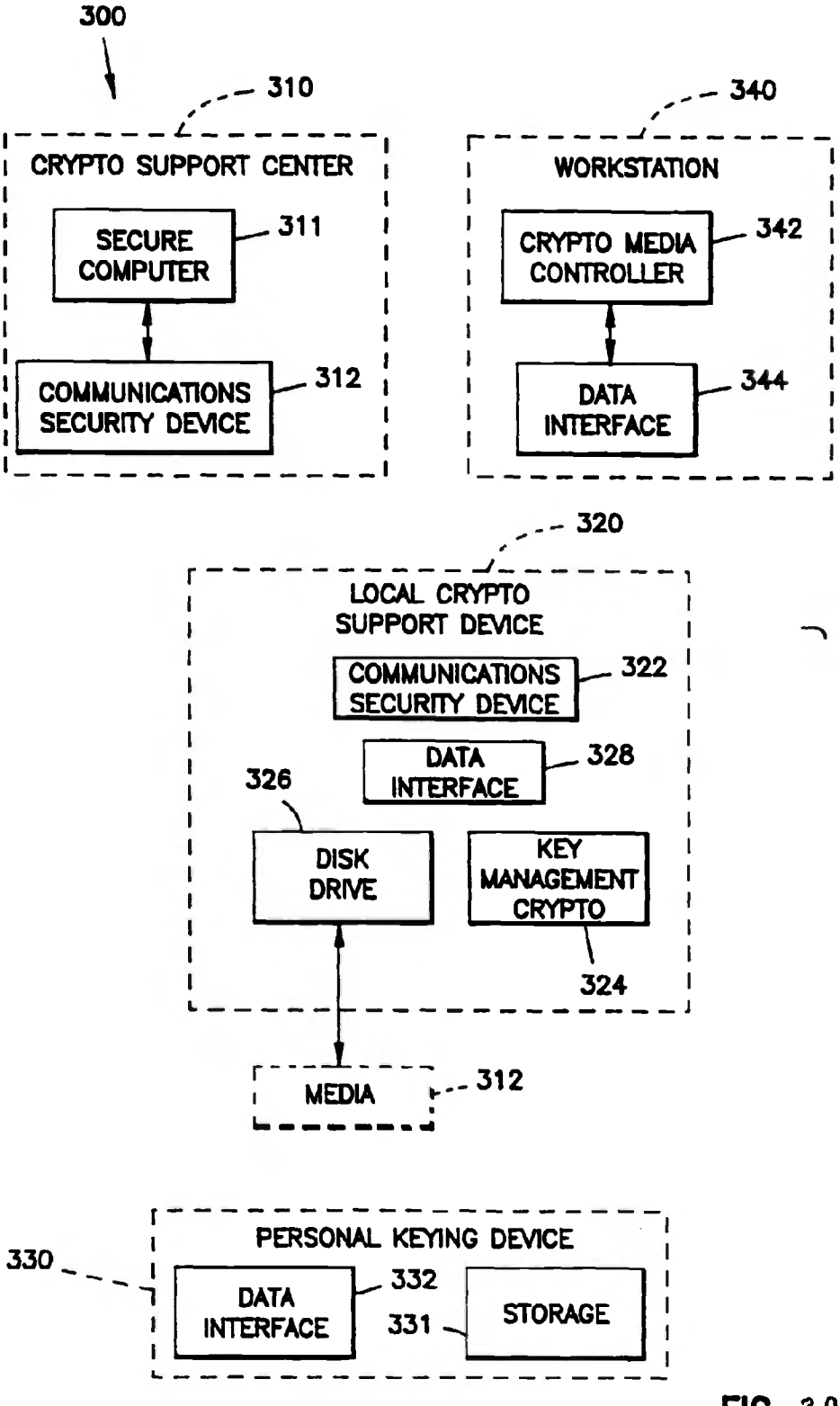

300
310
CRYPTO SUPPORT CENTER
SECURE COMPUTER
311
COMMUNICATIONS SECURITY DEVICE
312
340
WORKSTATION
CRYPTO MEDIA CONTROLLER
342
DATA INTERFACE
344
320
LOCAL CRYPTO SUPPORT DEVICE
COMMUNICATIONS SECURITY DEVICE
322
DATA INTERFACE
328
326
DISK DRIVE
KEY MANAGEMENT CRYPTO
324
MEDIA
312
330
PERSONAL KEYING DEVICE
DATA INTERFACE
332
331
STORAGE

FIG. 29

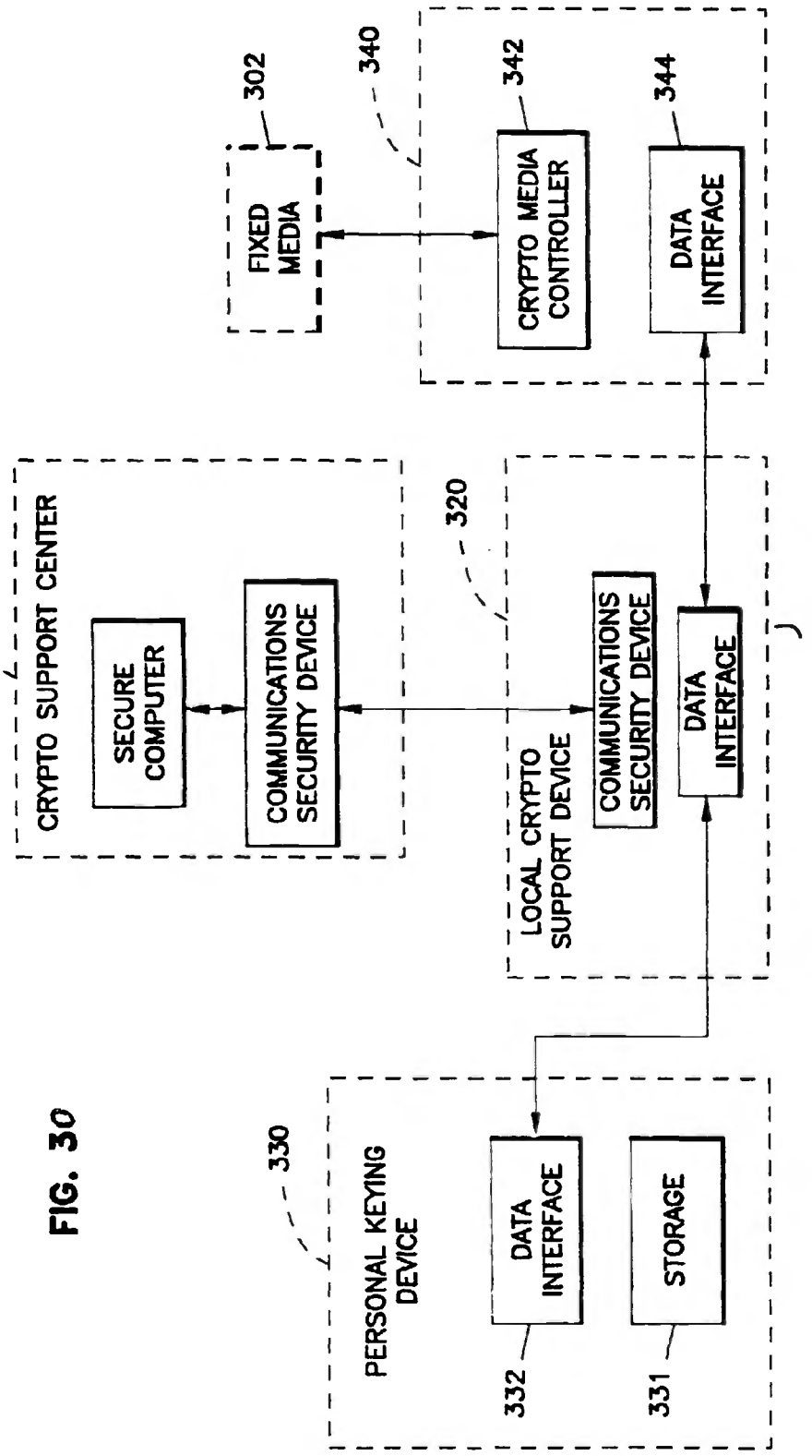

310
CRYPTO SUPPORT CENTER
SECURE COMPUTER
COMMUNICATIONS SECURITY DEVICE
320
LOCAL CRYPTO SUPPORT DEVICE
COMMUNICATIONS SECURITY DEVICE
DATA INTERFACE
340
342
CRYPTO MEDIA CONTROLLER
344
DATA INTERFACE
302
FIXED MEDIA
330
PERSONAL KEYING DEVICE
DATA INTERFACE
332
STORAGE
331

FIG. 30

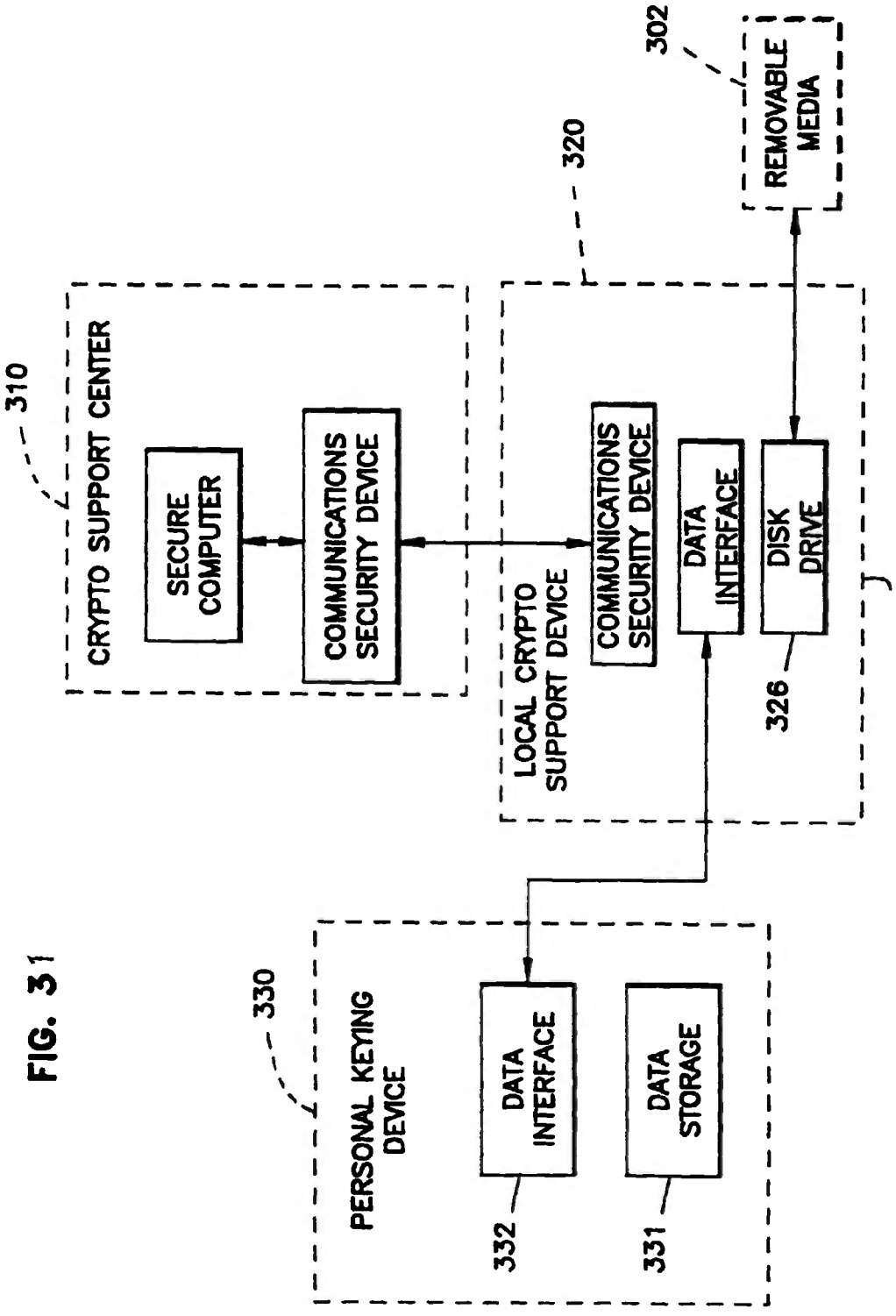

310
CRYPTO SUPPORT CENTER
SECURE COMPUTER
COMMUNICATIONS SECURITY DEVICE
320
LOCAL CRYPTO SUPPORT DEVICE
COMMUNICATIONS SECURITY DEVICE
DATA INTERFACE
DISK DRIVE
326
302
REMOVABLE MEDIA
330
PERSONAL KEYING DEVICE
DATA INTERFACE
332
DATA STORAGE
331

FIG. 31

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 March 2001 (29.03.2001)

PCT

(10) International Publication Number
WO 01/22201 A1

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number: PCT/US00/25748

(22) International Filing Date:
20 September 2000 (20.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/154,734 20 September 1999 (20.09.1999) US
60/200,396 27 April 2000 (27.04.2000) US

(71) Applicant: **ETHENTICA, INC.** [US/US]; 100 North Pointe Drive, Lake Forest, CA 92630 (US).

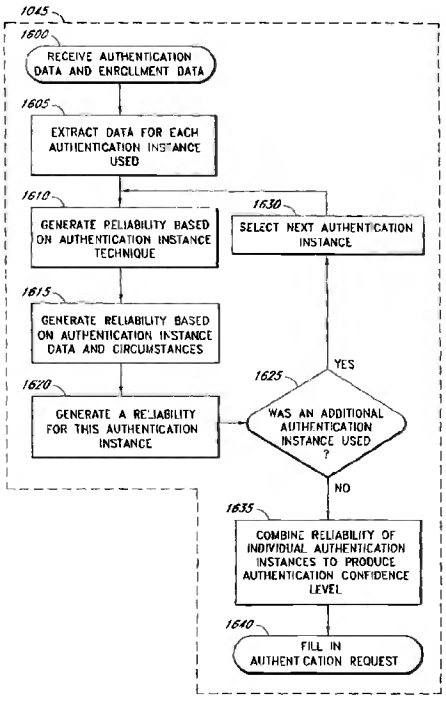
(72) Inventors: **DICKINSON, Alexander, G.**; 466 Monterey Drive, Laguna Beach, CA 92651 (US). **BERGER, Brian**; 23518 Sandstone, Mission Viejo, CA 92692 (US). **DOBSON, Robert, T., Jr.**; 15 Inverary, Dove Canyon, CA 92679 (US).

(74) Agent: **ALTMAN, Daniel, E.**; Knobbe, Martens, Olson & Bear, LLP, 620 Newport Center Drive, 16th Floor, Newport Beach, CA 92660 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, CZ (utility model), DE, DE (utility model), DK, DK (utility model), DM, DZ, EE, EE (utility model), ES, FI, FI (utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KR (utility model), KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (utility model), SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

[Continued on next page]

(54) Title: CONTEXT SENSITIVE DYNAMIC AUTHENTICATION IN A CRYPTOGRAPHIC SYSTEM



(57) Abstract: A system for performing authentication of a first user to a second user includes the ability for the first user to submit multiple instances of authentication data which are evaluated and then used to generate an overall level of confidence in the claimed identity of the first user. The individual authentication instances are evaluated based upon: the degree of match between the user provided by the first user during the authentication and the data provided by the first user during his enrollment; the inherent reliability of the authentication technique being used; the circumstances surrounding the generation of the authentication

[Continued on next page]

WO 01/22201 A1

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— *With international search report.*

— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

data by the first user; and the circumstances surrounding the generation of the enrollment data by the first user. This confidence level is compared with a required trust level which is based at least in part upon the requirements of the second user, and the authentication result is based upon this comparison.

CONTEXT SENSITIVE DYNAMIC AUTHENTICATION IN A CRYPTOGRAPHIC SYSTEM

Background of the Invention

Field of the Invention

The present invention relates to cryptographic authentication. More specifically, the present invention relates to the evaluation of an authentication event based upon the context of that event.

Description of the Related Art

With the continued increase in commercial and other transactions taking place across linked computer systems, it has become desirable to secure these transactions and the information related to these transactions. One form of security is to prevent access to systems which perform certain functions, for instance by requiring a password or PIN number in order to use an ATM. Another form of security is to protect data from being intercepted and used by those other than the intended recipients, for instance, when sending a credit card number electronically. Another form of security involves allowing someone to undeniably sign a document or otherwise assent to a transaction electronically.

All of these functions are related to the concept of authentication, or proof of identity. Authentication of electronic systems, particularly digital systems, is generally carried out using cryptographic techniques and protocols. Cryptography is the scrambling of information in such a specific way that it can only be unscrambled by someone who holds the appropriate unscrambling key. By exchanging messages which can only be decrypted by those with access to the proper key, cryptographic protocols can be used as a means to authenticate individuals.

Techniques of authentication vary in their ease of use and their reliability. For instance, a classical authentication technique in both electronic and non-electronic systems is the password. Anyone who knows the password for the club is assumed to be a member and is admitted. Anyone who doesn't know the password is prohibited. Similarly, someone who knows a particular individual's password is assumed to be that individual, while someone who doesn't is assumed to be someone else.

This technique is fairly simple to implement, and fairly simple to use. The individuals being authenticated need merely remember the password, and they can be authenticated by anyone else who knows the password. However, such a technique is also fairly unreliable; people may be forget their own password or overhear someone else's password.

Other techniques for authentication involve the use or control of a particular token, such as a particular key, either physical or electronic. Still other techniques are based upon some immutable physical characteristic of a user, such as a fingerprint or the sound of their voice. Some of these techniques are more reliable than others. For instance, fingerprints are more effective authenticators than passwords in most cases. However, analyzing a fingerprint to determine if someone is who they claim to be is much more complex than simply comparing passwords. Generally, the more reliable a technique of authentication is, the more cumbersome it is to use.

Therefore, there is a continued need for improved systems that provide appropriate levels of reliable authentication and security with the improved ease of use and reduced inconvenience to the users.

Summary of the Invention

Based on the foregoing, a need exists to provide a method for authenticating a user here the user is authenticated based on those aspects of his current circumstances which are most reliable for use in determining that the user is who he says he is. Accordingly, one aspect of the invention is to provide a secure server, or trust engine, having server-centric keys, or in other words, storing cryptographic keys and user authentication data on a server. According to this embodiment, a user accesses the trust engine in order to perform authentication and cryptographic functions, such as, for example, authentication, authorization, digital signing and generation, storage, and retrieval of certificates, encryption, notary-like and power-of-attorney-like actions, and the like.

Another aspect of the invention is to provide a reliable, or trusted, authentication process. Moreover, subsequent to a trustworthy positive authentication, a wide number of differing actions may be taken, from providing cryptographic technology, to system or device authorization and access, to permitting use or control of a wide number of electronic devices.

Another aspect of the invention is to provide cryptographic keys and authentication data in an environment where they are not lost, stolen, or compromised, thereby advantageously avoiding a need to continually reissue and manage new keys and authentication data. According to another aspect of the invention, the trust engine allows a user to use one key pair for multiple activities, vendors, and/or authentication requests. According to yet another aspect of the invention, the trust engine performs the majority of cryptographic processing, such as encrypting, authenticating, or signing, on the server side, thereby allowing clients to possess only minimal computing resources.

According to yet another aspect of the invention, the trust engine includes multiple depositories for storing portions of each cryptographic key and authentication data. The portions are created through a data splitting process that prohibits reconstruction without a predetermined portion from more than one depository. According to another embodiment, the multiple depositories are geographically remote such that a rogue employee or otherwise compromised system at one depository will not provide access to a user's key or authentication data.

According to yet another aspect, the authentication process advantageously allows the trust engine to process vendor and user authentication activities in parallel. According to yet another embodiment, the trust engine may advantageously track failed access attempts and thereby limit the number of times malicious intruders may attempt to subvert the system.

According to yet another aspect, the trust engine may include multiple instantiations where each trust engine may predict and share processing loads with the others. According to yet another embodiment, the trust engine may include a redundancy module for polling a plurality of authentication results to ensure that more than one system authenticates the user.

Therefore, one embodiment of the invention provides a method where data is obtained from the user during an authentication attempt along with data describing one or more aspects of the current circumstances surrounding the authentication attempt. This data is compared to previously received data associated with the circumstances of

the previous authentication attempts by this user. Based upon this comparison, a level of trust for the authentication attempt of the user.

According to another aspect of the invention, a method for authenticating a user is provided in which authentication data is obtained associated with an authentication operation, and metadata is obtained related to the authentication operation. This metadata is compared with data which has previously been received and a level of trust associated with the authentication operation is determined.

According to yet another aspect of the invention, the authentication data is comprised of data generated using more than one authentication technique. Data generated using different techniques is compared with different portions of the set of previously stored data in order to determine a level of trust for the authentication.

An additional aspect of the invention provides a system for graded authentication comprising user data and circumstantial data used by a trust engine. The user data is obtained from a user during previously successful authentication attempts, and circumstantial data associated with these previously successful authentication attempts is also received. The trust engine generates a level of trust associated with a current authentication attempt by comparing circumstantial data associated with the current authentication attempt with the circumstantial data associated with the previously successful authentication attempts.

In another embodiment of the invention, a method for grading an authentication operation is relying on a variable set of authentication techniques used to obtain authentication data is provided. The reliability of the set of authentication techniques available is defined, and authentication data is received during an authentication operation. The authentication data is generated using a subset of the available authentication techniques. The acceptability of the authentication data is determined for the subset of data generated by each of the techniques used, and the level of trust is defined for the authentication operation based upon the acceptability the authentication data and the reliability of the authentication techniques used to generated the authentication data.

Yet another aspect of the invention provides an apparatus for evaluating an authentication attempt. The apparatus comprises reliability data which is associated with a set of authentication techniques that may be used in an authentication attempt. It also comprises a plurality of authentication instances generated using a subset of the authentication techniques. A trust engine determines a level of match associated with each authentication instance. These levels of match are used to assign a level of trust for the authentication attempt. The level of trust is determined by the level of match associated with each authentication instance and the reliability of the technique used in each authentication instance.

According to yet another aspect of the invention, a method for grading an authentication attempt is provided in which the reliability of a set of authentication techniques that may be used in an authentication attempt is defined. A plurality of authentication instances generated using a subset of these authentication techniques is then received and a level of match is associated with each authentication instance. The level of trust of the authentication attempt is then defined based upon the level of match associated with each authentication instance and upon the reliability of the technique used in each authentication instance.

Brief Description of the Drawings

The present invention is described in more detail below in connection with the attached drawings, which are meant to illustrate and not to limit the invention, and in which:

FIGURE 1 illustrates a block diagram of a cryptographic system, according to aspects of an embodiment of the invention;

FIGURE 2 illustrates a block diagram of the trust engine of FIGURE 1, according to aspects of an embodiment of the invention;

FIGURE 3 illustrates a block diagram of the transaction engine of FIGURE 2, according to aspects of an embodiment of the invention;

FIGURE 4 illustrates a block diagram of the depository of FIGURE 2, according to aspects of an embodiment of the invention;

FIGURE 5 illustrates a block diagram of the authentication engine of FIGURE 2, according to aspects of an embodiment of the invention;

FIGURE 6 illustrates a block diagram of the cryptographic engine of FIGURE 2, according to aspects of an embodiment of the invention;

FIGURE 7 illustrates a block diagram of a depository system, according to aspects of another embodiment of the invention;

FIGURE 8 illustrates a flow chart of a data splitting process according to aspects of an embodiment of the invention;

FIGURE 9A illustrates a data flow of an enrollment process according to aspects of an embodiment of the invention;

FIGURE 9B illustrates a flow chart of an interoperability process according to aspects of an embodiment of the invention;

FIGURE 10 illustrates a data flow of an authentication process according to aspects of an embodiment of the invention;

FIGURE 11 illustrates a data flow of a signing process according to aspects of an embodiment of the invention.

FIGURE 12 illustrates a data flow and an encryption/decryption process according to aspects and yet another embodiment of the invention;

FIGURE 13 illustrates a simplified block diagram of a trust engine system according to aspects of another embodiment of the invention;

FIGURE 14 illustrates a simplified block diagram of a trust engine system according to aspects of another embodiment of the invention;

FIGURE 15 illustrates a block diagram of the redundancy module of FIGURE 14, according to aspects of an embodiment of the invention;

FIGURE 16 illustrates a process for evaluating authentications according to one aspect of the invention;

FIGURE 17 illustrates a process for assigning a value to an authentication according to one aspect as shown in FIGURE 16 of the invention;

FIGURE 18 illustrates a process for performing trust arbitrage in an aspect of the invention as shown in FIGURE 17; and

FIGURE 19 illustrates a sample transaction between a user and a vendor according to aspects of an embodiment of the invention where an initial web based contact leads to a sales contract signed by both parties.

FIGURE 20 illustrates a sample user system with a cryptographic service provider module which provides security functions to a user system.

Detailed Description of the Preferred Embodiment

One aspect of the present invention is to provide a cryptographic system where one or more secure servers, or a trust engine, stores cryptographic keys and user authentication data. Users access the functionality of conventional cryptographic systems through network access to the trust engine, however, the trust engine does not release actual keys and other authentication data and therefore, the keys and data remain secure. This server-centric storage of keys and authentication data provides for user-independent security, portability, availability, and straightforwardness.

Because users can be confident in, or trust, the cryptographic system to perform user and document authentication and other cryptographic functions, a wide variety of functionality may be incorporated into the system. For example, the trust engine provider can ensure against agreement repudiation by, for example, authenticating the agreement participants, digitally signing the agreement on behalf of or for the participants, and storing a record of the agreement digitally signed by each participant. In addition, the cryptographic system may monitor agreements and determine to apply varying degrees of authentication, based on, for example, price, user, vendor, geographic location, place of use, or the like.

To facilitate a complete understanding of the invention, the remainder of the detailed description describes the invention with reference to the figures, wherein like elements are referenced with like numerals throughout.

FIGURE 1 illustrates a block diagram of a cryptographic system 100, according to aspects of an embodiment of the invention. As shown in FIGURE 1, the cryptographic system 100 includes a user system 105, a trust engine 110, a certificate authority 115, and a vendor system 120, communicating through a communication link 125.

According to one embodiment of the invention, the user system 105 comprises a conventional general-purpose computer having one or more microprocessors, such as, for example, an Intel-based processor. Moreover, the user system 105 includes an appropriate operating system, such as, for example, an operating system capable of including graphics or windows, such as Windows, Unix, Linux, or the like. As shown in FIGURE 1, the user system 105 may include a biometric device 107. The biometric device 107 may advantageously capture a user's biometric and transfer the captured biometric to the trust engine 110. According to one embodiment of the invention, the biometric device may advantageously comprise a device having attributes and features similar to those disclosed in U.S. Patent

Application No. 08/926,277, filed on September 5, 1997, entitled *"RELIEF OBJECT IMAGE GENERATOR,"* U.S. Patent Application No. 09/558,634, filed on April 26, 2000, entitled *"IMAGING DEVICE FOR A RELIEF OBJECT AND SYSTEM AND METHOD OF USING THE IMAGE DEVICE,"* U.S. Patent Application No. 09/435,011, filed on November 5, 1999, entitled *"RELIEF OBJECT SENSOR ADAPTOR,"* and U.S. Patent Application No. 09/477,943, filed on January 5, 2000, entitled *"PLANAR OPTICAL IMAGE SENSOR AND SYSTEM FOR GENERATING AN ELECTRONIC IMAGE OF A RELIEF OBJECT FOR FINGERPRINT READING,"* all of which are owned by the instant assignee, and all of which are hereby incorporated by reference herein.

In addition, the user system 105 may connect to the communication link 125 through a conventional service provider, such as, for example, a dial up, digital subscriber line (DSL), cable modem, fiber connection, or the like. According to another embodiment, the user system 105 connects the communication link 125 through network connectivity such as, for example, a local or wide area network. According to one embodiment, the operating system includes a TCP/IP stack that handles all incoming and outgoing message traffic passed over the communication link 125.

Although the user system 105 is disclosed with reference to the foregoing embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives embodiments of the user system 105, including almost any computing device capable of sending or receiving information from another computer system. For example, the user system 105 may include a computer workstation, an interactive television, an interactive kiosk, a personal mobile computing device, such as a digital assistant, mobile phone, laptop, or the like, a wireless communications device, a smartcard, an embedded computing device, or the like, which can interact with the communication link 125. In such alternative systems, the operating systems will likely differ and be adapted for the particular device. However, according to one embodiment, the operating systems advantageously continue to provide the appropriate communications protocols needed to establish communication with the communication link 125.

FIGURE 1 illustrates the trust engine 110. According to one embodiment, the trust engine 110 comprises one or more secure servers for accessing and storing sensitive information, such as user authentication data and public and private cryptographic keys. According to one embodiment, the authentication data includes data designed to uniquely identify a user of the cryptographic system 100. For example, the authentication data may include a user identification number, one or more biometrics, and a series of questions and answers generated by the trust engine 110 or the user, but answered initially by the user at enrollment. The foregoing questions may include demographic data, such as place of birth, address, anniversary, or the like, personal data, such as mother's maiden name, favorite ice cream, or the like, or other data designed to uniquely identify the user. The trust engine 110 compares a user's authentication data associated with a current transaction, to the authentication data provided at an earlier time, such as, for example, during enrollment. The trust engine 110 may advantageously require the user to produce the authentication data at the time of each transaction, or, the trust engine 110 may advantageously allow the user to

periodically produce authentication data, such as at the beginning of a string of transactions or the logging onto a particular vendor website.

According to the embodiment where the user produces biometric data, the user provides a physical characteristic, such as a fingerprint or speech, to the biometric device 107. The biometric device advantageously produces an electronic pattern, or biometric, of the physical characteristic. The electronic pattern is transferred through the user system 105 to the trust engine 110 for either enrollment or authentication purposes.

Once the user produces the appropriate authentication data and the trust engine 110 determines a positive match between that authentication data (current authentication data) and the authentication data provided at the time of enrollment (enrollment authentication data), the trust engine 110 provides the user with complete cryptographic functionality. For example, the properly authenticated user may advantageously employ the trust engine 110 to perform hashing, digitally signing, encrypting and decrypting (often referred to only as encrypting), creating or distributing digital certificates, and the like. However, the private cryptographic keys used in the cryptographic functions will not be available outside the trust engine 110, thereby ensuring the integrity of the cryptographic keys.

According to one embodiment, the trust engine 110 generates and stores cryptographic keys. According to another embodiment, at least one cryptographic key is associated with each user. Moreover, when the cryptographic keys include public-key technology, each private key associated with a user is generated within, and not released from, the trust engine 110. Thus, so long as the user has access to the trust engine 110, the user may perform cryptographic functions using his or her private or public key. Such remote access advantageously allows users to remain completely mobile and access cryptographic functionality through practically any Internet connection, such as cellular and satellite phones, kiosks, laptops, hotel rooms and the like.

According to another embodiment, the trust engine 110 performs the cryptographic functionality using a key pair generated for the trust engine 110. According to this embodiment, the trust engine 110 first authenticates the user, and after the user has properly produced authentication data matching the enrollment authentication data, the trust engine 110 uses its own cryptographic key pair to perform cryptographic functions on behalf of the authenticated user.

A skilled artisan will recognize from the disclosure herein that the cryptographic keys may advantageously include some or all of symmetric keys, public keys, and private keys. In addition, a skilled artisan will recognize from the disclosure herein that the foregoing keys may be implemented with a wide number of algorithms available from commercial technologies, such as, for example, RSA, ELGAMAL, or the like.

FIGURE 1 also illustrates the certificate authority 115. According to one embodiment, the certificate authority 115 may advantageously comprise a trusted third-party organization or company that issues digital certificates, such as, for example, VeriSign, Baltimore, Entrust, or the like. The trust engine 110 may advantageously transmit requests for digital certificates, through one or more conventional digital certificate protocols, such as, for example, PKCS10, to the certificate authority 115. In response, the certificate authority 115 will issue a digital certificate in one or more of a number of differing protocols, such as, for example, PKCS7. According to one

embodiment of the invention, the trust engine 110 requests digital certificates from several or all of the prominent certificate authorities 115 such that the trust engine 110 has access to a digital certificate corresponding to the certificate standard of any requesting party.

According to another embodiment, the trust engine 110 internally performs certificate issuances. In this embodiment, the trust engine 110 may access a certificate system for generating certificates and/or may internally generate certificates when they are requested, such as, for example, at the time of key generation or in the certificate standard requested at the time of the request. The trust engine 110 will be disclosed in greater detail below.

FIGURE 1 also illustrates the vendor system 120. According to one embodiment, the vendor system 120 advantageously comprises a Web server. Typical Web servers generally serve content over the Internet using one of several internet markup languages or document format standards, such as the Hyper-Text Markup Language (HTML) or the Extensible Markup Language (XML). The Web server accepts requests from browsers like Netscape and Internet Explorer and then returns the appropriate electronic documents. A number of server or client-side technologies can be used to increase the power of the Web server beyond its ability to deliver standard electronic documents. For example, these technologies include Common Gateway Interface (CGI) scripts, Secure Sockets Layer (SSL) security, and Active Server Pages (ASPs). The vendor system 120 may advantageously provide electronic content relating to commercial, personal, educational, or other transactions.

Although the vendor system 120 is disclosed with reference to the foregoing embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein that the vendor system 120 may advantageously comprise any of the devices described with reference to the user system 105 or combination thereof.

FIGURE 1 also illustrates the communication link 125 connecting the user system 105, the trust engine 110, the certificate authority 115, and the vendor system 120. According to one embodiment, the communication link 125 preferably comprises the Internet. The Internet, as used throughout this disclosure is a global network of computers. The structure of the Internet, which is well known to those of ordinary skill in the art, includes a network backbone with networks branching from the backbone. These branches, in turn, have networks branching from them, and so on. Routers move information packets between network levels, and then from network to network, until the packet reaches the neighborhood of its destination. From the destination, the destination network's host directs the information packet to the appropriate terminal, or node. In one advantageous embodiment, the Internet routing hubs comprise domain name system (DNS) servers using Transmission Control Protocol/Internet Protocol (TCP/IP) as is well known in the art. The routing hubs connect to one or more other routing hubs via high-speed communication links.

One popular part of the Internet is the World Wide Web. The World Wide Web contains different computers, which store documents capable of displaying graphical and textual information. The computers that provide information on the World Wide Web are typically called "websites." A website is defined by an Internet address that has an associated electronic page. The electronic page can be identified by a Uniform Resource Locator (URL).

Generally, an electronic page is a document that organizes the presentation of text, graphical images, audio, video, and so forth.

Although the communication link 125 is disclosed in terms of its preferred embodiment, one of ordinary skill in the art will recognize from the disclosure herein that the communication link 125 may include a wide range of interactive communications links. For example, the communication link 125 may include interactive television networks, telephone networks, wireless data transmission systems, two-way cable systems, customized private or public computer networks, interactive kiosk networks, automatic teller machine networks, direct links, satellite or cellular networks, and the like.

FIGURE 2 illustrates a block diagram of the trust engine 110 of FIGURE 1 according to aspects of an embodiment of the invention. As shown in FIGURE 2, the trust engine 110 includes a transaction engine 205, a depository 210, an authentication engine 215, and a cryptographic engine 220. According to one embodiment of the invention, the trust engine 110 also includes mass storage 225. As further shown in FIGURE 2, the transaction engine 205 communicates with the depository 210, the authentication engine 215, and the cryptographic engine 220, along with the mass storage 225. In addition, the depository 210 communicates with the authentication engine 215, the cryptographic engine 220, and the mass storage 225. Moreover, the authentication engine 215 communicates with the cryptographic engine 220. According to one embodiment of the invention, some or all of the foregoing communications may advantageously comprise the transmission of XML documents to IP addresses that correspond to the receiving device. As mentioned in the foregoing, XML documents advantageously allow designers to create their own customized document tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations. Moreover, some or all of the foregoing communications may include conventional SSL technologies.

According to one embodiment, the transaction engine 205 comprises a data routing device, such as a conventional Web server available from Netscape, Microsoft, Apache, or the like. For example, the Web server may advantageously receive incoming data from the communication link 125. According to one embodiment of the invention, the incoming data is addressed to a front-end security system for the trust engine 110. For example, the front-end security system may advantageously include a firewall, an intrusion detection system searching for known attack profiles, and/or a virus scanner. After clearing the front-end security system, the data is received by the transaction engine 205 and routed to one of the depository 210, the authentication engine 215, the cryptographic engine 220, and the mass storage 225. In addition, the transaction engine 205 monitors incoming data from the authentication engine 215 and cryptographic engine 220, and routes the data to particular systems through the communication link 125. For example, the transaction engine 205 may advantageously route data to the user system 105, the certificate authority 115, or the vendor system 120.

According to one embodiment, the data is routed using conventional HTTP routing techniques, such as, for example, employing URLs or Uniform Resource Indicators (URIs). URIs are similar to URLs, however, URIs typically indicate the source of files or actions, such as, for example, executables, scripts, and the like. Therefore, according to

the one embodiment, the user system 105, the certificate authority 115, the vendor system 120, and the components of the trust engine 210, advantageously include sufficient data within communication URLs or URIs for the transaction engine 205 to properly route data throughout the cryptographic system.

Although the data routing is disclosed with reference to its preferred embodiment, a skilled artisan will recognize a wide number of possible data routing solutions or strategies. For example, XML or other data packets may advantageously be unpacked and recognized by their format, content, or the like, such that the transaction engine 205 may properly route data throughout the trust engine 110. Moreover, a skilled artisan will recognize that the data routing may advantageously be adapted to the data transfer protocols conforming to particular network systems, such as, for example, when the communication link 125 comprises a local network.

According to yet another embodiment of the invention, the transaction engine 205 includes conventional SSL encryption technologies, such that the foregoing systems may authenticate themselves, and vise-versa, with transaction engine 205, during particular communications. As will be used throughout this disclosure, the term "½ SSL" refers to communications where a server but not necessarily the client, is SSL authenticated, and the term "FULL SSL" refers to communications where the client and the server are SSL authenticated. When the instant disclosure uses the term "SSL", the communication may comprise ½ or FULL SSL.

As the transaction engine 205 routes data to the various components of the cryptographic system 100, the transaction engine 205 may advantageously create an audit trail. According to one embodiment, the audit trail includes a record of at least the type and format of data routed by the transaction engine 205 throughout the cryptographic system 100. Such audit data may advantageously be stored in the mass storage 225.

FIGURE 2 also illustrates the depository 210. According to one embodiment, the depository 210 comprises one or more data storage facilities, such as, for example, a directory server, a database server, or the like. As shown in FIGURE 2, the depository 210 stores cryptographic keys and enrollment authentication data. The cryptographic keys may advantageously correspond to the trust engine 110 or to users of the cryptographic system 100, such as the user or vendor. The enrollment authentication data may advantageously include data designed to uniquely identify a user, such as, user ID, passwords, answers to questions, biometric data, or the like. This enrollment authentication data may advantageously be acquired at enrollment of a user or another alternative later time. For example, the trust engine 110 may include periodic or other renewal or reissue of enrollment authentication data.

According to one embodiment, the communication from the transaction engine 205 to and from the authentication engine 215 and the cryptographic engine 220 comprises secure communication, such as, for example conventional SSL technology. In addition, as mentioned in the foregoing, the data of the communications to and from the depository 210 may be transferred using URLs, URIs, HTTP or XML documents, with any of the foregoing advantageously having data requests and formats embedded therein.

As mentioned above, the depository 210 may advantageously comprises a plurality of secure data storage facilities. In such an embodiment, the secure data storage facilities may be configured such that a compromise of the security in one individual data storage facility will not compromise the cryptographic keys or the authentication data

stored therein. For example, according to this embodiment, the cryptographic keys and the authentication data are mathematically operated on so as to statistically and substantially randomize the data stored in each data storage facility. According to one embodiment, the randomization of the data of an individual data storage facility renders that data undecipherable. Thus, compromise of an individual data storage facility produces only a randomized undecipherable number and does not compromise the security of any cryptographic keys or the authentication data as a whole.

FIGURE 2 also illustrates the trust engine 110 including the authentication engine 215. According to one embodiment, the authentication engine 215 comprises a data comparator configured to compare data from the transaction engine 205 with data from the depository 210. For example, during authentication, a user supplies current authentication data to the trust engine 110 such that the transaction engine 205 receives the current authentication data. As mentioned in the foregoing, the transaction engine 205 recognizes the data requests, preferably in the URL or URI, and routes the authentication data to the authentication engine 215. Moreover, upon request, the depository 210 forwards enrollment authentication data corresponding to the user to the authentication engine 215. Thus, the authentication engine 215 has both the current authentication data and the enrollment authentication data for comparison.

According to one embodiment, the communications to the authentication engine comprise secure communications, such as, for example, SSL technology. Additionally, security can be provided within the trust engine 110 components, such as, for example, super-encryption using public key technologies. For example, according to one embodiment, the user encrypts the current authentication data with the public key of the authentication engine 215. In addition, the depository 210 also encrypts the enrollment authentication data with the public key of the authentication engine 215. In this way, only the authentication engine's private key can be used to decrypt the transmissions.

As shown in FIGURE 2, the trust engine 110 also includes the cryptographic engine 220. According to one embodiment, the cryptographic engine comprises a cryptographic handling module, configured to advantageously provide conventional cryptographic functions, such as, for example, public-key infrastructure (PKI) functionality. For example, the cryptographic engine 220 may advantageously issue public and private keys for users of the cryptographic system 100. In this manner, the cryptographic keys are generated at the cryptographic engine 220 and forwarded to the depository 210 such that at least the private cryptographic keys are not available outside of the trust engine 110. According to another embodiment, the cryptographic engine 220 randomizes and splits at least the private cryptographic key data, thereby storing only the randomized split data. Similar to the splitting of the enrollment authentication data, the splitting process ensures the stored keys are not available outside the cryptographic engine 220. According to another embodiment, the functions of the cryptographic engine can be combined with and performed by the authentication engine 215.

According to one embodiment, communications to and from the cryptographic engine include secure communications, such as SSL technology. In addition, XML documents may advantageously be employed to transfer data and/or make cryptographic function requests.

FIGURE 2 also illustrates the trust engine 110 having the mass storage 225. As mentioned in the foregoing, the transaction engine 205 keeps data corresponding to an audit trail and stores such data in the mass storage 225. Similarly, according to one embodiment of the invention, the depository 210 keeps data corresponding to an audit trail and stores such data in the mass storage device 225. The depository audit trail data is similar to that of the transaction engine 205 in that the audit trail data comprises a record of the requests received by the depository 210 and the response thereof. In addition, the mass storage 225 may be used to store digital certificates having the public key of a user contained therein.

Although the trust engine 110 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize in the disclosure herein, a wide number of alternatives for the trust engine 110. For example, the trust engine 110, may advantageously perform only authentication, or alternatively, only some or all of the cryptographic functions, such as data encryption and decryption. According to such embodiments, one of the authentication engine 215 and the cryptographic engine 220 may advantageously be removed, thereby creating a more straightforward design for the trust engine 110. In addition, the cryptographic engine 220 may also communicate with a certificate authority such that the certificate authority is embodied within the trust engine 110. According to yet another embodiment, the trust engine 110 may advantageously perform authentication and one or more cryptographic functions, such as, for example, digital signing.

FIGURE 3 illustrates a block diagram of the transaction engine 205 of FIGURE 2, according to aspects of an embodiment of the invention. According to this embodiment, the transaction engine 205 comprises an operating system 305 having a handling thread and a listening thread. The operating system 305 may advantageously be similar to those found in conventional high volume servers, such as, for example, Web servers available from Apache. The listening thread monitors the incoming communication from one of the communication link 125, the authentication engine 215, and the cryptographic engine 220 for incoming data flow. The handling thread recognizes particular data structures of the incoming data flow, such as, for example, the foregoing data structures, thereby routing the incoming data to one of the communication link 125, the depository 210, the authentication engine 215, the cryptographic engine 220, or the mass storage 225. As shown in FIGURE 3, the incoming and outgoing data may advantageously be secured through, for example, SSL technology.

FIGURE 4 illustrates a block diagram of the depository 210 of FIGURE 2 according to aspects of an embodiment of the invention. According to this embodiment, the depository 210 comprises one or more lightweight directory access protocol (LDAP) servers. LDAP directory servers are available from a wide variety of manufacturers such as Netscape, ISO, and others. FIGURE 4 also shows that the directory server preferably stores data 405 corresponding to the cryptographic keys and data 410 corresponding to the enrollment authentication data. According to one embodiment, the depository 210 comprises a single logical memory structure indexing authentication data and

cryptographic key data to a unique user ID. The single logical memory structure preferably includes mechanisms to ensure a high degree of trust, or security, in the data stored therein. For example, the physical location of the depository 210 may advantageously include a wide number of conventional security measures, such as limited employee access, modern surveillance systems, and the like. In addition to, or in lieu of, the physical securities, the computer system or server may advantageously include software solutions to protect the stored data. For example, the depository 210 may advantageously create and store data 415 corresponding to an audit trail of actions taken. In addition, the incoming and outgoing communications may advantageously be encrypted with public key encryption coupled with conventional SSL technologies.

According to another embodiment, the depository 210 may comprise distinct and physically separated data storage facilities, as disclosed further with reference to FIGURE 7.

FIGURE 5 illustrates a block diagram of the authentication engine 215 of FIGURE 2 according to aspects of an embodiment of the invention. Similar to the transaction engine 205 of FIGURE 3, the authentication engine 215 comprises an operating system 505 having at least a listening and a handling thread of a modified version of a conventional Web server, such as, for example, Web servers available from Apache. As shown in FIGURE 5, the authentication engine 215 includes access to at least one private key 510. The private key 510 may advantageously be used for example, to decrypt data from the transaction engine 205 or the depository 210, which was encrypted with a corresponding public key of the authentication engine 215.

FIGURE 5 also illustrates the authentication engine 215 comprising a comparator 515, a data splitting module 520, and a data assembling module 525. According to the preferred embodiment of the invention, the comparator 515 includes technology capable of comparing potentially complex patterns related to the foregoing biometric authentication data. The technology may include hardware, software, or combined solutions for pattern comparisons, such as, for example, those representing finger print patterns or voice patterns. In addition, according to one embodiment, the comparator 515 of the authentication engine 215 may advantageously compare conventional hashes of documents in order to render a comparison result. According to one embodiment of the invention, the comparator 515 includes the application of heuristics 530 to the comparison. The heuristics 530 may advantageously address circumstances surrounding an authentication attempt, such as, for example, the time of day, IP address or subnet mask, purchasing profile, email address, processor serial number or ID, or the like.

Moreover, the nature of biometric data comparisons may result in varying degrees of confidence being produced from the matching of current biometric authentication data to enrollment data. For example, unlike a traditional password which may only return a positive or negative match, a fingerprint may be determined to be a partial match, *e.g.* a 90% match, a 75% match, or a 10% match, rather than simply being correct or incorrect. Other biometric identifiers such as voice print analysis or face recognition may share this property of probabilistic authentication, rather than absolute authentication.

When working with such probabilistic authentication or in other cases where an authentication is considered less than absolutely reliable, it is desirable to apply the heuristics 530 to determine whether the level of confidence in the authentication provided is sufficiently high to authenticate the transaction which is being made.

It will sometimes be the case that the transaction at issue is a relatively low value transaction where it is acceptable to be authenticated to a lower level of confidence. This could include a transaction which has a low dollar value associated with it (*e.g.,* a $10 purchase) or a transaction with low risk (*e.g.,* admission to a members-only web site).

Conversely, for authenticating other transactions, it may be desirable to require a high degree of confidence in the authentication before allowing the transaction to proceed. Such transactions may include transactions of large dollar value (*e.g.,* signing a multi-million dollar supply contract) or transaction with a high risk if an improper authentication occurs (*e.g.,* remotely logging onto a government computer).

The use of the heuristics 530 in combination with confidence levels and transactions values may be used as will be described below to allow the comparator to provide a dynamic context-sensitive authentication system.

According to another embodiment of the invention, the comparator 515 may advantageously track authentication attempts for a particular transaction. For example, when a transaction fails, the trust engine 110 may request the user to re-enter his or her current authentication data. The comparator 515 of the authentication engine 215 may advantageously employ an attempt limiter 535 to limit the number of authentication attempts, thereby prohibiting brute-force attempts to impersonate a user's authentication data. According to one embodiment, the attempt limiter 535 comprises a software module monitoring transactions for repeating authentication attempts and, for example, limiting the authentication attempts for a given transaction to three. Thus, the attempt limiter 535 will limit an automated attempt to impersonate an individual's authentication data to, for example, simply three "guesses." Upon three failures, the attempt limiter 535 may advantageously deny additional authentication attempts. Such denial may advantageously be implemented through, for example, the comparator 515 returning a negative result regardless of the current authentication data being transmitted. On the other hand, the transaction engine 205 may advantageously block any additional authentication attempts pertaining to a transaction in which three attempts have previously failed.

The authentication engine 215 also includes the data splitting module 520 and the data assembling module 525. The data splitting module 520 advantageously comprises a software, hardware, or combination module having the ability to mathematically operate on various data so as to substantially randomize and split the data into portions. According to one embodiment, original data is not recreatable from an individual portion. The data assembling module 525 advantageously comprises a software, hardware, or combination module configured to mathematically operate on the foregoing substantially randomized portions, such that the combination thereof provides the original deciphered data. According to one embodiment, the authentication engine 215 employs the data splitting module 520 to randomize and split enrollment authentication data into portions, and employs the data assembling module 525 to reassemble the portions into usable enrollment authentication data.

FIGURE 6 illustrates a block diagram of the cryptographic engine 220 of the trust engine 200 of FIGURE 2 according to aspects of one embodiment of the invention. Similar to the transaction engine 205 of FIGURE 3, the cryptographic engine 220 comprises an operating system 605 having at least a listening and a handling thread of a modified version of a conventional Web server, such as, for example, Web servers available from Apache. As shown in FIGURE 6, the cryptographic engine 220 comprises a data splitting module 610 and a data assembling module 620 that function similar to those of FIGURE 5. However, according to one embodiment, the data splitting module 610 and the data assembling module 620 process cryptographic key data, as opposed to the foregoing enrollment authentication data. Although, a skilled artisan will recognize from the disclosure herein that the data splitting module 910 and the data splitting module 620 may be combined with those of the authentication engine 215.

The cryptographic engine 220 also comprises a cryptographic handling module 625 configured to perform some or all of a wide number of cryptographic functions. According to one embodiment, the cryptographic handling module 625 may comprise software modules or programs, hardware, or both. According to another embodiment, the cryptographic handling module 625 may perform data comparisons, data hashing, data encryption or decryption, digital signature verification or creation, digital certificate generation, storage, or requests, cryptographic key generation, or the like. Moreover, a skilled artisan will recognize from the disclosure herein that the cryptographic handling module 625 may advantageously comprises a public-key infrastructure, such as Pretty Good Privacy (PGP), an RSA-based public-key system, or a wide number of alternative key management systems. In addition, the cryptographic handling module 625 may perform public-key encryption, symmetric-key encryption, or both. In addition to the foregoing, the cryptographic handling module 625 may include one or more computer programs or modules, hardware, or both, for implementing seamless, transparent, interoperability functions.

A skilled artisan will also recognize from the disclosure herein that the cryptographic functionality may include a wide number of functions generally surrounding cryptographic key management systems.

FIGURE 7 illustrates a simplified block diagram of a depository system 700 according to aspects of an embodiment of the invention. As shown in FIGURE 7, the depository system 700 advantageously comprises multiple data storage facilities, for example, data storage facilities D1, D2, D3, and D4. According to one embodiment of the invention, each of the data storage facilities D1 through D4 may advantageously comprise some or all of the elements disclosed with reference to the depository 210 of FIGURE 4. Similar to the depository 210, the data storage facilities D1 through D4 communicate with the transaction engine 205, the authentication engine 215, and the cryptographic engine 220, preferably through conventional SSL communication links transferring, for example, XML documents. Communications from the transaction engine 205 may advantageously include requests for data, wherein the request is advantageously broadcast to the IP address of each data storage facility D1 through D4. On the other hand, the transaction engine 205 may broadcast requests to particular data storage facilities based on a wide number of criteria, such as, for example, response time, server loads, maintenance schedules, or the like.

In response to requests for data from the transaction engine 205, the depository system 700 advantageously forwards stored data to the authentication engine 215 and the cryptographic engine 220. The respective data

assembling modules receive the forwarded data and assemble the data into useable formats. On the other hand, communications from the authentication engine 215 and the cryptographic engine 220 to the data storage facilities D1 through D4 may include the transmission of sensitive data to be stored. For example, according to one embodiment, the authentication engine 215 and the cryptographic engine 220 may advantageously employ their respective data splitting modules to divide sensitive data into undecipherable portions, and then transmit one or more undecipherable portions of the sensitive data to a particular data storage facility.

According to one embodiment, each data storage facility, D1 through D4, comprises a separate and independent storage system, such as, for example, a directory server. According to another embodiment of the invention, the depository system 700 comprises multiple geographically separated independent data storage systems. By distributing the sensitive data into distinct and independent storage facilities D1 through D4, some or all of which may be advantageously geographically separated, the depository system 700 provides redundancy along with additional security measures. For example, according to one embodiment, only data from two of the multiple data storage facilities, D1 through D4, are needed to decipher and reassemble the sensitive data. Thus, as many as two of the four data storage facilities D1 through D4 may be inoperative due to maintenance, system failure, power failure, or the like, without affecting the functionality of the trust engine 110. In addition, because, according to one embodiment, the data stored in each data storage facility is randomized and undecipherable, compromise of any individual data storage facility does not necessarily compromise the sensitive data. Moreover, in the embodiment having geographically separation of the data storage facilities, a compromise of multiple geographically remote facilities becomes increasingly difficult. In fact, even a rouge employee will be greatly challenged to subvert the needed multiple independent geographically remote data storage facilities.

Although the depository system 700 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the depository system 700. For example, the depository system 700 may comprise two or more data storage facilities. In addition, sensitive data may be mathematically operated such that portions from two or more data storage facilities are needed to reassemble and decipher the sensitive data.

As mentioned in the foregoing, the authentication engine 215 and the cryptographic engine 220 each include a data splitting module 520 and 610, respectively, for splitting sensitive data, such as, for example, the authentication data and the cryptographic key data. FIGURE 8 illustrates a flowchart of a data splitting process 800 performed by the data splitting module according to aspects of an embodiment of the invention. As shown in FIGURE 8, the data splitting process 800 begins at STEP 805 when sensitive data "S" is received by the data splitting module of the authentication engine 215 or the cryptographic engine 220. Preferably, in STEP 810, the data splitting module then generates a substantially random number, value, or string or set of bits, "A." For example, the random number A may be generated in a wide number of varying conventional techniques available to one of ordinary skill in the art, for producing high quality random numbers suitable for use in cryptographic applications. In addition, according to one embodiment, the random number A comprises a bit length equal to the bit length of the sensitive data, S.

In addition, in STEP 820 the data splitting process 800 generates another statistically random number "C." According to the preferred embodiment, the generation of the statistically random numbers A and C may advantageously be done in parallel. The data splitting module then combines the numbers A and C with the sensitive data S such that new numbers "B" and "D" are generated. For example, number B may comprise the binary combination of A XOR S and number D may comprise the binary combination of C XOR S. The foregoing combinations preferably occur in STEPS 825 and 830, respectively, and, according to one embodiment, the foregoing combinations also occur in parallel. The data splitting process 800 then proceeds to STEP 835 where the random numbers A and C and the numbers B and D are paired such that none of the pairings contain sufficient data, by themselves, to reorganize and decipher the original sensitive data S. For example, the numbers may be paired as follows: AC, AD, BC, and BD. According to one embodiment, each of the foregoing pairings is distributed to one of the depositories D1 through D4 of FIGURE 7. According to another embodiment, each of the foregoing pairings is randomly distributed to one of the depositories D1 through D4. For example, during a first data splitting process 800, the pairing AC may be sent to depository D2, through, for example, a random selection of D2's IP address. Then, during a second data splitting process 800, the pairing AC may be sent to depository D4, through, for example, a random selection of D4's IP address.

Based on the foregoing, the data splitting process 800 advantageously places portions of the sensitive data in each of the four data storage facilities D1 through D4, such that no single data storage facility D1 through D4 includes sufficient encrypted data to recreate the original sensitive data S. As mentioned in the foregoing, such randomization of the data into individually unusable encrypted portions increases security and provides for maintained trust in the data even if one of the data storage facilities, D1 through D4, is compromised.

Although the data splitting process 800 is disclosed with reference to its preferred embodiment, the invention is not intended to be limited thereby. Rather a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the data splitting process 800. For example, the data splitting process may advantageously split the data into two numbers, for example, random number A and number B and, randomly distribute A and B through two data storage facilities. Moreover, the data splitting process 800 may advantageously split the data among a wide number of data storage facilities through generation of an additional random numbers.

As mentioned in the foregoing, in order to recreate the sensitive data S, the data portions need to be derandomized and reorganized. This process may advantageously occur in the data assembling modules, 525 and 620, of the authentication engine 215 and the cryptographic engine 220, respectively. The data assembling module, for example, data assembly module 525, receives data portions from the data storage facilities D1 through D4, and reassembles the data into useable form. For example, according to one embodiment where the data splitting module 520 employed the data splitting process 800 of FIGURE 8, the data assembling module 525 uses data portions from at least two of the data storage facilities D1 through D4 to recreate the sensitive data S. For example, the pairings of AC, AD, BC, and BD, were distributed such that any two provide one of A and B, or, C and D. Noting that S = A XOR B or S = C XOR D indicates that when the data assembling module receives one of A and B, or, C and D,

the data assembling module 525 can advantageously reassemble the sensitive data S. Thus, the data assembling module 525 may assemble the sensitive data S, when, for example, it receives data portions from at least the first two of the data storage facilities D1 through D4 to respond to an assemble request by the trust engine 110.

Based on the above data splitting and assembling processes, the sensitive data S exists in usable format only in a limited area of the trust engine 110. For example, when the sensitive data S includes enrollment authentication data, usable, nonrandomized enrollment authentication data is available only in the authentication engine 215. Likewise, when the sensitive data S includes private cryptographic key data, usable, nonrandomized private cryptographic key data is available only in the cryptographic engine 220.

Although the data splitting and assembling processes are disclosed with reference to their preferred embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for splitting and reassembling the sensitive data S. For example, public-key encryption may be used to further secure the data at the data storage facilities D1 through D4.

FIGURE 9A illustrates a data flow of an enrollment process 900 according to aspects of an embodiment of the invention. As shown in FIGURE 9A, the enrollment process 900 begins at STEP 905 when a user desires to enroll with the trust engine 110 of the cryptographic system 100. According to this embodiment, the user system 105 advantageously includes a client-side applet, such as a Java-based, that queries the user to enter enrollment data, such as demographic data and enrollment authentication data. According to one embodiment, the enrollment authentication data includes user ID, password(s), biometric(s), or the like. According to one embodiment, during the querying process, the client-side applet preferably communicates with the trust engine 110 to ensure that a chosen user ID is unique. When the user ID is nonunique, the trust engine 110 may advantageously suggest a unique user ID. The client-side applet gathers the enrollment data and transmits the enrollment data, for example, through and XML document, to the trust engine 110, and in particular, to the transaction engine 205. According to one embodiment, the transmission is encoded with the public key of the authentication engine 215.

According to one embodiment, the user performs a single enrollment during STEP 905 of the enrollment process 900. For example, the user enrolls himself or herself as a particular person, such as Joe User. When Joe User desires to enroll as Joe User, CEO of Mega Corp., then according to this embodiment, Joe User enrolls a second time, receives a second unique user ID and the trust engine 110 does not associate the two identities. According to another embodiment of the invention, the enrollment process 900 provides for multiple user identities for a single user ID. Thus, in the above example, the trust engine 110 will advantageously associate the two identities of Joe User. As will be understood by a skilled artisan from the disclosure herein, a user may have many identities, for example, Joe User the head of household, Joe User the member of the Charitable Foundations, and the like. Even though the user may have multiple identities, according to this embodiment, the trust engine 110 preferably stores only one set of enrollment data. Moreover, users may advantageously add, edit/update, or delete identities as they are needed.

Although the enrollment process 900 is disclosed with reference to its preferred embodiment, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of

alternatives for gathering of enrollment data, and in particular, enrollment authentication data. For example, the applet may be common object model (COM) based applet or the like.

On the other hand, the enrollment process may include graded enrollment. For example, at a lowest level of enrollment, the user may enroll over the communication link 125 without producing documentation as to his or her identity. According to an increased level of enrollment, the user enrolls using a trusted third party, such as a digital notary. For example, and the user may appear in person to the trusted third party, produce credentials such as a birth certificate, driver's license, military ID, or the like, and the trusted third party may advantageously include, for example, their digital signature in enrollment submission. The trusted third party may include an actual notary, a government agency, such as the Post Office or Department of Motor Vehicles, a human resources person in a large company enrolling an employee, or the like. A skilled artisan will understand from the disclosure herein that a wide number of varying levels of enrollment may occur during the enrollment process 900.

After receiving the enrollment authentication data, at STEP 915, the transaction engine 205, using conventional FULL SSL technology forwards the enrollment authentication data to the authentication engine 215. In STEP 920, the authentication engine 215 decrypts the enrollment authentication data using the private key of the authentication engine 215. In addition, the authentication engine 215 employs the data splitting module to mathematically operate on the enrollment authentication data so as to split the data into at least two independently undecipherable, randomized, numbers. As mentioned in the foregoing, the at least two numbers may comprise a statistically random number and a binary XORed number. In STEP 925, the authentication engine 215 forwards each portion of the randomized numbers to one of the data storage facilities D1 through D4. As mentioned in the foregoing, the authentication engine 215 may also advantageously randomize which portions are transferred to which depositories.

Often during the enrollment process 900, the user will also desire to have a digital certificate issued such that he or she may receive encrypted documents from others outside the cryptographic system 100. As mentioned in the foregoing, the certificate authority 115 generally issues digital certificates according to one or more of several conventional standards. Generally, the digital certificate includes a public key of the user or system, which is known to everyone.

Whether the user requests a digital certificate at enrollment, or at another time, the request is transferred through the trust engine 110 to the authentication engine 215. According to one embodiment, the request includes an XML document having, for example, the proper name of the user. According to STEP 935, the authentication engine 215 transfers the request to the cryptographic engine 220 instructing the cryptographic engine 220 to generate a cryptographic key or key pair.

Upon request, at STEP 935, the cryptographic engine 220 generates at least one cryptographic key. According to one embodiment, the cryptographic handling module 625 generates a key pair, where one key is used as a private key, and one is used as a public key. The cryptographic engine 220 stores the private key and, according to one embodiment, a copy of the public key. In STEP 945, the cryptographic engine 220 transmits a request for a digital

certificate to the transaction engine 205. According to one embodiment, the request advantageously includes a standardized request, such as PKCS10, embedded in, for example, and XML document. The request for a digital certificate may advantageously correspond to one or more certificate authorities and the one or more standard formats the certificate authorities require.

In STEP 950 the transaction engine 205 forwards this request to the certificate authority 115, who, in STEP 955, returns a digital certificate. The return digital certificate may advantageously be in a standardized format, such as PKCS7, or in a proprietary format of one or more of the certificate authorities 115. In STEP 960, the digital certificate is received by the transaction engine 205, and a copy is forwarded to the user and a copy is stored with the trust engine 110. The trust engine 110 stores a copy of the certificate such that the trust engine 110 will not need to rely on the availability of the certificate authority 115. For example, when the user desires to send a digital certificate, or a third party requests the user's digital certificate, the request for the digital certificate is typically sent to the certificate authority 115. However, if the certificate authority 115 is conducting maintenance or has been victim of a failure or security compromise, the digital certificate may not be available.

At any time after issuing the cryptographic keys, the cryptographic engine 220 may advantageously employ the data splitting process 800 described above such that the cryptographic keys are split into independently undecipherable randomized numbers. Similar to the authentication data, at STEP 965 the cryptographic engine 220 transfers the randomized numbers to the data storage facilities D1 through D4.

A skilled artisan will recognize from the disclosure herein that the user may request a digital certificate anytime after enrollment. Moreover, the communications between systems may advantageously include FULL SSL or public-key encryption technologies. Moreover, the enrollment process may issue multiple digital certificates from multiple certificate authorities, including one or more proprietary certificate authorities internal or external to the trust engine 110.

As disclosed in STEPS 935 through 960, one embodiment of the invention includes the request for a certificate that is eventually stored on the trust engine 110. Because, according to one embodiment, the cryptographic handling module 625 issues the keys used by the trust engine 110, each certificate corresponds to a private key. Therefore, the trust engine 110 may advantageously provide for interoperability through monitoring the certificates owned by, or associated with, a user. For example, when the cryptographic engine 220 receives a request for a cryptographic function, the cryptographic handling module 625 may investigate the certificates owned by the requesting user to determine whether the user owns a private key matching the attributes of the request. When such a certificate exists, the cryptographic handling module 625 may use the certificate or the public or private keys associated therewith, to perform the requested function. When such a certificate does not exist, the cryptographic handling module 625 may advantageously and transparently perform a number of actions to attempt to remedy the lack of an appropriate key. For example, FIGURE 9B illustrates a flowchart of an interoperability process 970, which according to aspects of an embodiment of the invention, discloses the foregoing steps to ensure the cryptographic handling module 625 performs cryptographic functions using appropriate keys.

As shown in FIGURE 9B, the interoperability process 970 begins with STEP 972 where the cryptographic handling module 925 determines the type of certificate desired. According to one embodiment of the invention, the type of certificate may advantageously be specified in the request for cryptographic functions, or other data provided by the requestor. According to another embodiment, the certificate type may be ascertained by the data format of the request. For example, the cryptographic handling module 925 may advantageously recognize the request corresponds to a particular type.

According to one embodiment, the certificate type may include one or more algorithm standards, for example, RSA, ELGAMAL, or the like. In addition, the certificate type may include one or more key types, such as symmetric keys, public keys, strong encryption keys such as 256 bit keys, less secure keys, or the like. Moreover, the certificate type may include upgrades or replacements of one or more of the foregoing algorithm standards or keys, one or more message or data formats, one or more data encapsulation or encoding schemes, such as Base 32 or Base 64. The certificate type may also include compatibility with one or more third-party cryptographic applications or interfaces, one or more communication protocols, or one or more certificate standards or protocols. A skilled artisan will recognize from the disclosure herein that other differences may exist in certificate types, and translations to and from those differences may be implemented as disclosed herein.

Once the cryptographic handling module 625 determines the certificate type, the interoperability process 970 proceeds to STEP 974, and determines whether the user owns a certificate matching the type determined in STEP 974. When the user owns a matching certificate, for example, the trust engine 110 has access to the matching certificate through, for example, prior storage thereof, the cryptographic handling module 625 knows that a matching private key is also stored within the trust engine 110. For example, the matching private key may be stored within the depository 210 or depository system 700. The cryptographic handling module 625 may advantageously request the matching private key be assembled from, for example, the depository 210, and then in STEP 976, use the matching private key to perform cryptographic actions or functions. For example, as mentioned in the foregoing, the cryptographic handling module 625 may advantageously perform hashing, hash comparisons, data encryption or decryption, digital signature verification or creation, or the like.

When the user does not own a matching certificate, the interoperability process 970 proceeds to STEP 978 where the cryptographic handling module 625 determines whether the users owns a cross-certified certificate. According to one embodiment, cross-certification between certificate authorities occurs when a first certificate authority determines to trust certificates from a second certificate authority. In other words, the first certificate authority determines that certificates from the second certificate authority meets certain quality standards, and therefore, may be "certified" as equivalent to the first certificate authority's own certificates. Cross-certification becomes more complex when the certificate authorities issue, for example, certificates having levels of trust . For example, the first certificate authority may provide three levels of trust for a particular certificate, usually based on the degree of reliability in the enrollment process, while the second certificate authority may provide seven levels of trust. Cross-certification may advantageously track which levels and which the certificates from the second

certificate authority may be substituted for which levels and which certificates from the first. When the foregoing cross-certification is done officially and publicly between two certification authorities, the mapping of certificates and levels to one another is often called "chaining."

According to another embodiment of the invention, the cryptographic handling module 625 may advantageously develop cross-certifications outside those agreed upon by the certificate authorities. For example, the cryptographic handling module 625 may access a first certificate authority's certificate practice statement (CPS), or other published policy statement, and using, for example, the authentication tokens required by particular trust levels, match the first certificate authority's certificates to those of another certificate authority.

When, in STEP 978, the cryptographic handling module 625 determines that the users owns a cross-certified certificate, the interoperability process 970 proceeds to STEP 976, and performs the cryptographic action or function using the cross-certified public key, private key, or both. Alternatively, when the cryptographic handling module 625 determines that the users does not own a cross-certified certificate, the interoperability process 970 proceeds to STEP 980, where the cryptographic handling module 625 selects a certificate authority that issues the requested certificate type, or a certificate cross-certified thereto. In STEP 982, the cryptographic handling module 625 determines whether the user enrollment authentication data, discussed in the foregoing, meets the authentication requirements of the chosen certificate authority. For example, if the user enrolled over a network by, for example, answering demographic and other questions, the authentication data provided may establish a lower level of trust than a user providing biometric data and appearing before a third-party, such as, for example, a notary. According to one embodiment, the foregoing authentication requirements may advantageously be provided in the chosen authentication authority's CPS.

When the user has provided the trust engine 110 with enrollment authentication data meeting the requirements of chosen certificate authority, the interoperability process 970 proceeds to STEP 984, where the cryptographic handling module 625 acquires the certificate from the chosen certificate authority. According to one embodiment, the cryptographic handling module 625 acquires the certificate by following STEPS 945 through 960 of the enrollment process 900. For example, the cryptographic handling module 625 may advantageously employ one or more public keys from one or more of the key pairs already available to the cryptographic engine 220, to request the certificate from the certificate authority. According to another embodiment, the cryptographic handling module 625 may advantageously generate one or more new key pairs, and use the public keys corresponding thereto, to request the certificate from the certificate authority.

According to another embodiment, the trust engine 110 may advantageously include one or more certificate issuing modules capable of issuing one or more certificate types. According to this embodiment, the certificate issuing module may provide the foregoing certificate. When the cryptographic handling module 625 acquires the certificate, the interoperability process 970 proceeds to STEP 976, and performs the cryptographic action or function using the public key, private key, or both corresponding to the acquired certificate.

When the user, in STEP 982, has not provided the trust engine 110 with enrollment authentication data meeting the requirements of chosen certificate authority, the cryptographic handling module 625 determines, in STEP

986 whether there are other certificate authorities that have different authentication requirements. For example, the cryptographic handling module 625 may look for certificate authorities having lower authentication requirements, but still issue the chosen certificates, or cross-certifications thereof.

When the foregoing certificate authority having lower requirements exists, the interoperability process 970 proceeds to STEP 980 and chooses that certificate authority. Alternatively, when no such certificate authority exists, in STEP 988, the trust engine 110 may request additional authentication tokens from the user. For example, the trust engine 110 may request new enrollment authentication data comprising, for example, biometric data. Also, the trust engine 110 may request the user appear before a trusted third party and provide appropriate authenticating credentials, such as, for example, appearing before a notary with a drivers license, social security card, bank card, birth certificate, military ID, or the like. When the trust engine 110 receives updated authentication data, the interoperability process 970 proceeds to Step 984 and acquires the foregoing chosen certificate.

Through the foregoing interoperability process 970, the cryptographic handling module 625 advantageously provides seamless, transparent, translations and conversions between differing cryptographic systems. A skilled artisan will recognize from the disclosure herein, a wide number of advantages and implementations of the foregoing interoperable system. For example, the foregoing STEP 986 of the interoperability process 970 may advantageously include aspects of trust arbitrage, discussed in further detail below, where the certificate authority may under special circumstances accept lower levels of cross-certification. In addition, the interoperability process 970 may include ensuring interoperability between and employment of standard certificate revocations, such as employing certificate revocation lists (CRL), online certificate status protocols (OCSP), or the like.

FIGURE 10 illustrates a data flow of an authentication process 1000 according to aspects of an embodiment of the invention. According to one embodiment, the authentication process 1000 includes gathering current authentication data from a user and comparing that to the enrollment authentication data of the user. For example, the authentication process 1000 begins at STEP 1005 where a user desires to perform a transaction with, for example, a vendor. Such transactions may include, for example, selecting a purchase option, requesting access to a restricted area or device of the vendor system 120, or the like. At STEP 1010, a vendor provides the user with a transaction ID and an authentication request. The transaction ID may advantageously include a 192 bit quantity having a 32 bit timestamp concatenated with a 128 bit random quantity, or a "nonce," concatenated with a 32 bit vendor specific constant. Such a transaction ID uniquely identifies the transaction such that copycat transactions can be refused by the trust engine 110.

The authentication request may advantageously include what level of authentication is needed for a particular transaction. For example, the vendor may specify a particular level of confidence that is required for the transaction at issue. If authentication cannot be made to this level of confidence, as will be discussed below, the transaction will not occur without either further authentication by the user to raise the level of confidence, or a change in the terms of the authentication between the vendor and the server. These issues are discussed more completely below.

According to one embodiment, the transaction ID and the authentication request may advantageously generated by a vendor-side applet or other software program. In addition, the transmission of the transaction ID and authentication data may include one or more XML documents encrypted using conventional SSL technology, such as, for example, ½ SSL, or, in other words vendor-side authenticated SSL.

After the user system 105 receives the transaction ID and authentication request, the user system 105 gathers the current authentication data, potentially including current biometric information, from the user. The user system 105, at STEP 1015, encrypts at least the current authentication data " B' " and the transaction ID, with the public key of the authentication engine 215, and transfers that data to the trust engine 110. The transmission preferably comprises XML documents encrypted with at least conventional ½ SSL technology. In STEP 1020, the transaction engine 205 receives the transmission, preferably recognizes the data format or request in the URL or URI, and forwards the transmission to the authentication engine 215.

During STEPS 1015 and 1020, the vendor system 120, at STEP 1025, forwards the transaction ID and the authentication request to the trust engine 110, using the preferred FULL SSL technology. This communication may also include a vendor ID, although vendor identification may also be communicated through a non-random portion of the transaction ID. At STEPS 1030 and 1035, the transaction engine 205 receives the communication, creates a record in the audit trail, and generates a request for the user's enrollment authentication data to be assembled from the data storage facilities D1 through D4. At STEP 1040, the depository system 700 transfers the portions of the enrollment authentication data corresponding to the user to the authentication engine 215. At STEP 1045, the authentication engine 215 decrypts the transmission using its private key and compares the enrollment authentication data to the current authentication data provided by the user.

The comparison of STEP 1045 may advantageously apply heuristical context sensitive authentication, as referred to in the forgoing, and discussed in further detail below. For example, if the biometric information received does not match perfectly, a lower confidence match results. In particular embodiments, the level of confidence of the authentication is balanced against the nature of the transaction and the desires of both the user and the vendor. Again, this is discussed in greater detail below.

At STEP 1050, the authentication engine 215 fills in the authentication request with the result of the comparison of STEP 1045. According to one embodiment of the invention, the authentication request is filled with a YES/NO or TRUE/FALSE result of the authentication process 1000. In STEP 1055 the filled-in authentication request is returned to the vendor for the vendor to act upon, for example, allowing the user to complete the transaction that initiated the authentication request. According to one embodiment, a confirmation message is passed to the user.

Based on the foregoing, the authentication process 1000 advantageously keeps sensitive data secure and produces results configured to maintain the integrity of the sensitive data. For example, the sensitive data is assembled only inside the authentication engine 215. For example, the enrollment authentication data is undecipherable until it is assembled in the authentication engine 215 by the data assembling module, and the current authentication data is undecipherable until it is unwrapped by the conventional SSL technology and the private key of

the authentication engine 215. Moreover, the authentication result transmitted to the vendor does not include the sensitive data, and the user may not even know whether he or she produced valid authentication data.

Although the authentication process 1000 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the authentication process 1000. For example, the vendor may advantageously be replaced by almost any requesting application, even those residing with the user system 105. For example, a client application, such as Microsoft Word, may use an application program interface (API) or a cryptographic API (CAPI) to request authentication before unlocking a document. Alternatively, a mail server, a network, a cellular phone, a personal or mobile computing device, a workstation, or the like, may all make authentication requests that can be filled by the authentication process 1000. In fact, after providing the foregoing trusted authentication process 1000, the requesting application or device may provide access to or use of a wide number of electronic or computer devices or systems.

Moreover, the authentication process 1000 may employ a wide number of alternative procedures in the event of authentication failure. For example, authentication failure may maintain the same transaction ID and request that the user reenter his or her current authentication data. As mentioned in the foregoing, use of the same transaction ID allows the comparator of the authentication engine 215 to monitor and limit the number of authentication attempts for a particular transaction, thereby creating a more secure cryptographic system 100.

In addition, the authentication process 1000 may be advantageously be employed to develop elegant single sign-on solutions, such as, unlocking a sensitive data vault. For example, successful or positive authentication may provide the authenticated user the ability to automatically access any number of passwords for an almost limitless number of systems and applications. For example, authentication of a user may provide the user access to password, login, financial credentials, or the like, associated with multiple online vendors, a local area network, various personal computing devices, Internet service providers, auction providers, investment brokerages, or the like. By employing a sensitive data vault, users may choose truly large and random passwords because they no longer need to remember them through association. Rather, the authentication process 1000 provides access thereto. For example, a user may choose a random alphanumeric string that is twenty plus digits in length rather than something associated with a memorable data, name, etc.

According to one embodiment, a sensitive data vault associated with a given user may advantageously be stored in the data storage facilities of the depository 210, or split and stored in the depository system 700. According to this embodiment, after positive user authentication, the trust engine 110 serves the requested sensitive data, such as, for example, to the appropriate password to the requesting application. According to another embodiment, the trust engine 110 may include a separate system for storing the sensitive data vault. For example, the trust engine 110 may include a stand-alone software engine implementing the data vault functionality and figuratively residing "behind" the foregoing front-end security system of the trust engine 110. According to this embodiment, the software engine

serves the requested sensitive data after the software engine receives a signal indicating positive user authentication from the trust engine 110.

In yet another embodiment, the data vault may be implemented by a third-party system. Similar to the software engine embodiment, the third-party system may advantageously serve the requested sensitive data after the third-party system receives a signal indicating positive user authentication from the trust engine 110. According to yet another embodiment, the data vault may be implemented on the user system 105. A user-side software engine may advantageously serve the foregoing data after receiving a signal indicating positive user authentication from the trust engine 110.

Although the foregoing data vaults are disclosed with reference to alternative embodiments, a skilled artisan will recognize from the disclosure herein, a wide number of additional implementations thereof. For example, a particular data vault may include aspects from some or all of the foregoing embodiments. In addition, any of the foregoing data vaults may employ one or more authentication requests at varying times. For example, any of the data vaults may require authentication every one or more transactions, periodically, every one ore more sessions, every access to one or more Webpages or Websites, at one or more other specified intervals, or the like.

FIGURE 11 illustrates a data flow of a signing process 1100 according to aspects of an embodiment of the invention. As shown in FIGURE 11, the signing process 1100 includes steps similar to those of the authentication process 1000 described in the foregoing with reference to FIGURE 10. According to one embodiment of the invention, the signing process 1100 first authenticates the user and then performs one or more of several digital signing functions as will be discussed in further detail below. According to another embodiment, the signing process 1100 may advantageously store data related thereto, such as hashes of messages or documents, or the like. This data may advantageously be used in an audit or any other event, such as for example, when a participating party attempts to repudiate a transaction.

As shown in FIGURE 11, during the authentication steps, the user and vendor may advantageously agree on a message, such as, for example, a contract. During signing, the signing process 1100 advantageously ensures that the contract signed by the user is identical to the contract supplied by the vendor. Therefore, according to one embodiment, during authentication, the vendor and the user include a hash of their respective copies of the message or contract, in the data transmitted to the authentication engine 215. By employing only a hash of a message or contract, the trust engine 110 may advantageously store a significantly reduced amount of data, providing for a more efficient and cost effective cryptographic system. In addition, the stored hash may be advantageously compared to a hash of a document in question to determine whether the document in question matches one signed by any of the parties. The ability to determine whether the document is identical to one relating to a transaction provides for additional evidence that can be used against a claim for repudiation by a party to a transaction.

In STEP 1103, the authentication engine 215 assembles the enrollment authentication data and compares it to the current authentication data provided by the user. When the comparator of the authentication engine 215 indicates that the enrollment authentication data matches the current authentication data, the comparator of the

authentication engine 215 also compares the hash of the message supplied by the vendor to the hash of the message supplied by the user. Thus, the authentication engine 215 advantageously ensures that the message agreed to by the user is identical to that agreed to by the vendor.

In STEP 1105, the authentication engine 215 transmits a digital signature request to the cryptographic engine 220. According to one embodiment of the invention, the request includes a hash of the message or contract. However, a skill artisan will recognize from the disclosure herein that the cryptographic engine 220 may encrypt virtually any text to form the desired digital signature. Returning to STEP 1105, the digital signature request preferably comprises an XML document communicated through conventional SSL technologies.

In STEP 1110, the authentication engine 215 transmits a request to each of the data storage facilities D1 through D4, such that each of the data storage facilities D1 through D4 transmit their respective portion of the cryptographic key or keys corresponding to a signing party. According to another embodiment, the cryptographic engine 220 employs some or all of the steps of the interoperability process 970 discussed in the foregoing, such that the cryptographic engine 220 first determines the appropriate key or keys to request from the depository 210 or the depository system 700 for the signing party, and takes actions to provide appropriate matching keys. According to still another embodiment, the authentication engine 215 or the cryptographic engine 220 may advantageously request one or more of the keys associated with the signing party and stored in the depository 210 or depository system 700.

According to one embodiment, the signing party includes one or both the user and the vendor. In such case, the authentication engine 215 advantageously requests the cryptographic keys corresponding to the user and/or the vendor. According to another embodiment, the signing party includes the trust engine 110. In this embodiment, the trust engine 110 is certifying that the authentication process 1000 properly authenticated the user, vendor, or both. Therefore, the authentication engine 215 requests the cryptographic key of the trust engine 110, such as, for example, the key belonging to the cryptographic engine 220, to perform the digital signature. According to another embodiment, the trust engine 110 performs a digital notary-like function. In this embodiment, the signing party includes the user, vendor, or both, along with the trust engine 110. Thus, the trust engine 110 provides the digital signature of the user and/or vendor, and then indicates with its own digital signature that the user and/or vendor were properly authenticated. In this embodiment, the authentication engine 215 may advantageously request assembly of the cryptographic keys corresponding to the user, the vendor, or both. According to another embodiment, the authentication engine 215 may advantageously request assembly of the cryptographic keys corresponding to the trust engine 110.

According to another embodiment, the trust engine 110 performs power of attorney-like functions. For example, the trust engine 110 may digitally sign the message on behalf of a third party. In such case, the authentication engine 215 requests the cryptographic keys associated with the third party. According to this embodiment, the signing process 1100 may advantageously include authentication of the third party, before allowing power of attorney-like functions. In addition, the authentication process 1000 may include a check for third party

constraints, such as, for example, business logic or the like dictating when and in what circumstances a particular third-party's signature may be used.

Based on the foregoing, in STEP 1110, the authentication engine requested the cryptographic keys from the data storage facilities D1 through D4 corresponding to the signing party. In STEP 1115, the data storage facilities D1 through D4 transmit their respective portions of the cryptographic key corresponding to the signing party to the cryptographic engine 220. According to one embodiment, the foregoing transmissions include SSL technologies. According to another embodiment, the foregoing transmissions may advantageously be super-encrypted with the public key of the cryptographic engine 220.

In STEP 1120, the cryptographic engine 220 assembles the foregoing cryptographic keys of the signing party and encrypts the message therewith, thereby forming the digital signature(s). In STEP 1125 of the signing process 1100, the cryptographic engine 220 transmits the digital signature(s) to the authentication engine 215. In STEP 1130, the authentication engine 215 transmits the filled-in authentication request along with a copy of the hashed message and the digital signature(s) to the transaction engine 205. In STEP 1135, the transaction engine 205 transmits a receipt comprising the transaction ID, an indication of whether the authentication was successful, and the digital signature(s), to the vendor. According to one embodiment, the foregoing transmission may advantageously include the digital signature of the trust engine 110. For example, the trust engine 110 may encrypt the hash of the receipt with its private key, thereby forming a digital signature to be attached to the transmission to the vendor.

According to one embodiment, the transaction engine 205 also transmits a confirmation message to the user.

Although the signing process 1100 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the signing process 1100. For example, the vendor may be replaced with a user application, such as an email application. For example, the user may wish to digitally sign a particular email with his or her digital signature. In such an embodiment, the transmission throughout the signing process 1100 may advantageously include only one copy of a hash of the message. Moreover, a skilled artisan will recognize from the disclosure herein that a wide number of client applications may request digital signatures. For example, the client applications may comprise word processors, spreadsheets, emails, voicemail, access to restricted system areas, or the like.

In addition, a skilled artisan will recognize from the disclosure herein that STEPS 1105 through 1120 of the signing process 1100 may advantageously employ some or all of the steps of the interoperability process 970 of FIGURE 9B, thereby providing interoperability between differing cryptographic systems that may, for example, need to process the digital signature under differing signature types.

FIGURE 12 illustrates a data flow of an encryption/decryption process 1200 according to aspects of an embodiment of the invention. As shown in FIGURE 12, the decryption process 1200 begins by authenticating the user using the authentication process 1000. According to one embodiment, the authentication process 1000 includes in the authentication request, a synchronous session key. For example, in conventional PKI technologies, it is understood by

skilled artisans that encrypting or decrypting data using public and private keys is mathematically intensive and may require significant system resources. However, in symmetric key cryptographic systems, or systems where the sender and receiver of a message share a single common key that is used to encrypt and decrypt a message, the mathematical operations are significantly simpler and faster. Thus, in the conventional PKI technologies, the sender of a message will generate synchronous session key, and encrypt the message using the simpler, faster symmetric key system. Then, the sender will encrypt the session key with the public key of the receiver. The encrypted session key will be attached to the synchronously encrypted message and both data are sent to the receiver. The receiver uses his or her private key to decrypt the session key, and then uses the session key to decrypt the message. Based on the foregoing, the simpler and faster symmetric key system is used for the majority of the encryption/decryption processing. Thus, in the decryption process 1200, the decryption advantageously assumes that a synchronous key has been encrypted with the public key of the user. Thus, as mentioned in the foregoing, the encrypted session key is included in the authentication request.

Returning to the decryption process 1200, after the user has been authenticated in STEP 1205, the authentication engine 215 forwards the encrypted session key to the cryptographic engine 220. In STEP 1210, the authentication engine 215 forwards a request to each of the data storage facilities, D1 through D4, requesting the cryptographic key data of the user. In STEP 1215, each data storage facility, D1 through D4, transmits their respective portion of the cryptographic key to the cryptographic engine 220. According to one embodiment, the foregoing transmission is encrypted with the public key of the cryptographic engine 220.

In STEP 1220 of the decryption process 1200, the cryptographic engine 220 assembles the cryptographic key and decrypts the session key therewith. In STEP 1225, the cryptographic engine forwards the session key to the authentication engine 215. In STEP 1227, the authentication engine 215 fills in the authentication request including the decrypted session key, and transmits the filled-in authentication request to the transaction engine 205. In STEP 1230, the transaction engine 205 forwards the authentication request along with the session key to the requesting application or vendor. Then, according to one embodiment, the requesting application or vendor uses the session key to decrypt the encrypted message.

Although the decryption process 1200 is disclosed with reference to its preferred and alternative embodiments, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the decryption process 1200. For example, the decryption process 1200 may forego synchronous key encryption and rely on full public-key technology. In such an embodiment, the requesting application may transmit the entire message to the cryptographic engine 220, or, may employ some type of compression or reversible hash in order to transmit the message to the cryptographic engine 220. A skilled artisan will also recognize from the disclosure herein that the foregoing communications may advantageously include XML documents wrapped in SSL technology.

The encryption/decryption process 1200 also provides for encryption of documents or other data. Thus, in STEP 1235, a requesting application or vendor may advantageously transmit to the transaction engine 205 of the trust engine 110, a request for the public key of the user. The requesting application or vendor makes this request

because the requesting application or vendor uses the public key of the user, for example, to encrypt the session key that will be used to encrypt the document or message. As mentioned in the enrollment process 900, the transaction engine 205 stores a copy of the digital certificate of the user, for example, in the mass storage 225. Thus, in STEP 1240 of the encryption process 1200, the transaction engine 205 requests the digital certificate of the user from the mass storage 225. In STEP 1245, the mass storage 225 transmits the digital certificate corresponding to the user, to the transaction engine 205. In STEP 1250, the transaction engine 205 transmits the digital certificate to the requesting application or vendor. According to one embodiment, the encryption portion of the encryption process 1200 does not include the authentication of a user. This is because the requesting vendor needs only the public key of the user, and is not requesting any sensitive data.

A skilled artisan will recognize from the disclosure herein that if a particular user does not have a digital certificate, the trust engine 110 may employ some or all of the enrollment process 900 in order to generate a digital certificate for that particular user. Then, the trust engine 110 may initiate the encryption/decryption process 1200 and thereby provide the appropriate digital certificate. In addition, a skilled artisan will recognize from the disclosure herein that STEPS 1220 and 1235 through 1250 of the encryption/decryption process 1200 may advantageously employ some or all of the steps of the interoperability process of FIGURE 9B, thereby providing interoperability between differing cryptographic systems that may, for example, need to process the encryption.

FIGURE 13 illustrates a simplified block diagram of a trust engine system 1300 according to aspects of yet another embodiment of the invention. As shown in FIGURE 13, the trust engine system 1300 comprises a plurality of distinct trust engines 1305, 1310, 1315, and 1320, respectively. To facilitate a more complete understanding of the invention, FIGURE 13 illustrates each trust engine, 1305, 1310, 1315, and 1320 as having a transaction engine, a depository, and an authentication engine. However, a skilled artisan will recognize that each transaction engine may advantageously comprise some or all of the elements and communication channels disclosed with reference to FIGURES 1-8. For example, one embodiment may advantageously include trust engines having transaction engines, depositories, and cryptographic servers.

According to one embodiment of the invention, each of the trust engines 1305, 1310, 1315 and 1320 are geographically separated, such that, for example, the trust engine 1305 may reside in a first location, the trust engine 1310 may reside in a second location, the trust engine 1315 may reside in a third location, and the trust engine 1320 may reside in a fourth location. The foregoing geographic separation advantageously decreases system response time while increasing the security of the overall trust engine system 1300.

For example, when a user logs onto the cryptographic system 100, the user may be nearest the first location and may desire to be authenticated. As described with reference to FIGURE 10, to be authenticated, the user provides current authentication data, such as a biometric or the like, and the current authentication data is compared to that user's enrollment authentication data. Therefore, according to one example, the user advantageously provides current authentication data to the geographically nearest trust engine 1305. The transaction engine 1321 of the trust engine 1305 then forwards the current authentication data to the authentication engine 1322 also residing at the first

location. According to another embodiment, the transaction engine 1321 forwards the current authentication data to one or more of the authentication engines of the trust engines 1310, 1315, or 1320.

The transaction engine 1321 also requests the assembly of the enrollment authentication data from the depositories of, for example, each of the trust engines, 1305 through 1320. According to this embodiment, each depository provides its portion of the enrollment authentication data to the authentication engine 1322 of the trust engine 1305. The authentication engine 1322 then employs the encrypted data portions from, for example, the first two depositories to respond, and assembles the enrollment authentication data into deciphered form. The authentication engine 1322 compares the enrollment authentication data with the current authentication data and returns an authentication result to the transaction engine 1321 of the trust engine 1305.

Based on the above, the trust engine system 1300 employs the nearest one of a plurality of geographically separated trust engines, 1305 through 1320, to perform the authentication process. According to one embodiment of the invention, the routing of information to the nearest transaction engine may advantageously be performed at client-side applets executing on one or more of the user system 105, vendor system 120, or certificate authority 115. According to an alternative embodiment, a more sophisticated decision process may be employed to select from the trust engines 1305 through 1320. For example, the decision may be based on the availability, operability, speed of connections, load, performance, geographic proximity, or a combination thereof, of a given trust engine.

In this way, the trust engine system 1300 lowers its response time while maintaining the security advantages associated with geographically remote data storage facilities, such as those discussed with reference to FIGURE 7 where each data storage facility stores randomized portions of sensitive data. For example, a security compromise at, for example, the depository 1325 of the trust engine 1315 does not necessarily compromise the sensitive data of the trust engine system 1300. This is because the depository 1325 contains only non-decipherable randomized data that, without more, is entirely useless.

According to another embodiment, the trust engine system 1300 may advantageously include multiple cryptographic engines arranged similar to the authentication engines. The cryptographic engines may advantageously perform cryptographic functions such as those disclosed with reference to FIGURES 1-8. According to yet another embodiment, the trust engine system 1300 may advantageously replace the multiple authentication engines with multiple cryptographic engines, thereby performing cryptographic functions such as those disclosed with reference to FIGURES 1-8. According to yet another embodiment of the invention, the trust engine system 1300 may replace each multiple authentication engine with an engine having some or all of the functionality of the authentication engines, cryptographic engines, or both, as disclosed in the foregoing.

Although the trust engine system 1300 is disclosed with reference to its preferred and alternative embodiments, a skilled artisan will recognize that the trust engine system 1300 may comprise portions of trust engines 1305 through 1320. For example, the trust engine system 1300 may include one or more transaction engines, one or more depositories, one or more authentication engines, or one or more cryptographic engines.

FIGURE 14 illustrates a simplified block diagram of a trust engine system 1400 according to aspects of yet another embodiment of the invention. As shown in FIGURE 14, the trust engine system 1400 includes multiple trust engines 1405, 1410, 1415 and 1420. According to one embodiment, each of the trust engines 1405, 1410, 1415 and 1420, comprise some or all of the elements of trust engine 110 disclosed with reference to FIGURES 1-8. According to this embodiment, when the client side applets of the user system 105, the vendor system 120, or the certificate authority 115, communicate with the trust engine system 1400, those communications are sent to the IP address of each of the trust engines 1405 through 1420. Further, each transaction engine of each of the trust engines, 1405, 1410, 1415, and 1420, behaves similar to the transaction engine 1321 of the trust engine 1305 disclosed with reference to FIGURE 13. For example, during an authentication process, each transaction engine of each of the trust engines 1405, 1410, 1415, and 1420 transmits the current authentication data to their respective authentication engines and transmits a request to assemble the randomized data stored in each of the depositories of each of the trust engines 1405 through 1420. FIGURE 14 does not illustrate all of these communications, as such illustration would become overly complex. Continuing with the authentication process, each of the depositories then communicates its portion of the randomized data to each of the authentication engines of the each of the trust engines 1405 through 1420. Each of the authentication engines of the each of the trust engines employs its comparator to determine whether the current authentication data matches the enrollment authentication data provided by the depositories of each of the trust engines 1405 through 1420. According to this embodiment, the result of the comparison by each of the authentication engines is then transmitted to a redundancy module of the other three trust engines. For example, the result of the authentication engine from the trust engine 1405 is transmitted to the redundancy modules of the trust engines 1410, 1415, and 1420. Thus, the redundancy module of the trust engine 1405 likewise receives the result of the authentication engines from the trust engines 1410, 1415, and 1420.

FIGURE 15 illustrates a block diagram of the redundancy module of FIGURE 14. The redundancy module comprises a comparator configured to receive the authentication result from three authentication engines and transmit that result to the transaction engine of the fourth trust engine. The comparator compares the authentication result form the three authentication engines, and if two of the results agree, the comparator concludes that the authentication result should match that of the two agreeing authentication engines. This result is then transmitted back to the transaction engine corresponding to the trust engine not associated with the three authentication engines.

Based on the foregoing, the redundancy module determines an authentication result from data received from authentication engines that are preferably geographically remote from the trust engine of that the redundancy module. By providing such redundancy functionality, the trust engine system 1400 ensures that a compromise of the authentication engine of one of the trust engines 1405 through 1420, is insufficient to compromise the authentication result of the redundancy module of that particular trust engine. A skilled artisan will recognize that redundancy module functionality of the trust engine system 1400 may also be applied to the cryptographic engine of each of the trust engines 1405 through 1420. However, such cryptographic engine communication was not shown in FIGURE 14

to avoid complexity. Moreover, a skilled artisan will recognize a wide number of alternative authentication result conflict resolution algorithms for the comparator of FIGURE 15.

According to yet another embodiment of the invention, the trust engine system 1400 may advantageously employ the redundancy module during cryptographic comparison steps. For example, some or all of the foregoing redundancy module disclosure with reference to FIGURES 14 and 15 may advantageously be implemented during a hash comparison of documents provided by one or more parties during a particular transaction.

Although the foregoing invention has been described in terms of certain preferred and alternative embodiments, other embodiments will be apparent to those of ordinary skill in the art from the disclosure herein. For example, the trust engine 110 may issue short-term certificates, where the private cryptographic key is released to the user for a predetermined period of time. For example, current certificate standards include a validity field that can be set to expire after a predetermined amount of time. Thus, the trust engine 110 may release a private key to a user where the private key would be valid for, for example, 24 hours. According to such an embodiment, the trust engine 110 may advantageously issue a new cryptographic key pair to be associated with a particular user and then release the private key of the new cryptographic key pair. Then, once the private cryptographic key is released, the trust engine 110 immediately expires any internal valid use of such private key, as it is no longer securable by the trust engine 110.

In addition, a skilled artisan will recognize that the cryptographic system 100 or the trust engine 110 may include the ability to recognize types of devices, such as a laptop, a cell phone, a network or the like. According to one embodiment, such recognition may come from data supplied in the request for a particular service, such as, a request for authentication leading to access or use, a request for cryptographic functionality, or the like. According to one embodiment, the foregoing request may include a unique device identifier, such as, for example, a processor ID. Alternatively, the request may include data in a particular recognizable data format. For example, mobile and satellite phones often do not include the processing power for full X509.v3 heavy encryption certificates, and therefore do not request them. According to this embodiment, the trust engine 110 may recognize the type of data format presented, and respond only in kind.

In an additional aspect of the system described above, context sensitive authentication can be provided using various techniques as will be described below. Context sensitive authentication, for example as shown in FIGURE 16, provides the possibility of evaluating not only the actual data which is sent by the user when attempting to authenticate himself, but also the circumstances surrounding the generation and delivery of that data. Such techniques may also support transaction specific trust arbitrage between the user and trust engine 110 or between the vendor and trust engine 110, as will be described below.

As discussed above, authentication is the process of proving that a user is who he says he is. Generally, authentication requires demonstrating some fact to an authentication authority. The trust engine 110 of the present invention represents the authority to which a user must authenticate himself. The user must demonstrate to the trust engine 110 that he is who he says he is by either: knowing something that only the user should know (knowledge-

based authentication), having something that only the user should have (token-based authentication), or by being something that only the user should be (biometric-based authentication).

Examples of knowledge-based authentication include without limitation a password, PIN number, or lock combination. Examples of token-based authentication include without limitation a house key, a physical credit card, a driver's license, or a particular phone number. Examples of biometric-based authentication include without limitation a fingerprint, a voice analysis, or a retinal scan.

Each type of authentication has particular advantages and disadvantages, and each provides a different level of security. For example, it is generally harder to create a false fingerprint that matches someone else's than it is to overhear someone's password and repeat it. Each type of authentication also requires a different type of data to be known to the authenticating authority in order to verify someone using that form of authentication.

As used herein, "authentication" will refer broadly to the overall process of verifying someone's identity to be who he says he is. An "authentication technique" will refer to a particular type of authentication based upon a particular piece of knowledge, physical token, or biometric reading. "Authentication data" refers to information which is sent to or otherwise demonstrated to an authentication authority in order to establish identity. "Enrollment data" will refer to the data which is initially submitted to an authentication authority in order to establish a baseline for comparison with authentication data. An "authentication instance" will refer to the data associated with an attempt to authenticate by an authentication technique.

The internal protocols and communications involved in the process of authenticating a user is described with reference to FIGURE 10 above. The part of this process within which the context sensitive authentication takes place occurs within the comparison step shown as STEP 1045 of FIGURE 10. This step takes place within the authentication engine 215 and involves assembling the enrollment data 410 retrieved from the depository 210 and comparing the authentication data provided by the user to it. One particular embodiment of this process is shown in FIGURE 16 and described below.

The current authentication data provided by the user and the enrollment data retrieved from the depository 210 are received by the authentication engine 215 in STEP 1600 of FIGURE 16. Both of these sets of data may contain data which is related to separate techniques of authentication. The authentication engine 215 separates the authentication data associated with each individual authentication instance in STEP 1605. This is necessary so that the authentication data is compared with the appropriate subset of the enrollment data for the user (*e.g.* fingerprint authentication data should be compared with fingerprint enrollment data, rather than password enrollment data).

Generally, authenticating a user involves one or more individual authentication instances, depending on which authentication techniques are available to the user. These methods are limited by the enrollment data which were provided by the user during his enrollment process (if the user did not provide a retinal scan when enrolling, he will not be able to authenticate himself using a retinal scan), as well as the means which may be currently available to the user (*e.g.* if the user does not have a fingerprint reader at his current location, fingerprint authentication will not be practical). In some cases, a single authentication instance may be sufficient to authenticate a user; however, in

certain circumstances a combination of multiple authentication instances may be used in order to more confidently authenticate a user for a particular transaction.

Each authentication instance consists of data related to a particular authentication technique (*e.g.* fingerprint, password, smart card, etc.) and the circumstances which surround the capture and delivery of the data for that particular technique. For example, a particular instance of attempting to authenticate via password will generate not only the data related to the password itself, but also circumstantial data, known as "metadata", related to that password attempt. This circumstantial data includes information such as: the time at which the particular authentication instance took place, the network address from which the authentication information was delivered, as well as any other information as is known to those of skill in the art which may be determined about the origin of the authentication data (the type of connection, the processor serial number, etc.).

In many cases, only a small amount of circumstantial metadata will be available. For example, if the user is located on a network which uses proxies or network address translation or another technique which masks the address of the originating computer, only the address of the proxy or router may be determined. Similarly, in many cases information such as the processor serial number will not be available because of either limitations of the hardware or operating system being used, disabling of such features by the operator of the system, or other limitations of the connection between the user's system and the trust engine 110.

As shown in FIGURE 16, once the individual authentication instances represented within the authentication data are extracted and separated in STEP 1605, the authentication engine 215 evaluates each instance for its reliability in indicating that the user is who he claims to be. The reliability for a single authentication instance will generally be determined based on several factors. These may be grouped as factors relating to the reliability associated with the authentication technique, which are evaluated in STEP 1610, and factors relating to the reliability of the particular authentication data provided, which are evaluated in STEP 1615. The first group includes without limitation the inherent reliability of the authentication technique being used, and the reliability of the enrollment data being used with that method. The second group includes without limitation the degree of match between the enrollment data and the data provided with the authentication instance, and the metadata associated with that authentication instance. Each of these factors may vary independently of the others.

The inherent reliability of an authentication technique is based on how hard it is for an imposter to provide someone else's correct data, as well as the overall error rates for the authentication technique. For passwords and knowledge based authentication methods, this reliability is often fairly low because there is nothing that prevents someone from revealing their password to another person and for that second person to use that password. Even a more complex knowledge based system may have only moderate reliability since knowledge may be transferred from person to person fairly easily. Token based authentication, such as having a proper smart card or using a particular terminal to perform the authentication, is similarly of low reliability used by itself, since there is no guarantee that the right person is in possession of the proper token.

However, biometric techniques are more inherently reliable because it is generally difficult to provide someone else with the ability to use your fingerprints in a convenient manner, even intentionally. Because subverting biometric authentication techniques is more difficult, the inherent reliability of biometric methods is generally higher than that of purely knowledge or token based authentication techniques. However, even biometric techniques may have some occasions in which a false acceptance or false rejection is generated. These occurrences may be reflected by differing reliabilities for different implementations of the same biometric technique. For example, a fingerprint matching system provided by one company may provide a higher reliability than one provided by a different company because one uses higher quality optics or a better scanning resolution or some other improvement which reduces the occurrence of false acceptances or false rejections.

Note that this reliability may be expressed in different manners. The reliability is desirably expressed in some metric which can be used by the heuristics 530 and algorithms of the authentication engine 215 to calculate the confidence level of each authentication. One preferred mode of expressing these reliabilities is as a percentage or fraction. For instance, fingerprints might be assigned an inherent reliability of 97%, while passwords might only be assigned an inherent reliability of 50%. Those of skill in the art will recognize that these particular values are merely exemplary and may vary between specific implementations.

The second factor for which reliability must be assessed is the reliability of the enrollment. This is part of the "graded enrollment" process referred to above. This reliability factor reflects the reliability of the identification provided during the initial enrollment process. For instance, if the individual initially enrolls in a manner where they physically produce evidence of their identity to a notary or other public official, and enrollment data is recorded at that time and notarized, the data will be more reliable than data which is provided over a network during enrollment and only vouched for by a digital signature or other information which is not truly tied to the individual.

Other enrollment techniques with varying levels of reliability include without limitation: enrollment at a physical office of the trust engine 110 operator; enrollment at a user's place of employment; enrollment at a post office or passport office; enrollment through an affiliated or trusted party to the trust engine 110 operator; anonymous or pseudonymous enrollment in which the enrolled identity is not yet identified with a particular real individual, as well as such other means as are known in the art.

These factors reflect the trust between the trust engine 110 and the source of identification provided during the enrollment process. For instance, if enrollment is performed in association with an employer during the initial process of providing evidence of identity, this information may be considered extremely reliable for purposes within the company, but may be trusted to a lesser degree by a government agency, or by a competitor. Therefore, trust engines operated by each of these other organizations may assign different levels of reliability to this enrollment.

Similarly, additional data which is submitted across a network, but which is authenticated by other trusted data provided during a previous enrollment with the same trust engine 110 may be considered as reliable as the original enrollment data was, even though the latter data were submitted across an open network. In such circumstances, a subsequent notarization will effectively increase the level of reliability associated with the original enrollment data. In

this way for example, an anonymous or pseudonymous enrollment may then be raised to a full enrollment by demonstrating to some enrollment official the identity of the individual matching the enrolled data.

The reliability factors discussed above are generally values which may be determined in advance of any particular authentication instance. This is because they are based upon the enrollment and the technique, rather than the actual authentication. In one embodiment, the step of generating reliability based upon these factors involves looking up previously determined values for this particular authentication technique and the enrollment data of the user. In a further aspect of an advantageous embodiment of the present invention, such reliabilities may be included with the enrollment data itself. In this way, these factors are automatically delivered to the authentication engine 215 along with the enrollment data sent from the depository 210.

While these factors may generally be determined in advance of any individual authentication instance, they still have an effect on each authentication instance which uses that particular technique of authentication for that user. Furthermore, although the values may change over time (*e.g.* if the user re-enrolls in a more reliable fashion), they are not dependent on the authentication data itself. By contrast, the reliability factors associated with a single specific instance's data may vary on each occasion. These factors, as discussed below, must be evaluated for each new authentication in order to generate reliability scores in STEP 1615.

The reliability of the authentication data reflects the match between the data provided by the user in a particular authentication instance and the data provided during the authentication enrollment. This is the fundamental question of whether the authentication data matches the enrollment data for the individual the user is claiming to be. Normally, when the data do not match, the user is considered to not be successfully authenticated, and the authentication fails. The manner in which this is evaluated may change depending on the authentication technique used. The comparison of such data is performed by the comparator 515 function of the authentication engine 215 as shown in FIGURE 5.

For instance, matches of passwords are generally evaluated in a binary fashion. In other words, a password is either a perfect match, or a failed match. It is usually not desirable to accept as even a partial match a password which is close to the correct password if it is not exactly correct. Therefore, when evaluating a password authentication, the reliability of the authentication returned by the comparator 515 is typically either 100% (correct) or 0% (wrong), with no possibility of intermediate values.

Similar rules to those for passwords are generally applied to token based authentication methods, such as smart cards. This is because having a smart card which has a similar identifier or which is similar to the correct one, is still just as wrong as having any other incorrect token. Therefore tokens tend also to be binary authenticators: a user either has the right token, or he doesn't.

However, certain types of authentication data, such as questionnaires and biometrics, are generally not binary authenticators. For example, a fingerprint may match a reference fingerprint to varying degrees. To some extent, this may be due to variations in the quality of the data captured either during the initial enrollment or in subsequent authentications. (A fingerprint may be smudged or a person may have a still healing scar or burn on a

particular finger.) In other instances the data may match less than perfectly because the information itself is somewhat variable and based upon pattern matching. (A voice analysis may seem close but not quite right because of background noise, or the acoustics of the environment in which the voice is recorded, or because the person has a cold.) Finally, in situations where large amounts of data are being compared, it may simply be the case that much of the data matches well, but some doesn't. (A ten-question questionnaire may have resulted in eight correct answers to personal questions, but two incorrect answers.) For any of these reasons, the match between the enrollment data and the data for a particular authentication instance may be desirably assigned a partial match value by the comparator 515. In this way, the fingerprint might be said to be a 85% match, the voice print a 65% match, and the questionnaire an 80% match, for example.

This measure (degree of match) produced by the comparator 515 is the factor representing the basic issue of whether an authentication is correct or not. However, as discussed above, this is only one of the factors which may be used in determining the reliability of a given authentication instance. Note also that even though a match to some partial degree may be determined, that ultimately, it may be desirable to provide a binary result based upon a partial match. In an alternate mode of operation, it is also possible to treat partial matches as binary, *i.e.* either perfect (100%) or failed (0%) matches, based upon whether or not the degree of match passes a particular threshold level of match. Such a process may be used to provide a simple pass/fail level of matching for systems which would otherwise produce partial matches.

Another factor to be considered in evaluating the reliability of a given authentication instance concerns the circumstances under which the authentication data for this particular instance are provided. As discussed above, the circumstances refer to the metadata associated with a particular authentication instance. This may include without limitation such information as: the network address of the authenticator, to the extent that it can be determined; the time of the authentication; the mode of transmission of the authentication data (phone line, cellular, network, etc.); and the serial number of the system of the authenticator.

These factors can be used to produce a profile of the type of authentication that is normally requested by the user. Then, this information can be used to assess reliability in at least two manners. One manner is to consider whether the user is requesting authentication in a manner which is consistent with the normal profile of authentication by this user. If the user normally makes authentication requests from one network address during business days (when she is at work) and from a different network address during evenings or weekends (when she is at home), an authentication which occurs from the home address during the business day is less reliable because it is outside the normal authentication profile. Similarly, if the user normally authenticates using a fingerprint biometric and in the evenings, an authentication which originates during the day using only a password is less reliable.

An additional way in which the circumstantial metadata can be used to evaluate the reliability of an instance of authentication is to determine how much corroboration the circumstance provides that the authenticator is the individual he claims to be. For instance, if the authentication comes from a system with a serial number known to be associated with the user, this is a good circumstantial indicator that the user is who they claim to be. Conversely, if

the authentication is coming from a network address which is known to be in Los Angeles when the user is known to reside in London, this is an indication that this authentication is less reliable based on its circumstances.

It is also possible that a cookie or other electronic data may be placed upon the system being used by a user when they interact with a vendor system or with the trust engine 110. This data is written to the storage of the system of the user and may contain an identification which may be read by a Web browser or other software on the user system. If this data is allowed to reside on the user system between sessions (a "persistent cookie"), it may be sent with the authentication data as further evidence of the past use of this system during authentication of a particular user. In effect, the metadata of a given instance, particularly a persistent cookie, may form a sort of token based authenticator itself.

Once the appropriate reliability factors based on the technique and data of the authentication instance are generated as described above in STEPS 1610 and 1615 respectively, they are used to produce an overall reliability for the authentication instance provided in STEP 1620. One means of doing this is simply to express each reliability as a percentage and then to multiply them together.

For example, suppose the authentication data is being sent in from a network address known to be the user's home computer completely in accordance with the user's past authentication profile (100%), and the technique being used is fingerprint identification (97%), and the initial finger print data was registered through the user's employer with the trust engine 110 (90%), and the match between the authentication data and the original fingerprint template in the enrollment data is very good (99%). The overall reliability of this authentication instance could then be calculated as the product of these reliabilities: 100% * 97% * 90% * 99% = 86.4% reliability.

This calculated reliability represents the reliability of one single instance of authentication. The overall reliability of a single authentication instance may also be calculated using techniques which treat the different reliability factors differently, for example by using formulas where different weights are assigned to each reliability factor. Furthermore, those of skill in the art will recognize that the actual values used may represent values other than percentages and may use non-arithmetic systems. One embodiment may include a module used by an authentication requestor to set the weights for each factor and the algorithms used in establishing the overall reliability of the authentication instance.

The authentication engine 215 may use the above techniques and variations thereof to determine the reliability of a single authentication instance, indicated as STEP 1620. However, it may be useful in many authentication situations for multiple authentication instances to be provided at the same time. For example, while attempting to authenticate himself using the system of the present invention, a user may provide a user identification, fingerprint authentication data, a smart card, and a password. In such a case, three independent authentication instances are being provided to the trust engine 110 for evaluation. Proceeding to STEP 1625, if the authentication engine 215 determines that the data provided by the user includes more than one authentication instance, then each instance in turn will be selected as shown in STEP 1630 and evaluated as described above in STEPS 1610, 1615 and 1620.

Note that many of the reliability factors discussed may vary from one of these instances to another. For instance, the inherent reliability of these techniques is likely to be different, as well as the degree of match provided between the authentication data and the enrollment data. Furthermore, the user may have provided enrollment data at different times and under different circumstances for each of these techniques, providing different enrollment reliabilities for each of these instances as well. Finally, even though the circumstances under which the data for each of these instances is being submitted is the same, the use of such techniques may each fit the profile of the user differently, and so may be assigned different circumstantial reliabilities. (For example, the user may normally use their password and fingerprint, but not their smart card.)

As a result, the final reliability for each of these authentication instances may be different from one another. However, by using multiple instances together, the overall confidence level for the authentication will tend to increase.

Once the authentication engine has performed STEPS 1610 through 1620 for all of the authentication instances provided in the authentication data, the reliability of each instance is used in STEP 1635 to evaluate the overall authentication confidence level. This process of combining the individual authentication instance reliabilities into the authentication confidence level may be modeled by various methods relating the individual reliabilities produced, and may also address the particular interaction between some of these authentication techniques. (For example, multiple knowledge-based systems such as passwords may produce less confidence than a single password and even a fairly weak biometric, such as a basic voice analysis.)

One means in which the authentication engine 215 may combine the reliabilities of multiple concurrent authentication instances to generate a final confidence level is to multiply the unreliability of each instance to arrive at a total unreliability. The unreliability is generally the complementary percentage of the reliability. For example, a technique which is 84% reliable is 16% unreliable. The three authentication instances described above (fingerprint, smart card, password) which produce reliabilities of 86%, 75%, and 72% would have corresponding unreliabilities of (100 - 86)%, (100 - 75)% and (100 - 72)%, or 14%, 25%, and 28%, respectively. By multiplying these unreliabilities, we get a cumulative unreliability of 14% * 25% * 28% = .98% unreliability, which corresponds to a reliability of 99.02%.

In an additional mode of operation, additional factors and heuristics 530 may be applied within the authentication engine 215 to account for the interdependence of various authentication techniques. For example, if someone has unauthorized access to a particular home computer, they probably have access to the phone line at that address as well. Therefore, authenticating based on an originating phone number as well as upon the serial number of the authenticating system does not add much to the overall confidence in the authentication. However, knowledge based authentication is largely independent of token based authentication (*i.e.* if someone steals your cellular phone or keys, they are no more likely to know your PIN or password than if they hadn't).

Furthermore, different vendors or other authentication requestors may wish to weigh different aspects of the authentication differently. This may include the use of separate weighing factors or algorithms used in calculating the

reliability of individual instances as well as the use of different means to evaluate authentication events with multiple instances.

For instance, vendors for certain types of transactions, for instance corporate email systems, may desire to authenticate primarily based upon heuristics and other circumstantial data by default. Therefore, they may apply high weights to factors related to the metadata and other profile related information associated with the circumstances surrounding authentication events. This arrangement could be used to ease the burden on users during normal operating hours, by not requiring more from the user than that he be logged on to the correct machine during business hours. However, another vendor may weigh authentications coming from a particular technique most heavily, for instance fingerprint matching, because of a policy decision that such a technique is most suited to authentication for the particular vendor's purposes.

Such varying weights may be defined by the authentication requestor in generating the authentication request and sent to the trust engine 110 with the authentication request in one mode of operation. Such options could also be set as preferences during an initial enrollment process for the authentication requestor and stored within the authentication engine in another mode of operation.

Once the authentication engine 215 produces an authentication confidence level for the authentication data provided, this confidence level is used to complete the authentication request in STEP 1640, and this information is forwarded from the authentication engine 215 to the transaction engine 205 for inclusion in a message to the authentication requestor.

The process described above is merely exemplary, and those of skill in the art will recognize that the steps need not be performed in the order shown. Furthermore, certain steps, such as the evaluation of the reliability of each authentication instance provided, may be carried out in parallel with one another if circumstances permit.

In a further aspect of this invention, a method is provided to accommodate conditions when the authentication confidence level produced by the process described above fails to meet the required trust level of the vendor or other party requiring the authentication. In circumstances such as these where a gap exists between the level of confidence provided and the level of trust desired, the operator of the trust engine 110 is in a position to provide opportunities for one or both parties to provide alternate data or requirements in order to close this trust gap. This process will be referred to as "trust arbitrage" herein.

Trust arbitrage may take place within a framework of cryptographic authentication as described above with reference to FIGURES 10 and 11. As shown therein, a vendor or other party will request authentication of a particular user in association with a particular transaction. In one circumstance, the vendor simply requests an authentication, either positive or negative, and after receiving appropriate data from the user, the trust engine 110 will provide such a binary authentication. In circumstances such as these, the degree of confidence required in order to secure a positive authentication is determined based upon preferences set within the trust engine 110.

However, it is also possible that the vendor may request a particular level of trust in order to complete a particular transaction. This required level may be included with the authentication request (*e.g.* authenticate this user

to 96% confidence) or may be determined by the trust engine 110 based on other factors associated with the transaction (*i.e.* authenticate this user as appropriate for this transaction). One such factor might be the economic value of the transaction. For transactions which have greater economic value, a higher degree of trust may be required. Similarly, for transactions with high degrees of risk a high degree of trust may be required. Conversely, for transactions which are either of low risk or of low value, lower trust levels may be required by the vendor or other authentication requestor.

The process of trust arbitrage occurs between the steps of the trust engine 110 receiving the authentication data in STEP 1050 of FIGURE 10 and the return of an authentication result to the vendor in STEP 1055 of FIGURE 10. Between these steps, the process which leads to the evaluation of trust levels and the potential trust arbitrage occurs as shown in FIGURE 17. In circumstances where simple binary authentication is performed, the process shown in FIGURE 17 reduces to having the transaction engine 205 directly compare the authentication data provided with the enrollment data for the identified user as discussed above with reference to FIGURE 10, flagging any difference as a negative authentication.

As shown in FIGURE 17, the first step after receiving the data in STEP 1050 is for the transaction engine 205 to determine the trust level which is required for a positive authentication for this particular transaction in STEP 1710. This step may be performed by one of several different methods. The required trust level may be specified to the trust engine 110 by the authentication requestor at the time when the authentication request is made. The authentication requestor may also set a preference in advance which is stored within the depository 210 or other storage which is accessible by the transaction engine 205. This preference may then be read and used each time an authentication request is made by this authentication requestor. The preference may also be associated with a particular user as a security measure such that a particular level of trust is always required in order to authenticate that user, the user preference being stored in the depository 210 or other storage media accessible by the transaction engine 205. The required level may also be derived by the transaction engine 205 or authentication engine 215 based upon information provided in the authentication request, such as the value and risk level of the transaction to be authenticated.

In one mode of operation, a policy management module or other software which is used when generating the authentication request is used to specify the required degree of trust for the authentication of the transaction. This may be used to provide a series of rules to follow when assigning the required level of trust based upon the policies which are specified within the policy management module. One advantageous mode of operation is for such a module to be incorporated with the web server of a vendor in order to appropriately determine required level of trust for transactions initiated with the vendor's web server. In this way, transaction requests from users may be assigned a required trust level in accordance with the policies of the vendor and such information may be forwarded to the trust engine 110 along with the authentication request.

This required trust level correlates with the degree of certainty that the vendor wants to have that the individual authenticating is in fact who he identifies himself as. For example, if the transaction is one where the

vendor wants a fair degree of certainty because goods are changing hands, the vendor may require a trust level of 85%. For situation where the vendor is merely authenticating the user to allow him to view members only content or exercise privileges on a chat room, the downside risk may be small enough that the vendor requires only a 60% trust level. However, to enter into a production contract with a value of tens of thousands of dollars, the vendor may require a trust level of 99% or more.

This required trust level represents a metric to which the user must authenticate himself in order to complete the transaction. If the required trust level is 85% for example, the user must provide authentication to the trust engine 110 sufficient for the trust engine 110 to say with 85% confidence that the user is who they say they are. It is the balance between this required trust level and the authentication confidence level which produces either a positive authentication (to the satisfaction of the vendor) or a possibility of trust arbitrage.

As shown in FIGURE 17, after the transaction engine 205 receives the required trust level, it compares in STEP 1720 the required trust level to the authentication confidence level which the authentication engine 215 calculated for the current authentication (as discussed with reference to FIGURE 16). If the authentication confidence level is higher than the required trust level for the transaction in STEP 1730, then the process moves to STEP 1740 where a positive authentication for this transaction is produced by the transaction engine 205. A message to this effect will then be inserted into the authentication results and returned to the vendor by the transaction engine 205 as shown in STEP 1055 (see FIGURE 10).

However, if the authentication confidence level does not fulfill the required trust level in STEP 1730, then a confidence gap exists for the current authentication, and trust arbitrage is conducted in STEP 1750. Trust arbitrage is described more completely with reference to FIGURE 18 below. This process as described below takes place within the transaction engine 205 of the trust engine 110. Because no authentication or other cryptographic operations are needed to execute trust arbitrage (other than those required for the SSL communication between the transaction engine 205 and other components), the process may be performed outside the authentication engine 215. However, as will be discussed below, any reevaluation of authentication data or other cryptographic or authentication events will require the transaction engine 205 to resubmit the appropriate data to the authentication engine 215. Those of skill in the art will recognize that the trust arbitrage process could alternately be structured to take place partially or entirely within the authentication engine 215 itself.

As mentioned above, trust arbitrage is a process where the trust engine 110 mediates a negotiation between the vendor and user in an attempt to secure a positive authentication where appropriate. As shown in STEP 1805, the transaction engine 205 first determines whether or not the current situation is appropriate for trust arbitrage. This may be determined based upon the circumstances of the authentication, *e.g.* whether this authentication has already been through multiple cycles of arbitrage, as well as upon the preferences of either the vendor or user, as will be discussed further below.

In such circumstances where arbitrage is not possible, the process proceeds to STEP 1810 where the transaction engine 205 generates a negative authentication and then inserts it into the authentication results which

are sent to the vendor in STEP 1055 (see FIGURE 10). One limit which may be advantageously used to prevent authentications from pending indefinitely is to set a time-out period from the initial authentication request. In this way, any transaction which is not positively authenticated within the time limit is denied further arbitrage and negatively authenticated. Those of skill in the art will recognize that such a time limit may vary depending upon the circumstances of the transaction and the desires of the user and vendor. Limitations may also be placed upon the number of attempts that may be made at providing a successful authentication. Such limitations may be handled by an attempt limiter 535 as shown in FIGURE 5.

If arbitrage is not prohibited in STEP 1805, the transaction engine 205 will then engage in negotiation with one or both of the transacting parties. The transaction engine 205 may send a message to the user requesting some form of additional authentication in order to boost the authentication confidence level produced as shown in STEP 1820. In the simplest form, this may simply indicates that authentication was insufficient. A request to produce one or more additional authentication instances to improve the overall confidence level of the authentication may also be sent.

If the user provides some additional authentication instances in STEP 1825, then the transaction engine 205 adds these authentication instances to the authentication data for the transaction and forwards it to the authentication engine 215 as shown in STEP 1015 (see FIGURE 10), and the authentication is reevaluated based upon both the pre-existing authentication instances for this transaction and the newly provided authentication instances.

An additional type of authentication may be a request from the trust engine 110 to make some form of person-to-person contact between the trust engine 110 operator (or a trusted associate) and the user, for example, by phone call. This phone call or other non-computer authentication can be used to provide personal contact with the individual and also to conduct some form of questionnaire based authentication. This also may give the opportunity to verify an originating telephone number and potentially a voice analysis of the user when he calls in. Even if no additional authentication data can be provided, the additional context associated with the user's phone number may improve the reliability of the authentication context. Any revised data or circumstances based upon this phone call are fed into the trust engine 110 for use in consideration of the authentication request.

Additionally, in STEP 1820 the trust engine 110 may provide an opportunity for the user to purchase insurance, effectively buying a more confident authentication. The operator of the trust engine 110 may, at times, only want to make such an option available if the confidence level of the authentication is above a certain threshold to begin with. In effect, this user side insurance is a way for the trust engine 110 to vouch for the user when the authentication meets the normal required trust level of the trust engine 110 for authentication, but does not meet the required trust level of the vendor for this transaction. In this way, the user may still successfully authenticate to a very high level as may be required by the vendor, even though he only has authentication instances which produce confidence sufficient for the trust engine 110.

This function of the trust engine 110 allows the trust engine 110 to vouch for someone who is authenticated to the satisfaction of the trust engine 110, but not of the vendor. This is analogous to the function performed by a

notary in adding his signature to a document in order to indicate to someone reading the document at a later time that the person whose signature appears on the document is in fact the person who signed it. The signature of the notary testifies to the act of signing by the user. In the same way, the trust engine is providing an indication that the person transacting is who they say they are.

However, because the trust engine 110 is artificially boosting the level of confidence provided by the user, there is a greater risk to the trust engine 110 operator, since the user is not actually meeting the required trust level of the vendor. The cost of the insurance is designed to offset the risk of a false positive authentication to the trust engine 110 (who may be effectively notarizing the authentications of the user). The user pays the trust engine 110 operator to take the risk of authenticating to a higher level of confidence than has actually been provided.

Because such an insurance system allows someone to effectively buy a higher confidence rating from the trust engine 110, both vendors and users may wish to prevent the use of user side insurance in certain transactions. Vendors may wish to limit positive authentications to circumstances where they know that actual authentication data supports the degree of confidence which they require and so may indicate to the trust engine 110 that user side insurance is not to be allowed. Similarly, to protect his online identity, a user may wish to prevent the use of user side insurance on his account, or may wish to limit its use to situations where the authentication confidence level without the insurance is higher than a certain limit. This may be used as a security measure to prevent someone from overhearing a password or stealing a smart card and using them to falsely authenticate to a low level of confidence, and then purchasing insurance to produce a very high level of (false) confidence. These factors may be evaluated in determining whether user side insurance is allowed.

If user purchases insurance in STEP 1840, then the authentication confidence level is adjusted based upon the insurance purchased in STEP 1845, and the authentication confidence level and required trust level are again compared in STEP 1730 (see FIGURE 17). The process continues from there, and may lead to either a positive authentication in STEP 1740 (see FIGURE 17), or back into the trust arbitrage process in STEP 1750 for either further arbitrage (if allowed) or a negative authentication in STEP 1810 if further arbitrage is prohibited.

In addition to sending a message to the user in STEP 1820, the transaction engine 205 may also send a message to the vendor in STEP 1830 which indicates that a pending authentication is currently below the required trust level. The message may also offer various options on how to proceed to the vendor. One of these options is to simply inform the vendor of what the current authentication confidence level is and ask if the vendor wishes to maintain their current unfulfilled required trust level. This may be beneficial because in some cases, the vendor may have independent means for authenticating the transaction or may have been using a default set of requirements which generally result in a higher required level being initially specified than is actually needed for the particular transaction at hand.

For instance, it may be standard practice that all incoming purchase order transactions with the vendor are expected to meet a 98% trust level. However, if an order was recently discussed by phone between the vendor and a long-standing customer, and immediately thereafter the transaction is authenticated, but only to a 93% confidence

level, the vendor may wish to simply lower the acceptance threshold for this transaction, because the phone call effectively provides additional authentication to the vendor. In certain circumstances, the vendor may be willing to lower their required trust level, but not all the way to the level of the current authentication confidence. For instance, the vendor in the above example might consider that the phone call prior to the order might merit a 4% reduction in the degree of trust needed; however, this is still greater than the 93% confidence produced by the user.

If the vendor does adjust their required trust level in STEP 1835, then the authentication confidence level produced by the authentication and the required trust level are compared in STEP 1730 (see FIGURE 17). If the confidence level now exceeds the required trust level, a positive authentication may be generated in the transaction engine 205 in STEP 1740 (see FIGURE 17). If not, further arbitrage may be attempted as discussed above if it is permitted.

In addition to requesting an adjustment to the required trust level, the transaction engine 205 may also offer vendor side insurance to the vendor requesting the authentication. This insurance serves a similar purpose to that described above for the user side insurance. Here, however, rather than the cost corresponding to the risk being taken by the trust engine 110 in authenticating above the actual authentication confidence level produced, the cost of the insurance corresponds to the risk being taken by the vendor in accepting a lower trust level in the authentication.

Instead of just lowering their actual required trust level, the vendor has the option of purchasing insurance to protect itself from the additional risk associated with a lower level of trust in the authentication of the user. As described above, it may be advantageous for the vendor to only consider purchasing such insurance to cover the trust gap in conditions where the existing authentication is already above a certain threshold.

The availability of such vendor side insurance allows the vendor the option to either: lower his trust requirement directly at no additional cost to himself, bearing the risk of a false authentication himself (based on the lower trust level required); or, buying insurance for the trust gap between the authentication confidence level and his requirement, with the trust engine 110 operator bearing the risk of the lower confidence level which has been provided. By purchasing the insurance, the vendor effectively keeps his high trust level requirement, because the risk of a false authentication is shifted to the trust engine 110 operator.

If the vendor purchases insurance in STEP 1840, the authentication confidence level and required trust levels are compared in STEP 1730 (see FIGURE 17), and the process continues as described above.

Note that it is also possible that both the user and the vendor respond to messages from the trust engine 110. Those of skill in the art will recognize that there are multiple ways in which such situations can be handled. One advantageous mode of handling the possibility of multiple responses is simply to treat the responses in a first-come, first-served manner. For example, if the vendor responds with a lowered required trust level and immediately thereafter the user also purchases insurance to raise his authentication level, the authentication is first reevaluated based upon the lowered trust requirement from the vendor. If the authentication is now positive, the user's insurance purchase is ignored. In another advantageous mode of operation, the user might only be charged for the level of

insurance required to meet the new, lowered trust requirement of the vendor (if a trust gap remained even with the lowered vendor trust requirement).

If no response from either party is received during the trust arbitrage process at STEP 1850 within the time limit set for the authentication, the arbitrage is reevaluated in STEP 1805. This effectively begins the arbitrage process again. If the time limit was final or other circumstances prevent further arbitrage in STEP 1805, a negative authentication is generated by the transaction engine 205 in STEP 1810 and returned to the vendor in STEP 1055 (see FIGURE 10). If not, new messages may be sent to the user and vendor, and the process may be repeated as desired.

Note that for certain types of transactions, for instance, digitally signing documents which are not part of a transaction, there may not necessarily be a vendor or other third party; therefore the transaction is primarily between the user and the trust engine 110. In circumstances such as these, the trust engine 110 will have its own required trust level which must be satisfied in order to generate a positive authentication. However, in such circumstances, it will often not be desirable for the trust engine 110 to offer insurance to the user in order for him to raise the confidence of his own signature.

The process described above and shown in FIGURES 16 - 18 may be carried out using various communications modes as described above with reference to the trust engine 110. For instance, the messages may be web-based and sent using SSL connections between the trust engine 110 and applets downloaded in real time to browsers running on the user or vendor systems. In an alternate mode of operation, certain dedicated applications may be in use by the user and vendor which facilitate such arbitrage and insurance transactions. In another alternate mode of operation, secure email operations may be used to mediate the arbitrage described above, thereby allowing deferred evaluations and batch processing of authentications. Those of skill in the art will recognize that different communications modes may be used as are appropriate for the circumstances and authentication requirements of the vendor.

The following description with reference to FIGURE 19 describes a sample transaction which integrates the various aspects of the present invention as described above. This example illustrates the overall process between a user and a vendor as mediates by the trust engine 110. Although the various steps and components as described in detail above may be used to carry out the following transaction, the process illustrated focuses on the interaction between the trust engine 110, user and vendor.

The transaction begins when the user, while viewing web pages online, fills out an order form on the web site of the vendor in STEP 1900. The user wishes to submit this order form to the vendor, signed with his digital signature. In order to do this, the user submits the order form with his request for a signature to the trust engine 110 in STEP 1905. The user will also provide authentication data which will be used as described above to authenticate his identity.

In STEP 1910 the authentication data is compared to the enrollment data by the trust engine 110 as discussed above, and if a positive authentication is produced, the hash of the order form, signed with the private key of the user, is forwarded to the vendor along with the order form itself.

The vendor receives the signed form in STEP 1915, and then the vendor will generate an invoice or other contract related to the purchase to be made in STEP 1920. This contract is sent back to the user with a request for a signature in STEP 1925. The vendor also sends an authentication request for this contract transaction to the trust engine 110 in STEP 1930 including a hash of the contract which will be signed by both parties. To allow the contract to be digitally signed by both parties, the vendor also includes authentication data for itself so that the vendor's signature upon the contract can later be verified if necessary.

As discussed above, the trust engine 110 then verifies the authentication data provided by the vendor to confirm the vendor's identity, and if the data produces a positive authentication in STEP 1935, continues with STEP 1955 when the data is received from the user. If the vendor's authentication data does not match the enrollment data of the vendor to the desired degree, a message is returned to the vendor requesting further authentication. Trust arbitrage may be performed here if necessary, as described above, in order for the vendor to successfully authenticate itself to the trust engine 110.

When the user receives the contract in STEP 1940, he reviews it, generates authentication data to sign it if it is acceptable in STEP 1945, and then sends a hash of the contract and his authentication data to the trust engine 110 in STEP 1950. The trust engine 110 verifies the authentication data in STEP 1955 and if the authentication is good, proceeds to process the contract as described below. As discussed above with reference to FIGURES 17 and 18, trust arbitrage may be performed as appropriate to close any trust gap which exists between the authentication confidence level and the required authentication level for the transaction.

The trust engine 110 signs the hash of the contract with the user's private key, and sends this signed hash to the vendor in STEP 1960, signing the complete message on its own behalf, *i.e.* including a hash of the complete message (including the user's signature) encrypted with the private key 510 of the trust engine 110. This message is received by the vendor in STEP 1965. The message represents a signed contract (hash of contract encrypted using user's private key) and a receipt from the trust engine 110 (the hash of the message including the signed contract, encrypted using the trust engine 110's private key).

The trust engine 110 similarly prepares a hash of the contract with the vendor's private key in STEP 1970, and forwards this to the user, signed by the trust engine 110. In this way, the user also receives a copy of the contract, signed by the vendor, as well as a receipt, signed by the trust engine 110, for delivery of the signed contract in STEP 1975.

In addition to the foregoing, an additional aspect of the invention provides a cryptographic Service Provider Module (SPM) which may be available to a client side application as a means to access functions provided by the trust engine 110 described above. One advantageous way to provide such a service is for the cryptographic SPM is to mediate communications between a third party Application Programming Interface (API) and a trust engine 110 which is accessible via a network or other remote connection. A sample cryptographic SPM is described below with reference to FIGURE 20.

For example, on a typical system, a number of API's are available to programmers. Each API provides a set of function calls which may be made by an application 2000 running upon the system. Examples of API's which provide programming interfaces suitable for cryptographic functions, authentication functions, and other security function include the Cryptographic API (CAPI) 2010 provided by Microsoft with its Windows operating systems, and the Common Data Security Architecture (CDSA), sponsored by IBM, Intel and other members of the Open Group. CAPI will be used as an exemplary security API in the discussion that follows. However, the cryptographic SPM described could be used with CDSA or other security API's as are known in the art.

This API is used by a user system 105 or vendor system 120 when a call is made for a cryptographic function. Included among these functions may be requests associated with performing various cryptographic operations, such as encrypting a document with a particular key, signing a document, requesting a digital certificate, verifying a signature upon a signed document, and such other cryptographic functions as are described above or known to those of skill in the art.

Such cryptographic functions are normally performed locally to the system upon which CAPI 2010 is located. This is because generally the functions called require the use of either resources of the local user system 105, such as a fingerprint reader, or software functions which are programmed using libraries which are executed on the local machine. Access to these local resources is normally provided by one or more Service Provider Modules (SPM's) 2015, 2020 as referred to above which provide resources with which the cryptographic functions are carried out. Such SPM's may include software libraries 2015 to perform encrypting or decrypting operations, or drivers and applications 2020 which are capable of accessing specialized hardware 2025, such as biometric scanning devices. In much the way that CAPI 2010 provides functions which may be used by an application 2000 of the system 105, the SPM's 2015, 2020 provide CAPI with access to the lower level functions and resources associated with the available services upon the system.

In accordance with the invention, it is possible to provide a cryptographic SPM 2030 which is capable of accessing the cryptographic functions provided by the trust engine 110 and making these functions available to an application 2000 through CAPI 2010. Unlike embodiments where CAPI 2010 is only able to access resources which are locally available through SPM's 2015, 2020, a cryptographic SPM 2030 as described herein would be able to submit requests for cryptographic operations to a remotely-located, network-accessible trust engine 110 in order to perform the operations desired.

For instance, if an application 2000 has a need for a cryptographic operation, such as signing a document, the application 2000 makes a function call to the appropriate CAPI 2010 function. CAPI 2010 in turn will execute this function, making use of the resources which are made available to it by the SPM's 2015, 2020 and the cryptographic SPM 2030. In the case of a digital signature function, the cryptographic SPM 2030 will generate an appropriate request which will be sent to the trust engine 110 across the communication link 125.

The operations which occur between the cryptographic SPM 2030 and the trust engine 110 are the same operations that would be possible between any other system and the trust engine 110. However, these functions are

effectively made available to a user system 105 through CAPI 2010 such that they appear to be locally available upon the user system 105 itself. However, unlike ordinary SPM's 2015, 2020, the functions are being carried out on the remote trust engine 110 and the results relayed to the cryptographic SPM 2030 in response to appropriate requests across the communication link 125.

This cryptographic SPM 2030 makes a number of operations available to the user system 105 or a vendor system 120 which might not otherwise be available. These functions include without limitation: encryption and decryption of documents; issuance of digital certificates; digital signing of documents; verification of digital signatures; and such other operations as will be apparent to those of skill in the art.

Additionally, other combinations, admissions, substitutions and modifications will be apparent to the skilled artisan in view of the disclosure herein. Accordingly, the present invention is not intended to be limited by the reaction of the preferred embodiments but is to be defined by a reference to the appended claims.

WHAT IS CLAIMED IS:

1. A method of performing graded authentication of a user wherein the method obtains and evaluates circumstantial data associated with an authentication attempt, the method comprising:

obtaining user data from a user during an authentication attempt;

obtaining circumstantial data associated with the authentication attempt; and

determining a level of trust associated with the authentication attempt based on the comparison of the circumstantial data with previously stored data.

2. A method as in Claim 1 wherein said act of determining the level of trust associated with the authentication attempt is further based on the comparison of the user data with previously stored data.

3. A method as in Claim 1 wherein the previously stored data comprises in part a historical record of the circumstantial data obtained during previous successful authentications for the user.

4. A method as in Claim 1 wherein the circumstantial data comprises an identification associated with the system from which the user data is obtained.

5. A method as in Claim 4 wherein the identification comprises a processor serial number.

6. A method as in Claim 1 wherein the circumstantial data comprises an identification associated with the network location of the system from which the user data is obtained.

7. A method as in Claim 6 wherein the identification comprises an IP address.

8. A method as in Claim 1 wherein the circumstantial data comprises a time stamp associated with the time at which the user data was obtained.

9. A method as in Claim 1 wherein the circumstantial data comprises an identifier representing the medium over which the user data is obtained.

10. A method as in Claim 1 wherein the circumstantial data represents more than one circumstantial aspect of the authentication attempt.

11. A method as in Claim 10 wherein the circumstantial data comprises a time stamp associated with the time at which the user data was obtained and an identification associated with the network location of the system from which the user data is obtained.

12. A system for graded authentication comprising user data obtained from a user during at least one previously successful authentication attempt, circumstantial data associated with the at least one previously successful authentication attempt, and a trust engine which generates a level of trust associated with a current authentication attempt based on the comparison of circumstantial data associated with the current authentication attempt with the circumstantial data associated with the at least one previously successful authentication attempt.

13. A system as in Claim 12 wherein the user data represents an intent to assent to a transaction.

14. A system as in Claim 12 wherein the circumstantial data represents a network address associated with the authentication attempt.

15. A system as in Claim 12 wherein the circumstantial data represents a time stamp associated with the authentication attempt.

16. A system as in Claim 12 wherein the level of trust is also based upon the comparison of the user data to previously stored user data.

17. A method for authenticating a user comprising:

obtaining user data associated with an authentication operation;

obtaining metadata related to the authentication operation;

comparing the metadata with previously stored data; and

determining a level of trust associated with the authentication operation.

18. A method as in Claim 17 wherein the user data associated with the authentication operation represents the intent of a user to assent to a transaction.

19. A method as in Claim 18 wherein the metadata related to the authentication operation are made available at a later time to contest a repudiation of the authentication operation by the user.

20. A method as in Claim 17 wherein the act of determining a level of trust associated with the authentication operation comprises assigning a percentage to the authentication operation which represents the degree of confidence in the authentication of the user.

21. A method as in Claim 17 further comprising determining an intermediate level of trust associated with the metadata based upon the comparison of the metadata with previously stored data.

22. A method as in Claim 21 wherein the act of determining an intermediate level of trust comprises assigning a percentage to the metadata which represents the degree of correspondence between the metadata and the previously stored data.

23. A method as in Claim 22 wherein the act of determining a level of trust associated with the authentication operation comprises multiplying the percentage representing the intermediate level of trust and a percentage which represents a degree of correspondence between the user data and the previously stored data.

24. A method as in Claim 23 wherein additional factors are used to weight the percentage representing the intermediate level of trust and the percentage which represents the degree of correspondence between the user data and the previously stored data differently.

25. A method for authenticating a user comprising:

obtaining user data associated with an authentication operation;

obtaining metadata related to the authentication operation; and

determining a level of trust associated with the authentication operation based on the metadata.

26. A method as in Claim 25 wherein the act of determining a level of trust compares the metadata with previously stored data.

27. A method as in Claim 25 further comprising providing the user with a plurality of authentication techniques which may be used to generate the user data.

28. A method as in Claim 27 wherein the user data is generated using more than one of the plurality of authentication techniques.

29. A method as in Claim 28 wherein the user data generated using each authentication technique is compared with a different portion of a set of previously stored data.

30. A method for grading an authentication operation that relies on a variable set of authentication techniques to obtain authentication data, the method comprising:

defining the reliability of a set of authentication techniques that may be used in an authentication operation;

receiving authentication data during an authentication operation, said authentication data generated using a subset of the authentication techniques;

determining the acceptability of the authentication data generated by each of the subset of authentication techniques; and

defining the level of trust of the authentication operation based upon the acceptability of the authentication data and based upon the reliability of the authentication techniques used in generating the authentication data.

31. The method of Claim 30 wherein the act of determining the acceptability involves comparing the authentication data with previously stored enrollment data.

32. The method of Claim 31 wherein the act of defining the reliability of a set of authentication techniques is based upon a set of circumstances associated with the previously stored enrollment data.

33. The method of Claim 30 wherein the act of defining the reliability of a set of authentication techniques is based upon the circumstances associated with the generation of the authentication data.

34. An apparatus for evaluating an authentication attempt comprising:

reliability data associated with a set of authentication techniques that may be used in an authentication attempt;

a plurality of authentication instances generated using a subset of the authentication techniques; and

a trust engine which determines a level of match associated with each authentication instance and assigns a level of trust for the authentication attempt based upon the level of match associated with each authentication instance and the reliability of the technique used in each authentication instance.

35. An apparatus as in Claim 34 further comprising a required level of trust associated with the authentication attempt.

36. An apparatus as in Claim 35 wherein the trust engine further assigns a result for the authentication based upon a comparison of the level of trust associated with the authentication attempt and the required level of trust.

37. An apparatus as in Claim 35 wherein the required level of trust is determined by the trust engine based upon the risk associated with a successful authentication.

38. A method for grading an authentication attempt comprising:

defining the reliability of a set of authentication techniques that may be used in an authentication attempt;

receiving a plurality of authentication instances generated using a subset of the authentication techniques;

determining a level of match associated with each authentication instance; and

defining a level of trust of the authentication attempt based upon the level of match associated with each authentication instance and based upon the reliability of the technique used in each authentication instance.

39. A method as in Claim 38 further comprising defining a required level of trust for the authentication attempt.

40. A method as in Claim 39 further comprising assigning an authentication result to the authentication attempt based on a comparison of the level of trust for the authentication attempt and a required level of trust of the authentication attempt.

41. A method as in Claim 39 wherein the required level of trust for the authentication attempt is based upon the value associated with a successful authentication.

42. A method as in Claim 39 wherein the required level of trust for the authentication attempt is based upon the risk associated with a successful authentication.

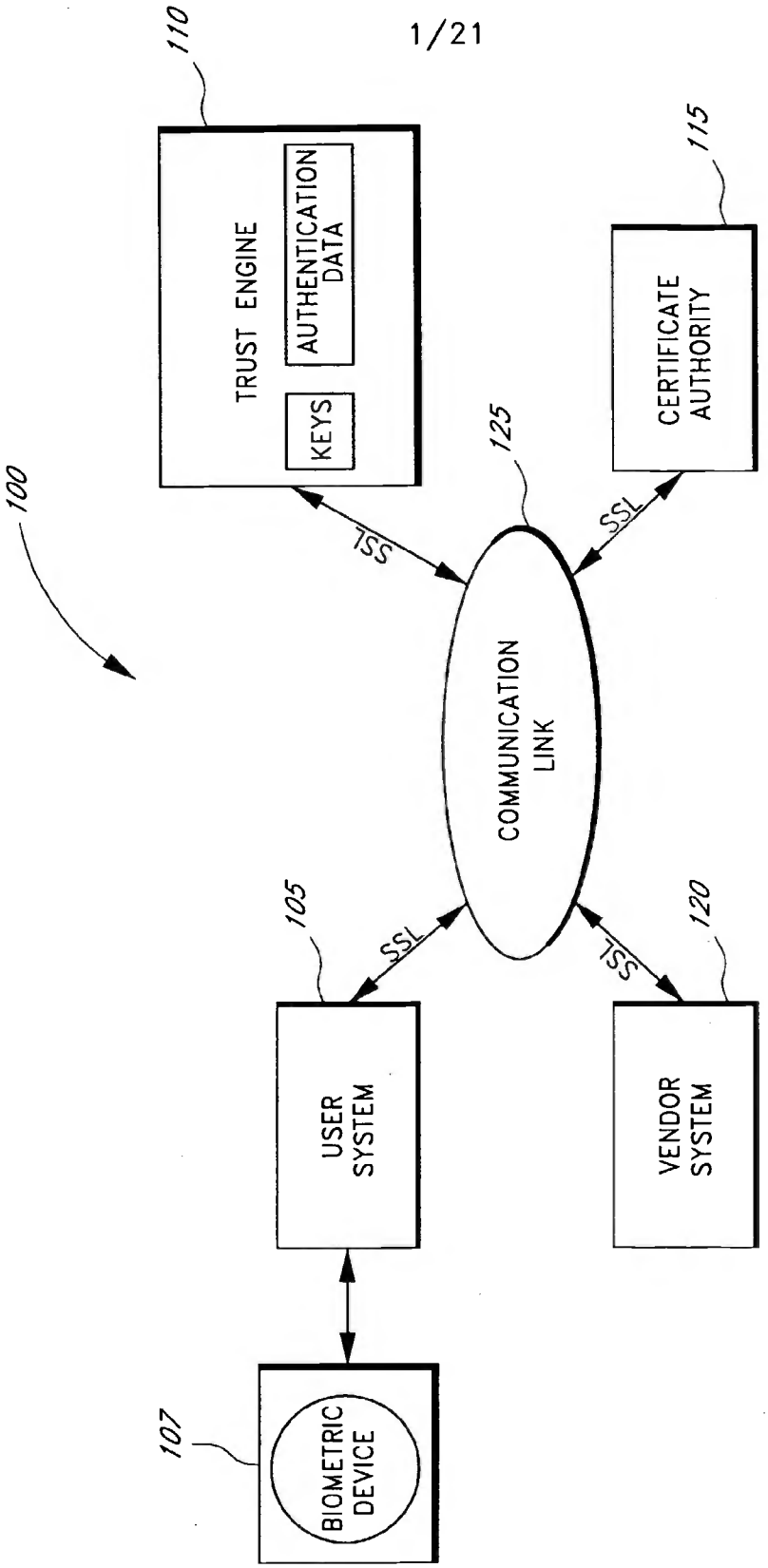

110
TRUST ENGINE
AUTHENTICATION DATA
KEYS
115
CERTIFICATE AUTHORITY
100
125
SSL
SSL
COMMUNICATION LINK
105
SSL
SSL
120
USER SYSTEM
VENDOR SYSTEM
107
BIOMETRIC DEVICE

FIG. 1

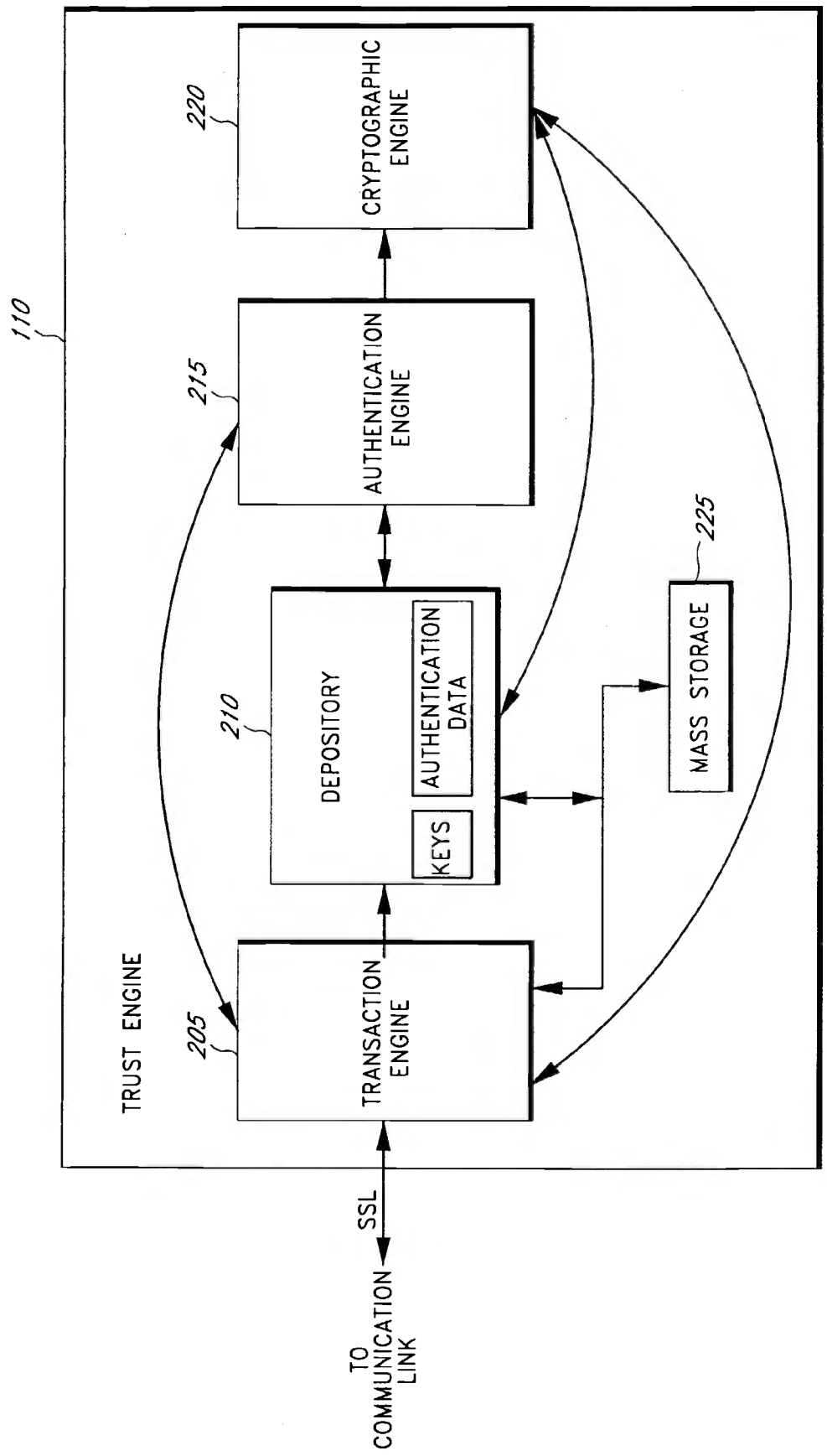

110
TRUST ENGINE
220
CRYPTOGRAPHIC ENGINE
215
AUTHENTICATION ENGINE
210
DEPOSITORY
AUTHENTICATION DATA
KEYS
225
MASS STORAGE
205
TRANSACTION ENGINE
SSL
TO COMMUNICATION LINK

FIG. 2

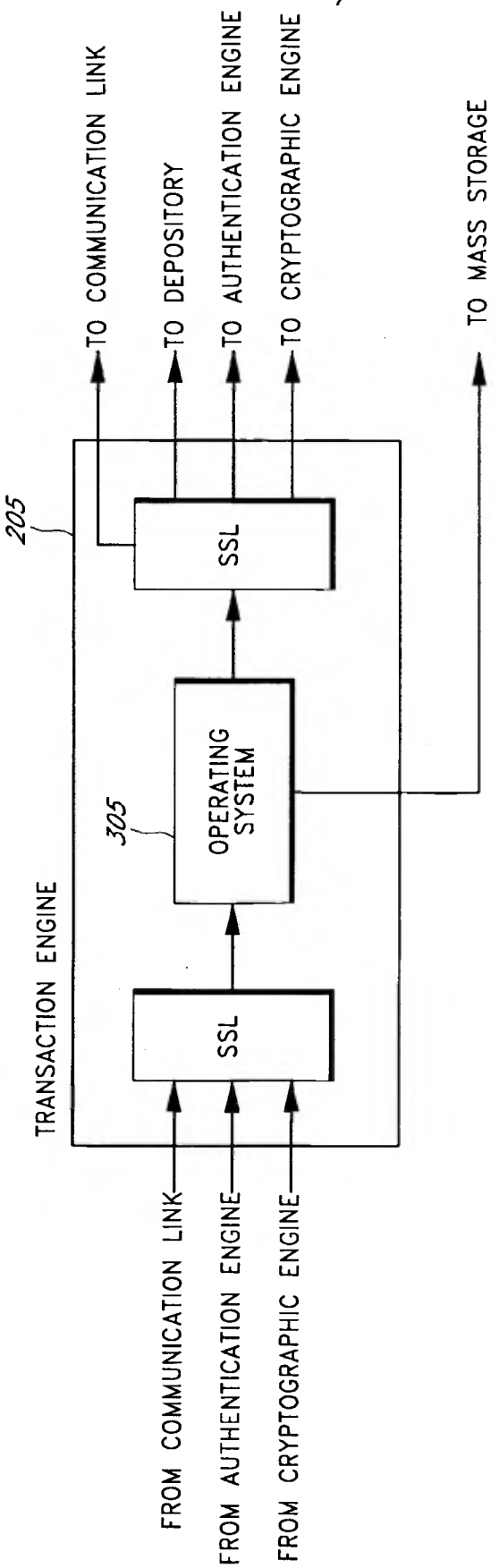

TO COMMUNICATION LINK
TO DEPOSITORY
TO AUTHENTICATION ENGINE
TO CRYPTOGRAPHIC ENGINE
TO MASS STORAGE
205
SSL
OPERATING SYSTEM
305
SSL
TRANSACTION ENGINE
FROM COMMUNICATION LINK
FROM AUTHENTICATION ENGINE
FROM CRYPTOGRAPHIC ENGINE

FIG. 3

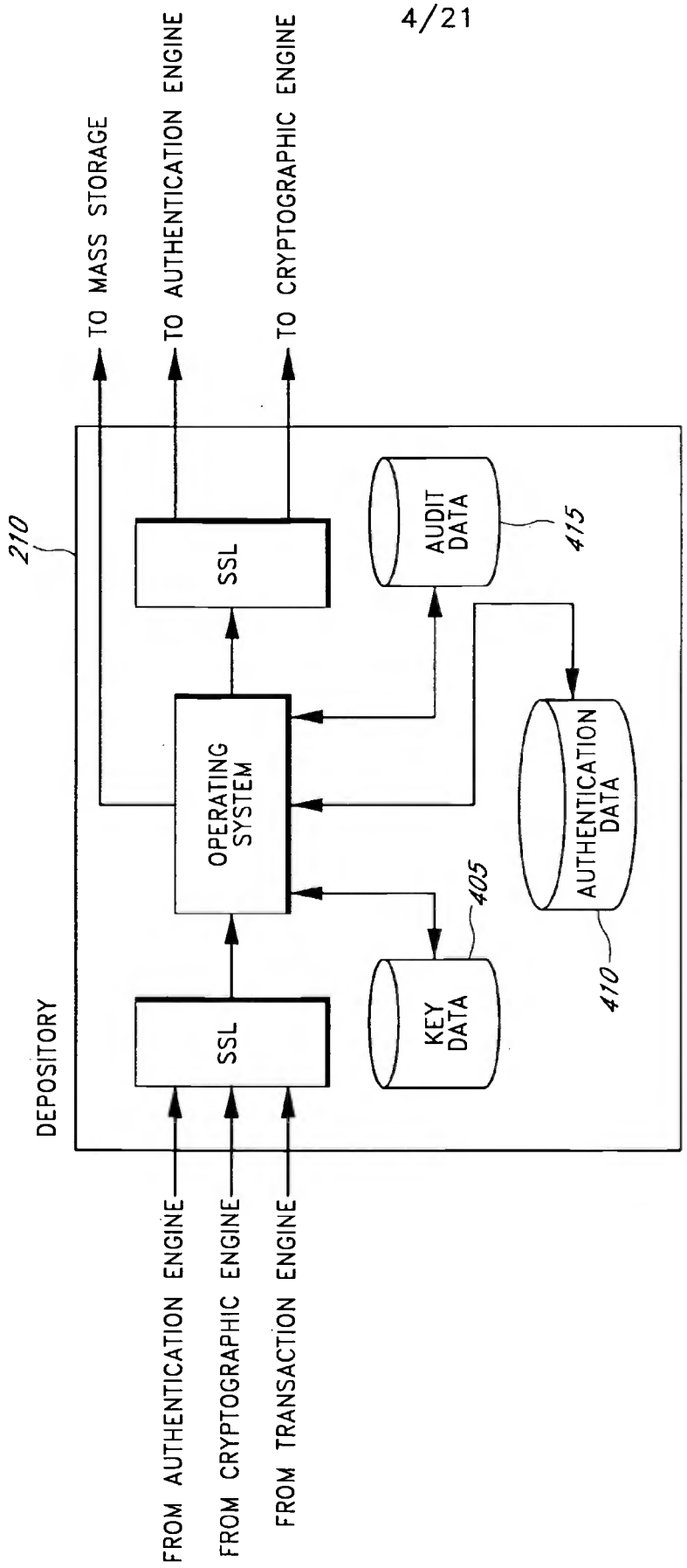

TO MASS STORAGE
TO AUTHENTICATION ENGINE
TO CRYPTOGRAPHIC ENGINE
210
SSL
AUDIT DATA
415
OPERATING SYSTEM
AUTHENTICATION DATA
405
KEY DATA
410
SSL
DEPOSITORY
FROM AUTHENTICATION ENGINE
FROM CRYPTOGRAPHIC ENGINE
FROM TRANSACTION ENGINE

FIG. 4

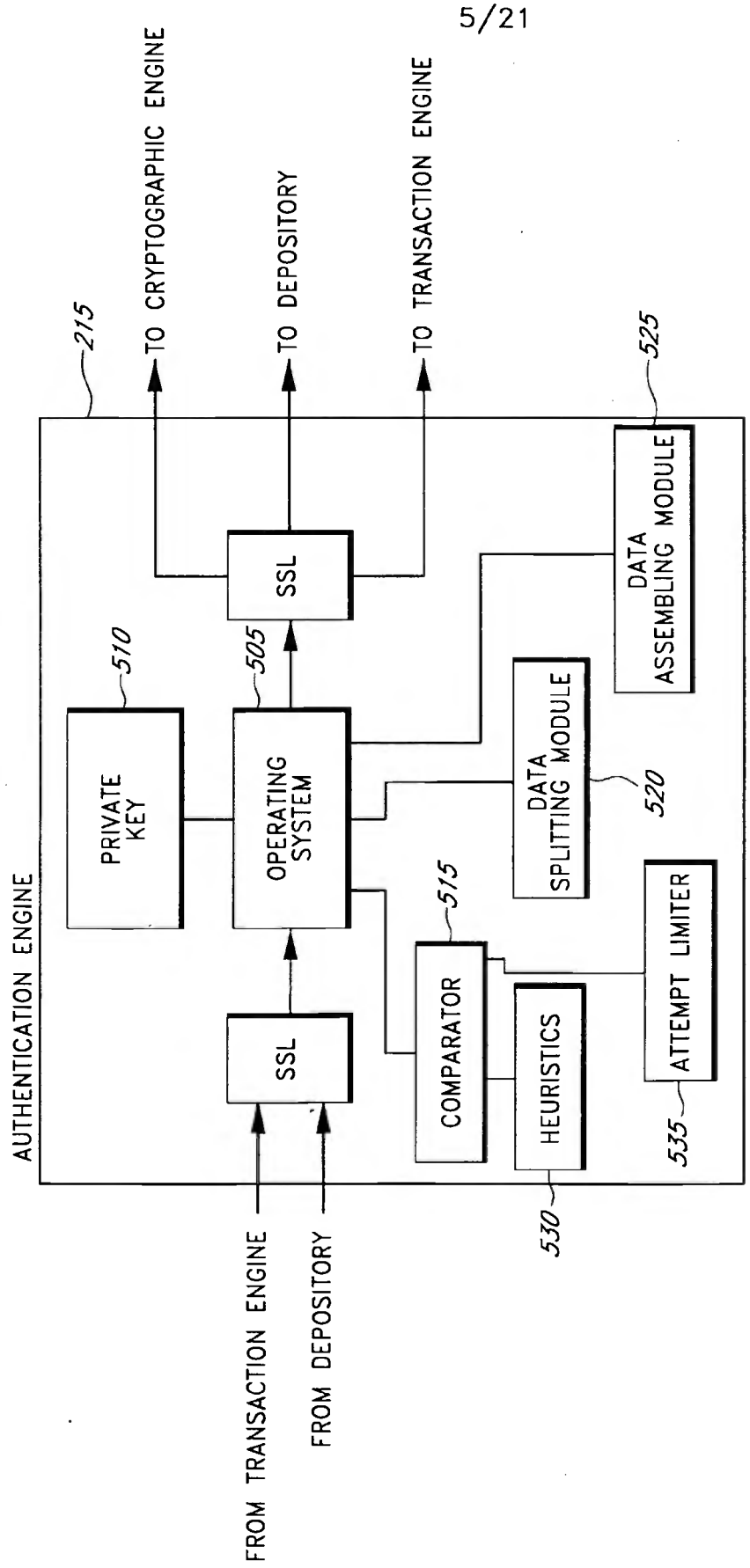

TO CRYPTOGRAPHIC ENGINE
TO DEPOSITORY
TO TRANSACTION ENGINE
215
525
AUTHENTICATION ENGINE
SSL
DATA ASSEMBLING MODULE
510
505
PRIVATE KEY
OPERATING SYSTEM
DATA SPLITTING MODULE
520
515
COMPARATOR
HEURISTICS
ATTEMPT LIMITER
535
530
SSL
FROM TRANSACTION ENGINE
FROM DEPOSITORY

FIG. 5

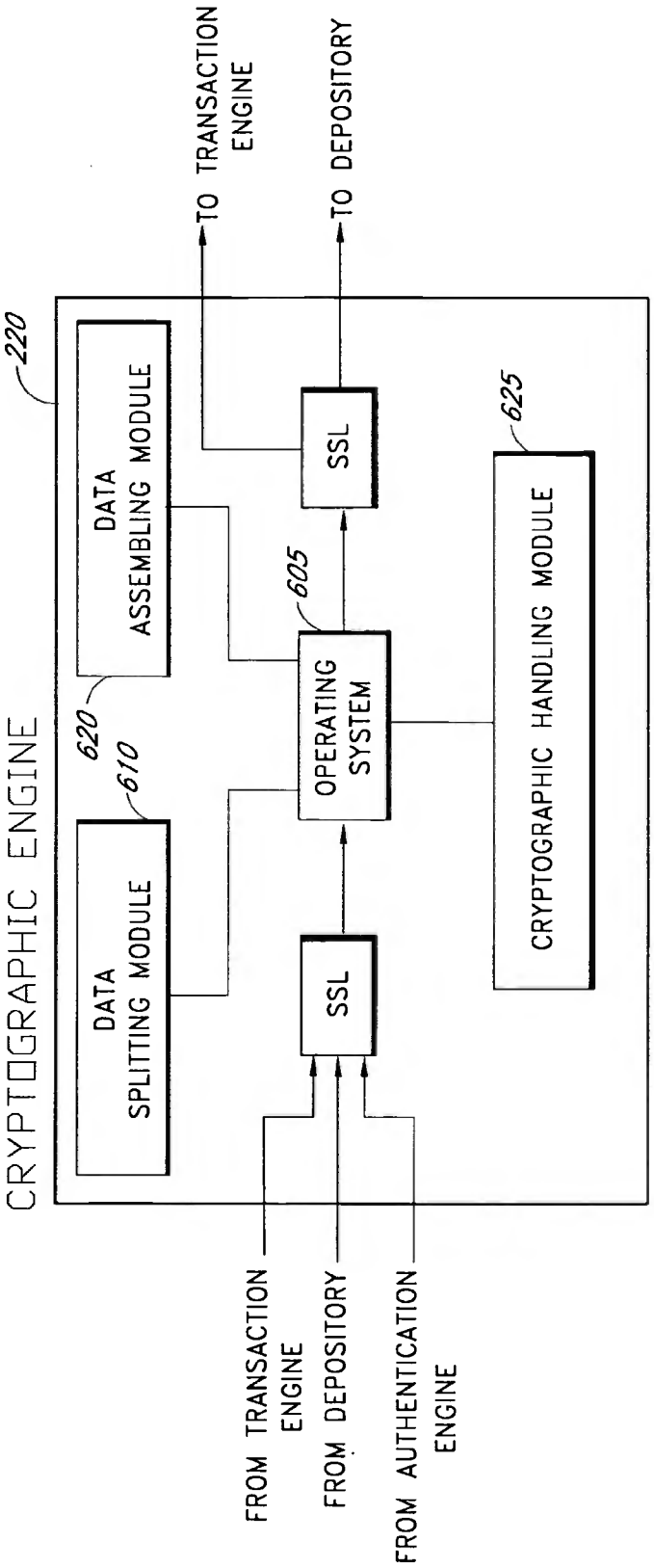

TO TRANSACTION ENGINE
TO DEPOSITORY
220
CRYPTOGRAPHIC ENGINE
DATA ASSEMBLING MODULE
620
610
DATA SPLITTING MODULE
605
OPERATING SYSTEM
SSL
SSL
625
CRYPTOGRAPHIC HANDLING MODULE
FROM TRANSACTION ENGINE
FROM DEPOSITORY
FROM AUTHENTICATION ENGINE

FIG. 6

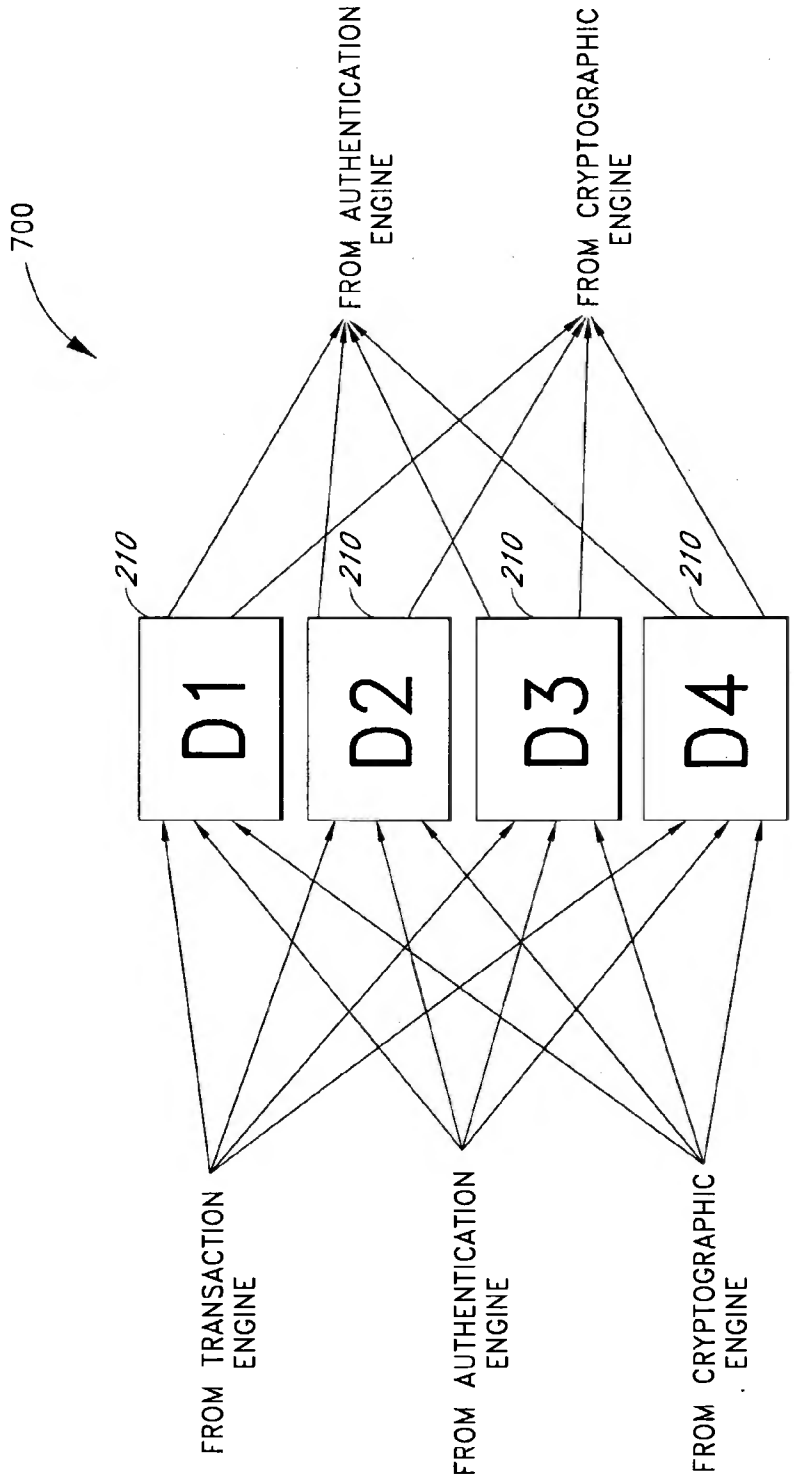

700
FROM AUTHENTICATION ENGINE
FROM CRYPTOGRAPHIC ENGINE
210
210
210
210
D1
D2
D3
D4
FROM TRANSACTION ENGINE
FROM AUTHENTICATION ENGINE
FROM CRYPTOGRAPHIC ENGINE

FIG. 7

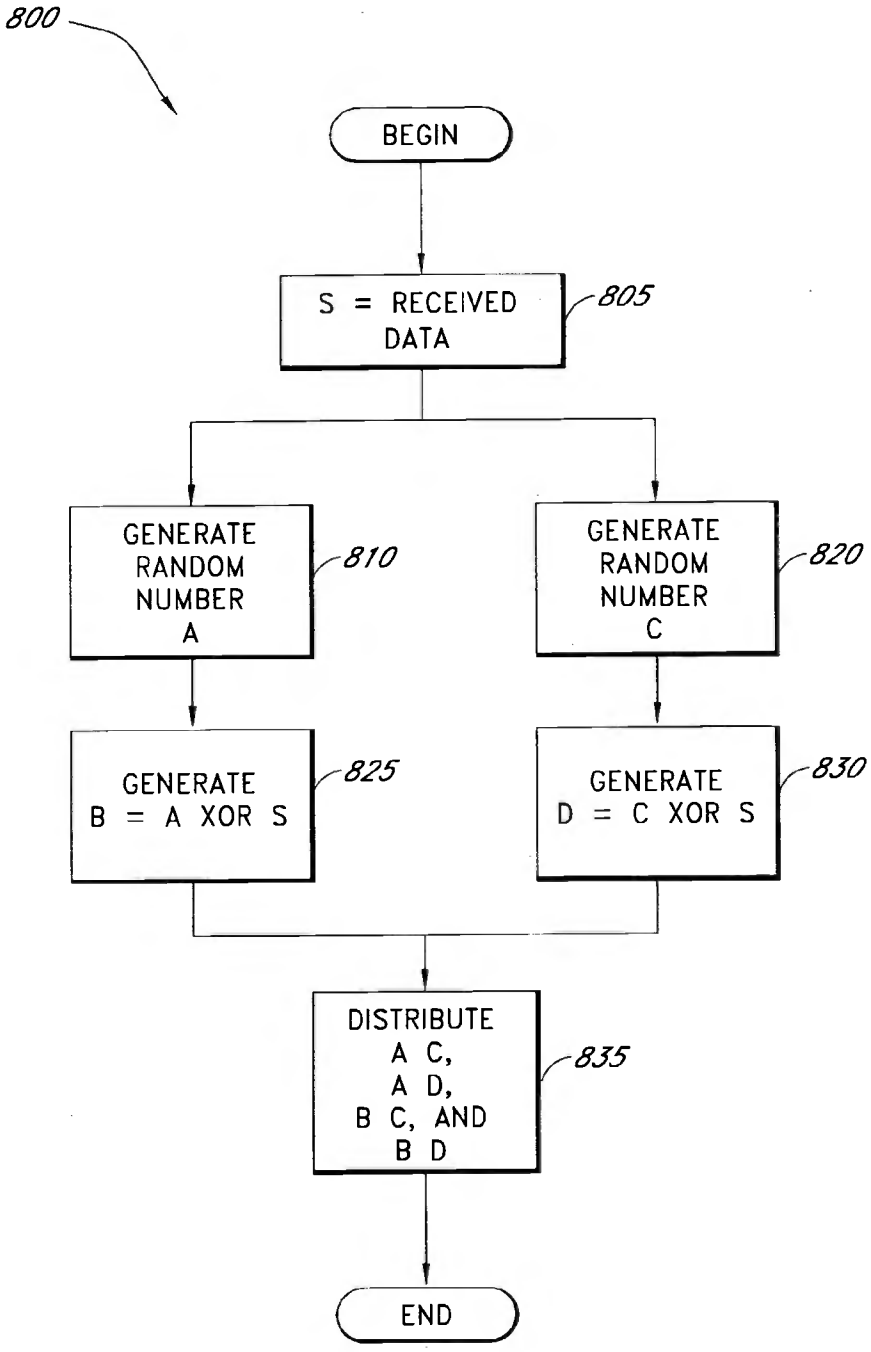
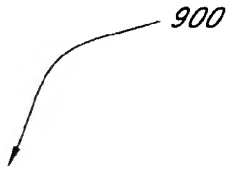

800
BEGIN
S = RECEIVED DATA
805
GENERATE RANDOM NUMBER A
810
GENERATE RANDOM NUMBER C
820
GENERATE B = A XOR S
825
GENERATE D = C XOR S
830
DISTRIBUTE A C, A D, B C, AND B D
835
END

FIG. 8



	ENROLLMENT DATA FLOW			
	SEND	RECEIVE	SSL	ACTION
905	USER	TRANSACTION ENGINE(TE)	1/2	TRANSMIT ENROLLMENT AUTHENTICATION DATA (B) AND THE USER ID (UID) ENCRYPTED WITH THE PUBLIC KEY OF THE AUTHENTICATION ENGINE (AE) AS (PUB_AE(UID,B))
915	TE	AE	FULL	FORWARD TRANSMISSION
920				AE DECRYPTS AND SPLITS FORWARDED DATA
925	AE	THE Xth DEPOSITORY (DX)	FULL	STORE RESPECTIVE PORTION OF DATA
	WHEN DIGITAL CERTIFICATE REQUESTED			
930	AE	CRYPTOGRAPHIC ENGINE (CE)	FULL	REQUEST KEY GENERATION
935				CE GENERATES AND SPLITS KEY
945	CE	TE	FULL	TRANSMIT REQUEST FOR DIGITAL CERTIFICATE
950	TE	CERTIFICATION AUTHORITY (CA)	1/2	TRANSMIT REQUEST
955	CA	TE	1/2	TRANSMIT DIGITAL CERTIFICATE
960	TE	USER	1/2	TRANSMIT DIGITAL CERTIFICATE
	TE	MS	FULL	STORE DIGITAL CERTIFICATE
965	TE	DX	FULL	STORE RESPECTIVE PORTION OF KEY

FIG. 9A

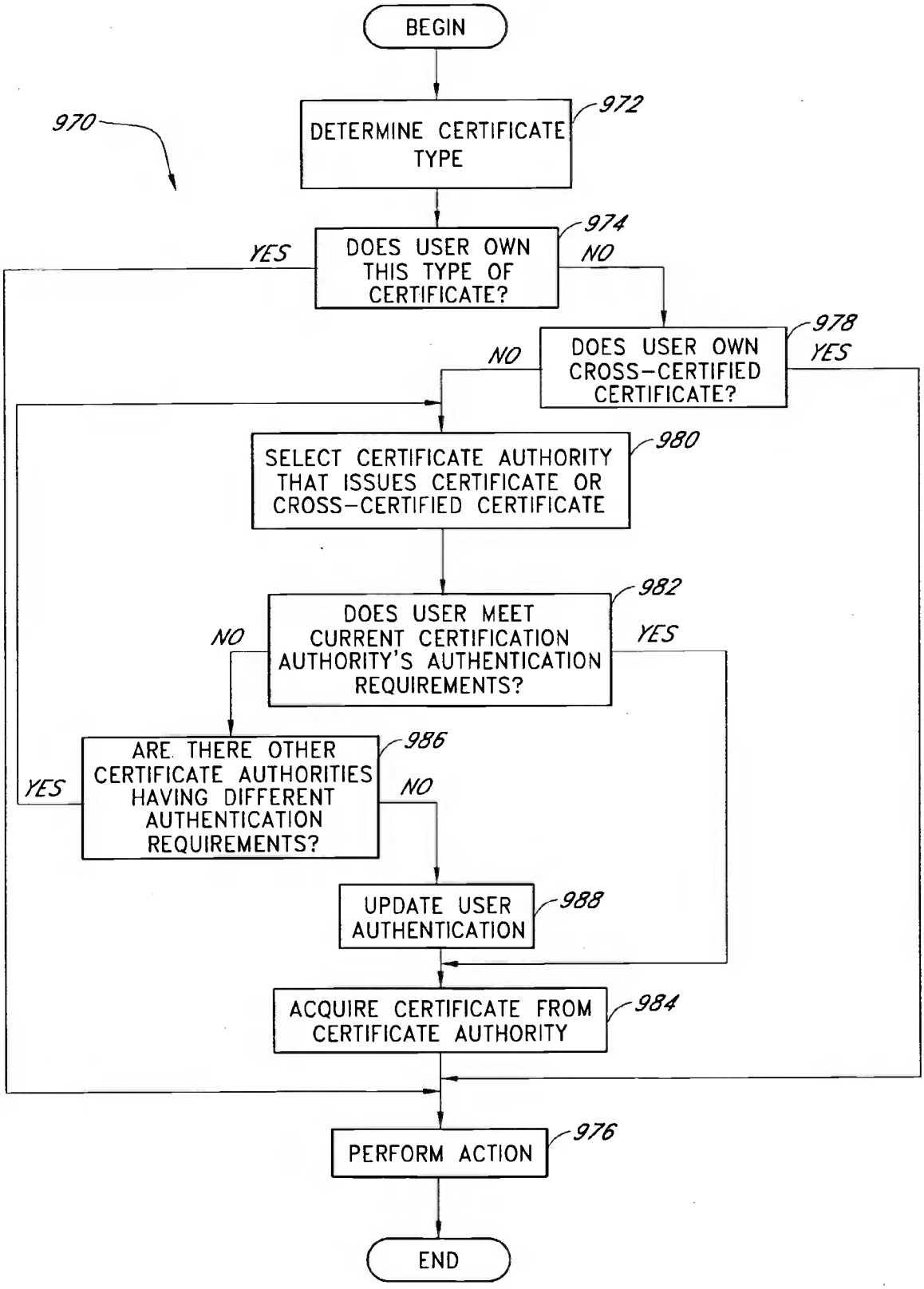
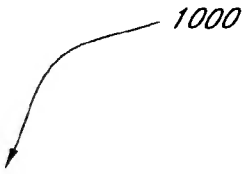

BEGIN
970
DETERMINE CERTIFICATE TYPE
972
DOES USER OWN THIS TYPE OF CERTIFICATE?
974
YES
NO
DOES USER OWN CROSS-CERTIFIED CERTIFICATE?
978
NO
YES
SELECT CERTIFICATE AUTHORITY THAT ISSUES CERTIFICATE OR CROSS-CERTIFIED CERTIFICATE
980
DOES USER MEET CURRENT CERTIFICATION AUTHORITY'S AUTHENTICATION REQUIREMENTS?
982
NO
YES
ARE THERE OTHER CERTIFICATE AUTHORITIES HAVING DIFFERENT AUTHENTICATION REQUIREMENTS?
986
YES
NO
UPDATE USER AUTHENTICATION
988
ACQUIRE CERTIFICATE FROM CERTIFICATE AUTHORITY
984
PERFORM ACTION
976
END

FIG. 9B



AUTHENTICATION DATA FLOW

	SEND	RECEIVE	SSL	ACTION
1005	USER	VENDOR	1/2	TRANSACTION OCCURS, SUCH AS SELECTING PURCHASE
1010	VENDOR	USER	1/2	TRANSMIT TRANSACTION ID (TID) AND AUTHENTICATION REQUEST (AR)
				AUTHENTICATION DATA (B') IS GATHERED FROM USER
1015	USER	TE	1/2	TRANSMIT TID AND B' WRAPPED IN THE PUBLIC KEY OF THE AUTHENTICATION ENGINE (AE), AS (PUB_AE(TID, B'))
1020	TE	AE	FULL	FORWARD TRANSMISSION
				ENROLLMENT AUTHENTICATION DATA (B) IS REQUESTED AND GATHERED
1025	VENDOR	TRANSACTION ENGINE (TE)	FULL	TRANSMITS TID, AR
1030	TE	MASS STORAGE(MS)	FULL	CREATE RECORD IN DATABASE
1035	TE	THE Xth DEPOSITORY(DX)	FULL	UID, TID
1040	DX	AE	FULL	TRANSMIT THE TID AND THE PORTION OF THE AUTHENTICATION DATA STORED AT ENROLLMENT (BX) AS (PUB_AE(TID, BX))
1045				AE ASSEMBLES B AND
1050				COMPARES TO B'
	AE	TE	FULL	TID, THE FILLED IN AR
	TE	VENDOR	FULL	TID, YES/NO
1055	TE	USER	1/2	TID, CONFIRMATION MESSAGE

FIG. 10

1100

SIGNING DATA FLOW			
SEND	RECEIVE	SSL	ACTION
USER	VENDOR	1/2	TRANSACTION OCCURS, SUCH AS AGREEING ON A DEAL
VENDOR	USER	1/2	TRANSMIT TRANSACTION IDENTIFICATION NUMBER (TID), AUTHENTICATION REQUEST (AR), AND AGREEMENT OR MESSAGE (M)
			CURRENT AUTHENTICATION DATA (B') AND A HASH OF THE MESSAGE RECEIVED BY THE USER (h(M')) IS GATHERED FROM USER
USER	TE	1/2	TRANSMIT TID, B', AR, AND h(M') WRAPPED IN THE PUBLIC KEY OF THE AUTHENTICATION ENGINE (AE) AS (PUB_AE(TID, B', h(M'))
TE	AE	FULL	FORWARD TRANSMISSION
			GATHER ENROLLMENT AUTHENTICATION DATA
VENDOR	TRANSACTION ENGINE (TE)	FULL	TRANSMITS UID, TID, AR, AND A HASH OF THE MESSAGE (h(M)).
TE	MASS STORAGE (MS)	FULL	CREATE RECORD IN DATABASE
TE	THE Xth DEPOSITORY(DX)	FULL	UID, TID
DX	AE	FULL	TRANSMIT THE TID AND THE PORTION OF THE AUTHENTICATION DATA STORED AT ENROLLMENT (BX),AS (PUB_AE(TID, BX))
			THE ORIGINAL VENDOR MESSAGE IS TRANSMITTED TO THE AE
TE	AE	FULL	TRANSMIT h(M)
			AE ASSEMBLES B, COMPARES TO B' AND COMPARES h(M) TO h(M')
AE	CRYPTOGRAPHIC ENGINE (CE)	FULL	REQUEST FOR DIGITAL SIGNATURE AND A MESSAGE TO BE SIGNED, FOR EXAMPLE, THE HASHED MESSAGE
AE	DX	FULL	TID, SIGNING UID
DX	CE	FULL	TRANSMIT THE PORTION OF THE CRYPTOGRAPHIC KEY CORRESPONDING TO THE SIGNING PARTY
			CE ASSEMBLES KEY AND SIGNS
CE	AE	FULL	TRANSMIT THE DIGITAL SIGNATURE (S) OF SIGNING PARTY
AE	TE	FULL	TID, THE FILLED IN AR, h(M), AND S
TE	VENDOR	FULL	TID, A RECEIPT=(TID, YES/NO, AND S), AND THE DIGITAL SIGNATURE OF THE TRUST ENGINE, FOR EXAMPLE, A HASH OF THE RECEIPT ENCRYPTED WITH THE TRUST ENGINE'S PRIVATE KEY (Priv_TE(h(RECEIPT)))
TE	USER	1/2	TID, CONFIRMATION MESSAGE

1103 1105 1110 1115 1120 1125 1130 1135 1140

FIG. 11



ENCRYPTION/DECRYPTION DATA FLOW			
SEND	RECEIVE	SSL	ACTION
DECRYPTION			
			PERFORM AUTHENTICATION DATA PROCESS 1000, INCLUDE THE SESSION KEY (SYNC) IN THE AR, WHERE THE SYNC HAS BEEN ENCRYPTED WITH THE PUBLIC KEY OF THE USER AS PUB_USER(SNYC)
			AUTHENTICATE THE USER
AE	CE	FULL	FORWARD PUB_USER(SYNC) TO CE
AE	DX	FULL	UID, TID
DX	CE	FULL	TRANSMIT THE TID AND THE PORTION OF THE PRIVATE KEY AS (PUB_AE(TID, KEY_USER))
			CE ASSEMBLES THE CRYPTIOGRAPHIC KEY AND DECRYPTS THE SYNC
CE	AE	FULL	TID, THE FILLED IN AR INCLUDING DECRYPTED SYNC
AE	TE	FULL	FORWARD TO TE
TE	REQUESTING APP/VENDOR	1/2	TID, YES/NO, SYNC
ENCRYPTION			
REQUESTING APP/VENDOR	TE	1/2	REQUEST FOR PUBLIC KEY OF USER
TE	MS	FULL	REQUEST DIGITAL CERTIFICATE
MS	TE	FULL	TRANSMIT DIGITAL CERTIFICATE
TE	REQUESTING APP/VENDOR	1/2	TRANSMIT DIGITAL CERTIFICATE

1205, 1210, 1215, 1220, 1225, 1230, 1235, 1240, 1245, 1250

FIG. 12

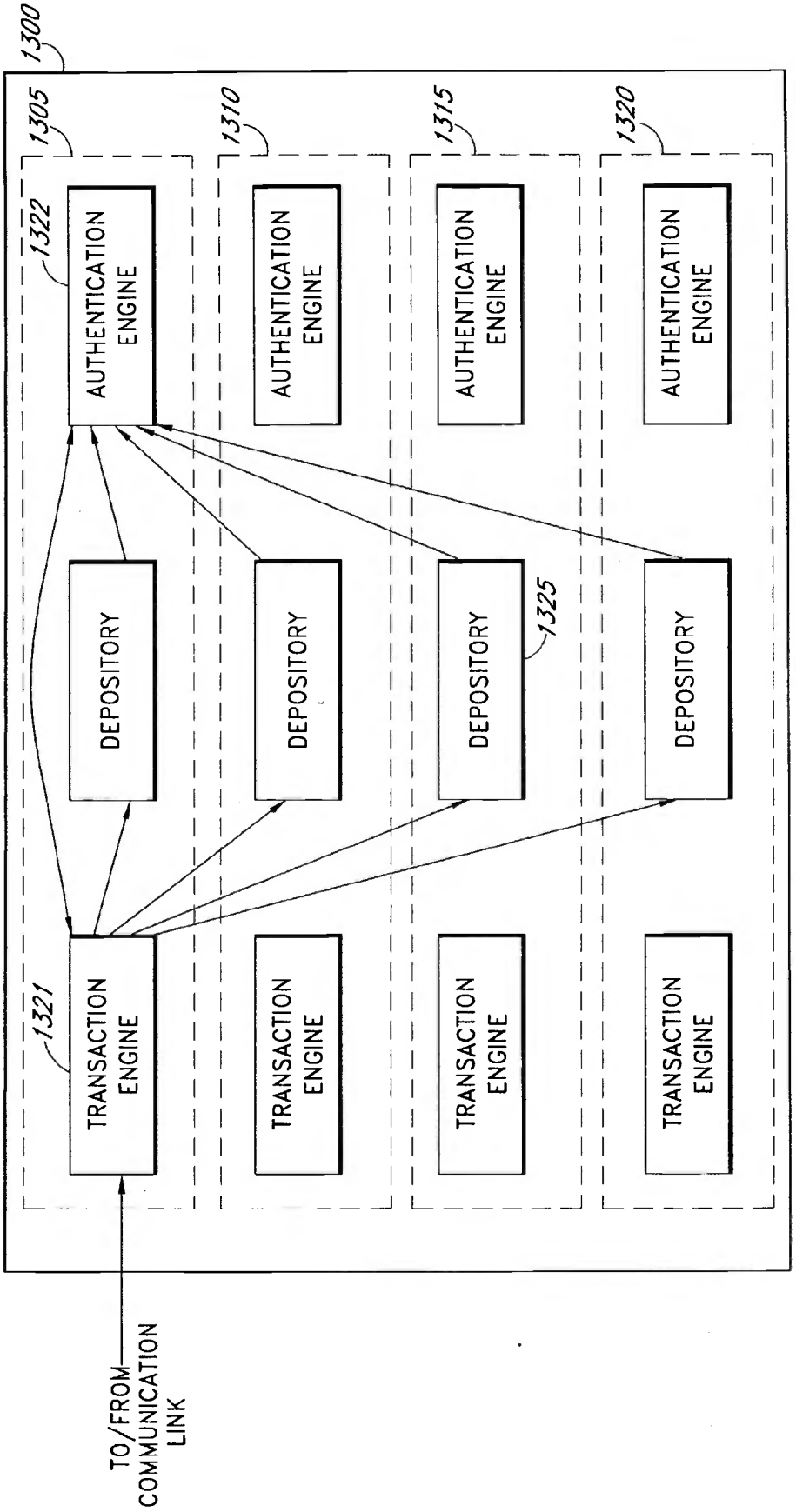

1300
1305
1310
1315
1320
1322
AUTHENTICATION ENGINE
AUTHENTICATION ENGINE
AUTHENTICATION ENGINE
AUTHENTICATION ENGINE
DEPOSITORY
DEPOSITORY
DEPOSITORY
1325
DEPOSITORY
1321
TRANSACTION ENGINE
TRANSACTION ENGINE
TRANSACTION ENGINE
TRANSACTION ENGINE
TO/FROM COMMUNICATION LINK

FIG. 13

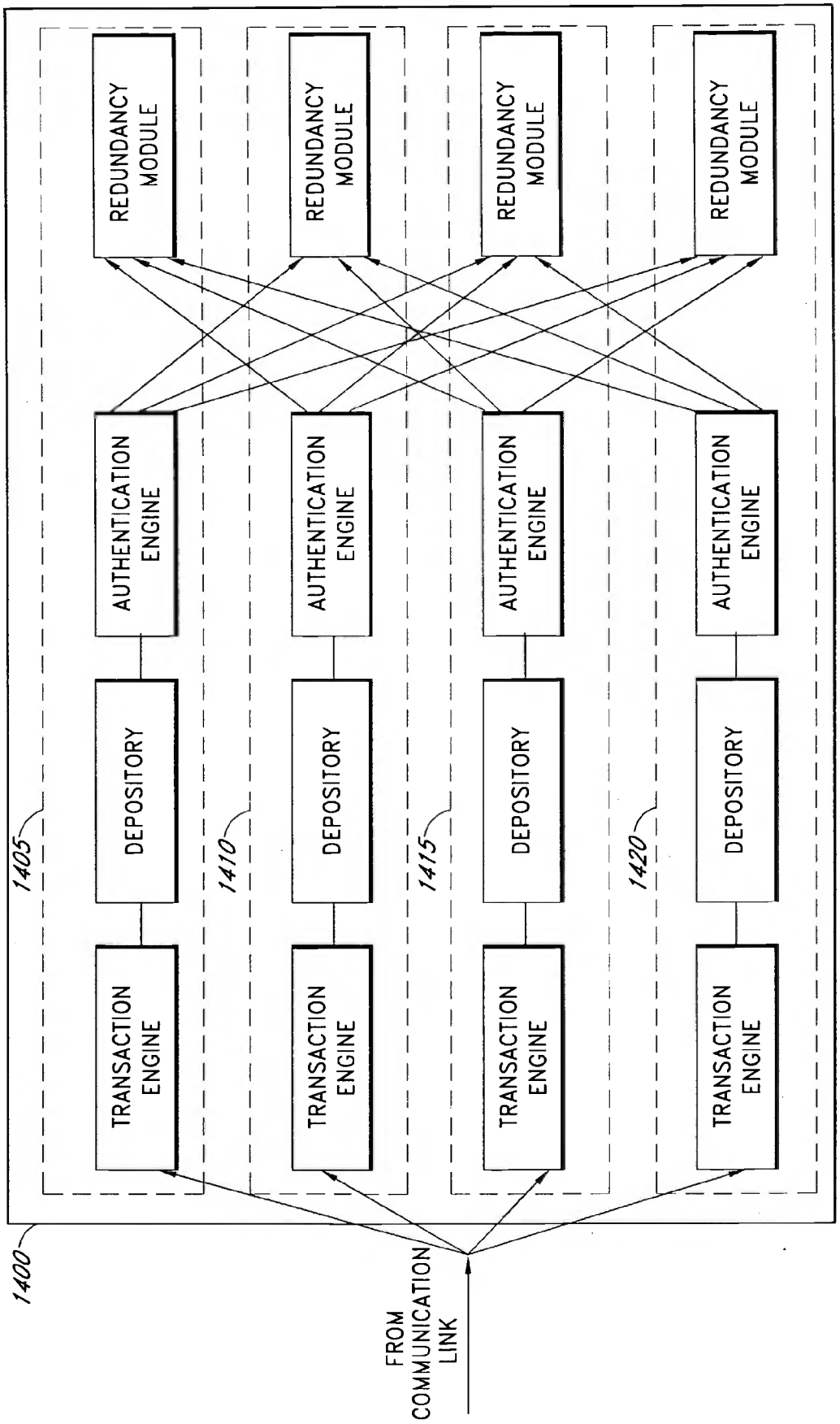

REDUNDANCY MODULE
REDUNDANCY MODULE
REDUNDANCY MODULE
REDUNDANCY MODULE
AUTHENTICATION ENGINE
AUTHENTICATION ENGINE
AUTHENTICATION ENGINE
AUTHENTICATION ENGINE
DEPOSITORY
DEPOSITORY
DEPOSITORY
DEPOSITORY
TRANSACTION ENGINE
TRANSACTION ENGINE
TRANSACTION ENGINE
TRANSACTION ENGINE
1405
1410
1415
1420
1400
FROM COMMUNICATION LINK

FIG. 14

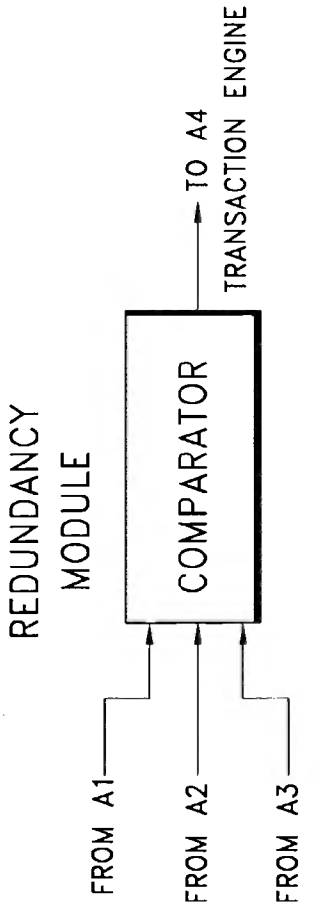

REDUNDANCY MODULE
COMPARATOR
TO A4 TRANSACTION ENGINE
FROM A1
FROM A2
FROM A3

FIG. 15

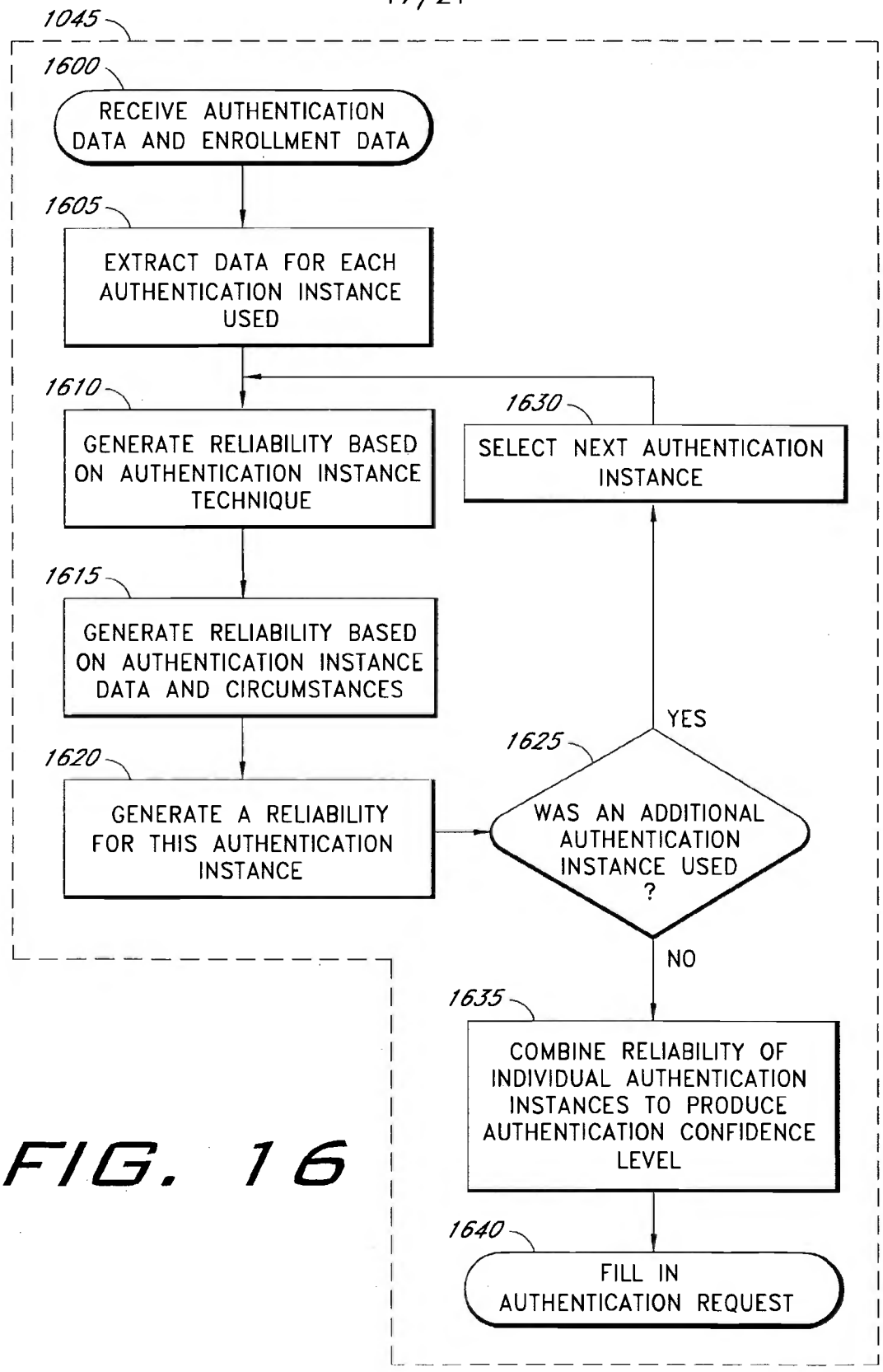

1045
1600
RECEIVE AUTHENTICATION DATA AND ENROLLMENT DATA
1605
EXTRACT DATA FOR EACH AUTHENTICATION INSTANCE USED
1610
GENERATE RELIABILITY BASED ON AUTHENTICATION INSTANCE TECHNIQUE
1615
GENERATE RELIABILITY BASED ON AUTHENTICATION INSTANCE DATA AND CIRCUMSTANCES
1620
GENERATE A RELIABILITY FOR THIS AUTHENTICATION INSTANCE
1625
WAS AN ADDITIONAL AUTHENTICATION INSTANCE USED ?
YES
1630
SELECT NEXT AUTHENTICATION INSTANCE
NO
1635
COMBINE RELIABILITY OF INDIVIDUAL AUTHENTICATION INSTANCES TO PRODUCE AUTHENTICATION CONFIDENCE LEVEL
1640
FILL IN AUTHENTICATION REQUEST

FIG. 16

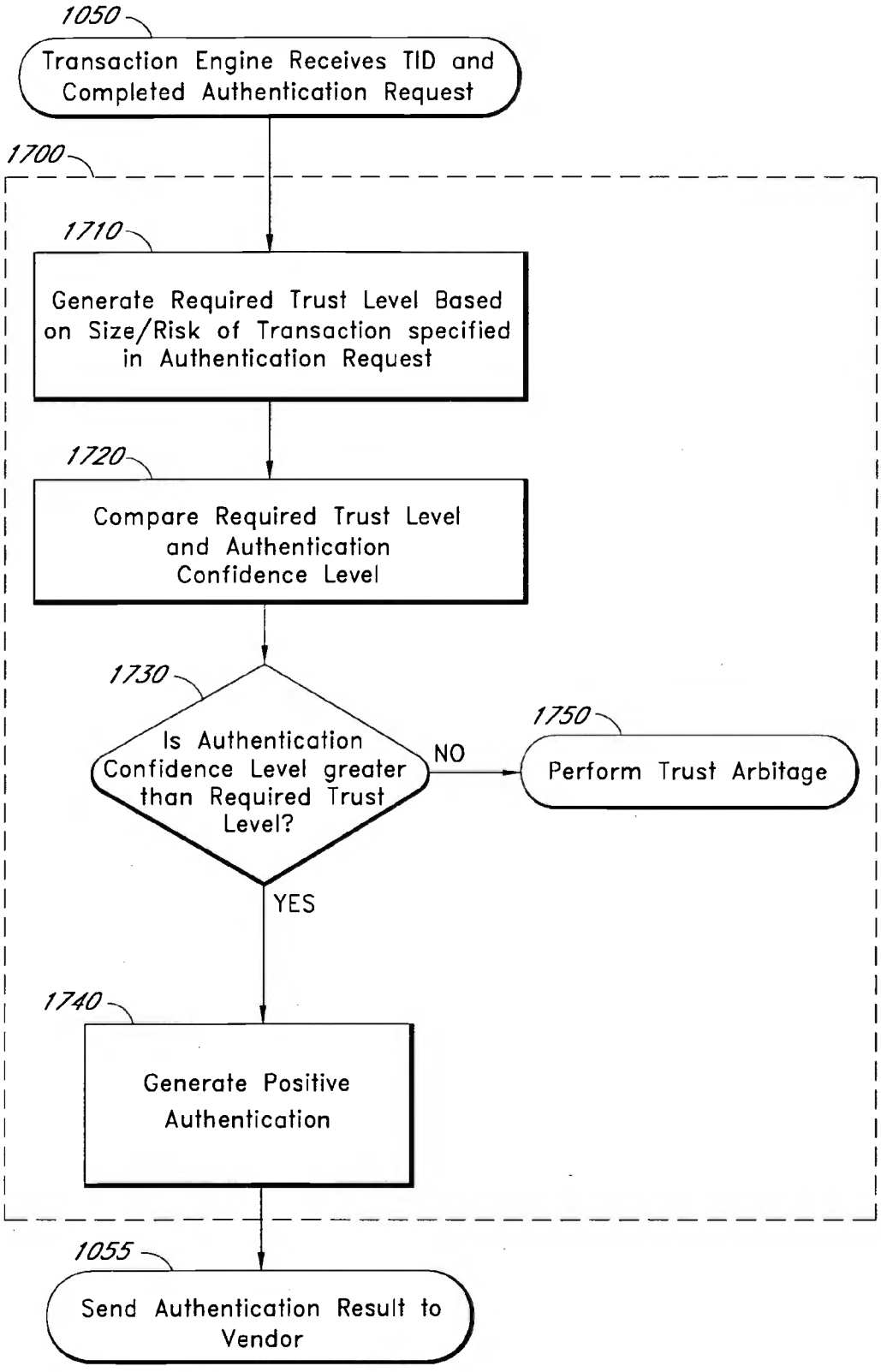

1050
Transaction Engine Receives TID and Completed Authentication Request
1700
1710
Generate Required Trust Level Based on Size/Risk of Transaction specified in Authentication Request
1720
Compare Required Trust Level and Authentication Confidence Level
1730
Is Authentication Confidence Level greater than Required Trust Level?
NO
1750
Perform Trust Arbitage
YES
1740
Generate Positive Authentication
1055
Send Authentication Result to Vendor

FIG. 17

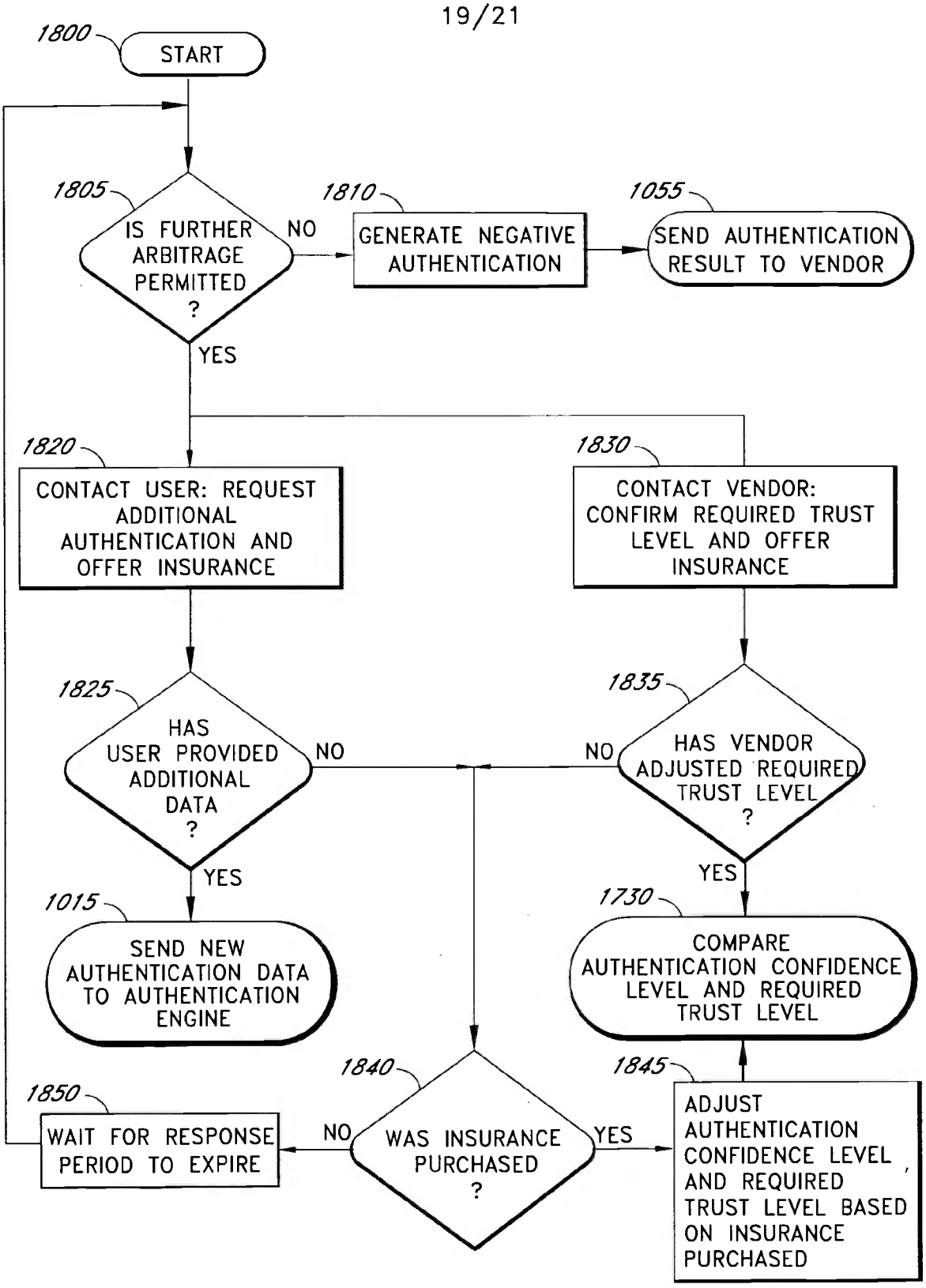

1800
START
1805
IS FURTHER ARBITRAGE PERMITTED ?
NO
1810
GENERATE NEGATIVE AUTHENTICATION
1055
SEND AUTHENTICATION RESULT TO VENDOR
YES
1820
CONTACT USER: REQUEST ADDITIONAL AUTHENTICATION AND OFFER INSURANCE
1830
CONTACT VENDOR: CONFIRM REQUIRED TRUST LEVEL AND OFFER INSURANCE
1825
HAS USER PROVIDED ADDITIONAL DATA ?
NO
1835
HAS VENDOR ADJUSTED REQUIRED TRUST LEVEL ?
NO
YES
YES
1015
SEND NEW AUTHENTICATION DATA TO AUTHENTICATION ENGINE
1730
COMPARE AUTHENTICATION CONFIDENCE LEVEL AND REQUIRED TRUST LEVEL
1840
WAS INSURANCE PURCHASED ?
1845
ADJUST AUTHENTICATION CONFIDENCE LEVEL AND REQUIRED TRUST LEVEL BASED ON INSURANCE PURCHASED
1850
WAIT FOR RESPONSE PERIOD TO EXPIRE
NO
YES

FIG. 18

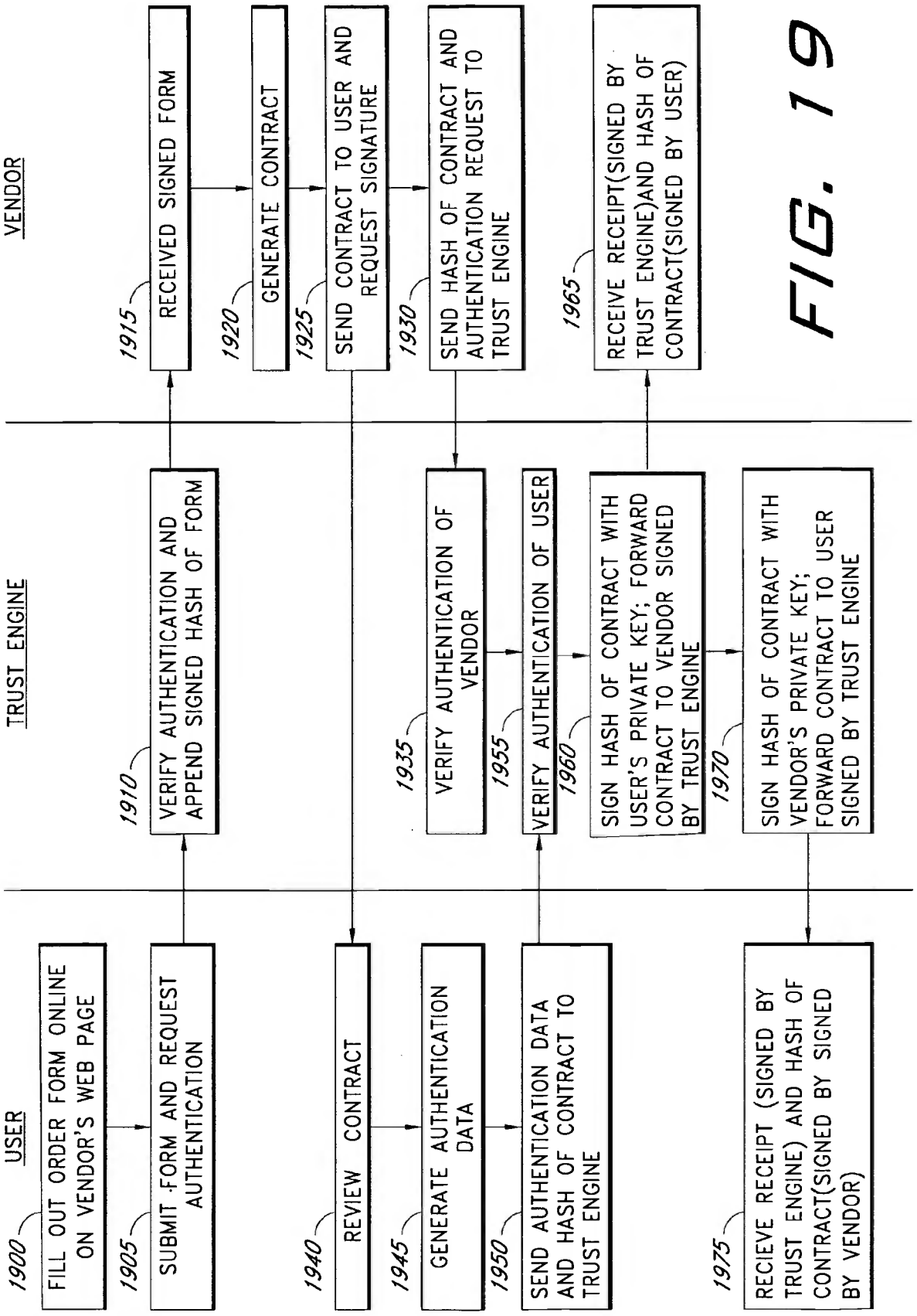

USER
TRUST ENGINE
VENDOR
1900
FILL OUT ORDER FORM ONLINE ON VENDOR'S WEB PAGE
1905
SUBMIT FORM AND REQUEST AUTHENTICATION
1910
VERIFY AUTHENTICATION AND APPEND SIGNED HASH OF FORM
1915
RECEIVED SIGNED FORM
1920
GENERATE CONTRACT
1925
SEND CONTRACT TO USER AND REQUEST SIGNATURE
1930
SEND HASH OF CONTRACT AND AUTHENTICATION REQUEST TO TRUST ENGINE
1935
VERIFY AUTHENTICATION OF VENDOR
1940
REVIEW CONTRACT
1945
GENERATE AUTHENTICATION DATA
1950
SEND AUTHENTICATION DATA AND HASH OF CONTRACT TO TRUST ENGINE
1955
VERIFY AUTHENTICATION OF USER
1960
SIGN HASH OF CONTRACT WITH USER'S PRIVATE KEY; FORWARD CONTRACT TO VENDOR SIGNED BY TRUST ENGINE
1965
RECEIVE RECEIPT(SIGNED BY TRUST ENGINE)AND HASH OF CONTRACT(SIGNED BY USER)
1970
SIGN HASH OF CONTRACT WITH VENDOR'S PRIVATE KEY; FORWARD CONTRACT TO USER SIGNED BY TRUST ENGINE
1975
RECIEVE RECEIPT (SIGNED BY TRUST ENGINE) AND HASH OF CONTRACT(SIGNED BY SIGNED BY VENDOR)

FIG. 19

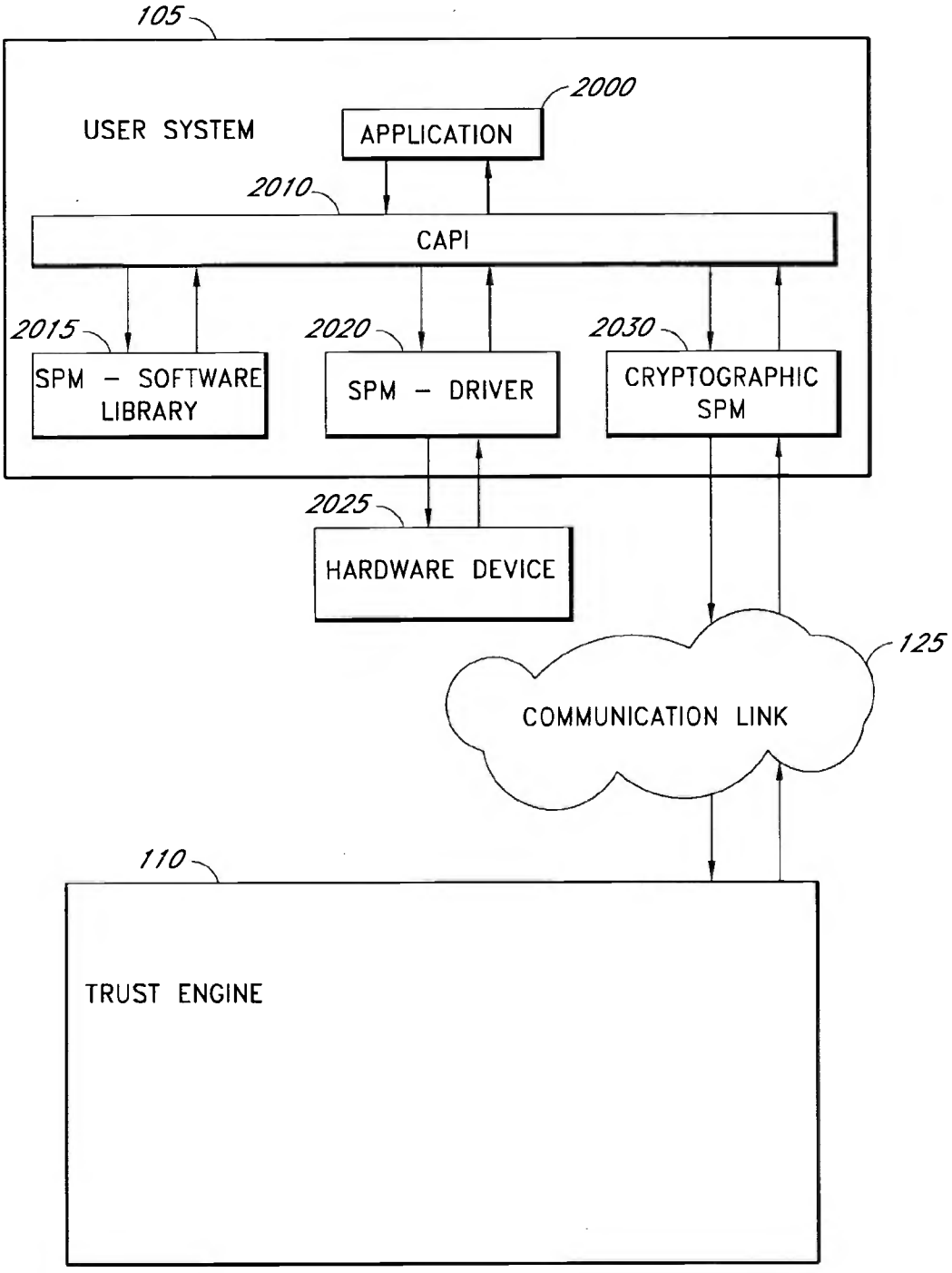

105
USER SYSTEM
2000
APPLICATION
2010
CAPI
2015
SPM – SOFTWARE LIBRARY
2020
SPM – DRIVER
2030
CRYPTOGRAPHIC SPM
2025
HARDWARE DEVICE
125
COMMUNICATION LINK
110
TRUST ENGINE

FIG. 20

INTERNATIONAL SEARCH REPORT

Internal application No
PCT/US 00/25748

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F G07F G07C

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 375 244 A (MCNAIR BRUCE E) 20 December 1994 (1994-12-20) abstract column 1, line 50 -column 3, line 5 column 5, line 21 - line 45 column 8, line 42 -column 11, line 15	1-42
X	GB 2 237 670 A (DE LA RUE SYST ;MIDLAND BANK PLC (GB)) 8 May 1991 (1991-05-08)	1-3,8, 10,12, 13,15-26
A	abstract page 2, line 28 -page 4, line 35	4-7,9, 11,14, 27-42

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
23 February 2001	02/03/2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Van Dop, E

1

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 386 104 A (SIME IAIN R F) 31 January 1995 (1995-01-31)	1-3,12, 13, 16-31, 34-41
A	abstract column 5, line 45 - line 65	4-11,14, 15,32,33
P,X	WO 99 65207 A (MICROSOFT CORP) 16 December 1999 (1999-12-16) abstract	1,2,6,7, 17-26
A	page 33, line 10 - line 30	3-5, 8-16, 27-42

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Internat pplication No
PCT/US 00/25748

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5375244 A	20-12-1994	NONE	
GB 2237670 A	08-05-1991	NONE	
US 5386104 A	31-01-1995	DE 69425944 D	26-10-2000
		EP 0652540 A	10-05-1995
		ES 2150474 T	01-12-2000
		JP 7192164 A	28-07-1995
WO 9965207 A	16-12-1999	NONE	

Form PCT/ISA/210 (patent family annex) (July 1992)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 793 367 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
03.09.1997 Bulletin 1997/36

(51) Int. Cl.[6]: **H04L 9/08**

(21) Application number: **97102667.9**

(22) Date of filing: **19.02.1997**

(84) Designated Contracting States:
DE FR GB

(30) Priority: **29.02.1996 JP 43315/96**

(71) Applicants:
- **Oki Electric Industry Co., Ltd.**
  Tokyo (JP)
- **CASIO COMPUTER COMPANY LIMITED**
  Shinjuku-ku Tokyo 160 (JP)

(72) Inventors:
- **Shona, Yoshihiro**
  7-12, Toranomon 1-chome, Minato-ku Tokyo (JP)
- **Kizaki, Masaharu,**
  c/o Casio Computer Co., Ltd.
  Higashiyamato-shi, Tokyo (JP)
- **Kawano, Kazuya,**
  c/o Casio Computer Co., Ltd.
  Higashiyamato-shi, Tokyo (JP)

(74) Representative: **Betten & Resch**
Reichenbachstrasse 19
80469 München (DE)

(54) Key distribution system and method

(57) A communication system (10) can maintain high security in communication between a communication apparatus (11) and a terminal (12). The communication system (10) includes the communication apparatus (11) and the terminal (12) provided with a memory unit (16) for storing data for specifying function of the communication apparatus. The communication apparatus (11) and the terminal (12) include random number generators (13, 13') for generating random numbers, encryption/decryption key preparing units (14, 14') for preparing encryption/decryption keys (Y1) on the basis of both random numbers (R1) and (R2) generated by the respective random number generators (13, 13') of the communication apparatus (11) and the terminal (12) and a secret key (X1) held in common by the communication apparatus (11) and the terminal (12), and encryption/decryption processing units (15, 15') for encrypting or decrypting communication data between the communication apparatus (11) and the terminal (12) containing the data by means of the encryption/decryption key (Y1), respectively.

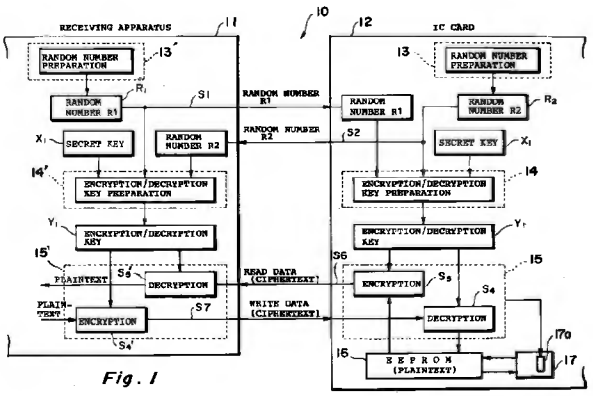


Fig. 1

EP 0 793 367 A2

Description

BACKGROUND OF THE INVENTION

The present invention relates to a communication system including, for example, a communication apparatus and various terminals represented by, for example, IC cards used in combination with the communication apparatus to specify the function of the communication apparatus, and a communication method.

There is proposed an attempt that a receiving apparatus capable of receiving a number of TV programs or radio programs is manufactured as a communication apparatus and the receiving apparatus and an IC card for placing restrictions on the receiving apparatus in accordance with contents of a contract are sold as a set.

Stored in the IC card are data defined in accordance with the contents of the contract which enable the receiving apparatus to receive only a specific program or to receive programs only during a predetermined term of the contract, for example.

Accordingly, when the IC card constituting a terminal is connected or loaded into the receiving apparatus combined with the IC card as the set, communication between the receiving apparatus and the IC card is performed. The receiving apparatus reads out the data for specifying the contract contents from the IC card by means of the communication, so that predetermined function is selectively given to the receiving apparatus on the basis of the data.

In the communication between the receiving apparatus and the IC card, in order to prevent the iniquity by forgery, the authentication for judging whether both of them have genuine relation with each other based on the regular contract or not is performed and, after the authentication, transmission and reception of data for specifying the contract contents is made therebetween.

Fig. 6 is a schematic diagram illustrating an example of a communication method in a communication system including an IC card and a receiving apparatus in a prior art.

As shown in Fig. 6, the conventional communication system adopts the mutual authentication.

An IC card 1 and a receiving apparatus 2 include random number generators 3 and 4, respectively. The receiving apparatus 2 transmits a random number R1 generated by the random number generator 4 to an authentication code preparing unit 5a of the IC card 1 and prepares at an authentication code preparing unit 6a thereof an authentication code X' on the basis of a first secret key X1 held in common by the IC card 1 and the receiving apparatus 2 and the random number R1. The IC card 1 which has received the random number R1 prepares at the authentication code preparing unit 5a thereof an authentication code X on the basis of the first secret key X1 and the random number R1 and sends the authentication code X to the receiving apparatus 2.

The receiving apparatus 2 which has received the authentication code X from the IC card 1 compares at a comparison unit 8 thereof the authentication code X with the authentication code X' prepared by the authentication code preparing unit 6a to thereby perform a first authentication A.

Further, the IC card 1 transmits a random number R2 prepared by a random number generator 3 thereof to an authentication code preparing unit 6b of the receiving apparatus 2 and prepares at an authentication code preparing unit 5b thereof an authentication code Y on the basis of a second secret key X2 held in common by the IC card 1 and the receiving apparatus 2 and the random number R2. The receiving apparatus 2 which has received the random number R2 prepares at the authentication code preparing unit 6b an authentication code Y' on the basis of the second secret key X2 and the random number R2 and transmits the authentication code Y' to the IC card 1.

The IC card 1 which has received the authentication code Y' from the receiving apparatus 2 compares at its comparison unit 7 the authentication code Y with the authentication code Y' prepared by the unit 6b of the receiving apparatus 2 to thereby perform a second authentication B.

After it has been confirmed that both of them have genuine relation with each other based on the regular contract in accordance with the mutual authentication A and B, data specifying the contract contents are exchanged therebetween in plaintext.

In the conventional system as described above, the secret keys X1 and X2 used in the first and second authentication A and B can be made identical, while different keys are used as the secret keys X1 and X2, respectively, so that cryptanalysis of the secret keys is made difficult.

In the conventional system, however, since the data specifying the contract contents are exchanged in plaintext, the plaintext data can be deciphered or understood relatively easily by monitoring the plaintext data with technical knowledge of a certain degree.

Even if the plaintext data are deciphered, alteration such as, for example, extension of a contract term, addition of contract programs or the like is made to the IC card 1 or the receiving apparatus 2 is directly modified on the basis of the deciphered plaintext data relatively easily.

Further, in the conventional system, the authentication codes X and Y' are prepared on the basis of the single random number R1 or R2 and the single secret key X1 or X2, respectively. Accordingly, it is relatively easy to cryptanalyze the secret key X1 or X2 by monitoring the communication between the communication apparatus and the terminal. The cryptanalysis of the secret key increases the possibility that the terminal such as the IC card is forged.

EX1010, Page 1360

SUMMARY OF THE INVENTION

Accordingly, it is an object of the present invention to provide a communication system and method capable of maintaining the higher security than in the prior art even if communication between a communication apparatus and a terminal is monitored.

Further, it is another object of the present invention to provide a communication system and method which make alteration or forgery difficult as compared with the prior art.

The communication system according to the present invention including a communication apparatus for reception or transmission and a terminal provided with a memory unit in which data for specifying function of the communication apparatus are stored, is characterized in that the communication apparatus and the terminal each comprise a random number generating unit for generating a random number, an encryption/decryption key preparing unit for preparing an encryption/decryption key on the basis of both random numbers generated by the respective random number generating units of the communication apparatus and the terminal and a secret key held in common by the communication apparatus and the terminal, and an encryption/decryption processing unit for encrypting or decrypting communication data between the communication apparatus and the terminal including the data by using the encryption/decryption key.

In the communication system according to the present invention, the respective encryption/decryption key preparing units provided in the communication apparatus and the terminal prepare the encryption/decryption keys on the basis of both the first and second random numbers and the secret key. Further, the communication apparatus and the terminal encrypt at the respective encryption/decryption processing units communication data between the communication apparatus and the terminal on the basis of the encryption/decryption keys prepared on the basis of both the random numbers and the secret key, so that communication between the communication apparatus and the terminal is performed by means of the ciphertext.

Accordingly, plaintext is not used in communication between the communication apparatus and the terminal as in the prior art and even if communication between the communication apparatus and the terminal is monitored, it is not easy to cryptanalyze data or the like for specifying contact contents of the communication apparatus.

Furthermore, since the data or the like for specifying the contract contents are encrypted by means of the first and second random numbers of two kinds and the secret key, it is extremely complicated and difficult to cryptanalyze ciphertext as compared with cryptanalysis of a conventional authentication code which is encrypted by a single random number and a secret key and does not include plaintext data information.

Consequently, the encrypted communication contents are not cryptanalysed by a relatively easy cryptanalysis as in the prior art.

Accordingly, it is not easy even for the user based on the regular contact to cryptanalyse communication contents between the terminal used by the user itself and the communication apparatus and alteration of the terminal and the communication apparatus based on cryptanalysis of the communication contents is prevented exactly.

The terminal may include a reading/writing unit for reading data from the memory unit and writing data into the memory unit. The reading/writing unit may include judgment means for controlling reading of data from and writing of data into the memory unit. Further, the encryption/decryption key preparing means provided in the terminal may include signal producing means for producing a first signal for permitting reading of data by the reading/writing unit and a second signal for permitting writing of data by the reading/writing unit to supply the signals to the judgment means.

The data for specifying the function of the communication apparatus may include data for specifying a receivable program of the communication apparatus by way of example.

Further, the data for specifying the function of the communication apparatus may include, as another example, data of a term for the receivable program specified by the communication apparatus.

The signal producing means sends to the judgment unit a signal for permitting reading or writing.

The signal producing means can produce first and second flags as the signal to the judgment unit. The reading/writing unit is permitted to read data from the memory unit in a set state of the first flag and to write data into the memory unit in a set state of the second flag.

The random generating unit provided in the terminal may include an arithmetic operation unit for generating the second random number by using the first random number generated by the random number generating unit provided in the communication apparatus and a portion for storing the random number rewritably.

The arithmetic operation unit may be constituted by an arithmetic circuit for generating the second random number by the arithmetic operation of the random number stored in the random number storage portion and the first random number.

Further, the arithmetic operation unit may include a conversion processing circuit for converting the second random number.

The random number in the random number storage portion can be successively updated by a value converted by the conversion processing circuit.

Furthermore, the conversion of the conversion processing circuit can adopt a bit replacement process or a bit inversion process.

The communication method according to the present invention between a communication apparatus for reception or transmission and a terminal provided

EX1010, Page 1361

with a memory unit in which data for specifying function of the communication apparatus are stored, is characterized in that first and second random numbers generated by the communication apparatus and the terminal, respectively, and a secret key held in common by the communication apparatus and the terminal are used to encrypt the data and the function of the communication apparatus is restricted by communication of the encrypted data.

The communication apparatus and the terminal each comprise a random number generating unit, an encryption/decryption key preparing unit for preparing an encryption/decryption key on the basis of the first and second random numbers generated by the respective random number generating units and the secret key, and an encryption/decryption processing unit for encrypting or decrypting communication data between the communication apparatus and the terminal including the data by using the encryption/decryption key, and when the terminal receives a command from the communication apparatus for reading out data from the memory unit, the terminal encrypts the read-out data from the memory unit by means of the encryption/decryption key to transmit the encrypted data to the communication apparatus, the terminal being responsive to a command from the communication apparatus for writing data into the memory unit to decrypt ciphertext corresponding to the data by means of the encryption/decryption key and to write the decrypted data into the memory unit when the terminal receives the command from the communication apparatus.

The reading of data from the terminal and the writing of data to the terminal cannot be performed at the same time. By preventing the writing of data from being performed at the same time as the reading of data, alteration of data to the terminal can be prevented exactly.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic diagram illustrating a communication system according to the present invention;

Fig. 2 is a schematic diagram similar to Fig. 1 illustrating another communication system according to the present invention;

Fig. 3 is a flow chart showing a data reading process of the communication system according to the present invention shown in Fig. 2;

Fig. 4 is a flow chart showing a data writing process of the communication system according to the present invention shown in Fig. 2;

Fig. 5 is a schematic diagram illustrating another example of a random number generating unit in a terminal of the communication system according to the present invention; and

Fig. 6 is a schematic diagram illustrating a conventional communication system.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The features which are considered characteristic of the invention will be more apparent from the following description taken in connection with embodiments shown in the accompanying drawings.

A communication system 10 according to the present invention illustrated in Fig. 1 as an example comprises a receiving apparatus 11 such as, for example, a television receiver and a terminal 12 in which data for specifying the function of the receiving apparatus are stored. In the following description, an IC card 12 is used as the terminal 12 by way of example.

The communication system 10 according to the present invention comprises the receiving apparatus 11 and the IC card 12 for storing data relative to, for example, a receiving channel or a receiving term of the receiving apparatus 11.

The receiving apparatus 11 includes a random number generator 13' for generating a random number R1, an encryption/decryption key preparing unit 14' for preparing an encryption/decryption key on the basis of a secret key X1 held in common by the receiving apparatus 11 and the IC card 12, and an encryption/decryption processing unit 15' for encrypting plaintext or decrypting cryptogram or ciphertext by using the encryption/decryption key Y1 prepared by the encryption/decryption key preparing unit 14'.

The IC card 12 combined with the receiving apparatus 11 includes a random number generator 13 for generating a random number R2, an encryption/decryption key preparing unit 14 for preparing an encryption/decryption key on the basis of the secret key X1, and an encryption/decryption processing unit 15 for encrypting plaintext or decrypting ciphertext by using the encryption/decryption key Y1 prepared by the encryption/decryption key preparing unit 14.

Further, the IC card 12 includes a memory unit 16 made of a rewritable memory such as, for example, an EEPROM (Electrically Erasable Programmable Read Only Memory) for storing programs or data and a reading/writing unit 17 for writing data into the memory unit and reading data from the memory unit.

The encryption/decryption key preparing units 14' and 14 of the receiving apparatus 11 and the IC card 12, respectively, prepare the encryption/decryption key on the basis of the respective secret key X1 by using the random numbers R1 and R2 generated by the respective random number generator 13' and 13 and the random numbers R2 and R1 received from the other unit by means of communication between the receiving apparatus 11 and the IC card 12, respectively.

Further, the encryption/decryption processing units 15' and 15 of the receiving apparatus 11 and the IC card 12, respectively, encrypt plaintext or decrypt ciphertext by using the encryption/decryption key Y1 prepared by the respective encryption/decryption key preparing unit 14' and 14, respectively.

EX1010, Page 1362

Operation of each unit of the communication system 10 is now described.

Before communication of cryptogram between the receiving apparatus 11 and the IC card 12, the first random number R1 generated by the random number generator 13' of the receiving apparatus 11 is sent to the IC card 12 (step S1). On the other hand, the second random number R2 generated by the random number generator 13 of the IC card 12 is sent to the receiving apparatus 11 (step S2). Consequently, the receiving apparatus 11 and the IC card 12 hold both of the first and second random numbers R1 and R2.

The encryption/decryption key preparing units 14' and 14 prepare the encryption/decryption key Y1 on the basis of the secret key X1 held in common by both of the receiving apparatus 11 and the IC card 12 by using both of the first and second random numbers R1 and R2 (steps S3 and S3').

The encryption/decryption processing units 15' and 15 of the receiving apparatus 11 and the IC card 12, respectively, perform encryption and decryption by using the encryption/decryption key Y1 (steps S4, S4' and S5, S5').

The encryption/decryption key preparing units 14' and 14 can be constituted by, for example, an XOR circuit which attains an exclusive OR of both the first and second random numbers R1 and R2 and the secret key X1. It is desirable that even if each of the random numbers R1 and R2 is varied slightly the encryption/decryption key Y1 prepared by the encryption/decryption key preparing units 14' and 14 is varied greatly.

More particularly, when one bit or so of a series of bits representative of the random number R1 or R2 to be inputted is varied, it is desirable to use the encryption/decryption key preparing unit 14' or 14 which prepares the encryption/decryption key Y1 which is an output corresponding to the inputted random number and constituted by a series ofbits having, for example, four to five bits of which values are varied.

The subsequent procedure is various depending on programs or applications in the IC card 12 for specifying the function of the receiving apparatus 11.

Among these applications, an application which allows the receiving apparatus 11 to receive a specific receiving channel on the basis of a contract by means of the IC card 12 is now described.

When the IC card 12 is loaded into the receiving apparatus 11, the reading/writing unit 17 of the IC card 12 reads out data relative to the receivable channel stored in the memory unit 16 in plaintext on the basis of a reading command from the receiving apparatus 11 to supply the data to the encryption/decryption processing unit 15. The encryption/decryption processing unit 15 encrypts the read plaintext data by using the encryption/decryption key Y1 (step S5). Then, the encrypted data or the read data are sent to the receiving apparatus 11 (step S6).

The read data received by the receiving apparatus 11 are decrypted at the encryption/decryption processing unit 15' of the receiving apparatus 11 by using the encryption/decryption key Y1 (step S5').

On the basis of the plaintext data decrypted by the above decryption, restrictions are placed on function of the receiving apparatus 11 so that the receiving apparatus 11 can receive only a predetermined channel specified by the data and the receiving apparatus 11 exhibits only predetermined function.

Further, when the receiving apparatus 11 receives a command of writing data to the IC card 12, the receiving apparatus 11 encrypts plaintext of the write data at the encryption/decryption processing unit 15' by using the encryption/decryption key Y1 (step S4'). The encrypted write data is sent to the IC card 12 (step S7).

The write data received by the IC card 12 is decrypted at the encryption/decryption processing unit 15 by using the encryption/decryption key Y1. The decrypted plaintext data is written in the memory unit 16 by the reading/writing unit 17.

The reading of data from the memory unit 16 and the writing of data to the memory unit 16 of the IC card 12 can be executed by the reading/writing unit 17 only when the processes at the above steps S1 to S3 have been completed normally.

Judgment as to whether the processes of the steps S1 to S3 have been completed normally or not can be made by judgment means 17a using, for example, one or a plurality of flags as described later.

In communication between the receiving apparatus 11 and the IC card 12, data are not exchanged in plaintext, and ciphertext is used to exchange data. Furthermore, since two kinds of the first and second random numbers are used to prepare the ciphertext, data cannot be cryptanalyzed easily as in the prior art even if cryptanalysis by monitoring of communication between the receiving apparatus 11 and the IC card 12 is tried.

Accordingly, the security of the communication system 10 is enhanced as compared with the prior art.

More particularly, the encryption/decryption key Y1 is varied in accordance with the random numbers R1 and R2 each time data is read and written. Accordingly, even if a number of read data or write data as described above are monitored and the encrypted communication information obtained by the monitoring and operation thereof are analyzed, it is substantially impossible to analyze commands, even if the commands are plaintext data, or the data itself.

Further, even if the commands of plaintext data or the data itself are not analyzed, there remains the possibility that contents of the communication information can be analyzed by the analysis of encrypted specific communication information and operation thereof. However, as described above, since the analyzed communication information includes a parameter of the encryption/decryption key Y1 varied each time data is read and written, it is extremely difficult to decrypt the communication information in order to understand the plaintext data corresponding to the communication information.

EX1010, Page 1363

Furthermore, even if the contents of the communication information including the parameter of the encryption/decryption key Y1 are analyzed and the communication information obtained with the intention of rewriting data in the memory unit 16 is inputted to the encryption/decryption processing unit 15 of the IC card 12, it is substantially impossible to rewrite data to intended contents since the encryption/decryption key Y1 upon analysis is different from the encryption/decryption key Y1 upon writing.

Accordingly, the security of communication between the receiving apparatus 11 and the IC card 12 can be enhanced and furthermore alteration or forgery by rewriting of data in the IC card 12 can be prevented exactly.

As described above, in the communication system 10, the encryption/decryption key Y1 is prepared from the two random numbers R1 and R2 and the secret key X1 and data encrypted by using the encryption/decryption key Y1 are exchanged between the receiving apparatus 11 and the IC card 12.

Accordingly, even when data having the same contents are read/written, values of the random numbers R1 and R2 used upon execution thereof are varied and a value of the encryption/decryption key Y1 is varied in accordance with the variation of the random numbers.

Accordingly, as described above, even if communication between the receiving apparatus 11 and the IC card 12 is monitored, cryptanalysis of ciphertext is not easy and high security can be ensured.

Further, even if contents of the communication information including the parameter of the encryption/decryption key Y1 are analyzed, since a value of the encryption/decryption key Y1 is varied each time data is read/written, alteration or forgery by rewriting of data in the IC card 12 can be prevented exactly and further alteration or forgery that the receiving apparatus 11 does not require the IC card 12 can be also prevented exactly.

The communication system 10 shown in Fig. 2 is basically identical with the communication system 10 shown in Fig. 1. In the communication system of Fig. 2, reading of data from and writing of data into the memory unit 16 of the IC card 12 is controlled by two flags from the encryption/decryption key preparing unit 14 of the IC card 12.

In Fig. 2, the encryption/decryption processing units 15' and 15, the memory unit 16 and the like are omitted for simplification of the drawing.

The encryption/decryption key preparing unit 14 of the IC card 12 includes signal producing means 14a for producing a first flag signal F1 which is a first signal and a second flag signal F2 which is a second signal.

Upon turning on of a power supply of the IC card 12 and upon resetting of the IC card 12, both the signals from the signal producing means 14a are set to a non-output state or a reset state "0" representing that reading and writing of data are not permitted.

When the encryption/decryption key preparing unit 14 of the IC card 12 prepares the encryption/decryption key Y1, the signal producing means 14a sets both the first and second flag signals F1 and F2 to "1" representing that reading or writing of data is permitted.

The reading/'writing unit 17 of the IC card 12 includes judgment means 17a which receives the flag signals F1 and F2.

When the reading/writing unit 17 receives an instruction for a data reading process for reading data from the memory unit 16 (Fig. 1), the reading/writing unit 17 changes the second flag F2 to the reset state "0" if the first flag F1 is in the set state "1". Further, when the reading/writing unit 17 receives an instruction for a data writing process for writing data into the memory unit 16, the reading/writing unit 17 changes the first flag F1 to the reset state "0" if the second flag F2 is in the set state "1".

The operating of reading data from and writing data into the memory unit 16 of the IC card 12 is now described with reference to Figs. 3 and 4.

When the encryption/decryption key Y1 is prepared by the encryption/decryption key preparing unit 14 of the IC card 12 as described above with reference to Fig. 1, the encryption/decryption key preparing unit 14 sets both the first and second flags F1 and F2 to the set state "1".

At this time, when the reading/writing unit 17 receives the command for the data reading process, the judgment means 17a judges whether the first flag F1 is in the set state or not (step S21) as shown in Fig. 3.

When the first flag F1 is in the reset state "0", the reading/'writing unit 17 terminates the data reading process.

On the other hand, when the first flag F1 is in the set state "1", the second flag F2 is changed to the reset state "0" (step S22). In this flag state, the reading/writing unit 17 reads out plaintext data from the memory unit 16 (Fig. 1). The plaintext data are encrypted by the encryption/decryption processing unit 15 by using the encryption/'decryption key Y1 as described above (step S23) and are then transmitted to the receiving apparatus 11 as ciphertext (step S24).

Further, as shown in Fig. 4, when the reading/writing unit 17 receives the command for the data writing process, the judgment means 17a judges whether the second flag F2 is in the set state or not (step S31).

When the second flag F2 is in the reset state "0", the reading/writing unit 17 terminates the data writing process.

On the other hand, when the second flag F2 is in the set state "1", the first flag F1 is changed to the reset state "0" (step S32). In this flag state, the encryption/decryption processing unit 15 of the IC card 12 decrypts the encrypted write data sent from the receiving apparatus 11 by using the encryption/decryption key Y1 (step S33). The decrypted plaintext data are written in the memory unit 16 by the reading/writing unit 17 (step S34).

In the communication system 10 shown in Fig. 2,

EX1010, Page 1364

the IC card 12 uses the first and second flags, so that simultaneous or parallel processing of the data reading process and the data writing process is impossible and only one of them can be selectively performed. Accordingly, the simultaneous processing of the data reading process and the data writing process using the same encryption/decryption key Y1 is impossible.

In other words, the encryption/decryption key Y1 is varied for each of the data reading process or the data writing process. Accordingly, even if contents of plaintext data are identical, encrypted communication data are varied for each communication since the encryption/decryption key Y1 is varied.

Accordingly, higher security can be ensured.

In addition, in the communication system 10 shown in Fig. 2, for example, even if a combination of one plaintext and ciphertext corresponding to the plaintext is clarified to thereby cryptanalyze an encryption/decryption key Y1 used at this time and altered data are written in order to alter the IC card 12, an encryption/'decryption key Y1 upon writing thereof is different from the cryptanalyzed encryption/'decryption key Y1.

Accordingly, it is impossible to alter data as desired and forgery of the IC card 12 by alteration of data can be prevented exactly.

In order to prevent the forgery of the IC card 12 by alteration of data, various proper measures can be adopted instead of adoption of the above flags to make it impossible to perform the simultaneous processing of the data reading process and the data writing process.

As the random number generators 13' and 13 of the receiving apparatus 11 and the IC card 12 shown in Figs. 1 and 2, a well-known random number generator for generating random numbers in hardware manner or a microprocessor for generating random numbers in software manner can be adopted.

However, in order to make small the IC card 12, the random number generator 13 of the IC card 12 is desirably configured as shown in Fig. 5.

The random number generator 13 of the IC card 12 includes, as shown in Fig. 5, a random number storage portion 18 constituted by a rewritable memory, for example, and an arithmetic operation portion 19.

The arithmetic operation portion 19 includes, in an example shown in Fig. 5, an arithmetic circuit 19a constituted by an XOR (exclusive-OR) circuit and a random number conversion circuit 19b for converting a random number.

An initial value R of an random number having 8 bits, for example, is stored in the random number storage portion 18 and the arithmetic circuit 19a generates a random number R2 by an arithmetic operation (XOR) of the random number R1 of, for example, 8 bits received from the receiving apparatus 11 and the initial value R from the random number storage portion 18.

Further, the random number R2 generated by the random number conversion circuit 19b is sent to the encryption/decryption key preparing unit 14' of the receiving apparatus 11 and the encryption/'decryption key preparing unit 14 of the IC card 12 in order to prepare the encryption/decryption key Y1 in the same manner as in the above example.

Further, the random number R2 prepared by the random number conversion circuit 19b is sent to the random number storage portion 18 in order to rewrite the initial value of the random number after converted by the random number conversion circuit 19b. Consequently, the initial value R of the random number is successively updated each time data is read or written subsequently.

Accordingly, security of communication between the receiving apparatus 11 and the IC card 12 can be further improved.

The conversion process by the random number conversion circuit 19b can adopt a bit inversion process that "1" or "0" of a specific bit of the random number R2 is inverted, a bit replacement process that a position of a specific bit is replaced with another position or another conversion process properly.

In the example shown in Fig. 5, the random number R1 generated by the receiving apparatus 11 can be utilized to generate the random number R2 in the IC card 12. Accordingly, the random number R2 can be prepared properly without provision of a microprocessor for executing a random number generating program in the IC card 12 and without provision of a random number generator having a complicated configuration as the random number generator 13 in the IC card 12.

Accordingly, the security of communication can be further enhanced and configuration of the IC card can be simplified.

In the foregoing description, the IC card is described as an example of the terminal, while various terminals such as a module, a package or the like including integrated circuits can be used regardless of a form thereof.

Further, as the memory unit of data used in various application programs and the random number storage portion, electrically rewritable non-volatile memory means such as a static random access memory (SRAM) with a backup power supply can be properly used in addition to the EEPROM.

Furthermore, the present invention is not limited to a specific encryption system or an interface system between the communication apparatus and the terminal and can be applied to any encryption system or a contact or non-contact type interface system.

Claims

1. A communication system including a communication apparatus for reception or transmission and a terminal provided with a memory unit in which data for specifying function of said communication apparatus are stored, wherein

said communication apparatus and said terminal each comprise:

EX1010, Page 1365

a random number generating unit for generating a random number;

an encryption/decryption key preparing unit for preparing an encryption/decryption key on the basis of both random numbers generated by said respective random number generating units of said communication apparatus and said terminal and a secret key held in common by said communication apparatus and said terminal; and

an encryption/decryption processing unit for encrypting or decrypting communication data between said communication apparatus and said terminal including said data by using said encryption/decryption key.

2. A communication system according to Claim 1, wherein said terminal further comprises a reading/writing unit for reading data from said memory unit and writing data into said memory unit,

said reading/writing unit including judgment means for controlling reading of data from and writing of data into said memory unit,

said encryption/decryption key preparing unit provided in said terminal including signal producing means for producing a signal for permitting reading and writing of data by said reading/writing unit to supply said signal to said judgment means.

3. A communication system according to Claim 2, wherein said signal producing means produces a first signal for permitting reading of data by said reading/writing unit to supply said first signal to said judgment means and produces a second signal for permitting of writing of data by said reading/writing unit to supply said second signal to said judgment means.

4. A communication system according to Claim 1, wherein said data for specifying the function of said communication apparatus comprises data for specifying a receivable program of said communication apparatus.

5. A communication system according to Claim 1, wherein said data for specifying the function of said communication apparatus comprises data of a term for a receivable program specified by said communication apparatus.

6. A communication system according to Claim 3, wherein said signal producing means produces first and second flags to supply said flags to said judgment means, and

said reading/writing unit is permitted to read data from said memory unit when said first flag is in a set state and to write data into said memory unit when said second flag is in a set state.

7. A communication system according to Claim 1 or 2, wherein said random number generating unit provided in said terminal includes an arithmetic operation portion for generating said second random number by using said first random number generated by said random number generating unit provided in said communication apparatus and a random number storage portion for storing random numbers rewritably.

8. A communication system according to Claim 7, wherein said arithmetic operation portion includes an arithmetic operation circuit for generating said second random number by means of an arithmetic operation of said random number stored in said random number storage unit and said first random number.

9. A communication system according to Claim 8, wherein said arithmetic operation portion includes a conversion processing circuit for converting said second random number, and said random number in said random number storage portion is updated so that said random number is equal to a converted value obtained from said conversion processing circuit.

10. A communication system according to Claim 9, wherein said conversion process by said conversion processing circuit is a bit replacement process.

11. A communication system according to Claim 9, wherein said conversion process by said conversion processing circuit is a bit inversion process.

12. A communication method between a communication apparatus for reception or transmission and a terminal provided with a memory unit in which data for specifying function of said communication apparatus are stored, wherein

first and second random numbers generated by said communication apparatus and said terminal, respectively, and a secret key held in common by said communication apparatus and said terminal are used to encrypt said data and the function of said communication apparatus is restricted by communication of said encrypted data.

13. A communication method according to Claim 12, wherein said communication apparatus and said terminal each comprise a random number generating unit, an encryption/decryption key preparing unit for preparing an encryption/decryption key on the basis of said first and second random numbers generated by said respective random number generating units and said secret key, and an encryption/'decryption processing unit for encrypting or decrypting communication data between said communication apparatus and said terminal including

EX1010, Page 1366

said data by using said encryption/'decryption key, and

when said terminal receives a command from said communication apparatus for reading out data from said memory unit, said terminal encrypts said read-out data from said memory unit by means of said encryption/decryption key to transmit said encrypted data to said communication apparatus,

said terminal being responsive to a command from said communication apparatus for writing data into said memory unit to decrypt ciphertext corresponding to the data by means of said encryption/'decryption key and to write said decrypted data into said memory unit when said terminal receives said command from said communication apparatus.

14. A communication method according to Claim 13, wherein the reading of data from said terminal and the writing of data to, said terminal cannot be performed at the same time.

EX1010, Page 1367

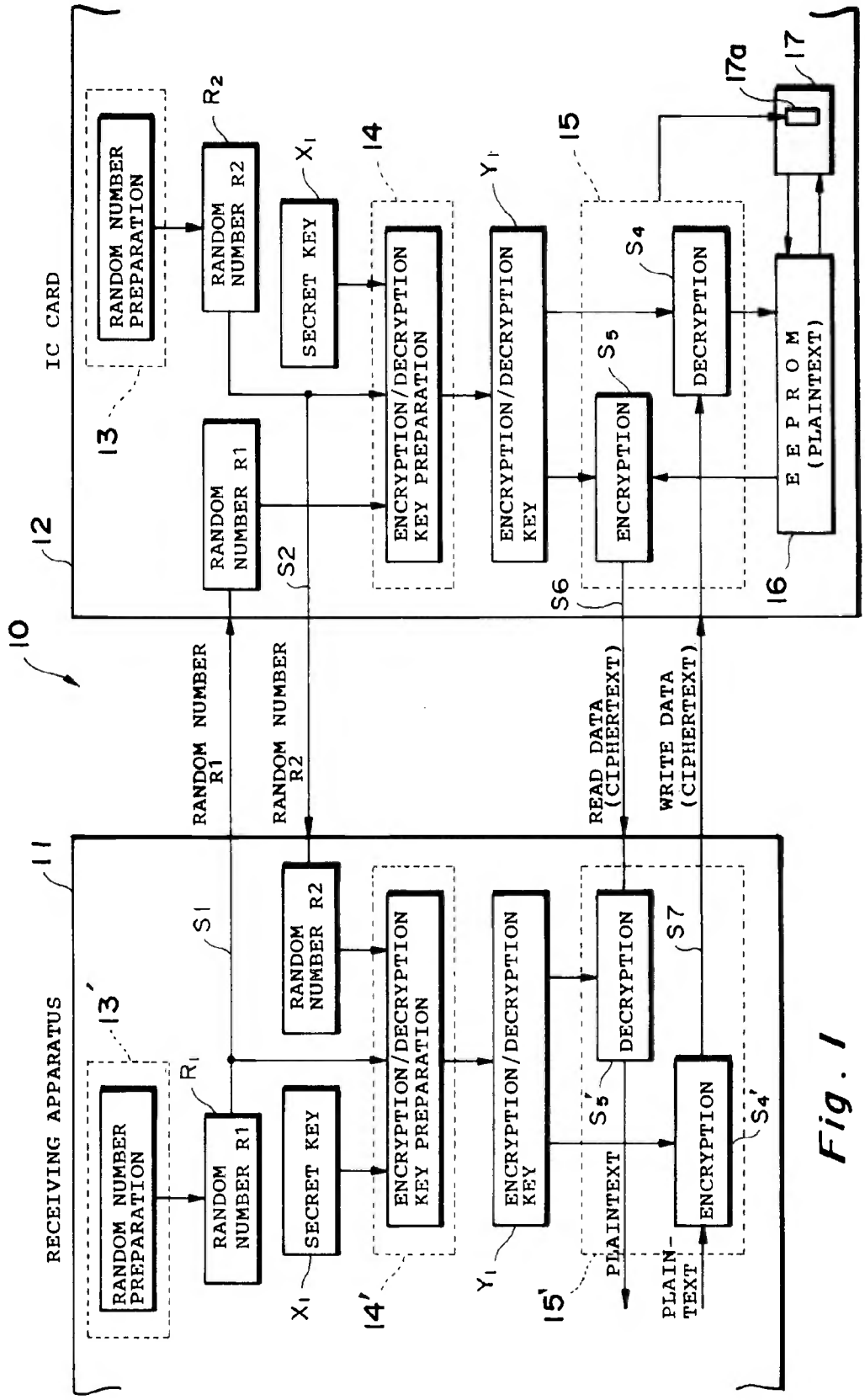

10
IC CARD
12
RANDOM NUMBER PREPARATION
13
RANDOM NUMBER R2
R2
RANDOM NUMBER R1
SECRET KEY
X1
14
ENCRYPTION/DECRYPTION KEY PREPARATION
ENCRYPTION/DECRYPTION KEY
Y1
15
ENCRYPTION
DECRYPTION
S4
S5
S6
S2
17a
17
EEPROM (PLAINTEXT)
16
RANDOM NUMBER R1
RANDOM NUMBER R2
READ DATA (CIPHERTEXT)
WRITE DATA (CIPHERTEXT)
11
RECEIVING APPARATUS
S1
13'
RANDOM NUMBER PREPARATION
R1
RANDOM NUMBER R1
SECRET KEY
X1
14'
ENCRYPTION/DECRYPTION KEY PREPARATION
RANDOM NUMBER R2
ENCRYPTION/DECRYPTION KEY
Y1
15'
DECRYPTION
S5'
S7
PLAINTEXT
PLAIN-TEXT
ENCRYPTION
S4'

Fig. 1

Fig. 2

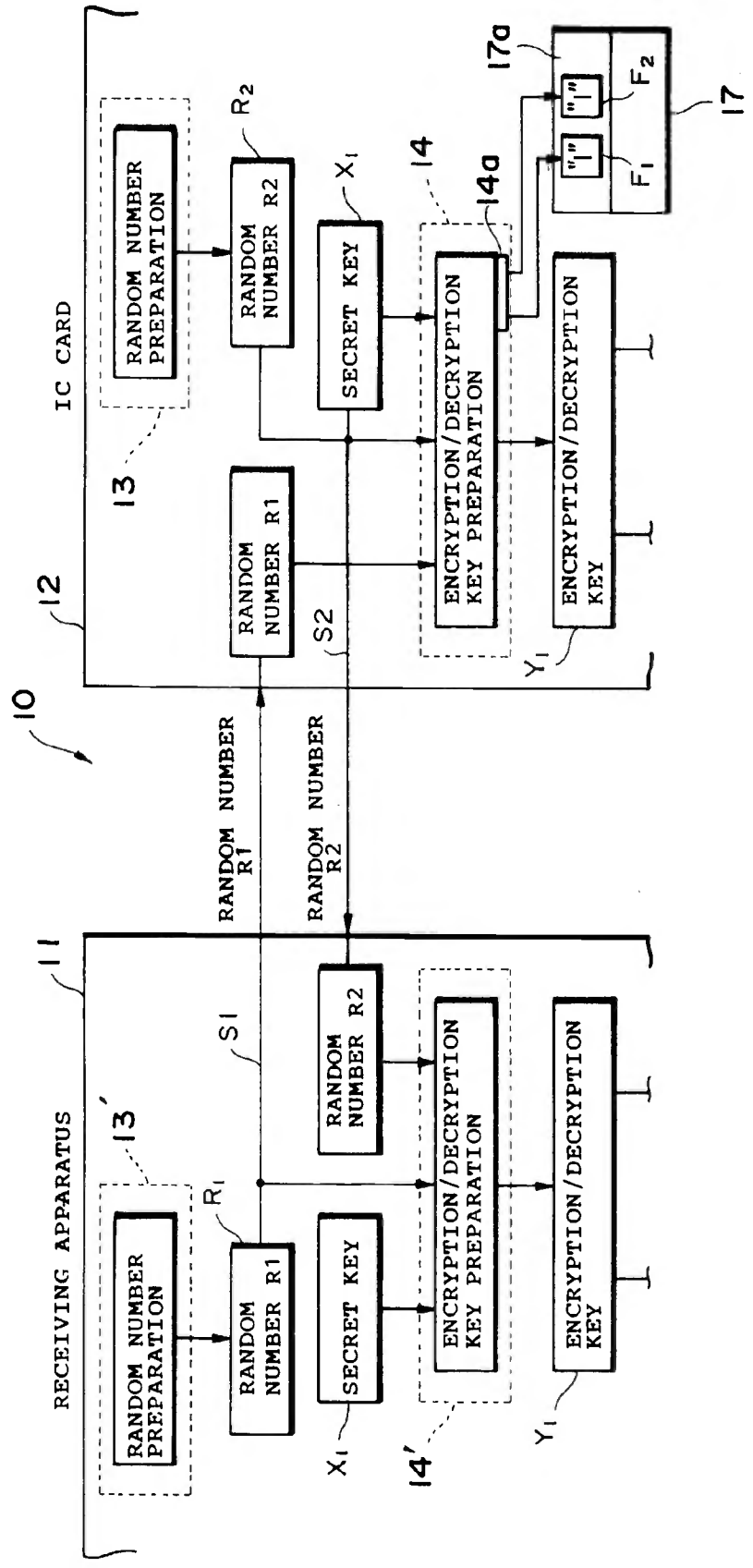

IC CARD
12
RANDOM NUMBER PREPARATION
13
RANDOM NUMBER R2
R2
RANDOM NUMBER R1
S2
SECRET KEY
X1
14
14a
ENCRYPTION/DECRYPTION KEY PREPARATION
ENCRYPTION/DECRYPTION KEY
Y1
17a
"1"
"1"
F2
F1
17
10
RANDOM NUMBER R1
RANDOM NUMBER R2
RECEIVING APPARATUS
11
S1
13'
RANDOM NUMBER PREPARATION
R1
RANDOM NUMBER R1
RANDOM NUMBER R2
SECRET KEY
X1
14'
ENCRYPTION/DECRYPTION KEY PREPARATION
ENCRYPTION/DECRYPTION KEY
Y1

Fig. 3

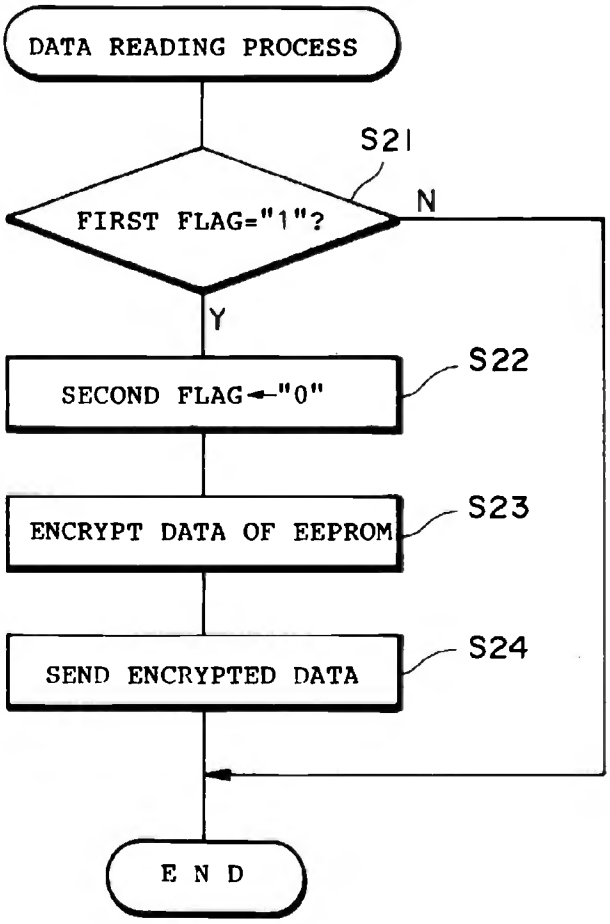

DATA READING PROCESS
S21
FIRST FLAG="1"?
N
Y
S22
SECOND FLAG←"0"
S23
ENCRYPT DATA OF EEPROM
S24
SEND ENCRYPTED DATA
E N D

Fig. 4

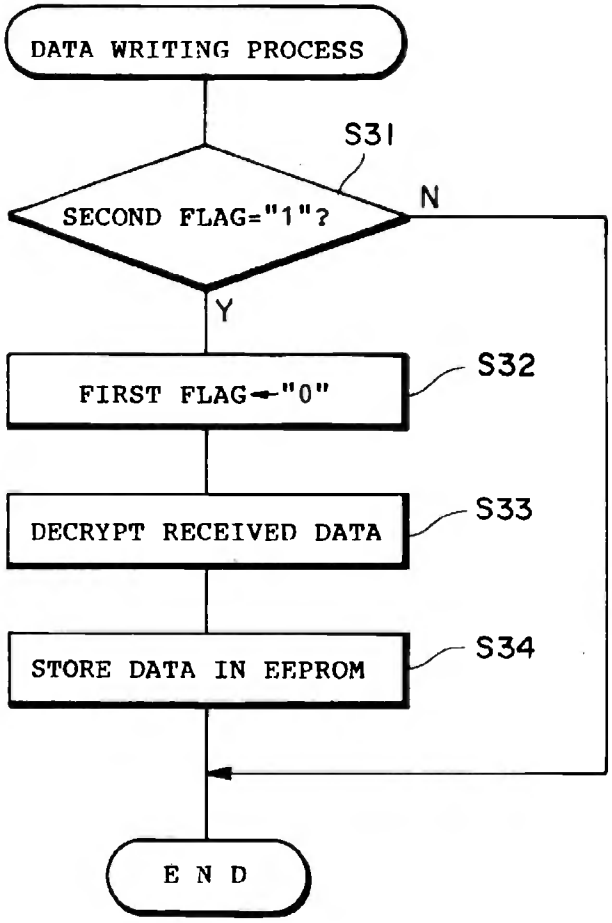

DATA WRITING PROCESS
S31
SECOND FLAG="1"?
N
Y
FIRST FLAG←"0"
S32
DECRYPT RECEIVED DATA
S33
STORE DATA IN EEPROM
S34
E N D

Fig. 5

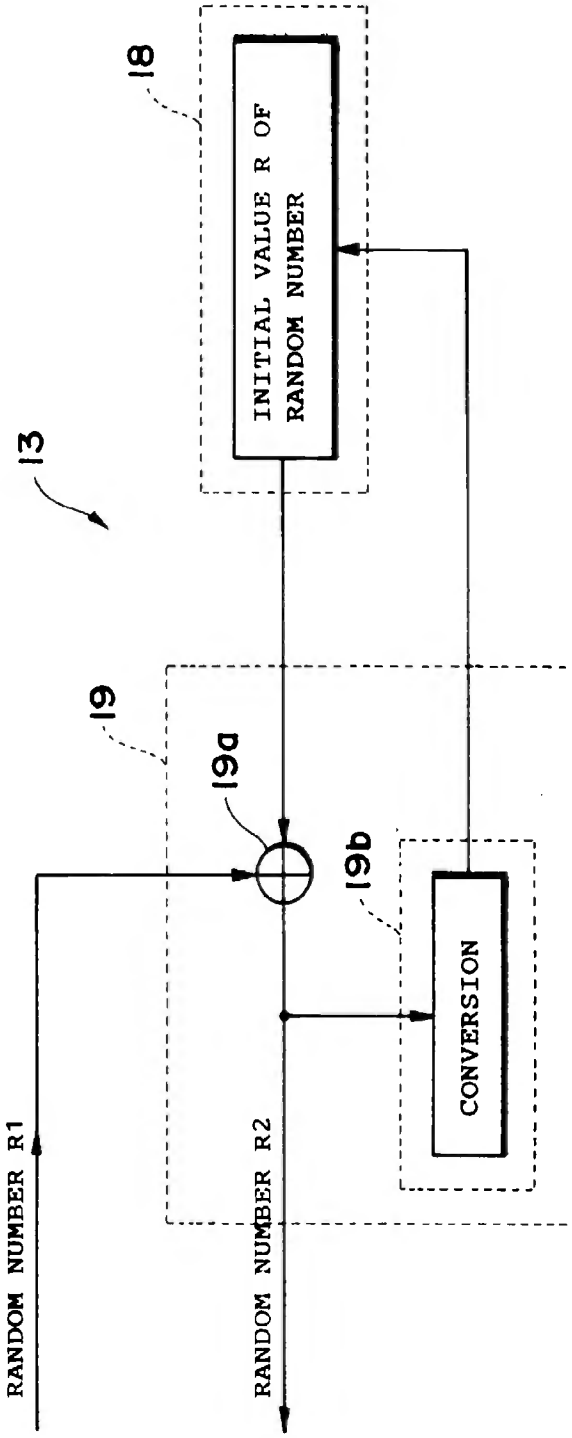
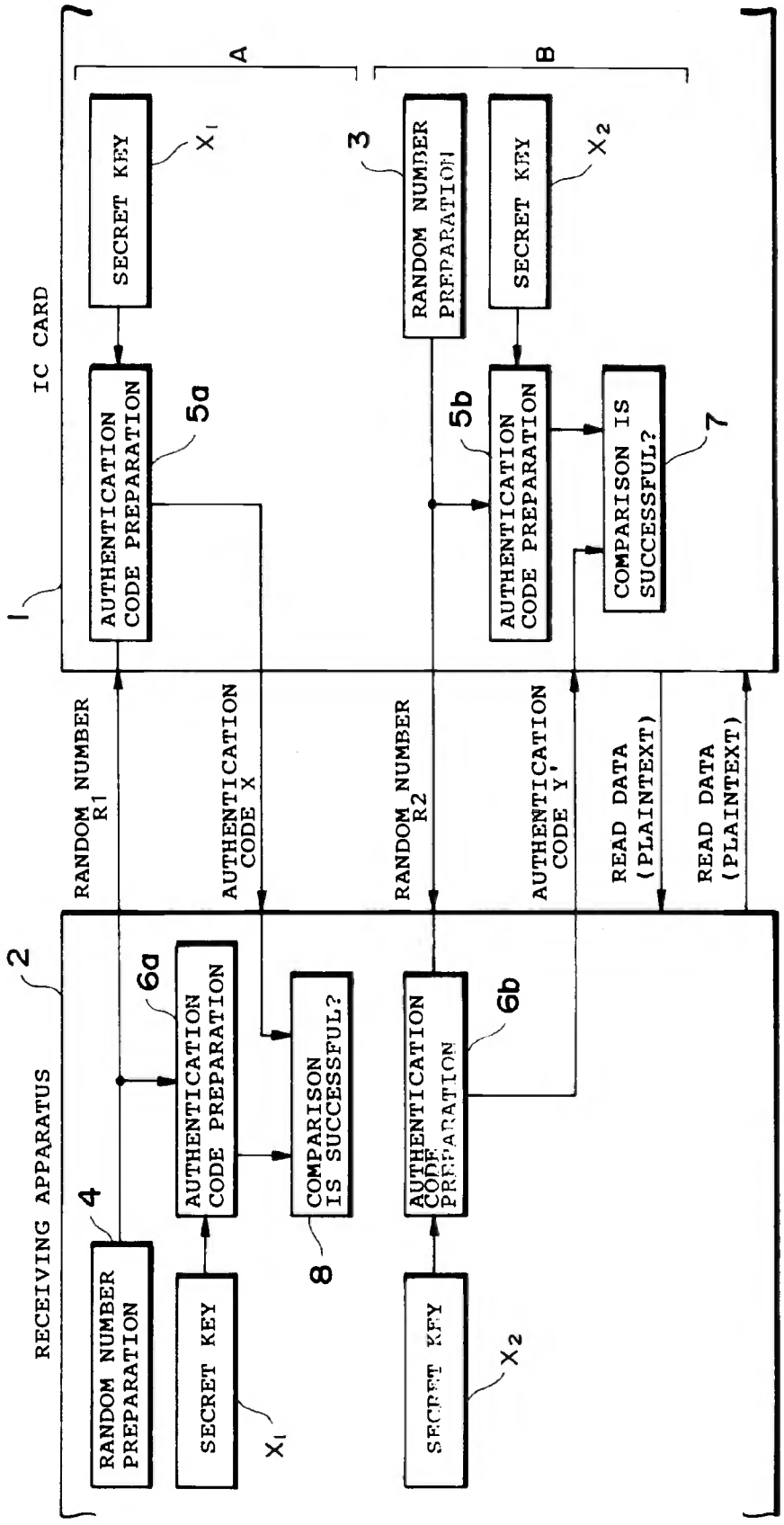

18
INITIAL VALUE R OF RANDOM NUMBER
13
19
19a
19b
CONVERSION
RANDOM NUMBER R1
RANDOM NUMBER R2

Fig. 6

PRIOR ART



(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 December 2000 (28.12.2000)

PCT

(10) International Publication Number
WO 00/79367 A1

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number: PCT/US00/17306

(22) International Filing Date: 21 June 2000 (21.06.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/140,608 23 June 1999 (23.06.1999) US
09/598,016 20 June 2000 (20.06.2000) US

(71) Applicant: **THE BRODIA GROUP** [US/US]; Suite 1530, 221 Main Street, San Francisco, CA 94105 (US).

(72) Inventors: **GOLDSTEIN, Theodore, Charles**; Suite 1530, 221 Main Street, San Francisco, CA 94105 (US). **RUBIN, Paul, H.**; Suite 1530, 221 Main Street, San Francisco, CA 94105 (US).

(74) Agent: **MEYER, Virginia, H.**; Meyer Intellectual Property Law, Suite 275, 475 Gate Five Road, Sausalito, CA 94965 (US).

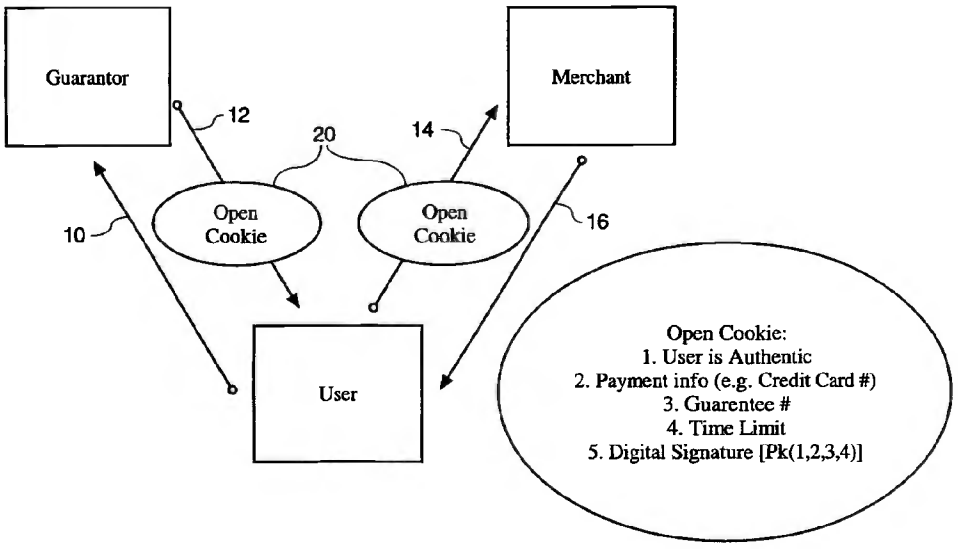
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AT (utility model), AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, CZ (utility model), DE, DE (utility model), DK, DK (utility model), DM, DZ, EE, EE (utility model), ES, FI, FI (utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KR (utility model), KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (utility model), SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*
— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR SECURE GUARANTEED TRANSACTIONS OVER A COMPUTER NETWORK



(57) Abstract: A trusted third-party authentication protocol that facilitates secure commercial transactions over computer networks between a merchant and a user. The method and system of the present invention generally involves three entities: 1) a guarantor, which acts as a trusted arbitrator, 2) a merchant, and 3) a consumer or user. In one embodiment, the guarantor authenticates users and provides authentication documents back to the user for use in transactions with merchants. The authentication document generated by the guarantor can be used as a means to validate a user's identity and/or to provide authorization/validation of a specific transaction.

WO 00/79367 A1

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND SYSTEM FOR SECURE GUARANTEED TRANSACTIONS OVER A COMPUTER NETWORK

FIELD OF THE INVENTION

The present invention relates to computer networks and, more particularly, relates to secure guaranteed transactions over computer networks.

BACKGROUND OF THE INVENTION

The increasing use of wide area networks such as the Internet has resulted in an explosion in the provision of on-line services. Computer users can access a vast wealth of information and services by utilizing a wide area network to establish a connection with other computers connected to the network.

The Internet is a global network of millions of computers belonging to various commercial and non-profit entities such as corporations, universities, and research organizations. The computer networks of the Internet are connected by gateways that handle data transfer and conversion of messages from a sending network to the protocols used by a receiving network. The Internet's collection of networks and gateways use the TCP/IP protocol. TCP/IP is an acronym for Transport Control Protocol/Internet Protocol, a software protocol developed by the Department of Defense.

Typically, the computers connected to a wide area network such as the Internet are identified as either servers or clients. A server is a computer that stores files that are available to other computers connected to the network. A client is a computer connected to the network that accesses the files and other resources provided by a server. To obtain information from a server, a client computer makes a request for a file or information located on the server using a specified protocol. Upon receipt of a properly formatted request, the server downloads the file to the client computer.

The World Wide Web is a system of Internet servers using specified Internet protocols and supporting specially formatted documents. The HyperText Transfer

Protocol ("HTTP") is the underlying protocol used by the World Wide Web. HTTP defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands. The other main standard of the World Wide Web is Hyper-Text Markup Language ("HTML"), which covers how documents and files are formatted and displayed. HTML supports links to other documents, as well as graphics, audio, and video files.

Users typically access the content contained on the Internet and the World Wide Web with an Internet Browser, which is a software application used to locate and display web pages. A Web page is a document on the World Wide Web. Every Web page or file on a web server is identified by a unique Uniform Resource Locator. A Uniform Resource Locator ("URL") is the global address of files and other resources on the Internet. The address indicates the protocol being used and specifies the IP address or the domain name where the file or resource is located. Typically, a URL identifies the name of the server and the path to a desired file on the server. For example, a URL for a web server may be constructed as follows: "http://<server>/<filepath>", where <server> identifies the server on which the file is located and <filepath> identifies the path to the file on the server. Thus, with the name of the server and the correct path to a file, a properly formatted URL accesses a desired file on a server connected to the World Wide Web.

There are myriad documents and files corresponding to a vast array of information and services accessible on the Internet. Indeed, the Internet has quickly become a means for not only obtaining information, but for conducting commercial transactions. For example, many consumers use the Internet to access on-line merchant sites and order books, compact discs and the like. Such commercial transactions conducted over the Internet require the consumer to provide some form of payment information, such as a credit card type, number and expiration date. However, one of the main disadvantages of conducting transactions over a computer network is the merchant's inability to verify that the user at the other end is who he purports to be and is authorized to make the purchase at issue with the method of payment he specifies. Accordingly, numerous authentication protocols involving computer networks are known in the art. See generally

Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2d Ed.* (John Wiley & Sons, Inc. 1996). For example, Kerberos is a trusted third-party authentication protocol. Kerberos is based on symmetric cryptography and shares different secret keys with every entity connected to the computer network. In Kerberos, knowledge of the secret key or a password that yields the secret key is proof of identity. However, the use of secret keys requires that all entities on the network must register and store secret keys with a central trusted authority. The large number of entities connected to the Internet, for example, renders the implementation of a central authenticating agency inapposite to the efficient operation of commercial transactions over the Internet. Accordingly, a need exists for a novel method and system for the secure guarantee of commercial transactions conducted over a large-scale computer network. The present invention substantially fulfills this need.

SUMMARY OF THE INVENTION

The present invention provides a trusted third-party authentication protocol that facilitates secure commercial transactions over computer networks between a merchant and a user. The method and system of the present invention generally involves three entities: 1) a guarantor, which acts as a trusted arbitrator, 2) a merchant, and 3) a consumer or user. In one embodiment, the guarantor authenticates users and provides authentication documents back to the user for use in transactions with merchants. The authentication document generated by the guarantor can be used as a means to validate a user's identity and/or to provide authorization/validation of a specific transaction. In one embodiment, when the user accesses the intended merchant's site on a computer network, the authentication document is uploaded to the merchant. In one embodiment, the merchant validates the authentication document and completes the transaction. In one embodiment, the present invention uses public or shared key encryption techniques and protocols to prepare and transmit encrypted authentication guarantees to merchants. According to one embodiment of the invention, these authentication documents are encrypted in a form that an intended merchant can decrypt. In one embodiment, the merchant decrypts the authentication document and processes the user's transaction

request if the authentication document is valid.

More specifically, the present invention, in one embodiment, provides a method for providing secure guaranteed transactions over a computer network. In one embodiment, a method according to the present invention generally comprises (a) authenticating a user and receiving a recipient merchant request; (b) generating an authentication document, if the user is authentic; (c) adding a digital signature to the authentication document; and, (d) transmitting the authentication document to the user. In one embodiment, the method further comprises encrypting the authentication document in a manner the recipient merchant can decrypt before transmitting it. According to one embodiment, when the user accesses the recipient merchant's site, the encrypted authentication document is uploaded. In embodiments involving encrypted authentication documents, the recipient merchant decrypts the authentication document with a private or shared key. The recipient merchant verifies the digital signature and may also validate the remainder of the authentication document.

In one embodiment, the encrypted authentication document is packaged as an open cookie before being transmitted to the user. The user's browser receives the open cookie, which is automatically uploaded to the recipient merchant site, when the user navigates to it. An open cookie, in one embodiment, is a string of text that is presented or available to other web sites and web servers. In another embodiment, the open cookie is a string of data inserted, typically as a parameter, into a URL or link corresponding to the intended or recipient merchant.

The authentication document, when included with a digital signature, acts as a guarantee by the issuing entity (here, called a guarantor) that the user in possession of it has been authenticated. For example, and in one embodiment, the guarantor may be a credit agency or other financial institution, while the user may have a credit or savings account with the guarantor. According to one embodiment of the invention, the user logs on to the guarantor's site and is prompted for a user name and a password for authentication purposes. Of course, other means of authentication may be used. The user's password or an encrypted representation thereof is checked against the password

value stored in the guarantor's database. If the values match, the user is authentic. In one embodiment, the user may then select a recipient merchant from which he desires to make a purchase and designate a method of payment, such as a credit card or an electronic check. The guarantor generates an authentication document, which includes for example, the identity of the user, a guarantee, a time stamp, and the user's credit card number and expiration date. In one embodiment, the authentication document also includes a transaction identifier or other challenge string supplied by the merchant that uniquely identifies the transaction being authenticated. As discussed more fully below, in one embodiment, the authentication document is signed by the guarantor and then encrypted with the public key of the recipient merchant.

As to the recipient merchant, the present invention contemplates that it trusts the guarantor and the guarantor's authentication of the user. Although the guarantor has been described as a financial institution, the guarantor, according to the invention, may be any entity trusted by the intended recipient merchant. Moreover, the functions performed by the guarantor may be incorporated into other electronic commerce related services, such as an electronic wallet provider. The recipient merchant may be a provider of goods, services and/or information. For example, the recipient merchant may be an on-line catalog site or an on-line services provider. Further, the recipient merchant may merely be a hosted application sitting on a web or Internet server allowing only authenticated users access to the application. According to this embodiment, the guarantor's authentication document serves as a form of ticket granting the user access to restricted services on a computer network. As one skilled in the art will recognize, the present invention centralizes the authentication of users and, therefore, minimizes the spread of password and other sensitive account information.

Another embodiment of the present invention is a system for providing secure guaranteed transactions over a computer network. In one embodiment, the system of the present invention comprises a database containing a list of user accounts and passwords corresponding to the user accounts and a server coupled to the database. In one embodiment, the server comprises (a) means for authenticating a user and receiving a

recipient merchant request; (b) means for generating an authentication document; (c) means for adding a digital signature to the authentication document; and (d) means for transmitting the authentication document to the user. One embodiment of the present invention contemplates that transactions are conducted between a user and a recipient merchant and that the merchant has at least one public encryption key. In this embodiment, the system further comprises (e) means for encrypting the authentication document with the public key of the recipient merchant. Other embodiments contemplate the user of symmetric encryption algorithms and an encryption key that is shared between the recipient merchant and the guarantor. In this embodiment, the server further comprises means for encrypting the authentication document with a shared encryption key. In one embodiment, the server further includes means for packaging the encrypted authentication document as an open cookie.

DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram broadly illustrating the flow of information among users, a guarantor and a merchant in one embodiment of the present invention.

Figure 2 is a block diagram illustrating a preferred embodiment of the present invention as applied to the Internet.

Figure 3 is a flow-chart diagram showing a preferred embodiment of a method according to the invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Figure 2 illustrates an embodiment of the present invention as applied to the Internet 60. Of course, the present invention can be applied across any computer network.

As Figure 2 illustrates, one embodiment of the present invention generally involves a guarantor, a merchant, and at least one user. In one embodiment, a user accesses an account stored on a database operably connected to servers 32 of guarantor site 30. Servers 32 receive and process authentication requests submitted by users. According to one embodiment, the user's account includes a user name, a password or an encrypted representation thereof, and payment information, such as a credit card type, number and

expiration date. The database can be any database known in the art. In one form, the database is implemented in hardware including a collection of computer programs enabling the storage, modification, and extraction of information on the database. Database hardware may range from personal computers (for small systems) to mainframes (for large systems). Servers 32 may be implemented in hardware or software, or preferably a combination of both. In one embodiment, the server is implemented in computer programs executing on programmable computers each comprising at least one processor, a data storage system (including volatile and non-volatile media), at least one input device, and at least one output device. In one preferred embodiment, servers 32 are web or Internet servers operably connected to the Internet 60.

In the embodiment shown in Figure 2, merchant site 40 is supported by web or Internet servers 42, which receive requests submitted by users and transmit files and other documents to users. In one embodiment, servers 42 are connected to the Internet 60.

As Figure 2 shows, one embodiment of the present invention works in conjunction with a conventional computer having Internet Browsing Software and a connection to the Internet. The user's computer can be any conventional personal computer known in the art. In one preferred embodiment, the user's computer is connected to the Internet via a dial-up connection or through a network line. Such communication could also be wireless. Additionally, suitable Internet browsers for use with the present invention include NETSCAPE NAVIGATOR®, MICROSOFT INTERNET EXPLORER®, and any other suitable browser, including those suitable for wireless technologies, employing protocols, such as the Wireless Access Protocol (WAP). The browser implemented on client computer 50 preferably includes the functionality required to receive and transmit open cookies. In addition, preferred browsers for use with the present invention support the SSL ("Secure Sockets Layer") protocol, the S-HTTP ("Secure HTTP") protocol, the TLS or the WTLS protocol, or any other similar protocol for transmitting confidential or private information over an open computer network. In one embodiment, communication of passwords and sensitive data, such as authentication documents, for example, between guarantor site 30 and client computer 50 (or between client computer 50 and merchant

site 40) employs the SSL protocol.

Figures 1 and 3 illustrate the operation of one embodiment of the present invention. In use, a user directs the browser on client computer 50 to guarantor site 30, as is conventional. In one form, server 32 prompts the user for a user name and a password to authenticate the user (Figure 3, step 100). Numerous authentication protocols are known in the art. The particular authentication protocol used is not critical to the invention. In one embodiment, the records corresponding to each user account contain the user name and a salted one-way hash of the user's password. Therefore, each user is authenticated by hashing the inputted password with the "salt" and comparing the result to the hash value stored in the user's record. If there is a match, the user is deemed to be authentic.

Upon proper authentication of the user, server 32, in one embodiment, prompts the user for the network address of the recipient merchant's site at which the user wishes to make a purchase or some other request. Server 32 accesses the record corresponding to the user and generates an authentication document (step 102). According to one embodiment, each user record may further include the user's full name, delivery and/or payment address, and preferred payment information. In one form, the authentication document includes the user's name, payment information (such as credit type, number and expiration date), a guarantee number, and a time limit beyond which the guarantee is no longer valid. In some embodiments, the authentication document may also include the user's preferred delivery address. In yet other embodiments, the guarantor may be a credit agency. According to this embodiment, the user can input an amount he contemplates spending. The guarantor can then include in the authentication document an indication that the user has a pre-approved spending limit of a certain specified amount. Of course, the authentication document can contain a vast array of information, including user preferences and additional hash values corresponding to the user's currently used machine number, as well as unique session IDs.

In step 104, guarantor 30 adds a digital signature to the authentication document. Various digital signature protocols are well known in the art. Digital signatures generally

involve the encryption of a document with a secret (symmetric encryption) or private (asymmetric encryption) key. In one embodiment, a digital signature is created with the guarantor's secret key using a key-dependent, one-way hash function. Under this embodiment, a recipient merchant also has knowledge of this secret key and is, therefore, able to validate the digital signature. In another embodiment, however, the digital signature is generated with the guarantor's private key using an asymmetric encryption algorithm. Suitable digital signature algorithms include, but are not limited to, RSA and DSA. Similar digital signature methods include hashing the authentication document and encrypting the resulting hash value with the guarantor's private key. However, any suitable digital signature can be incorporated into the encrypted authentication document. In one embodiment the signed authentication document is then transmitted to the user for use in a transaction with the intended merchant recipient.

In one embodiment, server 32 encrypts the signed authentication document in a form the recipient merchant can decrypt before sending it to the user. In one embodiment, server 32 encrypts the signed authentication document with a public key of the recipient merchant (step 106). Suitable public-key encryption algorithms include RSA and ElGamal. In another embodiment, the signed authentication document is encrypted with a symmetric encryption algorithm and a key shared between at guarantor site 30 and merchant site 40.

In one embodiment, the encrypted authentication document is packaged as an open cookie 20 (step 108) and transmitted to client computer 50 (step 110). In one embodiment, the open cookie 20 is transmitted simultaneously with a web page containing a hyperlink to the recipient merchant's web site. In one embodiment, the open cookie is a REDIRECT message comprising a string of data inserted as a parameter into a URL corresponding to merchant site 40. In one embodiment, the data is encrypted with either a private key (to be decrypted with a public key) or a shared key. In one embodiment, a user authenticates himself by entering a password and clicking on a link to a redirection engine at guarantor site 30. In one embodiment, the redirection engine computes the contents of the open cookie and composes a URL or link comprising the

recipient merchant site's address, the authentication document, and, in one embodiment, any other parameters required by merchant site 40.

When the user navigates his browser to recipient merchant site 40, the open cookie 20 is uploaded (step 112 & Figure 1, Ref. No. 14). In embodiments where the authentication document is encrypted, server 42 supporting merchant site 40 decrypts the authentication document in the open cookie 20 with the recipient merchant's private key or, depending on the embodiment, the shared key (step 114). Recipient merchant server 42 may then authenticate the guarantor's digital signature using conventional techniques known in the art (step 116). According to one embodiment, server 42 also validates the authentication document (step 118). Namely, and in one preferred embodiment, server 42 compares the actual time to the time limit specified in the authentication document. If the current time value is less than the time limit value, then the user's request may be processed. Validation step 118 may further include comparing the pre-approved credit limit specified in the authentication document to the purchase amount. In one form, server 42 can also use the relevant information contained in open cookie to fill in the various fields appearing in the order form provided by merchant site 40.

Additionally and in one embodiment, after the transaction is completed at merchant site 40, the user can be directed back to guarantor site 30. The user may specify another merchant site to which he wishes to navigate. In one embodiment, guarantor 30 re-encrypts the authentication document with the public key of the new specified merchant or generates an entirely new authentication document with a new time stamp or time limit. In other embodiments, merchant site 40 may re-encrypt the authentication document with the public key of the next intended recipient merchant.

Still further, embodiments of the present invention also contemplate the sharing of a private key and corresponding public key by more than one merchant site. Under this embodiment, each participating merchant site has knowledge of a shared private key. The guarantor encrypts the authentication document with the corresponding public key. Accordingly, each participating merchant has access to the encrypted authentication document and may validate it. Under this embodiment, the user may navigate among

participating merchant sites without the need for re-encryption of the authentication document.

In one embodiment, the authentication document can be used to authorize specific transactions. In one such embodiment of the present invention, merchant site 40 issues a unique merchant transaction identifier that guarantor site 30 incorporates into the authentication document. The merchant transaction identifier can be supplied in several ways. In one embodiment, merchant site 40 sends a merchant transaction identifier to guarantor site 30 by transmitting a HTTP REDIRECT message with transaction information, including the merchant transaction identifier to the user. For example, when a user transmits to merchant site 30 a request to order a product or service, merchant site 30 may construct a REDIRECT message. In one embodiment, the REDIRECT message comprises a page or other document including the computer network address of guarantor site 30 and the merchant transaction identifier and any other suitable transaction parameters. For example, the REDIRECT message may include a hypertext link or URL, such as "http://www.guarantor.com/?TRANSACTIONID=123456&AMOUNT=$789.00 &MERCHANT=www.merchant.com". When the user clicks on the link or otherwise activates the URL in the REDIRECT message, the merchant transaction identifier and any other parameters included in the message are transmitted to guarantor site 30 as part of the recipient merchant request. In one embodiment, guarantor site 30, if it approves the transaction, includes the merchant transaction identifier in the authentication document discussed above, signs the authentication document and transmits it to the user. In one embodiment, the authentication document is included as part of another REDIRECT message, which, when the link or URL in the message is activated, causes browser 62 to transmit the authentication document to merchant site 40. In one embodiment, merchant site 40 validates the authentication document, as discussed above, and completes the transaction.

In another embodiment, merchant site 40 provides the merchant transaction identifier in a field included in a form transmitted to the user. According to one

embodiment, the merchant transaction identifier is read from the form and transmitted to guarantor site 30 for inclusion in an authentication document. For example and in one embodiment, the merchant transaction identifier is embedded in a field specified in the Electronic Commerce Markup Language ("ECML") standard, an e-commerce standard available at www.ecml.org and published as the Internet Engineering Task Force RFC 2706 by Eastlake and Goldstein (incorporated by reference herein). The ECML standard specifies HTML fields that allow a merchant and a user to communicate. One such field, Ecom_ConsumerOrderID can be embedded as an HTML text field in an HTML form. For example, to represent the same merchant transaction identifier described above, a merchant transaction server embeds the following:

<input type=hidden name=Ecom_ConsumerOrderID value="transactionID=123456">. In one embodiment, the user's computer includes a software module, such as an electronic wallet, that scans the form for the merchant transaction identifier and transmits it to guarantor site 30. In one embodiment, the software module is configured to operate on order forms transmitted by merchants, including, in some embodiments, HTML forms complying with the ECML standard described above. In one such embodiment, the electronic wallet module, when the user directs it to complete the transaction, scans the order form for a merchant transaction identifier and transmits it to guarantor site 30. Guarantor site 30 composes and signs the authentication document, including the merchant transaction identifier, and transmits it to the user's computer. In one embodiment, the electronic wallet module completes the order form, includes or appends the authentication document, and transmits them back to the recipient merchant to complete the transaction. In embodiments involving the ECML standard, the software module inserts the authentication document into reserved fields in the order form. In another embodiment, guarantor site 30 composes and transmits to the user a URL including the computer network address corresponding to merchant site 40, the contents of the user's order, and the authentication document. When the user clicks on the URL, the order and the authentication document are transmitted to merchant site 40.

SUMMARY

With respect to the above-provided description, one skilled in the art will readily recognize that the present invention has application in a variety of contexts. The foregoing description illustrates the principles of the present invention and provides examples of its implementation. For example, although preferred embodiments are described as working in conjunction with an Internet browser, the present invention may be used in connection with any suitable software application for accessing files throughout a computer network. Accordingly, the above-provided description is not intended to limit the scope of the claims to the exact embodiments shown and described.

CLAIMS

What is claimed is:

1. A method for providing secure guaranteed transactions over a computer network, said transactions conducted between a user and a recipient merchant, said method comprising the steps of

(a) authenticating a user;

(b) receiving a recipient merchant request;

(c) generating an authentication document, if said user is authentic;

(d) adding a digital signature to said authentication document; and

(e) transmitting said authentication document to said user.

2. The method of claim 1, wherein said merchant has at least one encryption key, further comprising the step of

(f) encrypting said authentication document with the encryption key of said merchant before performing said transmitting step (e).

3. The method of claim 2 wherein said encryption key is a shared encryption key between a guarantor and said merchant.

4. The method of claim 2 wherein said encryption key is a public key of said merchant.

5. The method of claim 1, 2, 3, or 4 wherein said transmitting step (e) comprises the steps of

(e1) composing a link including a computer network address of the recipient merchant identified in said recipient merchant request and said authentication document; and,

(e2) transmitting said link to said user.

6. A method for providing secure guaranteed transactions over a computer network, said

transactions conducted between a user and a recipient merchant, said merchant having at least one public encryption key and a corresponding private key, said method comprising the steps of

(a) authenticating a user;

(b) receiving a recipient merchant request;

(c) generating an authentication document, if said user is authentic;

(d) adding a digital signature to said authentication document,

(e) encrypting said authentication document with the public key of said recipient merchant; and

(f) transmitting said encrypted authentication document to said user.

7. The method of claim 6 wherein said transmitting step (f) further comprises the steps of

(f1) packaging said encrypted authentication document as an open cookie; and

(f2) transmitting said open cookie to said user.

8. The method of claim 1, 5, 6 or 7 wherein said authentication document includes payment information corresponding to said user.

9. The method of claim 8 wherein said authentication document further includes a guarantee number and a time stamp.

10. The method of claim 8 wherein said authentication document further includes a guarantee number and a time limit.

11. The method of claim 1, 5, 6 or 7 wherein said authentication document is signed with a private key of said guarantor.

12. The method of claim 1, 5, 6 or 7 wherein said digital signature in step (d) is created with a secret key of said guarantor using a key-dependent one way hash-function, and

wherein said recipient merchant possesses said secret key.

13. The method of claim 1,5, 6 or 7 wherein said digital signature in step (d) is generated with a private key of said guarantor using an asymmetric algorithm.

14. The method of claim 1, 5, 6 or 7 wherein said adding step (d) comprises the steps of

(d1) hashing said authentication document;

(d2) creating a digital signature by applying a public-key algorithm to said hashed authentication document using a private key of said guarantor; and

(d3) adding said digital signature to said authentication document.

15. The method of claim 6 or 7 wherein said public key and said private key are shared by at least two recipient merchants.

16. A apparatus for providing secure guaranteed transactions over a computer network, said transactions conducted between a user and a recipient merchant, said user authenticated by a guarantor, said apparatus comprising

a database, said database containing a list user accounts and passwords, or encrypted representations thereof, corresponding to said user accounts;

a server operably coupled to said database, said server comprising

means for authenticating a user and receiving a recipient merchant request;

means for generating an authentication document;

means for adding a digital signature to said authentication document, and

means for transmitting said encrypted authentication document to said user.

17. The apparatus of claim 16 wherein said merchant having at least one public encryption key, further comprising means for encrypting said authentication document

with the public key of said recipient merchant.

18. The apparatus of claim 16 or 17 wherein said transmitting means further comprises means for packaging said authentication document as an open cookie.

19. The apparatus of claim 16 or 17 further comprising means for composing a link including a computer network address corresponding to the recipient merchant and the authentication document.

20. A method for providing secure guaranteed transactions over a computer network, said transactions conducted between a user and a recipient merchant, said merchant having at least one public encryption key, said user authenticated by a guarantor, said method comprising the steps of

(a) receiving an authentication document from a user, said authentication document encrypted with a public encryption key of said recipient merchant and including a digital signature;

(b) decrypting said authentication document with the corresponding private key of said recipient merchant;

(c) authenticating said digital signature; and

(d) processing the user's request, if said digital signature is authentic and said authentication document is valid.

21. The method of claim 20 wherein said authentication document is packaged as an open cookie and wherein said receiving step (a) comprises uploading said open cookie.

22. The method of claim 20 or 21 wherein said authentication document includes payment information, wherein said user request is a purchase request, and wherein said processing step (d) further comprises using said payment information in said authentication document to complete a transaction.

23. The method of claim 1 wherein said recipient merchant request includes a merchant transaction identifier; and wherein said authentication document generated in step (c) includes said merchant transaction identifier.

24. The method of claim 23 wherein said recipient merchant request originates from a merchant as a redirect message transmitted to said user.

25. A method for providing secure guaranteed transactions over a computer network, said transactions conducted between a user and a recipient merchant, said method comprising the steps of

(a) authenticating a user;

(b) receiving a recipient merchant request, said recipient merchant request including a merchant transaction identifier;

(c) generating an authentication document including said merchant transaction identifier, if said user is authentic;

(d) adding a digital signature to said authentication document; and

(e) transmitting said authentication document to said user.

26. The method of claim 25 wherein said recipient merchant request originates from a merchant as a redirect message transmitted to said user.

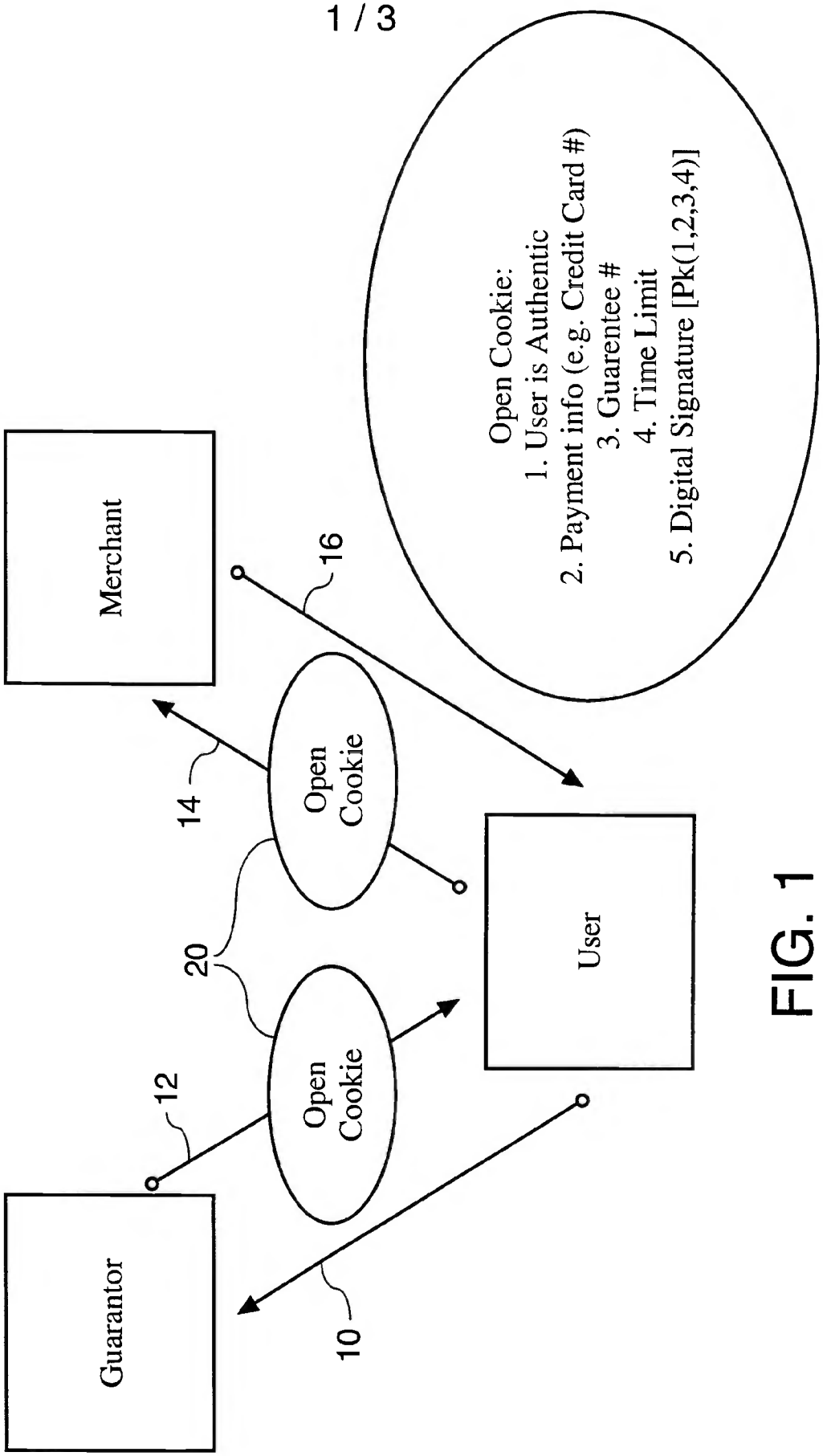

Open Cookie:
1. User is Authentic
2. Payment info (e.g. Credit Card #)
3. Guarentee #
4. Time Limit
5. Digital Signature [Pk(1,2,3,4)]
Merchant
16
14
Open Cookie
20
Open Cookie
User
12
Guarantor
10

FIG. 1

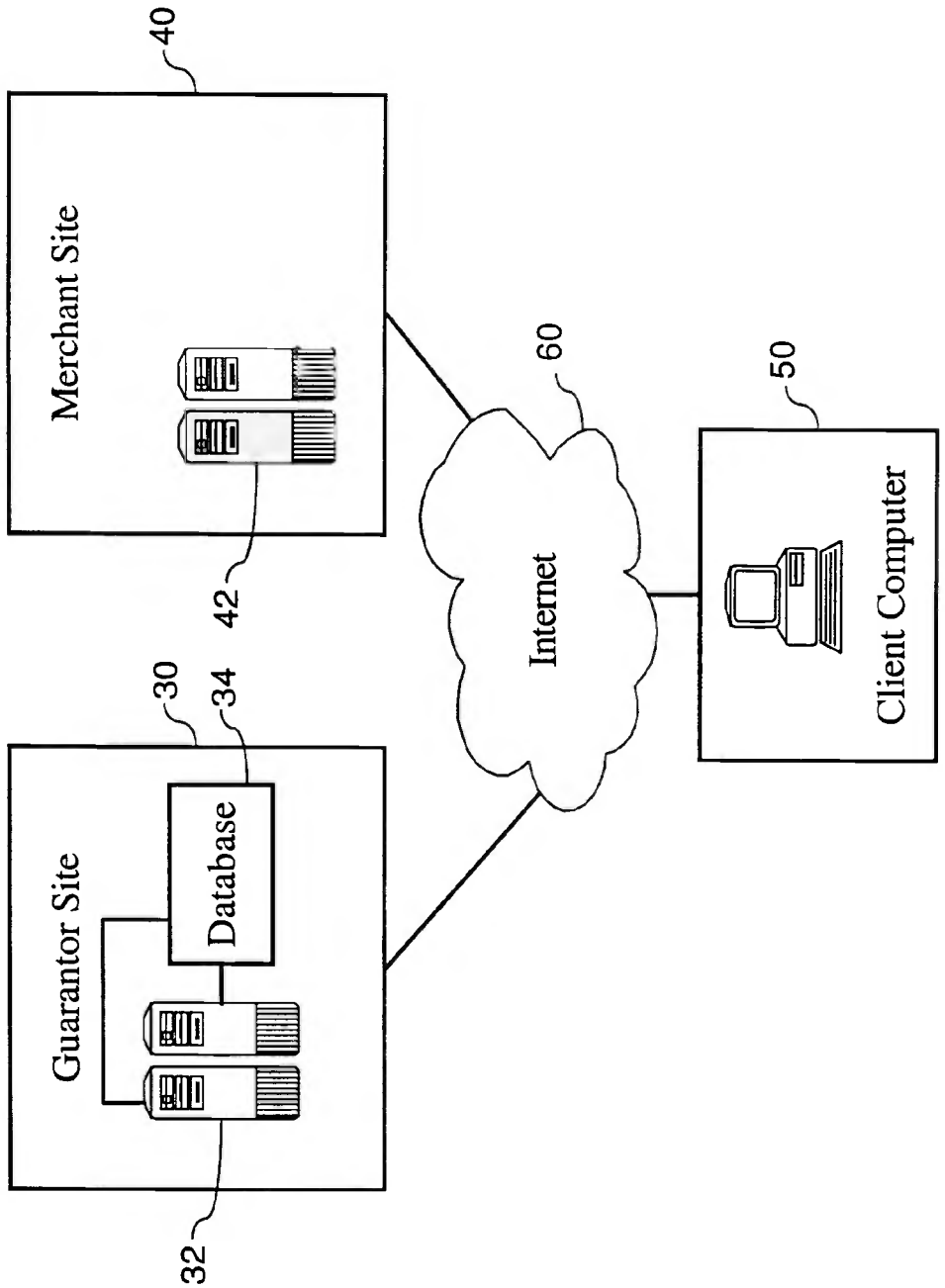

40
Merchant Site
42
60
50
Internet
Client Computer
30
34
Guarantor Site
Database
32

FIG. 2

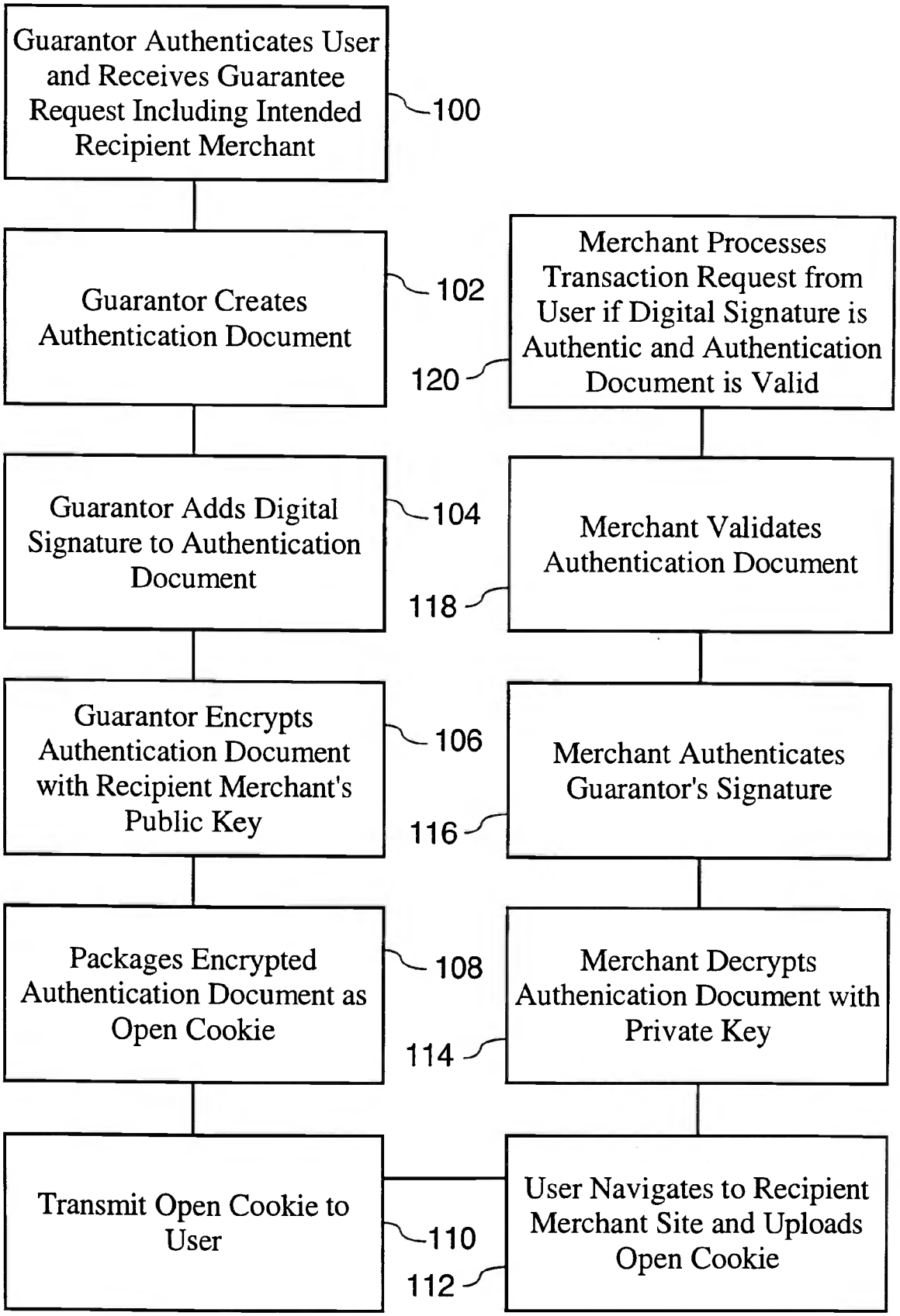

Guarantor Authenticates User and Receives Guarantee Request Including Intended Recipient Merchant
100
Guarantor Creates Authentication Document
102
Guarantor Adds Digital Signature to Authentication Document
104
Guarantor Encrypts Authentication Document with Recipient Merchant's Public Key
106
Packages Encrypted Authentication Document as Open Cookie
108
Transmit Open Cookie to User
110
User Navigates to Recipient Merchant Site and Uploads Open Cookie
112
Merchant Decrypts Authenication Document with Private Key
114
Merchant Authenticates Guarantor's Signature
116
Merchant Validates Authentication Document
118
Merchant Processes Transaction Request from User if Digital Signature is Authentic and Authentication Document is Valid
120

FIG. 3

INTERNATIONAL SEARCH REPORT

Intern al Application No
PCT/US 00/17306

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F G07F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 96 42041 A (OPEN MARKET INC) 27 December 1996 (1996-12-27)	1-3,5, 11-13, 16,19, 23-26
Y		4,6, 8-10,17, 20,22
A	abstract page 9 -page 19 figure 3	14

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 October 2000	18/10/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Jacobs, P

1

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inter ıal Application No
PCT/US 00/17306

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 98 58306 A (OYLER SCOTT ;GUTHRIE JOHN (US); TECHWAVE INC (US); KRISHNAN GANAPA) 23 December 1998 (1998-12-23)	4,6, 8-10,17, 20,22
A	abstract page 10, line 8 -page 15, line 3 page 30, line 10 -page 43, line 27 claim 25 figures 1-3	1-3,5, 11-16, 19,23,25
X	WO 98 47091 A (TRANSACTOR NETWORKS INC) 22 October 1998 (1998-10-22)	1-3,5, 8-13,16, 19,23,25
A	page 9, line 4 -page 16, line 27 page 29, line 9 -page 37, line 10 page 45, line 10 -page 46, line 12 figures 1,2	4,6,14, 17,20,22
X	US 5 802 497 A (MANASSE MARK S) 1 September 1998 (1998-09-01)	1,2,5, 8-10,16, 19,25
A	column 2, line 55 -column 6, line 35 claims 1,2 figures 1-5	3,4,6, 11-14, 17,20, 22,23
A	US 5 875 296 A (AULT MICHAEL BRADFORD ET AL) 23 February 1999 (1999-02-23)	1,5-7,9, 10,16, 18,19, 21,25
	abstract column 6, line 48 -column 9, line 58	

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

Internal Application No
PCT/US 00/17306

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9642041 A	27-12-1996	US 5708780 A	13-01-1998
		US 5812776 A	22-09-1998
		AU 694367 B	16-07-1998
		AU 5936796 A	09-01-1997
		CA 2221506 A	27-12-1996
		EP 0830774 A	25-03-1998
		JP 11507752 T	06-07-1999
WO 9858306 A	23-12-1998	US 6073124 A	06-06-2000
		AU 8150598 A	04-01-1999
WO 9847091 A	22-10-1998	US 6119229 A	12-09-2000
		AU 721052 B	22-06-2000
		AU 6963298 A	11-11-1998
		EP 0974118 A	26-01-2000
US 5802497 A	01-09-1998	BR 9606450 A	30-09-1997
		EP 0796480 A	24-09-1997
		IL 117195 A	20-06-1999
		JP 2984731 B	29-11-1999
		JP 9510814 T	28-10-1997
		WO 9703423 A	30-01-1997
US 5875296 A	23-02-1999	JP 3003997 B	31-01-2000
		JP 10257048 A	25-09-1998

PATENTS
Attorney Docket No. SFC-2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION

Applicants	:	Rick L. Orsini et al.
Application No.	:	11/258,839 Confirmation No. 2420
Filed	:	October 25, 2005
For	:	SECURE DATA PARSER METHOD AND SYSTEM
Art Unit	:	2432
Examiner	:	Samson B. Lemma

New York, N.Y. 10036
May 7, 2009

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

Sir:

Pursuant to 37 C.F.R. §§ 1.56 and 1.97, applicants hereby make the documents listed on the accompanying Form PTO/SB/08A of record in the above-identified patent application. Pursuant to 37 C.F.R. § 1.98(a)(2)(ii) no copies of the "U.S. Patent Documents" and "U.S. Patent Publications" are provided. Copies of the “Foreign” and "Non-Patent Literature” documents are submitted herewith. Applicants respectfully request that these documents be printed on any patent which may issue on this application. Applicants request that a copy of Form PTO/SB/08A, as considered and

initialed by the Examiner, be returned with the next communication.

Pursuant to 37 C.F.R. §1.97(b)(3), this Supplemental Information Disclosure Statement is filed before the mailing of a first Office action on the merits. Accordingly, no fee is believed to be due in connection with this submission. However, if for any reason a fee is due, the Director is hereby authorized to charge payment of any fee required in connection with this Statement to Deposit Account No. 06-1075. Consideration of the foregoing in relation to this patent application is respectfully requested.

Applicants reserve the right to establish the patentability of the claimed invention over any of the information provided herewith, and/or to prove that this information may not be prior art, and/or to prove that this information may not be enabling for the teachings purportedly offered.

Respectfully submitted,

/Brian E. Mack/

Brian E. Mack
Registration No. 57,189
Attorney for Applicants
Customer No. 1473
(212) 596-9000

Electronic Acknowledgement Receipt

EFS ID:	5295368
Application Number:	11258839
International Application Number:	
Confirmation Number:	2420
Title of Invention:	Secure data parser method and system
First Named Inventor/Applicant Name:	Rick L. Orsini
Customer Number:	01473
Filer:	Brian Edward Mack
Filer Authorized By:	
Attorney Docket Number:	SFC-2
Receipt Date:	07-MAY-2009
Filing Date:	25-OCT-2005
Time Stamp:	17:09:48
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	SIDS.pdf	5838 d386196e8accf22c7a4c72b7a4caef3aabf35e08	no	2

Warnings:	
Information:	

2	NPL Documents	CachinOnlineSecretSharing.pdf	4314537 9c81865e6a1cd2f971dd3f81887961a82550836d	no	9
Warnings:					
Information:					
3	NPL Documents	Crescenzo-Non-InteractiveandNon-MalleableCommitment.pdf	7001839 06387a380bb5802304ff54d6f8580774b965a5de	no	10
Warnings:					
Information:					
4	NPL Documents	Mayer-GeneralizedSecretSharing.pdf	5046510 e98139bdd98ddbc21efa2bbf6d5bebee9872c6f7	no	15
Warnings:					
Information:					
5	Foreign Reference	WO2009035674A1.pdf	9001438 79e476963f023048008b52f209f6b5973abb2d3a	no	242
Warnings:					
Information:					
6	Information Disclosure Statement (IDS) Filed (SB/08)	SFC_2_SB08.pdf	99155 a7bb8127bf4e26314963b23c7580c22a9168269e	no	2
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
		Total Files Size (in bytes):	25469317		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

<u>New Applications Under 35 U.S.C. 111</u>
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

<u>National Stage of an International Application under 35 U.S.C. 371</u>
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

<u>New International Application Filed with the USPTO as a Receiving Office</u>
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/08a (01-08)
Approved for use through 04/31/2009. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 2

Complete if Known	
Application Number	11/258,839 (Conf. No. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2432
Examiner Name	Samson B. Lemma
Attorney Docket Number	SFC-2

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		7,069,451 B1	06-27-2006	Ginter, et al.	
		7,391,865	06-24-2008	Orsini et al.	

U.S. PATENT PUBLICATIONS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		2002/0091640 A1	07-11-2002	Gupta	
		2003/0058274 A1	03-27-2003	Hill et al.	
		2006/0282681 A1	12-14-2006	Scheidt et al.	
		2007/0160198	07-12-2007	Orsini et al.	
		2008/0137857 A1	06-12-2008	Bellare et al.	
		2008/0183992 A1	07-31-2008	Martin et al.	
		2008/0244277 A1	10-02-2008	Orsini et al.	
		2009/0097661 A1	04-16-2009	Orsini et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3] "Number[4] "Kind Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]
		WO 2009/035674 A1	03-19-2009	Security First Corporation		

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
		Crescenzo, G. et al., "Non-Interactive and Non-Malleable Commitment," Proceedings of the 30th Annual ACM Symposium on Theory of Computing. Dallas, TX, May 23-26, 1998, [Proceedings of the 30th Annual ACM Symposium on Theory of Computing], New York, NY: ACM, US, PAGES 141-150; XP000970902; ISBN: 978-0-89791-962-3.	

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08a (01-08)
Approved for use through 04/31/2009. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet	2	of	2

Complete if Known	
Application Number	11/258,839 (Conf. No. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2432
Examiner Name	Samson B. Lemma
Attorney Docket Number	SFC-2

		Christian Cachin, "On-Line secret Sharing," Cryptography and Coding. IMA Conference, Proceedings, XX, XX, December 18, 1995, pages 190-198, XP002137681.	
		Mayer, A. et al., "Generalized Secret Sharing and Group-Key Distribution Using Short Keys," Compression and Complexity of Sequences 1997, Proceedings Salerno, Italy, June 11-13, 1997, Los Alamitos, CA, USA, IEEE Comput,. SOC, US, June 11, 1997, pages 30-44, XP010274905, ISBN: 978-0-8186-8132-5.	

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 March 2009 (19.03.2009)

PCT

(10) International Publication Number
WO 2009/035674 A1

(51) International Patent Classification:
H04L 9/00 (2006.01)

(21) International Application Number:
PCT/US2008/010677

(22) International Filing Date:
12 September 2008 (12.09.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/993,756 14 September 2007 (14.09.2007) US

(71) Applicant *(for all designated States except US)*: **SECURITY FIRST CORPORATION** [US/US]; 22362 Gilberto #130, Rancho Santa Margarita, California 92688 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **ORSINI, Rick, L.** [US/US]; 2100 Kings Forest Lane, Flower Mound, Texas 75028 (US). **O'HARE, Mark, S.** [US/US]; 8 Kennedy Court, Coto De Caza, California 92679 (US). **DAVENPORT, Roger** [US/US]; 3204 Cr 4119, Campbell, Texas 75422 (US).

(74) Agents: **INGERMAN, Jeffrey, H.** et al.; Ropes & Gray LLP, 1211 Avenue of the Americas, New York, New York 10036-8704 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(54) Title: SYSTEMS AND METHODS FOR MANAGING CRYPTOGRAPHIC KEYS

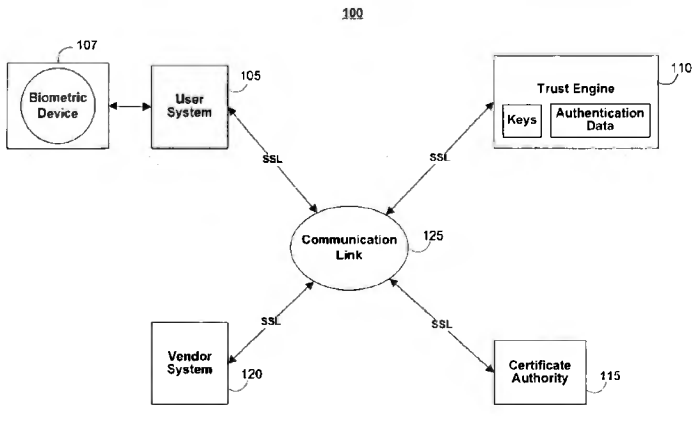


FIG. 1

(57) Abstract: A common interface for managing cryptographic keys is provided. A request to manage a cryptographic key may be received in a first interface format, translated to a common interface format, and then executed remotely from the first interface. Return arguments may then be translated from the common interface format to a format compatible with the first interface and communicated securely to the first interface. The cryptographic keys may be used in connection with a secure data parser that secures data by randomly distributing data within a data set into two or more shares.

WO 2009/035674 A1

SYSTEMS AND METHODS FOR MANAGING CRYPTOGRAPHIC KEYS

Cross-Reference to Related Applications

[0001] This application claims the benefit of U.S. provisional application No. 60/993,756, filed on September 14, 2007, which is hereby incorporated by reference herein in its entirety.

Field of the Invention

[0002] The present invention relates in general to a system for securing data from unauthorized access or use. More particularly, the present invention relates to a common interface for supporting cryptographic keys.

Background of the Invention

[0003] In today's society, individuals and businesses conduct an ever-increasing amount of activities on and over computer systems. These computer systems, including proprietary and non-proprietary computer networks, are often storing, archiving, and transmitting all types of sensitive information. Thus, an ever-increasing need exists for ensuring data stored and transmitted over these systems cannot be read or otherwise compromised.

[0004] One common solution for securing computer systems is to provide login and password functionality.

However, password management has proven to be quite costly with a large percentage of help desk calls relating to password issues. Moreover, passwords provide little security in that they are generally stored in a file susceptible to inappropriate access, through, for example, brute-force attacks.

[0005] Another solution for securing computer systems is to provide cryptographic infrastructures. Cryptography, in general, refers to protecting data by transforming, or encrypting, it into an unreadable format. Only those who possess the key(s) to the encryption can decrypt the data into a useable format. Cryptography is used to identify users, e.g., authentication, to allow access privileges, e.g., authorization, to create digital certificates and signatures, and the like. One popular cryptography system is a public key system that uses two keys, a public key known to everyone and a private key known only to the individual or business owner thereof. Generally, the data encrypted with one key is decrypted with the other and neither key is recreatable from the other.

[0006] Unfortunately, even the foregoing typical public-key cryptographic systems are still highly reliant on the user for security. For example, cryptographic systems issue the private key to the user, for example, through the user's browser. Unsophisticated users then generally store the private key on a hard drive accessible to others through an open computer system, such as, for example, the Internet. On the other hand, users may choose poor names for files containing their private key, such as, for example, "key." The result of the foregoing and

other acts is to allow the key or keys to be susceptible to compromise.

[0007] In addition to the foregoing compromises, a user may save his or her private key on a computer system configured with an archiving or backup system, potentially resulting in copies of the private key traveling through multiple computer storage devices or other systems. This security breach is often referred to as "key migration." Similar to key migration, many applications provide access to a user's private key through, at most, simple login and password access. As mentioned in the foregoing, login and password access often does not provide adequate security.

[0008] One solution for increasing the security of the foregoing cryptographic systems is to include biometrics as part of the authentication or authorization. Biometrics generally include measurable physical characteristics, such as, for example, finger prints or speech that can be checked by an automated system, such as, for example, pattern matching or recognition of finger print patterns or speech patterns. In such systems, a user's biometric and/or keys may be stored on mobile computing devices, such as, for example, a smartcard, laptop, personal digital assistant, or mobile phone, thereby allowing the biometric or keys to be usable in a mobile environment.

[0009] The foregoing mobile biometric cryptographic system still suffers from a variety of drawbacks. For example, the mobile user may lose or break the smartcard or portable computing device, thereby having his or her access to potentially important data entirely cut-off. Alternatively, a malicious person may steal the mobile user's smartcard or portable

computing device and use it to effectively steal the mobile user's digital credentials. On the other hand, the portable-computing device may be connected to an open system, such as the Internet, and, like passwords, the file where the biometric is stored may be susceptible to compromise through user inattentiveness to security or malicious intruders.

[0010] In addition, there are many ways to securely create, store, and manage individual cryptographic keys. For example, some applications may store a user's cryptographic keys in a key store or other data structure. The cryptographic keys in a user's key store may be accessed by a variety of applications. Some applications, however, may be incompatible with other applications or may compromise the security of a user's cryptographic keys by, for example, exposing one or more of the keys to corruption or unauthorized or unsecured access.

Summary of the Invention

[0011] Accordingly, a cryptographic system whose security is user-independent while still supporting mobile users is provided.

[0012] In addition, a common interface, for example, an application programming interface ("API"), is also provided that is capable of supporting multiple interfaces to a variety of cryptographic key providers and present cryptographic keys obtained from these key providers to a secure parser engine to be used, for example, for securing data for storage or transmission. Such secure parser engines are described in more detail in Orsini et al. U.S. Patent No. 7,391,865, U.S. Patent

Application No. 11/258,839, filed October 25, 2005, and U.S. Patent Application No. 11/602,667, filed November 20, 2006, all of which are hereby incorporated by reference herein in their entireties.

[0013] Accordingly, one aspect of the present invention is to provide a method for securing virtually any type of data from unauthorized access or use. The method comprises one or more steps of parsing, splitting and/or separating the data to be secured into two or more parts or portions. The method also comprises encrypting the data to be secured. Encryption of the data may be performed prior to or after the first parsing, splitting and/or separating of the data. In addition, the encrypting step may be repeated for one or more portions of the data. Similarly, the parsing, splitting and/or separating steps may be repeated for one or more portions of the data. The method also optionally comprises storing the parsed, split and/or separated data that has been encrypted in one location or in multiple locations. This method also optionally comprises reconstituting or re-assembling the secured data into its original form for authorized access or use. This method may be incorporated into the operations of any computer, server, engine or the like, that is capable of executing the desired steps of the method.

[0014] Another aspect of the present invention provides a system for securing virtually any type of data from unauthorized access or use. This system comprises a data splitting module, a cryptographic handling module, and, optionally, a data assembly module. The system may, in one embodiment, further

comprise one or more data storage facilities where secure data may be stored.

[0015] Accordingly, one aspect of the invention is to provide a secure server, or trust engine, having server-centric keys, or in other words, storing cryptographic keys and user authentication data on a server. According to this embodiment, a user accesses the trust engine in order to perform authentication and cryptographic functions, such as, but not limited to, for example, authentication, authorization, digital signing and generation, storage, and retrieval of certificates, encryption, notary-like and power-of-attorney-like actions, and the like.

[0016] Another aspect of the invention is to provide a reliable, or trusted, authentication process. Moreover, subsequent to a trustworthy positive authentication, a wide number of differing actions may be taken, from providing cryptographic technology, to system or device authorization and access, to permitting use or control of one or a wide number of electronic devices.

[0017] Another aspect of the invention is to provide cryptographic keys and authentication data in an environment where they are not lost, stolen, or compromised, thereby advantageously avoiding a need to continually reissue and manage new keys and authentication data. According to another aspect of the invention, the trust engine allows a user to use one key pair for multiple activities, vendors, and/or authentication requests. According to yet another aspect of the invention, the trust engine performs at least one step of cryptographic processing, such as, but not limited to, encrypting, authenticating, or

signing, on the server side, thereby allowing clients or users to possess only minimal computing resources.

[0018] According to yet another aspect of the invention, the trust engine includes one or multiple depositories for storing portions of each cryptographic key and authentication data. The portions are created through a data splitting process that prohibits reconstruction without a predetermined portion from more than one location in one depository or from multiple depositories. According to another embodiment, the multiple depositories may be geographically remote such that a rogue employee or otherwise compromised system at one depository will not provide access to a user's key or authentication data.

[0019] According to yet another embodiment, the authentication process advantageously allows the trust engine to process multiple authentication activities in parallel. According to yet another embodiment, the trust engine may advantageously track failed access attempts and thereby limit the number of times malicious intruders may attempt to subvert the system.

[0020] According to yet another embodiment, the trust engine may include multiple instantiations where each trust engine may predict and share processing loads with the others. According to yet another embodiment, the trust engine may include a redundancy module for polling a plurality of authentication results to ensure that more than one system authenticates the user.

[0021] Therefore, one aspect of the invention includes a secure cryptographic system, which may be remotely accessible, for storing data of any type, including, but not limited to, a plurality of private

cryptographic keys to be associated with a plurality of users. The cryptographic system associates each of the plurality of users with one or more different keys from the plurality of private cryptographic keys and performs cryptographic functions for each user using the associated one or more different keys without releasing the plurality of private cryptographic keys to the users. The cryptographic system comprises a depository system having at least one server which stores the data to be secured, such as a plurality of private cryptographic keys and a plurality of enrollment authentication data. Each enrollment authentication data identifies one of multiple users and each of the multiple users is associated with one or more different keys from the plurality of private cryptographic keys. The cryptographic system also may comprise an authentication engine which compares authentication data received by one of the multiple users to enrollment authentication data corresponding to the one of multiple users and received from the depository system, thereby producing an authentication result. The cryptographic system also may comprise a cryptographic engine which, when the authentication result indicates proper identification of the one of the multiple users, performs cryptographic functions on behalf of the one of the multiple users using the associated one or more different keys received from the depository system. The cryptographic system also may comprise a transaction engine connected to route data from the multiple users to the depository server system, the authentication engine, and the cryptographic engine.

[0022] Another aspect of the invention includes a secure cryptographic system that is optionally remotely accessible. The cryptographic system comprises a depository system having at least one server which stores at least one private key and any other data, such as, but not limited to, a plurality of enrollment authentication data, wherein each enrollment authentication data identifies one of possibly multiple users. The cryptographic system may also optionally comprise an authentication engine which compares authentication data received by users to enrollment authentication data corresponding to the user and received from the depository system, thereby producing an authentication result. The cryptographic system also comprises a cryptographic engine which, when the authentication result indicates proper identification of the user, performs cryptographic functions on behalf of the user using at least said private key, which may be received from the depository system. The cryptographic system may also optionally comprise a transaction engine connected to route data from the users to other engines or systems such as, but not limited to, the depository server system, the authentication engine, and the cryptographic engine.

[0023] Another aspect of the invention includes a method of facilitating cryptographic functions. The method comprises associating a user from multiple users with one or more keys from a plurality of private cryptographic keys stored on a secure location, such as a secure server. The method also comprises receiving authentication data from the user, and comparing the authentication data to authentication data corresponding to the user, thereby verifying the

identity of the user. The method also comprises utilizing the one or more keys to perform cryptographic functions without releasing the one or more keys to the user.

[0024] Another aspect of the invention includes an authentication system for uniquely identifying a user through secure storage of the user's enrollment authentication data. The authentication system comprises one or more data storage facilities, wherein each data storage facility includes a computer accessible storage medium which stores at least one of portions of enrollment authentication data. The authentication system also comprises an authentication engine which communicates with the data storage facility or facilities. The authentication engine comprises a data splitting module which operates on the enrollment authentication data to create portions, a data assembling module which processes the portions from at least one of the data storage facilities to assemble the enrollment authentication data, and a data comparator module which receives current authentication data from a user and compares the current authentication data with the assembled enrollment authentication data to determine whether the user has been uniquely identified.

[0025] Another aspect of the invention includes a cryptographic system. The cryptographic system comprises one or more data storage facilities, wherein each data storage facility includes a computer accessible storage medium which stores at least one portion of one ore more cryptographic keys. The cryptographic system also comprises a cryptographic engine which communicates with the data storage

facilities. The cryptographic engine also comprises a data splitting module which operate on the cryptographic keys to create portions, a data assembling module which processes the portions from at least one of the data storage facilities to assemble the cryptographic keys, and a cryptographic handling module which receives the assembled cryptographic keys and performs cryptographic functions therewith.

[0026] Another aspect of the invention includes a method of storing any type of data, including, but not limited to, authentication data in geographically remote secure data storage facilities thereby protecting the data against composition of any individual data storage facility. The method comprises receiving data at a trust engine, combining at the trust engine the data with a first substantially random value to form a first combined value, and combining the data with a second substantially random value to form a second combined value. The method comprises creating a first pairing of the first substantially random value with the second combined value, creating a second pairing of the first substantially random value with the second substantially random value, and storing the first pairing in a first secure data storage facility. The method comprises storing the second pairing in a second secure data storage facility remote from the first secure data storage facility.

[0027] Another aspect of the invention includes a method of storing any type of data, including, but not limited to, authentication data comprising receiving data, combining the data with a first set of bits to form a second set of bits, and combining the data with a third set of bits to form a fourth set of bits. The

method also comprises creating a first pairing of the first set of bits with the third set of bits. The method also comprises creating a second pairing of the first set of bits with the fourth set of bits, and storing one of the first and second pairings in a first computer accessible storage medium. The method also comprises storing the other of the first and second pairings in a second computer accessible storage medium.

[0028] Another aspect of the invention includes a method of storing cryptographic data in geographically remote secure data storage facilities thereby protecting the cryptographic data against comprise of any individual data storage facility. The method comprises receiving cryptographic data at a trust engine, combining at the trust engine the cryptographic data with a first substantially random value to form a first combined value, and combining the cryptographic data with a second substantially random value to form a second combined value. The method also comprises creating a first pairing of the first substantially random value with the second combined value, creating a second pairing of the first substantially random value with the second substantially random value, and storing the first pairing in a first secure data storage facility. The method also comprises storing the second pairing in a secure second data storage facility remote from the first secure data storage facility.

[0029] Another aspect of the invention includes a method of storing cryptographic data comprising receiving authentication data and combining the cryptographic data with a first set of bits to form a second set of bits. The method also comprises

combining the cryptographic data with a third set of bits to form a fourth set of bits, creating a first pairing of the first set of bits with the third set of bits, and creating a second pairing of the first set of bits with the fourth set of bits. The method also comprises storing one of the first and second pairings in a first computer accessible storage medium, and storing the other of the first and second pairings in a second computer accessible storage medium.

[0030] Another aspect of the invention includes a method of handling sensitive data of any type or form in a cryptographic system, wherein the sensitive data exists in a useable form only during actions by authorized users, employing the sensitive data. The method also comprises receiving in a software module, substantially randomized or encrypted sensitive data from a first computer accessible storage medium, and receiving in the software module, substantially randomized or encrypted data which may or may not be sensitive data, from one or more other computer accessible storage medium. The method also comprises processing the substantially randomized pre-encrypted sensitive data and the substantially randomized or encrypted data which may or may not be sensitive data, in the software module to assemble the sensitive data and employing the sensitive data in a software engine to perform an action. The action includes, but is not limited to, one of authenticating a user and performing a cryptographic function.

[0031] Another aspect of the invention includes a secure authentication system. The secure authentication system comprises a plurality of authentication engines. Each authentication engine

receives enrollment authentication data designed to uniquely identify a user to a degree of certainty. Each authentication engine receives current authentication data to compare to the enrollment authentication data, and each authentication engine determines an authentication result. The secure authentication system also comprises a redundancy system which receives the authentication result of at least two of the authentication engines and determines whether the user has been uniquely identified.

[0032] Another aspect of the invention includes a secure data in motion system whereby data may be transmitted in different portions that are secured in accordance with the present invention such that any one portion becoming compromised shall not provide sufficient data to restore the original data. This may be applied to any transmission of data, whether it be wired, wireless, or physical.

[0033] Another aspect of the invention includes integration of the secure data parser of the present invention into any suitable system where data is stored or communicated. For example, email system, RAID systems, video broadcasting systems, database systems, or any other suitable system may have the secure data parser integrated at any suitable level.

[0034] Another aspect of the invention includes using any suitable parsing and splitting algorithm to generate shares of data. Either random, pseudo-random, deterministic, or any combination thereof may be employed for parsing and splitting data.

Brief Description of the Drawings

[0035] The present invention is described in more detail below in connection with the attached drawings, which are meant to illustrate and not to limit the invention, and in which:

[0036] FIGURE 1 illustrates a block diagram of a cryptographic system, according to aspects of an embodiment of the invention;

[0037] FIGURE 2 illustrates a block diagram of the trust engine of FIGURE 1, according to aspects of an embodiment of the invention;

[0038] FIGURE 3 illustrates a block diagram of the transaction engine of FIGURE 2, according to aspects of an embodiment of the invention;

[0039] FIGURE 4 illustrates a block diagram of the depository of FIGURE 2, according to aspects of an embodiment of the invention;

[0040] FIGURE 5 illustrates a block diagram of the authentication engine of FIGURE 2, according to aspects of an embodiment of the invention;

[0041] FIGURE 6 illustrates a block diagram of the cryptographic engine of FIGURE 2, according to aspects of an embodiment of the invention;

[0042] FIGURE 7 illustrates a block diagram of a depository system, according to aspects of another embodiment of the invention;

[0043] FIGURE 8 illustrates a flow chart of a data splitting process according to aspects of an embodiment of the invention;

[0044] FIGURE 9, Panel A illustrates a data flow of an enrollment process according to aspects of an embodiment of the invention;

[0045] FIGURE 9, Panel B illustrates a flow chart of an interoperability process according to aspects of an embodiment of the invention;

[0046] FIGURE 10 illustrates a data flow of an authentication process according to aspects of an embodiment of the invention;

[0047] FIGURE 11 illustrates a data flow of a signing process according to aspects of an embodiment of the invention;

[0048] FIGURE 12 illustrates a data flow and an encryption/decryption process according to aspects and yet another embodiment of the invention;

[0049] FIGURE 13 illustrates a simplified block diagram of a trust engine system according to aspects of another embodiment of the invention;

[0050] FIGURE 14 illustrates a simplified block diagram of a trust engine system according to aspects of another embodiment of the invention;

[0051] FIGURE 15 illustrates a block diagram of the redundancy module of FIGURE 14, according to aspects of an embodiment of the invention;

[0052] FIGURE 16 illustrates a process for evaluating authentications according to one aspect of the invention;

[0053] FIGURE 17 illustrates a process for assigning a value to an authentication according to one aspect as shown in FIGURE 16 of the invention;

[0054] FIGURE 18 illustrates a process for performing trust arbitrage in an aspect of the invention as shown in FIGURE 17; and

[0055] FIGURE 19 illustrates a sample transaction between a user and a vendor according to aspects of an embodiment of the invention where an initial web based

contact leads to a sales contract signed by both parties.

[0056] FIGURE 20 illustrates a sample user system with a cryptographic service provider module which provides security functions to a user system.

[0057] FIGURE 21 illustrates a process for parsing, splitting and/or separating data with encryption and storage of the encryption master key with the data.

[0058] FIGURE 22 illustrates a process for parsing, splitting and/or separating data with encryption and storing the encryption master key separately from the data.

[0059] FIGURE 23 illustrates the intermediary key process for parsing, splitting and/or separating data with encryption and storage of the encryption master key with the data.

[0060] FIGURE 24 illustrates the intermediary key process for parsing, splitting and/or separating data with encryption and storing the encryption master key separately from the data.

[0061] FIGURE 25 illustrates utilization of the cryptographic methods and systems of the present invention with a small working group.

[0062] FIGURE 26 is a block diagram of an illustrative physical token security system employing the secure data parser in accordance with one embodiment of the present invention.

[0063] FIGURE 27 is a block diagram of an illustrative arrangement in which the secure data parser is integrated into a system in accordance with one embodiment of the present invention.

[0064] FIGURE 28 is a block diagram of an illustrative data in motion system in accordance with one embodiment of the present invention.

[0065] FIGURE 29 is a block diamgram of another illustrative data in motion system in accordance with one embodiment of the present invention.

[0066] FIGURE 30-32 are block diagrams of an illustrative system having the secure data parser integrated in accordance with one embodiment of the present invention.

[0067] FIGURE 33 is a process flow diagram of an illustrative process for parsing and splitting data in accordance with one embodiment of the present invention.

[0068] FIGURE 34 is a process flow diagram of an illustrative process for restoring portions of data into original data in accordance with one embodiment of the present invention.

[0069] FIGURE 35 is a process flow diagram of an illustrative process for splitting data at the bit level in accordance with one embodiment of the present invention.

[0070] FIGURE 36 is a process flow diagram of illustrative steps and features, that may be used in any suitable combination, with any suitable additions, deletions, or modifications in accordance with one embodiment of the present invention.

[0071] FIGURE 37 is a process flow diagram of illustrative steps and features, that may be used in any suitable combination, with any suitable additions, deletions, or modifications in accordance with one embodiment of the present invention.

[0072] FIGURE 38 is a simplified block diagram of the storage of key and data components within shares, that may be used in any suitable combination, with any suitable additions, deletions, or modifications in accordance with one embodiment of the present invention.

[0073] FIGURE 39 is a simplified block diagram of the storage of key and data components within shares using a workgroup key, that may be used in any suitable combination, with any suitable additions, deletions, or modifications in accordance with one embodiment of the present invention.

[0074] FIGURES 40A and 40B are simplified and illustrative process flow diagrams for header generation and data splitting for data in motion, that may be used in any suitable combination, with any suitable additions, deletions, or modifications in accordance with one embodiment of the present invention.

[0075] FIGURE 41 is a simplified block diagram of an illustrative share format, that may be used in any suitable combination, with any suitable additions, deletions, or modifications in accordance with one embodiment of the present invention.

[0076] FIGURE 42 is a process flow diagram of illustrative steps and features, that may be used in any suitable combination, with any suitable additions, deletions, or modifications for managing cryptographic keys in accordance with one embodiment of the present invention.

Detailed Description of the Invention

[0077] One aspect of the present invention is to provide a cryptographic system where one or more secure servers, or a trust engine, stores cryptographic keys and user authentication data. Users access the functionality of conventional cryptographic systems through network access to the trust engine, however, the trust engine does not release actual keys and other authentication data and therefore, the keys and data remain secure. This server-centric storage of keys and authentication data provides for user-independent security, portability, availability, and straightforwardness.

[0078] Because users can be confident in, or trust, the cryptographic system to perform user and document authentication and other cryptographic functions, a wide variety of functionality may be incorporated into the system. For example, the trust engine provider can ensure against agreement repudiation by, for example, authenticating the agreement participants, digitally signing the agreement on behalf of or for the participants, and storing a record of the agreement digitally signed by each participant. In addition, the cryptographic system may monitor agreements and determine to apply varying degrees of authentication, based on, for example, price, user, vendor, geographic location, place of use, or the like.

[0079] To facilitate a complete understanding of the invention, the remainder of the detailed description describes the invention with reference to the figures, wherein like elements are referenced with like numerals throughout.

[0080] FIGURE 1 illustrates a block diagram of a cryptographic system 100, according to aspects of an embodiment of the invention. As shown in FIGURE 1, the cryptographic system 100 includes a user system 105, a trust engine 110, a certificate authority 115, and a vendor system 120, communicating through a communication link 125.

[0081] According to one embodiment of the invention, the user system 105 comprises a conventional general-purpose computer having one or more microprocessors, such as, for example, an Intel-based processor. Moreover, the user system 105 includes an appropriate operating system, such as, for example, an operating system capable of including graphics or windows, such as Windows, Unix, Linux, or the like. As shown in FIGURE 1, the user system 105 may include a biometric device 107. The biometric device 107 may advantageously capture a user's biometric and transfer the captured biometric to the trust engine 110. According to one embodiment of the invention, the biometric device may advantageously comprise a device having attributes and features similar to those disclosed in U.S. Patent Application No. 08/926,277, filed on September 5, 1997, entitled *"RELIEF OBJECT IMAGE GENERATOR,"* U.S. Patent Application No. 09/558,634, filed on April 26, 2000, entitled *"IMAGING DEVICE FOR A RELIEF OBJECT AND SYSTEM AND METHOD OF USING THE IMAGE DEVICE,"* U.S. Patent Application No. 09/435,011, filed on November 5, 1999, entitled *"RELIEF OBJECT SENSOR ADAPTOR,"* and U.S. Patent Application No. 09/477,943, filed on January 5, 2000, entitled *"PLANAR OPTICAL IMAGE SENSOR AND SYSTEM FOR GENERATING AN ELECTRONIC IMAGE OF A RELIEF OBJECT*

FOR FINGERPRINT READING," all of which are owned by the instant assignee, and all of which are hereby incorporated by reference herein.

[0082] In addition, the user system 105 may connect to the communication link 125 through a conventional service provider, such as, for example, a dial up, digital subscriber line (DSL), cable modem, fiber connection, or the like. According to another embodiment, the user system 105 connects the communication link 125 through network connectivity such as, for example, a local or wide area network. According to one embodiment, the operating system includes a TCP/IP stack that handles all incoming and outgoing message traffic passed over the communication link 125.

[0083] Although the user system 105 is disclosed with reference to the foregoing embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives embodiments of the user system 105, including almost any computing device capable of sending or receiving information from another computer system. For example, the user system 105 may include, but is not limited to, a computer workstation, an interactive television, an interactive kiosk, a personal mobile computing device, such as a digital assistant, mobile phone, laptop, or the like, a wireless communications device, a smartcard, an embedded computing device, or the like, which can interact with the communication link 125. In such alternative systems, the operating systems will likely differ and be adapted for the particular device. However, according to one embodiment, the operating

systems advantageously continue to provide the appropriate communications protocols needed to establish communication with the communication link 125.

[0084] FIGURE 1 illustrates the trust engine 110. According to one embodiment, the trust engine 110 comprises one or more secure servers for accessing and storing sensitive information, which may be any type or form of data, such as, but not limited to text, audio, video, user authentication data and public and private cryptographic keys. According to one embodiment, the authentication data includes data designed to uniquely identify a user of the cryptographic system 100. For example, the authentication data may include a user identification number, one or more biometrics, and a series of questions and answers generated by the trust engine 110 or the user, but answered initially by the user at enrollment. The foregoing questions may include demographic data, such as place of birth, address, anniversary, or the like, personal data, such as mother's maiden name, favorite ice cream, or the like, or other data designed to uniquely identify the user. The trust engine 110 compares a user's authentication data associated with a current transaction, to the authentication data provided at an earlier time, such as, for example, during enrollment. The trust engine 110 may advantageously require the user to produce the authentication data at the time of each transaction, or, the trust engine 110 may advantageously allow the user to periodically produce authentication data, such as at the beginning of a string of transactions or the logging onto a particular vendor website.

[0085] According to the embodiment where the user produces biometric data, the user provides a physical characteristic, such as, but not limited to, facial scan, hand scan, ear scan, iris scan, retinal scan, vascular pattern, DNA, a fingerprint, writing or speech, to the biometric device 107. The biometric device advantageously produces an electronic pattern, or biometric, of the physical characteristic. The electronic pattern is transferred through the user system 105 to the trust engine 110 for either enrollment or authentication purposes.

[0086] Once the user produces the appropriate authentication data and the trust engine 110 determines a positive match between that authentication data (current authentication data) and the authentication data provided at the time of enrollment (enrollment authentication data), the trust engine 110 provides the user with complete cryptographic functionality. For example, the properly authenticated user may advantageously employ the trust engine 110 to perform hashing, digitally signing, encrypting and decrypting (often together referred to only as encrypting), creating or distributing digital certificates, and the like. However, the private cryptographic keys used in the cryptographic functions will not be available outside the trust engine 110, thereby ensuring the integrity of the cryptographic keys.

[0087] According to one embodiment, the trust engine 110 generates and stores cryptographic keys. According to another embodiment, at least one cryptographic key is associated with each user. Moreover, when the cryptographic keys include public-key technology, each private key associated with a user is generated within,

and not released from, the trust engine 110. Thus, so long as the user has access to the trust engine 110, the user may perform cryptographic functions using his or her private or public key. Such remote access advantageously allows users to remain completely mobile and access cryptographic functionality through practically any Internet connection, such as cellular and satellite phones, kiosks, laptops, hotel rooms and the like.

[0088] According to another embodiment, the trust engine 110 performs the cryptographic functionality using a key pair generated for the trust engine 110. According to this embodiment, the trust engine 110 first authenticates the user, and after the user has properly produced authentication data matching the enrollment authentication data, the trust engine 110 uses its own cryptographic key pair to perform cryptographic functions on behalf of the authenticated user.

[0089] A skilled artisan will recognize from the disclosure herein that the cryptographic keys may advantageously include some or all of symmetric keys, public keys, and private keys. In addition, a skilled artisan will recognize from the disclosure herein that the foregoing keys may be implemented with a wide number of algorithms available from commercial technologies, such as, for example, RSA, ELGAMAL, or the like.

[0090] FIGURE 1 also illustrates the certificate authority 115. According to one embodiment, the certificate authority 115 may advantageously comprise a trusted third-party organization or company that issues digital certificates, such as, for example, VeriSign,

Baltimore, Entrust, or the like. The trust engine 110 may advantageously transmit requests for digital certificates, through one or more conventional digital certificate protocols, such as, for example, PKCS10, to the certificate authority 115. In response, the certificate authority 115 will issue a digital certificate in one or more of a number of differing protocols, such as, for example, PKCS7. According to one embodiment of the invention, the trust engine 110 requests digital certificates from several or all of the prominent certificate authorities 115 such that the trust engine 110 has access to a digital certificate corresponding to the certificate standard of any requesting party.

[0091] According to another embodiment, the trust engine 110 internally performs certificate issuances. In this embodiment, the trust engine 110 may access a certificate system for generating certificates and/or may internally generate certificates when they are requested, such as, for example, at the time of key generation or in the certificate standard requested at the time of the request. The trust engine 110 will be disclosed in greater detail below.

[0092] FIGURE 1 also illustrates the vendor system 120. According to one embodiment, the vendor system 120 advantageously comprises a Web server. Typical Web servers generally serve content over the Internet using one of several internet markup languages or document format standards, such as the Hyper-Text Markup Language (HTML) or the Extensible Markup Language (XML). The Web server accepts requests from browsers like Netscape and Internet Explorer and then returns the appropriate electronic documents. A number of

server or client-side technologies can be used to increase the power of the Web server beyond its ability to deliver standard electronic documents. For example, these technologies include Common Gateway Interface (CGI) scripts, Secure Sockets Layer (SSL) security, and Active Server Pages (ASPs). The vendor system 120 may advantageously provide electronic content relating to commercial, personal, educational, or other transactions.

[0093] Although the vendor system 120 is disclosed with reference to the foregoing embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein that the vendor system 120 may advantageously comprise any of the devices described with reference to the user system 105 or combination thereof.

[0094] FIGURE 1 also illustrates the communication link 125 connecting the user system 105, the trust engine 110, the certificate authority 115, and the vendor system 120. According to one embodiment, the communication link 125 preferably comprises the Internet. The Internet, as used throughout this disclosure is a global network of computers. The structure of the Internet, which is well known to those of ordinary skill in the art, includes a network backbone with networks branching from the backbone. These branches, in turn, have networks branching from them, and so on. Routers move information packets between network levels, and then from network to network, until the packet reaches the neighborhood of its destination. From the destination, the destination network's host directs the information packet to the

appropriate terminal, or node. In one advantageous embodiment, the Internet routing hubs comprise domain name system (DNS) servers using Transmission Control Protocol/Internet Protocol (TCP/IP) as is well known in the art. The routing hubs connect to one or more other routing hubs via high-speed communication links.

[0095] One popular part of the Internet is the World Wide Web. The World Wide Web contains different computers, which store documents capable of displaying graphical and textual information. The computers that provide information on the World Wide Web are typically called "websites." A website is defined by an Internet address that has an associated electronic page. The electronic page can be identified by a Uniform Resource Locator (URL). Generally, an electronic page is a document that organizes the presentation of text, graphical images, audio, video, and so forth.

[0096] Although the communication link 125 is disclosed in terms of its preferred embodiment, one of ordinary skill in the art will recognize from the disclosure herein that the communication link 125 may include a wide range of interactive communications links. For example, the communication link 125 may include interactive television networks, telephone networks, wireless data transmission systems, two-way cable systems, customized private or public computer networks, interactive kiosk networks, automatic teller machine networks, direct links, satellite or cellular networks, and the like.

[0097] FIGURE 2 illustrates a block diagram of the trust engine 110 of FIGURE 1 according to aspects of an embodiment of the invention. As shown in FIGURE 2, the trust engine 110 includes a transaction engine 205, a

depository 210, an authentication engine 215, and a cryptographic engine 220. According to one embodiment of the invention, the trust engine 110 also includes mass storage 225. As further shown in FIGURE 2, the transaction engine 205 communicates with the depository 210, the authentication engine 215, and the cryptographic engine 220, along with the mass storage 225. In addition, the depository 210 communicates with the authentication engine 215, the cryptographic engine 220, and the mass storage 225. Moreover, the authentication engine 215 communicates with the cryptographic engine 220. According to one embodiment of the invention, some or all of the foregoing communications may advantageously comprise the transmission of XML documents to IP addresses that correspond to the receiving device. As mentioned in the foregoing, XML documents advantageously allow designers to create their own customized document tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations. Moreover, some or all of the foregoing communications may include conventional SSL technologies.

[0098] According to one embodiment, the transaction engine 205 comprises a data routing device, such as a conventional Web server available from Netscape, Microsoft, Apache, or the like. For example, the Web server may advantageously receive incoming data from the communication link 125. According to one embodiment of the invention, the incoming data is addressed to a front-end security system for the trust engine 110. For example, the front-end security system may advantageously include a firewall, an intrusion

detection system searching for known attack profiles, and/or a virus scanner. After clearing the front-end security system, the data is received by the transaction engine 205 and routed to one of the depository 210, the authentication engine 215, the cryptographic engine 220, and the mass storage 225. In addition, the transaction engine 205 monitors incoming data from the authentication engine 215 and cryptographic engine 220, and routes the data to particular systems through the communication link 125. For example, the transaction engine 205 may advantageously route data to the user system 105, the certificate authority 115, or the vendor system 120.

[0099] According to one embodiment, the data is routed using conventional HTTP routing techniques, such as, for example, employing URLs or Uniform Resource Indicators (URIs). URIs are similar to URLs, however, URIs typically indicate the source of files or actions, such as, for example, executables, scripts, and the like. Therefore, according to the one embodiment, the user system 105, the certificate authority 115, the vendor system 120, and the components of the trust engine 210, advantageously include sufficient data within communication URLs or URIs for the transaction engine 205 to properly route data throughout the cryptographic system.

[0100] Although the data routing is disclosed with reference to its preferred embodiment, a skilled artisan will recognize a wide number of possible data routing solutions or strategies. For example, XML or other data packets may advantageously be unpacked and recognized by their format, content, or the like, such that the transaction engine 205 may properly route data

throughout the trust engine 110. Moreover, a skilled artisan will recognize that the data routing may advantageously be adapted to the data transfer protocols conforming to particular network systems, such as, for example, when the communication link 125 comprises a local network.

[0101] According to yet another embodiment of the invention, the transaction engine 205 includes conventional SSL encryption technologies, such that the foregoing systems may authenticate themselves, and vise-versa, with transaction engine 205, during particular communications. As will be used throughout this disclosure, the term "½ SSL" refers to communications where a server but not necessarily the client, is SSL authenticated, and the term "FULL SSL" refers to communications where the client and the server are SSL authenticated. When the instant disclosure uses the term "SSL", the communication may comprise ½ or FULL SSL.

[0102] As the transaction engine 205 routes data to the various components of the cryptographic system 100, the transaction engine 205 may advantageously create an audit trail. According to one embodiment, the audit trail includes a record of at least the type and format of data routed by the transaction engine 205 throughout the cryptographic system 100. Such audit data may advantageously be stored in the mass storage 225.

[0103] FIGURE 2 also illustrates the depository 210. According to one embodiment, the depository 210 comprises one or more data storage facilities, such as, for example, a directory server, a database server, or the like. As shown in FIGURE 2, the depository 210 stores cryptographic keys and enrollment authentication

data. The cryptographic keys may advantageously correspond to the trust engine 110 or to users of the cryptographic system 100, such as the user or vendor. The enrollment authentication data may advantageously include data designed to uniquely identify a user, such as, user ID, passwords, answers to questions, biometric data, or the like. This enrollment authentication data may advantageously be acquired at enrollment of a user or another alternative later time. For example, the trust engine 110 may include periodic or other renewal or reissue of enrollment authentication data.

[0104] According to one embodiment, the communication from the transaction engine 205 to and from the authentication engine 215 and the cryptographic engine 220 comprises secure communication, such as, for example conventional SSL technology. In addition, as mentioned in the foregoing, the data of the communications to and from the depository 210 may be transferred using URLs, URIs, HTTP or XML documents, with any of the foregoing advantageously having data requests and formats embedded therein.

[0105] As mentioned above, the depository 210 may advantageously comprises a plurality of secure data storage facilities. In such an embodiment, the secure data storage facilities may be configured such that a compromise of the security in one individual data storage facility will not compromise the cryptographic keys or the authentication data stored therein. For example, according to this embodiment, the cryptographic keys and the authentication data are mathematically operated on so as to statistically and substantially randomize the data stored in each data

storage facility. According to one embodiment, the randomization of the data of an individual data storage facility renders that data undecipherable. Thus, compromise of an individual data storage facility produces only a randomized undecipherable number and does not compromise the security of any cryptographic keys or the authentication data as a whole.

[0106] FIGURE 2 also illustrates the trust engine 110 including the authentication engine 215. According to one embodiment, the authentication engine 215 comprises a data comparator configured to compare data from the transaction engine 205 with data from the depository 210. For example, during authentication, a user supplies current authentication data to the trust engine 110 such that the transaction engine 205 receives the current authentication data. As mentioned in the foregoing, the transaction engine 205 recognizes the data requests, preferably in the URL or URI, and routes the authentication data to the authentication engine 215. Moreover, upon request, the depository 210 forwards enrollment authentication data corresponding to the user to the authentication engine 215. Thus, the authentication engine 215 has both the current authentication data and the enrollment authentication data for comparison.

[0107] According to one embodiment, the communications to the authentication engine comprise secure communications, such as, for example, SSL technology. Additionally, security can be provided within the trust engine 110 components, such as, for example, super-encryption using public key technologies. For example, according to one embodiment, the user encrypts the current

authentication data with the public key of the authentication engine 215. In addition, the depository 210 also encrypts the enrollment authentication data with the public key of the authentication engine 215. In this way, only the authentication engine's private key can be used to decrypt the transmissions.

[0108] As shown in FIGURE 2, the trust engine 110 also includes the cryptographic engine 220. According to one embodiment, the cryptographic engine comprises a cryptographic handling module, configured to advantageously provide conventional cryptographic functions, such as, for example, public-key infrastructure (PKI) functionality. For example, the cryptographic engine 220 may advantageously issue public and private keys for users of the cryptographic system 100. In this manner, the cryptographic keys are generated at the cryptographic engine 220 and forwarded to the depository 210 such that at least the private cryptographic keys are not available outside of the trust engine 110. According to another embodiment, the cryptographic engine 220 randomizes and splits at least the private cryptographic key data, thereby storing only the randomized split data. Similar to the splitting of the enrollment authentication data, the splitting process ensures the stored keys are not available outside the cryptographic engine 220. According to another embodiment, the functions of the cryptographic engine can be combined with and performed by the authentication engine 215.

[0109] According to one embodiment, communications to and from the cryptographic engine include secure communications, such as SSL technology. In addition, XML documents may advantageously be employed to

transfer data and/or make cryptographic function requests.

[0110] FIGURE 2 also illustrates the trust engine 110 having the mass storage 225. As mentioned in the foregoing, the transaction engine 205 keeps data corresponding to an audit trail and stores such data in the mass storage 225. Similarly, according to one embodiment of the invention, the depository 210 keeps data corresponding to an audit trail and stores such data in the mass storage device 225. The depository audit trail data is similar to that of the transaction engine 205 in that the audit trail data comprises a record of the requests received by the depository 210 and the response thereof. In addition, the mass storage 225 may be used to store digital certificates having the public key of a user contained therein.

[0111] Although the trust engine 110 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize in the disclosure herein, a wide number of alternatives for the trust engine 110. For example, the trust engine 110, may advantageously perform only authentication, or alternatively, only some or all of the cryptographic functions, such as data encryption and decryption. According to such embodiments, one of the authentication engine 215 and the cryptographic engine 220 may advantageously be removed, thereby creating a more straightforward design for the trust engine 110. In addition, the cryptographic engine 220 may also communicate with a certificate authority such that the certificate authority is embodied within the trust engine 110. According to yet another embodiment,

the trust engine 110 may advantageously perform authentication and one or more cryptographic functions, such as, for example, digital signing.

[0112] FIGURE 3 illustrates a block diagram of the transaction engine 205 of FIGURE 2, according to aspects of an embodiment of the invention. According to this embodiment, the transaction engine 205 comprises an operating system 305 having a handling thread and a listening thread. The operating system 305 may advantageously be similar to those found in conventional high volume servers, such as, for example, Web servers available from Apache. The listening thread monitors the incoming communication from one of the communication link 125, the authentication engine 215, and the cryptographic engine 220 for incoming data flow. The handling thread recognizes particular data structures of the incoming data flow, such as, for example, the foregoing data structures, thereby routing the incoming data to one of the communication link 125, the depository 210, the authentication engine 215, the cryptographic engine 220, or the mass storage 225. As shown in FIGURE 3, the incoming and outgoing data may advantageously be secured through, for example, SSL technology.

[0113] FIGURE 4 illustrates a block diagram of the depository 210 of FIGURE 2 according to aspects of an embodiment of the invention. According to this embodiment, the depository 210 comprises one or more lightweight directory access protocol (LDAP) servers. LDAP directory servers are available from a wide variety of manufacturers such as Netscape, ISO, and others. FIGURE 4 also shows that the directory server preferably stores data 405 corresponding to the

cryptographic keys and data 410 corresponding to the enrollment authentication data. According to one embodiment, the depository 210 comprises a single logical memory structure indexing authentication data and cryptographic key data to a unique user ID. The single logical memory structure preferably includes mechanisms to ensure a high degree of trust, or security, in the data stored therein. For example, the physical location of the depository 210 may advantageously include a wide number of conventional security measures, such as limited employee access, modern surveillance systems, and the like. In addition to, or in lieu of, the physical securities, the computer system or server may advantageously include software solutions to protect the stored data. For example, the depository 210 may advantageously create and store data 415 corresponding to an audit trail of actions taken. In addition, the incoming and outgoing communications may advantageously be encrypted with public key encryption coupled with conventional SSL technologies.

[0114] According to another embodiment, the depository 210 may comprise distinct and physically separated data storage facilities, as disclosed further with reference to FIGURE 7.

[0115] FIGURE 5 illustrates a block diagram of the authentication engine 215 of FIGURE 2 according to aspects of an embodiment of the invention. Similar to the transaction engine 205 of FIGURE 3, the authentication engine 215 comprises an operating system 505 having at least a listening and a handling thread of a modified version of a conventional Web server, such as, for example, Web servers available from

Apache. As shown in FIGURE 5, the authentication engine 215 includes access to at least one private key 510. The private key 510 may advantageously be used for example, to decrypt data from the transaction engine 205 or the depository 210, which was encrypted with a corresponding public key of the authentication engine 215.

[0116] FIGURE 5 also illustrates the authentication engine 215 comprising a comparator 515, a data splitting module 520, and a data assembling module 525. According to the preferred embodiment of the invention, the comparator 515 includes technology capable of comparing potentially complex patterns related to the foregoing biometric authentication data. The technology may include hardware, software, or combined solutions for pattern comparisons, such as, for example, those representing finger print patterns or voice patterns. In addition, according to one embodiment, the comparator 515 of the authentication engine 215 may advantageously compare conventional hashes of documents in order to render a comparison result. According to one embodiment of the invention, the comparator 515 includes the application of heuristics 530 to the comparison. The heuristics 530 may advantageously address circumstances surrounding an authentication attempt, such as, for example, the time of day, IP address or subnet mask, purchasing profile, email address, processor serial number or ID, or the like.

[0117] Moreover, the nature of biometric data comparisons may result in varying degrees of confidence being produced from the matching of current biometric authentication data to enrollment data. For example,

unlike a traditional password which may only return a positive or negative match, a fingerprint may be determined to be a partial match, e.g. a 90% match, a 75% match, or a 10% match, rather than simply being correct or incorrect. Other biometric identifiers such as voice print analysis or face recognition may share this property of probabilistic authentication, rather than absolute authentication.

[0118] When working with such probabilistic authentication or in other cases where an authentication is considered less than absolutely reliable, it is desirable to apply the heuristics 530 to determine whether the level of confidence in the authentication provided is sufficiently high to authenticate the transaction which is being made.

[0119] It will sometimes be the case that the transaction at issue is a relatively low value transaction where it is acceptable to be authenticated to a lower level of confidence. This could include a transaction which has a low dollar value associated with it (e.g., a $10 purchase) or a transaction with low risk (e.g., admission to a members-only web site).

[0120] Conversely, for authenticating other transactions, it may be desirable to require a high degree of confidence in the authentication before allowing the transaction to proceed. Such transactions may include transactions of large dollar value (e.g., signing a multi-million dollar supply contract) or transaction with a high risk if an improper authentication occurs (e.g., remotely logging onto a government computer).

[0121] The use of the heuristics 530 in combination with confidence levels and transactions values may be

used as will be described below to allow the comparator to provide a dynamic context-sensitive authentication system.

[0122] According to another embodiment of the invention, the comparator 515 may advantageously track authentication attempts for a particular transaction. For example, when a transaction fails, the trust engine 110 may request the user to re-enter his or her current authentication data. The comparator 515 of the authentication engine 215 may advantageously employ an attempt limiter 535 to limit the number of authentication attempts, thereby prohibiting brute-force attempts to impersonate a user's authentication data. According to one embodiment, the attempt limiter 535 comprises a software module monitoring transactions for repeating authentication attempts and, for example, limiting the authentication attempts for a given transaction to three. Thus, the attempt limiter 535 will limit an automated attempt to impersonate an individual's authentication data to, for example, simply three "guesses." Upon three failures, the attempt limiter 535 may advantageously deny additional authentication attempts. Such denial may advantageously be implemented through, for example, the comparator 515 returning a negative result regardless of the current authentication data being transmitted. On the other hand, the transaction engine 205 may advantageously block any additional authentication attempts pertaining to a transaction in which three attempts have previously failed.

[0123] The authentication engine 215 also includes the data splitting module 520 and the data assembling module 525. The data splitting module 520

advantageously comprises a software, hardware, or combination module having the ability to mathematically operate on various data so as to substantially randomize and split the data into portions. According to one embodiment, original data is not recreatable from an individual portion. The data assembling module 525 advantageously comprises a software, hardware, or combination module configured to mathematically operate on the foregoing substantially randomized portions, such that the combination thereof provides the original deciphered data. According to one embodiment, the authentication engine 215 employs the data splitting module 520 to randomize and split enrollment authentication data into portions, and employs the data assembling module 525 to reassemble the portions into usable enrollment authentication data.

[0124] FIGURE 6 illustrates a block diagram of the cryptographic engine 220 of the trust engine 200 of FIGURE 2 according to aspects of one embodiment of the invention. Similar to the transaction engine 205 of FIGURE 3, the cryptographic engine 220 comprises an operating system 605 having at least a listening and a handling thread of a modified version of a conventional Web server, such as, for example, Web servers available from Apache. As shown in FIGURE 6, the cryptographic engine 220 comprises a data splitting module 610 and a data assembling module 620 that function similar to those of FIGURE 5. However, according to one embodiment, the data splitting module 610 and the data assembling module 620 process cryptographic key data, as opposed to the foregoing enrollment authentication data. Although, a skilled artisan will recognize from the disclosure herein that the data splitting module

910 and the data splitting module 620 may be combined with those of the authentication engine 215.

[0125] The cryptographic engine 220 also comprises a cryptographic handling module 625 configured to perform one, some or all of a wide number of cryptographic functions. According to one embodiment, the cryptographic handling module 625 may comprise software modules or programs, hardware, or both. According to another embodiment, the cryptographic handling module 625 may perform data comparisons, data parsing, data splitting, data separating, data hashing, data encryption or decryption, digital signature verification or creation, digital certificate generation, storage, or requests, cryptographic key generation, or the like. Moreover, a skilled artisan will recognize from the disclosure herein that the cryptographic handling module 825 may advantageously comprises a public-key infrastructure, such as Pretty Good Privacy (PGP), an RSA-based public-key system, or a wide number of alternative key management systems. In addition, the cryptographic handling module 625 may perform public-key encryption, symmetric-key encryption, or both. In addition to the foregoing, the cryptographic handling module 625 may include one or more computer programs or modules, hardware, or both, for implementing seamless, transparent, interoperability functions.

[0126] A skilled artisan will also recognize from the disclosure herein that the cryptographic functionality may include a wide number or variety of functions generally relating to cryptographic key management systems.

[0127] FIGURE 7 illustrates a simplified block diagram of a depository system 700 according to aspects of an embodiment of the invention. As shown in FIGURE 7, the depository system 700 advantageously comprises multiple data storage facilities, for example, data storage facilities D1, D2, D3, and D4. However, it is readily understood by those of ordinary skill in the art that the depository system may have only one data storage facility. According to one embodiment of the invention, each of the data storage facilities D1 through D4 may advantageously comprise some or all of the elements disclosed with reference to the depository 210 of FIGURE 4. Similar to the depository 210, the data storage facilities D1 through D4 communicate with the transaction engine 205, the authentication engine 215, and the cryptographic engine 220, preferably through conventional SSL. Communication links transferring, for example, XML documents. Communications from the transaction engine 205 may advantageously include requests for data, wherein the request is advantageously broadcast to the IP address of each data storage facility D1 through D4. On the other hand, the transaction engine 205 may broadcast requests to particular data storage facilities based on a wide number of criteria, such as, for example, response time, server loads, maintenance schedules, or the like.

[0128] In response to requests for data from the transaction engine 205, the depository system 700 advantageously forwards stored data to the authentication engine 215 and the cryptographic engine 220. The respective data assembling modules receive the forwarded data and assemble the data into useable

formats. On the other hand, communications from the authentication engine 215 and the cryptographic engine 220 to the data storage facilities D1 through D4 may include the transmission of sensitive data to be stored. For example, according to one embodiment, the authentication engine 215 and the cryptographic engine 220 may advantageously employ their respective data splitting modules to divide sensitive data into undecipherable portions, and then transmit one or more undecipherable portions of the sensitive data to a particular data storage facility.

[0129] According to one embodiment, each data storage facility, D1 through D4, comprises a separate and independent storage system, such as, for example, a directory server. According to another embodiment of the invention, the depository system 700 comprises multiple geographically separated independent data storage systems. By distributing the sensitive data into distinct and independent storage facilities D1 through D4, some or all of which may be advantageously geographically separated, the depository system 700 provides redundancy along with additional security measures. For example, according to one embodiment, only data from two of the multiple data storage facilities, D1 through D4, are needed to decipher and reassemble the sensitive data. Thus, as many as two of the four data storage facilities D1 through D4 may be inoperative due to maintenance, system failure, power failure, or the like, without affecting the functionality of the trust engine 110. In addition, because, according to one embodiment, the data stored in each data storage facility is randomized and undecipherable, compromise of any individual data

storage facility does not necessarily compromise the sensitive data. Moreover, in the embodiment having geographical separation of the data storage facilities, a compromise of multiple geographically remote facilities becomes increasingly difficult. In fact, even a rogue employee will be greatly challenged to subvert the needed multiple independent geographically remote data storage facilities.

[0130] Although the depository system 700 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the depository system 700. For example, the depository system 700 may comprise one, two or more data storage facilities. In addition, sensitive data may be mathematically operated such that portions from two or more data storage facilities are needed to reassemble and decipher the sensitive data.

[0131] As mentioned in the foregoing, the authentication engine 215 and the cryptographic engine 220 each include a data splitting module 520 and 610, respectively, for splitting any type or form of sensitive data, such as, for example, text, audio, video, the authentication data and the cryptographic key data. FIGURE 8 illustrates a flowchart of a data splitting process 800 performed by the data splitting module according to aspects of an embodiment of the invention. As shown in FIGURE 8, the data splitting process 800 begins at step 805 when sensitive data "S" is received by the data splitting module of the authentication engine 215 or the cryptographic engine 220. Preferably, in step 810, the data splitting

module then generates a substantially random number, value, or string or set of bits, "A." For example, the random number A may be generated in a wide number of varying conventional techniques available to one of ordinary skill in the art, for producing high quality random numbers suitable for use in cryptographic applications. In addition, according to one embodiment, the random number A comprises a bit length which may be any suitable length, such as shorter, longer or equal to the bit length of the sensitive data, S.

[0132] In addition, in step 820 the data splitting process 800 generates another statistically random number "C." According to the preferred embodiment, the generation of the statistically random numbers A and C may advantageously be done in parallel. The data splitting module then combines the numbers A and C with the sensitive data S such that new numbers "B" and "D" are generated. For example, number B may comprise the binary combination of A XOR S and number D may comprise the binary combination of C XOR S. The XOR function, or the "exclusive-or" function, is well known to those of ordinary skill in the art. The foregoing combinations preferably occur in steps 825 and 830, respectively, and, according to one embodiment, the foregoing combinations also occur in parallel. The data splitting process 800 then proceeds to step 835 where the random numbers A and C and the numbers B and D are paired such that none of the pairings contain sufficient data, by themselves, to reorganize and decipher the original sensitive data S. For example, the numbers may be paired as follows: AC, AD, BC, and BD. According to one embodiment, each of the foregoing

pairings is distributed to one of the depositories D1 through D4 of FIGURE 7. According to another embodiment, each of the foregoing pairings is randomly distributed to one of the depositories D1 through D4. For example, during a first data splitting process 800, the pairing AC may be sent to depository D2, through, for example, a random selection of D2's IP address. Then, during a second data splitting process 800, the pairing AC may be sent to depository D4, through, for example, a random selection of D4's IP address. In addition, the pairings may all be stored on one depository, and may be stored in separate locations on said depository.

[0133] Based on the foregoing, the data splitting process 800 advantageously places portions of the sensitive data in each of the four data storage facilities D1 through D4, such that no single data storage facility D1 through D4 includes sufficient encrypted data to recreate the original sensitive data S. As mentioned in the foregoing, such randomization of the data into individually unusable encrypted portions increases security and provides for maintained trust in the data even if one of the data storage facilities, D1 through D4, is compromised.

[0134] Although the data splitting process 800 is disclosed with reference to its preferred embodiment, the invention is not intended to be limited thereby. Rather a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the data splitting process 800. For example, the data splitting process may advantageously split the data into two numbers, for example, random number A and number B and, randomly distribute A and B through two

data storage facilities. Moreover, the data splitting process 800 may advantageously split the data among a wide number of data storage facilities through generation of additional random numbers. The data may be split into any desired, selected, predetermined, or randomly assigned size unit, including but not limited to, a bit, bits, bytes, kilobytes, megabytes or larger, or any combination or sequence of sizes. In addition, varying the sizes of the data units resulting from the splitting process may render the data more difficult to restore to a useable form, thereby increasing security of sensitive data. It is readily apparent to those of ordinary skill in the art that the split data unit sizes may be a wide variety of data unit sizes or patterns of sizes or combinations of sizes. For example, the data unit sizes may be selected or predetermined to be all of the same size, a fixed set of different sizes, a combination of sizes, or randomly generates sizes. Similarly, the data units may be distributed into one or more shares according to a fixed or predetermined data unit size, a pattern or combination of data unit sizes, or a randomly generated data unit size or sizes per share.

[0135] As mentioned in the foregoing, in order to recreate the sensitive data S, the data portions need to be derandomized and reorganized. This process may advantageously occur in the data assembling modules, 525 and 620, of the authentication engine 215 and the cryptographic engine 220, respectively. The data assembling module, for example, data assembly module 525, receives data portions from the data storage facilities D1 through D4, and reassembles the data into useable form. For example, according to one embodiment

where the data splitting module 520 employed the data splitting process 800 of FIGURE 8, the data assembling module 525 uses data portions from at least two of the data storage facilities D1 through D4 to recreate the sensitive data S. For example, the pairings of AC, AD, BC, and BD, were distributed such that any two provide one of A and B, or, C and D. Noting that S = A XOR B or S = C XOR D indicates that when the data assembling module receives one of A and B, or, C and D, the data assembling module 525 can advantageously reassemble the sensitive data S. Thus, the data assembling module 525 may assemble the sensitive data S, when, for example, it receives data portions from at least the first two of the data storage facilities D1 through D4 to respond to an assemble request by the trust engine 110.

[0136] Based on the above data splitting and assembling processes, the sensitive data S exists in usable format only in a limited area of the trust engine 110. For example, when the sensitive data S includes enrollment authentication data, usable, nonrandomized enrollment authentication data is available only in the authentication engine 215. Likewise, when the sensitive data S includes private cryptographic key data, usable, nonrandomized private cryptographic key data is available only in the cryptographic engine 220.

[0137] Although the data splitting and assembling processes are disclosed with reference to their preferred embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for splitting and reassembling the sensitive data S. For example, public-key encryption

may be used to further secure the data at the data storage facilities D1 through D4. In addition, it is readily apparent to those of ordinary skill in the art that the data splitting module described herein is also a separate and distinct embodiment of the present invention that may be incorporated into, combined with or otherwise made part of any pre-existing computer systems, software suites, database, or combinations thereof, or other embodiments of the present invention, such as the trust engine, authentication engine, and transaction engine disclosed and described herein.

[0138] FIGURE 9A illustrates a data flow of an enrollment process 900 according to aspects of an embodiment of the invention. As shown in FIGURE 9A, the enrollment process 900 begins at step 905 when a user desires to enroll with the trust engine 110 of the cryptographic system 100. According to this embodiment, the user system 105 advantageously includes a client-side applet, such as a Java-based, that queries the user to enter enrollment data, such as demographic data and enrollment authentication data. According to one embodiment, the enrollment authentication data includes user ID, password(s), biometric(s), or the like. According to one embodiment, during the querying process, the client-side applet preferably communicates with the trust engine 110 to ensure that a chosen user ID is unique. When the user ID is nonunique, the trust engine 110 may advantageously suggest a unique user ID. The client-side applet gathers the enrollment data and transmits the enrollment data, for example, through and XML document, to the trust engine 110, and in particular, to the transaction engine 205. According

to one embodiment, the transmission is encoded with the public key of the authentication engine 215.

[0139] According to one embodiment, the user performs a single enrollment during step 905 of the enrollment process 900. For example, the user enrolls himself or herself as a particular person, such as Joe User. When Joe User desires to enroll as Joe User, CEO of Mega Corp., then according to this embodiment, Joe User enrolls a second time, receives a second unique user ID and the trust engine 110 does not associate the two identities. According to another embodiment of the invention, the enrollment process 900 provides for multiple user identities for a single user ID. Thus, in the above example, the trust engine 110 will advantageously associate the two identities of Joe User. As will be understood by a skilled artisan from the disclosure herein, a user may have many identities, for example, Joe User the head of household, Joe User the member of the Charitable Foundations, and the like. Even though the user may have multiple identities, according to this embodiment, the trust engine 110 preferably stores only one set of enrollment data. Moreover, users may advantageously add, edit/update, or delete identities as they are needed.

[0140] Although the enrollment process 900 is disclosed with reference to its preferred embodiment, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for gathering of enrollment data, and in particular, enrollment authentication data. For example, the applet may be common object model (COM) based applet or the like.

[0141] On the other hand, the enrollment process may include graded enrollment. For example, at a lowest level of enrollment, the user may enroll over the communication link 125 without producing documentation as to his or her identity. According to an increased level of enrollment, the user enrolls using a trusted third party, such as a digital notary. For example, and the user may appear in person to the trusted third party, produce credentials such as a birth certificate, driver's license, military ID, or the like, and the trusted third party may advantageously include, for example, their digital signature in enrollment submission. The trusted third party may include an actual notary, a government agency, such as the Post Office or Department of Motor Vehicles, a human resources person in a large company enrolling an employee, or the like. A skilled artisan will understand from the disclosure herein that a wide number of varying levels of enrollment may occur during the enrollment process 900.

[0142] After receiving the enrollment authentication data, at step 915, the transaction engine 205, using conventional FULL SSL technology forwards the enrollment authentication data to the authentication engine 215. In step 920, the authentication engine 215 decrypts the enrollment authentication data using the private key of the authentication engine 215. In addition, the authentication engine 215 employs the data splitting module to mathematically operate on the enrollment authentication data so as to split the data into at least two independently undecipherable, randomized, numbers. As mentioned in the foregoing, at least two numbers may comprise a statistically random

number and a binary X0Red number. In step 925, the authentication engine 215 forwards each portion of the randomized numbers to one of the data storage facilities D1 through D4. As mentioned in the foregoing, the authentication engine 215 may also advantageously randomize which portions are transferred to which depositories.

[0143] Often during the enrollment process 900, the user will also desire to have a digital certificate issued such that he or she may receive encrypted documents from others outside the cryptographic system 100. As mentioned in the foregoing, the certificate authority 115 generally issues digital certificates according to one or more of several conventional standards. Generally, the digital certificate includes a public key of the user or system, which is known to everyone.

[0144] Whether the user requests a digital certificate at enrollment, or at another time, the request is transferred through the trust engine 110 to the authentication engine 215. According to one embodiment, the request includes an XML document having, for example, the proper name of the user. According to step 935, the authentication engine 215 transfers the request to the cryptographic engine 220 instructing the cryptographic engine 220 to generate a cryptographic key or key pair.

[0145] Upon request, at step 935, the cryptographic engine 220 generates at least one cryptographic key. According to one embodiment, the cryptographic handling module 625 generates a key pair, where one key is used as a private key, and one is used as a public key. The cryptographic engine 220 stores the private key and,

according to one embodiment, a copy of the public key. In step 945, the cryptographic engine 220 transmits a request for a digital certificate to the transaction engine 205. According to one embodiment, the request advantageously includes a standardized request, such as PKCS10, embedded in, for example, an XML document. The request for a digital certificate may advantageously correspond to one or more certificate authorities and the one or more standard formats the certificate authorities require.

[0146] In step 950 the transaction engine 205 forwards this request to the certificate authority 115, who, in step 955, returns a digital certificate. The return digital certificate may advantageously be in a standardized format, such as PKCS7, or in a proprietary format of one or more of the certificate authorities 115. In step 960, the digital certificate is received by the transaction engine 205, and a copy is forwarded to the user and a copy is stored with the trust engine 110. The trust engine 110 stores a copy of the certificate such that the trust engine 110 will not need to rely on the availability of the certificate authority 115. For example, when the user desires to send a digital certificate, or a third party requests the user's digital certificate, the request for the digital certificate is typically sent to the certificate authority 115. However, if the certificate authority 115 is conducting maintenance or has been victim of a failure or security compromise, the digital certificate may not be available.

[0147] At any time after issuing the cryptographic keys, the cryptographic engine 220 may advantageously employ the data splitting process 800 described above

such that the cryptographic keys are split into independently undecipherable randomized numbers. Similar to the authentication data, at step 965 the cryptographic engine 220 transfers the randomized numbers to the data storage facilities D1 through D4.

[0148] A skilled artisan will recognize from the disclosure herein that the user may request a digital certificate anytime after enrollment. Moreover, the communications between systems may advantageously include FULL SSL or public-key encryption technologies. Moreover, the enrollment process may issue multiple digital certificates from multiple certificate authorities, including one or more proprietary certificate authorities internal or external to the trust engine 110.

[0149] As disclosed in steps 935 through 960, one embodiment of the invention includes the request for a certificate that is eventually stored on the trust engine 110. Because, according to one embodiment, the cryptographic handling module 625 issues the keys used by the trust engine 110, each certificate corresponds to a private key. Therefore, the trust engine 110 may advantageously provide for interoperability through monitoring the certificates owned by, or associated with, a user. For example, when the cryptographic engine 220 receives a request for a cryptographic function, the cryptographic handling module 625 may investigate the certificates owned by the requesting user to determine whether the user owns a private key matching the attributes of the request. When such a certificate exists, the cryptographic handling module 625 may use the certificate or the public or private keys associated therewith, to perform the requested

function. When such a certificate does not exist, the cryptographic handling module 625 may advantageously and transparently perform a number of actions to attempt to remedy the lack of an appropriate key. For example, FIGURE 9B illustrates a flowchart of an interoperability process 970, which according to aspects of an embodiment of the invention, discloses the foregoing steps to ensure the cryptographic handling module 625 performs cryptographic functions using appropriate keys.

[0150] As shown in FIGURE 9B, the interoperability process 970 begins with step 972 where the cryptographic handling module 925 determines the type of certificate desired. According to one embodiment of the invention, the type of certificate may advantageously be specified in the request for cryptographic functions, or other data provided by the requestor. According to another embodiment, the certificate type may be ascertained by the data format of the request. For example, the cryptographic handling module 925 may advantageously recognize the request corresponds to a particular type.

[0151] According to one embodiment, the certificate type may include one or more algorithm standards, for example, RSA, ELGAMAL, or the like. In addition, the certificate type may include one or more key types, such as symmetric keys, public keys, strong encryption keys such as 256 bit keys, less secure keys, or the like. Moreover, the certificate type may include upgrades or replacements of one or more of the foregoing algorithm standards or keys, one or more message or data formats, one or more data encapsulation or encoding schemes, such as Base 32 or Base 64. The

certificate type may also include compatibility with one or more third-party cryptographic applications or interfaces, one or more communication protocols, or one or more certificate standards or protocols. A skilled artisan will recognize from the disclosure herein that other differences may exist in certificate types, and translations to and from those differences may be implemented as disclosed herein.

[0152] Once the cryptographic handling module 625 determines the certificate type, the interoperability process 970 proceeds to step 974, and determines whether the user owns a certificate matching the type determined in step 974. When the user owns a matching certificate, for example, the trust engine 110 has access to the matching certificate through, for example, prior storage thereof, the cryptographic handling module 825 knows that a matching private key is also stored within the trust engine 110. For example, the matching private key may be stored within the depository 210 or depository system 700. The cryptographic handling module 625 may advantageously request the matching private key be assembled from, for example, the depository 210, and then in step 976, use the matching private key to perform cryptographic actions or functions. For example, as mentioned in the foregoing, the cryptographic handling module 625 may advantageously perform hashing, hash comparisons, data encryption or decryption, digital signature verification or creation, or the like.

[0153] When the user does not own a matching certificate, the interoperability process 970 proceeds to step 978 where the cryptographic handling module 625 determines whether the users owns a cross-certified

certificate. According to one embodiment, cross-certification between certificate authorities occurs when a first certificate authority determines to trust certificates from a second certificate authority. In other words, the first certificate authority determines that certificates from the second certificate authority meets certain quality standards, and therefore, may be "certified" as equivalent to the first certificate authority's own certificates. Cross-certification becomes more complex when the certificate authorities issue, for example, certificates having levels of trust. For example, the first certificate authority may provide three levels of trust for a particular certificate, usually based on the degree of reliability in the enrollment process, while the second certificate authority may provide seven levels of trust. Cross-certification may advantageously track which levels and which certificates from the second certificate authority may be substituted for which levels and which certificates from the first. When the foregoing cross-certification is done officially and publicly between two certification authorities, the mapping of certificates and levels to one another is often called "chaining."

[0154] According to another embodiment of the invention, the cryptographic handling module 625 may advantageously develop cross-certifications outside those agreed upon by the certificate authorities. For example, the cryptographic handling module 625 may access a first certificate authority's certificate practice statement (CPS), or other published policy statement, and using, for example, the authentication tokens required by particular trust levels, match the

first certificate authority's certificates to those of another certificate authority.

[0155] When, in step 978, the cryptographic handling module 625 determines that the users owns a cross-certified certificate, the interoperability process 970 proceeds to step 976, and performs the cryptographic action or function using the cross-certified public key, private key, or both. Alternatively, when the cryptographic handling module 625 determines that the users does not own a cross-certified certificate, the interoperability process 970 proceeds to step 980, where the cryptographic handling module 625 selects a certificate authority that issues the requested certificate type, or a certificate cross-certified thereto. In step 982, the cryptographic handling module 625 determines whether the user enrollment authentication data, discussed in the foregoing, meets the authentication requirements of the chosen certificate authority. For example, if the user enrolled over a network by, for example, answering demographic and other questions, the authentication data provided may establish a lower level of trust than a user providing biometric data and appearing before a third-party, such as, for example, a notary. According to one embodiment, the foregoing authentication requirements may advantageously be provided in the chosen authentication authority's CPS.

[0156] When the user has provided the trust engine 110 with enrollment authentication data meeting the requirements of chosen certificate authority, the interoperability process 970 proceeds to step 984, where the cryptographic handling module 825 acquires the certificate from the chosen certificate authority.

According to one embodiment, the cryptographic handling module 625 acquires the certificate by following steps 945 through 960 of the enrollment process 900. For example, the cryptographic handling module 625 may advantageously employ one or more public keys from one or more of the key pairs already available to the cryptographic engine 220, to request the certificate from the certificate authority. According to another embodiment, the cryptographic handling module 625 may advantageously generate one or more new key pairs, and use the public keys corresponding thereto, to request the certificate from the certificate authority.

[0157] According to another embodiment, the trust engine 110 may advantageously include one or more certificate issuing modules capable of issuing one or more certificate types. According to this embodiment, the certificate issuing module may provide the foregoing certificate. When the cryptographic handling module 625 acquires the certificate, the interoperability process 970 proceeds to step 976, and performs the cryptographic action or function using the public key, private key, or both corresponding to the acquired certificate.

[0158] When the user, in step 982, has not provided the trust engine 110 with enrollment authentication data meeting the requirements of chosen certificate authority, the cryptographic handling module 625 determines, in step 986 whether there are other certificate authorities that have different authentication requirements. For example, the cryptographic handling module 625 may look for certificate authorities having lower authentication

requirements, but still issue the chosen certificates, or cross-certifications thereof.

[0159] When the foregoing certificate authority having lower requirements exists, the interoperability process 970 proceeds to step 980 and chooses that certificate authority. Alternatively, when no such certificate authority exists, in step 988, the trust engine 110 may request additional authentication tokens from the user. For example, the trust engine 110 may request new enrollment authentication data comprising, for example, biometric data. Also, the trust engine 110 may request the user appear before a trusted third party and provide appropriate authenticating credentials, such as, for example, appearing before a notary with a drivers license, social security card, bank card, birth certificate, military ID, or the like. When the trust engine 110 receives updated authentication data, the interoperability process 970 proceeds to step 984 and acquires the foregoing chosen certificate.

[0160] Through the foregoing interoperability process 970, the cryptographic handling module 625 advantageously provides seamless, transparent, translations and conversions between differing cryptographic systems. A skilled artisan will recognize from the disclosure herein, a wide number of advantages and implementations of the foregoing interoperable system. For example, the foregoing step 986 of the interoperability process 970 may advantageously include aspects of trust arbitrage, discussed in further detail below, where the certificate authority may under special circumstances accept lower levels of cross-certification. In

addition, the interoperability process 970 may include ensuring interoperability between and employment of standard certificate revocations, such as employing certificate revocation lists (CRL), online certificate status protocols (OCSP), or the like.

[0161] FIGURE 10 illustrates a data flow of an authentication process 1000 according to aspects of an embodiment of the invention. According to one embodiment, the authentication process 1000 includes gathering current authentication data from a user and comparing that to the enrollment authentication data of the user. For example, the authentication process 1000 begins at step 1005 where a user desires to perform a transaction with, for example, a vendor. Such transactions may include, for example, selecting a purchase option, requesting access to a restricted area or device of the vendor system 120, or the like. At step 1010, a vendor provides the user with a transaction ID and an authentication request. The transaction ID may advantageously include a 192 bit quantity having a 32 bit timestamp concatenated with a 128 bit random quantity, or a "nonce," concatenated with a 32 bit vendor specific constant. Such a transaction ID uniquely identifies the transaction such that copycat transactions can be refused by the trust engine 110.

[0162] The authentication request may advantageously include what level of authentication is needed for a particular transaction. For example, the vendor may specify a particular level of confidence that is required for the transaction at issue. If authentication cannot be made to this level of confidence, as will be discussed below, the transaction

will not occur without either further authentication by the user to raise the level of confidence, or a change in the terms of the authentication between the vendor and the server. These issues are discussed more completely below.

[0163] According to one embodiment, the transaction ID and the authentication request may be advantageously generated by a vendor-side applet or other software program. In addition, the transmission of the transaction ID and authentication data may include one or more XML documents encrypted using conventional SSL technology, such as, for example, ½ SSL, or, in other words vendor-side authenticated SSL.

[0164] After the user system 105 receives the transaction ID and authentication request, the user system 105 gathers the current authentication data, potentially including current biometric information, from the user. The user system 105, at step 1015, encrypts at least the current authentication data "B" and the transaction ID, with the public key of the authentication engine 215, and transfers that data to the trust engine 110. The transmission preferably comprises XML documents encrypted with at least conventional ½ SSL technology. In step 1020, the transaction engine 205 receives the transmission, preferably recognizes the data format or request in the URL or URI, and forwards the transmission to the authentication engine 215.

[0165] During steps 1015 and 1020, the vendor system 120, at step 1025, forwards the transaction ID and the authentication request to the trust engine 110, using the preferred FULL SSL technology. This communication may also include a vendor ID, although vendor

identification may also be communicated through a non-random portion of the transaction ID. At steps 1030 and 1035, the transaction engine 205 receives the communication, creates a record in the audit trail, and generates a request for the user's enrollment authentication data to be assembled from the data storage facilities D1 through D4. At step 1040, the depository system 700 transfers the portions of the enrollment authentication data corresponding to the user to the authentication engine 215. At step 1045, the authentication engine 215 decrypts the transmission using its private key and compares the enrollment authentication data to the current authentication data provided by the user.

[0166] The comparison of step 1045 may advantageously apply heuristical context sensitive authentication, as referred to in the forgoing, and discussed in further detail below. For example, if the biometric information received does not match perfectly, a lower confidence match results. In particular embodiments, the level of confidence of the authentication is balanced against the nature of the transaction and the desires of both the user and the vendor. Again, this is discussed in greater detail below.

[0167] At step 1050, the authentication engine 215 fills in the authentication request with the result of the comparison of step 1045. According to one embodiment of the invention, the authentication request is filled with a YES/NO or TRUE/FALSE result of the authentication process 1000. In step 1055 the filled-in authentication request is returned to the vendor for the vendor to act upon, for example,

allowing the user to complete the transaction that initiated the authentication request. According to one embodiment, a confirmation message is passed to the user.

[0168] Based on the foregoing, the authentication process 1000 advantageously keeps sensitive data secure and produces results configured to maintain the integrity of the sensitive data. For example, the sensitive data is assembled only inside the authentication engine 215. For example, the enrollment authentication data is undecipherable until it is assembled in the authentication engine 215 by the data assembling module, and the current authentication data is undecipherable until it is unwrapped by the conventional SSL technology and the private key of the authentication engine 215. Moreover, the authentication result transmitted to the vendor does not include the sensitive data, and the user may not even know whether he or she produced valid authentication data.

[0169] Although the authentication process 1000 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the authentication process 1000. For example, the vendor may advantageously be replaced by almost any requesting application, even those residing with the user system 105. For example, a client application, such as Microsoft Word, may use an application program interface (API) or a cryptographic API (CAPI) to request authentication before unlocking a document. Alternatively, a mail server, a network, a

cellular phone, a personal or mobile computing device, a workstation, or the like, may all make authentication requests that can be filled by the authentication process 1000. In fact, after providing the foregoing trusted authentication process 1000, the requesting application or device may provide access to or use of a wide number of electronic or computer devices or systems.

[0170] Moreover, the authentication process 1000 may employ a wide number of alternative procedures in the event of authentication failure. For example, authentication failure may maintain the same transaction ID and request that the user reenter his or her current authentication data. As mentioned in the foregoing, use of the same transaction ID allows the comparator of the authentication engine 215 to monitor and limit the number of authentication attempts for a particular transaction, thereby creating a more secure cryptographic system 100.

[0171] In addition, the authentication process 1000 may be advantageously be employed to develop elegant single sign-on solutions, such as, unlocking a sensitive data vault. For example, successful or positive authentication may provide the authenticated user the ability to automatically access any number of passwords for an almost limitless number of systems and applications. For example, authentication of a user may provide the user access to password, login, financial credentials, or the like, associated with multiple online vendors, a local area network, various personal computing devices, Internet service providers, auction providers, investment brokerages, or the like. By employing a sensitive data vault, users may choose

truly large and random passwords because they no longer need to remember them through association. Rather, the authentication process 1000 provides access thereto. For example, a user may choose a random alphanumeric string that is twenty plus digits in length rather than something associated with a memorable data, name, etc.

[0172] According to one embodiment, a sensitive data vault associated with a given user may advantageously be stored in the data storage facilities of the depository 210, or split and stored in the depository system 700. According to this embodiment, after positive user authentication, the trust engine 110 serves the requested sensitive data, such as, for example, to the appropriate password to the requesting application. According to another embodiment, the trust engine 110 may include a separate system for storing the sensitive data vault. For example, the trust engine 110 may include a stand-alone software engine implementing the data vault functionality and figuratively residing "behind" the foregoing front-end security system of the trust engine 110. According to this embodiment, the software engine serves the requested sensitive data after the software engine receives a signal indicating positive user authentication from the trust engine 110.

[0173] In yet another embodiment, the data vault may be implemented by a third-party system. Similar to the software engine embodiment, the third-party system may advantageously serve the requested sensitive data after the third-party system receives a signal indicating positive user authentication from the trust engine 110. According to yet another embodiment, the data vault may be implemented on the user system 105. A user-side

software engine may advantageously serve the foregoing data after receiving a signal indicating positive user authentication from the trust engine 110.

[0174] Although the foregoing data vaults are disclosed with reference to alternative embodiments, a skilled artisan will recognize from the disclosure herein, a wide number of additional implementations thereof. For example, a particular data vault may include aspects from some or all of the foregoing embodiments. In addition, any of the foregoing data vaults may employ one or more authentication requests at varying times. For example, any of the data vaults may require authentication every one or more transactions, periodically, every one or more sessions, every access to one or more Webpages or Websites, at one or more other specified intervals, or the like.

[0175] FIGURE 11 illustrates a data flow of a signing process 1100 according to aspects of an embodiment of the invention. As shown in FIGURE 11, the signing process 1100 includes steps similar to those of the authentication process 1000 described in the foregoing with reference to FIGURE 10. According to one embodiment of the invention, the signing process 1100 first authenticates the user and then performs one or more of several digital signing functions as will be discussed in further detail below. According to another embodiment, the signing process 1100 may advantageously store data related thereto, such as hashes of messages or documents, or the like. This data may advantageously be used in an audit or any other event, such as for example, when a participating party attempts to repudiate a transaction.

[0176] As shown in FIGURE 11, during the authentication steps, the user and vendor may advantageously agree on a message, such as, for example, a contract. During signing, the signing process 1100 advantageously ensures that the contract signed by the user is identical to the contract supplied by the vendor. Therefore, according to one embodiment, during authentication, the vendor and the user include a hash of their respective copies of the message or contract, in the data transmitted to the authentication engine 215. By employing only a hash of a message or contract, the trust engine 110 may advantageously store a significantly reduced amount of data, providing for a more efficient and cost effective cryptographic system. In addition, the stored hash may be advantageously compared to a hash of a document in question to determine whether the document in question matches one signed by any of the parties. The ability to determine whether the document is identical to one relating to a transaction provides for additional evidence that can be used against a claim for repudiation by a party to a transaction.

[0177] In step 1103, the authentication engine 215 assembles the enrollment authentication data and compares it to the current authentication data provided by the user. When the comparator of the authentication engine 215 indicates that the enrollment authentication data matches the current authentication data, the comparator of the authentication engine 215 also compares the hash of the message supplied by the vendor to the hash of the message supplied by the user. Thus, the authentication engine 215 advantageously ensures

that the message agreed to by the user is identical to that agreed to by the vendor.

[0178] In step 1105, the authentication engine 215 transmits a digital signature request to the cryptographic engine 220. According to one embodiment of the invention, the request includes a hash of the message or contract. However, a skilled artisan will recognize from the disclosure herein that the cryptographic engine 220 may encrypt virtually any type of data, including, but not limited to, video, audio, biometrics, images or text to form the desired digital signature. Returning to step 1105, the digital signature request preferably comprises an XML document communicated through conventional SSL technologies.

[0179] In step 1110, the authentication engine 215 transmits a request to each of the data storage facilities D1 through D4, such that each of the data storage facilities D1 through D4 transmit their respective portion of the cryptographic key or keys corresponding to a signing party. According to another embodiment, the cryptographic engine 220 employs some or all of the steps of the interoperability process 970 discussed in the foregoing, such that the cryptographic engine 220 first determines the appropriate key or keys to request from the depository 210 or the depository system 700 for the signing party, and takes actions to provide appropriate matching keys. According to still another embodiment, the authentication engine 215 or the cryptographic engine 220 may advantageously request one or more of the keys associated with the signing party and stored in the depository 210 or depository system 700.

[0180] According to one embodiment, the signing party includes one or both the user and the vendor. In such case, the authentication engine 215 advantageously requests the cryptographic keys corresponding to the user and/or the vendor. According to another embodiment, the signing party includes the trust engine 110. In this embodiment, the trust engine 110 is certifying that the authentication process 1000 properly authenticated the user, vendor, or both. Therefore, the authentication engine 215 requests the cryptographic key of the trust engine 110, such as, for example, the key belonging to the cryptographic engine 220, to perform the digital signature. According to another embodiment, the trust engine 110 performs a digital notary-like function. In this embodiment, the signing party includes the user, vendor, or both, along with the trust engine 110. Thus, the trust engine 110 provides the digital signature of the user and/or vendor, and then indicates with its own digital signature that the user and/or vendor were properly authenticated. In this embodiment, the authentication engine 215 may advantageously request assembly of the cryptographic keys corresponding to the user, the vendor, or both. According to another embodiment, the authentication engine 215 may advantageously request assembly of the cryptographic keys corresponding to the trust engine 110.

[0181] According to another embodiment, the trust engine 110 performs power of attorney-like functions. For example, the trust engine 110 may digitally sign the message on behalf of a third party. In such case, the authentication engine 215 requests the cryptographic keys associated with the third party.

According to this embodiment, the signing process 1100 may advantageously include authentication of the third party, before allowing power of attorney-like functions. In addition, the authentication process 1000 may include a check for third party constraints, such as, for example, business logic or the like dictating when and in what circumstances a particular third-party's signature may be used.

[0182] Based on the foregoing, in step 1110, the authentication engine requested the cryptographic keys from the data storage facilities D1 through D4 corresponding to the signing party. In step 1115, the data storage facilities D1 through D4 transmit their respective portions of the cryptographic key corresponding to the signing party to the cryptographic engine 220. According to one embodiment, the foregoing transmissions include SSL technologies. According to another embodiment, the foregoing transmissions may advantageously be super-encrypted with the public key of the cryptographic engine 220.

[0183] In step 1120, the cryptographic engine 220 assembles the foregoing cryptographic keys of the signing party and encrypts the message therewith, thereby forming the digital signature(s). In step 1125 of the signing process 1100, the cryptographic engine 220 transmits the digital signature(s) to the authentication engine 215. In step 1130, the authentication engine 215 transmits the filled-in authentication request along with a copy of the hashed message and the digital signature(s) to the transaction engine 205. In step 1135, the transaction engine 205 transmits a receipt comprising the transaction ID, an indication of whether the authentication was

successful, and the digital signature(s), to the vendor. According to one embodiment, the foregoing transmission may advantageously include the digital signature of the trust engine 110. For example, the trust engine 110 may encrypt the hash of the receipt with its private key, thereby forming a digital signature to be attached to the transmission to the vendor.

[0184] According to one embodiment, the transaction engine 205 also transmits a confirmation message to the user. Although the signing process 1100 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the signing process 1100. For example, the vendor may be replaced with a user application, such as an email application. For example, the user may wish to digitally sign a particular email with his or her digital signature. In such an embodiment, the transmission throughout the signing process 1100 may advantageously include only one copy of a hash of the message. Moreover, a skilled artisan will recognize from the disclosure herein that a wide number of client applications may request digital signatures. For example, the client applications may comprise word processors, spreadsheets, emails, voicemail, access to restricted system areas, or the like.

[0185] In addition, a skilled artisan will recognize from the disclosure herein that steps 1105 through 1120 of the signing process 1100 may advantageously employ some or all of the steps of the interoperability

process 970 of FIGURE 9B, thereby providing interoperability between differing cryptographic systems that may, for example, need to process the digital signature under differing signature types.

[0186] FIGURE 12 illustrates a data flow of an encryption/decryption process 1200 according to aspects of an embodiment of the invention. As shown in FIGURE 12, the decryption process 1200 begins by authenticating the user using the authentication process 1000. According to one embodiment, the authentication process 1000 includes in the authentication request, a synchronous session key. For example, in conventional PKI technologies, it is understood by skilled artisans that encrypting or decrypting data using public and private keys is mathematically intensive and may require significant system resources. However, in symmetric key cryptographic systems, or systems where the sender and receiver of a message share a single common key that is used to encrypt and decrypt a message, the mathematical operations are significantly simpler and faster. Thus, in the conventional PKI technologies, the sender of a message will generate synchronous session key, and encrypt the message using the simpler, faster symmetric key system. Then, the sender will encrypt the session key with the public key of the receiver. The encrypted session key will be attached to the synchronously encrypted message and both data are sent to the receiver. The receiver uses his or her private key to decrypt the session key, and then uses the session key to decrypt the message. Based on the foregoing, the simpler and faster symmetric key system is used for the majority of the encryption/decryption processing.

Thus, in the decryption process 1200, the decryption advantageously assumes that a synchronous key has been encrypted with the public key of the user. Thus, as mentioned in the foregoing, the encrypted session key is included in the authentication request.

[0187] Returning to the decryption process 1200, after the user has been authenticated in step 1205, the authentication engine 215 forwards the encrypted session key to the cryptographic engine 220. In step 1210, the authentication engine 215 forwards a request to each of the data storage facilities, D1 through D4, requesting the cryptographic key data of the user. In step 1215, each data storage facility, D1 through D4, transmits their respective portion of the cryptographic key to the cryptographic engine 220. According to one embodiment, the foregoing transmission is encrypted with the public key of the cryptographic engine 220.

[0188] In step 1220 of the decryption process 1200, the cryptographic engine 220 assembles the cryptographic key and decrypts the session key therewith. In step 1225, the cryptographic engine forwards the session key to the authentication engine 215. In step 1227, the authentication engine 215 fills in the authentication request including the decrypted session key, and transmits the filled-in authentication request to the transaction engine 205. In step 1230, the transaction engine 205 forwards the authentication request along with the session key to the requesting application or vendor. Then, according to one embodiment, the requesting application or vendor uses the session key to decrypt the encrypted message.

[0189] Although the decryption process 1200 is disclosed with reference to its preferred and

alternative embodiments, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the decryption process 1200. For example, the decryption process 1200 may forego synchronous key encryption and rely on full public-key technology. In such an embodiment, the requesting application may transmit the entire message to the cryptographic engine 220, or, may employ some type of compression or reversible hash in order to transmit the message to the cryptographic engine 220. A skilled artisan will also recognize from the disclosure herein that the foregoing communications may advantageously include XML documents wrapped in SSL technology.

[0190] The encryption/decryption process 1200 also provides for encryption of documents or other data. Thus, in step 1235, a requesting application or vendor may advantageously transmit to the transaction engine 205 of the trust engine 110, a request for the public key of the user. The requesting application or vendor makes this request because the requesting application or vendor uses the public key of the user, for example, to encrypt the session key that will be used to encrypt the document or message. As mentioned in the enrollment process 900, the transaction engine 205 stores a copy of the digital certificate of the user, for example, in the mass storage 225. Thus, in step 1240 of the encryption process 1200, the transaction engine 205 requests the digital certificate of the user from the mass storage 225. In step 1245, the mass storage 225 transmits the digital certificate corresponding to the user, to the transaction engine 205. In step 1250, the transaction engine 205 transmits the digital certificate to the requesting

application or vendor. According to one embodiment, the encryption portion of the encryption process 1200 does not include the authentication of a user. This is because the requesting vendor needs only the public key of the user, and is not requesting any sensitive data.

[0191] A skilled artisan will recognize from the disclosure herein that if a particular user does not have a digital certificate, the trust engine 110 may employ some or all of the enrollment process 900 in order to generate a digital certificate for that particular user. Then, the trust engine 110 may initiate the encryption/decryption process 1200 and thereby provide the appropriate digital certificate. In addition, a skilled artisan will recognize from the disclosure herein that steps 1220 and 1235 through 1250 of the encryption/decryption process 1200 may advantageously employ some or all of the steps of the interoperability process of FIGURE 9B, thereby providing interoperability between differing cryptographic systems that may, for example, need to process the encryption.

[0192] FIGURE 13 illustrates a simplified block diagram of a trust engine system 1300 according to aspects of yet another embodiment of the invention. As shown in FIGURE 13, the trust engine system 1300 comprises a plurality of distinct trust engines 1305, 1310, 1315, and 1320, respectively. To facilitate a more complete understanding of the invention, FIGURE 13 illustrates each trust engine, 1305, 1310, 1315, and 1320 as having a transaction engine, a depository, and an authentication engine. However, a skilled artisan will recognize that each transaction engine may advantageously comprise some, a combination, or all of

the elements and communication channels disclosed with reference to FIGURES 1-8. For example, one embodiment may advantageously include trust engines having one or more transaction engines, depositories, and cryptographic servers or any combinations thereof.

[0193] According to one embodiment of the invention, each of the trust engines 1305, 1310, 1315 and 1320 are geographically separated, such that, for example, the trust engine 1305 may reside in a first location, the trust engine 1310 may reside in a second location, the trust engine 1315 may reside in a third location, and the trust engine 1320 may reside in a fourth location. The foregoing geographic separation advantageously decreases system response time while increasing the security of the overall trust engine system 1300.

[0194] For example, when a user logs onto the cryptographic system 100, the user may be nearest the first location and may desire to be authenticated. As described with reference to FIGURE 10, to be authenticated, the user provides current authentication data, such as a biometric or the like, and the current authentication data is compared to that user's enrollment authentication data. Therefore, according to one example, the user advantageously provides current authentication data to the geographically nearest trust engine 1305. The transaction engine 1321 of the trust engine 1305 then forwards the current authentication data to the authentication engine 1322 also residing at the first location. According to another embodiment, the transaction engine 1321 forwards the current authentication data to one or more of the authentication engines of the trust engines 1310, 1315, or 1320.

[0195] The transaction engine 1321 also requests the assembly of the enrollment authentication data from the depositories of, for example, each of the trust engines, 1305 through 1320. According to this embodiment, each depository provides its portion of the enrollment authentication data to the authentication engine 1322 of the trust engine 1305. The authentication engine 1322 then employs the encrypted data portions from, for example, the first two depositories to respond, and assembles the enrollment authentication data into deciphered form. The authentication engine 1322 compares the enrollment authentication data with the current authentication data and returns an authentication result to the transaction engine 1321 of the trust engine 1305.

[0196] Based on the above, the trust engine system 1300 employs the nearest one of a plurality of geographically separated trust engines, 1305 through 1320, to perform the authentication process. According to one embodiment of the invention, the routing of information to the nearest transaction engine may advantageously be performed at client-side applets executing on one or more of the user system 105, vendor system 120, or certificate authority 115. According to an alternative embodiment, a more sophisticated decision process may be employed to select from the trust engines 1305 through 1320. For example, the decision may be based on the availability, operability, speed of connections, load, performance, geographic proximity, or a combination thereof, of a given trust engine.

[0197] In this way, the trust engine system 1300 lowers its response time while maintaining the security

advantages associated with geographically remote data storage facilities, such as those discussed with reference to FIGURE 7 where each data storage facility stores randomized portions of sensitive data. For example, a security compromise at, for example, the depository 1325 of the trust engine 1315 does not necessarily compromise the sensitive data of the trust engine system 1300. This is because the depository 1325 contains only non-decipherable randomized data that, without more, is entirely useless.

[0198] According to another embodiment, the trust engine system 1300 may advantageously include multiple cryptographic engines arranged similar to the authentication engines. The cryptographic engines may advantageously perform cryptographic functions such as those disclosed with reference to FIGURES 1-8. According to yet another embodiment, the trust engine system 1300 may advantageously replace the multiple authentication engines with multiple cryptographic engines, thereby performing cryptographic functions such as those disclosed with reference to FIGURES 1-8. According to yet another embodiment of the invention, the trust engine system 1300 may replace each multiple authentication engine with an engine having some or all of the functionality of the authentication engines, cryptographic engines, or both, as disclosed in the foregoing,

[0199] Although the trust engine system 1300 is disclosed with reference to its preferred and alternative embodiments, a skilled artisan will recognize that the trust engine system 1300 may comprise portions of trust engines 1305 through 1320. For example, the trust engine system 1300 may include

one or more transaction engines, one or more depositories, one or more authentication engines, or one or more cryptographic engines or combinations thereof.

[0200] FIGURE 14 illustrates a simplified block diagram of a trust engine System 1400 according to aspects of yet another embodiment of the invention. As shown in FIGURE 14, the trust engine system 1400 includes multiple trust engines 1405, 1410, 1415 and 1420. According to one embodiment, each of the trust engines 1405, 1410, 1415 and 1420, comprise some or all of the elements of trust engine 110 disclosed with reference to FIGURES 1-8. According to this embodiment, when the client side applets of the user system 105, the vendor system 120, or the certificate authority 115, communicate with the trust engine system 1400, those communications are sent to the IP address of each of the trust engines 1405 through 1420. Further, each transaction engine of each of the trust engines, 1405, 1410, 1415, and 1420, behaves similar to the transaction engine 1321 of the trust engine 1305 disclosed with reference to FIGURE 13. For example, during an authentication process, each transaction engine of each of the trust engines 1405, 1410, 1415, and 1420 transmits the current authentication data to their respective authentication engines and transmits a request to assemble the randomized data stored in each of the depositories of each of the trust engines 1405 through 1420. FIGURE 14 does not illustrate all of these communications; as such illustration would become overly complex. Continuing with the authentication process, each of the depositories then communicates its portion of the randomized data to each of the

authentication engines of the each of the trust engines 1405 through 1420. Each of the authentication engines of the each of the trust engines employs its comparator to determine whether the current authentication data matches the enrollment authentication data provided by the depositories of each of the trust engines 1405 through 1420. According to this embodiment, the result of the comparison by each of the authentication engines is then transmitted to a redundancy module of the other three trust engines. For example, the result of the authentication engine from the trust engine 1405 is transmitted to the redundancy modules of the trust engines 1410, 1415, and 1420. Thus, the redundancy module of the trust engine 1405 likewise receives the result of the authentication engines from the trust engines 1410, 1415, and 1420.

[0201] FIGURE 15 illustrates a block diagram of the redundancy module of FIGURE 14. The redundancy module comprises a comparator configured to receive the authentication result from three authentication engines and transmit that result to the transaction engine of the fourth trust engine. The comparator compares the authentication result form the three authentication engines, and if two of the results agree, the comparator concludes that the authentication result should match that of the two agreeing authentication engines. This result is then transmitted back to the transaction engine corresponding to the trust engine not associated with the three authentication engines.

[0202] Based on the foregoing, the redundancy module determines an authentication result from data received from authentication engines that are preferably geographically remote from the trust engine of that the

redundancy module. By providing such redundancy functionality, the trust engine system 1400 ensures that a compromise of the authentication engine of one of the trust engines 1405 through 1420, is insufficient to compromise the authentication result of the redundancy module of that particular trust engine. A skilled artisan will recognize that redundancy module functionality of the trust engine system 1400 may also be applied to the cryptographic engine of each of the trust engines 1405 through 1420. However, such cryptographic engine communication was not shown in FIGURE 14 to avoid complexity. Moreover, a skilled artisan will recognize a wide number of alternative authentication result conflict resolution algorithms for the comparator of FIGURE 15 are suitable for use in the present invention.

[0203] According to yet another embodiment of the invention, the trust engine system 1400 may advantageously employ the redundancy module during cryptographic comparison steps. For example, some or all of the foregoing redundancy module disclosure with reference to FIGURES 14 and 15 may advantageously be implemented during a hash comparison of documents provided by one or more parties during a particular transaction.

[0204] Although the foregoing invention has been described in terms of certain preferred and alternative embodiments, other embodiments will be apparent to those of ordinary skill in the art from the disclosure herein. For example, the trust engine 110 may issue short-term certificates, where the private cryptographic key is released to the user for a predetermined period of time. For example, current

certificate standards include a validity field that can be set to expire after a predetermined amount of time. Thus, the trust engine 110 may release a private key to a user where the private key would be valid for, for example, 24 hours. According to such an embodiment, the trust engine 110 may advantageously issue a new cryptographic key pair to be associated with a particular user and then release the private key of the new cryptographic key pair. Then, once the private cryptographic key is released, the trust engine 110 immediately expires any internal valid use of such private key, as it is no longer securable by the trust engine 110.

[0205] In addition, a skilled artisan will recognize that the cryptographic system 100 or the trust engine 110 may include the ability to recognize any type of devices, such as, but not limited to, a laptop, a cell phone, a network, a biometric device or the like. According to one embodiment, such recognition may come from data supplied in the request for a particular service, such as, a request for authentication leading to access or use, a request for cryptographic functionality, or the like. According to one embodiment, the foregoing request may include a unique device identifier, such as, for example, a processor ID. Alternatively, the request may include data in a particular recognizable data format. For example, mobile and satellite phones often do not include the processing power for full X509.v3 heavy encryption certificates, and therefore do not request them. According to this embodiment, the trust engine 110 may recognize the type of data format presented, and respond only in kind.

[0206] In an additional aspect of the system described above, context sensitive authentication can be provided using various techniques as will be described below. Context sensitive authentication, for example as shown in FIGURE 16, provides the possibility of evaluating not only the actual data which is sent by the user when attempting to authenticate himself, but also the circumstances surrounding the generation and delivery of that data. Such techniques may also support transaction specific trust arbitrage between the user and trust engine 110 or between the vendor and trust engine 110, as will be described below.

[0207] As discussed above, authentication is the process of proving that a user is who he says he is. Generally, authentication requires demonstrating some fact to an authentication authority. The trust engine 110 of the present invention represents the authority to which a user must authenticate himself. The user must demonstrate to the trust engine 110 that he is who he says he is by either: knowing something that only the user should know (knowledge-based authentication), having something that only the user should have (token-based authentication), or by being something that only the user should be (biometric-based authentication).

[0208] Examples of knowledge-based authentication include without limitation a password, PIN number, or lock combination. Examples of token-based authentication include without limitation a house key, a physical credit card, a driver's license, or a particular phone number. Examples of biometric-based authentication include without limitation a fingerprint, handwriting analysis, facial scan, hand

scan, ear scan, iris scan, vascular pattern, DNA, a voice analysis, or a retinal scan.

[0209] Each type of authentication has particular advantages and disadvantages, and each provides a different level of security. For example, it is generally harder to create a false fingerprint that matches someone else's than it is to overhear someone's password and repeat it. Each type of authentication also requires a different type of data to be known to the authenticating authority in order to verify someone using that form of authentication.

[0210] As used herein, "authentication" will refer broadly to the overall process of verifying someone's identity to be who he says he is. An "authentication technique" will refer to a particular type of authentication based upon a particular piece of knowledge, physical token, or biometric reading. "Authentication data" refers to information which is sent to or otherwise demonstrated to an authentication authority in order to establish identity. "Enrollment data" will refer to the data which is initially submitted to an authentication authority in order to establish a baseline for comparison with authentication data. An "authentication instance" will refer to the data associated with an attempt to authenticate by an authentication technique.

[0211] The internal protocols and communications involved in the process of authenticating a user is described with reference to FIGURE 10 above. The part of this process within which the context sensitive authentication takes place occurs within the comparison step shown as step 1045 of FIGURE 10. This step takes place within the authentication engine 215 and involves

assembling the enrollment data 410 retrieved from the depository 210 and comparing the authentication data provided by the user to it. One particular embodiment of this process is shown in FIGURE 16 and described below.

[0212] The current authentication data provided by the user and the enrollment data retrieved from the depository 210 are received by the authentication engine 215 in step 1600 of FIGURE 16. Both of these sets of data may contain data which is related to separate techniques of authentication. The authentication engine 215 separates the authentication data associated with each individual authentication instance in step 1605. This is necessary so that the authentication data is compared with the appropriate subset of the enrollment data for the user (e.g. fingerprint authentication data should be compared with fingerprint enrollment data, rather than password enrollment data).

[0213] Generally, authenticating a user involves one or more individual authentication instances, depending on which authentication techniques are available to the user. These methods are limited by the enrollment data which were provided by the user during his enrollment process (if the user did not provide a retinal scan when enrolling, he will not be able to authenticate himself using a retinal scan), as well as the means which may be currently available to the user (e.g. if the user does not have a fingerprint reader at his current location, fingerprint authentication will not be practical). In some cases, a single authentication instance may be sufficient to authenticate a user; however, in certain circumstances a combination of

multiple authentication instances may be used in order to more confidently authenticate a user for a particular transaction.

[0214] Each authentication instance consists of data related to a particular authentication technique (e.g. fingerprint, password, smart card, etc.) and the circumstances which surround the capture and delivery of the data for that particular technique. For example, a particular instance of attempting to authenticate via password will generate not only the data related to the password itself, but also circumstantial data, known as "metadata", related to that password attempt. This circumstantial data includes information such as: the time at which the particular authentication instance took place, the network address from which the authentication information was delivered, as well as any other information as is known to those of skill in the art which may be determined about the origin of the authentication data (the type of connection, the processor serial number, etc.).

[0215] In many cases, only a small amount of circumstantial metadata will be available. For example, if the user is located on a network which uses proxies or network address translation or another technique which masks the address of the originating computer, only the address of the proxy or router may be determined. Similarly, in many cases information such as the processor serial number will not be available because of either limitations of the hardware or operating system being used, disabling of such features by the operator of the system, or other

limitations of the connection between the user's system and the trust engine 110.

[0216] As shown in FIGURE 16, once the individual authentication instances represented within the authentication data are extracted and separated in step 1605, the authentication engine 215 evaluates each instance for its reliability in indicating that the user is who he claims to be. The reliability for a single authentication instance will generally be determined based on several factors. These may be grouped as factors relating to the reliability associated with the authentication technique, which are evaluated in step 1610, and factors relating to the reliability of the particular authentication data provided, which are evaluated in step 1815. The first group includes without limitation the inherent reliability of the authentication technique being used, and the reliability of the enrollment data being used with that method. The second group includes without limitation the degree of match between the enrollment data and the data provided with the authentication instance, and the metadata associated with that authentication instance. Each of these factors may vary independently of the others.

[0217] The inherent reliability of an authentication technique is based on how hard it is for an imposter to provide someone else's correct data, as well as the overall error rates for the authentication technique. For passwords and knowledge based authentication methods, this reliability is often fairly low because there is nothing that prevents someone from revealing their password to another person and for that second person to use that password. Even a more complex

knowledge based system may have only moderate reliability since knowledge may be transferred from person to person fairly easily. Token based authentication, such as having a proper smart card or using a particular terminal to perform the authentication, is similarly of low reliability used by itself, since there is no guarantee that the right person is in possession of the proper token.

[0218] However, biometric techniques are more inherently reliable because it is generally difficult to provide someone else with the ability to use your fingerprints in a convenient manner, even intentionally. Because subverting biometric authentication techniques is more difficult, the inherent reliability of biometric methods is generally higher than that of purely knowledge or token based authentication techniques. However, even biometric techniques may have some occasions in which a false acceptance or false rejection is generated. These occurrences may be reflected by differing reliabilities for different implementations of the same biometric technique. For example, a fingerprint matching system provided by one company may provide a higher reliability than one provided by a different company because one uses higher quality optics or a better scanning resolution or some other improvement which reduces the occurrence of false acceptances or false rejections.

[0219] Note that this reliability may be expressed in different manners. The reliability is desirably expressed in some metric which can be used by the heuristics 530 and algorithms of the authentication engine 215 to calculate the confidence level of each

authentication. One preferred mode of expressing these reliabilities is as a percentage or fraction. For instance, fingerprints might be assigned an inherent reliability of 97%, while passwords might only be assigned an inherent reliability of 50%. Those of skill in the art will recognize that these particular values are merely exemplary and may vary between specific implementations.

[0220] The second factor for which reliability must be assessed is the reliability of the enrollment. This is part of the "graded enrollment" process referred to above. This reliability factor reflects the reliability of the identification provided during the initial enrollment process. For instance, if the individual initially enrolls in a manner where they physically produce evidence of their identity to a notary or other public official, and enrollment data is recorded at that time and notarized, the data will be more reliable than data which is provided over a network during enrollment and only vouched for by a digital signature or other information which is not truly tied to the individual.

[0221] Other enrollment techniques with varying levels of reliability include without limitation: enrollment at a physical office of the trust engine 110 operator; enrollment at a user's place of employment; enrollment at a post office or passport office; enrollment through an affiliated or trusted party to the trust engine 110 operator; anonymous or pseudonymous enrollment in which the enrolled identity is not yet identified with a particular real individual, as well as such other means as are known in the art.

[0222] These factors reflect the trust between the trust engine 110 and the source of identification provided during the enrollment process. For instance, if enrollment is performed in association with an employer during the initial process of providing evidence of identity, this information may be considered extremely reliable for purposes within the company, but may be trusted to a lesser degree by a government agency, or by a competitor. Therefore, trust engines operated by each of these other organizations may assign different levels of reliability to this enrollment.

[0223] Similarly, additional data which is submitted across a network, but which is authenticated by other trusted data provided during a previous enrollment with the same trust engine 110 may be considered as reliable as the original enrollment data was, even though the latter data were submitted across an open network. In such circumstances, a subsequent notarization will effectively increase the level of reliability associated with the original enrollment data. In this way for example, an anonymous or pseudonymous enrollment may then be raised to a full enrollment by demonstrating to some enrollment official the identity of the individual matching the enrolled data.

[0224] The reliability factors discussed above are generally values which may be determined in advance of any particular authentication instance. This is because they are based upon the enrollment and the technique, rather than the actual authentication. In one embodiment, the step of generating reliability based upon these factors involves looking up previously determined values for this particular authentication

technique and the enrollment data of the user. In a further aspect of an advantageous embodiment of the present invention, such reliabilities may be included with the enrollment data itself. In this way, these factors are automatically delivered to the authentication engine 215 along with the enrollment data sent from the depository 210.

[0225] While these factors may generally be determined in advance of any individual authentication instance, they still have an effect on each authentication instance which uses that particular technique of authentication for that user. Furthermore, although the values may change over time (e.g. if the user re-enrolls in a more reliable fashion), they are not dependent on the authentication data itself. By contrast, the reliability factors associated with a single specific instance's data may vary on each occasion. These factors, as discussed below, must be evaluated for each new authentication in order to generate reliability scores in step 1815.

[0226] The reliability of the authentication data reflects the match between the data provided by the user in a particular authentication instance and the data provided during the authentication enrollment. This is the fundamental question of whether the authentication data matches the enrollment data for the individual the user is claiming to be. Normally, when the data do not match, the user is considered to not be successfully authenticated, and the authentication fails. The manner in which this is evaluated may change depending on the authentication technique used. The comparison of such data is performed by the

comparator 515 function of the authentication engine 215 as shown in FIGURE 5.

[0227] For instance, matches of passwords are generally evaluated in a binary fashion. In other words, a password is either a perfect match, or a failed match. It is usually not desirable to accept as even a partial match a password which is close to the correct password if it is not exactly correct. Therefore, when evaluating a password authentication, the reliability of the authentication returned by the comparator 515 is typically either 100% (correct) or 0% (wrong), with no possibility of intermediate values.

[0228] Similar rules to those for passwords are generally applied to token based authentication methods, such as smart cards. This is because having a smart card which has a similar identifier or which is similar to the correct one, is still just as wrong as having any other incorrect token. Therefore tokens tend also to be binary authenticators: a user either has the right token, or he doesn't.

[0229] However, certain types of authentication data, such as questionnaires and biometrics, are generally not binary authenticators. For example, a fingerprint may match a reference fingerprint to varying degrees. To some extent, this may be due to variations in the quality of the data captured either during the initial enrollment or in subsequent authentications. (A fingerprint may be smudged or a person may have a still healing scar or burn on a particular finger.) In other instances the data may match less than perfectly because the information itself is somewhat variable and based upon pattern matching. (A voice analysis may seem close but not

quite right because of background noise, or the acoustics of the environment in which the voice is recorded, or because the person has a cold.) Finally, in situations where large amounts of data are being compared, it may simply be the case that much of the data matches well, but some doesn't. (A ten-question questionnaire may have resulted in eight correct answers to personal questions, but two incorrect answers.) For any of these reasons, the match between the enrollment data and the data for a particular authentication instance may be desirably assigned a partial match value by the comparator 515. In this way, the fingerprint might be said to be a 85% match, the voice print a 65% match, and the questionnaire an 80% match, for example.

[0230] This measure (degree of match) produced by the comparator 515 is the factor representing the basic issue of whether an authentication is correct or not. However, as discussed above, this is only one of the factors which may be used in determining the reliability of a given authentication instance. Note also that even though a match to some partial degree may be determined, that ultimately, it may be desirable to provide a binary result based upon a partial match. In an alternate mode of operation, it is also possible to treat partial matches as binary, i.e. either perfect (100%) or failed (0%) matches, based upon whether or not the degree of match passes a particular threshold level of match. Such a process may be used to provide a simple pass/fail level of matching for systems which would otherwise produce partial matches.

[0231] Another factor to be considered in evaluating the reliability of a given authentication instance

concerns the circumstances under which the authentication data for this particular instance are provided. As discussed above, the circumstances refer to the metadata associated with a particular authentication instance. This may include without limitation such information as: the network address of the authenticator, to the extent that it can be determined; the time of the authentication; the mode of transmission of the authentication data (phone line, cellular, network, etc.); and the serial number of the system of the authenticator.

[0232] These factors can be used to produce a profile of the type of authentication that is normally requested by the user. Then, this information can be used to assess reliability in at least two manners. One manner is to consider whether the user is requesting authentication in a manner which is consistent with the normal profile of authentication by this user. If the user normally makes authentication requests from one network address during business days (when she is at work) and from a different network address during evenings or weekends (when she is at home), an authentication which occurs from the home address during the business day is less reliable because it is outside the normal authentication profile. Similarly, if the user normally authenticates using a fingerprint biometric and in the evenings, an authentication which originates during the day using only a password is less reliable.

[0233] An additional way in which the circumstantial metadata can be used to evaluate the reliability of an instance of authentication is to determine how much corroboration the circumstance provides that the

authenticator is the individual he claims to be. For instance, if the authentication comes from a system with a serial number known to be associated with the user, this is a good circumstantial indicator that the user is who they claim to be. Conversely, if the authentication is coming from a network address which is known to be in Los Angeles when the user is known to reside in London, this is an indication that this authentication is less reliable based on its circumstances.

[0234] It is also possible that a cookie or other electronic data may be placed upon the system being used by a user when they interact with a vendor system or with the trust engine 110. This data is written to the storage of the system of the user and may contain an identification which may be read by a Web browser or other software on the user system. If this data is allowed to reside on the user system between sessions (a "persistent cookie"), it may be sent with the authentication data as further evidence of the past use of this system during authentication of a particular user. In effect, the metadata of a given instance, particularly a persistent cookie, may form a sort of token based authenticator itself.

[0235] Once the appropriate reliability factors based on the technique and data of the authentication instance are generated as described above in steps 1610 and 1615 respectively, they are used to produce an overall reliability for the authentication instance provided in step 1620. One means of doing this is simply to express each reliability as a percentage and then to multiply them together.

[0236] For example, suppose the authentication data is being sent in from a network address known to be the user's home computer completely in accordance with the user's past authentication profile (100%), and the technique being used is fingerprint identification (97%), and the initial finger print data was roistered through the user's employer with the trust engine 110 (90%), and the match between the authentication data and the original fingerprint template in the enrollment data is very good (99%). The overall reliability of this authentication instance could then be calculated as the product of these reliabilities: 100% * 97% * 90% * 99% - 86.4% reliability.

[0237] This calculated reliability represents the reliability of one single instance of authentication. The overall reliability of a single authentication instance may also be calculated using techniques which treat the different reliability factors differently, for example by using formulas where different weights are assigned to each reliability factor. Furthermore, those of skill in the art will recognize that the actual values used may represent values other than percentages and may use non-arithmetic systems. One embodiment may include a module used by an authentication requestor to set the weights for each factor and the algorithms used in establishing the overall reliability of the authentication instance.

[0238] The authentication engine 215 may use the above techniques and variations thereof to determine the reliability of a single authentication instance, indicated as step 1620. However, it may be useful in many authentication situations for multiple authentication instances to be provided at the same

time. For example, while attempting to authenticate himself using the system of the present invention, a user may provide a user identification, fingerprint authentication data, a smart card, and a password. In such a case, three independent authentication instances are being provided to the trust engine 110 for evaluation. Proceeding to step 1625, if the authentication engine 215 determines that the data provided by the user includes more than one authentication instance, then each instance in turn will be selected as shown in step 1630 and evaluated as described above in steps 1610, 1615 and 1620.

[0239] Note that many of the reliability factors discussed may vary from one of these instances to another. For instance, the inherent reliability of these techniques is likely to be different, as well as the degree of match provided between the authentication data and the enrollment data. Furthermore, the user may have provided enrollment data at different times and under different circumstances for each of these techniques, providing different enrollment reliabilities for each of these instances as well. Finally, even though the circumstances under which the data for each of these instances is being submitted is the same, the use of such techniques may each fit the profile of the user differently, and so may be assigned different circumstantial reliabilities. (For example, the user may normally use their password and fingerprint, but not their smart card.)

[0240] As a result, the final reliability for each of these authentication instances may be different from One another. However, by using multiple instances

together, the overall confidence level for the authentication will tend to increase.

[0241] Once the authentication engine has performed steps 1610 through 1620 for all of the authentication instances provided in the authentication data, the reliability of each instance is used in step 1635 to evaluate the overall authentication confidence level. This process of combining the individual authentication instance reliabilities into the authentication confidence level may be modeled by various methods relating the individual reliabilities produced, and may also address the particular interaction between some of these authentication techniques. (For example, multiple knowledge-based systems such as passwords may produce less confidence than a single password and even a fairly weak biometric, such as a basic voice analysis.)

[0242] One means in which the authentication engine 215 may combine the reliabilities of multiple concurrent authentication instances to generate a final confidence level is to multiply the unreliability of each instance to arrive at a total unreliability. The unreliability is generally the complementary percentage of the reliability. For example, a technique which is 84% reliable is 16% unreliable. The three authentication instances described above (fingerprint, smart card, password)which produce reliabilities of 86%, 75%, and 72% would have corresponding unreliabilities of (100- 86)%, (100- 75)% and (100- 72)%, or 14%, 25%, and 28%, respectively. By multiplying these unreliabilities, we get a cumulative unreliability of 14% * 25% * 28% - .98% unreliability, which corresponds to a reliability of 99.02%.

[0243] In an additional mode of operation, additional factors and heuristics 530 may be applied within the authentication engine 215 to account for the interdependence of various authentication techniques. For example, if someone has unauthorized access to a particular home computer, they probably have access to the phone line at that address as well. Therefore, authenticating based on an originating phone number as well as upon the serial number of the authenticating system does not add much to the overall confidence in the authentication. However, knowledge based authentication is largely independent of token based authentication (i.e. if someone steals your cellular phone or keys, they are no more likely to know your PIN or password than if they hadn't).

[0244] Furthermore, different vendors or other authentication requestors may wish to weigh different aspects of the authentication differently. This may include the use of separate weighing factors or algorithms used in calculating the reliability of individual instances as well as the use of different means to evaluate authentication events with multiple instances.

[0245] For instance, vendors for certain types of transactions, for instance corporate email systems, may desire to authenticate primarily based upon heuristics and other circumstantial data by default. Therefore, they may apply high weights to factors related to the metadata and other profile related information associated with the circumstances surrounding authentication events. This arrangement could be used to ease the burden on users during normal operating hours, by not requiring more from the user than that he

be logged on to the correct machine during business hours. However, another vendor may weigh authentications coming from a particular technique most heavily, for instance fingerprint matching, because of a policy decision that such a technique is most suited to authentication for the particular vendor's purposes.

[0246] Such varying weights may be defined by the authentication requestor in generating the authentication request and sent to the trust engine 110 with the authentication request in one mode of operation. Such options could also be set as preferences during an initial enrollment process for the authentication requestor and stored within the authentication engine in another mode of operation.

[0247] Once the authentication engine 215 produces an authentication confidence level for the authentication data provided, this confidence level is used to complete the authentication request in step 1640, and this information is forwarded from the authentication engine 215 to the transaction engine 205 for inclusion in a message to the authentication requestor.

[0248] The process described above is merely exemplary, and those of skill in the art will recognize that the steps need not be performed in the order shown or that only certain of the steps are desired to be performed, or that a variety of combinations of steps may be desired. Furthermore, certain steps, such as the evaluation of the reliability of each authentication instance provided, may be carried out in parallel with one another if circumstances permit.

[0249] In a further aspect of this invention, a method is provided to accommodate conditions when the

authentication confidence level produced by the process described above fails to meet the required trust level of the vendor or other party requiring the authentication. In circumstances such as these where a gap exists between the level of confidence provided and the level of trust desired, the operator of the trust engine 110 is in a position to provide opportunities for one or both parties to provide alternate data or requirements in order to close this trust gap. This process will be referred to as "trust arbitrage" herein.

[0250] Trust arbitrage may take place within a framework of cryptographic authentication as described above with reference to FIGURES 10 and 11. As shown therein, a vendor or other party will request authentication of a particular user in association with a particular transaction. In one circumstance, the vendor simply requests an authentication, either positive or negative, and after receiving appropriate data from the user, the trust engine 110 will provide such a binary authentication. In circumstances such as these, the degree of confidence required in order to secure a positive authentication is determined based upon preferences set within the trust engine 110.

[0251] However, it is also possible that the vendor may request a particular level of trust in order to complete a particular transaction. This required level may be included with the authentication request (e.g. authenticate this user to 98% confidence) or may be determined by the trust engine 110 based on other factors associated with the transaction (i.e. authenticate this user as appropriate for this transaction). One such factor might be the economic

value of the transaction. For transactions which have greater economic value, a higher degree of trust may be required. Similarly, for transactions with high degrees of risk a high degree of trust may be required. Conversely, for transactions which are either of low risk or of low value, lower trust levels may be required by the vendor or other authentication requestor.

[0252] The process of trust arbitrage occurs between the steps of the trust engine 110 receiving the authentication data in step 1050 of FIGURE 10 and the return of an authentication result to the vendor in step 1055 of FIGURE 10. Between these steps, the process which leads to the evaluation of trust levels and the potential trust arbitrage occurs as shown in FIGURE 17. In circumstances where simple binary authentication is performed, the process shown in FIGURE 17 reduces to having the transaction engine 205 directly compare the authentication data provided with the enrollment data for the identified user as discussed above with reference to FIGURE 10, flagging any difference as a negative authentication.

[0253] As shown in FIGURE 17, the first step after receiving the data in step 1050 is for the transaction engine 205 to determine the trust level which is required for a positive authentication for this particular transaction in step 1710. This step may be performed by one of several different methods. The required trust level may be specified to the trust engine 110 by the authentication requestor at the time when the authentication request is made. The authentication requestor may also set a preference in advance which is stored within the depository 210 or

other storage which is accessible by the transaction engine 205. This preference may then be read and used each time an authentication request is made by this authentication requestor. The preference may also be associated with a particular user as a security measure such that a particular level of trust is always required in order to authenticate that user, the user preference being stored in the depository 210 or other storage media accessible by the transaction engine 205. The required level may also be derived by the transaction engine 205 or authentication engine 215 based upon information provided in the authentication request, such as the value and risk level of the transaction to be authenticated.

[0254] In one mode of operation, a policy management module or other software which is used when generating the authentication request is used to specify the required degree of trust for the authentication of the transaction. This may be used to provide a series of rules to follow when assigning the required level of trust based upon the policies which are specified within the policy management module. One advantageous mode of operation is for such a module to be incorporated with the web server of a vendor in order to appropriately determine required level of trust for transactions initiated with the vendor's web server. In this way, transaction requests from users may be assigned a required trust level in accordance with the policies of the vendor and such information may be forwarded to the trust engine 110 along with the authentication request.

[0255] This required trust level correlates with the degree of certainty that the vendor wants to have that

the individual authenticating is in fact who he identifies himself as. For example, if the transaction is one where the vendor wants a fair degree of certainty because goods are changing hands, the vendor may require a trust level of 85%. For situation where the vendor is merely authenticating the user to allow him to view members only content or exercise privileges on a chat room, the downside risk may be small enough that the vendor requires only a 60% trust level. However, to enter into a production contract with a value of tens of thousands of dollars, the vendor may require a trust level of 99% or more.

[0256] This required trust level represents a metric to which the user must authenticate himself in order to complete the transaction. If the required trust level is 85% for example, the user must provide authentication to the trust engine 110 sufficient for the trust engine 110 to say with 85% confidence that the user is who they say they are. It is the balance between this required trust level and the authentication confidence level which produces either a positive authentication (to the satisfaction of the vendor) or a possibility of trust arbitrage.

[0257] As shown in FIGURE 17, after the transaction engine 205 receives the required trust level, it compares in step 1720 the required trust level to the authentication confidence level which the authentication engine 215 calculated for the current authentication (as discussed with reference to FIGURE 16). If the authentication confidence level is higher than the required trust level for the transaction in step 1730, then the process moves to step 1740 where a positive authentication for this transaction is

produced by the transaction engine 205. A message to this effect will then be inserted into the authentication results and returned to the vendor by the transaction engine 205 as shown in step 1055 (see FIGURE 10).

[0258] However, if the authentication confidence level does not fulfill the required trust level in step 1730, then a confidence gap exists for the current authentication, and trust arbitrage is conducted in step 1750. Trust arbitrage is described more completely with reference to FIGURE 18 below. This process as described below takes place within the transaction engine 205 of the trust engine 110. Because no authentication or other cryptographic operations are needed to execute trust arbitrage (other than those required for the SSL communication between the transaction engine 205 and other components), the process may be performed outside the authentication engine 215. However, as will be discussed below, any reevaluation of authentication data or other cryptographic or authentication events will require the transaction engine 205 to resubmit the appropriate data to the authentication engine 215. Those of skill in the art will recognize that the trust arbitrage process could alternately be structured to take place partially or entirely within the authentication engine 215 itself.

[0259] As mentioned above, trust arbitrage is a process where the trust engine 110 mediates a negotiation between the vendor and user in an attempt to secure a positive authentication where appropriate. As shown in step 1805, the transaction engine 205 first determines whether or not the current situation is

appropriate for trust arbitrage. This may be determined based upon the circumstances of the authentication, e.g. whether this authentication has already been through multiple cycles of arbitrage, as well as upon the preferences of either the vendor or user, as will be discussed further below.

[0260] In such circumstances where arbitrage is not possible, the process proceeds to step 1810 where the transaction engine 205 generates a negative authentication and then inserts it into the authentication results which are sent to the vendor in step 1055 (see FIGURE 10). One limit which may be advantageously used to prevent authentications from pending indefinitely is to set a time-out period from the initial authentication request. In this way, any transaction which is not positively authenticated within the time limit is denied further arbitrage and negatively authenticated. Those of skill in the art will recognize that such a time limit may vary depending upon the circumstances of the transaction and the desires of the user and vendor. Limitations may also be placed upon the number of attempts that may be made at providing a successful authentication. Such limitations may be handled by an attempt limiter 535 as shown in FIGURE 5.

[0261] If arbitrage is not prohibited in step 1805, the transaction engine 205 will then engage in negotiation with one or both of the transacting parties. The transaction engine 205 may send a message to the user requesting some form of additional authentication in order to boost the authentication confidence level produced as shown in step 1820. In the simplest form, this may simply indicates that

authentication was insufficient. A request to produce one or more additional authentication instances to improve the overall confidence level of the authentication may also be sent.

[0262] If the user provides some additional authentication instances in step 1825, then the transaction engine 205 adds these authentication instances to the authentication data for the transaction and forwards it to the authentication engine 215 as shown in step 1015 (see FIGURE 10), and the authentication is reevaluated based upon both the pre-existing authentication instances for this transaction and the newly provided authentication instances.

[0263] An additional type of authentication may be a request from the trust engine 110 to make some form of person-to-person contact between the trust engine 110 operator (or a trusted associate) and the user, for example, by phone call. This phone call or other non-computer authentication can be used to provide personal contact with the individual and also to conduct some form of questionnaire based authentication. This also may give the opportunity to verify an originating telephone number and potentially a voice analysis of the user when he calls in. Even if no additional authentication data can be provided, the additional context associated with the user's phone number may improve the reliability of the authentication context. Any revised data or circumstances based upon this phone call are fed into the trust engine 110 for use in consideration of the authentication request.

[0264] Additionally, in step 1820 the trust engine 110 may provide an opportunity for the user to purchase insurance, effectively buying a more confident authentication. The operator of the trust engine 110 may, at times, only want to make such an option available if the confidence level of the authentication is above a certain threshold to begin with. In effect, this user side insurance is a way for the trust engine 110 to vouch for the user when the authentication meets the normal required trust level of the trust engine 110 for authentication, but does not meet the required trust level of the vendor for this transaction. In this way, the user may still successfully authenticate to a very high level as may be required by the vendor, even though he only has authentication instances which produce confidence sufficient for the trust engine 110.

[0265] This function of the trust engine 110 allows the trust engine 110 to vouch for someone who is authenticated to the satisfaction of the trust engine 110, but not of the vendor. This is analogous to the function performed by a notary in adding his signature to a document in order to indicate to someone reading the document at a later time that the person whose signature appears on the document is in fact the person who signed it. The signature of the notary testifies to the act of signing by the user. In the same way, the trust engine is providing an indication that the person transacting is who they say they are.

[0266] However, because the trust engine 110 is artificially boosting the level of confidence provided by the user, there is a greater risk to the trust engine 110 operator, since the user is not actually meeting the required trust level of the vendor. The

cost of the insurance is designed to offset the risk of a false positive authentication to the trust engine 110 (who may be effectively notarizing the authentications of the user). The user pays the trust engine 110 operator to take the risk of authenticating to a higher level of confidence than has actually been provided.

[0267] Because such an insurance system allows someone to effectively buy a higher confidence rating from the trust engine 110, both vendors and users may wish to prevent the use of user side insurance in certain transactions. Vendors may wish to limit positive authentications to circumstances where they know that actual authentication data supports the degree of confidence which they require and so may indicate to the trust engine 110 that user side insurance is not to be allowed. Similarly, to protect his online identity, a user may wish to prevent the use of user side insurance on his account, or may wish to limit its use to situations where the authentication confidence level without the insurance is higher than a certain limit. This may be used as a security measure to prevent someone from overhearing a password or stealing a smart card and using them to falsely authenticate to a low level of confidence, and then purchasing insurance to produce a very high level of (false) confidence. These factors may be evaluated in determining whether user side insurance is allowed.

[0268] If user purchases insurance in step 1840, then the authentication confidence level is adjusted based upon the insurance purchased in step 1845, and the authentication confidence level and required trust level are again compared in step 1730 (see FIGURE 17). The process continues from there, and may lead to

either a positive authentication in step 1740 (see FIGURE 17), or back into the trust arbitrage process in step 1750 for either further arbitrage (if allowed) or a negative authentication in step 1810 if further arbitrage is prohibited.

[0269] In addition to sending a message to the user in step 1820, the transaction engine 205 may also send a message to the vendor in step 1830 which indicates that a pending authentication is currently below the required trust level. The message may also offer various options on how to proceed to the vendor. One of these Options is to simply inform the vendor of what the current authentication confidence level is and ask if the vendor wishes to maintain their current unfulfilled required trust level. This may be beneficial because in some cases, the vendor may have independent means for authenticating the transaction or may have been using a default set of requirements which generally result in a higher required level being initially specified than is actually needed for the particular transaction at hand.

[0270] For instance, it may be standard practice that all incoming purchase order transactions with the vendor are expected to meet a 98% trust level. However, if an order was recently discussed by phone between the vendor and a long-standing customer, and immediately thereafter the transaction is authenticated, but only to a 93% confidence level, the vendor may wish to simply lower the acceptance threshold for this transaction, because the phone call effectively provides additional authentication to the vendor. In certain circumstances, the vendor may be willing to lower their required trust level, but not

all the way to the level of the current authentication confidence. For instance, the vendor in the above example might consider that the phone call prior to the order might merit a 4% reduction in the degree of trust needed; however, this is still greater than the 93% confidence produced by the user.

[0271] If the vendor does adjust their required trust level in step 1835, then the authentication confidence level produced by the authentication and the required trust level are compared in step 1730 (see FIGURE 17). If the confidence level now exceeds the required trust level, a positive authentication may be generated in the transaction engine 205 in step 1740 (see FIGURE 17). If not, further arbitrage may be attempted as discussed above if it is permitted.

[0272] In addition to requesting an adjustment to the required trust level, the transaction engine 205 may also offer vendor side insurance to the vendor requesting the authentication. This insurance serves a similar purpose to that described above for the user side insurance. Here, however, rather than the cost corresponding to the risk being taken by the trust engine 110 in authenticating above the actual authentication confidence level produced, the cost of the insurance corresponds to the risk being taken by the vendor in accepting a lower trust level in the authentication.

[0273] Instead of just lowering their actual required trust level, the vendor has the option of purchasing insurance to protect itself from the additional risk associated with a lower level of trust in the authentication of the user. As described above, it may be advantageous for the vendor to only consider

purchasing such insurance to cover the trust gap in conditions where the existing authentication is already above a certain threshold.

[0274] The availability of such vendor side insurance allows the vendor the option to either: lower his trust requirement directly at no additional cost to himself, bearing the risk of a false authentication himself (based on the lower trust level required); or, buying insurance for the trust gap between the authentication confidence level and his requirement, with the trust engine 110 operator bearing the risk of the lower confidence level which has been provided. By purchasing the insurance, the vendor effectively keeps his high trust level requirement; because the risk of a false authentication is shifted to the trust engine 110 operator.

[0275] If the vendor purchases insurance in step 1840, the authentication confidence level and required trust levels are compared in step 1730 (see FIGURE 17), and the process continues as described above.

[0276] Note that it is also possible that both the user and the vendor respond to messages from the trust engine 110. Those of skill in the art will recognize that there are multiple ways in which such situations can be handled. One advantageous mode of handling the possibility of multiple responses is simply to treat the responses in a first-come, first-served manner. For example, if the vendor responds with a lowered required trust level and immediately thereafter the user also purchases insurance to raise his authentication level, the authentication is first reevaluated based upon the lowered trust requirement from the vendor. If the authentication is now

positive, the user's insurance purchase is ignored. In another advantageous mode of operation, the user might only be charged for the level of insurance required to meet the new, lowered trust requirement of the vendor (if a trust gap remained even with the lowered vendor trust requirement).

[0277] If no response from either party is received during the trust arbitrage process at step 1850 within the time limit set for the authentication, the arbitrage is reevaluated in step 1805. This effectively begins the arbitrage process again. If the time limit was final or other circumstances prevent further arbitrage in step 1805, a negative authentication is generated by the transaction engine 205 in step 1810 and returned to the vendor in step 1055 (see FIGURE 10). If not, new messages may be sent to the user and vendor, and the process may be repeated as desired.

[0278] Note that for certain types of transactions, for instance, digitally signing documents which are not part of a transaction, there may not necessarily be a vendor or other third party; therefore the transaction is primarily between the user and the trust engine 110. In circumstances such as these, the trust engine 110 will have its own required trust level which must be satisfied in order to generate a positive authentication. However, in such circumstances, it will often not be desirable for the trust engine 110 to offer insurance to the user in order for him to raise the confidence of his own signature.

[0279] The process described above and shown in FIGURES 16-18 may be carried out using various communications modes as described above with reference

to the trust engine 110. For instance, the messages may be web-based and sent using SSL connections between the trust engine 110 and applets downloaded in real time to browsers running on the user or vendor systems. In an alternate mode of operation, certain dedicated applications may be in use by the user and vendor which facilitate such arbitrage and insurance transactions. In another alternate mode of operation, secure email operations may be used to mediate the arbitrage described above, thereby allowing deferred evaluations and batch processing of authentications. Those of skill in the art will recognize that different communications modes may be used as are appropriate for the circumstances and authentication requirements of the vendor.

[0280] The following description with reference to FIGURE 19 describes a sample transaction which integrates the various aspects of the present invention as described above. This example illustrates the overall process between a user and a vendor as mediates by the trust engine 110. Although the various steps and components as described in detail above may be used to carry out the following transaction, the process illustrated focuses on the interaction between the trust engine 110, user and vendor.

[0281] The transaction begins when the user, while viewing web pages online, fills out an order form on the web site of the vendor in step 1900. The user wishes to submit this order form to the vendor, signed with his digital signature. In order to do this, the user submits the order form with his request for a signature to the trust engine 110 in step 1905. The user will also provide authentication data which will

be used as described above to authenticate his identity.

[0282] In step 1910 the authentication data is compared to the enrollment data by the trust engine 110 as discussed above, and if a positive authentication is produced, the hash of the order form, signed with the private key of the user, is forwarded to the vendor along with the order form itself.

[0283] The vendor receives the signed form in step 1915, and then the vendor will generate an invoice or other contract related to the purchase to be made in step 1920. This contract is sent back to the user with a request for a signature in step 1925. The vendor also sends an authentication request for this contract transaction to the trust engine 110 in step 1930 including a hash of the contract which will be signed by both parties. To allow the contract to be digitally signed by both parties, the vendor also includes authentication data for itself so that the vendor's signature upon the contract can later be verified if necessary.

[0284] As discussed above, the trust engine 110 then verifies the authentication data provided by the vendor to confirm the vendor's identity, and if the data produces a positive authentication in step 1935, continues with step 1955 when the data is received from the user. If the vendor's authentication data does not match the enrollment data of the vendor to the desired degree, a message is returned to the vendor requesting further authentication. Trust arbitrage may be performed here if necessary, as described above, in order for the vendor to successfully authenticate itself to the trust engine 110.

[0285] When the user receives the contract in step 1940, he reviews it, generates authentication data to sign it if it is acceptable in step 1945, and then sends a hash of the contract and his authentication data to the trust engine 110 in step 1950. The trust engine 110 verifies the authentication data in step 1955 and if the authentication is good, proceeds to process the contract as described below. As discussed above with reference to FIGURES 17 and 18, trust arbitrage may be performed as appropriate to close any trust gap which exists between the authentication confidence level and the required authentication level for the transaction.

[0286] The trust engine 110 signs the hash of the contract with the user's private key, and sends this signed hash to the vendor in step 1960, signing the complete message on its own behalf, i.e., including a hash of the complete message (including the user's signature) encrypted with the private key 510 of the trust engine 110. This message is received by the vendor in step 1965. The message represents a signed contract (hash of contract encrypted using user's private key) and a receipt from the trust engine 110 (the hash of the message including the signed contract, encrypted using the trust engine 110's private key).

[0287] The trust engine 110 similarly prepares a hash of the contract with the vendor's private key in step 1970, and forwards this to the user, signed by the trust engine 110. In this way, the user also receives a copy of the contract, signed by the vendor, as well as a receipt, signed by the trust engine 110, for delivery of the signed contract in step 1975.

[0288] In addition to the foregoing, an additional aspect of the invention provides a cryptographic Service Provider Module (SPM) which may be available to a client side application as a means to access functions provided by the trust engine 110 described above. One advantageous way to provide such a service is for the cryptographic SPM is to mediate communications between a third party Application Programming Interface (API) and a trust engine 110 which is accessible via a network or other remote connection. A sample cryptographic SPM is described below with reference to FIGURE 20.

[0289] For example, on a typical system, a number of API's are available to programmers. Each API provides a set of function calls which may be made by an application 2000 running upon the system. Examples of API's which provide programming interfaces suitable for cryptographic functions, authentication functions, and other security function include the Cryptographic API (CAPI) 2010 provided by Microsoft with its Windows operating systems, and the Common Data Security Architecture (CDSA), sponsored by IBM, Intel and other members of the Open Group. CAPI will be used as an exemplary security API in the discussion that follows. However, the cryptographic SPM described could be used with CDSA or other security API's as are known in the art.

[0290] This API is used by a user system 105 or vendor system 120 when a call is made for a cryptographic function. Included among these functions may be requests associated with performing various cryptographic operations, such as encrypting a document with a particular key, signing a document, requesting a

digital certificate, verifying a signature upon a signed document, and such other cryptographic functions as are described herein or known to those of skill in the art.

[0291] Such cryptographic functions are normally performed locally to the system upon which CAPI 2010 is located. This is because generally the functions called require the use of either resources of the local user system 105, such as a fingerprint reader, or software functions which are programmed using libraries which are executed on the local machine. Access to these local resources is normally provided by one or more Service Provider Modules (SPM's) 2015, 2020 as referred to above which provide resources with which the cryptographic functions are carried out. Such SPM's may include software libraries 2015 to perform encrypting or decrypting operations, or drivers and applications 2020 which are capable of accessing specialized hardware 2025, such as biometric scanning devices. In much the way that CAPI 2010 provides functions which may be used by an application 2000 of the system 105, the SPM's 2015, 2020 provide CAPI with access to the lower level functions and resources associated with the available services upon the system.

[0292] In accordance with the invention, it is possible to provide a cryptographic SPM 2030 which is capable of accessing the cryptographic functions provided by the trust engine 110 and making these functions available to an application 2000 through CAPI 2010. Unlike embodiments where CAPI 2010 is only able to access resources which are locally available through SPM's 2015, 2020, a cryptographic SPM 2030 as described herein would be able to submit requests for

cryptographic operations to a remotely-located, network-accessible trust engine 110 in order to perform the operations desired.

[0293] For instance, if an application 2000 has a need for a cryptographic operation, such as signing a document, the application 2000 makes a function call to the appropriate CAPI 2010 function. CAPI 2010 in turn will execute this function, making use of the resources which are made available to it by the SPM's 2015, 2020 and the cryptographic SPM 2030. In the case of a digital signature function, the cryptographic SPM 2030 will generate an appropriate request which will be sent to the trust engine 110 across the communication link 125.

[0294] The operations which occur between the cryptographic SPM 2030 and the trust engine 110 are the same operations that would be possible between any other system and the trust engine 110. However, these functions are effectively made available to a user system 105 through CAPI 2010 such that they appear to be locally available upon the user system 105 itself. However, unlike ordinary SPM's 2015, 2020, the functions are being carried out on the remote trust engine 110 and the results relayed to the cryptographic SPM 2030 in response to appropriate requests across the communication link 125.

[0295] This cryptographic SPM 2030 makes a number of operations available to the user system 105 or a vendor system 120 which might not otherwise be available. These functions include without limitation: encryption and decryption of documents; issuance of digital certificates; digital signing of documents; verification of digital signatures; and such other

operations as will be apparent to those of skill in the art.

[0296] In a separate embodiment, the present invention comprises a complete system for performing the data securing methods of the present invention on any data set. The computer system of this embodiment comprises a data splitting module that comprises the functionality shown in FIGURE 8 and described herein. In one embodiment of the present invention, the data splitting module, sometimes referred to herein as a secure data parser, comprises a parser program or software suite which comprises data splitting, encryption and decryption, reconstitution or reassembly functionality. This embodiment may further comprise a data storage facility or multiple data storage facilities, as well. The data splitting module, or secure data parser, comprises a cross-platform software module suite which integrates within an electronic infrastructure, or as an add-on to any application which requires the ultimate security of its data elements. This parsing process operates on any type of data set, and on any and all file types, or in a database on any row, column or cell of data in that database.

[0297] The parsing process of the present invention may, in one embodiment, be designed in a modular tiered fashion, and any encryption process is suitable for use in the process of the present invention. The modular tiers of the parsing and splitting process of the present invention may include, but are not limited to, 1) cryptographic split, dispersed and securely stored in multiple locations; 2) encrypt, cryptographically split, dispersed and securely stored in multiple

locations; 3) encrypt, cryptographically split, encrypt each share, then dispersed and securely stored in multiple locations; and 4) encrypt, cryptographically split, encrypt each share with a different type of encryption than was used in the first step, then dispersed and securely stored in multiple locations.

[0298] The process comprises, in one embodiment, splitting of the data according to the contents of a generated random number, or key and performing the same cryptographic splitting of the key used in the encryption of splitting of the data to be secured into two or more portions, or shares, of parsed and split data, and in one embodiment, preferably into four or more portions of parsed and split data, encrypting all of the portions, then scattering and storing these portions back into the database, or relocating them to any named device, fixed or removable, depending on the requestor's need for privacy and security. Alternatively, in another embodiment, encryption may occur prior to the splitting of the data set by the splitting module or secure data parser. The original data processed as described in this embodiment is encrypted and obfuscated and is secured. The dispersion of the encrypted elements, if desired, can be virtually anywhere, including, but not limited to, a single server or data storage device, or among separate data storage facilities or devices. Encryption key management in one embodiment may be included within the software suite, or in another embodiment may be integrated into an existing infrastructure or any other desired location.

[0299] A cryptographic split (cryptosplit) partitions the data into N number of shares. The

partitioning can be on any size unit of data, including an individual bit, bits, bytes, kilobytes, megabytes, or larger units, as well as any pattern or combination of data unit sizes whether predetermined or randomly generated. The units can also be of different sized, based on either a random or predetermined set of values. This means the data can be viewed as a sequence of these units. In this manner the size of the data units themselves may render the data more secure, for example by using one or more predetermined or randomly generated pattern, sequence or combination of data unit sizes. The units are then distributed (either randomly or by a predetermined set of values) into the N shares. This distribution could also involve a shuffling of the order of the units in the shares. It is readily apparent to those of ordinary skill in the art that the distribution of the data units into the shares may be performed according to a wide variety of possible selections, including but not limited to size-fixed, predetermined sizes, or one or more combination, pattern or sequence of data unit sizes that are predetermined or randomly generated.

[0300] One example of this cryptographic split process, or cryptosplit, would be to consider the data to be 23 bytes in size, with the data unit size chosen to be one byte, and with the number of shares selected to be 4. Each byte would be distributed into one of the 4 shares. Assuming a random distribution, a key would be obtained to create a sequence of 23 random numbers (r1, r2, r3 through r23), each with a value between 1 and 4 corresponding to the four shares. Each of the units of data (in this example 23 individual bytes of data) is associated with one of the 23 random

numbers corresponding to one of the four shares. The distribution of the bytes of data into the four shares would occur by placing the first byte of the data into share number r1, byte two into share r2, byte three into share r3, through the 23rd byte of data into share r23. It is readily apparent to those of ordinary skill in the art that a wide variety of other possible steps or combination or sequence of steps, including the size of the data units, may be used in the cryptosplit process of the present invention, and the above example is a non-limiting description of one process for cryptosplitting data. To recreate the original data, the reverse operation would be performed.

[0301] In another embodiment of the cryptosplit process of the present invention, an option for the cryptosplitting process is to provide sufficient redundancy in the shares such that only a subset of the shares are needed to reassemble or restore the data to its original or useable form. As a non-limiting example, the cryptosplit may be done as a "3 of 4" cryptosplit such that only three of the four shares are necessary to reassemble or restore the data to its original or useable form. This is also referred to as a "M of N cryptosplit" wherein N is the total number of shares, and M is at least one less than N. It is readily apparent to those of ordinary skill in the art that there are many possibilities for creating this redundancy in the cryptosplitting process of the present invention.

[0302] In one embodiment of the cryptosplitting process of the present invention, each unit of data is stored in two shares, the primary share and the backup share. Using the "3 of 4" cryptosplitting process

described above, any one share can be missing, and this is sufficient to reassemble or restore the original data with no missing data units since only three of the total four shares are required. As described herein, a random number is generated that corresponds to one of the shares. The random number is associated with a data unit, and stored in the corresponding share, based on a key. One key is used, in this embodiment, to generate the primary and backup share random number. As described herein for the cryptosplitting process of the present invention, a set of random numbers (also referred to as primary share numbers) from 0 to 3 are generated equal to the number of data units. Then another set of random numbers is generated (also referred to as backup share numbers) from 1 to 3 equal to the number of data units. Each unit of data is then associated with a primary share number and a backup share number. Alternatively, a set of random numbers may be generated that is fewer than the number of data units, and repeating the random number set, but this may reduce the security of the sensitive data. The primary share number is used to determine into which share the data unit is stored. The backup share number is combined with the primary share number to create a third share number between 0 and 3, and this number is used to determine into which share the data unit is stored. In this example, the equation to determine the third share number is:

(primary share number + backup share number) MOD 4 = third share number.

[0303] In the embodiment described above where the primary share number is between 0 and 3, and the backup share number is between 1 and 3 ensures that the third

share number is different from the primary share number. This results in the data unit being stored in two different shares. It is readily apparent to those of ordinary skill in the art that there are many ways of performing redundant cryptosplitting and non-redundant cryptosplitting in addition to the embodiments disclosed herein. For example, the data units in each share could be shuffled utilizing a different algorithm. This data unit shuffling may be performed as the original data is split into the data units, or after the data units are placed into the shares, or after the share is full, for example.

[0304] The various cryptosplitting processes and data shuffling processes described herein, and all other embodiments of the cryptosplitting and data shuffling methods of the present invention may be performed on data units of any size, including but not limited to, as small as an individual bit, bits, bytes, kilobytes, megabytes or larger.

[0305] An example of one embodiment of source code that would perform the cryptosplitting process described herein is:

```
DATA [1:24] - array of bytes with the data to be split
SHARES[0:3; 1:24] - 2-dimensionalarray with each row
representing one of the shares
RANDOM[1:24] - array random numbers in the range of
0..3
S1 = 1;
S2 = 1;
S3 = 1;
S4 = 1;

For J = 1 to 24 do
```

```
Begin
IF RANDOM[J[ ==0 then
     Begin
     SHARES[1,S1] = DATA [J];
     S1 = S1 + 1;
     End
ELSE IF RANDOM[J[ ==1 then
     Begin
     SHARES[2,S2] = DATA [J];
     S2 = S2 + 1;
     END
ELSE IF RANDOM[J[ ==2 then
     Begin
     Shares[3,S3] = data [J];
     S3 = S3 + 1;
     End
Else   begin
     Shares[4,S4] = data [J];
     S4 = S4 + 1;
     End;
END;
```

[0306] An example of one embodiment of source code that would perform the cryptosplitting RAID process described herein is:

[0307] Generate two sets of numbers, PrimaryShare is 0 to 3, BackupShare is 1 to 3. Then put each data unit into share[primaryshare[1]] and share[(primaryshare[1]+backupshare[1]) mod 4, with the same process as in cryptosplitting described above. This method will be scalable to any size N, where only N-1 shares are necessary to restore the data.

[0308] The retrieval, recombining, reassembly or reconstituting of the encrypted data elements may

utilize any number of authentication techniques, including, but not limited to, biometrics, such as fingerprint recognition, facial scan, hand scan, iris scan, retinal scan, ear scan, vascular pattern recognition or DNA analysis. The data splitting and/or parser modules of the present invention may be integrated into a wide variety of infrastructure products or applications as desired.

[0309] Traditional encryption technologies known in the art rely on one or more key used to encrypt the data and render it unusable without the key. The data, however, remains whole and intact and subject to attack. The secure data parser of the present invention, in one embodiment, addresses this problem by performing a cryptographic parsing and splitting of the encrypted file into two or more portions or shares, and in another embodiment, preferably four or more shares, adding another layer of encryption to each share of the data, then storing the shares in different physical and/or logical locations. When one or more data shares are physically removed from the system, either by using a removable device, such as a data storage device, or by placing the share under another party's control, any possibility of compromise of secured data is effectively removed.

[0310] An example of one embodiment of the secure data parser of the present invention and an example of how it may be utilized is shown in FIGURE 21 and described below. However, it is readily apparent to those of ordinary skill in the art that the secure data parser of the present invention may be utilized in a wide variety of ways in addition to the non-limiting example below. As a deployment option, and in one

embodiment, the secure data parser may be implemented with external session key management or secure internal storage of session keys. Upon implementation, a Parser Master Key will be generated which will be used for securing the application and for encryption purposes. It should be also noted that the incorporation of the Parser Master key in the resulting secured data allows for a flexibility of sharing of secured data by individuals within a workgroup, enterprise or extended audience.

[0311] As shown in Figure 21, this embodiment of the present invention shows the steps of the process performed by the secure data parser on data to store the session master key with the parsed data:

[0312] 1. Generating a session master key and encrypt the data using RS1 stream cipher.

[0313] 2. Separating the resulting encrypted data into four shares or portions of parsed data according to the pattern of the session master key.

[0314] 3. In this embodiment of the method, the session master key will be stored along with the secured data shares in a data depository. Separating the session master key according to the pattern of the Parser Master Key and append the key data to the encrypted parsed data.

[0315] 4. The resulting four shares of data will contain encrypted portions of the original data and portions of the session master key. Generate a stream cipher key for each of the four data shares.

[0316] 5. Encrypting each share, then store the encryption keys in different locations from the encrypted data portions or shares: Share 1 gets Key 4,

Share 2 gets Key 1, Share 3 gets Key 2, Share 4 gets Key 3.

[0317] To restore the original data format, the steps are reversed.

[0318] It is readily apparent to those of ordinary skill in the art that certain steps of the methods described herein may be performed in different order, or repeated multiple times, as desired. It is also readily apparent to those skilled in the art that the portions of the data may be handled differently from one another. For example, multiple parsing steps may be performed on only one portion of the parsed data. Each portion of parsed data may be uniquely secured in any desirable way provided only that the data may be reassembled, reconstituted, reformed, decrypted or restored to its original or other usable form.

[0319] As shown in FIGURE 22 and described herein, another embodiment of the present invention comprises the steps of the process performed by the secure data parser on data to store the session master key data in one or more separate key management table:

[0320] 1. Generating a session master key and encrypt the data using RS1 stream cipher.

[0321] 2. Separating the resulting encrypted data into four shares or portions of parsed data according to the pattern of the session master key.

[0322] 3. In this embodiment of the method of the present invention, the session master key will be stored in a separate key management table in a data depository. Generating a unique transaction ID for this transaction. Storing the transaction ID and session master key in a separate key management table. Separating the transaction ID according to the pattern

of the Parser Master Key and append the data to the encrypted parsed or separated data.

[0323] 4. The resulting four shares of data will contain encrypted portions of the original data and portions of the transaction ID.

[0324] 5. Generating a stream cipher key for each of the four data shares.

[0325] 6. Encrypting each share, then store the encryption keys in different locations from the encrypted data portions or shares: Share 1 gets Key 4, Share 2 gets Key 1, Share 3 gets Key 2, Share 4 gets Key 3.

[0326] To restore the original data format, the steps are reversed.

[0327] It is readily apparent to those of ordinary skill in the art that certain steps of the method described herein may be performed in different order, or repeated multiple times, as desired. It is also readily apparent to those skilled in the art that the portions of the data may be handled differently from one another. For example, multiple separating or parsing steps may be performed on only one portion of the parsed data. Each portion of parsed data may be uniquely secured in any desirable way provided only that the data may be reassembled, reconstituted, reformed, decrypted or restored to its original or other usable form.

[0328] As shown in Figure 23, this embodiment of the present invention shows the steps of the process performed by the secure data parser on data to store the session master key with the parsed data:

[0329] 1. Accessing the parser master key associated with the authenticated user

[0330] 2. Generating a unique Session Master key

[0331] 3. Derive an Intermediary Key from an exclusive OR function of the Parser Master Key and Session Master key

[0332] 4. Optional encryption of the data using an existing or new encryption algorithm keyed with the Intermediary Key.

[0333] 5. Separating the resulting optionally encrypted data into four shares or portions of parsed data according to the pattern of the Intermediary key.

[0334] 6. In this embodiment of the method, the session master key will be stored along with the secured data shares in a data depository. Separating the session master key according to the pattern of the Parser Master Key and append the key data to the optionally encrypted parsed data shares.

[0335] 7. The resulting multiple shares of data will contain optionally encrypted portions of the original data and portions of the session master key.

[0336] 8. Optionally generate an encryption key for each of the four data shares.

[0337] 9. Optionally encrypting each share with an existing or new encryption algorithm, then store the encryption keys in different locations from the encrypted data portions or shares: for example, Share 1 gets Key 4, Share 2 gets Key 1, Share 3 gets Key 2, Share 4 gets Key 3.

[0338] To restore the original data format, the steps are reversed.

[0339] It is readily apparent to those of ordinary skill in the art that certain steps of the methods described herein may be performed in different order, or repeated multiple times, as desired. It is also

readily apparent to those skilled in the art that the portions of the data may be handled differently from one another. For example, multiple parsing steps may be performed on only one portion of the parsed data. Each portion of parsed data may be uniquely secured in any desirable way provided only that the data may be reassembled, reconstituted, reformed, decrypted or restored to its original or other usable form.

[0340] As shown in FIGURE 24 and described herein, another embodiment of the present invention comprises the steps of the process performed by the secure data parser on data to store the session master key data in one or more separate key management table:

[0341] 1. Accessing the Parser Master Key associated with the authenticated user

[0342] 2. Generating a unique Session Master Key

[0343] 3. Derive an Intermediary Key from an exclusive OR function of the Parser Master Key and Session Master key

[0344] 4. Optionally encrypt the data using an existing or new encryption algorithm keyed with the Intermediary Key.

[0345] 5. Separating the resulting optionally encrypted data into four shares or portions of parsed data according to the pattern of the Intermediary Key.

[0346] 6. In this embodiment of the method of the present invention, the session master key will be stored in a separate key management table in a data depository. Generating a unique transaction ID for this transaction. Storing the transaction ID and session master key in a separate key management table or passing the Session Master Key and transaction ID back to the calling program for external management.

Separating the transaction ID according to the pattern of the Parser Master Key and append the data to the optionally encrypted parsed or separated data.

[0347] 7. The resulting four shares of data will contain optionally encrypted portions of the original data and portions of the transaction ID.

[0348] 8. Optionally generate an encryption key for each of the four data shares.

[0349] 9. Optionally encrypting each share, then store the encryption keys in different locations from the encrypted data portions or shares. For example: Share 1 gets Key 4, Share 2 gets Key 1, Share 3 gets Key 2, Share 4 gets Key 3.

[0350] To restore the original data format, the steps are reversed.

[0351] It is readily apparent to those of ordinary skill in the art that certain steps of the method described herein may be performed in different order, or repeated multiple times, as desired. It is also readily apparent to those skilled in the art that the portions of the data may be handled differently from one another. For example, multiple separating or parsing steps may be performed on only one portion of the parsed data. Each portion of parsed data may be uniquely secured in any desirable way provided only that the data may be reassembled, reconstituted, reformed, decrypted or restored to its original or other usable form.

[0352] A wide variety of encryption methodologies are suitable for use in the methods of the present invention, as is readily apparent to those skilled in the art. The One Time Pad algorithm, is often considered one of the most secure encryption methods,

and is suitable for use in the method of the present invention. Using the One Time Pad algorithm requires that a key be generated which is as long as the data to be secured. The use of this method may be less desirable in certain circumstances such as those resulting in the generation and management of very long keys because of the size of the data set to be secured. In the One-Time Pad (OTP) algorithm, the simple exclusive-or function, XOR, is used. For two binary streams x and y of the same length, x XOR y means the bitwise exclusive-or of x and y.

[0353] At the bit level is generated:

0 XOR 0 = 0

0 XOR 1 = 1

1 XOR 0 = 1

1 XOR 1 = 0

[0354] An example of this process is described herein for an n-byte secret, s, (or data set) to be split. The process will generate an n-byte random value, a, and then set:

b = a XOR s.

[0355] Note that one can derive "s" via the equation:

s = a XOR b.

[0356] The values a and b are referred to as shares or portions and are placed in separate depositories. Once the secret s is split into two or more shares, it is discarded in a secure manner.

[0357] The secure data parser of the present invention may utilize this function, performing multiple XOR functions incorporating multiple distinct secret key values: K1, K2, K3, Kn, K5. At the beginning of the operation, the data to be secured is

passed through the first encryption operation, secure data = data XOR secret key 5:

S = D XOR K5

[0358] In order to securely store the resulting encrypted data in, for example, four shares, S1, S2, S3, Sn, the data is parsed and split into "n" segments, or shares, according to the value of K5. This operation results in "n" pseudorandom shares of the original encrypted data. Subsequent XOR functions may then be performed on each share with the remaining secret key values, for example: Secure data segment 1 = encrypted data share 1 XOR secret key 1:

SD1 = S1 XOR K1

SD2 = S2 XOR K2

SD3 = S3 XOR K3

SDn = Sn XOR Kn.

[0359] In one embodiment, it may not be desired to have any one depository contain enough information to decrypt the information held there, so the key required to decrypt the share is stored in a different data depository:

Depository 1: SD1, Kn

Depository 2: SD2, K1

Depository 3: SD3, K2

Depository n: SDn, K3.

[0360] Additionally, appended to each share may be the information required to retrieve the original session encryption key, K5. Therefore, in the key management example described herein, the original session master key is referenced by a transaction ID split into "n" shares according to the contents of the installation dependant Parser Master Key (TID1, TID2, TID3, TIDn):

Depository 1: SD1, Kn, TID1
Depository 2: SD2, K1, TID2
Depository 3: SD3, K2, TID3
Depository n: SDn, K3, TIDn.

[0361] In the incorporated session key example described herein, the session master key is split into "n" shares according to the contents of the installation dependant Parser Master Key (SK1, SK2, SK3, SKn):

Depository 1: SD1, Kn, SK1
Depository 2: SD2, K1, SK2
Depository 3: SD3, K2, SK3
Depository n: SDn, K3, SKn.

[0362] Unless all four shares are retrieved, the data cannot be reassembled according to this example. Even if all four shares are captured, there is no possibility of reassembling or restoring the original information without access to the session master key and the Parser Master Key.

[0363] This example has described an embodiment of the method of the present invention, and also describes, in another embodiment, the algorithm used to place shares into depositories so that shares from all depositories can be combined to form the secret authentication material. The computations needed are very simple and fast. However, with the One Time Pad (OTP) algorithm there may be circumstances that cause it to be less desirable, such as a large data set to be secured, because the key size is the same size as the data to be stored. Therefore, there would be a need to store and transmit about twice the amount of the original data which may be less desirable under certain circumstances.

Stream Cipher RS1

[0364] The stream cipher RS1 splitting technique is very similar to the OTP splitting technique described herein. Instead of an n-byte random value, an n' = min(n, 16)-byte random value is generated and used to key the RS1 Stream Cipher algorithm. The advantage of the RS1 Stream Cipher algorithm is that a pseudorandom key is generated from a much smaller seed number. The speed of execution of the RS1 Stream Cipher encryption is also rated at approximately 10 times the speed of the well known in the art Triple DES encryption without compromising security. The RS1 Stream Cipher algorithm is well known in the art, and may be used to generate the keys used in the XOR function. The RS1 Stream Cipher algorithm is interoperable with other commercially available stream cipher algorithms, such as the RC4™ stream cipher algorithm of RSA Security, Inc and is suitable for use in the methods of the present invention.

[0365] Using the key notation above, K1 thru K5 are now an n' byte random values and we set:

SD1 = S1 XOR E(K1)

SD2 = S2 XOR E(K2)

SD3 = S3 XOR E(K3)

SDn = Sn XOR E(Kn)

where E(K1) thru E(Kn) are the first n' bytes of output from the RS1 Stream Cipher algorithm keyed by K1 thru Kn. The shares are now placed into data depositories as described herein.

[0366] In this stream cipher RS1 algorithm, the required computations needed are nearly as simple and fast as the OTP algorithm. The benefit in this example

using the RS1 Stream Cipher is that the system needs to store and transmit on average only about 16 bytes more than the size of the original data to be secured per share. When the size of the original data is more than 16 bytes, this RS1 algorithm is more efficient than the OTP algorithm because it is simply shorter. It is readily apparent to those of ordinary skill in the art that a wide variety of encryption methods or algorithms are suitable for use in the present invention, including, but not limited to RS1, OTP, RC4™, Triple DES and AES.

[0367] There are major advantages provided by the data security methods and computer systems of the present invention over traditional encryption methods. One advantage is the security gained from moving shares of the data to different locations on one or more data depositories or storage devices, that may be in different logical, physical or geographical locations. When the shares of data are split physically and under the control of different personnel, for example, the possibility of compromising the data is greatly reduced.

[0368] Another advantage provided by the methods and system of the present invention is the combination of the steps of the method of the present invention for securing data to provide a comprehensive process of maintaining security of sensitive data. The data is encrypted with a secure key and split into one or more shares, and in one embodiment, four shares, according to the secure key. The secure key is stored safely with a reference pointer which is secured into four shares according to a secure key. The data shares are then encrypted individually and the keys are stored

safely with different encrypted shares. When combined, the entire process for securing data according to the methods disclosed herein becomes a comprehensive package for data security.

[0369] The data secured according to the methods of the present invention is readily retrievable and restored, reconstituted, reassembled, decrypted, or otherwise returned into its original or other suitable form for use. In order to restore the original data, the following items may be utilized:

[0370] 1. All shares or portions of the data set.

[0371] 2. Knowledge of and ability to reproduce the process flow of the method used to secure the data.

[0372] 3. Access to the session master key.

[0373] 4. Access to the Parser Master Key.

[0374] Therefore, it may be desirable to plan a secure installation wherein at least one of the above elements may be physically separated from the remaining components of the system (under the control of a different system administrator for example).

[0375] Protection against a rogue application invoking the data securing methods application may be enforced by use of the Parser Master Key. A mutual authentication handshake between the secure data parser and the application may be required in this embodiment of the present invention prior to any action taken.

[0376] The security of the system dictates that there be no "backdoor" method for recreation of the original data. For installations where data recovery issues may arise, the secure data parser can be enhanced to provide a mirror of the four shares and session master key depository. Hardware options such as RAID (redundant array of inexpensive disks, used to

spread information over several disks) and software options such as replication can assist as well in the data recovery planning.

Key Management

[0377] In one embodiment of the present invention, the data securing method uses three sets of keys for an encryption operation. Each set of keys may have individual key storage, retrieval, security and recovery options, based on the installation. The keys that may be used, include, but are not limited to:

The Parser Master Key

[0378] This key is an individual key associated with the installation of the secure data parser. It is installed on the server on which the secure data parser has been deployed. There are a variety of options suitable for securing this key including, but not limited to, a smart card, separate hardware key store, standard key stores, custom key stores or within a secured database table, for example.

The Session Master Key

[0379] A Session Master Key may be generated each time data is secured. The Session Master Key is used to encrypt the data prior to the parsing and splitting operations. It may also be incorporated (if the Session Master Key is not integrated into the parsed data) as a means of parsing the encrypted data. The Session Master Key may be secured in a variety of manners, including, but not limited to, a standard key store, custom key store, separate database table, or secured within the encrypted shares, for example.

The Share Encryption Keys

[0380] For each share or portions of a data set that is created, an individual Share Encryption Key may be generated to further encrypt the shares. The Share Encryption Keys may be stored in different shares than the share that was encrypted.

[0381] It is readily apparent to those of ordinary skill in the art that the data securing methods and computer system of the present invention are widely applicable to any type of data in any setting or environment. In addition to commercial applications conducted over the Internet or between customers and vendors, the data securing methods and computer systems of the present invention are highly applicable to non-commercial or private settings or environments. Any data set that is desired to be kept secure from any unauthorized user may be secured using the methods and systems described herein. For example, access to a particular database within a company or organization may be advantageously restricted to only selected users by employing the methods and systems of the present invention for securing data. Another example is the generation, modification or access to documents wherein it is desired to restrict access or prevent unauthorized or accidental access or disclosure outside a group of selected individuals, computers or workstations. These and other examples of the ways in which the methods and systems of data securing of the present invention are applicable to any non-commercial or commercial environment or setting for any setting, including, but not limited to any organization, government agency or corporation.

[0382] In another embodiment of the present invention, the data securing method uses three sets of

keys for an encryption operation. Each set of keys may have individual key storage, retrieval, security and recovery options, based on the installation. The keys that may be used, include, but are not limited to:

1. The Parser Master Key

[0383] This key is an individual key associated with the installation of the secure data parser. It is installed on the server on which the secure data parser has been deployed. There are a variety of options suitable for securing this key including, but not limited to, a smart card, separate hardware key store, standard key stores, custom key stores or within a secured database table, for example.

2. The Session Master Key

[0384] A Session Master Key may be generated each time data is secured. The Session Master Key is used in conjunction with the Parser Master key to derive the Intermediary Key. The Session Master Key may be secured in a variety of manners, including, but not limited to, a standard key store, custom key store, separate database table, or secured within the encrypted shares, for example.

3. The Intermediary Key

[0385] An Intermediary Key may be generated each time data is secured. The Intermediary Key is used to encrypt the data prior to the parsing and splitting operation. It may also be incorporated as a means of parsing the encrypted data.

4. The Share Encryption Keys

[0386] For each share or portions of a data set that is created, an individual Share Encryption Key may be generated to further encrypt the shares. The Share

Encryption Keys may be stored in different shares than the share that was encrypted.

[0387] It is readily apparent to those of ordinary skill in the art that the data securing methods and computer system of the present invention are widely applicable to any type of data in any setting or environment. In addition to commercial applications conducted over the Internet or between customers and vendors, the data securing methods and computer systems of the present invention are highly applicable to non-commercial or private settings or environments. Any data set that is desired to be kept secure from any unauthorized user may be secured using the methods and systems described herein. For example, access to a particular database within a company or organization may be advantageously restricted to only selected users by employing the methods and systems of the present invention for securing data. Another example is the generation, modification or access to documents wherein it is desired to restrict access or prevent unauthorized or accidental access or disclosure outside a group of selected individuals, computers or workstations. These and other examples of the ways in which the methods and systems of data securing of the present invention are applicable to any non-commercial or commercial environment or setting for any setting, including, but not limited to any organization, government agency or corporation.

Workgroup, Project, Individual PC/Laptop or Cross Platform Data Security

[0388] The data securing methods and computer systems of the present invention are also useful in securing data by workgroup, project, individual

PC/Laptop and any other platform that is in use in, for example, businesses, offices, government agencies, or any setting in which sensitive data is created, handled or stored. The present invention provides methods and computer systems to secure data that is known to be sought after by organizations, such as the U.S. Government, for implementation across the entire government organization or between governments at a state or federal level.

[0389] The data securing methods and computer systems of the present invention provide the ability to not only parse and split flat files but also data fields, sets and or table of any type. Additionally, all forms of data are capable of being secured under this process, including, but not limited to, text, video, images, biometrics and voice data. Scalability, speed and data throughput of the methods of securing data of the present invention are only limited to the hardware the user has at their disposal.

[0390] In one embodiment of the present invention, the data securing methods are utilized as described below in a workgroup environment. In one embodiment, as shown in FIGURE 23 and described below, the Workgroup Scale data securing method of the present invention uses the private key management functionality of the TrustEngine to store the user/group relationships and the associated private keys (Parser Group Master Keys) necessary for a group of users to share secure data. The method of the present invention has the capability to secure data for an enterprise, workgroup, or individual user, depending on how the Parser Master Key was deployed.

[0391] In one embodiment, additional key management and user/group management programs may be provided, enabling wide scale workgroup implementation with a single point of administration and key management. Key generation, management and revocation are handled by the single maintenance program, which all become especially important as the number of users increase. In another embodiment, key management may also be set up across one or several different system administrators, which may not allow any one person or group to control data as needed. This allows for the management of secured data to be obtained by roles, responsibilities, membership, rights, etc., as defined by an organization, and the access to secured data can be limited to just those who are permitted or required to have access only to the portion they are working on, while others, such as managers or executives, may have access to all of the secured data. This embodiment allows for the sharing of secured data among different groups within a company or organization while at the same time only allowing certain selected individuals, such as those with the authorized and predetermined roles and responsibilities, to observe the data as a whole. In addition, this embodiment of the methods and systems of the present invention also allows for the sharing of data among, for example, separate companies, or separate departments or divisions of companies, or any separate organization departments, groups, agencies, or offices, or the like, of any government or organization or any kind, where some sharing is required, but not any one party may be permitted to have access to all the data. Particularly apparent examples of the need and utility for such a method and

system of the present invention are to allow sharing, but maintain security, in between government areas, agencies and offices, and between different divisions, departments or offices of a large company, or any other organization, for example.

[0392] An example of the applicability of the methods of the present invention on a smaller scale is as follows. A Parser Master key is used as a serialization or branding of the secure data parser to an organization. As the scale of use of the Parser Master key is reduced from the whole enterprise to a smaller workgroup, the data securing methods described herein are used to share files within groups of users.

[0393] In the example shown in FIGURE 25 and described below, there are six users defined along with their title or role within the organization. The side bar represents five possible groups that the users can belong to according to their role. The arrow represents membership by the user in one or more of the groups.

[0394] When configuring the secure data parser for use in this example, the system administrator accesses the user and group information from the operating system by a maintenance program. This maintenance program generates and assigns Parser Group Master Keys to users based on their membership in groups.

[0395] In this example, there are three members in the Senior Staff group. For this group, the actions would be:

[0396] 1. Access Parser Group Master Key for the Senior Staff group (generate a key if not available);

[0397] 2. Generate a digital certificate associating CEO with the Senior Staff group;

[0398] 3. Generate a digital certificate associating CFO with the Senior Staff group;

[0399] 4. Generate a digital certificate associating Vice President, Marketing with the Senior Staff group.

[0400] The same set of actions would be done for each group, and each member within each group. When the maintenance program is complete, the Parser Group Master Key becomes a shared credential for each member of the group. Revocation of the assigned digital certificate may be done automatically when a user is removed from a group through the maintenance program without affecting the remaining members of the group.

[0401] Once the shared credentials have been defined, the parsing and splitting process remains the same. When a file, document or data element is to be secured, the user is prompted for the target group to be used when securing the data. The resulting secured data is only accessible by other members of the target group. This functionality of the methods and systems of the present invention may be used with any other computer system or software platform, any may be, for example, integrated into existing application programs or used standalone for file security.

[0402] It is readily apparent to those of ordinary skill in the art that any one or combination of encryption algorithms are suitable for use in the methods and systems of the present invention. For example, the encryption steps may, in one embodiment, be repeated to produce a multi-layered encryption scheme. In addition, a different encryption algorithm, or combination of encryption algorithms, may be used in repeat encryption steps such that different encryption

algorithms are applied to the different layers of the multi-layered encryption scheme. As such, the encryption scheme itself may become a component of the methods of the present invention for securing sensitive data from unauthorized use or access.

[0403] The secure data parser may include as an internal component, as an external component, or as both an error-checking component. For example, in one suitable approach, as portions of data are created using the secure data parser in accordance with the present invention, to assure the integrity of the data within a portion, a hash value is taken at preset intervals within the portion and is appended to the end of the interval. The hash value is a predictable and reproducible numeric representation of the data. If any bit within the data changes, the hash value would be different. A scanning module (either as a stand-alone component external to the secure data parser or as an internal component) may then scan the portions of data generated by the secure data parser. Each portion of data (or alternatively, less than all portions of data according to some interval or by a random or pseudo-random sampling) is compared to the appended hash value or values and an action may be taken. This action may include a report of values that match and do not match, an alert for values that do not match, or invoking of some external or internal program to trigger a recovery of the data. For example, recovery of the data could be performed by invoking a recovery module based on the concept that fewer than all portions may be needed to generate original data in accordance with the present invention.

[0404] Any other suitable integrity checking may be implemented using any suitable integrity information appended anywhere in all or a subset of data portions. Integrity information may include any suitable information that can be used to determine the integrity of data portions. Examples of integrity information may include hash values computed based on any suitable parameter (e.g., based on respective data portions), digital signature information, message authentication code (MAC) information, any other suitable information, or any combination thereof.

[0405] The secure data parser of the present invention may be used in any suitable application. Namely, the secure data parser described herein has a variety of applications in different areas of computing and technology. Several such areas are discussed below. It will be understood that these are merely illustrative in nature and that any other suitable applications may make use of the secure data parser. It will further be understood that the examples described are merely illustrative embodiments that may be modified in any suitable way in order to satisfy any suitable desires. For example, parsing and splitting may be based on any suitable units, such as by bits, by bytes, by kilobytes, by megabytes, by any combination thereof, or by any other suitable unit.

[0406] The secure data parser of the present invention may be used to implement secure physical tokens, whereby data stored in a physical token may be required in order to access additional data stored in another storage area. In one suitable approach, a physical token, such as a compact USB flash drive, a floppy disk, an optical disk, a smart card, or any

other suitable physical token, may be used to store one of at least two portions of parsed data in accordance with the present invention. In order to access the original data, the USB flash drive would need to be accessed. Thus, a personal computer holding one portion of parsed data would need to have the USB flash drive, having the other portion of parsed data, attached before the original data can be accessed. FIGURE 26 illustrates this application. Storage area 2500 includes a portion of parsed data 2502. Physical token 2504, having a portion of parsed data 2506 would need to be coupled to storage area 2500 using any suitable communications interface 2508 (e.g., USB, serial, parallel, Bluetooth, IR, IEEE 1394, Ethernet, or any other suitable communications interface) in order to access the original data. This is useful in a situation where, for example, sensitive data on a computer is left alone and subject to unauthorized access attempts. By removing the physical token (e.g., the USB flash drive), the sensitive data is inaccessible. It will be understood that any other suitable approach for using physical tokens may be used.

[0407] The secure data parser of the present invention may be used to implement a secure authentication system whereby user enrollment data (e.g., passwords, private encryption keys, fingerprint templates, biometric data or any other suitable user enrollment data) is parsed and split using the secure data parser. The user enrollment data may be parsed and split whereby one or more portions are stored on a smart card, a government Common Access Card, any suitable physical storage device (e.g., magnetic or

optical disk, USB key drive, etc.), or any other suitable device. One or more other portions of the parsed user enrollment data may be stored in the system performing the authentication. This provides an added level of security to the authentication process (e.g., in addition to the biometric authentication information obtained from the biometric source, the user enrollment data must also be obtained via the appropriate parsed and split data portion).

[0408] The secure data parser of the present invention may be integrated into any suitable existing system in order to provide the use of its functionality in each system's respective environment. FIGURE 27 shows a block diagram of an illustrative system 2600, which may include software, hardware, or both for implementing any suitable application. System 2600 may be an existing system in which secure data parser 2602 may be retrofitted as an integrated component. Alternatively, secure data parser 2602 may be integrated into any suitable system 2600 from, for example, its earliest design stage. Secure data parser 2600 may be integrated at any suitable level of system 2600. For example, secure data parser 2602 may be integrated into system 2600 at a sufficiently back-end level such that the presence of secure data parser 2602 may be substantially transparent to an end user of system 2600. Secure data parser 2602 may be used for parsing and splitting data among one or more storage devices 2604 in accordance with the present invention. Some illustrative examples of systems having the secure data parser integrated therein are discussed below.

[0409] The secure data parser of the present invention may be integrated into an operating system

kernel (e.g., Linux, Unix, or any other suitable commercial or proprietary operating system). This integration may be used to protect data at the device level whereby, for example, data that would ordinarily be stored in one or more devices is separated into a certain number of portions by the secure data parser integrated into the operating system and stored among the one or more devices. When original data is attempted to be accessed, the appropriate software, also integrated into the operating system, may recombine the parsed data portions into the original data in a way that may be transparent to the end user.

[0410] The secure data parser of the present invention may be integrated into a volume manager or any other suitable component of a storage system to protect local and networked data storage across any or all supported platforms. For example, with the secure data parser integrated, a storage system may make use of the redundancy offered by the secure data parser (i.e., which is used to implement the feature of needing fewer than all separated portions of data in order to reconstruct the original data) to protect against data loss. The secure data parser also allows all data written to storage devices, whether using redundancy or not, to be in the form of multiple portions that are generated according to the parsing of the present invention. When original data is attempted to be accessed, the appropriate software, also integrated into the volume manager or other suitable component of the storage system, may recombine the parsed data portions into the original data in a way that may be transparent to the end user.

[0411] In one suitable approach, the secure data parser of the present invention may be integrated into a RAID controller (as either hardware or software). This allows for the secure storage of data to multiple drives while maintaining fault tolerance in case of drive failure.

[0412] The secure data parser of the present invention may be integrated into a database in order to, for example, protect sensitive table information. For example, in one suitable approach, data associated with particular cells of a database table (e.g., individual cells, one or more particular columns, one or more particular rows, any combination thereof, or an entire database table) may be parsed and separated according to the present invention (e.g., where the different portions are stored on one or more storage devices at one or more locations or on a single storage device). Access to recombine the portions in order to view the original data may be granted by traditional authentication methods (e.g., username and password query).

[0413] The secure parser of the present invention may be integrated in any suitable system that involves data in motion (i.e., transfer of data from one location to another). Such systems include, for example, email, streaming data broadcasts, and wireless (e.g., WiFi) communications. With respect to email, in one suitable approach, the secure parser may be used to parse outgoing messages (i.e., containing text, binary data, or both (e.g., files attached to an email message)) and sending the different portions of the parsed data along different paths thus creating multiple streams of data. If any one of these streams

of data is compromised, the original message remains secure because the system may require that more than one of the portions be combined, in accordance with the present invention, in order to generate the original data. In another suitable approach, the different portions of data may be communicated along one path sequentially so that if one portion is obtained, it may not be sufficient to generate the original data. The different portions arrive at the intended recipient's location and may be combined to generate the original data in accordance with the present invention.

[0414] FIGURES 28 and 29 are illustrative block diagrams of such email systems. FIGURE 28 shows a sender system 2700, which may include any suitable hardware, such as a computer terminal, personal computer, handheld device (e.g., PDA, Blackberry), cellular telephone, computer network, any other suitable hardware, or any combination thereof. Sender system 2700 is used to generate and/or store a message 2704, which may be, for example, an email message, a binary data file (e.g., graphics, voice, video, etc.), or both. Message 2704 is parsed and split by secure data parser 2702 in accordance with the present invention. The resultant data portions may be communicated across one or more separate communications paths 2706 over network 2708 (e.g., the Internet, an intranet, a LAN, WiFi, Bluetooth, any other suitable hard-wired or wireless communications means, or any combination thereof) to recipient system 2710. The data portions may be communicated parallel in time or alternatively, according to any suitable time delay between the communication of the different data portions. Recipient system 2710 may be any suitable

hardware as described above with respect to sender system 2700. The separate data portions carried along communications paths 2706 are recombined at recipient system 2710 to generate the original message or data in accordance with the present invention.

[0415] FIGURE 29 shows a sender system 2800, which may include any suitable hardware, such as a computer terminal, personal computer, handheld device (e.g., PDA), cellular telephone, computer network, any other suitable hardware, or any combination thereof. Sender system 2800 is used to generate and/or store a message 2804, which may be, for example, an email message, a binary data file (e.g., graphics, voice, video, etc.), or both. Message 2804 is parsed and split by secure data parser 2802 in accordance with the present invention. The resultant data portions may be communicated across a single communications paths 2806 over network 2808 (e.g., the Internet, an intranet, a LAN, WiFi, Bluetooth, any other suitable communications means, or any combination thereof) to recipient system 2810. The data portions may be communicated serially across communications path 2806 with respect to one another. Recipient system 2810 may be any suitable hardware as described above with respect to sender system 2800. The separate data portions carried along communications path 2806 are recombined at recipient system 2810 to generate the original message or data in accordance with the present invention.

[0416] It will be understood that the arrangement of FIGURES 28 and 29 are merely illustrative. Any other suitable arrangement may be used. For example, in another suitable approach, the features of the systems of FIGURES 28 and 29 may be combined whereby the multi-

path approach of FIGURE 28 is used and in which one or more of communications paths 2706 are used to carry more than one portion of data as communications path 2806 does in the context of FIGURE 29.

[0417] The secure data parser may be integrated at any suitable level of a data-in motion system. For example, in the context of an email system, the secure parser may be integrated at the user-interface level (e.g., into Microsoft® Outlook), in which case the user may have control over the use of the secure data parser features when using email. Alternatively, the secure parser may be implemented in a back-end component such as at the exchange server, in which case messages may be automatically parsed, split, and communicated along different paths in accordance with the present invention without any user intervention.

[0418] Similarly, in the case of streaming broadcasts of data (e.g., audio, video), the outgoing data may be parsed and separated into multiple streams each containing a portion of the parsed data. The multiple streams may be transmitted along one or more paths and recombined at the recipient's location in accordance with the present invention. One of the benefits of this approach is that it avoids the relatively large overhead associated with traditional encryption of data followed by transmission of the encrypted data over a single communications channel. The secure parser of the present invention allows data in motion to be sent in multiple parallel streams, increasing speed and efficiency.

[0419] It will be understand that the secure data parser may be integrated for protection of and fault tolerance of any type of data in motion through any

transport medium, including, for example, wired, wireless, or physical. For example, voice over Internet protocol (VoIP) applications may make use of the secure data parser of the present invention. Wireless or wired data transport from or to any suitable personal digital assistant (PDA) devices such as Blackberries and SmartPhones may be secured using the secure data parser of the present invention. Communications using wireless 802.11 protocols for peer to peer and hub based wireless networks, satellite communications, point to point wireless communications, Internet client/server communications, or any other suitable communications may involve the data in motion capabilities of the secure data parser in accordance with the present invention. Data communication between computer peripheral device (e.g., printer, scanner, monitor, keyboard, network router, biometric authentication device (e.g., fingerprint scanner), or any other suitable peripheral device) between a computer and a computer peripheral device, between a computer peripheral device and any other suitable device, or any combination thereof may make use of the data in motion features of the present invention.

[0420] The data in motion features of the present invention may also apply to physical transportation of secure shares using for example, separate routes, vehicles, methods, any other suitable physical transportation, or any combination thereof. For example, physical transportation of data may take place on digital/magnetic tapes, floppy disks, optical disks, physical tokens, USB drives, removable hard drives, consumer electronic devices with flash memory (e.g., Apple IPODs or other MP3 players), flash memory, any

other suitable medium used for transporting data, or any combination thereof.

[0421] The secure data parser of the present invention may provide security with the ability for disaster recovery. According to the present invention, fewer than all portions of the separated data generated by the secure data parser may be necessary in order to retrieve the original data. That is, out of m portions stored, n may be the minimum number of these m portions necessary to retrieve the original data, where n <= m. For example, if each of four portions is stored in a different physical location relative to the other three portions, then, if n=2 in this example, two of the locations may be compromised whereby data is destroyed or inaccessible, and the original data may still be retrieved from the portions in the other two locations. Any suitable value for n or m may be used.

[0422] In addition, the n of m feature of the present invention may be used to create a "two man rule" whereby in order to avoid entrusting a single individual or any other entity with full access to what may be sensitive data, two or more distinct entities, each with a portion of the separated data parsed by the secure parser of the present invention may need to agree to put their portions together in order to retrieve the original data.

[0423] The secure data parser of the present invention may be used to provide a group of entities with a group-wide key that allows the group members to access particular information authorized to be accessed by that particular group. The group key may be one of the data portions generated by the secure parser in accordance with the present invention that may be

required to be combined with another portion centrally stored, for example in order to retrieve the information sought. This feature allows for, for example, secure collaboration among a group. It may be applied in for example, dedicated networks, virtual private networks, intranets, or any other suitable network.

[0424] Specific applications of this use of the secure parser include, for example, coalition information sharing in which, for example, multi-national friendly government forces are given the capability to communicate operational and otherwise sensitive data on a security level authorized to each respective country over a single network or a dual network (i.e., as compared to the many networks involving relatively substantial manual processes currently used). This capability is also applicable for companies or other organizations in which information needed to be known by one or more specific individuals (within the organization or without) may be communicated over a single network without the need to worry about unauthorized individuals viewing the information.

[0425] Another specific application includes a multi-level security hierarchy for government systems. That is, the secure parser of the present invention may provide for the ability to operate a government system at different levels of classified information (e.g., unclassified, classified, secret, top secret) using a single network. If desired, more networks may be used (e.g., a separate network for top secret), but the present invention allows for substantially fewer than

current arrangement in which a separate network is used for each level of classification.

[0426] It will be understood that any combination of the above described applications of the secure parser of the present invention may be used. For example, the group key application can be used together with the data in motion security application (i.e., whereby data that is communicated over a network can only be accessed by a member of the respective group and where, while the data is in motion, it is split among multiple paths (or sent in sequential portions) in accordance with the present invention).

[0427] The secure data parser of the present invention may be integrated into any middleware application to enable applications to securely store data to different database products or to different devices without modification to either the applications or the database. Middleware is a general term for any product that allows two separate and already existing programs to communicate. For example, in one suitable approach, middleware having the secure data parser integrated, may be used to allow programs written for a particular database to communicate with other databases without custom coding.

[0428] The secure data parser of the present invention may be implemented having any combination of any suitable capabilities, such as those discussed herein. In some embodiments of the present invention, for example, the secure data parser may be implemented having only certain capabilities whereas other capabilities may be obtained through the use of external software, hardware, or both interfaced either directly or indirectly with the secure data parser.

[0429] FIGURE 30, for example, shows an illustrative implementation of the secure data parser as secure data parser 3000. Secure data parser 3000 may be implemented with very few built-in capabilities. As illustrated, secure data parser 3000 may include built-in capabilities for parsing and splitting data into portions (also referred to herein as shares) of data using module 3002 in accordance with the present invention. Secure data parser 3000 may also include built in capabilities for performing redundancy in order to be able to implement, for example, the m of n feature described above (i.e., recreating the original data using fewer than all shares of parsed and split data) using module 3004. Secure data parser 3000 may also include share distribution capabilities using module 3006 for placing the shares of data into buffers from which they are sent for communication to a remote location, for storage, etc. in accordance with the present invention. It will be understood that any other suitable capabilities may be built into secure data parser 3000.

[0430] Assembled data buffer 3008 may be any suitable memory used to store the original data (although not necessarily in its original form) that will be parsed and split by secure data parser 3000. In a splitting operation, assembled data buffer 3008 provides input to secure data parser 3008. In a restore operation, assembled data buffer 3008 may be used to store the output of secure data parser 3000.

[0431] Split shares buffers 3010 may be one or more memory modules that may be used to store the multiple shares of data that resulted from the parsing and splitting of original data. In a splitting operation,

split shares buffers 3010 hold the output of the secure data parser. In a restore operation, split shares buffers hold the input to secure data parser 3000.

[0432] It will be understood that any other suitable arrangement of capabilities may be built-in for secure data parser 3000. Any additional features may be built-in and any of the features illustrated may be removed, made more robust, made less robust, or may otherwise be modified in any suitable way. Buffers 3008 and 3010 are likewise merely illustrative and may be modified, removed, or added to in any suitable way.

[0433] Any suitable modules implemented in software, hardware or both may be called by or may call to secure data parser 3000. If desired, even capabilities that are built into secure data parser 3000 may be replaced by one or more external modules. As illustrated, some external modules include random number generator 3012, cipher feedback key generator 3014, hash algorithm 3016, any one or more types of encryption 3018, and key management 3020. It will be understood that these are merely illustrative external modules. Any other suitable modules may be used in addition to or in place of those illustrated.

[0434] Cipher feedback key generator 3014 may, externally to secure data parser 3000, generate for each secure data parser operation, a unique key, or random number (using, for example, random number generator 3012), to be used as a seed value for an operation that extends an original session key size (e.g., a value of 128, 256, 512, or 1024 bits) into a value equal to the length of the data to be parsed and split. Any suitable algorithm may be used for the

cipher feedback key generation, including, for example, the AES cipher feedback key generation algorithm.

[0435] In order to facilitate integration of secure data parser 3000 and its external modules (i.e., secure data parser layer 3026) into an application layer 3024 (e.g., email application, database application, etc.), a wrapping layer that may make use of, for example, API function calls may be used. Any other suitable arrangement for facilitating integration of secure data parser layer 3026 into application layer 3024 may be used.

[0436] FIGURE 31 illustratively shows how the arrangement of FIGURE 30 may be used when a write (e.g., to a storage device), insert (e.g., in a database field), or transmit (e.g., across a network) command is issued in application layer 3024. At step 3100 data to be secured is identified and a call is made to the secure data parser. The call is passed through wrapper layer 3022 where at step 3102, wrapper layer 3022 streams the input data identified at step 3100 into assembled data buffer 3008. Also at step 3102, any suitable share information, filenames, any other suitable information, or any combination thereof may be stored (e.g., as information 3106 at wrapper layer 3022). Secure data processor 3000 then parses and splits the data it takes as input from assembled data buffer 3008 in accordance with the present invention. It outputs the data shares into split shares buffers 3010. At step 3104, wrapper layer 3022 obtains from stored information 3106 any suitable share information (i.e., stored by wrapper 3022 at step 3102) and share location(s) (e.g., from one or more configuration files). Wrapper layer 3022 then writes

the output shares (obtained from split shares buffers 3010) appropriately (e.g., written to one or more storage devices, communicated onto a network, etc.).

[0437] FIGURE 32 illustratively shows how the arrangement of FIGURE 30 may be used when a read (e.g., from a storage device), select (e.g., from a database field), or receive (e.g., from a network) occurs. At step 3200, data to be restored is identified and a call to secure data parser 3000 is made from application layer 3024. At step 3202, from wrapper layer 3022, any suitable share information is obtained and share location is determined. Wrapper layer 3022 loads the portions of data identified at step 3200 into split shares buffers 3010. Secure data parser 3000 then processes these shares in accordance with the present invention (e.g., if only three of four shares are available, then the redundancy capabilities of secure data parser 3000 may be used to restore the original data using only the three shares). The restored data is then stored in assembled data buffer 3008. At step 3204, application layer 3022 converts the data stored in assembled data buffer 3008 into its original data format (if necessary) and provides the original data in its original format to application layer 3024.

[0438] It will be understood that the parsing and splitting of original data illustrated in FIGURE 31 and the restoring of portions of data into original data illustrated in FIGURE 32 is merely illustrative. Any other suitable processes, components, or both may be used in addition to or in place of those illustrated.

[0439] FIGURE 33 is a block diagram of an illustrative process flow for parsing and splitting original data into two or more portions of data in

accordance with one embodiment of the present invention. As illustrated, the original data desired to be parsed and split is plain text 3306 (i.e., the word "SUMMIT" is used as an example). It will be understood that any other type of data may be parsed and split in accordance with the present invention. A session key 3300 is generated. If the length of session key 3300 is not compatible with the length of original data 3306, then cipher feedback session key 3304 may be generated.

[0440] In one suitable approach, original data 3306 may be encrypted prior to parsing, splitting, or both. For example, as FIGURE 33 illustrates, original data 3306 may be XORed with any suitable value (e.g., with cipher feedback session key 3304, or with any other suitable value). It will be understood that any other suitable encryption technique may be used in place of or in addition to the XOR technique illustrate. It will further be understood that although FIGURE 33 is illustrated in terms of byte by byte operations, the operation may take place at the bit level or at any other suitable level. It will further be understood that, if desired, there need not be any encryption whatsoever of original data 3306.

[0441] The resultant encrypted data (or original data if no encryption took place) is then hashed to determine how to split the encrypted (or original) data among the output buckets (e.g., of which there are four in the illustrated example). In the illustrated example, the hashing takes place by bytes and is a function of cipher feedback session key 3304. It will be understood that this is merely illustrative. The hashing may be performed at the bit level, if desired.

The hashing may be a function of any other suitable value besides cipher feedback session key 3304. In another suitable approach, hashing need not be used. Rather, any other suitable technique for splitting data may be employed.

[0442] FIGURE 34 is a block diagram of an illustrative process flow for restoring original data 3306 from two or more parsed and split portions of original data 3306 in accordance with one embodiment of the present invention. The process involves hashing the portions in reverse (i.e., to the process of FIGURE 33) as a function of cipher feedback session key 3304 to restore the encrypted original data (or original data if there was no encryption prior to the parsing and splitting). The encryption key may then be used to restore the original data (i.e., in the illustrated example, cipher feedback session key 3304 is used to decrypt the XOR encryption by XORing it with the encrypted data). This the restores original data 3306.

[0443] FIGURE 35 shows how bit-splitting may be implemented in the example of FIGURES 33 and 34. A hash may be used (e.g., as a function of the cipher feedback session key, as a function of any other suitable value) to determine a bit value at which to split each byte of data. It will be understood that this is merely one illustrative way in which to implement splitting at the bit level. Any other suitable technique may be used.

[0444] It will be understood that any reference to hash functionality made herein may be made with respect to any suitable hash algorithm. These include for example, MD5 and SHA-1. Different hash algorithms may

be used at different times and by different components of the present invention.

[0445] After a split point has been determined in accordance with the above illustrative procedure or through any other procedure or algorithm, a determination may be made with regard to which data portions to append each of the left and right segments. Any suitable algorithm may be used for making this determination. For example, in one suitable approach, a table of all possible distributions (e.g., in the form of pairings of destinations for the left segment and for the right segment) may be created, whereby a destination share value for each of the left and right segment may be determined by using any suitable hash function on corresponding data in the session key, cipher feedback session key, or any other suitable random or pseudo-random value, which may be generated and extended to the size of the original data. For example, a hash function of a corresponding byte in the random or pseudo-random value may be made. The output of the hash function is used to determine which pairing of destinations (i.e., one for the left segment and one for the right segment) to select from the table of all the destination combinations. Based on this result, each segment of the split data unit is appended to the respective two shares indicated by the table value selected as a result of the hash function.

[0446] Redundancy information may be appended to the data portions in accordance with the present invention to allow for the restoration of the original data using fewer than all the data portions. For example, if two out of four portions are desired to be sufficient for restoration of data, then additional data from the

shares may be accordingly appended to each share in, for example, a round-robin manner (e.g., where the size of the original data is 4MB, then share 1 gets its own shares as well as those of shares 2 and 3; share 2 gets its own share as well as those of shares 3 and 4; share 3 gets its own share as well as those of shares 4 and 1; and share 4 gets its own shares as well as those of shares 1 and 2). Any such suitable redundancy may be used in accordance with the present invention.

[0447] It will be understood that any other suitable parsing and splitting approach may be used to generate portions of data from an original data set in accordance with the present invention. For example, parsing and splitting may be randomly or pseudo-randomly processed on a bit by bit basis. A random or pseudo-random value may be used (e.g., session key, cipher feedback session key, etc.) whereby for each bit in the original data, the result of a hash function on corresponding data in the random or pseudo-random value may indicate to which share to append the respective bit. In one suitable approach the random or pseudo-random value may be generated as, or extended to, 8 times the size of the original data so that the hash function may be performed on a corresponding byte of the random or pseudo-random value with respect to each bit of the original data. Any other suitable algorithm for parsing and splitting data on a bit by bit level may be used in accordance with the present invention. It will further be appreciated that redundancy data may be appended to the data shares such as, for example, in the manner described immediately above in accordance with the present invention.

[0448] In one suitable approach, parsing and splitting need not be random or pseudo-random. Rather, any suitable deterministic algorithm for parsing and splitting data may be used. For example, breaking up the original data into sequential shares may be employed as a parsing and splitting algorithm. Another example is to parse and split the original data bit by bit, appending each respective bit to the data shares sequentially in a round-robin manner. It will further be appreciated that redundancy data may be appended to the data shares such as, for example, in the manner described above in accordance with the present invention.

[0449] In one embodiment of the present invention, after the secure data parser generates a number of portions of original data, in order to restore the original data, certain one or more of the generated portions may be mandatory. For example, if one of the portions is used as an authentication share (e.g., saved on a physical token device), and if the fault tolerance feature of the secure data parser is being used (i.e., where fewer than all portions are necessary to restore the original data), then even though the secure data parser may have access to a sufficient number of portions of the original data in order to restore the original data, it may require the authentication share stored on the physical token device before it restores the original data. It will be understood that any number and types of particular shares may be required based on, for example, application, type of data, user, any other suitable factors, or any combination thereof.

[0450] In one suitable approach, the secure data parser or some external component to the secure data parser may encrypt one or more portions of the original data. The encrypted portions may be required to be provided and decrypted in order to restore the original data. The different encrypted portions may be encrypted with different encryption keys. For example, this feature may be used to implement a more secure "two man rule" whereby a first user would need to have a particular share encrypted using a first encryption and a second user would need to have a particular share encrypted using a second encryption key. In order to access the original data, both users would need to have their respective encryption keys and provide their respective portions of the original data. In one suitable approach, a public key may be used to encrypt one or more data portions that may be a mandatory share required to restore the original data. A private key may then be used to decrypt the share in order to be used to restore to the original data.

[0451] Any such suitable paradigm may be used that makes use of mandatory shares where fewer than all shares are needed to restore original data.

[0452] In one suitable embodiment of the present invention, distribution of data into a finite number of shares of data may be processed randomly or pseudo-randomly such that from a statistical perspective, the probability that any particular share of data receives a particular unit of data is equal to the probability that any one of the remaining shares will receive the unit of data. As a result, each share of data will have an approximately equal amount of data bits.

[0453] According to another embodiment of the present invention, each of the finite number of shares of data need not have an equal probability of receiving units of data from the parsing and splitting of the original data. Rather certain one or more shares may have a higher or lower probability than the remaining shares. As a result, certain shares may be larger or smaller in terms of bit size relative to other shares. For example, in a two-share scenario, one share may have a 1% probability of receiving a unit of data whereas the second share has a 99% probability. It should follow, therefore that once the data units have been distributed by the secure data parser among the two share, the first share should have approximately 1% of the data and the second share 99%. Any suitable probabilities may be used in accordance with the present invention.

[0454] It will be understood that the secure data parser may be programmed to distribute data to shares according to an exact (or near exact) percentage as well. For example, the secure data parser may be programmed to distribute 80% of data to a first share and the remaining 20% of data to a second share.

[0455] According to another embodiment of the present invention, the secure data parser may generate data shares, one or more of which have predefined sizes. For example, the secure data parser may split original data into data portions where one of the portions is exactly 256 bits. In one suitable approach, if it is not possible to generate a data portion having the requisite size, then the secure data parser may pad the portion to make it the correct size. Any suitable size may be used.

[0456] In one suitable approach, the size of a data portion may be the size of an encryption key, a splitting key, any other suitable key, or any other suitable data element.

[0457] As previously discussed, the secure data parser may use keys in the parsing and splitting of data. For purposes of clarity and brevity, these keys shall be referred to herein as "splitting keys." For example, the Session Master Key, previously introduced, is one type of splitting key. Also, as previously discussed, splitting keys may be secured within shares of data generated by the secure data parser. Any suitable algorithms for securing splitting keys may be used to secure them among the shares of data. For example, the Shamir algorithm may be used to secure the splitting keys whereby information that may be used to reconstruct a splitting key is generated and appended to the shares of data. Any other such suitable algorithm may be used in accordance with the present invention.

[0458] Similarly, any suitable encryption keys may be secured within one or more shares of data according to any suitable algorithm such as the Shamir algorithm. For example, encryption keys used to encrypt a data set prior to parsing and splitting, encryption keys used to encrypt a data portions after parsing and splitting, or both may be secured using, for example, the Shamir algorithm or any other suitable algorithm.

[0459] According to one embodiment of the present invention, an All or Nothing Transform (AoNT), such as a Full Package Transform, may be used to further secure data by transforming splitting keys, encryption keys, any other suitable data elements, or any combination

thereof. For example, an encryption key used to encrypt a data set prior to parsing and splitting in accordance with the present invention may be transformed by an AoNT algorithm. The transformed encryption key may then be distributed among the data shares according to, for example, the Shamir algorithm or any other suitable algorithm. In order to reconstruct the encryption key, the encrypted data set must be restored (e.g., not necessarily using all the data shares if redundancy was used in accordance with the present invention) in order to access the necessary information regarding the transformation in accordance with AoNTs as is well known by one skilled in the art. When the original encryption key is retrieved, it may be used to decrypt the encrypted data set to retrieve the original data set. It will be understood that the fault tolerance features of the present invention may be used in conjunction with the AoNT feature. Namely, redundancy data may be included in the data portions such that fewer than all data portions are necessary to restore the encrypted data set.

[0460] It will be understood that the AoNT may be applied to encryption keys used to encrypt the data portions following parsing and splitting either in place of or in addition to the encryption and AoNT of the respective encryption key corresponding to the data set prior to parsing and splitting. Likewise, AoNT may be applied to splitting keys.

[0461] In one embodiment of the present invention, encryption keys, splitting keys, or both as used in accordance with the present invention may be further encrypted using, for example, a workgroup key in order

to provide an extra level of security to a secured data set.

[0462] In one embodiment of the present invention, an audit module may be provided that tracks whenever the secure data parser is invoked to split data.

[0463] FIGURE 36 illustrates possible options 3600 for using the components of the secure data parser in accordance with the invention. Each combination of options is outlined below and labeled with the appropriate step numbers from FIGURE 36. The secure data parser may be modular in nature, allowing for any known algorithm to be used within each of the function blocks shown in FIGURE 36. For example, other key splitting (e.g., secret sharing) algorithms such as Blakely may be used in place of Shamir, or the AES encryption could be replaced by other known encryption algorithms such as Triple DES. The labels shown in the example of FIGURE 36 merely depict one possible combination of algorithms for use in one embodiment of the invention. It should be understood that any suitable algorithm or combination of algorithms may be used in place of the labeled algorithms.

[0464] 1) 3610, 3612, 3614, 3615, 3616, 3617, 3618, 3619

[0465] Using previously encrypted data at step 3610, the data may be eventually split into a predefined number of shares. If the split algorithm requires a key, a split encryption key may be generated at step 3612 using a cryptographically secure pseudo-random number generator. The split encryption key may optionally be transformed using an All or Nothing Transform (AoNT) into a transform split key at step 3614 before being key split to the predefined number of

shares with fault tolerance at step 3615. The data may then be split into the predefined number of shares at step 3616. A fault tolerant scheme may be used at step 3617 to allow for regeneration of the data from less than the total number of shares. Once the shares are created, authentication/integrity information may be embedded into the shares at step 3618. Each share may be optionally post-encrypted at step 3619.

[0466] 2) 3111, 3612, 3614, 3615, 3616, 3617, 3618, 3619

[0467] In some embodiments, the input data may be encrypted using an encryption key provided by a user or an external system. The external key is provided at step 3611. For example, the key may be provided from an external key store. If the split algorithm requires a key, the split encryption key may be generated using a cryptographically secure pseudo-random number generator at step 3612. The split key may optionally be transformed using an All or Nothing Transform (AoNT) into a transform split encryption key at step 3614 before being key split to the predefined number of shares with fault tolerance at step 3615. The data is then split to a predefined number of shares at step 3616. A fault tolerant scheme may be used at step 3617 to allow for regeneration of the data from less than the total number of shares. Once the shares are created, authentication/integrity information may be embedded into the shares at step 3618. Each share may be optionally post-encrypted at step 3619.

[0468] 3) 3612, 3613, 3614, 3615, 3612, 3614, 3615, 3616, 3617, 3618, 3619

[0469] In some embodiments, an encryption key may be generated using a cryptographically secure pseudo-

random number generator at step 3612 to transform the data. Encryption of the data using the generated encryption key may occur at step 3613. The encryption key may optionally be transformed using an All or Nothing Transform (AoNT) into a transform encryption key at step 3614. The transform encryption key and/or generated encryption key may then be split into the predefined number of shares with fault tolerance at step 3615. If the split algorithm requires a key, generation of the split encryption key using a cryptographically secure pseudo-random number generator may occur at step 3612. The split key may optionally be transformed using an All or Nothing Transform (AoNT) into a transform split encryption key at step 3614 before being key split to the predefined number of shares with fault tolerance at step 3615. The data may then be split into a predefined number of shares at step 3616. A fault tolerant scheme may be used at step 3617 to allow for regeneration of the data from less than the total number of shares. Once the shares are created, authentication/integrity information will be embedded into the shares at step 3618. Each share may then be optionally post-encrypted at step 3619.

[0470] 4) 3612, 3614, 3615, 3616, 3617, 3618, 3619

[0471] In some embodiments, the data may be split into a predefined number of shares. If the split algorithm requires a key, generation of the split encryption key using a cryptographically secure pseudo-random number generator may occur at step 3612. The split key may optionally be transformed using an All or Nothing Transform (AoNT) into a transformed split key at step 3614 before being key split into the predefined number of shares with fault tolerance at step 3615.

The data may then be split at step 3616. A fault tolerant scheme may be used at step 3617 to allow for regeneration of the data from less than the total number of shares. Once the shares are created, authentication/integrity information may be embedded into the shares at step 3618. Each share may be optionally post-encrypted at step 3619.

[0472] Although the above four combinations of options are preferably used in some embodiments of the invention, any other suitable combinations of features, steps, or options may be used with the secure data parser in other embodiments.

[0473] The secure data parser may offer flexible data protection by facilitating physical separation. Data may be first encrypted, then split into shares with "m of n" fault tolerance. This allows for regeneration of the original information when less than the total number of shares is available. For example, some shares may be lost or corrupted in transmission. The lost or corrupted shares may be recreated from fault tolerance or integrity information appended to the shares, as discussed in more detail below.

[0474] In order to create the shares, a number of keys are optionally utilized by the secure data parser. These keys may include one or more of the following:

[0475] Pre-encryption key: When pre-encryption of the shares is selected, an external key may be passed to the secure data parser. This key may be generated and stored externally in a key store (or other location) and may be used to optionally encrypt data prior to data splitting.

[0476] Split encryption key: This key may be generated internally and used by the secure data parser

to encrypt the data prior to splitting. This key may then be stored securely within the shares using a key split algorithm.

[0477] Split session key: This key is not used with an encryption algorithm; rather, it may be used to key the data partitioning algorithms when random splitting is selected. When a random split is used, a split session key may be generated internally and used by the secure data parser to partition the data into shares. This key may be stored securely within the shares using a key splitting algorithm.

[0478] Post encryption key: When post encryption of the shares is selected, an external key may be passed to the secure data parser and used to post encrypt the individual shares. This key may be generated and stored externally in a key store or other suitable location.

[0479] In some embodiments, when data is secured using the secure data parser in this way, the information may only be reassembled provided that all of the required shares and external encryption keys are present.

[0480] FIGURE 37 shows illustrative overview process 3700 for using the secure data parser of the present invention in some embodiments. As described above, two well-suited functions for secure data parser 3706 may include encryption 3702 and backup 3704. As such, secure data parser 3706 may be integrated with a RAID or backup system or a hardware or software encryption engine in some embodiments.

[0481] The primary key processes associated with secure data parser 3706 may include one or more of pre-encryption process 3708, encrypt/transform process

3710, key secure process 3712, parse/distribute process 3714, fault tolerance process 3716, share authentication process 3716, and post-encryption process 3720. These processes may be executed in several suitable orders or combinations, as detailed in FIGURE 36. The combination and order of processes used may depend on the particular application or use, the level of security desired, whether optional pre-encryption, post-encryption, or both, are desired, the redundancy desired, the capabilities or performance of an underlying or integrated system, or any other suitable factor or combination of factors.

[0482] The output of illustrative process 3700 may be two or more shares 3722. As described above, data may be distributed to each of these shares randomly (or pseudo-randomly) in some embodiments. In other embodiments, a deterministic algorithm (or some suitable combination of random, pseudo-random, and deterministic algorithms) may be used.

[0483] In addition to the individual protection of information assets, there is sometimes a requirement to share information among different groups of users or communities of interest. It may then be necessary to either control access to the individual shares within that group of users or to share credentials among those users that would only allow members of the group to reassemble the shares. To this end, a workgroup key may be deployed to group members in some embodiments of the invention. The workgroup key should be protected and kept confidential, as compromise of the workgroup key may potentially allow those outside the group to access information. Some systems and methods for workgroup key deployment and protection are discussed

below.

[0484] The workgroup key concept allows for enhanced protection of information assets by encrypting key information stored within the shares. Once this operation is performed, even if all required shares and external keys are discovered, an attacker has no hope of recreating the information without access to the workgroup key.

[0485] FIGURE 38 shows illustrative block diagram 3800 for storing key and data components within the shares. In the example of diagram 3800, the optional pre-encrypt and post-encrypt steps are omitted, although these steps may be included in other embodiments.

[0486] The simplified process to split the data includes encrypting the data using encryption key 3804 at encryption stage 3802. Portions of encryption key 3804 may then be split and stored within shares 3810 in accordance with the present invention. Portions of split encryption key 3806 may also be stored within shares 3810. Using the split encryption key, data 3808 is then split and stored in shares 3810.

[0487] In order to restore the data, split encryption key 3806 may be retrieved and restored in accordance with the present invention. The split operation may then be reversed to restore the ciphertext. Encryption key 3804 may also be retrieved and restored, and the ciphertext may then be decrypted using the encryption key.

[0488] When a workgroup key is utilized, the above process may be changed slightly to protect the encryption key with the workgroup key. The encryption key may then be encrypted with the workgroup key prior

to being stored within the shares. The modified steps are shown in illustrative block diagram 3900 of FIGURE 39.

[0489] The simplified process to split the data using a workgroup key includes first encrypting the data using the encryption key at stage 3902. The encryption key may then be encrypted with the workgroup key at stage 3904. The encryption key encrypted with the workgroup key may then be split into portions and stored with shares 3912. Split key 3908 may also be split and stored in shares 3912. Finally, portions of data 3910 are split and stored in shares 3912 using split key 3908.

[0490] In order to restore the data, the split key may be retrieved and restored in accordance with the present invention. The split operation may then be reversed to restore the ciphertext in accordance with the present invention. The encryption key (which was encrypted with the workgroup key) may be retrieved and restored. The encryption key may then be decrypted using the workgroup key. Finally, the ciphertext may be decrypted using the encryption key.

[0491] There are several secure methods for deploying and protecting workgroup keys. The selection of which method to use for a particular application depends on a number of factors. These factors may include security level required, cost, convenience, and the number of users in the workgroup. Some commonly used techniques used in some embodiments are provided below:

[0492] Hardware-based Key Storage

Hardware-based solutions generally provide the strongest guarantees for the security of encryption/decryption keys in an encryption system.

Examples of hardware-based storage solutions include tamper-resistant key token devices which store keys in a portable device (e.g., smartcard/dongle), or non-portable key storage peripherals. These devices are designed to prevent easy duplication of key material by unauthorized parties. Keys may be generated by a trusted authority and distributed to users, or generated within the hardware. Additionally, many key storage systems provide for multi-factor authentication, where use of the keys requires access both a physical object (token) and a passphrase or biometric.

[0493] Software-based Key Storage

While dedicated hardware-based storage may be desirable for high-security deployments or applications, other deployments may elect to store keys directly on local hardware (e.g., disks, RAM or non-volatile RAM stores such as USB drives). This provides a lower level of protection against insider attacks, or in instances where an attacker is able to directly access the encryption machine.

[0494] To secure keys on disk, software-based key management often protects keys by storing them in encrypted form under a key derived from a combination of other authentication metrics, including: passwords and passphrases, presence of other keys (e.g., from a hardware-based solution), biometrics, or any suitable combination of the foregoing. The level of security provided by such techniques may range from the relatively weak key protection mechanisms provided by some operating systems (e.g., MS Windows and Linux), to more robust solutions implemented using multi-factor authentication.

[0495] The secure data parser of the present invention may be advantageously used in a number of applications and technologies. For example, email system, RAID systems, video broadcasting systems, database systems, or any other suitable system may have the secure data parser integrated at any suitable level. As previously discussed, it will be understand that the secure data parser may also be integrated for protection and fault tolerance of any type of data in motion through any transport medium, including, for example, wired, wireless, or physical transport mediums. As one example, voice over Internet protocol (VoIP) applications may make use of the secure data parser of the present invention to solve problems relating to echoes and delays that are commonly found in VoIP. The need for network retry on dropped packets may be eliminated by using fault tolerance, which guarantees packet delivery even with the loss of a predetermined number of shares. Packets of data (e.g., network packets) may also be efficiently split and restored "on-the-fly" with minimal delay and buffering, resulting in a comprehensive solution for various types of data in motion. The secure data parser may act on network data packets, network voice packets, file system data blocks, or any other suitable unit of information. In addition to being integrated with a VoIP application, the secure data parser may be integrated with a file-sharing application (e.g., a peer-to-peer file-sharing application), a video broadcasting application, an electronic voting or polling application (which may implement an electronic voting protocol and blind signatures, such as the Sensus protocol), an email application, or any other

network application that may require or desire secure communication.

[0496] In some embodiments, support for network data in motion may be provided by the secure data parser of the present invention in two distinct phases -- a header generation phase and a data partitioning phase. Simplified header generation process 4000 and simplified data partitioning process 4010 are shown in FIGURES 40A and 40B, respectively. One or both of these processes may be performed on network packets, file system blocks, or any other suitable information.

[0497] In some embodiments, header generation process 4000 may be performed one time at the initiation of a network packet stream. At step 4002, a random (or pseudo-random) split encryption key, K, may be generated. The split encryption key, K, may then be optionally encrypted (e.g., using the workgroup key described above) at AES key wrap step 4004. Although an AES key wrap may be used in some embodiments, any suitable key encryption or key wrap algorithm may be used in other embodiments. AES key wrap step 4004 may operate on the entire split encryption key, K, or the split encryption key may be parsed into several blocks (e.g., 64-bit blocks). AES key wrap step 4004 may then operate on blocks of the split encryption key, if desired.

[0498] At step 4006, a secret sharing algorithm (e.g., Shamir) may be used to split the split encryption key, K, into key shares. Each key share may then be embedded into one of the output shares (e.g., in the share headers). Finally, a share integrity block and (optionally) a post-authentication tag (e.g., MAC) may be appended to the header block of each share.

Each header block may be designed to fit within a single data packet.

[0499] After header generation is complete (e.g., using simplified header generation process 4000), the secure data parser may enter the data partitioning phase using simplified data splitting process 4010. Each incoming data packet or data block in the stream is encrypted using the split encryption key, K, at step 4012. At step 4014, share integrity information (e.g., a hash H) may be computed on the resulting ciphertext from step 4012. For example, a SHA-256 hash may be computed. At step 4106, the data packet or data block may then be partitioned into two or more data shares using one of the data splitting algorithms described above in accordance with the present invention. In some embodiments, the data packet or data block may be split so that each data share contains a substantially random distribution of the encrypted data packet or data block. The integrity information (e.g., hash H) may then be appended to each data share. An optional post-authentication tag (e.g., MAC) may also be computed and appended to each data share in some embodiments.

[0500] Each data share may include metadata, which may be necessary to permit correct reconstruction of the data blocks or data packets. This information may be included in the share header. The metadata may include such information as cryptographic key shares, key identities, share nonces, signatures/MAC values, and integrity blocks. In order to maximize bandwidth efficiency, the metadata may be stored in a compact binary format.

[0501] For example, in some embodiments, the share header includes a cleartext header chunk, which is not encrypted and may include such elements as the Shamir key share, per-session nonce, per-share nonce, key identifiers (e.g., a workgroup key identifier and a post-authentication key identifier). The share header may also include an encrypted header chunk, which is encrypted with the split encryption key. An integrity header chunk, which may include integrity checks for any number of the previous blocks (e.g., the previous two blocks) may also be included in the header. Any other suitable values or information may also be included in the share header.

[0502] As shown in illustrative share format 4100 of FIGURE 41, header block 4102 may be associated with two or more output blocks 4104. Each header block, such as header block 4102, may be designed to fit within a single network data packet. In some embodiments, after header block 4102 is transmitted from a first location to a second location, the output blocks may then be transmitted. Alternatively, header block 4102 and output blocks 4104 may be transmitted at the same time in parallel. The transmission may occur over one or more similar or dissimilar communications paths.

[0503] Each output block may include data portion 4106 and integrity/authenticity portion 4108. As described above, each data share may be secured using a share integrity portion including share integrity information (e.g., a SHA-256 hash) of the encrypted, pre-partitioned data. To verify the integrity of the outputs blocks at recovery time, the secure data parser may compare the share integrity blocks of each share and then invert the split algorithm. The hash of the

recovered data may then be verified against the share hash.

[0504] Although some common applications of the secure data parser are described above, it should be clearly understood that the present invention may be integrated with any network application in order to increase security, fault-tolerance, anonymity, or any suitable combination of the foregoing.

[0505] Another aspect of the present invention is a common interface (for example, a common application programming interface ("API")) for use in securely creating, storing, and managing cryptographic keys across a plurality of platforms, interfaces, or both platforms and interfaces. The common interface may be provided by hardware, software, firmware, or any combination of hardware, software, and firmware. For example, in some embodiments, a key provider application, utility, or mechanism may provide the common interface between those systems that securely store or manage cryptographic keys and the secure data parser. The key provider application, utility, or mechanism may also permit interoperability between various key providers and key stores.

[0506] Several designs goals may be achieved in connection with the common interface, including provable security and reliance on standard cryptography. For example, cryptographic keys may be handled and transferred securely using proven industry standard mechanisms for key management. The integrity of any key provider application or mechanism may also be maintained by not permitting the common interface to interfere with the features or functionality of the underlying key provider application, utility, or

mechanism. In some embodiments, a single interface may be used for all key providers. Changes to the underlying key provider programs or the secure data parser may not impact the common interface in some embodiments. Interface changes in the key provider or secure data parser engine may have only a minimal impact on the common interface. For example, the design of the common interface (or the application, utility, or mechanism that provides the common interface) may be highly modular. Both the composition of the individual provider binaries and the mechanisms that the common interface uses to interface to these binaries may be highly modular in nature.

[0507] In some embodiments, the common interface includes functions that generate, store, retrieve, and pass cryptographic keys between applications or other interfaces or platforms. These functions may include one or more of the following illustrative functions:

[0508] OPEN - The OPEN function may be used to initiate communication with a target application or interface through the application interface binary associated with the application or interface.

[0509] CLOSE - The CLOSE function may be used to halt communication with the target application or interface.

[0510] GENERATE KEY - The GENERATE KEY function may be used to create a cryptographic key.

[0511] RETRIEVE KEY - The RETRIEVE KEY function may be used to access a key to be used in a split or restore function of a secure data parser engine.

[0512] STORE KEY - The STORE KEY function may be used to place a key within the secure data parser engine's key store (or on other target media, for

example, removable media, such as a USB flash drive or smart card).

[0513] DELETE KEY - The DELETE KEY function may be used to remove a key from the secure data parser engine's key store.

[0514] In some embodiments, each application or interface may be configured to be used by the key provider application, utility, or mechanism using an individual application interface binary. This binary (e.g., program) may translate requests from the common interface to the individual requirements of the requesting application or interface. As such, the key provider application, utility, or mechanism may seamlessly (and securely) support cryptographic key creation, deletion, storage, and management across a wide variety of platforms and interfaces.

[0515] In order for a cryptographic key to be accessible to the secure data parser engine, the key may first be placed in the secure parser engine's key store and identified for use. Each application or interface may have individualized requirements specified in the application or interface's application interface binary. As each application is added to the key provider application, utility, or mechanism, it may be supplied with a translation module that details the translation between the common interface and the unique application interface requirements.

[0516] FIGURE 42 shows illustrative process 4200 for managing a cryptographic key using a common interface. At step 4202, a request may be received from a first mechanism or interface to manage a cryptographic key. For example, a request to generate, retrieve, store, or delete a cryptographic key may be received in a first

interface format. The request may be received, for example, over a secure communications channel or via a secured communications session. If the request is received over a network, such as the Internet, an network security protocol (such as SSL, TLS, SSH, or IPsec) may be implemented.

[0517] At step 4204, the received request may be translated to a common interface request. For example, the request may be translated from a first interface format to one or more of the common interface functions described above. At step 4206, the common interface request may be authenticated. For example, the user or originating network address may be cryptographically authenticated using any authentication protocol or cryptographic handshake. The request may also be authenticated by verifying that the request is in a valid common interface format or originated from an authorized source. For example, valid requests may be cryptographically signed in some embodiments, in which case step 4206 may include verifying a cryptographic signature associated with the request. This may help prevent third-party attacks where an invalidly formed common interface request may attempt to bypass authentication. The secure data parser (or the system or mechanism executing the secure data parser) may also maintain a table of authorized clients or requesting interfaces. Clients or requesting interfaces may be authenticated using any suitable authentication protocol.

[0518] At step 4208, a determination may be made whether the common interface request has been authenticated. For example, a authentication token may be provided after a successful authentication in some

embodiments. On subsequent requests from the same interface or requesting client, the authentication token may be presented in lieu of initiating a new authentication session. The authentication token may also be associated with some expiration date, expiration time, or both. As such, the authentication token may expire after some predetermined length of time after issuance. The key provider application, utility, or mechanism may enforce all expiration dates and times associated with authentication tokens by accessing the current time from a system clock or timer. In other embodiments, each request may be authenticated regardless of the presentation of a valid and unexpired authentication token. If the request has not been authenticated, illustrative process 4200 may return to step 4202.

[0519] If, at step 4208, a determination is made that the request has been authenticated, then at step 4210 the secure data parser's cryptographic keys may be accessed. As described above, cryptographic keys may be stored or secured in a variety of ways. For example, all or some of the secure data parser's keys may be secured in a standard key store, a separate hardware key store, a smart card, a custom key store, a separate database table, or secured within one or more encrypted shares.

[0520] After accessing the secure data parser's cryptographic keys, at step 4212 the common interface request may be executed. For example, one or more of the common interface functions described above may be executed. These functions may, *inter alia*, generate a new key, retrieve an existing key, store a key, or delete an existing key from the set of the secure data

parser's cryptographic keys. Any other suitable command or action may also be executed at step 4212. At step 4214, a determination is made whether the common interface request includes any return arguments. For example, some requests may return a cryptographic key, a memory address or pointer to a cryptographic key, or any other suitable information (e.g., a success or failure return code). If no arguments are to be returned at step 4214, illustrative process 4200 may return to step 4202 and await a new request.

[0521] If one or more arguments are to be returned, at step 4216 the return arguments may be translated to a format compatible with the first interface. For example, an application interface binary may convert the return argument from the common interface format to the interface format of the requesting client. At step 4218, the return arguments may be transmitted to the first interface over a secure communications path or communications session. If the request was received over a network, such as the Internet, the return argument or arguments may be transmitted over a network connection implementing a network security protocol (such as SSL, TLS, SSH, or IPsec).

[0522] Additionally, other combinations, additions, substitutions and modifications will be apparent to the skilled artisan in view of the disclosure herein. Accordingly, the present invention is not intended to be limited by the reaction of the preferred embodiments but is to be defined by a reference to the appended claims.

What is Claimed is:

1. A method for managing cryptographic keys, the method comprising:

receiving, from a first interface, a request to manage at least one cryptographic key stored remote from the first interface;

translating the request from a first interface format to a common interface format;

authenticating the request by at least verifying that the request originated from an authorized source; and

executing the translated request in the common interface format.

2. The method of claim 1 wherein executing the translated request comprises retrieving the at least one cryptographic key.

3. The method of claim 1 wherein executing the translated request comprises generating the at least one cryptographic key.

4. The method of claim 1 wherein executing the translated request comprises deleting the at least one cryptographic key.

5. The method of claim 1 wherein executing the translated request comprises storing the at least one cryptographic key in a key store.

6. The method of claim 1 wherein executing the translated request comprises storing the at least one cryptographic key on removable media.

7. The method of claim 1 further comprising securing a data set using the at least one cryptographic key, wherein securing the data set comprises:

encrypting the data set using the at least one cryptographic key;

generating a random or pseudo-random value;

distributing, based, at least in part, on the random or pseudorandom value, encrypted data in the data set into two or more shares; and

storing the two or more shares separately on at least one data depository.

8. The method of claim 7 wherein storing the two or more shares separately on at least one data depository comprises storing the two or more shares on at least two geographically separated data depositories.

9. The method of claim 1 further comprising translating at least one return argument of the executed request from a common interface format to a first interface format.

10. The method of claim 9 wherein the at least one return argument of the executed request comprises at least one cryptographic key.

11. The method of claim 9 further comprising transmitting the at least one return argument to the first interface over a secure communications path.

12. The method of claim 1 wherein authenticating the request comprises implementing an authentication protocol or cryptographic handshake.

13. The method of claim 1 wherein authenticating the request comprises verifying a cryptographic signature associated with the request.

14. The method of claim 1 wherein authenticating the request comprises validating an authentication token.

15. The method of claim 14 wherein validating an authentication token comprises enforcing an expiration date or expiration time associated with the authentication token.

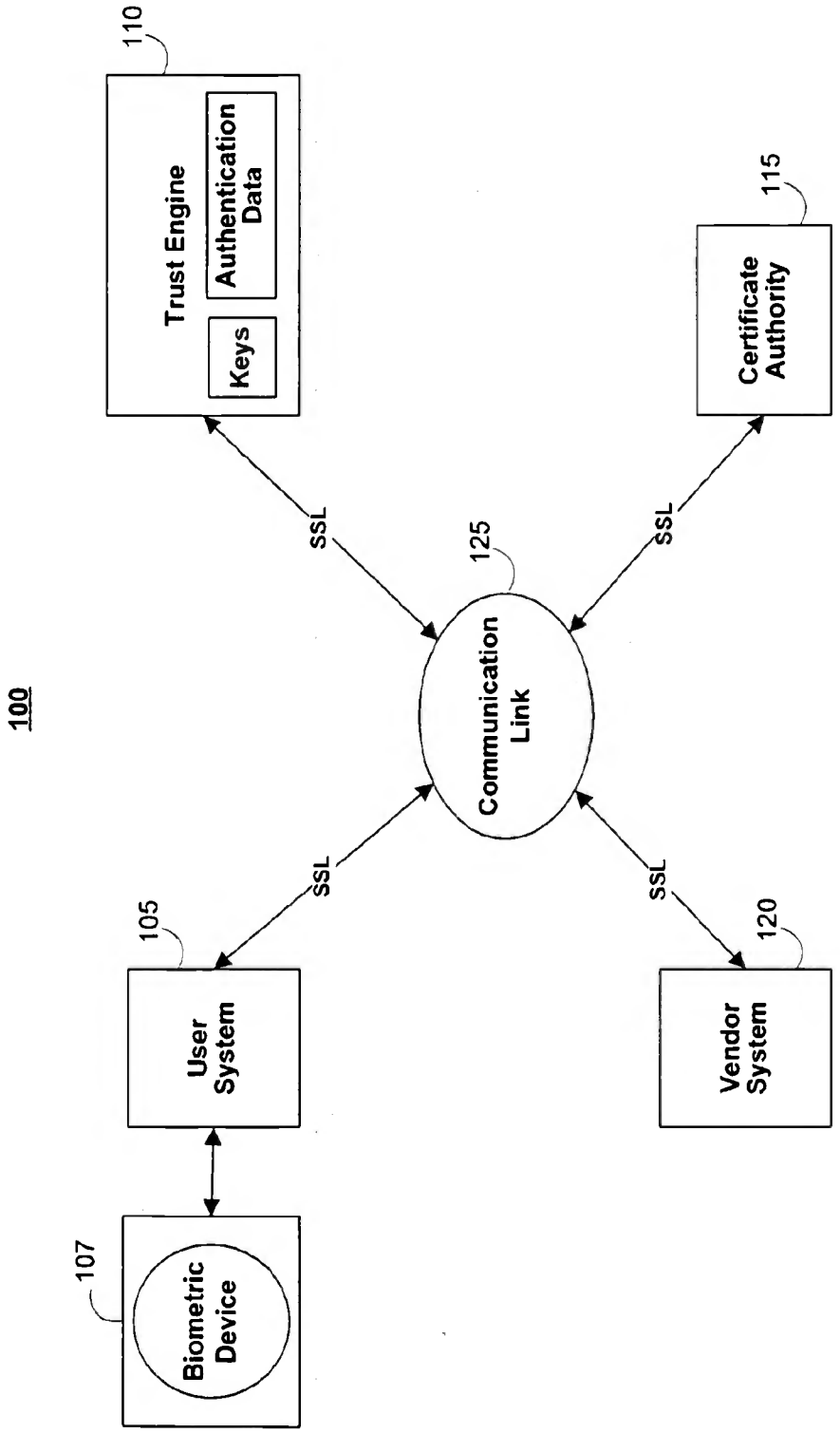

100
110
Trust Engine
Authentication Data
Keys
115
Certificate Authority
SSL
SSL
125
Communication Link
SSL
SSL
105
User System
120
Vendor System
107
Biometric Device

FIG. 1

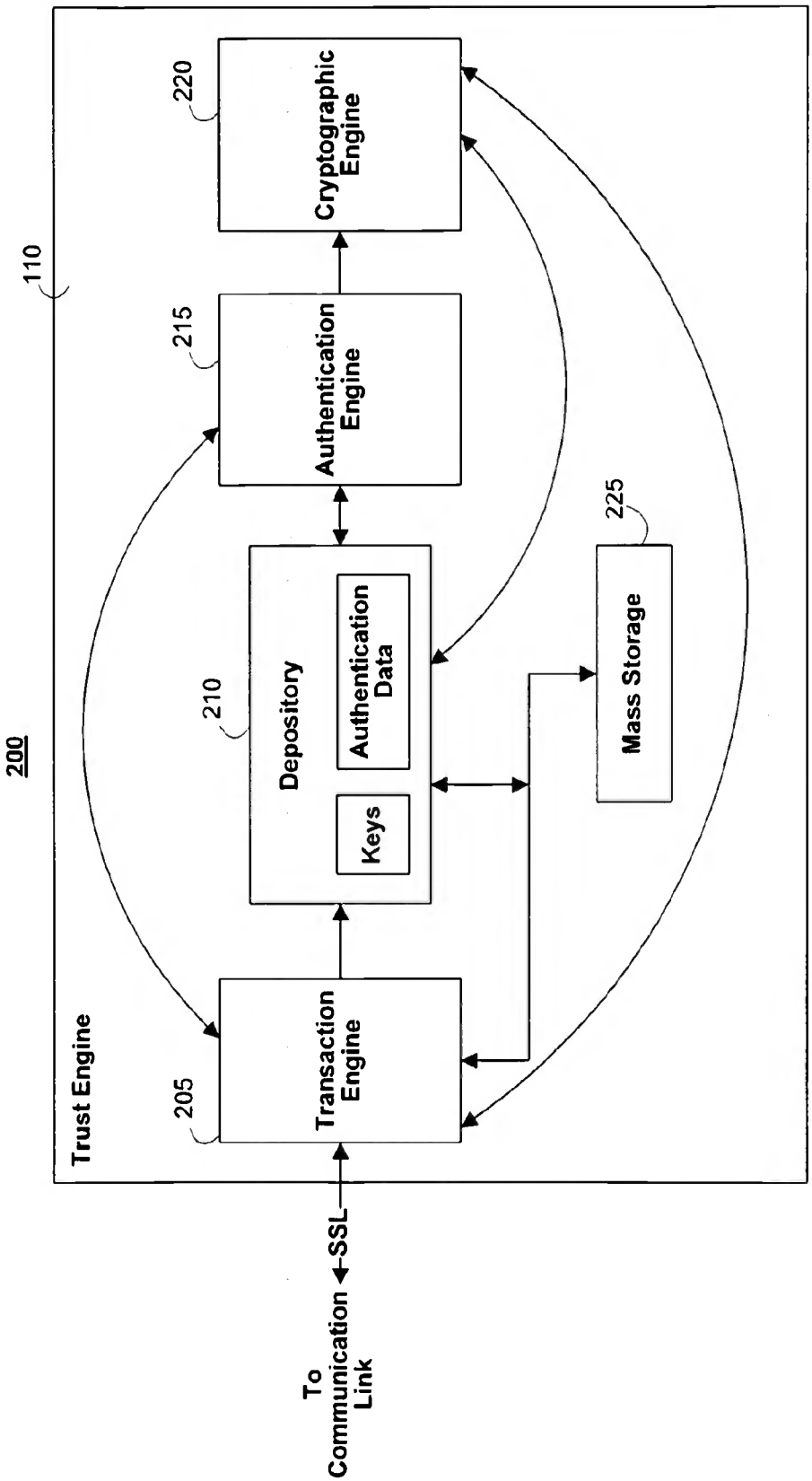

200
110
Trust Engine
205
Transaction Engine
210
Depository
Authentication Data
Keys
215
Authentication Engine
220
Cryptographic Engine
225
Mass Storage
SSL
To Communication Link

FIG. 2

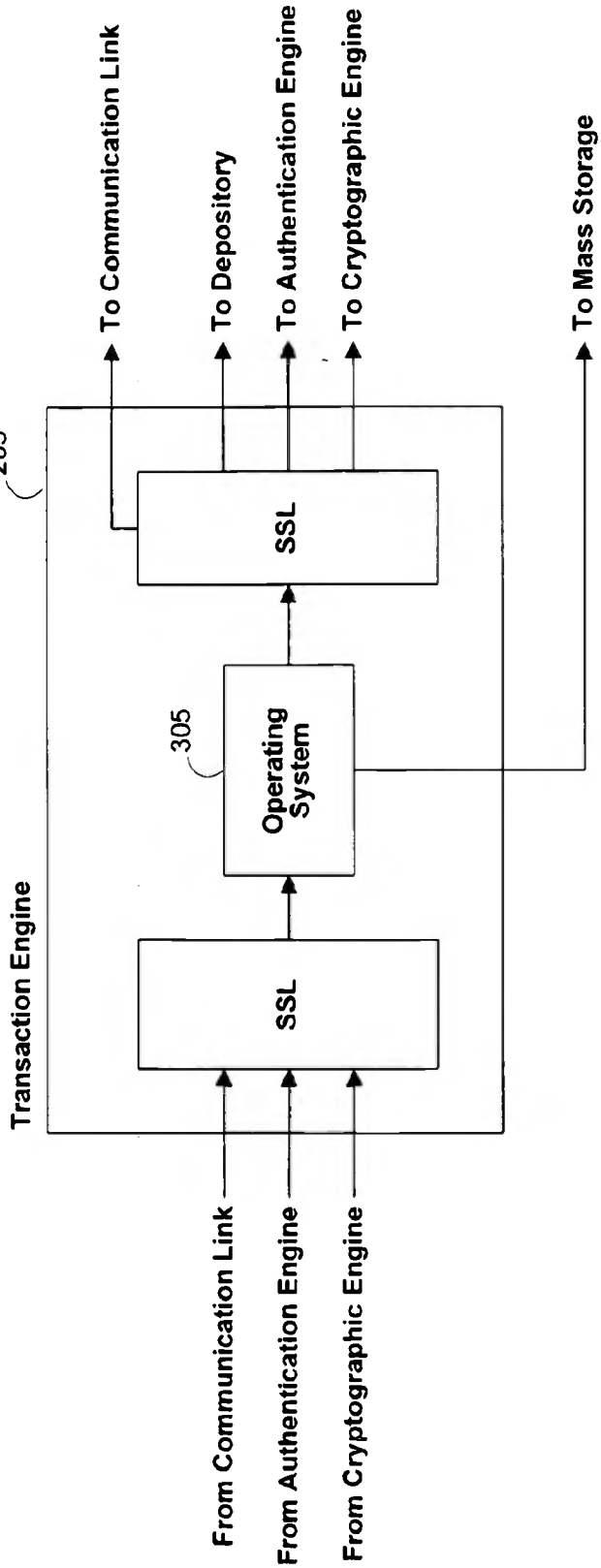

To Communication Link
To Depository
To Authentication Engine
To Cryptographic Engine
To Mass Storage
205
SSL
305
Operating System
SSL
Transaction Engine
From Communication Link
From Authentication Engine
From Cryptographic Engine

FIG. 3

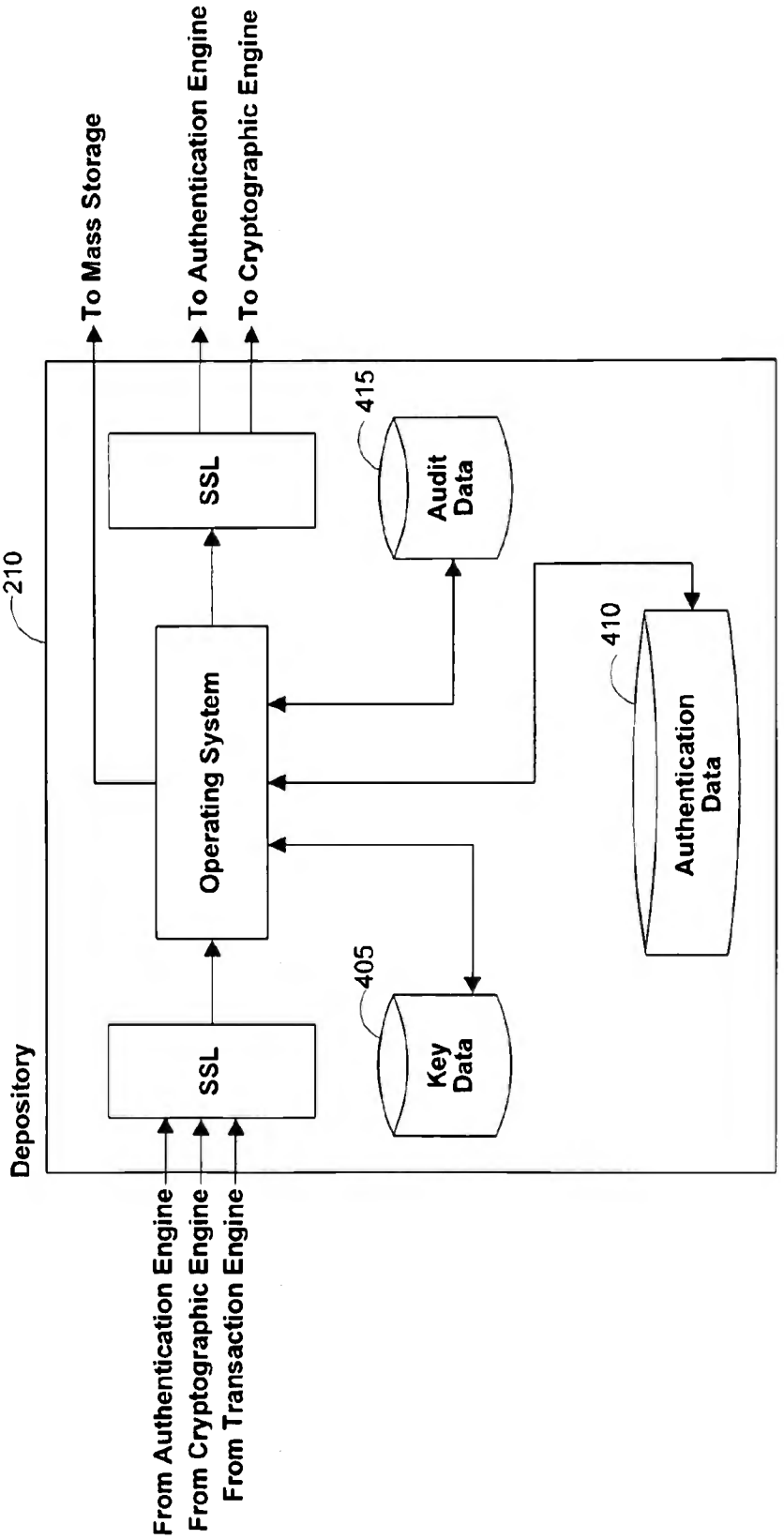

To Mass Storage
To Authentication Engine
To Cryptographic Engine
210
SSL
415
Audit Data
Operating System
410
Authentication Data
405
Key Data
SSL
Depository
From Authentication Engine
From Cryptographic Engine
From Transaction Engine

FIG. 4

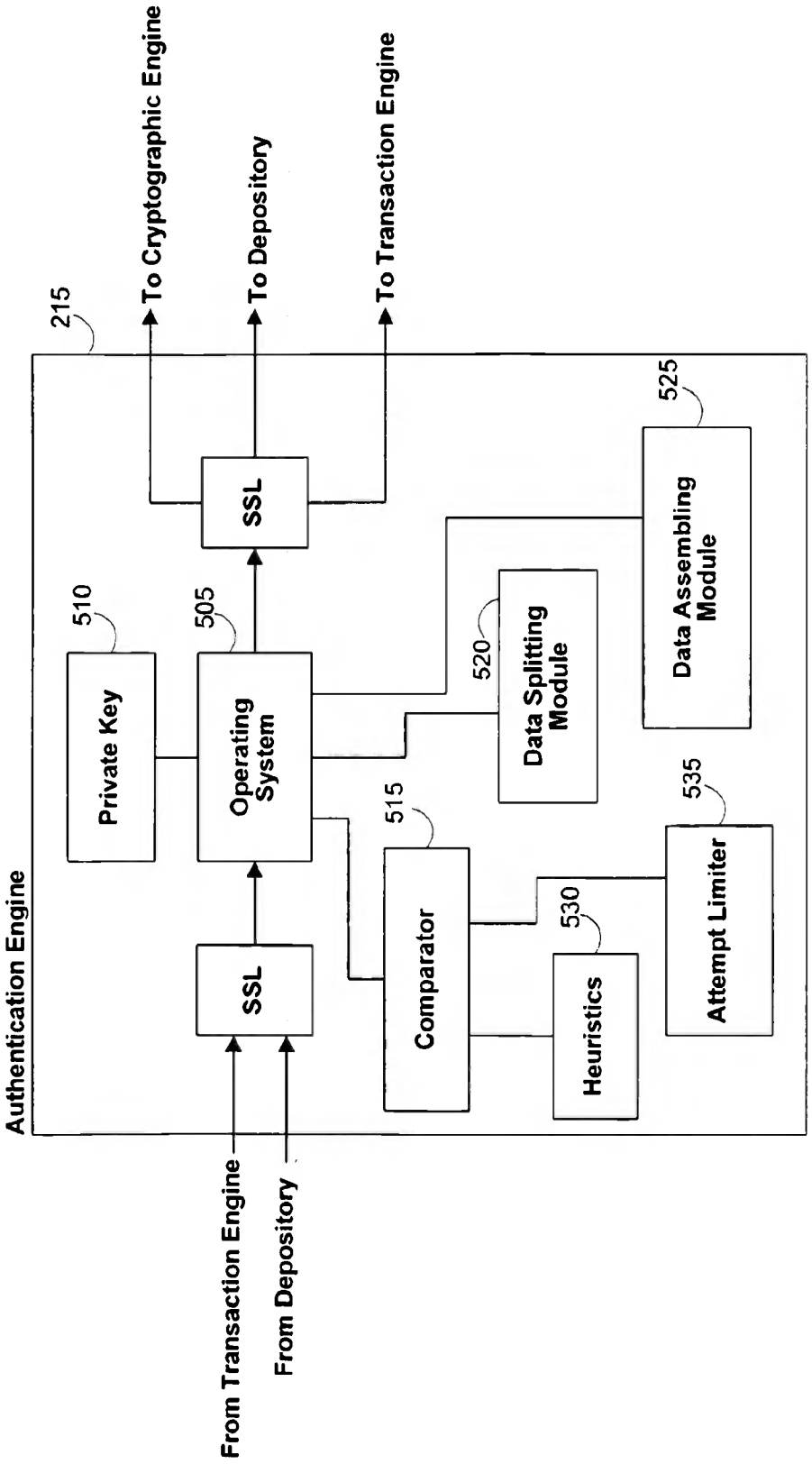

Authentication Engine
215
To Cryptographic Engine
To Depository
To Transaction Engine
From Transaction Engine
From Depository
SSL
SSL
Private Key
510
Operating System
505
Comparator
515
Heuristics
530
Attempt Limiter
535
Data Splitting Module
520
Data Assembling Module
525

FIG. 5

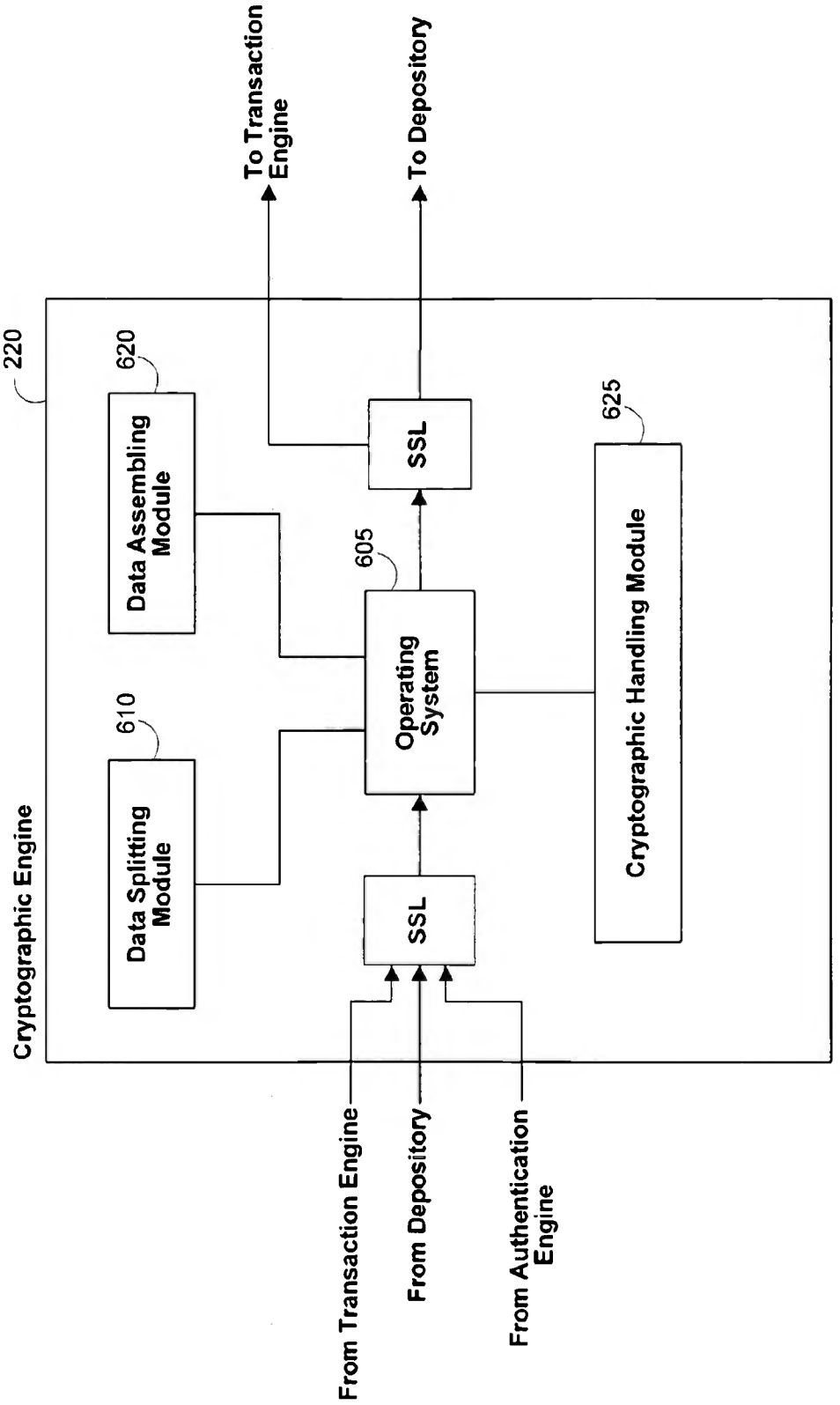

To Transaction Engine
To Depository
220
620
625
SSL
Data Assembling Module
605
Operating System
Cryptographic Handling Module
610
Data Splitting Module
SSL
Cryptographic Engine
From Transaction Engine
From Depository
From Authentication Engine

FIG. 6

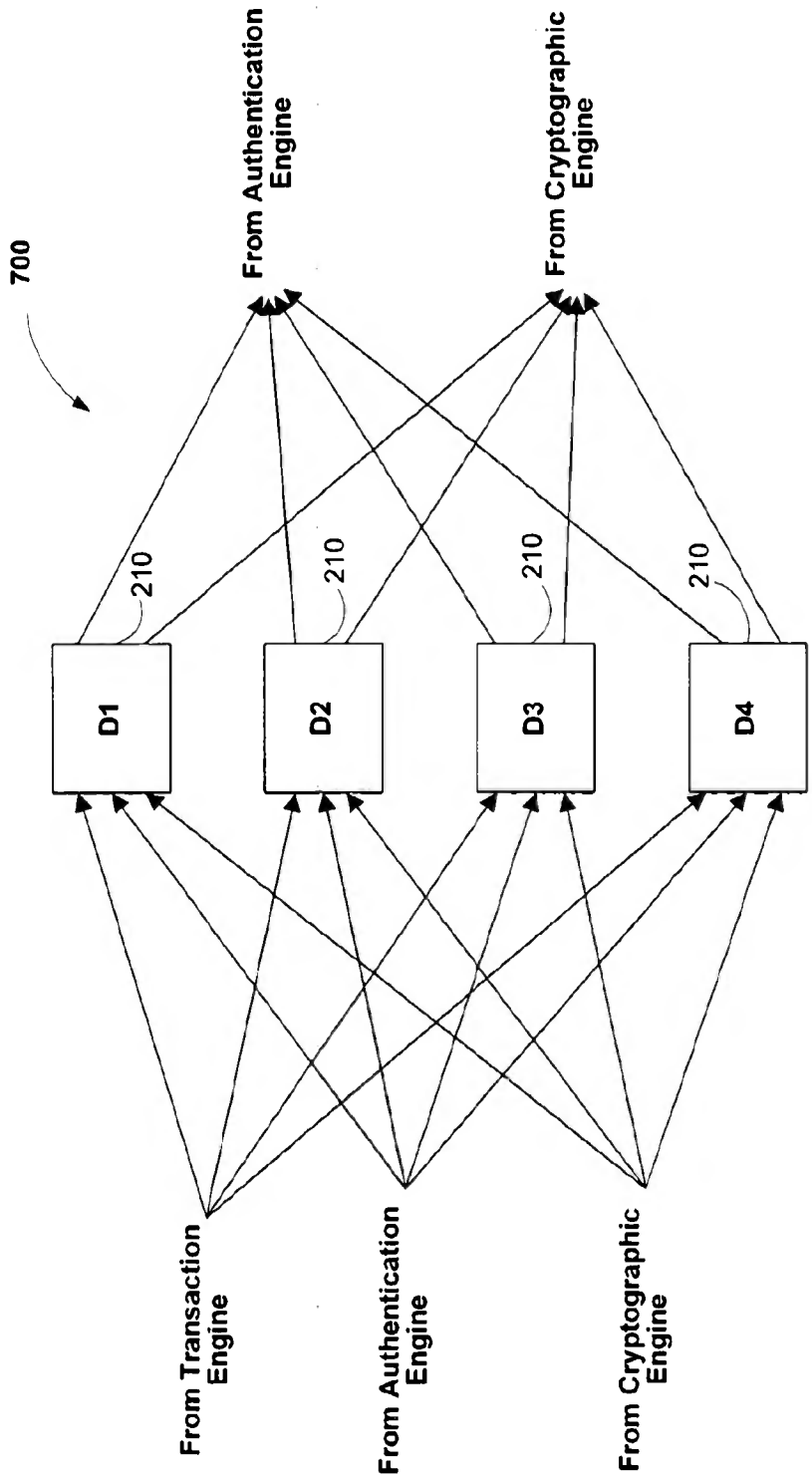

700
From Authentication Engine
From Cryptographic Engine
210
210
210
210
D1
D2
D3
D4
From Transaction Engine
From Authentication Engine
From Cryptographic Engine

FIG. 7

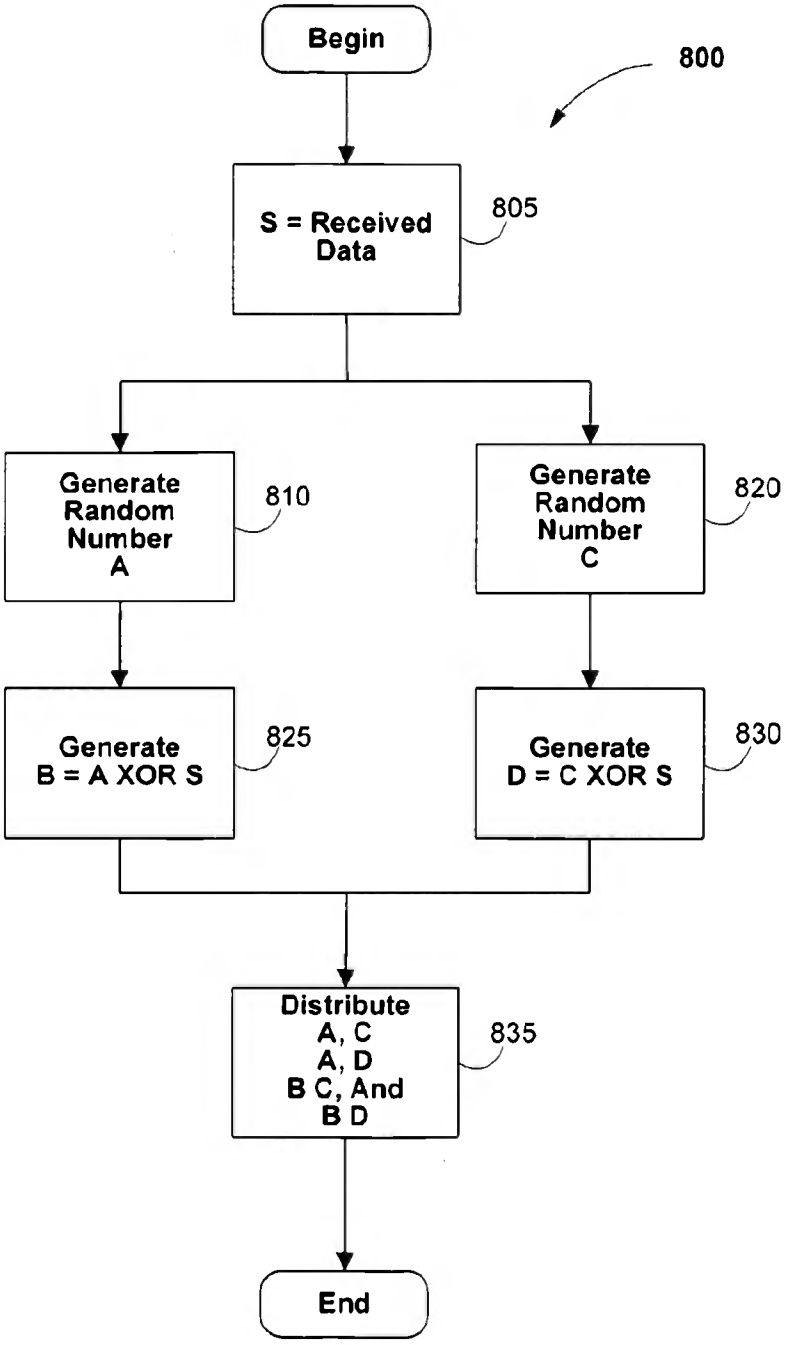
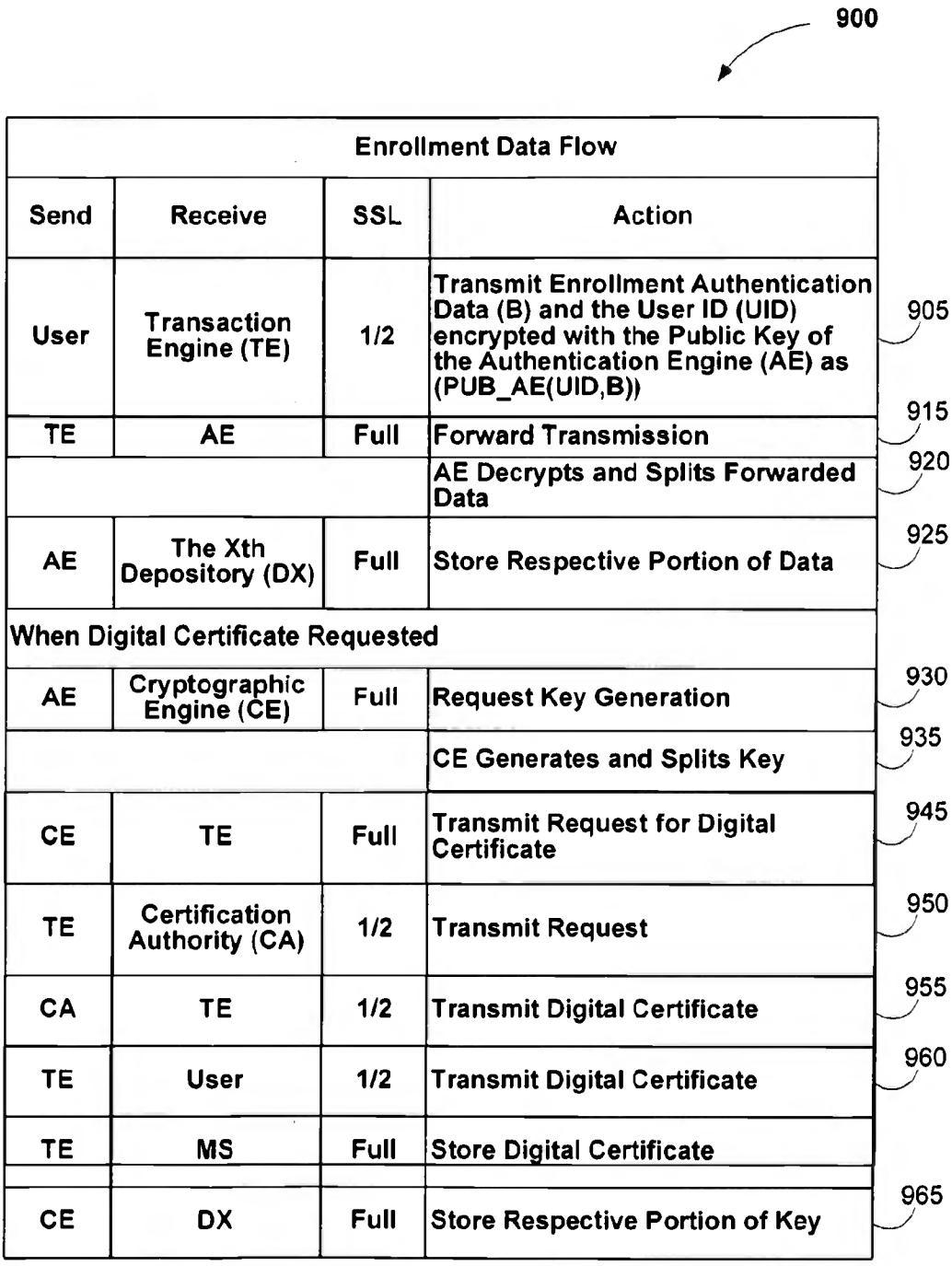

Begin
800
S = Received Data
805
Generate Random Number A
810
Generate Random Number C
820
Generate B = A XOR S
825
Generate D = C XOR S
830
Distribute A, C A, D B C, And B D
835
End

FIG. 8



Enrollment Data Flow			
Send	Receive	SSL	Action
User	Transaction Engine (TE)	1/2	Transmit Enrollment Authentication Data (B) and the User ID (UID) encrypted with the Public Key of the Authentication Engine (AE) as (PUB_AE(UID,B))
TE	AE	Full	Forward Transmission
			AE Decrypts and Splits Forwarded Data
AE	The Xth Depository (DX)	Full	Store Respective Portion of Data
When Digital Certificate Requested			
AE	Cryptographic Engine (CE)	Full	Request Key Generation
			CE Generates and Splits Key
CE	TE	Full	Transmit Request for Digital Certificate
TE	Certification Authority (CA)	1/2	Transmit Request
CA	TE	1/2	Transmit Digital Certificate
TE	User	1/2	Transmit Digital Certificate
TE	MS	Full	Store Digital Certificate
CE	DX	Full	Store Respective Portion of Key

FIG. 9, Panel A

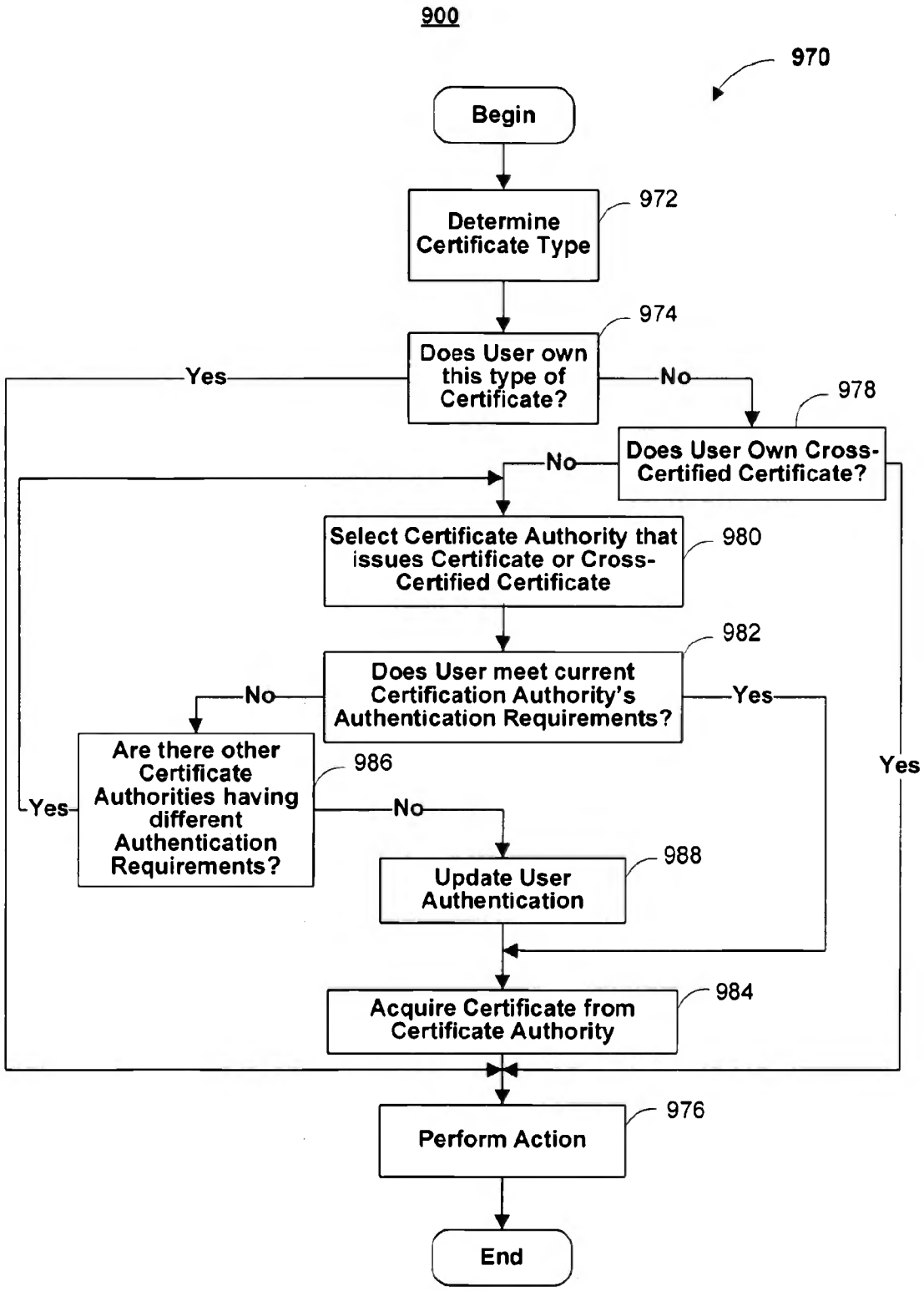

900
970
Begin
Determine Certificate Type
972
Does User own this type of Certificate?
974
Yes
No
Does User Own Cross-Certified Certificate?
978
No
Yes
Select Certificate Authority that issues Certificate or Cross-Certified Certificate
980
Does User meet current Certification Authority's Authentication Requirements?
982
No
Yes
Are there other Certificate Authorities having different Authentication Requirements?
986
Yes
No
Update User Authentication
988
Acquire Certificate from Certificate Authority
984
Perform Action
976
End

FIG. 9, Panel B

1000

	Authentication Data Flow			
	SEND	RECEIVE	SSL	ACTION
1005	User	Vendor	1/2	Transaction occurs, such as selecting purchase
1010	Vendor	User	1/2	Transmit transaction ID (TID) and authentication request (AR)
				Authentication data (B') is gathered from User
1015	User	TE	1/2	Transmit TID and B' wrapped in the Public Key of the Authentication Engine (AE), as (PUB_AE(TID, B'))
1020	TE	AE	Full	Forward transmission
				Enrollment authentication data (B) is requested and gathered
1025	Vendor	Transaction Engine (TE)	Full	Transmits TID, AR
1030	TE	Mass Storage (MS)	Full	Create Record in database
1035	TE	The Xth Depository (DX)	Full	UID, TID
1040	DX	AE	Full	Transmit the TID and the portion of the authentication data stored at enrollment (BX) as (PUB_AE(TID, BX))
1045				AE assembles B and compares to B'
1050	AE	TE	Full	TID, the filled in AR
1055	TE	Vendor	Full	TID, Yes/No
	TE	User	1/2	TID, confirmation message

FIG. 10

1100

Signing Data Flow			
SEND	RECEIVE	SSL	ACTION
User	Vendor	1/2	Transaction occurs, such as agreeing on a deal
Vendor	User	1/2	Transmit transaction identification number (TID), authentication request (AR), and agreement or message (M)
			Current authentication data (B') and a hash of the message received by the User (h(M')) is is gathered from User
User	TE	1/2	Transmit TID, B', AR, and h(M') wrapped in the Public Key of the Authentication Engine (AE), as (PUB_AE(TID, B', h(M'))
TE	AE	Full	Forward transmission
			Gather enrollment authentication data
Vendor	Transaction Engine (TE)	Full	Transmits UID, TID, AR, and a hash of the message (h(M')).
TE	Mass Storage (MS)	Full	Create Record in database
TE	The Xth Depository (DX)	Full	UID, TID
DX	AE	Full	Transmit the TID and the portion of the authentication data stored at Enrollment (BX), as (PUB_AE(TID, BX))
			The original vendor message is transmitted to the AE
TE	AE	Full	Transmit h(M)
1103			AE assembles B, compares to B' and compares h(M) to h(M')
1105 AE	Cryptographic Engine (CE)	Full	Request for digital signature and a message to be signed, for example, the hashed message
1110 AE	DX	Full	TID, signing UID
1115 DX	CE	Full	Transmit the portion of the Crypto-graphic Key corresponding to the signing party
1120			CE assembles key and signs
1125 CE	AE	Full	Transmit the digital signature (S) of signing party
1130 AE	TE	Full	TID, the filled in AR, h(M), and S
1135 TE	Vendor	Full	TID, a receipt=(TID, Yes/No, and S), and the digital signature of the trust engine, for example, a hash of the receipt encrypted with the trust engine's Private Key (Priv_TE(h(receipt)))
1140 TE	User	1/2	TID, confirmation message

FIG. 11

1200

Encryption/Decryption Data Flow				
Send	Receive	SSL	Action	
Decryption				
			Perform Authentication Data Process 1000, include the Session Key (sync) in the AR, where the sync has been encrypted with the Public Key of the User as PUB_USER(SYNC)	
			Authenticate the User	1205
AE	CE	Full	Forward PUB_USER(SYNC) to CE	1210
AE	DX	Full	UID, TID	1215
DX	CE	Full	Transmit the TID and the portion of the Private Key as (PUB_AE(TID, KEY_USER))	1220
			CE assembles the Cryptographic Key and decrypts the sync	1225
CE	AE	Full	TID, the filled in AR including decrypted sync	1230
AE	TE	Full	Forward to TE	
TE	Requesting APP/Vendor	1/2	TID, Yes/No, Sync	
Encryption				
Requesting APP/ Vendor	TE	1/2	Request for Public Key of User	1235
TE	MS	Full	Request Digital Certificate	1240
MS	TE	Full	Transmit Digital Certificate	1245
TE	Requesting APP/Vendor	1/2	Transmit Digital Certificate	1250

FIG. 12

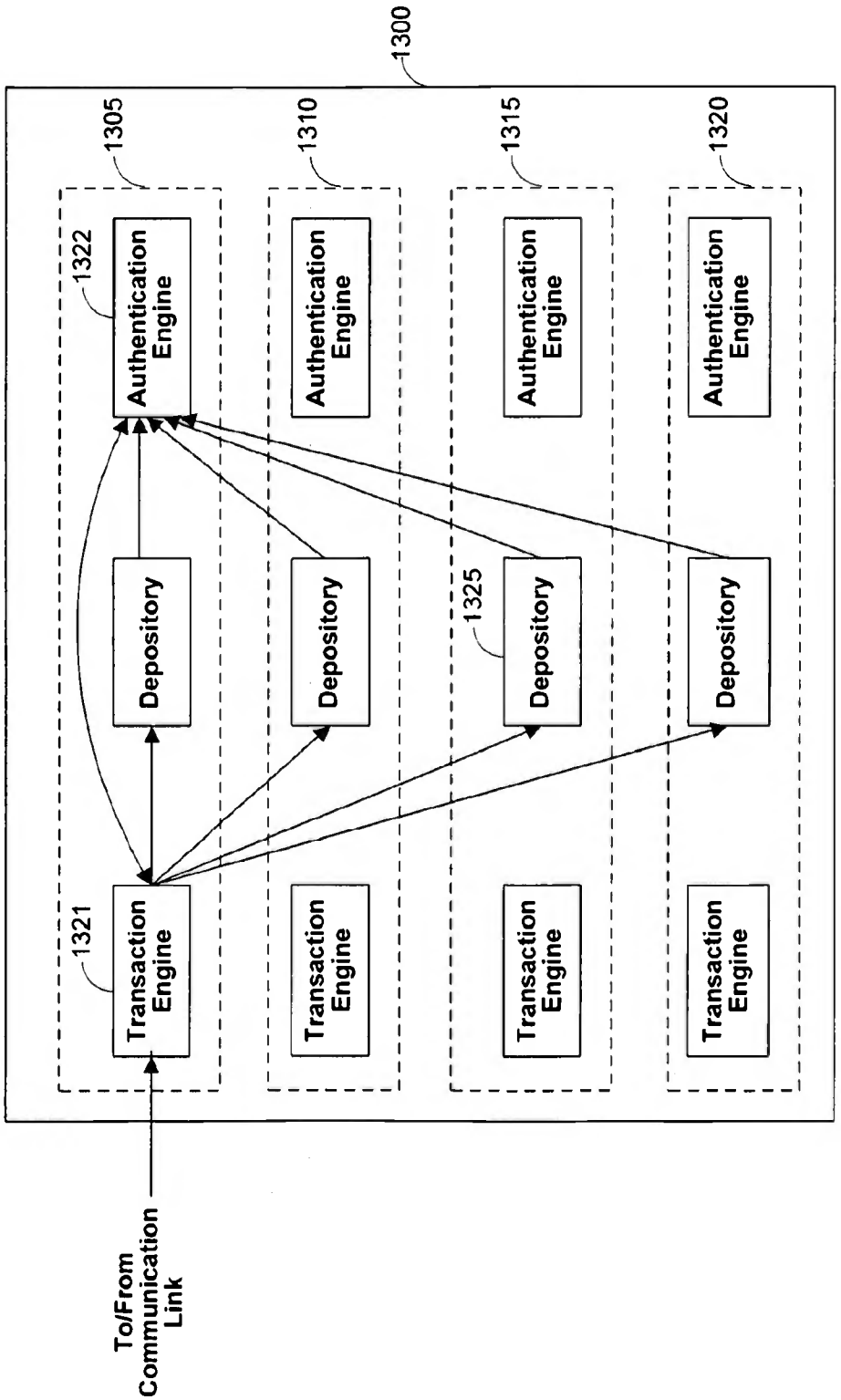

1300
1305
1310
1315
1320
1322
Authentication Engine
Depository
1325
1321
Transaction Engine
To/From Communication Link

FIG. 13

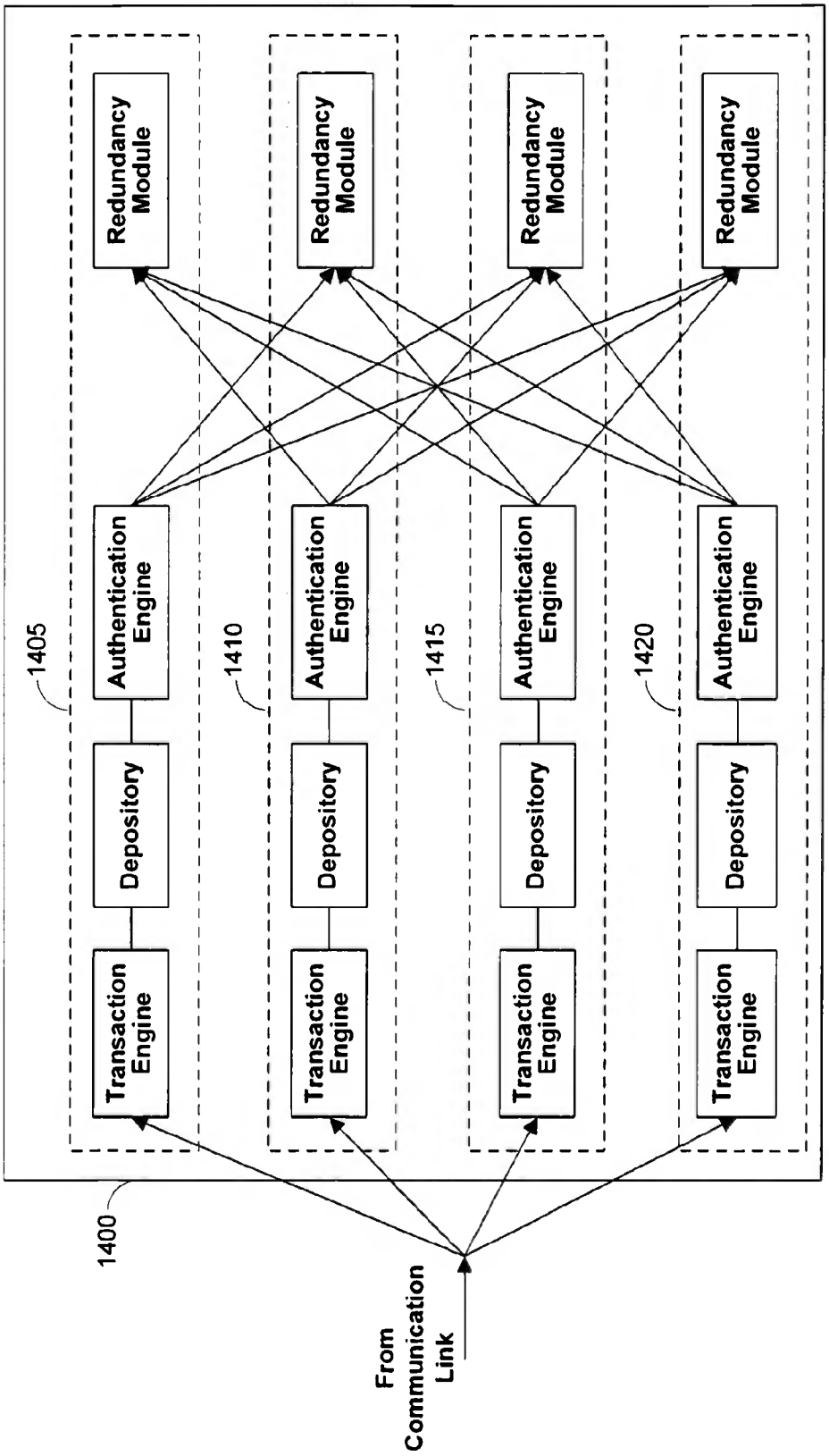

Redundancy Module
Redundancy Module
Redundancy Module
Redundancy Module
Authentication Engine
Authentication Engine
Authentication Engine
Authentication Engine
Depository
Depository
Depository
Depository
Transaction Engine
Transaction Engine
Transaction Engine
Transaction Engine
1405
1410
1415
1420
1400
From Communication Link

FIG. 14

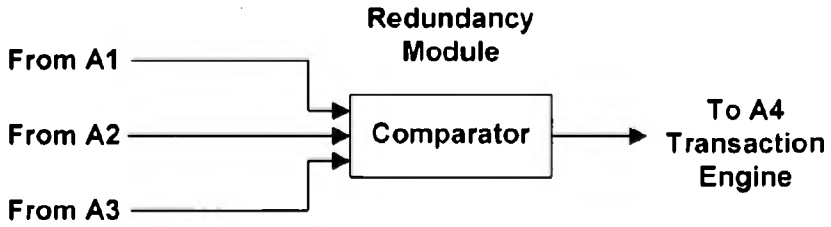

Redundancy
Module
From A1
From A2
From A3
Comparator
To A4
Transaction
Engine

FIG. 15

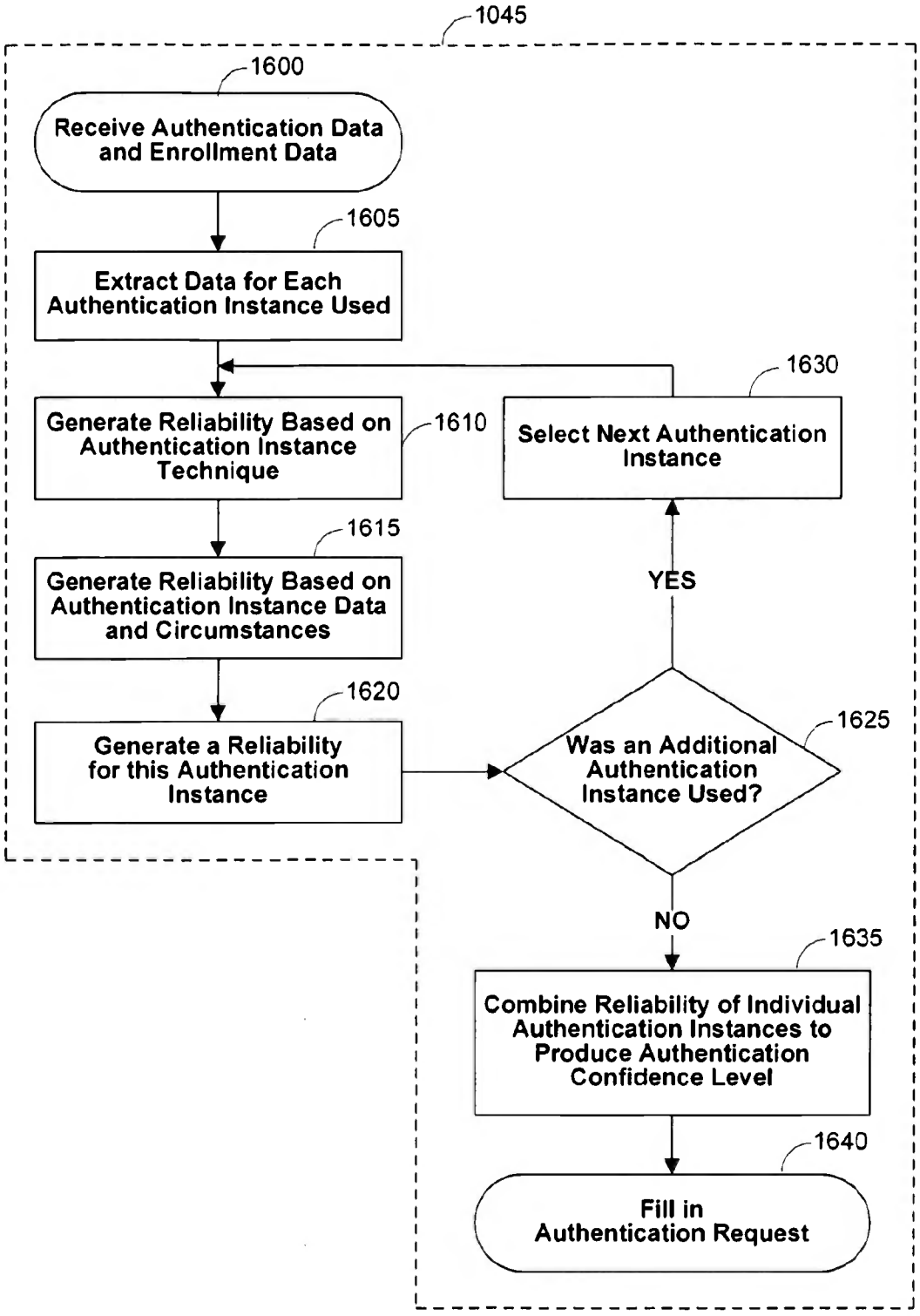

1045
1600
Receive Authentication Data and Enrollment Data
1605
Extract Data for Each Authentication Instance Used
Generate Reliability Based on Authentication Instance Technique
1610
Generate Reliability Based on Authentication Instance Data and Circumstances
1615
Generate a Reliability for this Authentication Instance
1620
Was an Additional Authentication Instance Used?
1625
YES
Select Next Authentication Instance
1630
NO
Combine Reliability of Individual Authentication Instances to Produce Authentication Confidence Level
1635
Fill in Authentication Request
1640

FIG. 16

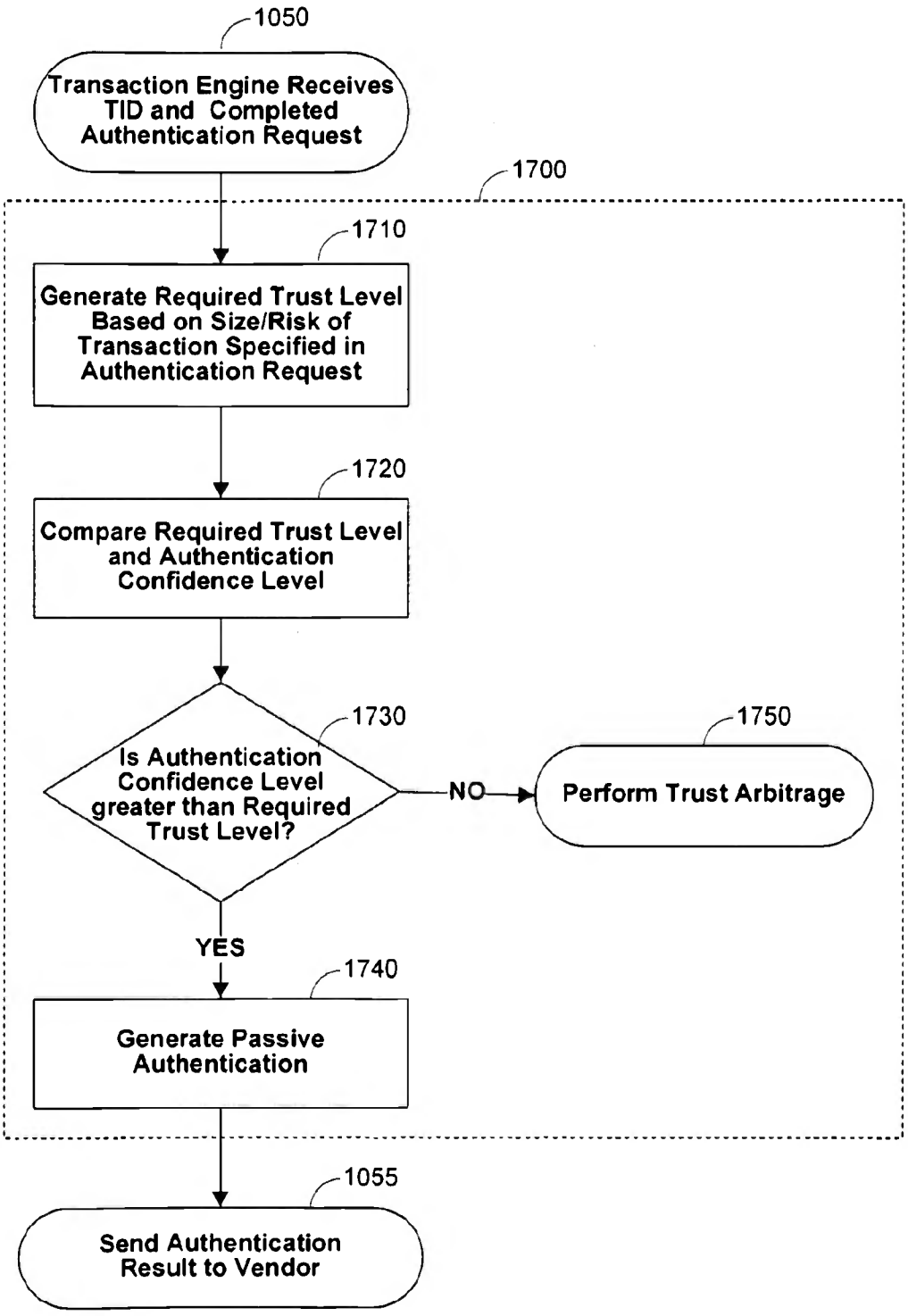

1050
Transaction Engine Receives TID and Completed Authentication Request
1700
1710
Generate Required Trust Level Based on Size/Risk of Transaction Specified in Authentication Request
1720
Compare Required Trust Level and Authentication Confidence Level
1730
Is Authentication Confidence Level greater than Required Trust Level?
NO
1750
Perform Trust Arbitrage
YES
1740
Generate Passive Authentication
1055
Send Authentication Result to Vendor

FIG. 17

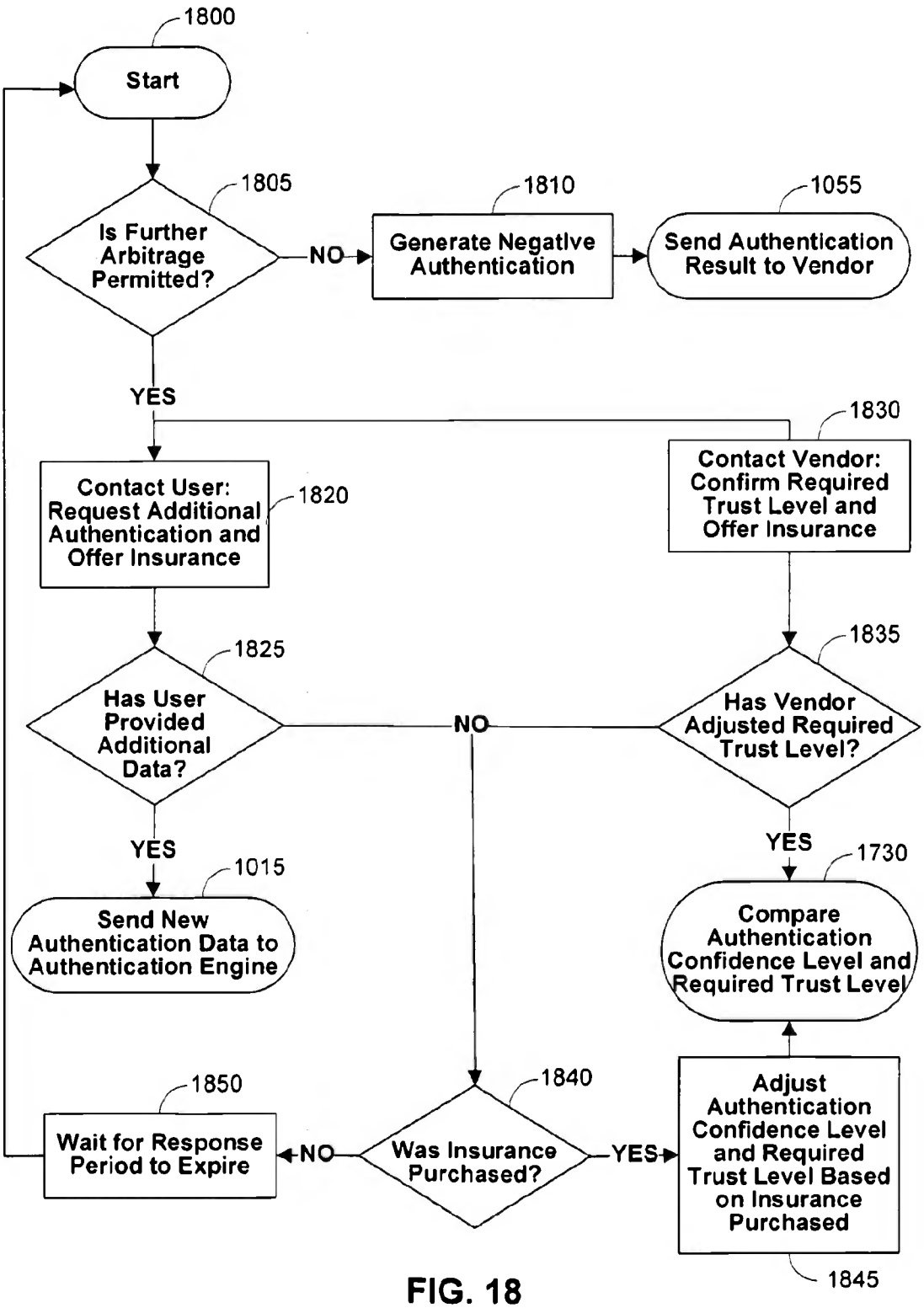

1800
Start
1805
Is Further Arbitrage Permitted?
NO
1810
Generate Negative Authentication
1055
Send Authentication Result to Vendor
YES
1830
Contact Vendor: Confirm Required Trust Level and Offer Insurance
Contact User: Request Additional Authentication and Offer Insurance
1820
1825
Has User Provided Additional Data?
NO
1835
Has Vendor Adjusted Required Trust Level?
YES
1015
Send New Authentication Data to Authentication Engine
YES
1730
Compare Authentication Confidence Level and Required Trust Level
1850
Wait for Response Period to Expire
NO
1840
Was Insurance Purchased?
YES
Adjust Authentication Confidence Level and Required Trust Level Based on Insurance Purchased
1845

FIG. 18

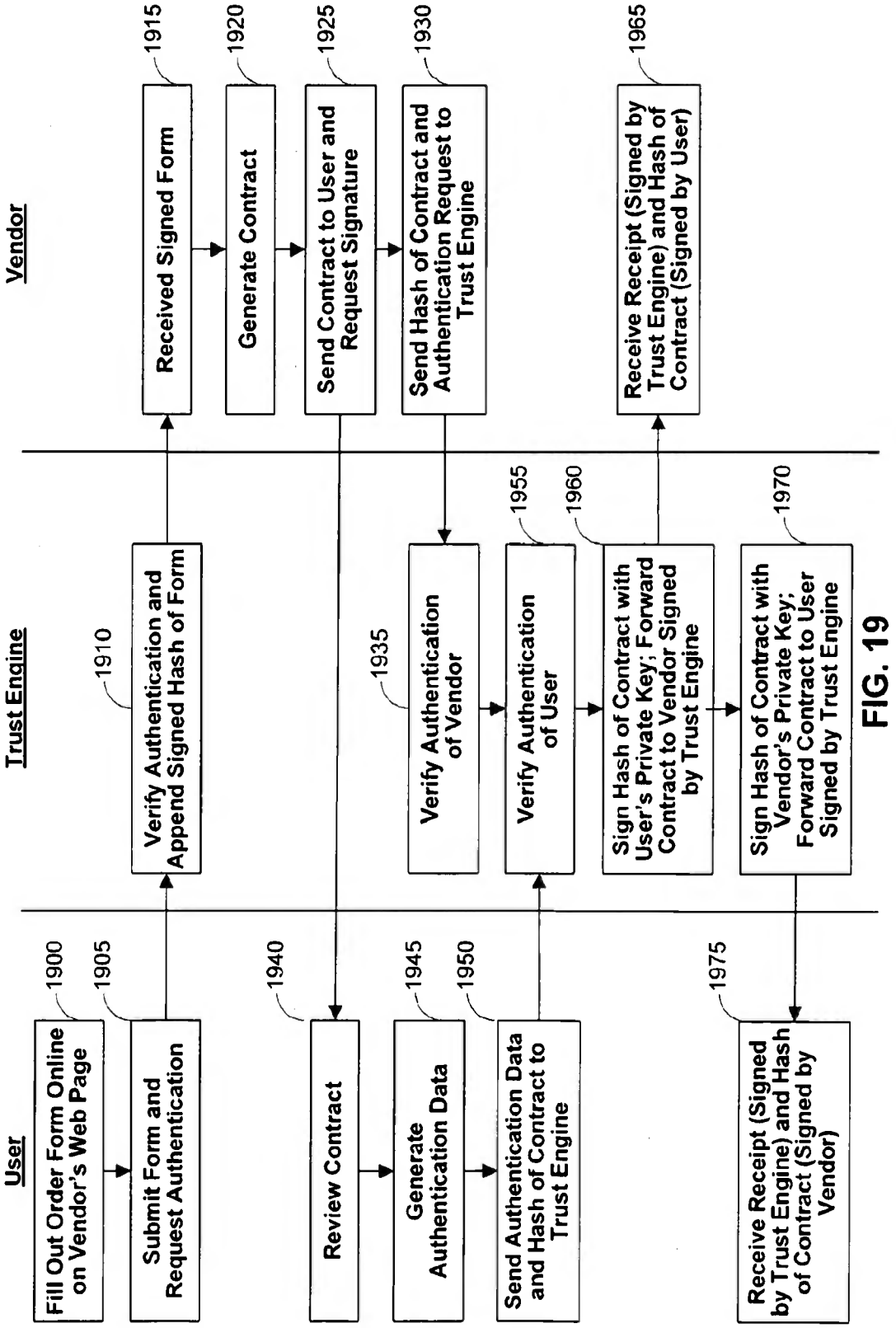

User
Trust Engine
Vendor
Fill Out Order Form Online on Vendor's Web Page
1900
Submit Form and Request Authentication
1905
Verify Authentication and Append Signed Hash of Form
1910
Received Signed Form
1915
Generate Contract
1920
Send Contract to User and Request Signature
1925
Send Hash of Contract and Authentication Request to Trust Engine
1930
Review Contract
1940
Generate Authentication Data
1945
Send Authentication Data and Hash of Contract to Trust Engine
1950
Verify Authentication of Vendor
1935
Verify Authentication of User
1955
Sign Hash of Contract with User's Private Key; Forward Contract to Vendor Signed by Trust Engine
1960
Receive Receipt (Signed by Trust Engine) and Hash of Contract (Signed by User)
1965
Sign Hash of Contract with Vendor's Private Key; Forward Contract to User Signed by Trust Engine
1970
Receive Receipt (Signed by Trust Engine) and Hash of Contract (Signed by Vendor)
1975

FIG. 19

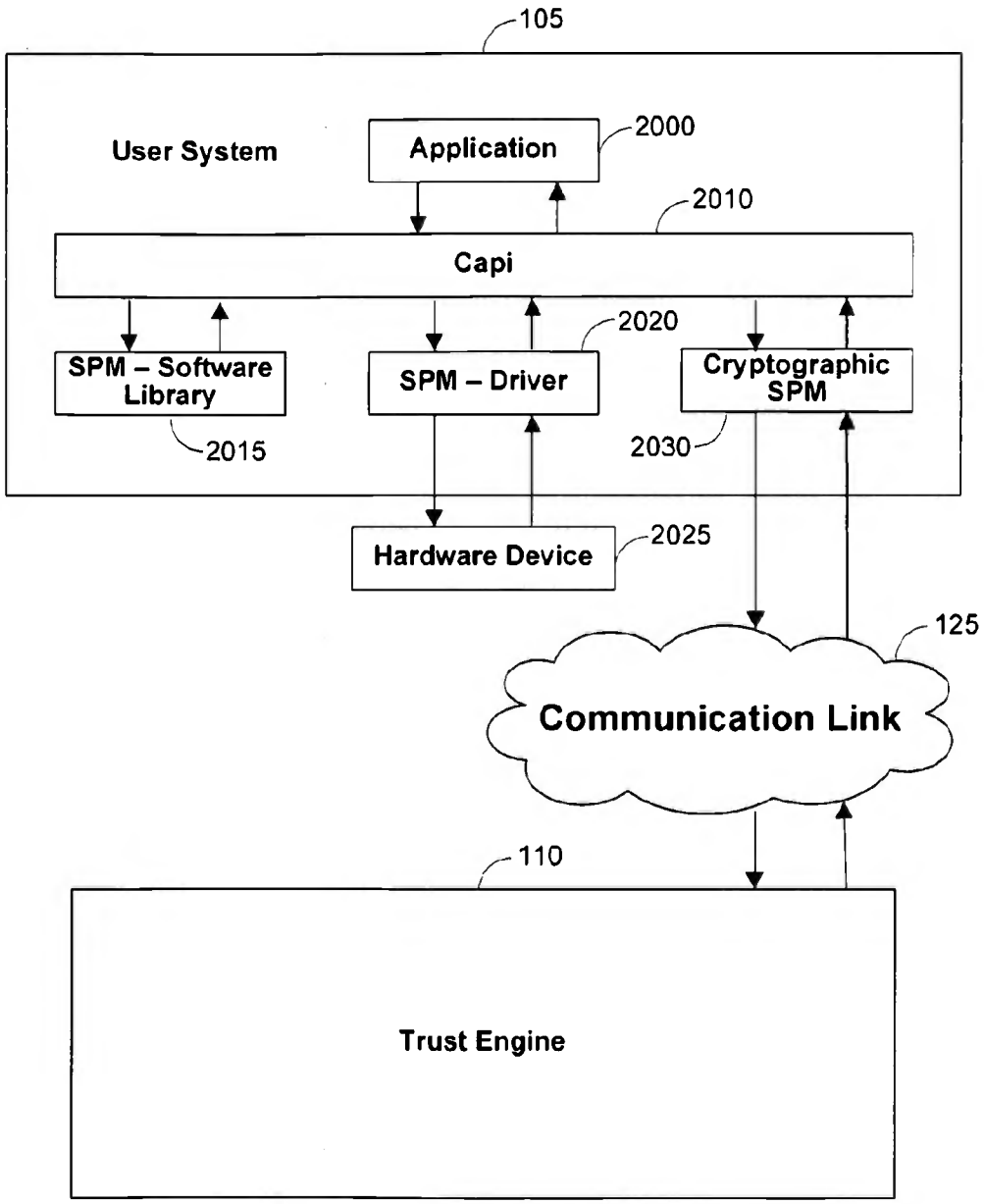

105
User System
Application
2000
2010
Capi
2020
SPM – Software Library
2015
SPM – Driver
Cryptographic SPM
2030
Hardware Device
2025
125
Communication Link
110
Trust Engine

FIG. 20

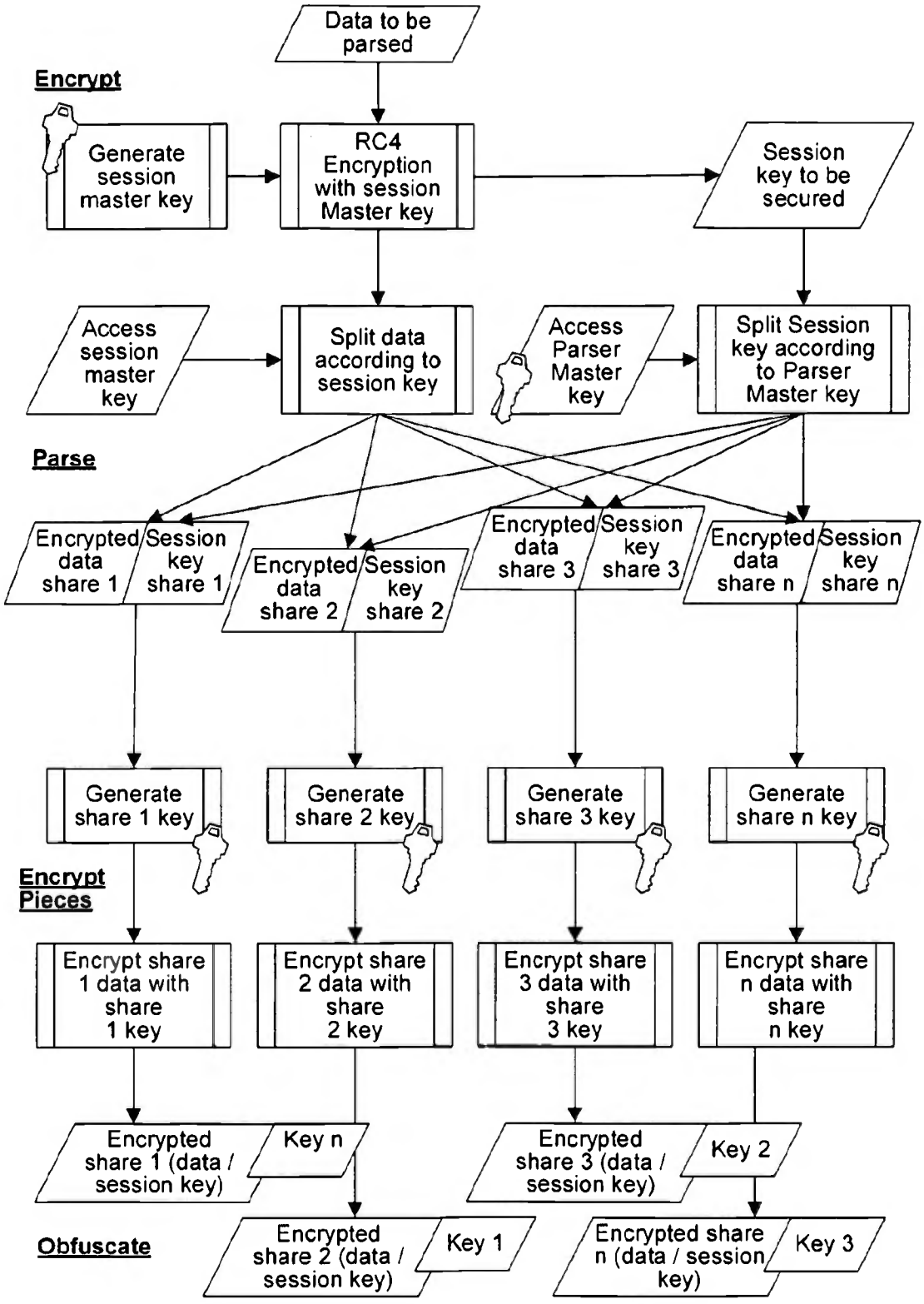

Data to be parsed
Encrypt
Generate session master key
RC4 Encryption with session Master key
Session key to be secured
Access session master key
Split data according to session key
Access Parser Master key
Split Session key according to Parser Master key
Parse
Encrypted data share 1
Session key share 1
Encrypted data share 2
Session key share 2
Encrypted data share 3
Session key share 3
Encrypted data share n
Session key share n
Generate share 1 key
Generate share 2 key
Generate share 3 key
Generate share n key
Encrypt Pieces
Encrypt share 1 data with share 1 key
Encrypt share 2 data with share 2 key
Encrypt share 3 data with share 3 key
Encrypt share n data with share n key
Encrypted share 1 (data / session key)
Key n
Encrypted share 3 (data / session key)
Key 2
Obfuscate
Encrypted share 2 (data / session key)
Key 1
Encrypted share n (data / session key)
Key 3

FIG. 21

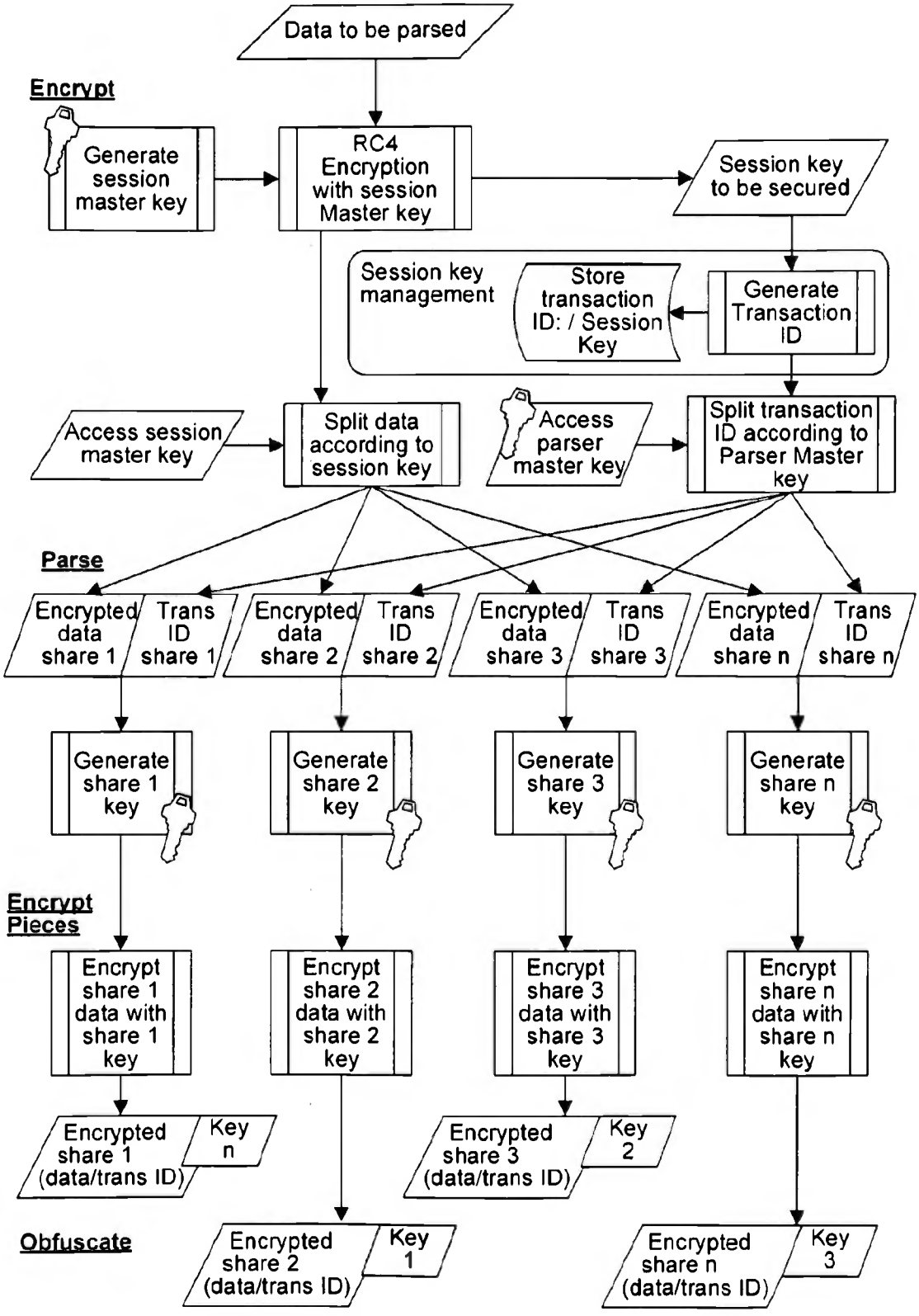

Data to be parsed
Encrypt
Generate session master key
RC4 Encryption with session Master key
Session key to be secured
Session key management
Store transaction ID: / Session Key
Generate Transaction ID
Access session master key
Split data according to session key
Access parser master key
Split transaction ID according to Parser Master key
Parse
Encrypted data share 1
Trans ID share 1
Encrypted data share 2
Trans ID share 2
Encrypted data share 3
Trans ID share 3
Encrypted data share n
Trans ID share n
Generate share 1 key
Generate share 2 key
Generate share 3 key
Generate share n key
Encrypt Pieces
Encrypt share 1 data with share 1 key
Encrypt share 2 data with share 2 key
Encrypt share 3 data with share 3 key
Encrypt share n data with share n key
Encrypted share 1 (data/trans ID)
Key n
Encrypted share 3 (data/trans ID)
Key 2
Obfuscate
Encrypted share 2 (data/trans ID)
Key 1
Encrypted share n (data/trans ID)
Key 3

FIG. 22

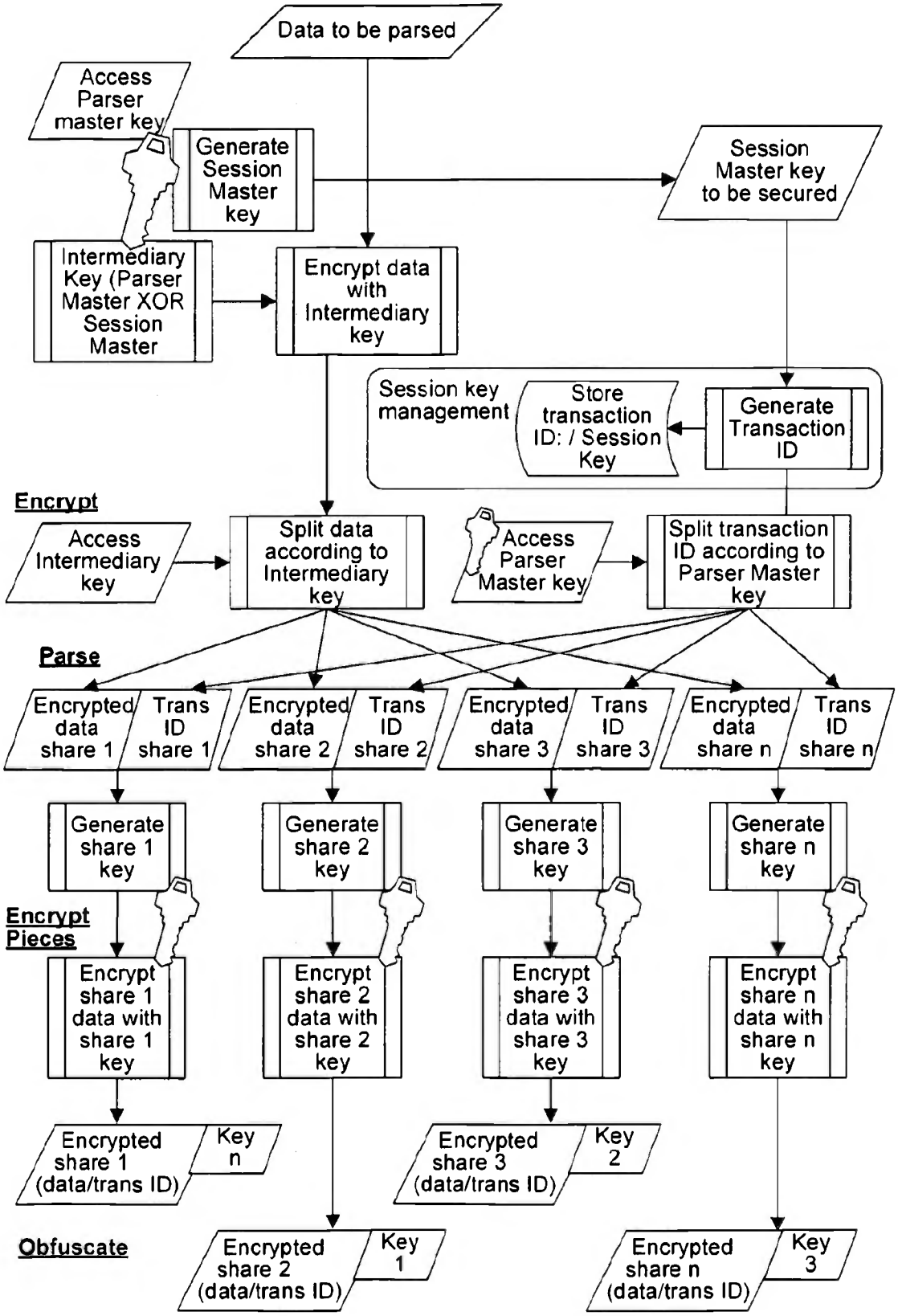

Data to be parsed
Access Parser master key
Generate Session Master key
Session Master key to be secured
Intermediary Key (Parser Master XOR Session Master
Encrypt data with Intermediary key
Session key management
Store transaction ID: / Session Key
Generate Transaction ID
Encrypt
Access Intermediary key
Split data according to Intermediary key
Access Parser Master key
Split transaction ID according to Parser Master key
Parse
Encrypted data share 1
Trans ID share 1
Encrypted data share 2
Trans ID share 2
Encrypted data share 3
Trans ID share 3
Encrypted data share n
Trans ID share n
Generate share 1 key
Generate share 2 key
Generate share 3 key
Generate share n key
Encrypt Pieces
Encrypt share 1 data with share 1 key
Encrypt share 2 data with share 2 key
Encrypt share 3 data with share 3 key
Encrypt share n data with share n key
Encrypted share 1 (data/trans ID)
Key n
Encrypted share 3 (data/trans ID)
Key 2
Obfuscate
Encrypted share 2 (data/trans ID)
Key 1
Encrypted share n (data/trans ID)
Key 3

FIG. 23

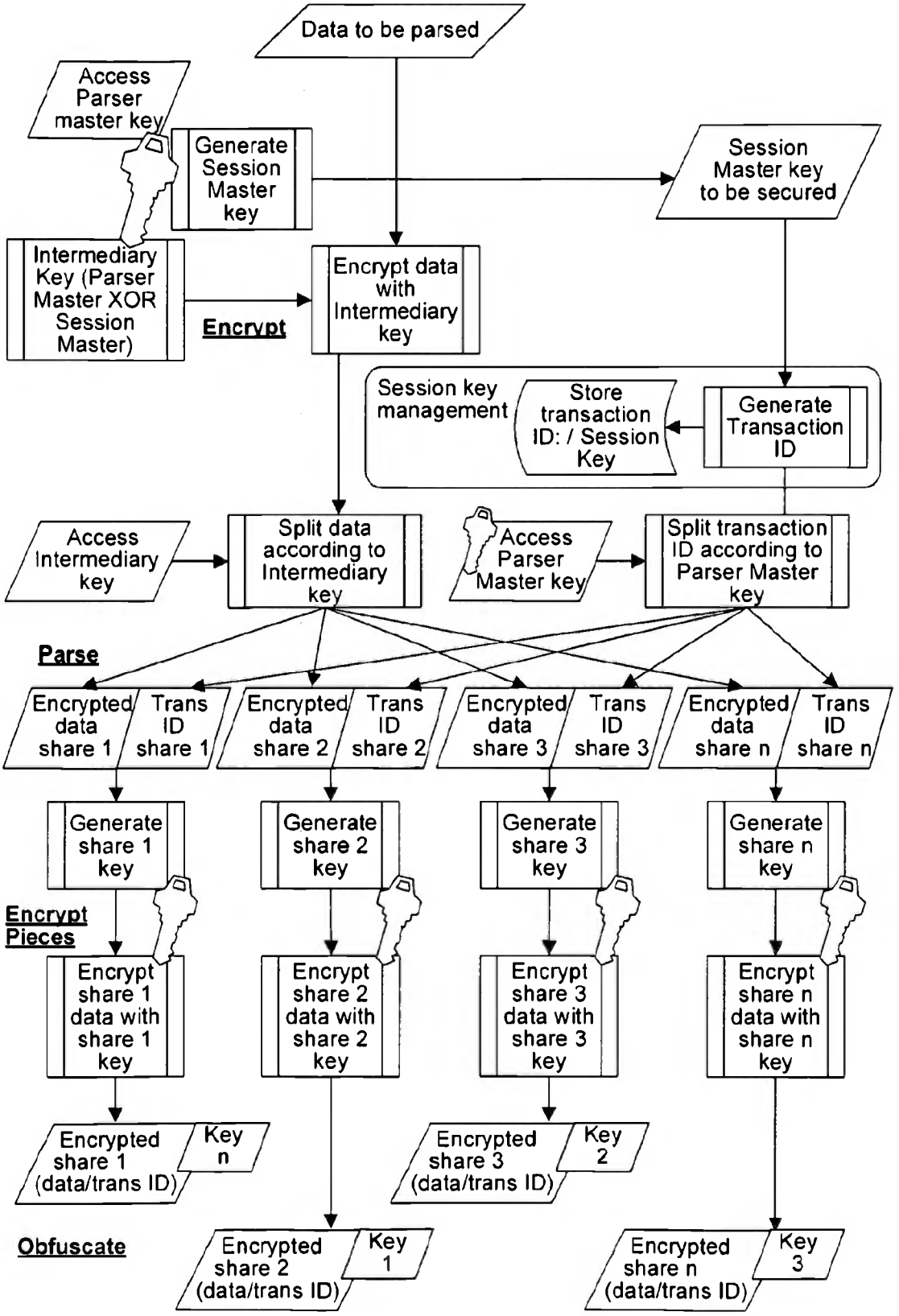

Data to be parsed
Access Parser master key
Generate Session Master key
Session Master key to be secured
Intermediary Key (Parser Master XOR Session Master)
Encrypt
Encrypt data with Intermediary key
Session key management
Store transaction ID: / Session Key
Generate Transaction ID
Access Intermediary key
Split data according to Intermediary key
Access Parser Master key
Split transaction ID according to Parser Master key
Parse
Encrypted data share 1
Trans ID share 1
Encrypted data share 2
Trans ID share 2
Encrypted data share 3
Trans ID share 3
Encrypted data share n
Trans ID share n
Generate share 1 key
Generate share 2 key
Generate share 3 key
Generate share n key
Encrypt Pieces
Encrypt share 1 data with share 1 key
Encrypt share 2 data with share 2 key
Encrypt share 3 data with share 3 key
Encrypt share n data with share n key
Encrypted share 1 (data/trans ID)
Key n
Encrypted share 3 (data/trans ID)
Key 2
Obfuscate
Encrypted share 2 (data/trans ID)
Key 1
Encrypted share n (data/trans ID)
Key 3

FIG. 24

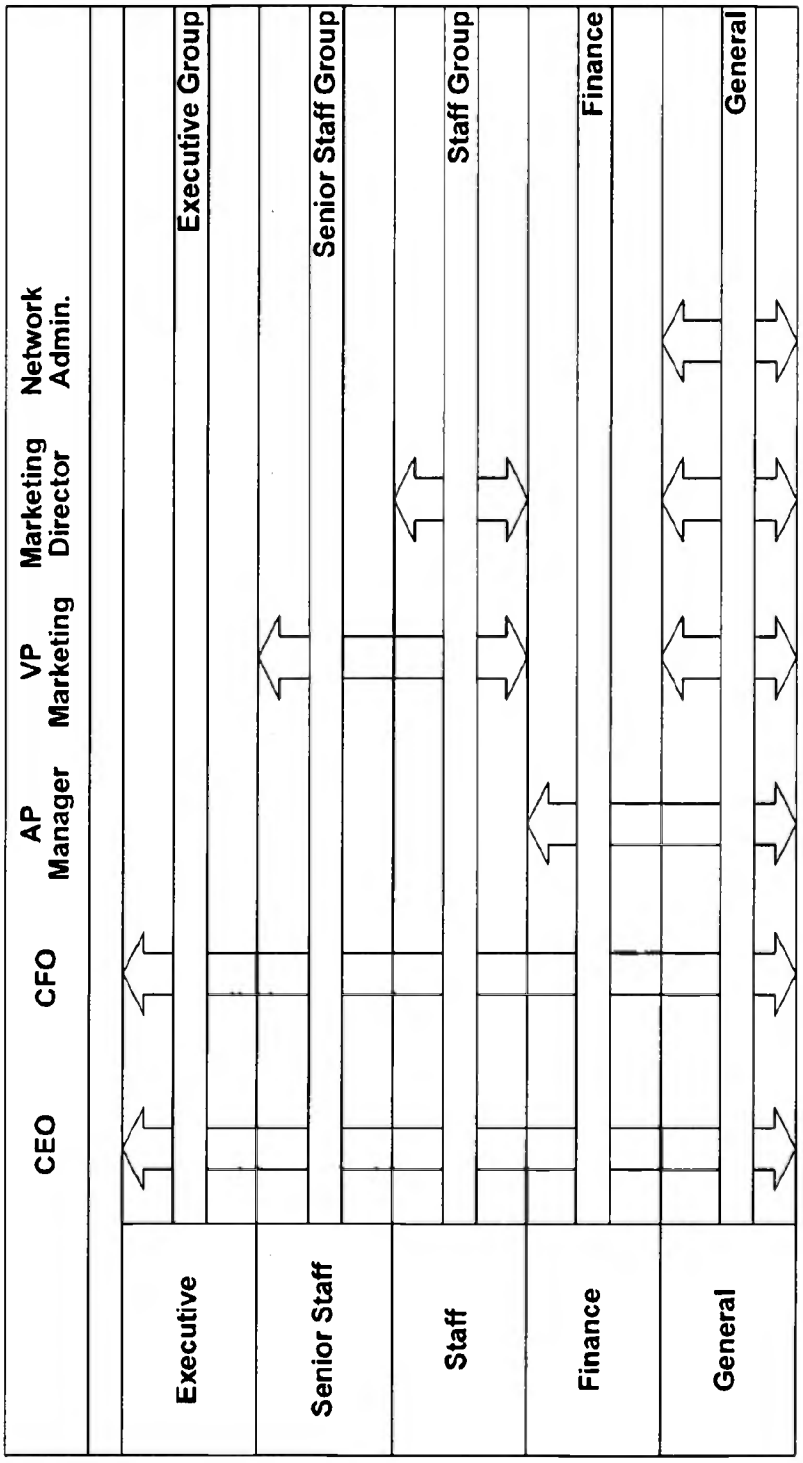

CEO
CFO
AP Manager
VP Marketing
Marketing Director
Network Admin.
Executive Group
Senior Staff Group
Staff Group
Finance
General
Executive
Senior Staff
Staff
Finance
General

FIG. 25

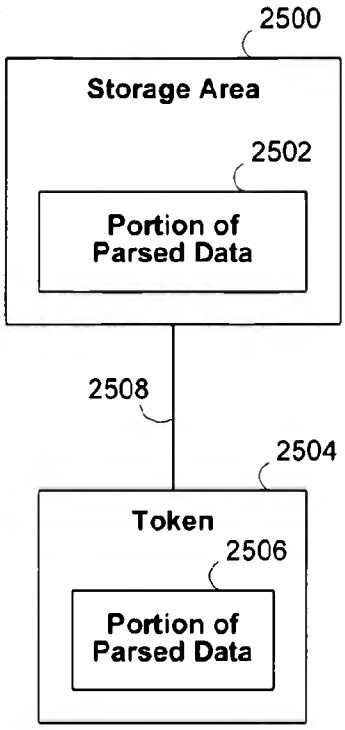

2500
Storage Area
2502
Portion of Parsed Data
2508
2504
Token
2506
Portion of Parsed Data

FIG. 26

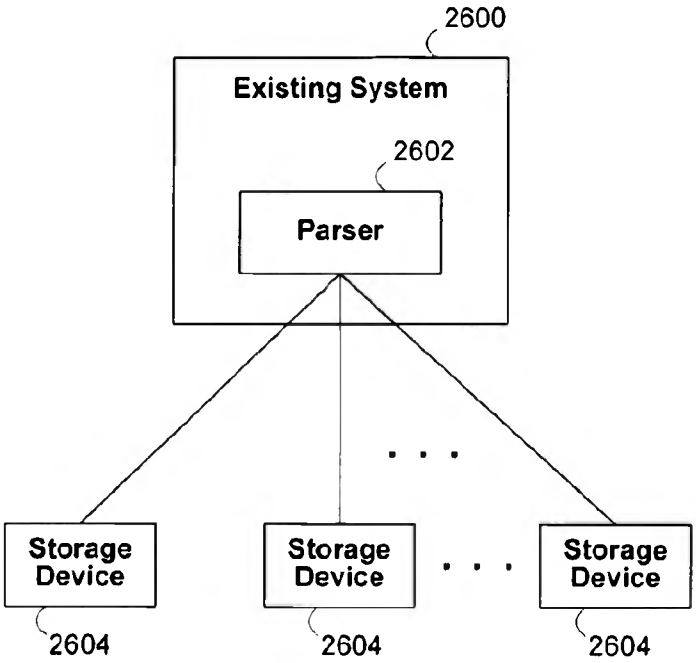

2600
Existing System
2602
Parser
Storage Device
2604
Storage Device
2604
Storage Device
2604

FIG. 27

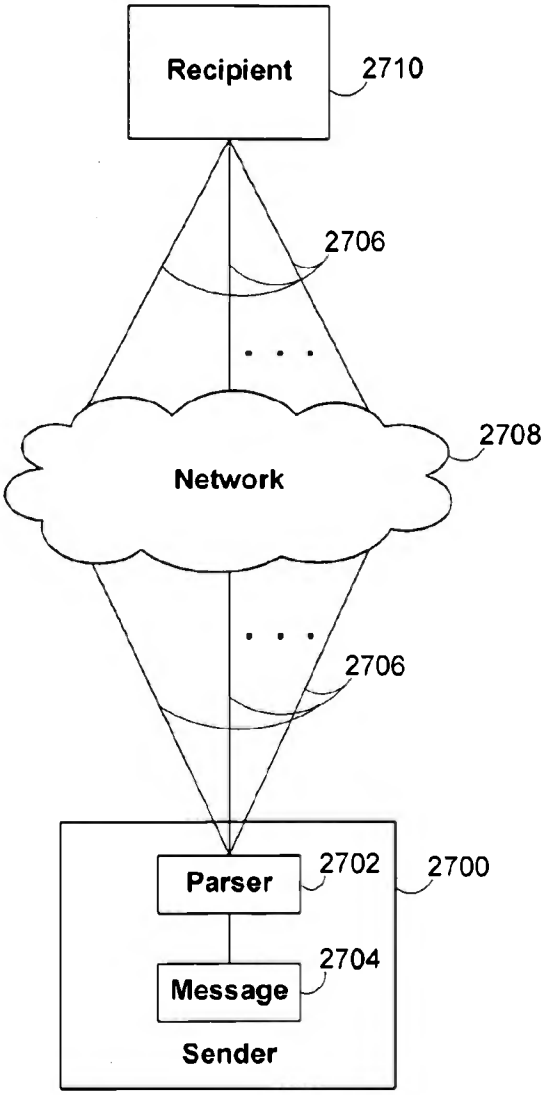

Recipient
2710
2706
2708
Network
2706
Parser
2702
2700
2704
Message
Sender

FIG. 28

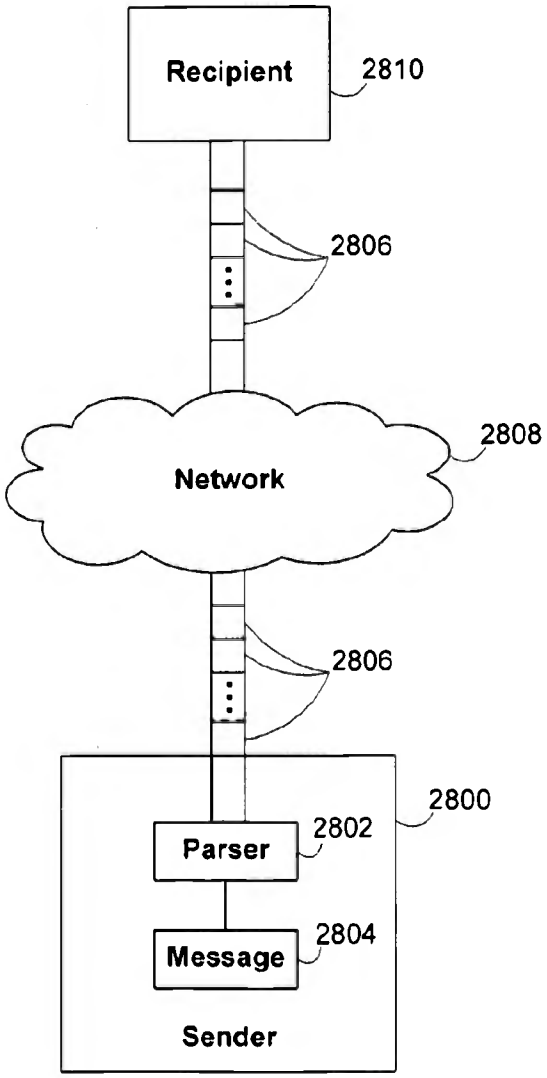

Recipient
2810
2806
2808
Network
2806
2800
Parser
2802
Message
2804
Sender

FIG. 29

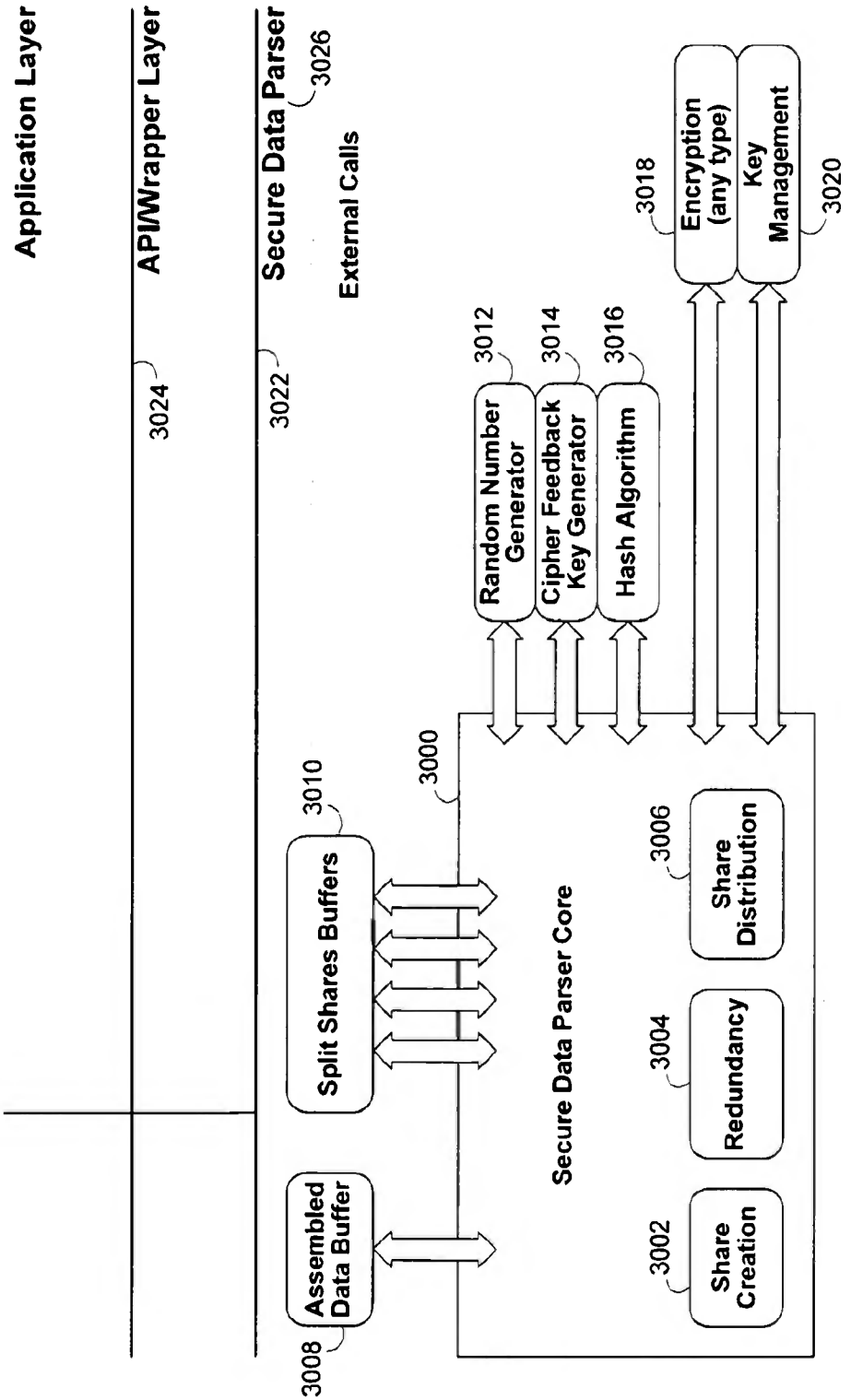

Application Layer
API/Wrapper Layer
Secure Data Parser
3026
External Calls
3024
3022
3012
3014
3016
3018
3020
Random Number Generator
Cipher Feedback Key Generator
Hash Algorithm
Encryption (any type)
Key Management
3000
3010
Split Shares Buffers
Secure Data Parser Core
3006
Share Distribution
3004
Redundancy
3002
Share Creation
Assembled Data Buffer
3008

FIG. 30

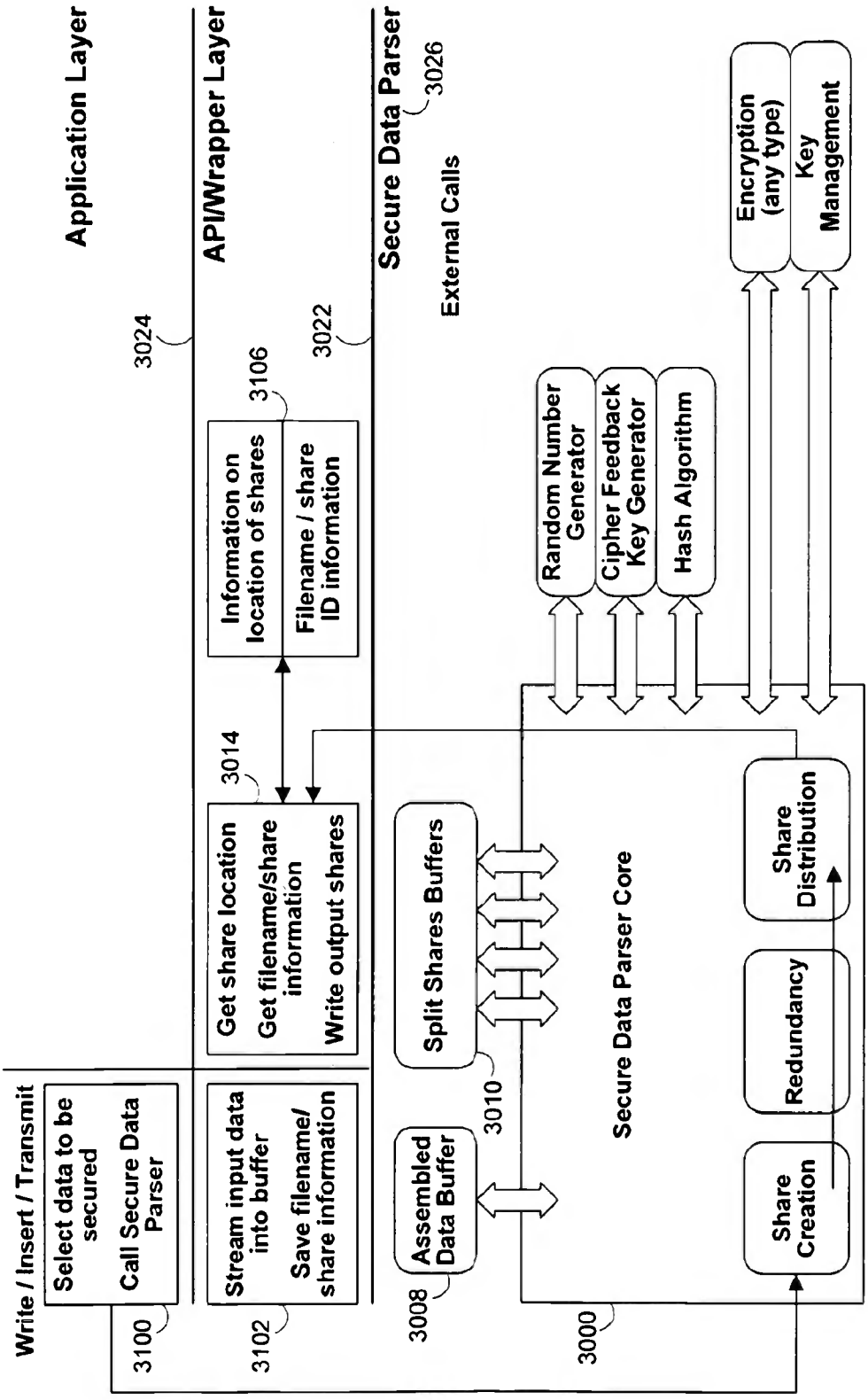

Application Layer
API/Wrapper Layer
Secure Data Parser
3026
External Calls
3024
3022
3106
3014
3100
3102
3008
3010
3000
Write / Insert / Transmit
Select data to be secured
Call Secure Data Parser
Stream input data into buffer
Save filename/ share information
Get share location
Get filename/share information
Write output shares
Information on location of shares
Filename / share ID information
Assembled Data Buffer
Split Shares Buffers
Secure Data Parser Core
Share Creation
Redundancy
Share Distribution
Random Number Generator
Cipher Feedback Key Generator
Hash Algorithm
Encryption (any type)
Key Management

FIG. 31

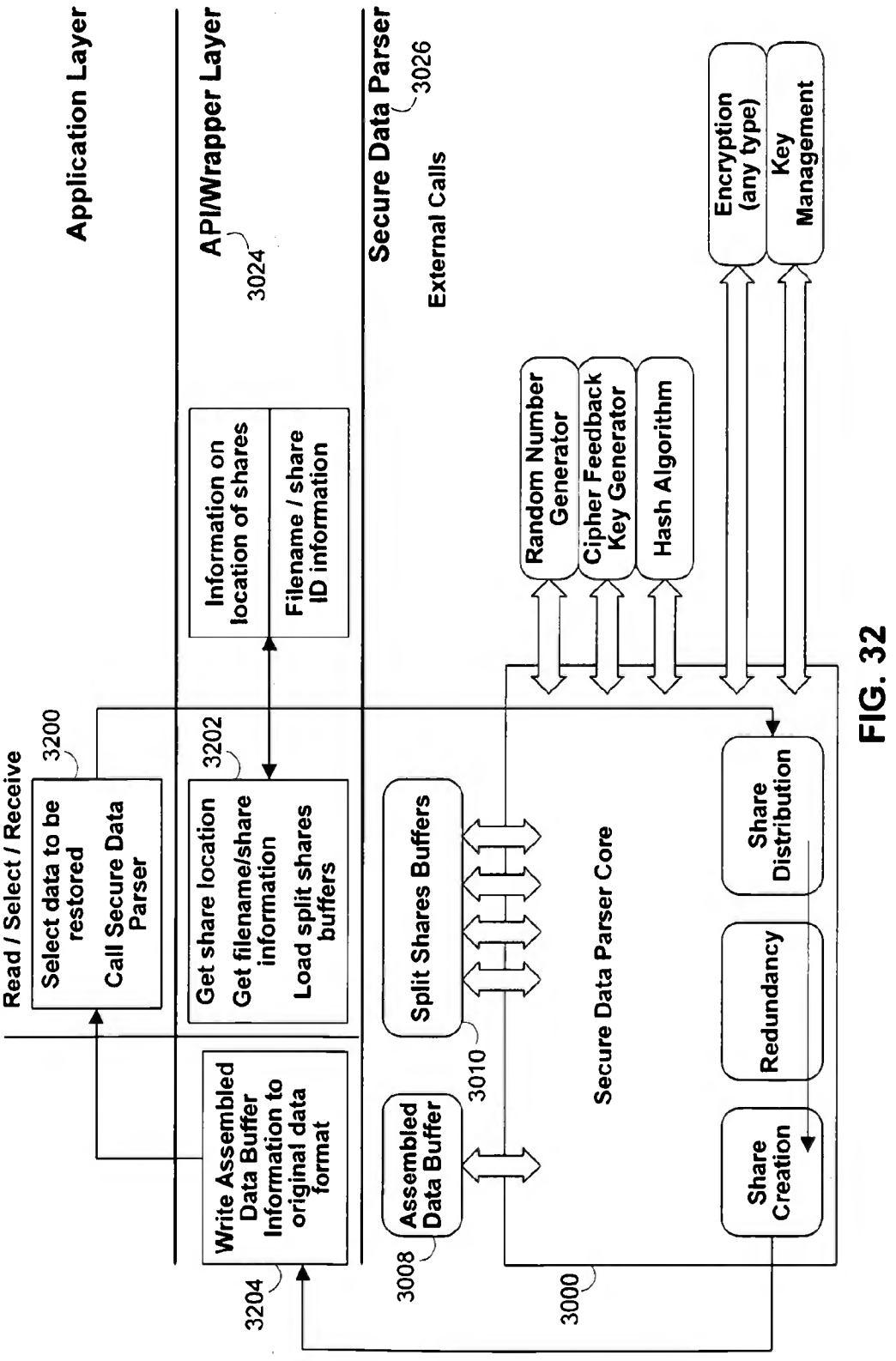

Application Layer
API/Wrapper Layer
3024
Secure Data Parser
3026
External Calls
Read / Select / Receive
3200
Select data to be restored
Call Secure Data Parser
3202
Get share location
Get filename/share information
Load split shares buffers
Information on location of shares
Filename / share ID information
Write Assembled Data Buffer Information to original data format
3204
Split Shares Buffers
3010
Assembled Data Buffer
3008
Secure Data Parser Core
3000
Random Number Generator
Cipher Feedback Key Generator
Hash Algorithm
Encryption (any type)
Key Management
Share Distribution
Redundancy
Share Creation

FIG. 32

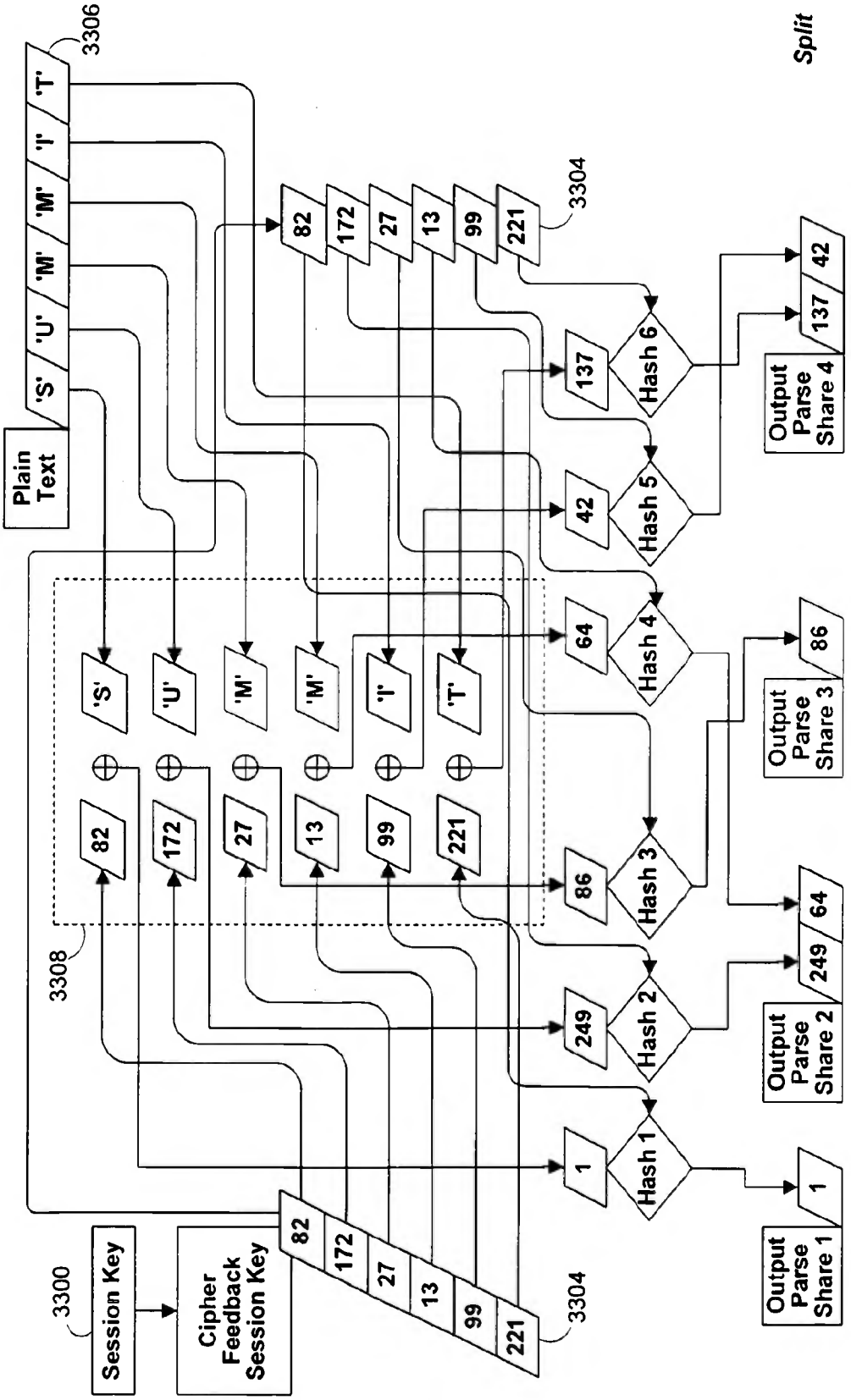

Split
3306
Plain Text
'S' 'U' 'M' 'M' 'I' 'T'
3304
82 172 27 13 99 221
3308
Session Key
3300
Cipher Feedback Session Key
Hash 1
Hash 2
Hash 3
Hash 4
Hash 5
Hash 6
Output Parse Share 1
Output Parse Share 2
Output Parse Share 3
Output Parse Share 4

FIG. 33

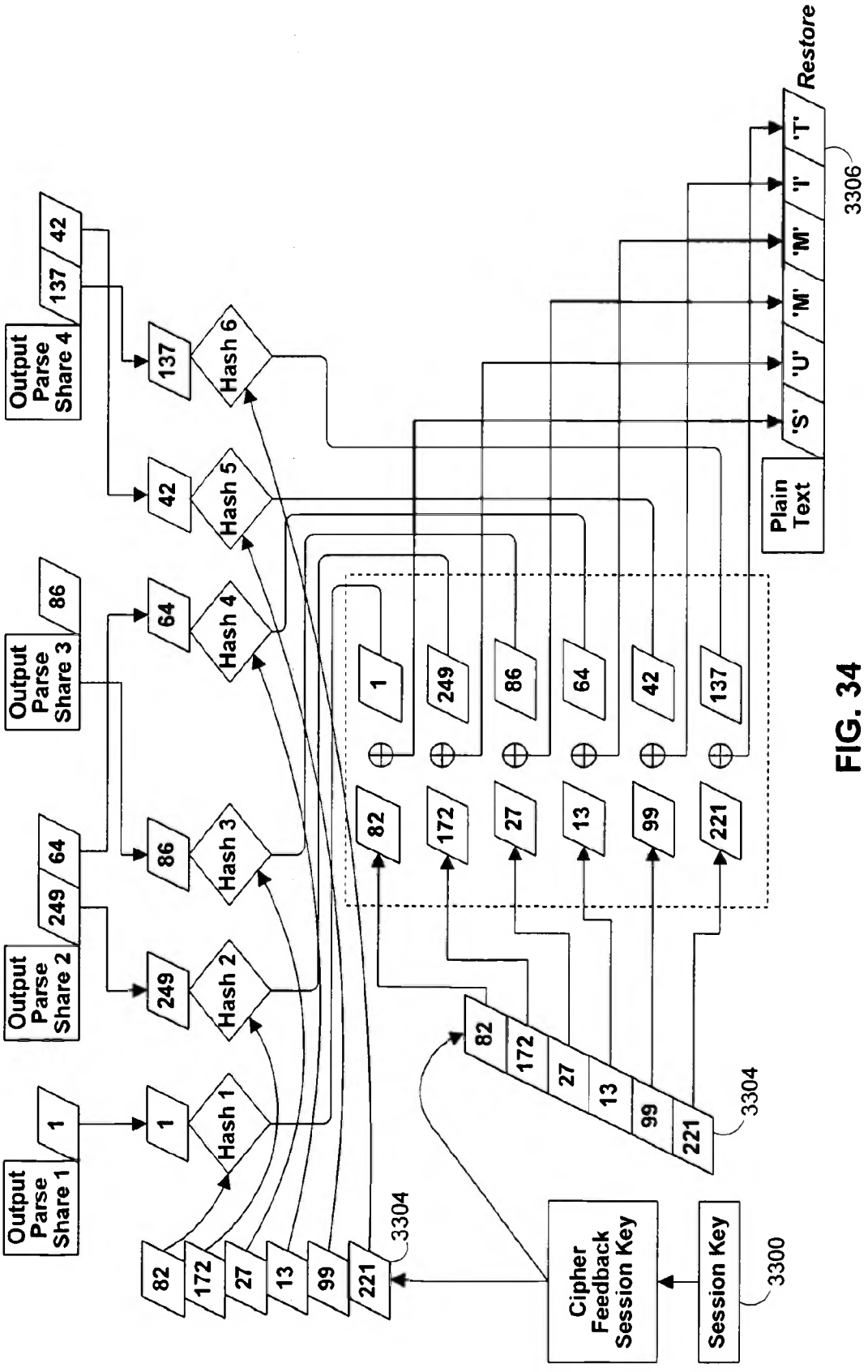

Output Parse Share 1
Output Parse Share 2
Output Parse Share 3
Output Parse Share 4
Hash 1
Hash 2
Hash 3
Hash 4
Hash 5
Hash 6
Cipher Feedback Session Key
Session Key
3300
3304
3306
Plain Text
Restore
'S'
'U'
'M'
'M'
'I'
'T'

FIG. 34

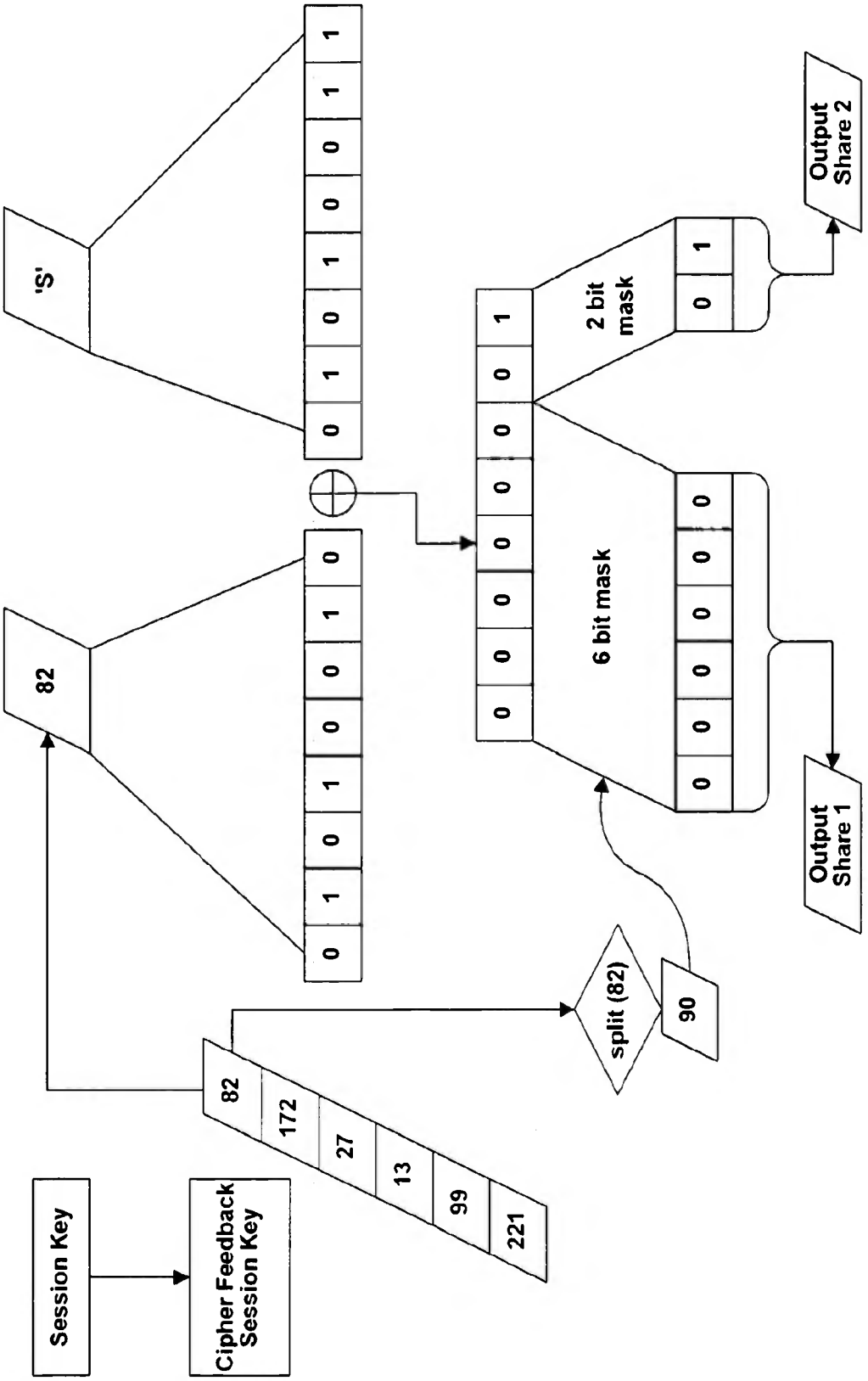

Session Key
Cipher Feedback Session Key
82
172
27
13
99
221
split (82)
90
'S'
82
6 bit mask
2 bit mask
Output Share 1
Output Share 2

FIG. 35

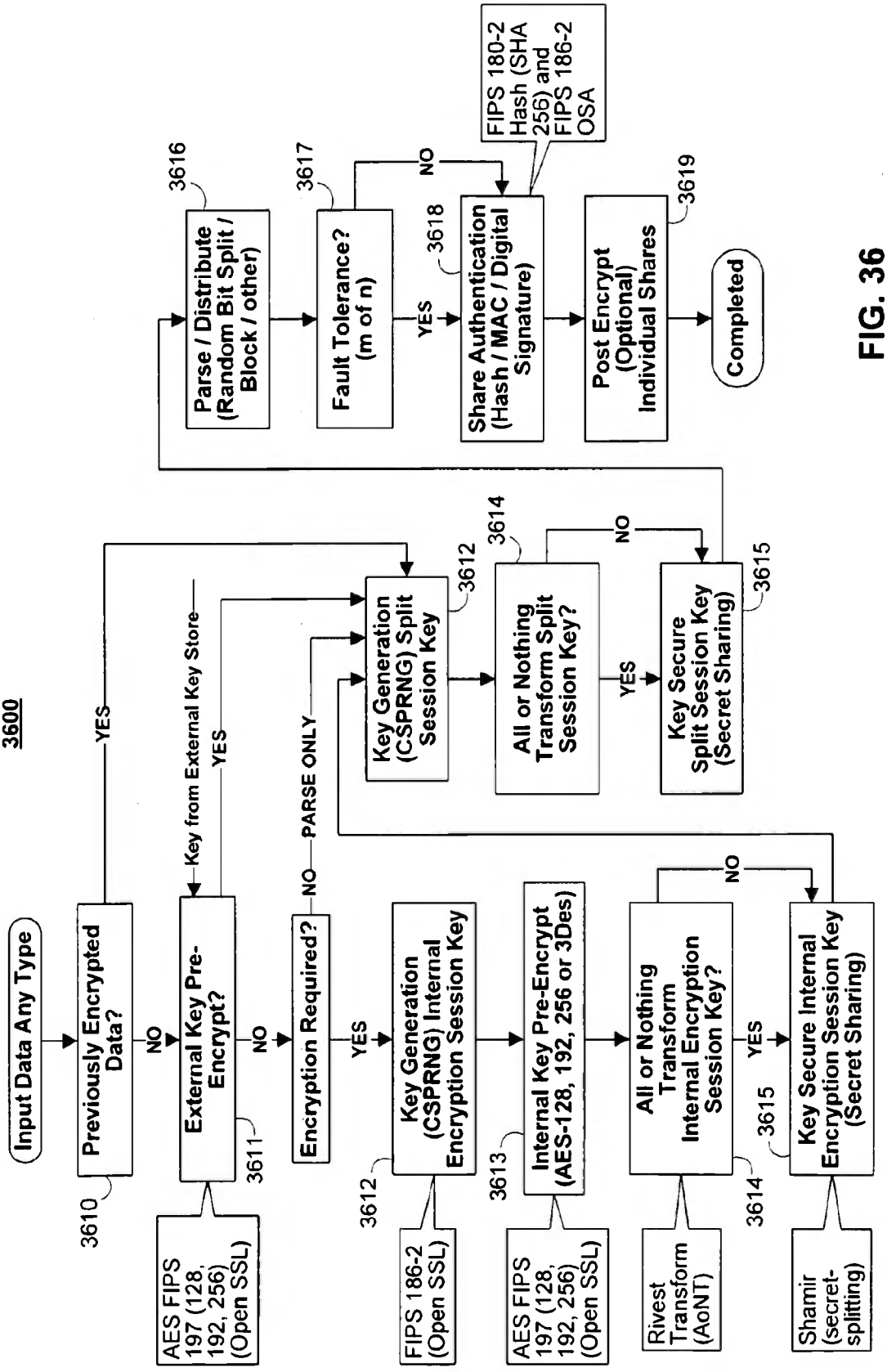

3600
Input Data Any Type
Previously Encrypted Data?
YES
NO
3610
External Key Pre-Encrypt?
Key from External Key Store
YES
NO
3611
AES FIPS 197 (128, 192, 256) (Open SSL)
Encryption Required?
PARSE ONLY
NO
YES
Key Generation (CSPRNG) Split Session Key
3612
FIPS 186-2 (Open SSL)
All or Nothing Transform Split Session Key?
3614
YES
NO
Key Secure Split Session Key (Secret Sharing)
3615
Key Generation (CSPRNG) Internal Encryption Session Key
3612
Internal Key Pre-Encrypt (AES-128, 192, 256 or 3Des)
3613
AES FIPS 197 (128, 192, 256) (Open SSL)
All or Nothing Transform Internal Encryption Session Key?
3614
NO
YES
Rivest Transform (AoNT)
Key Secure Internal Encryption Session Key (Secret Sharing)
3615
Shamir (secret-splitting)
Parse / Distribute (Random Bit Split / Block / other)
3616
Fault Tolerance? (m of n)
3617
YES
NO
3618
FIPS 180-2 Hash (SHA 256) and FIPS 186-2 OSA
Share Authentication (Hash / MAC / Digital Signature)
Post Encrypt (Optional) Individual Shares
3619
Completed

FIG. 36

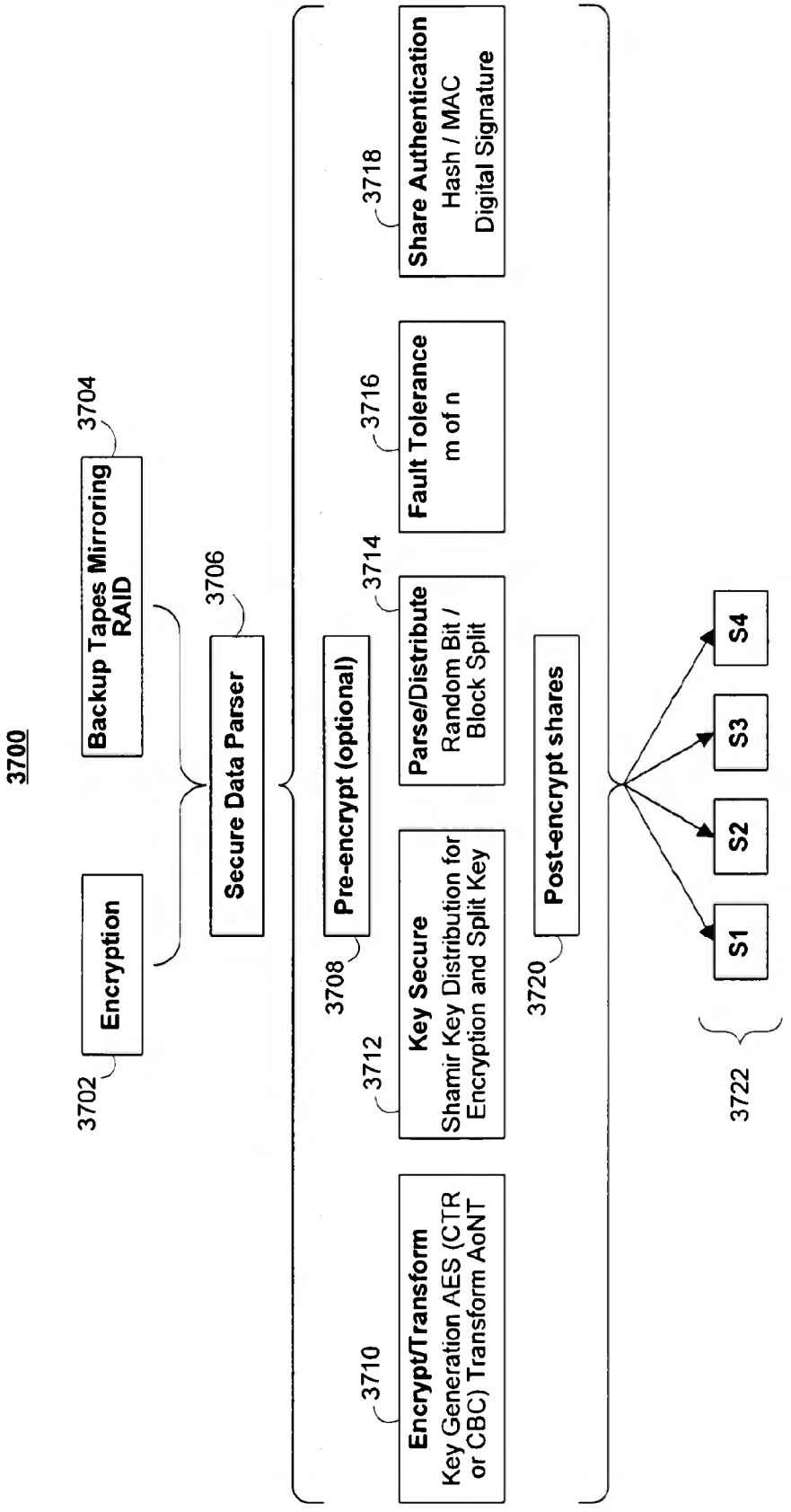

3700
Encryption
3702
Backup Tapes Mirroring RAID
3704
Secure Data Parser
3706
Pre-encrypt (optional)
3708
Key Secure
Shamir Key Distribution for Encryption and Split Key
3712
Encrypt/Transform
Key Generation AES (CTR or CBC) Transform AoNT
3710
Parse/Distribute
Random Bit / Block Split
3714
Fault Tolerance
m of n
3716
Share Authentication
Hash / MAC
Digital Signature
3718
Post-encrypt shares
3720
S1
S2
S3
S4
3722

FIG. 37

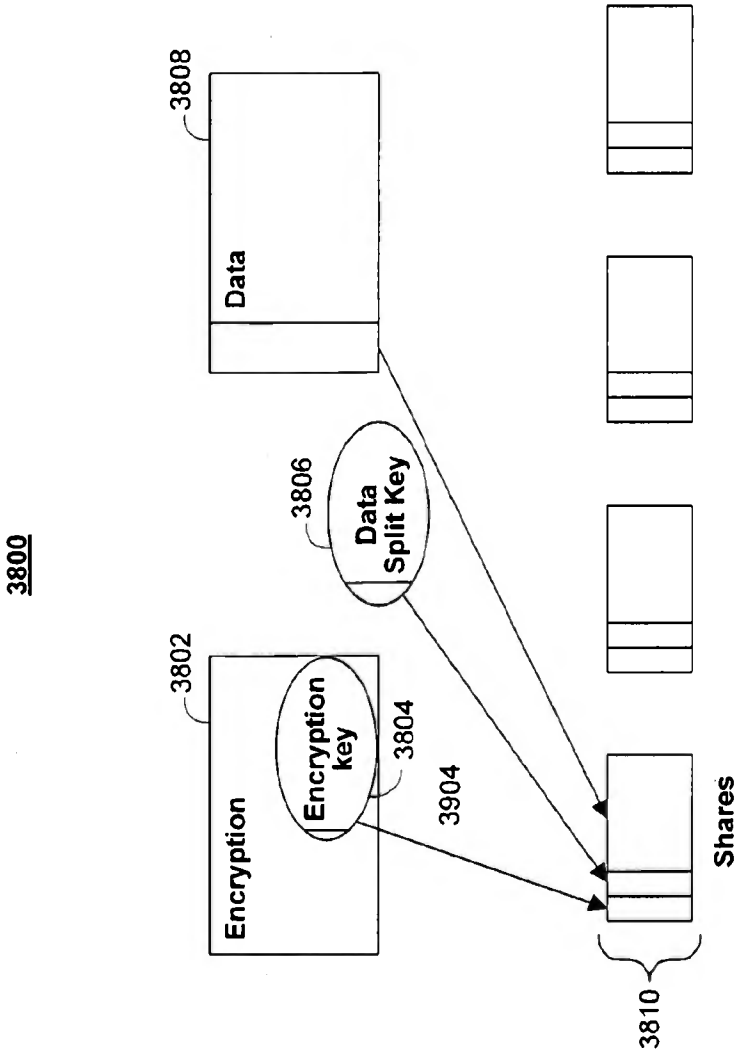

3800
3808
Data
3806
Data Split Key
3802
Encryption
Encryption key
3804
3904
Shares
3810

FIG. 38

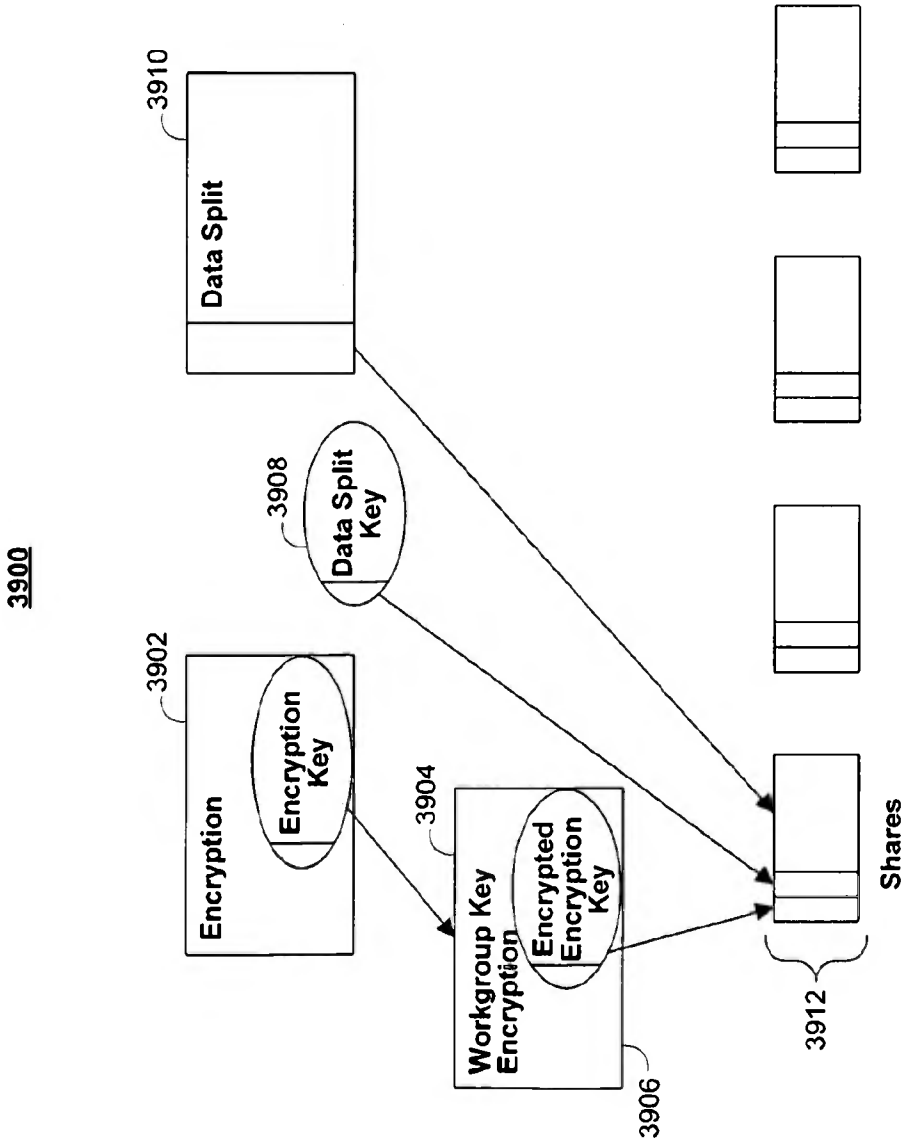

3900
3910
Data Split
3908
Data Split Key
3902
Encryption
Encryption Key
3904
Workgroup Key Encryption
Encrypted Encryption Key
3906
3912
Shares

FIG. 39

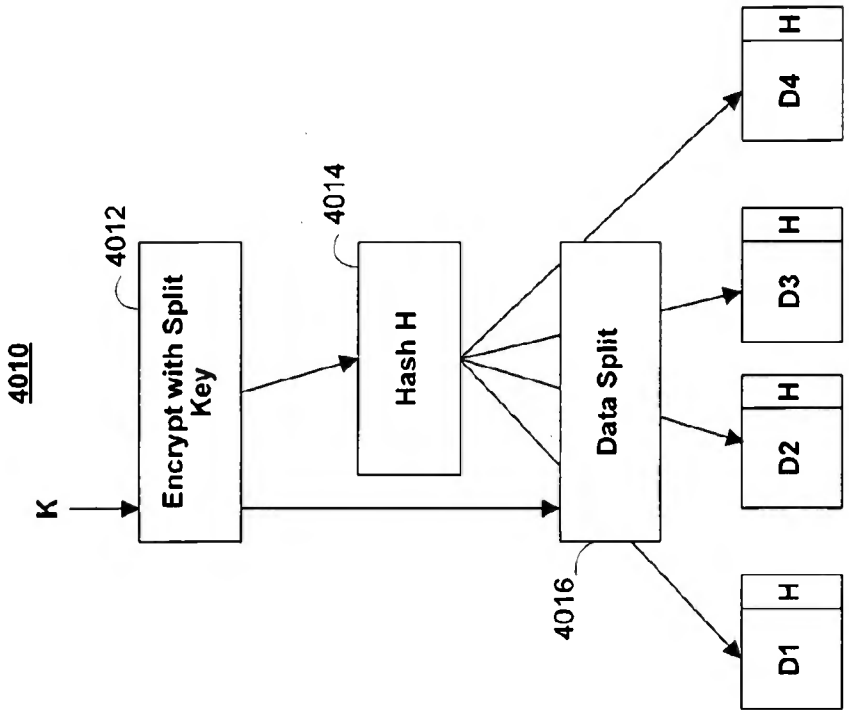
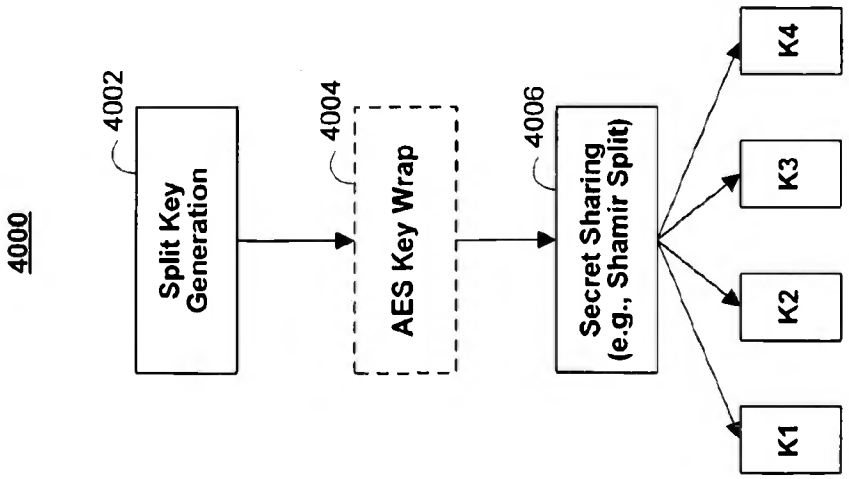

4010
K
4012
Encrypt with Split Key
4014
Hash H
Data Split
4016
D1
H
D2
H
D3
H
D4
H

FIG. 40A


4000
4002
Split Key Generation
4004
AES Key Wrap
4006
Secret Sharing (e.g., Shamir Split)
K1
K2
K3
K4

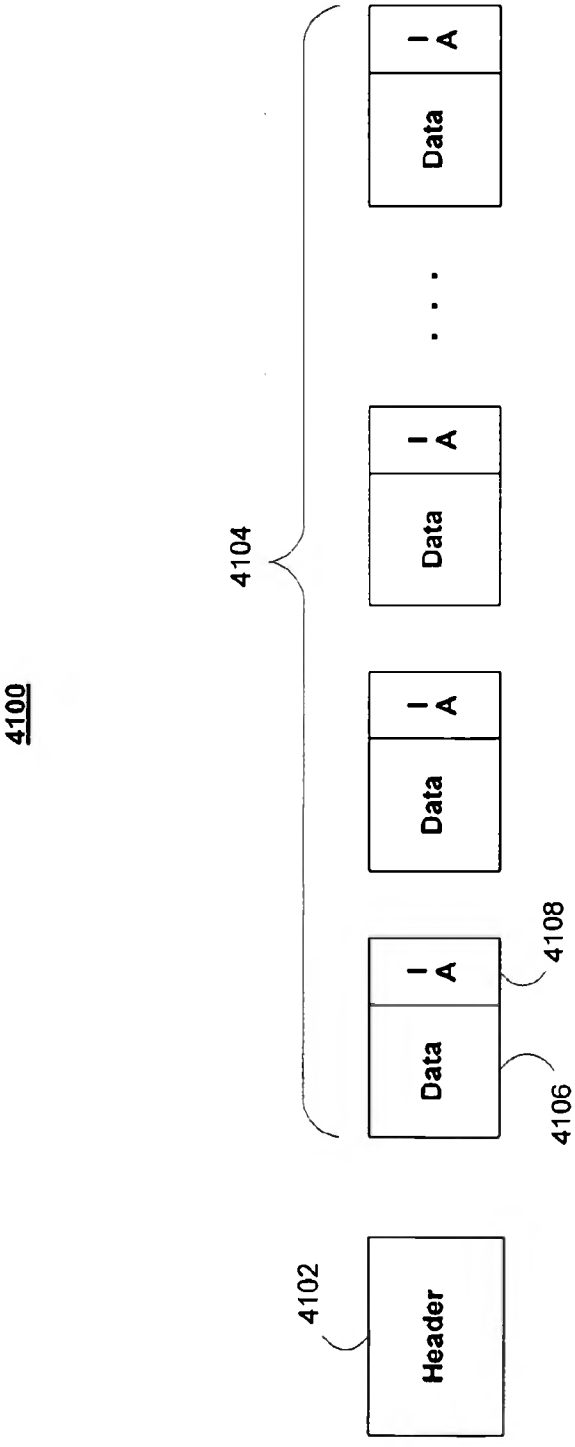


FIG. 41

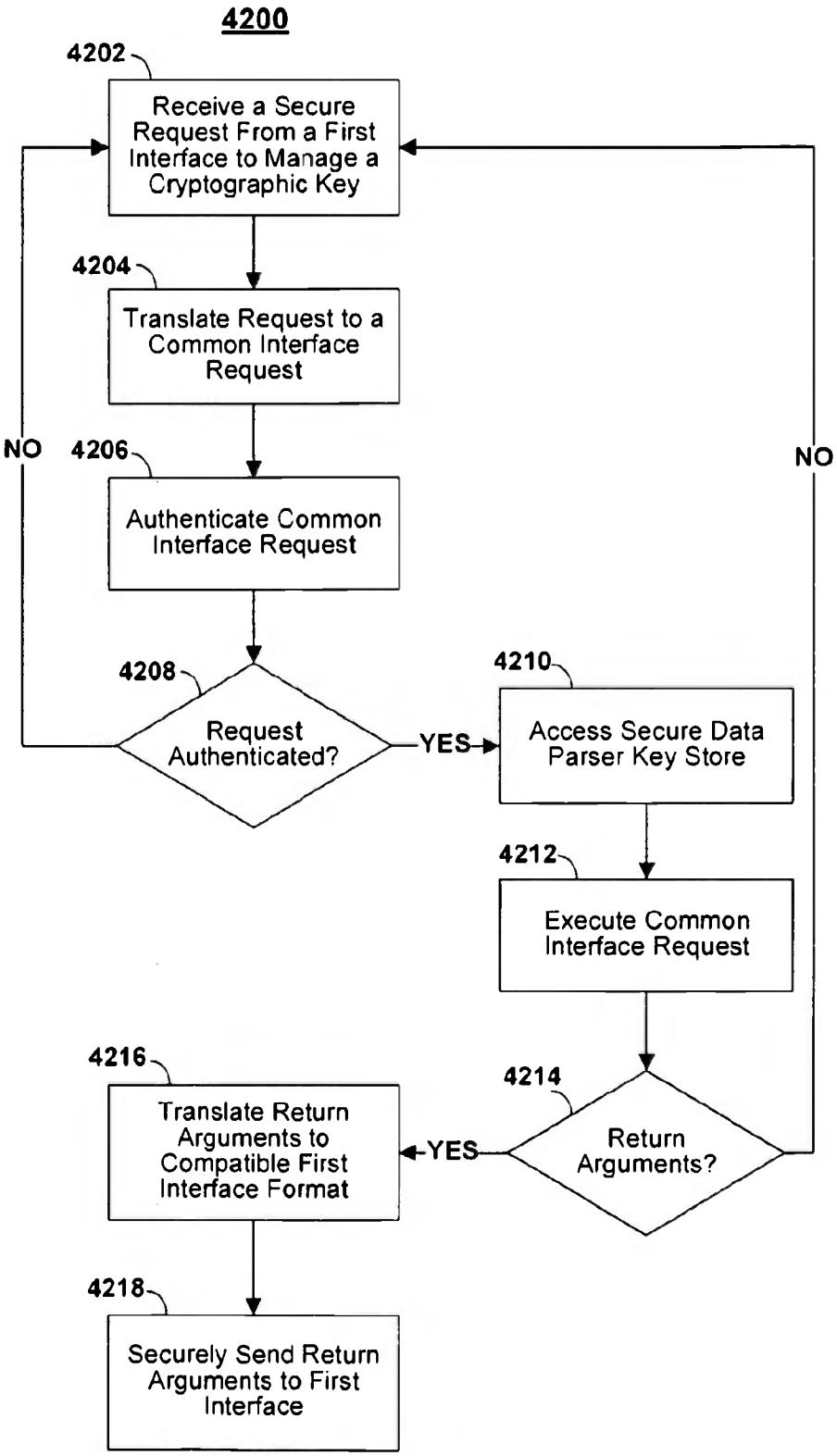

4200
4202
Receive a Secure Request From a First Interface to Manage a Cryptographic Key
4204
Translate Request to a Common Interface Request
NO
4206
Authenticate Common Interface Request
NO
4208
Request Authenticated?
YES
4210
Access Secure Data Parser Key Store
4212
Execute Common Interface Request
4214
Return Arguments?
YES
4216
Translate Return Arguments to Compatible First Interface Format
4218
Securely Send Return Arguments to First Interface

Figure 42

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 08/10677

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - H04L 9/00 (2008.04)
USPC - 713/171, 726/5
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
USPC: 713/171, 726/5

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC: 713/150, 168, 171; 726/2, 3, 4, 5, 26; 705/50, 76, 72, 64

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
PubWEST(USPT,PGPB,EPAB,JPAB); DialogPRO; WIPO, EPO, CITESEER, Google patents, Google scholar Search Terms Used: cryptographic key, first interface format, first format, common interface format, common format, second interface format, second format, format, cryptographic key management, managing cryptographic Key

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2003/0058274 A1 (HILL et al.) 27 March 2003 (27.03.2003) entire document especially abstract, para [0002], para [0004]-[0008], para [0016]-[0020], para [0025], para [0067]-[0069].	1-15
Y	US 2007/0160198 A1 (ORSINI et al.) 12 July 2007 (12.07.2007) entire document especially abstract, para [0151], para [0201], para [0424]-[0425], para [0501], para [0521].	1-15
A	US 2002/0091640 A1 (GUPTA) 11 July 2002 (11.07.2002) entire document	1-15

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 November 2008 (11.11.2008)	21 NOV 2008

Name and mailing address of the ISA/US	Authorized officer:
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (April 2007)

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 December 2000 (14.12.2000)

PCT

(10) International Publication Number
WO 00/76118 A1

(51) International Patent Classification[7]: H04L 9/06, 9/32

(21) International Application Number: PCT/US00/15869

(22) International Filing Date: 8 June 2000 (08.06.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/138,412 8 June 1999 (08.06.1999) US

(71) Applicant: GENERAL INSTRUMENT CORPORATION [US/US]; 101 Tournament Drive, Horsham, PA 19044 (US).

(72) Inventors: SPRUNK, Eric, J.; 6421 Cayenne Lane, Carlsbad, CA 92009 (US). QIU, Xin; 13750 Ruette Le Parc #D, Del Mar, CA 92014 (US).

(74) Agents: KULAS, Charles, J. et al.; Townsend and Townsend and Crew LLP, 8th floor, Two Embarcadero Center, San Francisco, CA 94111 (US).

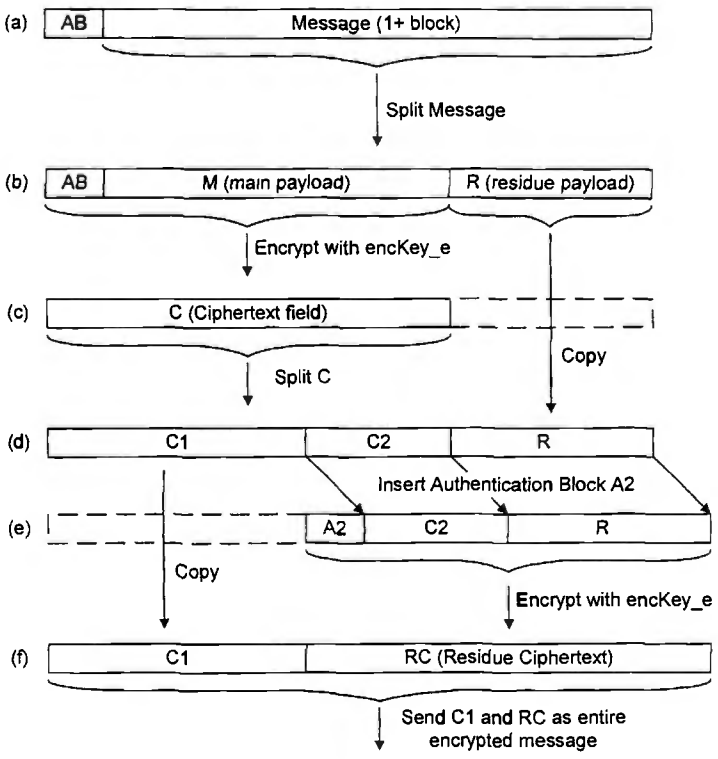
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*

[Continued on next page]

(54) Title: SELF AUTHENTICATION CIPHERTEXT CHAINING



(57) Abstract: Existing key encryption approaches are extended by using overlapping portions of encrypted information. Another provision inserts one or more bits of data to ensure correct encryption/decryption. The inserted data can also be used for authentication.

WO 00/76118 A1

— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SELF AUTHENTICATION CIPHERTEXT CHAINING

CROSS-REFERENCES TO RELATED APPLICATIONS

This application claims priority from U.S. Provisional Patent Application No. 60/138,412, filed June 8, 1999, the disclosure of which is incorporated herein in its entirety by reference for all purposes.

FIELD OF THE INVENTION

The present invention relates to cryptographic systems in general and in particular to a system for encrypting information efficiently using encryption keys having a fixed modulus size.

BACKGROUND OF THE INVENTION

Encryption is the process of converting a message from plaintext to ciphertext in such a way that only those that are authorized readers can decrypt the plaintext from the ciphertext. Often, encryption is used to secure a message that is expected to be transported through an untrusted channel or stored on an insecure data storage medium. The term "message" often refers to a communication between a sender and a receiver but as used here the term refers to any data that might need to be secured between the time and/or place of its creation or acceptance by the sender and the time and/or place of its receipt by the receiver. Thus, a message could be an e-mail communication, a program, a dataset, an image, a collection of data objects treated as a single message, a stream of data, or combinations of the above or similar objects.

One method of determining whether or not the receiver is authorized to read, or otherwise access, the plaintext of the message is the use of "keys". Typically a key is representable by data, such as a string of bits. An example is a 128-bit key, which is a string of 128 bits. Using this method, the sender would use an encryptor to encrypt the plaintext of the message into the ciphertext in such a way that any recipient of the ciphertext, authorized or not, that did not have knowledge of the key could not decrypt the plaintext from the ciphertext without some threshold of computing effort and/or time.

It is well understood that, except for a limited class of encryption schemes such as using one time pads, the plaintext can be extracted from the ciphertext without the key with enough computing effort and/or time. For example, an attacker (i.e., an unauthorized recipient) could attempt to decrypt the message by serially decrypting using

each possible key. However, in a well-designed encryption system, the amount of computing effort needed to decrypt without the key costs more than the value of having decrypted the message or would take so much time that the value of keeping the message secure has passed before the message is decrypted.

There are several aspects of message security that an encryption system provides. One aspect is secrecy, in that the plaintext of a message can be kept from unauthorized readers even if the reader has possession of the ciphertext of the message. Another aspect is authentication, in that the recipient of the ciphertext can verify that the message was actually sent by the purported sender. Yet another aspect is integrity, in that the recipient can verify that the message was not modified after leaving the control of the sender. In some instances, only one aspect is used. For example, a digital signature process creates a data sequence that authenticates a message and that message is often sent "in the clear" so that anyone can read the message. Thus, the message is not kept secret, but it can still be authenticated. Although a system does not always encrypt a message before transport or storage, as is the case for digital signatures, the system is nonetheless generically referred to as an encryption system.

Encryption systems are often classified into private key systems and public key systems, often referred to as symmetric key systems and asymmetric key systems, respectively. In a private key system, the key is used by the sender to encrypt the message and the same key is used by the receiver to decrypt or verify the message. As a result, the key must be kept secret from unauthorized entities. With public key systems, the key is a pair of key parts comprising a public part and a private part. The public part is not necessarily kept secret and can be used to verify messages and perform other processes on a message, but typically the private key is needed to extract plaintext from the ciphertext of a secret message.

One example of a public key standard is the widely used RSA standard. One advantage of using a standard public key system is that many components of the system are readily available, such as e-mail encryptors, key managers, encoders, decoders, verifiers, and the like. However, a problem with many standard encryption systems is that they operate on the message in blocks of fixed sizes per key length, requiring padding when the message to be sent does not fill an integer number of blocks exactly. Random data should be used for padding, to avoid easy attacks on decrypting the message without the key.

The use of fixed size blocks is not a problem where messages are always sized to be an integer number of blocks, but where the messages are not an integer number of blocks, but instead comprise zero or more full blocks and a partial block, the partial block must be padded up to a whole block before processing. Where the amount of processing to encrypt, decrypt or verify a message is a function of the number of blocks and the amount of processing needed for a block is considerable, a processing routine might perform many unnecessary operations on a partial block if the message portion of the partial block is much smaller than the block size.

For example, the block size is often dictated by a key modulus used to encrypt a block. If the key modulus is 512 bits, messages will be encoded in 512 bit blocks. If a message to be encrypted happens to be 1025 bits long, the message would be encrypted into three 512-bit blocks, one of which would represent only one bit of the message.

SUMMARY OF THE INVENTION

In an encryption system according to one embodiment of the present invention, existing fixed key modulus size encryption approaches are extended to use overlapping portions of encrypted information in generating encrypted messages. In another aspect of the invention, the encryption system can insert one or more bits of data to ensure correct encryption/decryption and the inserted data can be used for authentication.

A further understanding of the nature and the advantages of the inventions disclosed herein may be realized by reference to the remaining portions of the specification and the attached drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of an encryption system as might be used to implement an embodiment of the present invention.

Fig. 2 is a flow diagram illustrating a process of encrypting a full block and a residue block of a message.

Fig. 3 is a flow diagram illustrating a variation of the process shown in Fig. 2, without the use of a secondary authentication block.

Fig. 4 is a flow diagram illustrating a process of decrypting a full block and a residue block of a message encrypted as shown in Fig. 2.

DESCRIPTION OF THE SPECIFIC EMBODIMENTS

Fig. 1 is a block diagram of an encryption system 10, wherein plaintext 12 of a message is encrypted by an encryptor 14 using an encryption key to produce an encryptor output 16 that could be ciphertext, verification data (such as a digital signature), or both. Encryptor output 16 is conveyed to a channel and/or storage medium 18, where such channel or storage medium is untrusted and therefore considered insecure. A decryptor 20 receives encryptor output 16 from channel and/or storage medium 18 and, using a decryption key, produces plaintext or authorization indications 22.

In an encryption/decryption operation, the encryptor might encrypt plaintext, which the decryptor would decrypt to obtain the plaintext. In another operation, the encryptor might generate a digital signature that the decryptor could use to verify a message and issue an authenticated/unauthenticated signal.

In a fixed modulus encryption operation, plaintext 12 is divided into blocks of the modulus size and where plaintext 12 is of a size not evenly divisible by the modulus, the encryption of the last full block and the partial remaining block of plaintext 12 is performed as shown in Fig. 2.

Fig. 2 is a series of transformations of the full block and the partial remaining block resulting in the encryption of that data. Fig. 2(a) illustrates a message and an authentication block, AB. AB could represent a number of data elements about the message. In one example, AB is a concatenation of a zero bit or byte (to prevent overflows), a unit identity address (such as a MAC address), a sequence number (to prevent replay attacks), and other data about the message. In the description below, the following variables are used to represent lengths of various elements:

AB_len	length of AB field
Payload_len	length of the data to be encoded (a full block and a partial block)
Enc_len	length of blocks used in encoding process

As shown in Fig. 2(b), the encryptor logically splits the message into a main payload M and a residual payload R, where the split is done so that the length of M, len(M), is such that len(M) + AB_len = Enc_len. With that split, the length of R, is R_len = Payload_len – len(M) = Payload_len - Enc_len + AB_len.

As shown in Fig. 2(c), AB || M (where "||" is a concatenation operator) is encrypted using an encryption key encKey_e to produce a ciphertext field C where len(C) = Enc_len. In one embodiment, the blocks are encoded using an RSA encoding process. For example, C might be (AB || M)^encKey_e mod modulus_n, where values of

Enc_len can be expressed as unique numbers less than modulus_n. As used here, lengths can be in any units, but a common measurement of data length is in bits.

As shown in Fig. 2(d), C is then divided into two fields, C1 and C2, where the lengths of C1 and C2 are such that the following equations are satisfied:

$$len(C2) = Enc\_len - A2\_len - len(R)$$
$$= 2*Enc\_len - A2\_len - Payload\_len - AB\_len$$
$$len(C1) = len(C) - len(C2)$$

where A2_len is the length of a secondary authentication block, A2, shown in Fig. 2(e).

Since the length of A2 || C2 || R is Enc_len, that concatenation can be encrypted using encKey_e, to produce a residue ciphertext, RC. As illustrated by Fig. 2(f), C1 and RC can be provided to a decryptor.

The insertion of a secondary authentication block, such as A2 in Fig. 2(e), is optional. Where A2 is not used, the authentication block AB would authenticate R, due to the overlap, since the cryptographic effects of AB feed through from ciphertext C2, which is combined with R and encrypted as shown by Figs. 2(e)-(f). So long as there are enough bits in C2 for the feedthrough effect to be cryptographically valid (e.g., C2 being 128 bits or more), then the inclusion of A2 in the generation of RC is not needed to authenticate R. However, using A2 would be useful where C2 is too small. Thus, it should be understood that in the figures, len(A2) could range from zero to some positive value. Fig. 3 illustrates the feedthrough effect.

Fig. 4 is a flow diagram illustrating a process of decrypting a full block and a residue block of a message encrypted as shown in Fig. 2. As illustrated by Figs. 4(a)-(b), the received block is split into a C1 portion and an RC portion. The decryptor can properly split its input into C1 and RC knowing only len(C1 || RC) and Enc_len, since len(RC) = Enc_len.

As shown in Figs. 4(c)-(g), RC is decrypted and segmented into A2, C2 and R. The segmentation can be performed if the decryptor knows A2_len and either len(R) or Payload_len, Enc_len, AB_len, from which len(R) can be calculated. If A2 is used and cannot be verified, the message is discarded. Otherwise, the message is parsed into C1 and C2.

Once C1 and C2 are identified, they can be concatenated to form C, which can then be decrypted to produce AB and M. Finally, M and R can be combined to reconstruct the original plaintext message. If AB cannot be verified, the message is discarded. One case where the message is not verified is where the value for a key

sequence is stored in AB and the message has a sequence number lower than, or out of order relative to, a prior received sequence number.

In the process of decrypting (see Fig. 4(c)), the decryptor verifies A2 and discards the message if A2 is other than expected. One cause for A2 being an unexpected value is if the ciphertext message had been altered as it passed from the encryptor to the decryptor. In one embodiment, A2 is simply a null value, such as a "0" bit or a "00" byte. Because of some overlap between C and RC through C2, the authentication of R can be done by just verifying that A2 is as expected and AB is as expected. In effect, this allows both blocks to be authenticated using just one authentication block, AB, resulting in bandwidth and processing savings. In operation, the strength of AB for authenticating the partial block is related to the amount of overlap, i.e., len(C2). The overlap, len(C2), should preferably be at least 128 bits.

One use of the system described above is for securely passing keys to a remote security chip that is only accessible over an untrusted channel.

Although the invention has been described with reference to particular embodiments thereof, these embodiments are merely illustrative, and not limiting, of the present invention, the scope of which is to be determined solely by the appended claims.

WHAT IS CLAIMED IS:

1. A method for encrypting information using encryption keys, wherein each key encrypts a portion of information of a predetermined block length, the method comprising using a first key to encrypt a first portion of a message;

adding at least one bit of information to the encrypted first portion of the message;

using a second key to encrypt a second portion of the message wherein the second portion overlaps with the first portion and also includes the added one or more bits of information.

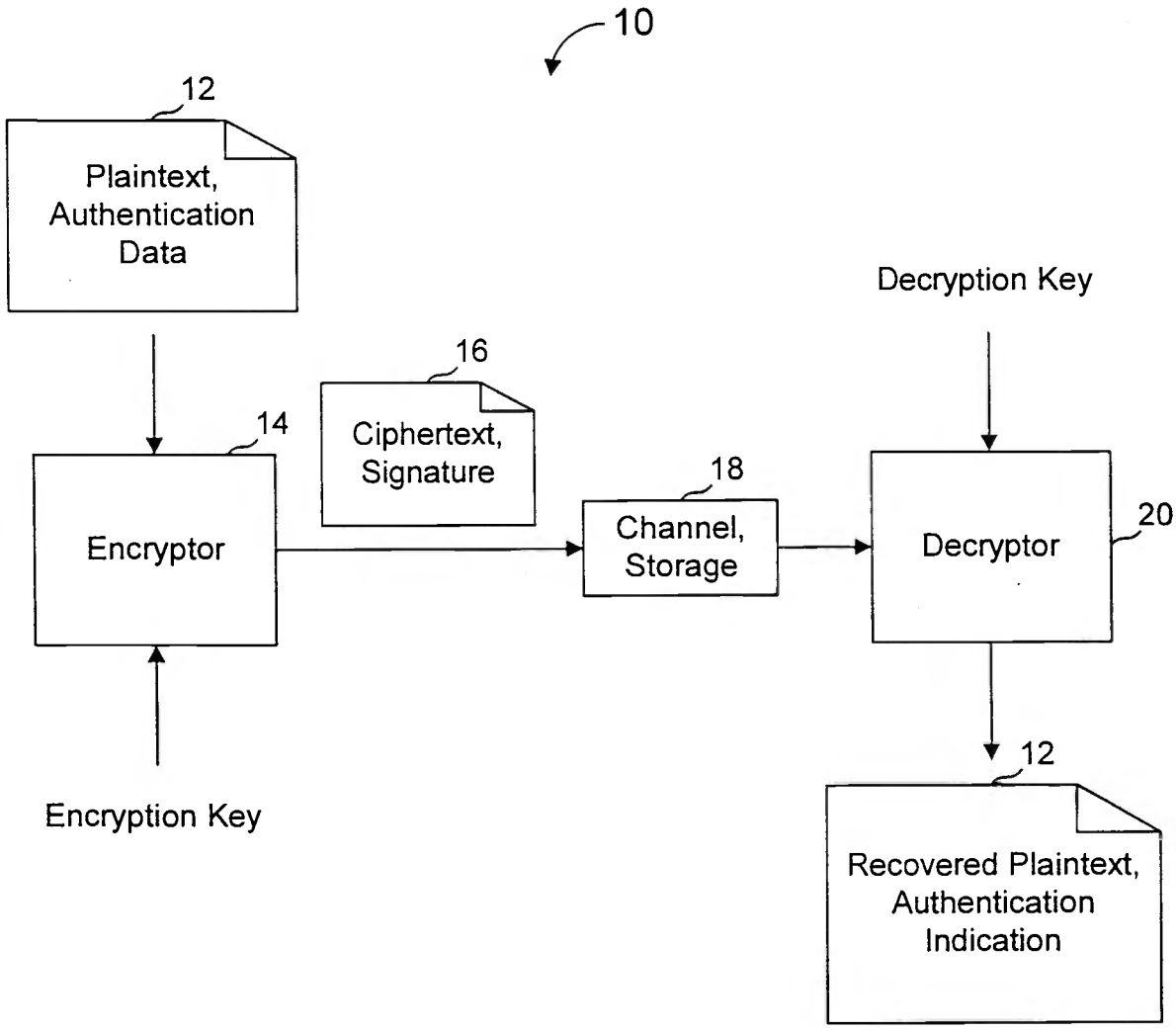

10
12
Plaintext, Authentication Data
Decryption Key
16
Ciphertext, Signature
14
18
20
Encryptor
Channel, Storage
Decryptor
12
Encryption Key
Recovered Plaintext, Authentication Indication

Fig. 1

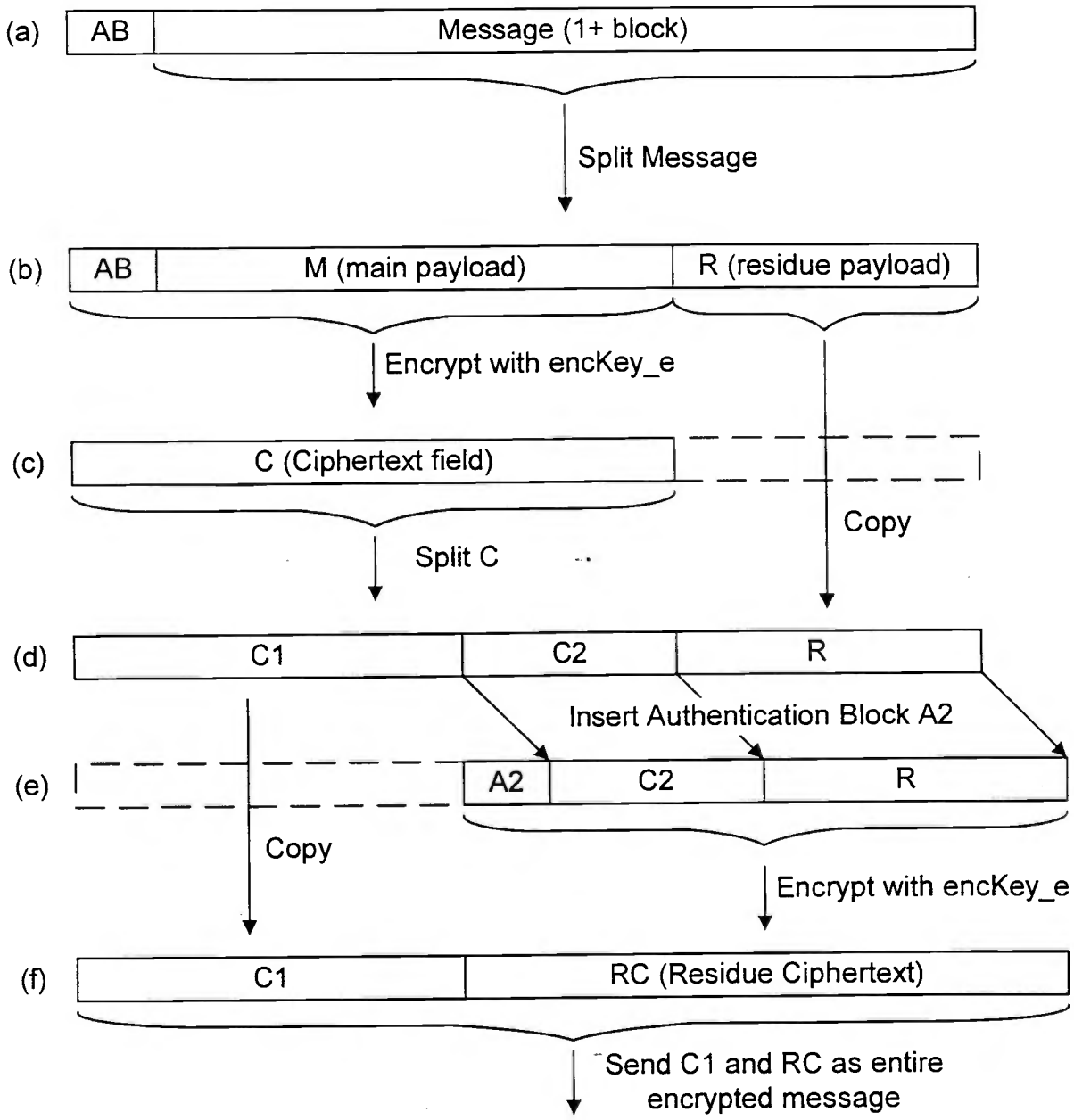

(a)
AB
Message (1+ block)
Split Message
(b)
AB
M (main payload)
R (residue payload)
Encrypt with encKey_e
(c)
C (Ciphertext field)
Copy
Split C
(d)
C1
C2
R
Insert Authentication Block A2
(e)
A2
C2
R
Copy
Encrypt with encKey_e
(f)
C1
RC (Residue Ciphertext)
Send C1 and RC as entire encrypted message

Fig. 2

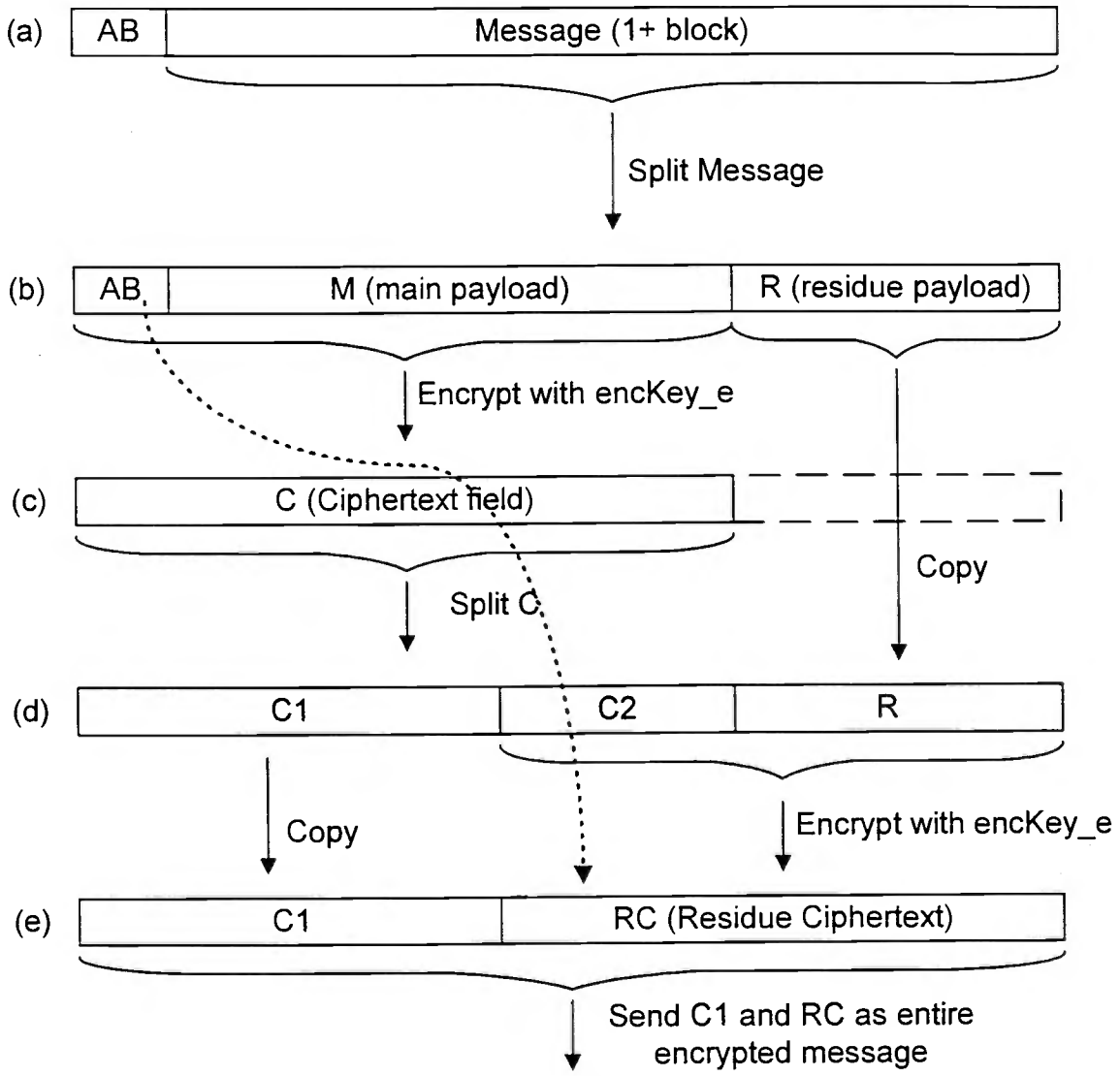

(a)
AB
Message (1+ block)
Split Message
(b)
AB
M (main payload)
R (residue payload)
Encrypt with encKey_e
(c)
C (Ciphertext field)
Copy
Split C
(d)
C1
C2
R
Copy
Encrypt with encKey_e
(e)
C1
RC (Residue Ciphertext)
Send C1 and RC as entire encrypted message

Fig. 3

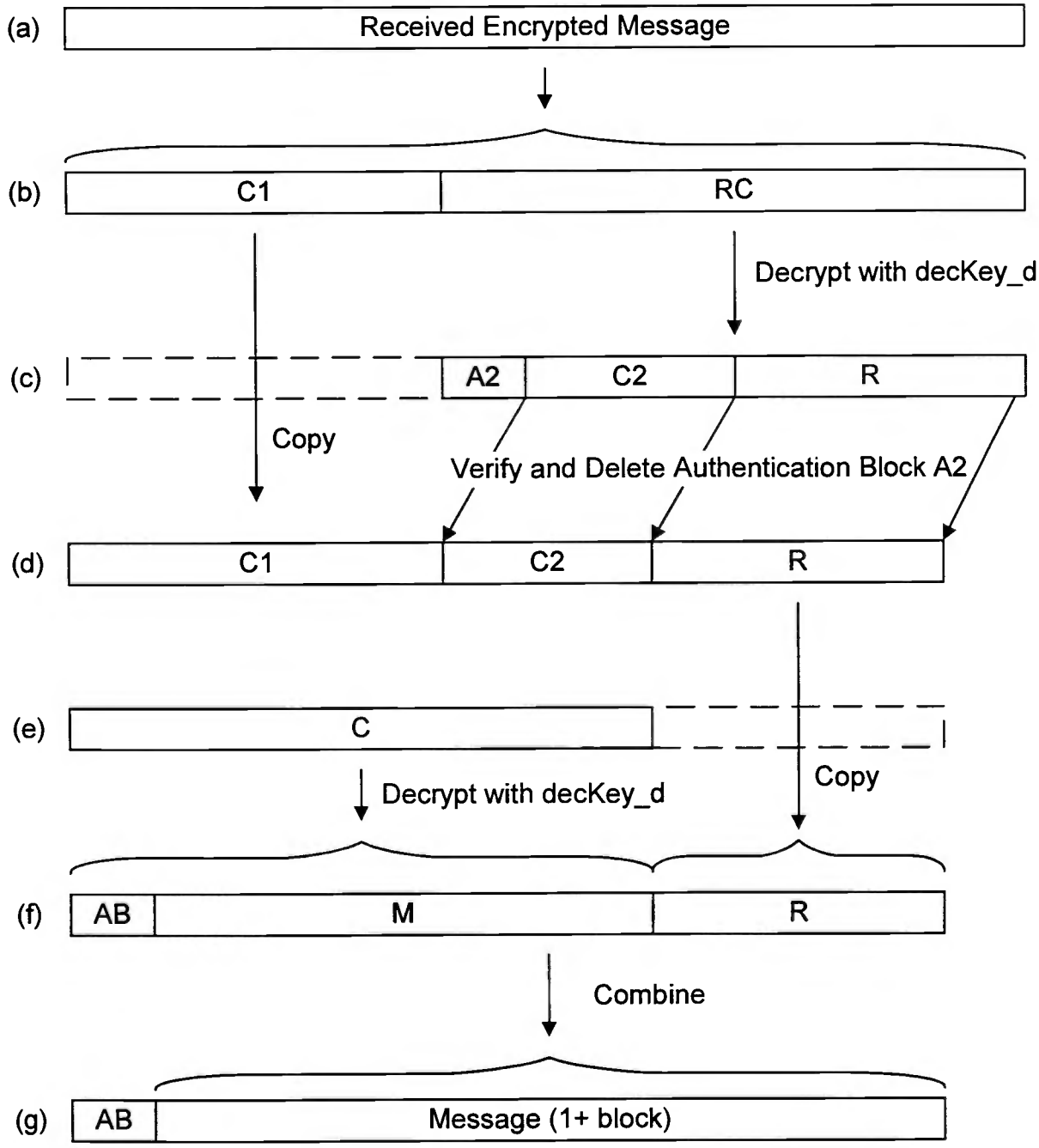

(a)
Received Encrypted Message
(b)
C1
RC
Decrypt with decKey_d
(c)
A2
C2
R
Copy
Verify and Delete Authentication Block A2
(d)
C1
C2
R
(e)
C
Copy
Decrypt with decKey_d
(f)
AB
M
R
Combine
(g)
AB
Message (1+ block)

Fig. 4

INTERNATIONAL SEARCH REPORT

Interr nal Application No
PCT/US 00/15869

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L9/06 H04L9/32

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, INSPEC, PAJ, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	FEISTEL H ET AL: "Some cryptographic techniques for machine-to-machine data communications" PROCEEDINGS OF THE IEEE, NOV. 1975, USA, vol. 63, no. 11, pages 1545-1554, XP002150274 ISSN: 0018-9219 page 1550, left-hand column, last paragraph -page 1551, left-hand column, line 6 ---	1
Y	US 5 495 533 A (LINEHAN MARK H ET AL) 27 February 1996 (1996-02-27) column 8, line 57 -column 9, line 2 ---	1
A	US 5 412 728 A (BESNARD CHRISTIAN ET AL) 2 May 1995 (1995-05-02) column 3, line 61 -column 5, line 7 ---	1
	-/--	

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
17 October 2000	03/11/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Holper, G

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Interr nal Application No
PCT/US 00/15869

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	EHRSAM W F ET AL: "FRACTIONAL BLOCK-SIZE CIPHER" IBM TECHNICAL DISCLOSURE BULLETIN,US,IBM CORP. NEW YORK, vol. 22, no. 2, 1 June 1979 (1979-06-01), pages 640-642, XP002051943 ISSN: 0018-8689 the whole document -----	1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

...rmation on patent family members

Interr nal Application No
PCT/US 00/15869

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5495533 A	27-02-1996	NONE	
US 5412728 A	02-05-1995	FR 2677200 A	04-12-1992
		AU 1883192 A	08-01-1993
		CA 2110378 A	10-12-1992
		EP 0586472 A	16-03-1994
		WO 9222159 A	10-12-1992

Form PCT/ISA/210 (patent family annex) (July 1992)

PTO/SB/08a (01-08)
Approved for use through 04/31/2009. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 1

Complete if Known	
Application Number	11/258,839 (Conf. No. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2432
Examiner Name	Samson B. Lemma
Attorney Docket Number	SFC-2

U.S. PATENT PUBLICATIONS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		2003/0051159 A2	03-13-2003	McGown et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3] "Number[4] "Kind Code[5] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]
		WO 2009/035674 A1	03-19-2009	Security First Corporation		
		WO 00/76118	12-14-2000	Gen Instrument Corp.		
		WO 00/36786	06-22-2000	Philips Electronics		

NON PATENT LITERATURE DOCUMENTS

Examiner Initials	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
		Menezes "Handbook of Applied Cryptography" 1997, CRC Press LLC, XP001525007	
		Brainard "A New Approach for Authentication with Short Secrets" RSA Laboratories, Bedford, MA (13 pages)	

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Electronic Acknowledgement Receipt

EFS ID:	5558157
Application Number:	11258839
International Application Number:	
Confirmation Number:	2420
Title of Invention:	Secure data parser method and system
First Named Inventor/Applicant Name:	Rick L. Orsini
Customer Number:	01473
Filer:	Brian Edward Mack
Filer Authorized By:	
Attorney Docket Number:	SFC-2
Receipt Date:	22-JUN-2009
Filing Date:	25-OCT-2005
Time Stamp:	13:00:14
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	SUPP-IDS-LTR.pdf	11544 0b12fe7d578eef4ea21697af052132a52fa32f37	no	2

Warnings:	
Information:	

2	Information Disclosure Statement (IDS) Filed (SB/08)	SB-08.pdf	86286 fec172cdd0009455d8df62e386bf81c5a2b430e5	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	NPL Documents	Menezes_Handbook_of_applied_cryptography.pdf	3787084 b82e232d8c8549531866011175cb24f441a833fc3	no	61
Warnings:					
Information:					
4	Foreign Reference	WO2000036786.pdf	887536 01c10f2914dc07d8765e91d2ef8ec6fd03e12cbd	no	25
Warnings:					
Information:					
5	Foreign Reference	WO2009035674A1.pdf	9001438 79e476963f023048008b52f209f6b5973abb2d3a	no	242
Warnings:					
Information:					
6	NPL Documents	BRAINARD_A_New_Two_Server_Approach.pdf	161269 cb907c0068c26959292dea57f95daee1becb1138	no	13
Warnings:					
Information:					
7	Foreign Reference	WO2000076118.pdf	512900 6893391640b2f47fe2af4b0bd7e950e1d32441c2	no	16
Warnings:					
Information:					
		Total Files Size (in bytes):	14448057		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : H04L 9/06	A1	(11) International Publication Number: **WO 00/36786**
		(43) International Publication Date: 22 June 2000 (22.06.00)

(21) International Application Number: PCT/US99/28288

(22) International Filing Date: 30 November 1999 (30.11.99)

(30) Priority Data:
09/212,151 15 December 1998 (15.12.98) US

(71) Applicant: PHILIPS ELECTRONICS NORTH AMERICA CORPORATION [US/US]; 1251 Avenue Of The Americas, New York, Ny 10020 (US).

(72) Inventors: GOFF, Lonnie, C.; 1433 S. Mill Avenue, Tempe, AZ 95281 (US). CORNELIUS, Steven, E.; 224 West Smoke Tree Road, Gilbert, AZ 85233 (US).

(74) Agent: WELLER, Douglas, L.; 431 Magnolia Lane, Santa Clara, CA 95051–5637 (US).

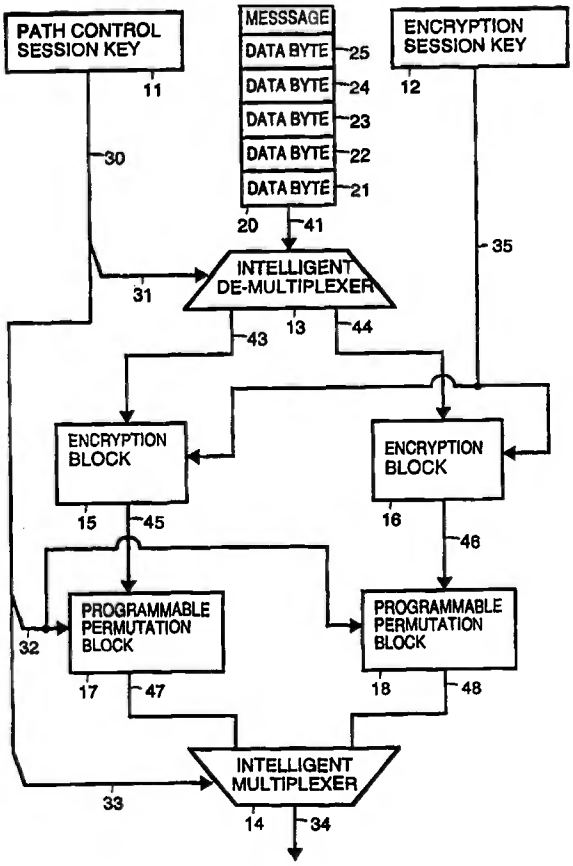
(81) Designated States: CN, JP, KP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: CRYPTOGRAPHIC DEVICE WITH ENCRYPTION BLOCKS CONNECTED IN PARALLEL

(57) Abstract

A cryptographic device includes a de–multiplexer, a plurality of encryption blocks, a plurality of permutation blocks, and a multiplexer. The encryption blocks encrypt data to produce encrypted data. The de–multiplexer receives data portion from a plaintext message and directs the data portions to one of the encryption blocks, based on a value within a path control session key. Each permutation block is associated with an encryption block. Each permutation block permutes encrypted data from the encryption block associated therewith. The multiplexer receives data portions from each of the plurality of permutation blocks to produce an encrypted output data stream.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

CRYPTOGRAPHIC DEVICE WITH ENCRYPTION BLOCKS CONNECTED IN PARALLEL

TECHNICAL FIELD

The present invention concerns encryption/decryption of data and pertains particularly to the parallel connection of encryption blocks to enhance security and performance of cryptographic devices.

BACKGROUND

In order to protect against theft or misuse, secure information within a computing system can be encrypted before being stored within the computing system and/or transmitted to another computing system. Before a computing system uses encrypted information, the encrypted information is decrypted. For example, encryption and decryption is often performed in accordance with the Data Encryption Standard (DES). See for example, *Data Encryption Standard (DES)*, Federal Information Processing Standards Publication (FIPS PUB) 46-2, December 30, 1993 available from the U.S. Department of Commerce, Technology Administration, National Institute of Standards and Technology. See also *DES Modes of Operation*, Federal Information Processing Standards Publication (FIPS PUB) 81, December 2, 1980 available from the U.S. Department of Commerce, National Bureau of Standards.

There are various techniques used to attack security systems. For example, in a brute-force ciphertext-only attacks, all possible combinations of keys can be tried in order to discover the plain text. Depending on the length of a secure key and other factors, an eavesdropper can, in some cases, successfully perform a DES brute-force ciphertext-only attack in a matter of hours.

Plaintext attacks (e.g. Differential Cryptanalysis) can be more efficient than brute-force attacks. In plaintext attacks, the correlation between plaintext and its resultant ciphertext is used in more clever ways than brute force attacks in order to discover the sender's key. Depending on the length of a secure key and other factors, an eavesdropper can often successfully perform some form of intelligent plaintext DES attack in 3 to 15 minutes.

It is desirable, therefore, to increase the security of information in order to make the information less vulnerable to discovery by an unauthorized user.

SUMMARY OF THE INVENTION

In accordance with the preferred embodiment of the present invention, a cryptographic device includes a de-multiplexer, a plurality of encryption blocks, a plurality of permutation blocks, and a multiplexer. The encryption blocks encrypt data to produce encrypted data. The de-multiplexer receives data portions from a plaintext message and directs the data portions to one of the encryption blocks, based on a value within a path control session key. Each permutation block is associated with an encryption block. Each permutation block permutes encrypted data from the encryption block associated therewith. The multiplexer receives data portions from each of the plurality of permutation blocks to produce an encrypted output data stream.

For example, each data portion directed by the de-multiplexer means is one byte in length. Each encryption block in the plurality of encryption blocks encrypt data eight bytes at a time.

In the preferred embodiment, two separate keys are used. An encryption session key is used to perform encryption. Different portions of the path control session key are used to control the de-multiplexer, the multiplexer and the permutation blocks.

The present invention offers significant additional protection against attacks on secrecy. Brute-force ciphertext-only attacks are greatly hindered. In a simple two encryption block system there are 12,870 different ways that the eight byte block output from one encryption block can appear in a sixteen byte transmission (i.e. combined with the eight byte output from the other encryption block). Since there are 40,320 different ways (i.e. permutations) that these eight bytes can be arranged there are, therefore, approximately 520 million combinations of ciphertext associated with a single block transmission. If an eavesdropper can successfully perform a decryption

brute-force ciphertext-only attack in 3.5 hours on a single encryption block, when two encryption blocks are connected in parallel, and permutation is performed on each encryption block, it will take almost 208,000 years to perform the same attack (3.5 hours x 518,918,400 combinations = 207,900 years).

Plaintext attacks on this system also get hindered by embodiments of the present invention. Since there are plaintext attacks (e.g. Differential Cryptanalysis) which can be more efficient than brute-force attacks, the correlation between plaintext and ciphertext is further obscured by adding the de-multiplexing means to the input of the encryption blocks. If an eavesdropper can successfully perform some form of intelligent plaintext encryption attack in 3 to 15 minutes on a single encryption block, when two encryption blocks are connected in parallel, and permutation is performed on each encryption block, it will take approximately 38 million years to perform the same attack (3 minutes x 12870 x 518,918,400 = 38,332,900 years). Clearly, when the present invention is utilized, plaintext attacks become less efficient than brute force ciphertext-only attacks.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a simplified block diagram of a cryptographic device in which encryption of data and permutation of encrypted data is performed in parallel data streams for greater performance and for greater security in accordance with the preferred embodiment of the present invention.

Figure 2 is a simplified block diagram which shows a preferred embodiment of the present invention in which four parallel data streams are used for encryption of data and permutation of encrypted data.

DISCLOSURE OF THE INVENTION

Figure 1 is a simplified block diagram illustrating techniques which are used to combine multiple encryption blocks for greater performance and for greater security. The techniques add to the cryptographic complexity of the system. In the preferred embodiment, the encryption blocks perform encryption/decryption in accordance with the Data Encryption Standard (DES) algorithm. However, in alternative embodiments of the present invention, other encryption algorithms may be used.

A plaintext message 20 to be encrypted includes a number of bytes, as represented by a data byte 21, a data byte 22, a data byte 23, a data byte 24 and a data byte 25. By way of an eight bit data path 41, plaintext message 20 is input into an intelligent de-multiplexer 13, one byte at a time.

A path control session key 11 is, for example 56 bits. Path control session key 11 is placed on a 56 bit data path 30. Sixteen bits of path control session key 11 are forwarded via sixteen bit data path 31 to intelligent de-multiplexer 13 to be used as a sixteen bit de-multiplexer key. The sixteen bit de-multiplexer key is used to steer each of sixteen bytes of plaintext message 20 to either an encryption block 15 via a data path 43 or an encryption block 16 via a data path 44. The first bit of the sixteen bit de-multiplexer key is used to direct the first incoming byte. If the first bit of the sixteen bit de-multiplexer key is a zero, the first byte of plaintext message 20 is passed by intelligent de-multiplexer 13 to encryption block 15. If the first bit of the sixteen bit de-multiplexer key is a one, the first byte of plaintext message 20 is passed to encryption block 16. The second bit of the sixteen bit de-multiplexer key is used to direct the second incoming byte. If the second bit of the sixteen bit de-multiplexer key is a zero, the second byte of plaintext message 20 is passed by intelligent de-multiplexer 13 to encryption block 15. If the second bit of the sixteen bit de-multiplexer key is a one, the second byte of plaintext message

20 is passed by intelligent de-multiplexer 13 to encryption block 16. And so on.

This pattern continues until encryption block 15 and encryption block 16 have each received eight bytes of the sixteen bytes of plaintext message 20. In the preferred embodiment of the present invention, the sixteen bit de-multiplexer key is edited, if necessary, so that there are an equal number of "1"s and "0"s guaranteeing that encryption block 15 and encryption block 16 each will receive eight bytes of the sixteen bytes of plaintext message 20. There are 12,870 possible ways that the sixteen bytes of plaintext message 20 can be segregated into the two byte streams.

Encryption block 15 encrypts the received eight bytes of data. The encryption is performed using an encryption session key 12 received by encryption block 15 via a data path 35. The encrypted eight bytes of data is forwarded to a programmable permutation block 17 via one byte data path 45.

Likewise, encryption block 16 encrypts the received eight bytes of data. The encryption is performed using encryption session key 12 received by encryption block 16 via data path 35. The encrypted eight bytes of data is forwarded to a programmable permutation block 18 via a one byte data path 46. Programmable permutation block 17 transposes the eight encrypted data bytes coming from encryption block 15. Programmable permutation block 17 deals with only eight bytes at a time. It takes only three bits to specify all of the transposed addresses of a single byte. Permutation block 17, therefore, requires twenty-four properly distributed address bits to perform the eight byte transposition. Likewise, Programmable permutation block 18 transposes the eight encrypted data bytes coming from encryption block 16. Permutation block 18 also requires twenty-four properly distributed address bits to perform the eight byte transposition.

Twenty-four bits of path control session key 11 are forwarded via twenty-four bit data path 32 to permutation block 17 and permutation block 18 to be used as a twenty-four bit permutation key. In the preferred embodiment, if the twenty-four bits of the permutation key used to control permutation blocks 17 and 18 are not properly distributed, the three bit addresses which would cause address collisions are edited using one-time pad techniques until all potential collisions are resolved.

More specifically, each of permutation block 17 and permutation block 18 uses eight addressable eight bit wide registers that store the information from encryption block 15 and encryption block 16, respectively. It takes three of the twenty-four bits of the permutation key to address each incoming byte. The first three bits of the twenty-four bits of the permutation key specify which register the first byte is stored in. The second three bits specify the register for the second byte. And so on.

In order to prevent multiple bytes from getting stored into the same register, each three bit segment must be unique. An edit is performed on the permutation key whenever a new key is defined (i.e. loaded). When a duplicate three bit register address is found, the three bits are changed. It is important that the algorithm that performs this modification is, itself, unpredictable. One way to implement the algorithm is to add three bits from another part of the key to the "duplicate" three bit address and check to see if the result is unique. If the result is not unique, another three bits are added and another check is made to see if the result is unique. Because the key is unknown and unpredictable, It will not be possible for an attacker to predict how duplicate addresses are fixed.

There are 8! (40,320) possible permutations generated by each of permutation block 17 and permutation block 18.

Intelligent multiplexer 14 receives eight bytes of output from permutation block 17 via an one byte data path 47. Intelligent multiplexer 14 also receives eight bytes of output from permutation block 18 via a one byte data path 48. Intelligent multiplexer 14 combines the output from permutation block 17 and the output from permutation block 18 into a sixteen byte data stream.

Sixteen bits of path control session key 11 are forwarded, via sixteen bit data path 33, to intelligent multiplexer 14 to be used as a sixteen bit multiplexer key. The sixteen bit multiplexer key is used to retrieve bytes of data from either programmable permutation block 17 via a data path 47 or programmable permutation block 18 via a data path 48. The first bit of the sixteen bit multiplexer key is used to direct the first outgoing byte. If the first bit of the sixteen bit multiplexer key is a zero, the first byte of plaintext message 20 is received by intelligent multiplexer 14 from programmable permutation block 17. If the first bit of the sixteen bit multiplexer key is a one, the first byte of plaintext message 20 is received from programmable permutation block 18. The second bit of the sixteen bit multiplexer key is used to direct the second outgoing byte. If the second bit of the sixteen bit multiplexer key is a zero, the second byte of plaintext message 20 is received from intelligent multiplexer 14 by programmable permutation block 17. If the second bit of the sixteen bit multiplexer key is a one, the second byte of plaintext message 20 is received by intelligent multiplexer 14 from programmable permutation block 18.

This pattern continues until programmable permutation block 17 and programmable permutation block 18 have each provided eight bytes of the sixteen bytes of plaintext message 20. In the preferred embodiment of the present invention, transmission control defaults to the non-empty permutation block when one of the permutation blocks is emptied. It is

therefore unnecessary to edit the multiplexer key so that there are an equal number of "1"s and "0"s. There are 12,870 possible ways that the eight byte outputs from permutation block 17 and permutation block 18 can be combined into a single sixteen byte data stream.

Additionally, while the embodiment of the present invention is shown in Figure 1 to have two parallel data streams each utilizing an encryption block and a permutation block, in alternative embodiments, more than two parallel data streams each utilizing an encryption block and a permutation block are used.

This is illustrated, for example, by Figure 2 which shows a simplified embodiment of the present invention which utilizes four parallel data streams.

By way of an eight bit data path 51, a plaintext message is input into an intelligent de-multiplexer 53, one byte at a time.

A portion of a path control session key (not shown) is used as a sixty-four bit de-multiplexer key. The sixty-four bit de-multiplexer key is used to steer each byte of the plaintext message to either an encryption block 55 an encryption block 56, an encryption block 57 or an encryption block 58. The first two bits of the sixty-four bit de-multiplexer key are used to direct the first incoming byte. If the first two bits of the sixty-four bit de-multiplexer key indicate a value of zero (00_2), the first byte of the plaintext message is passed by intelligent de-multiplexer 53 to encryption block 55. If the first two bits of the sixty-four bit de-multiplexer key are equal to a value of one (01_2), the first byte of the plaintext message is passed to encryption block 56. If the first two bits of the sixty-four bit de-multiplexer key indicate a value of two (10_2), the first byte of the plaintext message is passed by intelligent de-multiplexer 53 to encryption block 57. If the first two bits of the sixty-four bit de-multiplexer key are equal to a value of three (11_2), the first byte of the

plaintext message is passed to encryption block 58. The second two bits of the sixty-four bit de-multiplexer key is used to direct the second incoming byte. If the second two bits of the sixty-four bit de-multiplexer key indicate a value of zero (00_2), the second byte of the plaintext message is passed by intelligent de-multiplexer 53 to encryption block 55. If the second two bits of the sixty-four bit de-multiplexer key are equal to a value of one (01_2), the second byte of the plaintext message is passed to encryption block 56. If the second two bits of the sixty-four bit de-multiplexer key indicate a value of two (10_2), the second byte of the plaintext message is passed by intelligent de-multiplexer 53 to encryption block 57. If the second two bits of the sixty-four bit de-multiplexer key are equal to a value of three (11_2), the second byte of the plaintext message is passed to encryption block 58. And so on.

This pattern continues until encryption block 55, encryption block 56, encryption block 57 and encryption block 58 have each received eight bytes of the plaintext message. In the preferred embodiment of the present invention, the sixty-four bit de-multiplexer key is edited, if necessary, so that encryption block 55, encryption block 56, encryption block 57 and encryption block 58 each will receive eight bytes of the thirty-two bytes of the plaintext message. Encryption block 55 encrypts the received eight bytes of data. The encryption is performed using an encryption session key (not shown). The encrypted eight bytes of data is forwarded to a programmable permutation block 59.

Likewise, encryption block 56 encrypts the received eight bytes of data. The encryption is performed using the encryption session key . The encrypted eight bytes of data is forwarded to a programmable permutation block 60.

Encryption block 57 encrypts the received eight bytes of data. The encryption is performed using the encryption session key . The encrypted eight bytes of data is forwarded to a programmable permutation block 61.

Encryption block 58 encrypts the received eight bytes of data. The encryption is performed using the encryption session key . The encrypted eight bytes of data is forwarded to a programmable permutation block 62.

Programmable permutation block 59 transposes the eight encrypted data bytes coming from encryption block 55. Programmable permutation block 59 deals with only eight bytes at a time. It takes only three bits to specify all of the transposed addresses of a single byte. Permutation block 59, therefore, requires twenty-four properly distributed address bits to perform the eight byte transposition. Likewise, Programmable permutation block 60 transposes the eight encrypted data bytes coming from encryption block 56. Permutation block 60 also requires twenty-four properly distributed address bits to perform the eight byte transposition. Programmable permutation block 61 transposes the eight encrypted data bytes coming from encryption block 57. Permutation block 61 also requires twenty-four properly distributed address bits to perform the eight byte transposition. Programmable permutation block 62 transposes the eight encrypted data bytes coming from encryption block 58. Permutation block 62 also requires twenty-four properly distributed address bits to perform the eight byte transposition.

In order to prevent multiple bytes from getting stored into the same register, each three bit segment must be unique. An edit is performed on the permutation key whenever a new key is defined (i.e. loaded). When a duplicate three bit register address is found, the three bits are changed. It is important that the algorithm that performs this modification is, itself, unpredictable. One way to implement the algorithm is to add three bits from another part of the key to the "duplicate" three bit address and check to see if

the result is unique. If the result is not unique, another three bits are added and another check is made to see if the result is unique. Because the key is unknown and unpredictable, It will not be possible for an attacker to predict how duplicate addresses are fixed.

Intelligent multiplexer 54 receives eight bytes of output from each of permutation block 59, permutation block 60, permutation block 61 and permutation block 62. Intelligent multiplexer 54 combines the output from permutation block 59, the output from permutation block 60, the output from permutation block 61 and the output from permutation block 62 into a 32-byte data stream.

Sixty-four bits of the path control session key is used as a sixty-four bit multiplexer key. The sixty-four bit multiplexer key is used to retrieve bytes from programmable permutation block 59, programmable permutation block 60, programmable permutation block 61 and programmable permutation block 62.

As can be seen by the example shown in Figure 2, the path control session key will grow bigger as more encryption blocks are added in order to insure that all bytes from each encryption block get transmitted before the path control session key is recycled. The path control session key should be edited before it is loaded into the multiplex control register to insure that all bytes from all blocks get transmitted. For example, eight encryption blocks connected in parallel will require a 192 bit session key (three control bits/byte x eight bytes/encryption block x 8 encryption blocks = 192 bits).

The foregoing discussion discloses and describes merely exemplary methods and embodiments of the present invention. As will be understood by those familiar with the art, the invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof.

For example, in an alternative embodiment of the present invention key-specified permutation logic can be added to the input of encryption block 15 and encryption block 16. This would give additional protection against a plaintext attack.

Accordingly, the disclosure of the present invention is intended to be illustrative, but not limiting, of the scope of the invention, which is set forth in the following claims.

CLAIMS

We Claim:

1. A cryptographic device comprising:

a first encryption block for encrypting data to produce first encrypted data;

a second encryption block for encrypting data to produce second encrypted data;

a de-multiplexer means for receiving data portions from a message and directing the data portions to one of the first encryption block and to the second encryption block, based on a value within a path control session key;

a first permutation block which permutes the first encrypted data to produce first permuted data;

a second permutation block which permutes the second encrypted data to produce second permuted data; and,

a multiplexer means for receiving data portions from the first encrypted data and from the second encrypted data to produce an encrypted output data stream.

2. A cryptographic device as in claim 1 wherein the first encryption block and the second encryption block each use an encryption session key to perform encryption.

3. A cryptographic device as in claim 1 wherein each data portion directed by the de-multiplexer means is one byte in length.

4. A cryptographic device as in claim 1 wherein the first encryption block and the second encryption block each encrypt data eight bytes at a time.

5. A cryptographic device as in claim 1 wherein the first permutation block and the second permutation block perform permutations based on a value within the path control session key.

6. A cryptographic device as in claim 1 wherein the multiplexer means selects an order in which the data portions from the first permutation block and the second permutation block are received based on a value within the path control session key.

7. A cryptographic device comprising:

a plurality of encryption blocks, each for encrypting data to produce encrypted data;

a de-multiplexer means for receiving data portions from a message and directing the data portions to one of the plurality of encryption blocks, based on a value within a path control session key;

a plurality of permutation blocks, each permutation block being associated with an encryption block from the plurality of encryption blocks, each permutation block permuting encrypted data from the encryption block associated therewith; and

a multiplexer means for receiving data portions from each of the plurality of permutation blocks to produce an encrypted output data stream.

8. A cryptographic device as in claim 7 wherein each encryption block in the plurality of encryption blocks uses an encryption session key to perform encryption.

9. A cryptographic device as in claim 7 wherein each data portion directed by the de-multiplexer means is one byte in length.

10. A cryptographic device as in claim 7 wherein each encryption block in the plurality of encryption blocks encrypt data eight bytes at a time.

11. A cryptographic device as in claim 7 wherein each permutation block in the plurality of permutation blocks performs permutations based on a value within the path control session key.

12. A cryptographic device as in claim 7 wherein the multiplexer means selects an order in which the data portions from the plurality of permutation blocks are received based on a value within the path control session key.

13. A method for encrypting data comprising, cryptographic device comprising:

(a) receiving data portions from a message and directing the data portions to one of a plurality of encryption blocks, based on a value within a path control session key;

(b) encrypting data to produce encrypted data within each encryption block from the plurality of encryption blocks;

(c) permuting encryption data from each encryption block from the plurality of encryption blocks; and

(d) receiving data portions from each of the plurality of permutation blocks to produce an encrypted output data stream.

14. A method as in claim 13 wherein step (b) includes each encryption block in the plurality of encryption blocks using an encryption session key to perform encryption.

15. A method as in claim 13 wherein in step (a) each data portion received is one byte in length.

16. A method as in claim 13 wherein in step (b) each encryption block in the plurality of encryption blocks encrypt data eight bytes at a time.

17. A method as in claim 13 wherein in step (c) permutations are based on a value within the path control session key.

18. A method as in claim 13 wherein step (d) includes selecting an order in which the data portions from the plurality of permutation blocks are received based on a value within the path control session key.

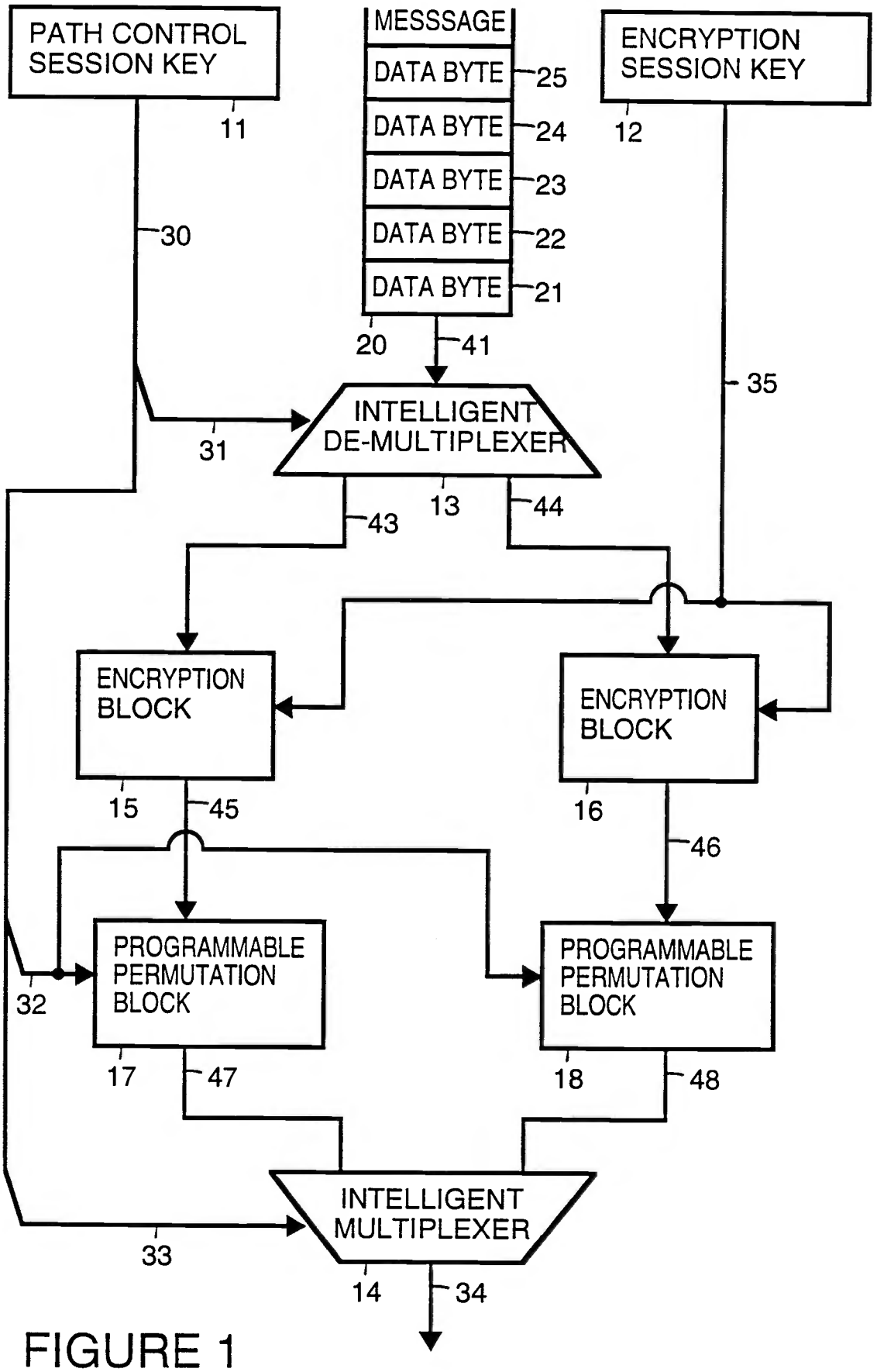

PATH CONTROL SESSION KEY
11
MESSSAGE
DATA BYTE
25
DATA BYTE
24
DATA BYTE
23
DATA BYTE
22
DATA BYTE
21
20
41
ENCRYPTION SESSION KEY
12
30
35
31
INTELLIGENT DE-MULTIPLEXER
13
43
44
ENCRYPTION BLOCK
15
45
ENCRYPTION BLOCK
16
46
32
PROGRAMMABLE PERMUTATION BLOCK
17
47
PROGRAMMABLE PERMUTATION BLOCK
18
48
INTELLIGENT MULTIPLEXER
33
14
34

FIGURE 1

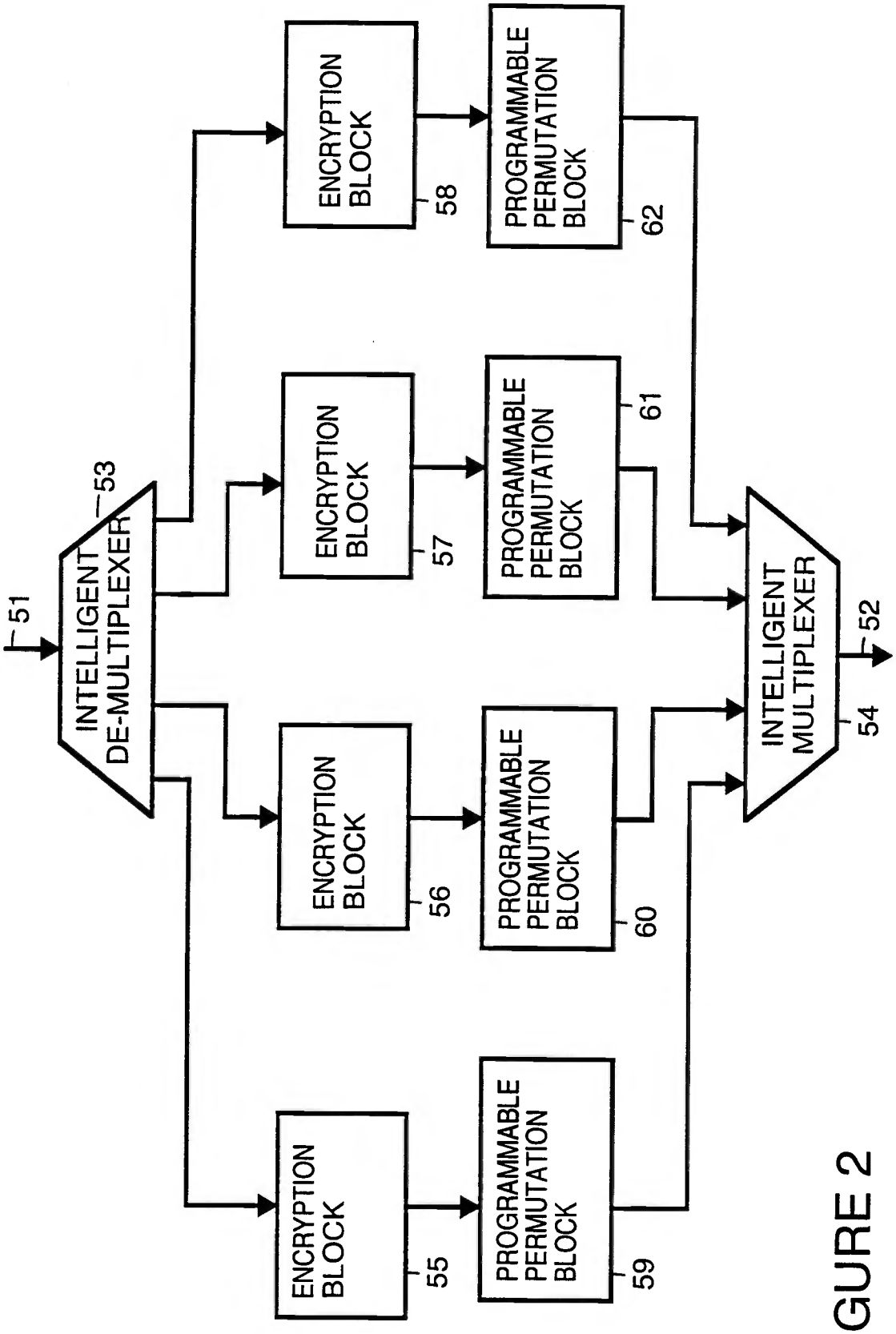

INTELLIGENT DE-MULTIPLEXER
53
51
ENCRYPTION BLOCK
58
PROGRAMMABLE PERMUTATION BLOCK
62
ENCRYPTION BLOCK
57
PROGRAMMABLE PERMUTATION BLOCK
61
ENCRYPTION BLOCK
56
PROGRAMMABLE PERMUTATION BLOCK
60
ENCRYPTION BLOCK
55
PROGRAMMABLE PERMUTATION BLOCK
59
INTELLIGENT MULTIPLEXER
54
52

FIGURE 2

INTERNATIONAL SEARCH REPORT

Inte onal Application No
PCT/US 99/28288

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L9/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 351 299 A (MATSUZAKI NATSUME ET AL) 27 September 1994 (1994-09-27) column 3, line 62 -column 6, line 60	1,7,13
A	"HIGH PERFORMANCE CRYPTOGRAPHIC HARDWARE USING PIPELINED DATA ENCRYPTION STANDARD UNITS" IBM TECHNICAL DISCLOSURE BULLETIN,US,IBM CORP. NEW YORK, vol. 38, no. 1, 1 January 1995 (1995-01-01), pages 503-507, XP000498851 ISSN: 0018-8689 the whole document	1,7,13

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 May 2000	31/05/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Carnerero Álvaro, F

1

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inte. onal Application No
PCT/US 99/28288

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	ELI BIHAM ET AL: "DIFFERENTIAL CRYPTANALYSIS OF DES-LIKE CRYPTOSYSTEMS" ADVANCES IN CRYPTOLOGY - PROCEEDINGS OF CRYPTO,DE,BERLIN, SPRINGER VERLAG, vol. CONF. 10, 1990, pages 2-21, XP000260014 the whole document	1-18

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inter ɔnal Application No
PCT/US 99/28288

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5351299 A	27-09-1994	JP 5344118 A	24-12-1993
		JP 6118872 A	28-04-1994

Form PCT/ISA/210 (patent family annex) (July 1992)

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 March 2009 (19.03.2009)

PCT

(10) International Publication Number
WO 2009/035674 A1

(51) International Patent Classification:
H04L 9/00 (2006.01)

(21) International Application Number:
PCT/US2008/010677

(22) International Filing Date:
12 September 2008 (12.09.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/993,756 14 September 2007 (14.09.2007) US

(71) Applicant *(for all designated States except US)*: **SECURITY FIRST CORPORATION** [US/US]; 22362 Gilberto #130, Rancho Santa Margarita, California 92688 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **ORSINI, Rick, L.** [US/US]; 2100 Kings Forest Lane, Flower Mound, Texas 75028 (US). **O'HARE, Mark, S.** [US/US]; 8 Kennedy Court, Coto De Caza, California 92679 (US). **DAVENPORT, Roger** [US/US]; 3204 Cr 4119, Campbell, Texas 75422 (US).

(74) Agents: **INGERMAN, Jeffrey, H.** et al.; Ropes & Gray LLP, 1211 Avenue of the Americas, New York, New York 10036-8704 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(54) Title: SYSTEMS AND METHODS FOR MANAGING CRYPTOGRAPHIC KEYS

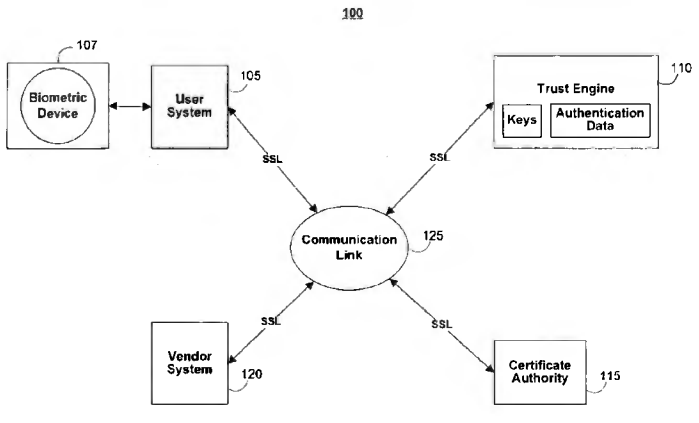


FIG. 1

(57) Abstract: A common interface for managing cryptographic keys is provided. A request to manage a cryptographic key may be received in a first interface format, translated to a common interface format, and then executed remotely from the first interface. Return arguments may then be translated from the common interface format to a format compatible with the first interface and communicated securely to the first interface. The cryptographic keys may be used in connection with a secure data parser that secures data by randomly distributing data within a data set into two or more shares.

WO 2009/035674 A1

SYSTEMS AND METHODS FOR MANAGING CRYPTOGRAPHIC KEYS

Cross-Reference to Related Applications

[0001] This application claims the benefit of U.S. provisional application No. 60/993,756, filed on September 14, 2007, which is hereby incorporated by reference herein in its entirety.

Field of the Invention

[0002] The present invention relates in general to a system for securing data from unauthorized access or use. More particularly, the present invention relates to a common interface for supporting cryptographic keys.

Background of the Invention

[0003] In today's society, individuals and businesses conduct an ever-increasing amount of activities on and over computer systems. These computer systems, including proprietary and non-proprietary computer networks, are often storing, archiving, and transmitting all types of sensitive information. Thus, an ever-increasing need exists for ensuring data stored and transmitted over these systems cannot be read or otherwise compromised.

[0004] One common solution for securing computer systems is to provide login and password functionality.

However, password management has proven to be quite costly with a large percentage of help desk calls relating to password issues. Moreover, passwords provide little security in that they are generally stored in a file susceptible to inappropriate access, through, for example, brute-force attacks.

[0005] Another solution for securing computer systems is to provide cryptographic infrastructures. Cryptography, in general, refers to protecting data by transforming, or encrypting, it into an unreadable format. Only those who possess the key(s) to the encryption can decrypt the data into a useable format. Cryptography is used to identify users, e.g., authentication, to allow access privileges, e.g., authorization, to create digital certificates and signatures, and the like. One popular cryptography system is a public key system that uses two keys, a public key known to everyone and a private key known only to the individual or business owner thereof. Generally, the data encrypted with one key is decrypted with the other and neither key is recreatable from the other.

[0006] Unfortunately, even the foregoing typical public-key cryptographic systems are still highly reliant on the user for security. For example, cryptographic systems issue the private key to the user, for example, through the user's browser. Unsophisticated users then generally store the private key on a hard drive accessible to others through an open computer system, such as, for example, the Internet. On the other hand, users may choose poor names for files containing their private key, such as, for example, "key." The result of the foregoing and

other acts is to allow the key or keys to be susceptible to compromise.

[0007] In addition to the foregoing compromises, a user may save his or her private key on a computer system configured with an archiving or backup system, potentially resulting in copies of the private key traveling through multiple computer storage devices or other systems. This security breach is often referred to as "key migration." Similar to key migration, many applications provide access to a user's private key through, at most, simple login and password access. As mentioned in the foregoing, login and password access often does not provide adequate security.

[0008] One solution for increasing the security of the foregoing cryptographic systems is to include biometrics as part of the authentication or authorization. Biometrics generally include measurable physical characteristics, such as, for example, finger prints or speech that can be checked by an automated system, such as, for example, pattern matching or recognition of finger print patterns or speech patterns. In such systems, a user's biometric and/or keys may be stored on mobile computing devices, such as, for example, a smartcard, laptop, personal digital assistant, or mobile phone, thereby allowing the biometric or keys to be usable in a mobile environment.

[0009] The foregoing mobile biometric cryptographic system still suffers from a variety of drawbacks. For example, the mobile user may lose or break the smartcard or portable computing device, thereby having his or her access to potentially important data entirely cut-off. Alternatively, a malicious person may steal the mobile user's smartcard or portable

computing device and use it to effectively steal the mobile user's digital credentials. On the other hand, the portable-computing device may be connected to an open system, such as the Internet, and, like passwords, the file where the biometric is stored may be susceptible to compromise through user inattentiveness to security or malicious intruders.

[0010] In addition, there are many ways to securely create, store, and manage individual cryptographic keys. For example, some applications may store a user's cryptographic keys in a key store or other data structure. The cryptographic keys in a user's key store may be accessed by a variety of applications. Some applications, however, may be incompatible with other applications or may compromise the security of a user's cryptographic keys by, for example, exposing one or more of the keys to corruption or unauthorized or unsecured access.

Summary of the Invention

[0011] Accordingly, a cryptographic system whose security is user-independent while still supporting mobile users is provided.

[0012] In addition, a common interface, for example, an application programming interface ("API"), is also provided that is capable of supporting multiple interfaces to a variety of cryptographic key providers and present cryptographic keys obtained from these key providers to a secure parser engine to be used, for example, for securing data for storage or transmission. Such secure parser engines are described in more detail in Orsini et al. U.S. Patent No. 7,391,865, U.S. Patent

Application No. 11/258,839, filed October 25, 2005, and U.S. Patent Application No. 11/602,667, filed November 20, 2006, all of which are hereby incorporated by reference herein in their entireties.

[0013] Accordingly, one aspect of the present invention is to provide a method for securing virtually any type of data from unauthorized access or use. The method comprises one or more steps of parsing, splitting and/or separating the data to be secured into two or more parts or portions. The method also comprises encrypting the data to be secured. Encryption of the data may be performed prior to or after the first parsing, splitting and/or separating of the data. In addition, the encrypting step may be repeated for one or more portions of the data. Similarly, the parsing, splitting and/or separating steps may be repeated for one or more portions of the data. The method also optionally comprises storing the parsed, split and/or separated data that has been encrypted in one location or in multiple locations. This method also optionally comprises reconstituting or re-assembling the secured data into its original form for authorized access or use. This method may be incorporated into the operations of any computer, server, engine or the like, that is capable of executing the desired steps of the method.

[0014] Another aspect of the present invention provides a system for securing virtually any type of data from unauthorized access or use. This system comprises a data splitting module, a cryptographic handling module, and, optionally, a data assembly module. The system may, in one embodiment, further

comprise one or more data storage facilities where secure data may be stored.

[0015] Accordingly, one aspect of the invention is to provide a secure server, or trust engine, having server-centric keys, or in other words, storing cryptographic keys and user authentication data on a server. According to this embodiment, a user accesses the trust engine in order to perform authentication and cryptographic functions, such as, but not limited to, for example, authentication, authorization, digital signing and generation, storage, and retrieval of certificates, encryption, notary-like and power-of-attorney-like actions, and the like.

[0016] Another aspect of the invention is to provide a reliable, or trusted, authentication process. Moreover, subsequent to a trustworthy positive authentication, a wide number of differing actions may be taken, from providing cryptographic technology, to system or device authorization and access, to permitting use or control of one or a wide number of electronic devices.

[0017] Another aspect of the invention is to provide cryptographic keys and authentication data in an environment where they are not lost, stolen, or compromised, thereby advantageously avoiding a need to continually reissue and manage new keys and authentication data. According to another aspect of the invention, the trust engine allows a user to use one key pair for multiple activities, vendors, and/or authentication requests. According to yet another aspect of the invention, the trust engine performs at least one step of cryptographic processing, such as, but not limited to, encrypting, authenticating, or

signing, on the server side, thereby allowing clients or users to possess only minimal computing resources.

[0018] According to yet another aspect of the invention, the trust engine includes one or multiple depositories for storing portions of each cryptographic key and authentication data. The portions are created through a data splitting process that prohibits reconstruction without a predetermined portion from more than one location in one depository or from multiple depositories. According to another embodiment, the multiple depositories may be geographically remote such that a rogue employee or otherwise compromised system at one depository will not provide access to a user's key or authentication data.

[0019] According to yet another embodiment, the authentication process advantageously allows the trust engine to process multiple authentication activities in parallel. According to yet another embodiment, the trust engine may advantageously track failed access attempts and thereby limit the number of times malicious intruders may attempt to subvert the system.

[0020] According to yet another embodiment, the trust engine may include multiple instantiations where each trust engine may predict and share processing loads with the others. According to yet another embodiment, the trust engine may include a redundancy module for polling a plurality of authentication results to ensure that more than one system authenticates the user.

[0021] Therefore, one aspect of the invention includes a secure cryptographic system, which may be remotely accessible, for storing data of any type, including, but not limited to, a plurality of private

cryptographic keys to be associated with a plurality of users. The cryptographic system associates each of the plurality of users with one or more different keys from the plurality of private cryptographic keys and performs cryptographic functions for each user using the associated one or more different keys without releasing the plurality of private cryptographic keys to the users. The cryptographic system comprises a depository system having at least one server which stores the data to be secured, such as a plurality of private cryptographic keys and a plurality of enrollment authentication data. Each enrollment authentication data identifies one of multiple users and each of the multiple users is associated with one or more different keys from the plurality of private cryptographic keys. The cryptographic system also may comprise an authentication engine which compares authentication data received by one of the multiple users to enrollment authentication data corresponding to the one of multiple users and received from the depository system, thereby producing an authentication result. The cryptographic system also may comprise a cryptographic engine which, when the authentication result indicates proper identification of the one of the multiple users, performs cryptographic functions on behalf of the one of the multiple users using the associated one or more different keys received from the depository system. The cryptographic system also may comprise a transaction engine connected to route data from the multiple users to the depository server system, the authentication engine, and the cryptographic engine.

[0022] Another aspect of the invention includes a secure cryptographic system that is optionally remotely accessible. The cryptographic system comprises a depository system having at least one server which stores at least one private key and any other data, such as, but not limited to, a plurality of enrollment authentication data, wherein each enrollment authentication data identifies one of possibly multiple users. The cryptographic system may also optionally comprise an authentication engine which compares authentication data received by users to enrollment authentication data corresponding to the user and received from the depository system, thereby producing an authentication result. The cryptographic system also comprises a cryptographic engine which, when the authentication result indicates proper identification of the user, performs cryptographic functions on behalf of the user using at least said private key, which may be received from the depository system. The cryptographic system may also optionally comprise a transaction engine connected to route data from the users to other engines or systems such as, but not limited to, the depository server system, the authentication engine, and the cryptographic engine.

[0023] Another aspect of the invention includes a method of facilitating cryptographic functions. The method comprises associating a user from multiple users with one or more keys from a plurality of private cryptographic keys stored on a secure location, such as a secure server. The method also comprises receiving authentication data from the user, and comparing the authentication data to authentication data corresponding to the user, thereby verifying the

identity of the user. The method also comprises utilizing the one or more keys to perform cryptographic functions without releasing the one or more keys to the user.

[0024] Another aspect of the invention includes an authentication system for uniquely identifying a user through secure storage of the user's enrollment authentication data. The authentication system comprises one or more data storage facilities, wherein each data storage facility includes a computer accessible storage medium which stores at least one of portions of enrollment authentication data. The authentication system also comprises an authentication engine which communicates with the data storage facility or facilities. The authentication engine comprises a data splitting module which operates on the enrollment authentication data to create portions, a data assembling module which processes the portions from at least one of the data storage facilities to assemble the enrollment authentication data, and a data comparator module which receives current authentication data from a user and compares the current authentication data with the assembled enrollment authentication data to determine whether the user has been uniquely identified.

[0025] Another aspect of the invention includes a cryptographic system. The cryptographic system comprises one or more data storage facilities, wherein each data storage facility includes a computer accessible storage medium which stores at least one portion of one ore more cryptographic keys. The cryptographic system also comprises a cryptographic engine which communicates with the data storage

facilities. The cryptographic engine also comprises a data splitting module which operate on the cryptographic keys to create portions, a data assembling module which processes the portions from at least one of the data storage facilities to assemble the cryptographic keys, and a cryptographic handling module which receives the assembled cryptographic keys and performs cryptographic functions therewith.

[0026] Another aspect of the invention includes a method of storing any type of data, including, but not limited to, authentication data in geographically remote secure data storage facilities thereby protecting the data against composition of any individual data storage facility. The method comprises receiving data at a trust engine, combining at the trust engine the data with a first substantially random value to form a first combined value, and combining the data with a second substantially random value to form a second combined value. The method comprises creating a first pairing of the first substantially random value with the second combined value, creating a second pairing of the first substantially random value with the second substantially random value, and storing the first pairing in a first secure data storage facility. The method comprises storing the second pairing in a second secure data storage facility remote from the first secure data storage facility.

[0027] Another aspect of the invention includes a method of storing any type of data, including, but not limited to, authentication data comprising receiving data, combining the data with a first set of bits to form a second set of bits, and combining the data with a third set of bits to form a fourth set of bits. The

method also comprises creating a first pairing of the first set of bits with the third set of bits. The method also comprises creating a second pairing of the first set of bits with the fourth set of bits, and storing one of the first and second pairings in a first computer accessible storage medium. The method also comprises storing the other of the first and second pairings in a second computer accessible storage medium.

[0028] Another aspect of the invention includes a method of storing cryptographic data in geographically remote secure data storage facilities thereby protecting the cryptographic data against comprise of any individual data storage facility. The method comprises receiving cryptographic data at a trust engine, combining at the trust engine the cryptographic data with a first substantially random value to form a first combined value, and combining the cryptographic data with a second substantially random value to form a second combined value. The method also comprises creating a first pairing of the first substantially random value with the second combined value, creating a second pairing of the first substantially random value with the second substantially random value, and storing the first pairing in a first secure data storage facility. The method also comprises storing the second pairing in a secure second data storage facility remote from the first secure data storage facility.

[0029] Another aspect of the invention includes a method of storing cryptographic data comprising receiving authentication data and combining the cryptographic data with a first set of bits to form a second set of bits. The method also comprises

combining the cryptographic data with a third set of bits to form a fourth set of bits, creating a first pairing of the first set of bits with the third set of bits, and creating a second pairing of the first set of bits with the fourth set of bits. The method also comprises storing one of the first and second pairings in a first computer accessible storage medium, and storing the other of the first and second pairings in a second computer accessible storage medium.

[0030] Another aspect of the invention includes a method of handling sensitive data of any type or form in a cryptographic system, wherein the sensitive data exists in a useable form only during actions by authorized users, employing the sensitive data. The method also comprises receiving in a software module, substantially randomized or encrypted sensitive data from a first computer accessible storage medium, and receiving in the software module, substantially randomized or encrypted data which may or may not be sensitive data, from one or more other computer accessible storage medium. The method also comprises processing the substantially randomized pre-encrypted sensitive data and the substantially randomized or encrypted data which may or may not be sensitive data, in the software module to assemble the sensitive data and employing the sensitive data in a software engine to perform an action. The action includes, but is not limited to, one of authenticating a user and performing a cryptographic function.

[0031] Another aspect of the invention includes a secure authentication system. The secure authentication system comprises a plurality of authentication engines. Each authentication engine

receives enrollment authentication data designed to uniquely identify a user to a degree of certainty. Each authentication engine receives current authentication data to compare to the enrollment authentication data, and each authentication engine determines an authentication result. The secure authentication system also comprises a redundancy system which receives the authentication result of at least two of the authentication engines and determines whether the user has been uniquely identified.

[0032] Another aspect of the invention includes a secure data in motion system whereby data may be transmitted in different portions that are secured in accordance with the present invention such that any one portion becoming compromised shall not provide sufficient data to restore the original data. This may be applied to any transmission of data, whether it be wired, wireless, or physical.

[0033] Another aspect of the invention includes integration of the secure data parser of the present invention into any suitable system where data is stored or communicated. For example, email system, RAID systems, video broadcasting systems, database systems, or any other suitable system may have the secure data parser integrated at any suitable level.

[0034] Another aspect of the invention includes using any suitable parsing and splitting algorithm to generate shares of data. Either random, pseudo-random, deterministic, or any combination thereof may be employed for parsing and splitting data.

Brief Description of the Drawings

[0035] The present invention is described in more detail below in connection with the attached drawings, which are meant to illustrate and not to limit the invention, and in which:

[0036] FIGURE 1 illustrates a block diagram of a cryptographic system, according to aspects of an embodiment of the invention;

[0037] FIGURE 2 illustrates a block diagram of the trust engine of FIGURE 1, according to aspects of an embodiment of the invention;

[0038] FIGURE 3 illustrates a block diagram of the transaction engine of FIGURE 2, according to aspects of an embodiment of the invention;

[0039] FIGURE 4 illustrates a block diagram of the depository of FIGURE 2, according to aspects of an embodiment of the invention;

[0040] FIGURE 5 illustrates a block diagram of the authentication engine of FIGURE 2, according to aspects of an embodiment of the invention;

[0041] FIGURE 6 illustrates a block diagram of the cryptographic engine of FIGURE 2, according to aspects of an embodiment of the invention;

[0042] FIGURE 7 illustrates a block diagram of a depository system, according to aspects of another embodiment of the invention;

[0043] FIGURE 8 illustrates a flow chart of a data splitting process according to aspects of an embodiment of the invention;

[0044] FIGURE 9, Panel A illustrates a data flow of an enrollment process according to aspects of an embodiment of the invention;

[0045] FIGURE 9, Panel B illustrates a flow chart of an interoperability process according to aspects of an embodiment of the invention;

[0046] FIGURE 10 illustrates a data flow of an authentication process according to aspects of an embodiment of the invention;

[0047] FIGURE 11 illustrates a data flow of a signing process according to aspects of an embodiment of the invention;

[0048] FIGURE 12 illustrates a data flow and an encryption/decryption process according to aspects and yet another embodiment of the invention;

[0049] FIGURE 13 illustrates a simplified block diagram of a trust engine system according to aspects of another embodiment of the invention;

[0050] FIGURE 14 illustrates a simplified block diagram of a trust engine system according to aspects of another embodiment of the invention;

[0051] FIGURE 15 illustrates a block diagram of the redundancy module of FIGURE 14, according to aspects of an embodiment of the invention;

[0052] FIGURE 16 illustrates a process for evaluating authentications according to one aspect of the invention;

[0053] FIGURE 17 illustrates a process for assigning a value to an authentication according to one aspect as shown in FIGURE 16 of the invention;

[0054] FIGURE 18 illustrates a process for performing trust arbitrage in an aspect of the invention as shown in FIGURE 17; and

[0055] FIGURE 19 illustrates a sample transaction between a user and a vendor according to aspects of an embodiment of the invention where an initial web based

contact leads to a sales contract signed by both parties.

[0056] FIGURE 20 illustrates a sample user system with a cryptographic service provider module which provides security functions to a user system.

[0057] FIGURE 21 illustrates a process for parsing, splitting and/or separating data with encryption and storage of the encryption master key with the data.

[0058] FIGURE 22 illustrates a process for parsing, splitting and/or separating data with encryption and storing the encryption master key separately from the data.

[0059] FIGURE 23 illustrates the intermediary key process for parsing, splitting and/or separating data with encryption and storage of the encryption master key with the data.

[0060] FIGURE 24 illustrates the intermediary key process for parsing, splitting and/or separating data with encryption and storing the encryption master key separately from the data.

[0061] FIGURE 25 illustrates utilization of the cryptographic methods and systems of the present invention with a small working group.

[0062] FIGURE 26 is a block diagram of an illustrative physical token security system employing the secure data parser in accordance with one embodiment of the present invention.

[0063] FIGURE 27 is a block diagram of an illustrative arrangement in which the secure data parser is integrated into a system in accordance with one embodiment of the present invention.

[0064] FIGURE 28 is a block diagram of an illustrative data in motion system in accordance with one embodiment of the present invention.

[0065] FIGURE 29 is a block diamgram of another illustrative data in motion system in accordance with one embodiment of the present invention.

[0066] FIGURE 30-32 are block diagrams of an illustrative system having the secure data parser integrated in accordance with one embodiment of the present invention.

[0067] FIGURE 33 is a process flow diagram of an illustrative process for parsing and splitting data in accordance with one embodiment of the present invention.

[0068] FIGURE 34 is a process flow diagram of an illustrative process for restoring portions of data into original data in accordance with one embodiment of the present invention.

[0069] FIGURE 35 is a process flow diagram of an illustrative process for splitting data at the bit level in accordance with one embodiment of the present invention.

[0070] FIGURE 36 is a process flow diagram of illustrative steps and features, that may be used in any suitable combination, with any suitable additions, deletions, or modifications in accordance with one embodiment of the present invention.

[0071] FIGURE 37 is a process flow diagram of illustrative steps and features, that may be used in any suitable combination, with any suitable additions, deletions, or modifications in accordance with one embodiment of the present invention.

[0072] FIGURE 38 is a simplified block diagram of the storage of key and data components within shares, that may be used in any suitable combination, with any suitable additions, deletions, or modifications in accordance with one embodiment of the present invention.

[0073] FIGURE 39 is a simplified block diagram of the storage of key and data components within shares using a workgroup key, that may be used in any suitable combination, with any suitable additions, deletions, or modifications in accordance with one embodiment of the present invention.

[0074] FIGURES 40A and 40B are simplified and illustrative process flow diagrams for header generation and data splitting for data in motion, that may be used in any suitable combination, with any suitable additions, deletions, or modifications in accordance with one embodiment of the present invention.

[0075] FIGURE 41 is a simplified block diagram of an illustrative share format, that may be used in any suitable combination, with any suitable additions, deletions, or modifications in accordance with one embodiment of the present invention.

[0076] FIGURE 42 is a process flow diagram of illustrative steps and features, that may be used in any suitable combination, with any suitable additions, deletions, or modifications for managing cryptographic keys in accordance with one embodiment of the present invention.

Detailed Description of the Invention

[0077] One aspect of the present invention is to provide a cryptographic system where one or more secure servers, or a trust engine, stores cryptographic keys and user authentication data. Users access the functionality of conventional cryptographic systems through network access to the trust engine, however, the trust engine does not release actual keys and other authentication data and therefore, the keys and data remain secure. This server-centric storage of keys and authentication data provides for user-independent security, portability, availability, and straightforwardness.

[0078] Because users can be confident in, or trust, the cryptographic system to perform user and document authentication and other cryptographic functions, a wide variety of functionality may be incorporated into the system. For example, the trust engine provider can ensure against agreement repudiation by, for example, authenticating the agreement participants, digitally signing the agreement on behalf of or for the participants, and storing a record of the agreement digitally signed by each participant. In addition, the cryptographic system may monitor agreements and determine to apply varying degrees of authentication, based on, for example, price, user, vendor, geographic location, place of use, or the like.

[0079] To facilitate a complete understanding of the invention, the remainder of the detailed description describes the invention with reference to the figures, wherein like elements are referenced with like numerals throughout.

[0080] FIGURE 1 illustrates a block diagram of a cryptographic system 100, according to aspects of an embodiment of the invention. As shown in FIGURE 1, the cryptographic system 100 includes a user system 105, a trust engine 110, a certificate authority 115, and a vendor system 120, communicating through a communication link 125.

[0081] According to one embodiment of the invention, the user system 105 comprises a conventional general-purpose computer having one or more microprocessors, such as, for example, an Intel-based processor. Moreover, the user system 105 includes an appropriate operating system, such as, for example, an operating system capable of including graphics or windows, such as Windows, Unix, Linux, or the like. As shown in FIGURE 1, the user system 105 may include a biometric device 107. The biometric device 107 may advantageously capture a user's biometric and transfer the captured biometric to the trust engine 110. According to one embodiment of the invention, the biometric device may advantageously comprise a device having attributes and features similar to those disclosed in U.S. Patent Application No. 08/926,277, filed on September 5, 1997, entitled *"RELIEF OBJECT IMAGE GENERATOR,"* U.S. Patent Application No. 09/558,634, filed on April 26, 2000, entitled *"IMAGING DEVICE FOR A RELIEF OBJECT AND SYSTEM AND METHOD OF USING THE IMAGE DEVICE,"* U.S. Patent Application No. 09/435,011, filed on November 5, 1999, entitled *"RELIEF OBJECT SENSOR ADAPTOR,"* and U.S. Patent Application No. 09/477,943, filed on January 5, 2000, entitled *"PLANAR OPTICAL IMAGE SENSOR AND SYSTEM FOR GENERATING AN ELECTRONIC IMAGE OF A RELIEF OBJECT*

FOR FINGERPRINT READING," all of which are owned by the instant assignee, and all of which are hereby incorporated by reference herein.

[0082] In addition, the user system 105 may connect to the communication link 125 through a conventional service provider, such as, for example, a dial up, digital subscriber line (DSL), cable modem, fiber connection, or the like. According to another embodiment, the user system 105 connects the communication link 125 through network connectivity such as, for example, a local or wide area network. According to one embodiment, the operating system includes a TCP/IP stack that handles all incoming and outgoing message traffic passed over the communication link 125.

[0083] Although the user system 105 is disclosed with reference to the foregoing embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives embodiments of the user system 105, including almost any computing device capable of sending or receiving information from another computer system. For example, the user system 105 may include, but is not limited to, a computer workstation, an interactive television, an interactive kiosk, a personal mobile computing device, such as a digital assistant, mobile phone, laptop, or the like, a wireless communications device, a smartcard, an embedded computing device, or the like, which can interact with the communication link 125. In such alternative systems, the operating systems will likely differ and be adapted for the particular device. However, according to one embodiment, the operating

systems advantageously continue to provide the appropriate communications protocols needed to establish communication with the communication link 125.

[0084] FIGURE 1 illustrates the trust engine 110. According to one embodiment, the trust engine 110 comprises one or more secure servers for accessing and storing sensitive information, which may be any type or form of data, such as, but not limited to text, audio, video, user authentication data and public and private cryptographic keys. According to one embodiment, the authentication data includes data designed to uniquely identify a user of the cryptographic system 100. For example, the authentication data may include a user identification number, one or more biometrics, and a series of questions and answers generated by the trust engine 110 or the user, but answered initially by the user at enrollment. The foregoing questions may include demographic data, such as place of birth, address, anniversary, or the like, personal data, such as mother's maiden name, favorite ice cream, or the like, or other data designed to uniquely identify the user. The trust engine 110 compares a user's authentication data associated with a current transaction, to the authentication data provided at an earlier time, such as, for example, during enrollment. The trust engine 110 may advantageously require the user to produce the authentication data at the time of each transaction, or, the trust engine 110 may advantageously allow the user to periodically produce authentication data, such as at the beginning of a string of transactions or the logging onto a particular vendor website.

[0085] According to the embodiment where the user produces biometric data, the user provides a physical characteristic, such as, but not limited to, facial scan, hand scan, ear scan, iris scan, retinal scan, vascular pattern, DNA, a fingerprint, writing or speech, to the biometric device 107. The biometric device advantageously produces an electronic pattern, or biometric, of the physical characteristic. The electronic pattern is transferred through the user system 105 to the trust engine 110 for either enrollment or authentication purposes.

[0086] Once the user produces the appropriate authentication data and the trust engine 110 determines a positive match between that authentication data (current authentication data) and the authentication data provided at the time of enrollment (enrollment authentication data), the trust engine 110 provides the user with complete cryptographic functionality. For example, the properly authenticated user may advantageously employ the trust engine 110 to perform hashing, digitally signing, encrypting and decrypting (often together referred to only as encrypting), creating or distributing digital certificates, and the like. However, the private cryptographic keys used in the cryptographic functions will not be available outside the trust engine 110, thereby ensuring the integrity of the cryptographic keys.

[0087] According to one embodiment, the trust engine 110 generates and stores cryptographic keys. According to another embodiment, at least one cryptographic key is associated with each user. Moreover, when the cryptographic keys include public-key technology, each private key associated with a user is generated within,

and not released from, the trust engine 110. Thus, so long as the user has access to the trust engine 110, the user may perform cryptographic functions using his or her private or public key. Such remote access advantageously allows users to remain completely mobile and access cryptographic functionality through practically any Internet connection, such as cellular and satellite phones, kiosks, laptops, hotel rooms and the like.

[0088] According to another embodiment, the trust engine 110 performs the cryptographic functionality using a key pair generated for the trust engine 110. According to this embodiment, the trust engine 110 first authenticates the user, and after the user has properly produced authentication data matching the enrollment authentication data, the trust engine 110 uses its own cryptographic key pair to perform cryptographic functions on behalf of the authenticated user.

[0089] A skilled artisan will recognize from the disclosure herein that the cryptographic keys may advantageously include some or all of symmetric keys, public keys, and private keys. In addition, a skilled artisan will recognize from the disclosure herein that the foregoing keys may be implemented with a wide number of algorithms available from commercial technologies, such as, for example, RSA, ELGAMAL, or the like.

[0090] FIGURE 1 also illustrates the certificate authority 115. According to one embodiment, the certificate authority 115 may advantageously comprise a trusted third-party organization or company that issues digital certificates, such as, for example, VeriSign,

Baltimore, Entrust, or the like. The trust engine 110 may advantageously transmit requests for digital certificates, through one or more conventional digital certificate protocols, such as, for example, PKCS10, to the certificate authority 115. In response, the certificate authority 115 will issue a digital certificate in one or more of a number of differing protocols, such as, for example, PKCS7. According to one embodiment of the invention, the trust engine 110 requests digital certificates from several or all of the prominent certificate authorities 115 such that the trust engine 110 has access to a digital certificate corresponding to the certificate standard of any requesting party.

[0091] According to another embodiment, the trust engine 110 internally performs certificate issuances. In this embodiment, the trust engine 110 may access a certificate system for generating certificates and/or may internally generate certificates when they are requested, such as, for example, at the time of key generation or in the certificate standard requested at the time of the request. The trust engine 110 will be disclosed in greater detail below.

[0092] FIGURE 1 also illustrates the vendor system 120. According to one embodiment, the vendor system 120 advantageously comprises a Web server. Typical Web servers generally serve content over the Internet using one of several internet markup languages or document format standards, such as the Hyper-Text Markup Language (HTML) or the Extensible Markup Language (XML). The Web server accepts requests from browsers like Netscape and Internet Explorer and then returns the appropriate electronic documents. A number of

server or client-side technologies can be used to increase the power of the Web server beyond its ability to deliver standard electronic documents. For example, these technologies include Common Gateway Interface (CGI) scripts, Secure Sockets Layer (SSL) security, and Active Server Pages (ASPs). The vendor system 120 may advantageously provide electronic content relating to commercial, personal, educational, or other transactions.

[0093] Although the vendor system 120 is disclosed with reference to the foregoing embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein that the vendor system 120 may advantageously comprise any of the devices described with reference to the user system 105 or combination thereof.

[0094] FIGURE 1 also illustrates the communication link 125 connecting the user system 105, the trust engine 110, the certificate authority 115, and the vendor system 120. According to one embodiment, the communication link 125 preferably comprises the Internet. The Internet, as used throughout this disclosure is a global network of computers. The structure of the Internet, which is well known to those of ordinary skill in the art, includes a network backbone with networks branching from the backbone. These branches, in turn, have networks branching from them, and so on. Routers move information packets between network levels, and then from network to network, until the packet reaches the neighborhood of its destination. From the destination, the destination network's host directs the information packet to the

appropriate terminal, or node. In one advantageous embodiment, the Internet routing hubs comprise domain name system (DNS) servers using Transmission Control Protocol/Internet Protocol (TCP/IP) as is well known in the art. The routing hubs connect to one or more other routing hubs via high-speed communication links.

[0095] One popular part of the Internet is the World Wide Web. The World Wide Web contains different computers, which store documents capable of displaying graphical and textual information. The computers that provide information on the World Wide Web are typically called "websites." A website is defined by an Internet address that has an associated electronic page. The electronic page can be identified by a Uniform Resource Locator (URL). Generally, an electronic page is a document that organizes the presentation of text, graphical images, audio, video, and so forth.

[0096] Although the communication link 125 is disclosed in terms of its preferred embodiment, one of ordinary skill in the art will recognize from the disclosure herein that the communication link 125 may include a wide range of interactive communications links. For example, the communication link 125 may include interactive television networks, telephone networks, wireless data transmission systems, two-way cable systems, customized private or public computer networks, interactive kiosk networks, automatic teller machine networks, direct links, satellite or cellular networks, and the like.

[0097] FIGURE 2 illustrates a block diagram of the trust engine 110 of FIGURE 1 according to aspects of an embodiment of the invention. As shown in FIGURE 2, the trust engine 110 includes a transaction engine 205, a

depository 210, an authentication engine 215, and a cryptographic engine 220. According to one embodiment of the invention, the trust engine 110 also includes mass storage 225. As further shown in FIGURE 2, the transaction engine 205 communicates with the depository 210, the authentication engine 215, and the cryptographic engine 220, along with the mass storage 225. In addition, the depository 210 communicates with the authentication engine 215, the cryptographic engine 220, and the mass storage 225. Moreover, the authentication engine 215 communicates with the cryptographic engine 220. According to one embodiment of the invention, some or all of the foregoing communications may advantageously comprise the transmission of XML documents to IP addresses that correspond to the receiving device. As mentioned in the foregoing, XML documents advantageously allow designers to create their own customized document tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations. Moreover, some or all of the foregoing communications may include conventional SSL technologies.

[0098] According to one embodiment, the transaction engine 205 comprises a data routing device, such as a conventional Web server available from Netscape, Microsoft, Apache, or the like. For example, the Web server may advantageously receive incoming data from the communication link 125. According to one embodiment of the invention, the incoming data is addressed to a front-end security system for the trust engine 110. For example, the front-end security system may advantageously include a firewall, an intrusion

detection system searching for known attack profiles, and/or a virus scanner. After clearing the front-end security system, the data is received by the transaction engine 205 and routed to one of the depository 210, the authentication engine 215, the cryptographic engine 220, and the mass storage 225. In addition, the transaction engine 205 monitors incoming data from the authentication engine 215 and cryptographic engine 220, and routes the data to particular systems through the communication link 125. For example, the transaction engine 205 may advantageously route data to the user system 105, the certificate authority 115, or the vendor system 120.

[0099] According to one embodiment, the data is routed using conventional HTTP routing techniques, such as, for example, employing URLs or Uniform Resource Indicators (URIs). URIs are similar to URLs, however, URIs typically indicate the source of files or actions, such as, for example, executables, scripts, and the like. Therefore, according to the one embodiment, the user system 105, the certificate authority 115, the vendor system 120, and the components of the trust engine 210, advantageously include sufficient data within communication URLs or URIs for the transaction engine 205 to properly route data throughout the cryptographic system.

[0100] Although the data routing is disclosed with reference to its preferred embodiment, a skilled artisan will recognize a wide number of possible data routing solutions or strategies. For example, XML or other data packets may advantageously be unpacked and recognized by their format, content, or the like, such that the transaction engine 205 may properly route data

throughout the trust engine 110. Moreover, a skilled artisan will recognize that the data routing may advantageously be adapted to the data transfer protocols conforming to particular network systems, such as, for example, when the communication link 125 comprises a local network.

[0101] According to yet another embodiment of the invention, the transaction engine 205 includes conventional SSL encryption technologies, such that the foregoing systems may authenticate themselves, and vise-versa, with transaction engine 205, during particular communications. As will be used throughout this disclosure, the term "½ SSL" refers to communications where a server but not necessarily the client, is SSL authenticated, and the term "FULL SSL" refers to communications where the client and the server are SSL authenticated. When the instant disclosure uses the term "SSL", the communication may comprise ½ or FULL SSL.

[0102] As the transaction engine 205 routes data to the various components of the cryptographic system 100, the transaction engine 205 may advantageously create an audit trail. According to one embodiment, the audit trail includes a record of at least the type and format of data routed by the transaction engine 205 throughout the cryptographic system 100. Such audit data may advantageously be stored in the mass storage 225.

[0103] FIGURE 2 also illustrates the depository 210. According to one embodiment, the depository 210 comprises one or more data storage facilities, such as, for example, a directory server, a database server, or the like. As shown in FIGURE 2, the depository 210 stores cryptographic keys and enrollment authentication

data. The cryptographic keys may advantageously correspond to the trust engine 110 or to users of the cryptographic system 100, such as the user or vendor. The enrollment authentication data may advantageously include data designed to uniquely identify a user, such as, user ID, passwords, answers to questions, biometric data, or the like. This enrollment authentication data may advantageously be acquired at enrollment of a user or another alternative later time. For example, the trust engine 110 may include periodic or other renewal or reissue of enrollment authentication data.

[0104] According to one embodiment, the communication from the transaction engine 205 to and from the authentication engine 215 and the cryptographic engine 220 comprises secure communication, such as, for example conventional SSL technology. In addition, as mentioned in the foregoing, the data of the communications to and from the depository 210 may be transferred using URLs, URIs, HTTP or XML documents, with any of the foregoing advantageously having data requests and formats embedded therein.

[0105] As mentioned above, the depository 210 may advantageously comprises a plurality of secure data storage facilities. In such an embodiment, the secure data storage facilities may be configured such that a compromise of the security in one individual data storage facility will not compromise the cryptographic keys or the authentication data stored therein. For example, according to this embodiment, the cryptographic keys and the authentication data are mathematically operated on so as to statistically and substantially randomize the data stored in each data

storage facility. According to one embodiment, the randomization of the data of an individual data storage facility renders that data undecipherable. Thus, compromise of an individual data storage facility produces only a randomized undecipherable number and does not compromise the security of any cryptographic keys or the authentication data as a whole.

[0106] FIGURE 2 also illustrates the trust engine 110 including the authentication engine 215. According to one embodiment, the authentication engine 215 comprises a data comparator configured to compare data from the transaction engine 205 with data from the depository 210. For example, during authentication, a user supplies current authentication data to the trust engine 110 such that the transaction engine 205 receives the current authentication data. As mentioned in the foregoing, the transaction engine 205 recognizes the data requests, preferably in the URL or URI, and routes the authentication data to the authentication engine 215. Moreover, upon request, the depository 210 forwards enrollment authentication data corresponding to the user to the authentication engine 215. Thus, the authentication engine 215 has both the current authentication data and the enrollment authentication data for comparison.

[0107] According to one embodiment, the communications to the authentication engine comprise secure communications, such as, for example, SSL technology. Additionally, security can be provided within the trust engine 110 components, such as, for example, super-encryption using public key technologies. For example, according to one embodiment, the user encrypts the current

authentication data with the public key of the authentication engine 215. In addition, the depository 210 also encrypts the enrollment authentication data with the public key of the authentication engine 215. In this way, only the authentication engine's private key can be used to decrypt the transmissions.

[0108] As shown in FIGURE 2, the trust engine 110 also includes the cryptographic engine 220. According to one embodiment, the cryptographic engine comprises a cryptographic handling module, configured to advantageously provide conventional cryptographic functions, such as, for example, public-key infrastructure (PKI) functionality. For example, the cryptographic engine 220 may advantageously issue public and private keys for users of the cryptographic system 100. In this manner, the cryptographic keys are generated at the cryptographic engine 220 and forwarded to the depository 210 such that at least the private cryptographic keys are not available outside of the trust engine 110. According to another embodiment, the cryptographic engine 220 randomizes and splits at least the private cryptographic key data, thereby storing only the randomized split data. Similar to the splitting of the enrollment authentication data, the splitting process ensures the stored keys are not available outside the cryptographic engine 220. According to another embodiment, the functions of the cryptographic engine can be combined with and performed by the authentication engine 215.

[0109] According to one embodiment, communications to and from the cryptographic engine include secure communications, such as SSL technology. In addition, XML documents may advantageously be employed to

transfer data and/or make cryptographic function requests.

[0110] FIGURE 2 also illustrates the trust engine 110 having the mass storage 225. As mentioned in the foregoing, the transaction engine 205 keeps data corresponding to an audit trail and stores such data in the mass storage 225. Similarly, according to one embodiment of the invention, the depository 210 keeps data corresponding to an audit trail and stores such data in the mass storage device 225. The depository audit trail data is similar to that of the transaction engine 205 in that the audit trail data comprises a record of the requests received by the depository 210 and the response thereof. In addition, the mass storage 225 may be used to store digital certificates having the public key of a user contained therein.

[0111] Although the trust engine 110 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize in the disclosure herein, a wide number of alternatives for the trust engine 110. For example, the trust engine 110, may advantageously perform only authentication, or alternatively, only some or all of the cryptographic functions, such as data encryption and decryption. According to such embodiments, one of the authentication engine 215 and the cryptographic engine 220 may advantageously be removed, thereby creating a more straightforward design for the trust engine 110. In addition, the cryptographic engine 220 may also communicate with a certificate authority such that the certificate authority is embodied within the trust engine 110. According to yet another embodiment,

the trust engine 110 may advantageously perform authentication and one or more cryptographic functions, such as, for example, digital signing.

[0112] FIGURE 3 illustrates a block diagram of the transaction engine 205 of FIGURE 2, according to aspects of an embodiment of the invention. According to this embodiment, the transaction engine 205 comprises an operating system 305 having a handling thread and a listening thread. The operating system 305 may advantageously be similar to those found in conventional high volume servers, such as, for example, Web servers available from Apache. The listening thread monitors the incoming communication from one of the communication link 125, the authentication engine 215, and the cryptographic engine 220 for incoming data flow. The handling thread recognizes particular data structures of the incoming data flow, such as, for example, the foregoing data structures, thereby routing the incoming data to one of the communication link 125, the depository 210, the authentication engine 215, the cryptographic engine 220, or the mass storage 225. As shown in FIGURE 3, the incoming and outgoing data may advantageously be secured through, for example, SSL technology.

[0113] FIGURE 4 illustrates a block diagram of the depository 210 of FIGURE 2 according to aspects of an embodiment of the invention. According to this embodiment, the depository 210 comprises one or more lightweight directory access protocol (LDAP) servers. LDAP directory servers are available from a wide variety of manufacturers such as Netscape, ISO, and others. FIGURE 4 also shows that the directory server preferably stores data 405 corresponding to the

cryptographic keys and data 410 corresponding to the enrollment authentication data. According to one embodiment, the depository 210 comprises a single logical memory structure indexing authentication data and cryptographic key data to a unique user ID. The single logical memory structure preferably includes mechanisms to ensure a high degree of trust, or security, in the data stored therein. For example, the physical location of the depository 210 may advantageously include a wide number of conventional security measures, such as limited employee access, modern surveillance systems, and the like. In addition to, or in lieu of, the physical securities, the computer system or server may advantageously include software solutions to protect the stored data. For example, the depository 210 may advantageously create and store data 415 corresponding to an audit trail of actions taken. In addition, the incoming and outgoing communications may advantageously be encrypted with public key encryption coupled with conventional SSL technologies.

[0114] According to another embodiment, the depository 210 may comprise distinct and physically separated data storage facilities, as disclosed further with reference to FIGURE 7.

[0115] FIGURE 5 illustrates a block diagram of the authentication engine 215 of FIGURE 2 according to aspects of an embodiment of the invention. Similar to the transaction engine 205 of FIGURE 3, the authentication engine 215 comprises an operating system 505 having at least a listening and a handling thread of a modified version of a conventional Web server, such as, for example, Web servers available from

Apache. As shown in FIGURE 5, the authentication engine 215 includes access to at least one private key 510. The private key 510 may advantageously be used for example, to decrypt data from the transaction engine 205 or the depository 210, which was encrypted with a corresponding public key of the authentication engine 215.

[0116] FIGURE 5 also illustrates the authentication engine 215 comprising a comparator 515, a data splitting module 520, and a data assembling module 525. According to the preferred embodiment of the invention, the comparator 515 includes technology capable of comparing potentially complex patterns related to the foregoing biometric authentication data. The technology may include hardware, software, or combined solutions for pattern comparisons, such as, for example, those representing finger print patterns or voice patterns. In addition, according to one embodiment, the comparator 515 of the authentication engine 215 may advantageously compare conventional hashes of documents in order to render a comparison result. According to one embodiment of the invention, the comparator 515 includes the application of heuristics 530 to the comparison. The heuristics 530 may advantageously address circumstances surrounding an authentication attempt, such as, for example, the time of day, IP address or subnet mask, purchasing profile, email address, processor serial number or ID, or the like.

[0117] Moreover, the nature of biometric data comparisons may result in varying degrees of confidence being produced from the matching of current biometric authentication data to enrollment data. For example,

unlike a traditional password which may only return a positive or negative match, a fingerprint may be determined to be a partial match, e.g. a 90% match, a 75% match, or a 10% match, rather than simply being correct or incorrect. Other biometric identifiers such as voice print analysis or face recognition may share this property of probabilistic authentication, rather than absolute authentication.

[0118] When working with such probabilistic authentication or in other cases where an authentication is considered less than absolutely reliable, it is desirable to apply the heuristics 530 to determine whether the level of confidence in the authentication provided is sufficiently high to authenticate the transaction which is being made.

[0119] It will sometimes be the case that the transaction at issue is a relatively low value transaction where it is acceptable to be authenticated to a lower level of confidence. This could include a transaction which has a low dollar value associated with it (e.g., a $10 purchase) or a transaction with low risk (e.g., admission to a members-only web site).

[0120] Conversely, for authenticating other transactions, it may be desirable to require a high degree of confidence in the authentication before allowing the transaction to proceed. Such transactions may include transactions of large dollar value (e.g., signing a multi-million dollar supply contract) or transaction with a high risk if an improper authentication occurs (e.g., remotely logging onto a government computer).

[0121] The use of the heuristics 530 in combination with confidence levels and transactions values may be

used as will be described below to allow the comparator to provide a dynamic context-sensitive authentication system.

[0122] According to another embodiment of the invention, the comparator 515 may advantageously track authentication attempts for a particular transaction. For example, when a transaction fails, the trust engine 110 may request the user to re-enter his or her current authentication data. The comparator 515 of the authentication engine 215 may advantageously employ an attempt limiter 535 to limit the number of authentication attempts, thereby prohibiting brute-force attempts to impersonate a user's authentication data. According to one embodiment, the attempt limiter 535 comprises a software module monitoring transactions for repeating authentication attempts and, for example, limiting the authentication attempts for a given transaction to three. Thus, the attempt limiter 535 will limit an automated attempt to impersonate an individual's authentication data to, for example, simply three "guesses." Upon three failures, the attempt limiter 535 may advantageously deny additional authentication attempts. Such denial may advantageously be implemented through, for example, the comparator 515 returning a negative result regardless of the current authentication data being transmitted. On the other hand, the transaction engine 205 may advantageously block any additional authentication attempts pertaining to a transaction in which three attempts have previously failed.

[0123] The authentication engine 215 also includes the data splitting module 520 and the data assembling module 525. The data splitting module 520

advantageously comprises a software, hardware, or combination module having the ability to mathematically operate on various data so as to substantially randomize and split the data into portions. According to one embodiment, original data is not recreatable from an individual portion. The data assembling module 525 advantageously comprises a software, hardware, or combination module configured to mathematically operate on the foregoing substantially randomized portions, such that the combination thereof provides the original deciphered data. According to one embodiment, the authentication engine 215 employs the data splitting module 520 to randomize and split enrollment authentication data into portions, and employs the data assembling module 525 to reassemble the portions into usable enrollment authentication data.

[0124] FIGURE 6 illustrates a block diagram of the cryptographic engine 220 of the trust engine 200 of FIGURE 2 according to aspects of one embodiment of the invention. Similar to the transaction engine 205 of FIGURE 3, the cryptographic engine 220 comprises an operating system 605 having at least a listening and a handling thread of a modified version of a conventional Web server, such as, for example, Web servers available from Apache. As shown in FIGURE 6, the cryptographic engine 220 comprises a data splitting module 610 and a data assembling module 620 that function similar to those of FIGURE 5. However, according to one embodiment, the data splitting module 610 and the data assembling module 620 process cryptographic key data, as opposed to the foregoing enrollment authentication data. Although, a skilled artisan will recognize from the disclosure herein that the data splitting module

910 and the data splitting module 620 may be combined with those of the authentication engine 215.

[0125] The cryptographic engine 220 also comprises a cryptographic handling module 625 configured to perform one, some or all of a wide number of cryptographic functions. According to one embodiment, the cryptographic handling module 625 may comprise software modules or programs, hardware, or both. According to another embodiment, the cryptographic handling module 625 may perform data comparisons, data parsing, data splitting, data separating, data hashing, data encryption or decryption, digital signature verification or creation, digital certificate generation, storage, or requests, cryptographic key generation, or the like. Moreover, a skilled artisan will recognize from the disclosure herein that the cryptographic handling module 825 may advantageously comprises a public-key infrastructure, such as Pretty Good Privacy (PGP), an RSA-based public-key system, or a wide number of alternative key management systems. In addition, the cryptographic handling module 625 may perform public-key encryption, symmetric-key encryption, or both. In addition to the foregoing, the cryptographic handling module 625 may include one or more computer programs or modules, hardware, or both, for implementing seamless, transparent, interoperability functions.

[0126] A skilled artisan will also recognize from the disclosure herein that the cryptographic functionality may include a wide number or variety of functions generally relating to cryptographic key management systems.

[0127] FIGURE 7 illustrates a simplified block diagram of a depository system 700 according to aspects of an embodiment of the invention. As shown in FIGURE 7, the depository system 700 advantageously comprises multiple data storage facilities, for example, data storage facilities D1, D2, D3, and D4. However, it is readily understood by those of ordinary skill in the art that the depository system may have only one data storage facility. According to one embodiment of the invention, each of the data storage facilities D1 through D4 may advantageously comprise some or all of the elements disclosed with reference to the depository 210 of FIGURE 4. Similar to the depository 210, the data storage facilities D1 through D4 communicate with the transaction engine 205, the authentication engine 215, and the cryptographic engine 220, preferably through conventional SSL. Communication links transferring, for example, XML documents. Communications from the transaction engine 205 may advantageously include requests for data, wherein the request is advantageously broadcast to the IP address of each data storage facility D1 through D4. On the other hand, the transaction engine 205 may broadcast requests to particular data storage facilities based on a wide number of criteria, such as, for example, response time, server loads, maintenance schedules, or the like.

[0128] In response to requests for data from the transaction engine 205, the depository system 700 advantageously forwards stored data to the authentication engine 215 and the cryptographic engine 220. The respective data assembling modules receive the forwarded data and assemble the data into useable

formats. On the other hand, communications from the authentication engine 215 and the cryptographic engine 220 to the data storage facilities D1 through D4 may include the transmission of sensitive data to be stored. For example, according to one embodiment, the authentication engine 215 and the cryptographic engine 220 may advantageously employ their respective data splitting modules to divide sensitive data into undecipherable portions, and then transmit one or more undecipherable portions of the sensitive data to a particular data storage facility.

[0129] According to one embodiment, each data storage facility, D1 through D4, comprises a separate and independent storage system, such as, for example, a directory server. According to another embodiment of the invention, the depository system 700 comprises multiple geographically separated independent data storage systems. By distributing the sensitive data into distinct and independent storage facilities D1 through D4, some or all of which may be advantageously geographically separated, the depository system 700 provides redundancy along with additional security measures. For example, according to one embodiment, only data from two of the multiple data storage facilities, D1 through D4, are needed to decipher and reassemble the sensitive data. Thus, as many as two of the four data storage facilities D1 through D4 may be inoperative due to maintenance, system failure, power failure, or the like, without affecting the functionality of the trust engine 110. In addition, because, according to one embodiment, the data stored in each data storage facility is randomized and undecipherable, compromise of any individual data

storage facility does not necessarily compromise the sensitive data. Moreover, in the embodiment having geographical separation of the data storage facilities, a compromise of multiple geographically remote facilities becomes increasingly difficult. In fact, even a rogue employee will be greatly challenged to subvert the needed multiple independent geographically remote data storage facilities.

[0130] Although the depository system 700 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the depository system 700. For example, the depository system 700 may comprise one, two or more data storage facilities. In addition, sensitive data may be mathematically operated such that portions from two or more data storage facilities are needed to reassemble and decipher the sensitive data.

[0131] As mentioned in the foregoing, the authentication engine 215 and the cryptographic engine 220 each include a data splitting module 520 and 610, respectively, for splitting any type or form of sensitive data, such as, for example, text, audio, video, the authentication data and the cryptographic key data. FIGURE 8 illustrates a flowchart of a data splitting process 800 performed by the data splitting module according to aspects of an embodiment of the invention. As shown in FIGURE 8, the data splitting process 800 begins at step 805 when sensitive data "S" is received by the data splitting module of the authentication engine 215 or the cryptographic engine 220. Preferably, in step 810, the data splitting

module then generates a substantially random number, value, or string or set of bits, "A." For example, the random number A may be generated in a wide number of varying conventional techniques available to one of ordinary skill in the art, for producing high quality random numbers suitable for use in cryptographic applications. In addition, according to one embodiment, the random number A comprises a bit length which may be any suitable length, such as shorter, longer or equal to the bit length of the sensitive data, S.

[0132] In addition, in step 820 the data splitting process 800 generates another statistically random number "C." According to the preferred embodiment, the generation of the statistically random numbers A and C may advantageously be done in parallel. The data splitting module then combines the numbers A and C with the sensitive data S such that new numbers "B" and "D" are generated. For example, number B may comprise the binary combination of A XOR S and number D may comprise the binary combination of C XOR S. The XOR function, or the "exclusive-or" function, is well known to those of ordinary skill in the art. The foregoing combinations preferably occur in steps 825 and 830, respectively, and, according to one embodiment, the foregoing combinations also occur in parallel. The data splitting process 800 then proceeds to step 835 where the random numbers A and C and the numbers B and D are paired such that none of the pairings contain sufficient data, by themselves, to reorganize and decipher the original sensitive data S. For example, the numbers may be paired as follows: AC, AD, BC, and BD. According to one embodiment, each of the foregoing

pairings is distributed to one of the depositories D1 through D4 of FIGURE 7. According to another embodiment, each of the foregoing pairings is randomly distributed to one of the depositories D1 through D4. For example, during a first data splitting process 800, the pairing AC may be sent to depository D2, through, for example, a random selection of D2's IP address. Then, during a second data splitting process 800, the pairing AC may be sent to depository D4, through, for example, a random selection of D4's IP address. In addition, the pairings may all be stored on one depository, and may be stored in separate locations on said depository.

[0133] Based on the foregoing, the data splitting process 800 advantageously places portions of the sensitive data in each of the four data storage facilities D1 through D4, such that no single data storage facility D1 through D4 includes sufficient encrypted data to recreate the original sensitive data S. As mentioned in the foregoing, such randomization of the data into individually unusable encrypted portions increases security and provides for maintained trust in the data even if one of the data storage facilities, D1 through D4, is compromised.

[0134] Although the data splitting process 800 is disclosed with reference to its preferred embodiment, the invention is not intended to be limited thereby. Rather a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the data splitting process 800. For example, the data splitting process may advantageously split the data into two numbers, for example, random number A and number B and, randomly distribute A and B through two

data storage facilities. Moreover, the data splitting process 800 may advantageously split the data among a wide number of data storage facilities through generation of additional random numbers. The data may be split into any desired, selected, predetermined, or randomly assigned size unit, including but not limited to, a bit, bits, bytes, kilobytes, megabytes or larger, or any combination or sequence of sizes. In addition, varying the sizes of the data units resulting from the splitting process may render the data more difficult to restore to a useable form, thereby increasing security of sensitive data. It is readily apparent to those of ordinary skill in the art that the split data unit sizes may be a wide variety of data unit sizes or patterns of sizes or combinations of sizes. For example, the data unit sizes may be selected or predetermined to be all of the same size, a fixed set of different sizes, a combination of sizes, or randomly generates sizes. Similarly, the data units may be distributed into one or more shares according to a fixed or predetermined data unit size, a pattern or combination of data unit sizes, or a randomly generated data unit size or sizes per share.

[0135] As mentioned in the foregoing, in order to recreate the sensitive data S, the data portions need to be derandomized and reorganized. This process may advantageously occur in the data assembling modules, 525 and 620, of the authentication engine 215 and the cryptographic engine 220, respectively. The data assembling module, for example, data assembly module 525, receives data portions from the data storage facilities D1 through D4, and reassembles the data into useable form. For example, according to one embodiment

where the data splitting module 520 employed the data splitting process 800 of FIGURE 8, the data assembling module 525 uses data portions from at least two of the data storage facilities D1 through D4 to recreate the sensitive data S. For example, the pairings of AC, AD, BC, and BD, were distributed such that any two provide one of A and B, or, C and D. Noting that S = A XOR B or S = C XOR D indicates that when the data assembling module receives one of A and B, or, C and D, the data assembling module 525 can advantageously reassemble the sensitive data S. Thus, the data assembling module 525 may assemble the sensitive data S, when, for example, it receives data portions from at least the first two of the data storage facilities D1 through D4 to respond to an assemble request by the trust engine 110.

[0136] Based on the above data splitting and assembling processes, the sensitive data S exists in usable format only in a limited area of the trust engine 110. For example, when the sensitive data S includes enrollment authentication data, usable, nonrandomized enrollment authentication data is available only in the authentication engine 215. Likewise, when the sensitive data S includes private cryptographic key data, usable, nonrandomized private cryptographic key data is available only in the cryptographic engine 220.

[0137] Although the data splitting and assembling processes are disclosed with reference to their preferred embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for splitting and reassembling the sensitive data S. For example, public-key encryption

may be used to further secure the data at the data storage facilities D1 through D4. In addition, it is readily apparent to those of ordinary skill in the art that the data splitting module described herein is also a separate and distinct embodiment of the present invention that may be incorporated into, combined with or otherwise made part of any pre-existing computer systems, software suites, database, or combinations thereof, or other embodiments of the present invention, such as the trust engine, authentication engine, and transaction engine disclosed and described herein.

[0138] FIGURE 9A illustrates a data flow of an enrollment process 900 according to aspects of an embodiment of the invention. As shown in FIGURE 9A, the enrollment process 900 begins at step 905 when a user desires to enroll with the trust engine 110 of the cryptographic system 100. According to this embodiment, the user system 105 advantageously includes a client-side applet, such as a Java-based, that queries the user to enter enrollment data, such as demographic data and enrollment authentication data. According to one embodiment, the enrollment authentication data includes user ID, password(s), biometric(s), or the like. According to one embodiment, during the querying process, the client-side applet preferably communicates with the trust engine 110 to ensure that a chosen user ID is unique. When the user ID is nonunique, the trust engine 110 may advantageously suggest a unique user ID. The client-side applet gathers the enrollment data and transmits the enrollment data, for example, through and XML document, to the trust engine 110, and in particular, to the transaction engine 205. According

to one embodiment, the transmission is encoded with the public key of the authentication engine 215.

[0139] According to one embodiment, the user performs a single enrollment during step 905 of the enrollment process 900. For example, the user enrolls himself or herself as a particular person, such as Joe User. When Joe User desires to enroll as Joe User, CEO of Mega Corp., then according to this embodiment, Joe User enrolls a second time, receives a second unique user ID and the trust engine 110 does not associate the two identities. According to another embodiment of the invention, the enrollment process 900 provides for multiple user identities for a single user ID. Thus, in the above example, the trust engine 110 will advantageously associate the two identities of Joe User. As will be understood by a skilled artisan from the disclosure herein, a user may have many identities, for example, Joe User the head of household, Joe User the member of the Charitable Foundations, and the like. Even though the user may have multiple identities, according to this embodiment, the trust engine 110 preferably stores only one set of enrollment data. Moreover, users may advantageously add, edit/update, or delete identities as they are needed.

[0140] Although the enrollment process 900 is disclosed with reference to its preferred embodiment, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for gathering of enrollment data, and in particular, enrollment authentication data. For example, the applet may be common object model (COM) based applet or the like.

[0141] On the other hand, the enrollment process may include graded enrollment. For example, at a lowest level of enrollment, the user may enroll over the communication link 125 without producing documentation as to his or her identity. According to an increased level of enrollment, the user enrolls using a trusted third party, such as a digital notary. For example, and the user may appear in person to the trusted third party, produce credentials such as a birth certificate, driver's license, military ID, or the like, and the trusted third party may advantageously include, for example, their digital signature in enrollment submission. The trusted third party may include an actual notary, a government agency, such as the Post Office or Department of Motor Vehicles, a human resources person in a large company enrolling an employee, or the like. A skilled artisan will understand from the disclosure herein that a wide number of varying levels of enrollment may occur during the enrollment process 900.

[0142] After receiving the enrollment authentication data, at step 915, the transaction engine 205, using conventional FULL SSL technology forwards the enrollment authentication data to the authentication engine 215. In step 920, the authentication engine 215 decrypts the enrollment authentication data using the private key of the authentication engine 215. In addition, the authentication engine 215 employs the data splitting module to mathematically operate on the enrollment authentication data so as to split the data into at least two independently undecipherable, randomized, numbers. As mentioned in the foregoing, at least two numbers may comprise a statistically random

number and a binary X0Red number. In step 925, the authentication engine 215 forwards each portion of the randomized numbers to one of the data storage facilities D1 through D4. As mentioned in the foregoing, the authentication engine 215 may also advantageously randomize which portions are transferred to which depositories.

[0143] Often during the enrollment process 900, the user will also desire to have a digital certificate issued such that he or she may receive encrypted documents from others outside the cryptographic system 100. As mentioned in the foregoing, the certificate authority 115 generally issues digital certificates according to one or more of several conventional standards. Generally, the digital certificate includes a public key of the user or system, which is known to everyone.

[0144] Whether the user requests a digital certificate at enrollment, or at another time, the request is transferred through the trust engine 110 to the authentication engine 215. According to one embodiment, the request includes an XML document having, for example, the proper name of the user. According to step 935, the authentication engine 215 transfers the request to the cryptographic engine 220 instructing the cryptographic engine 220 to generate a cryptographic key or key pair.

[0145] Upon request, at step 935, the cryptographic engine 220 generates at least one cryptographic key. According to one embodiment, the cryptographic handling module 625 generates a key pair, where one key is used as a private key, and one is used as a public key. The cryptographic engine 220 stores the private key and,

according to one embodiment, a copy of the public key. In step 945, the cryptographic engine 220 transmits a request for a digital certificate to the transaction engine 205. According to one embodiment, the request advantageously includes a standardized request, such as PKCS10, embedded in, for example, an XML document. The request for a digital certificate may advantageously correspond to one or more certificate authorities and the one or more standard formats the certificate authorities require.

[0146] In step 950 the transaction engine 205 forwards this request to the certificate authority 115, who, in step 955, returns a digital certificate. The return digital certificate may advantageously be in a standardized format, such as PKCS7, or in a proprietary format of one or more of the certificate authorities 115. In step 960, the digital certificate is received by the transaction engine 205, and a copy is forwarded to the user and a copy is stored with the trust engine 110. The trust engine 110 stores a copy of the certificate such that the trust engine 110 will not need to rely on the availability of the certificate authority 115. For example, when the user desires to send a digital certificate, or a third party requests the user's digital certificate, the request for the digital certificate is typically sent to the certificate authority 115. However, if the certificate authority 115 is conducting maintenance or has been victim of a failure or security compromise, the digital certificate may not be available.

[0147] At any time after issuing the cryptographic keys, the cryptographic engine 220 may advantageously employ the data splitting process 800 described above

such that the cryptographic keys are split into independently undecipherable randomized numbers. Similar to the authentication data, at step 965 the cryptographic engine 220 transfers the randomized numbers to the data storage facilities D1 through D4.

[0148] A skilled artisan will recognize from the disclosure herein that the user may request a digital certificate anytime after enrollment. Moreover, the communications between systems may advantageously include FULL SSL or public-key encryption technologies. Moreover, the enrollment process may issue multiple digital certificates from multiple certificate authorities, including one or more proprietary certificate authorities internal or external to the trust engine 110.

[0149] As disclosed in steps 935 through 960, one embodiment of the invention includes the request for a certificate that is eventually stored on the trust engine 110. Because, according to one embodiment, the cryptographic handling module 625 issues the keys used by the trust engine 110, each certificate corresponds to a private key. Therefore, the trust engine 110 may advantageously provide for interoperability through monitoring the certificates owned by, or associated with, a user. For example, when the cryptographic engine 220 receives a request for a cryptographic function, the cryptographic handling module 625 may investigate the certificates owned by the requesting user to determine whether the user owns a private key matching the attributes of the request. When such a certificate exists, the cryptographic handling module 625 may use the certificate or the public or private keys associated therewith, to perform the requested

function. When such a certificate does not exist, the cryptographic handling module 625 may advantageously and transparently perform a number of actions to attempt to remedy the lack of an appropriate key. For example, FIGURE 9B illustrates a flowchart of an interoperability process 970, which according to aspects of an embodiment of the invention, discloses the foregoing steps to ensure the cryptographic handling module 625 performs cryptographic functions using appropriate keys.

[0150] As shown in FIGURE 9B, the interoperability process 970 begins with step 972 where the cryptographic handling module 925 determines the type of certificate desired. According to one embodiment of the invention, the type of certificate may advantageously be specified in the request for cryptographic functions, or other data provided by the requestor. According to another embodiment, the certificate type may be ascertained by the data format of the request. For example, the cryptographic handling module 925 may advantageously recognize the request corresponds to a particular type.

[0151] According to one embodiment, the certificate type may include one or more algorithm standards, for example, RSA, ELGAMAL, or the like. In addition, the certificate type may include one or more key types, such as symmetric keys, public keys, strong encryption keys such as 256 bit keys, less secure keys, or the like. Moreover, the certificate type may include upgrades or replacements of one or more of the foregoing algorithm standards or keys, one or more message or data formats, one or more data encapsulation or encoding schemes, such as Base 32 or Base 64. The

certificate type may also include compatibility with one or more third-party cryptographic applications or interfaces, one or more communication protocols, or one or more certificate standards or protocols. A skilled artisan will recognize from the disclosure herein that other differences may exist in certificate types, and translations to and from those differences may be implemented as disclosed herein.

[0152] Once the cryptographic handling module 625 determines the certificate type, the interoperability process 970 proceeds to step 974, and determines whether the user owns a certificate matching the type determined in step 974. When the user owns a matching certificate, for example, the trust engine 110 has access to the matching certificate through, for example, prior storage thereof, the cryptographic handling module 825 knows that a matching private key is also stored within the trust engine 110. For example, the matching private key may be stored within the depository 210 or depository system 700. The cryptographic handling module 625 may advantageously request the matching private key be assembled from, for example, the depository 210, and then in step 976, use the matching private key to perform cryptographic actions or functions. For example, as mentioned in the foregoing, the cryptographic handling module 625 may advantageously perform hashing, hash comparisons, data encryption or decryption, digital signature verification or creation, or the like.

[0153] When the user does not own a matching certificate, the interoperability process 970 proceeds to step 978 where the cryptographic handling module 625 determines whether the users owns a cross-certified

certificate. According to one embodiment, cross-certification between certificate authorities occurs when a first certificate authority determines to trust certificates from a second certificate authority. In other words, the first certificate authority determines that certificates from the second certificate authority meets certain quality standards, and therefore, may be "certified" as equivalent to the first certificate authority's own certificates. Cross-certification becomes more complex when the certificate authorities issue, for example, certificates having levels of trust. For example, the first certificate authority may provide three levels of trust for a particular certificate, usually based on the degree of reliability in the enrollment process, while the second certificate authority may provide seven levels of trust. Cross-certification may advantageously track which levels and which certificates from the second certificate authority may be substituted for which levels and which certificates from the first. When the foregoing cross-certification is done officially and publicly between two certification authorities, the mapping of certificates and levels to one another is often called "chaining."

[0154] According to another embodiment of the invention, the cryptographic handling module 625 may advantageously develop cross-certifications outside those agreed upon by the certificate authorities. For example, the cryptographic handling module 625 may access a first certificate authority's certificate practice statement (CPS), or other published policy statement, and using, for example, the authentication tokens required by particular trust levels, match the

first certificate authority's certificates to those of another certificate authority.

[0155] When, in step 978, the cryptographic handling module 625 determines that the users owns a cross-certified certificate, the interoperability process 970 proceeds to step 976, and performs the cryptographic action or function using the cross-certified public key, private key, or both. Alternatively, when the cryptographic handling module 625 determines that the users does not own a cross-certified certificate, the interoperability process 970 proceeds to step 980, where the cryptographic handling module 625 selects a certificate authority that issues the requested certificate type, or a certificate cross-certified thereto. In step 982, the cryptographic handling module 625 determines whether the user enrollment authentication data, discussed in the foregoing, meets the authentication requirements of the chosen certificate authority. For example, if the user enrolled over a network by, for example, answering demographic and other questions, the authentication data provided may establish a lower level of trust than a user providing biometric data and appearing before a third-party, such as, for example, a notary. According to one embodiment, the foregoing authentication requirements may advantageously be provided in the chosen authentication authority's CPS.

[0156] When the user has provided the trust engine 110 with enrollment authentication data meeting the requirements of chosen certificate authority, the interoperability process 970 proceeds to step 984, where the cryptographic handling module 825 acquires the certificate from the chosen certificate authority.

According to one embodiment, the cryptographic handling module 625 acquires the certificate by following steps 945 through 960 of the enrollment process 900. For example, the cryptographic handling module 625 may advantageously employ one or more public keys from one or more of the key pairs already available to the cryptographic engine 220, to request the certificate from the certificate authority. According to another embodiment, the cryptographic handling module 625 may advantageously generate one or more new key pairs, and use the public keys corresponding thereto, to request the certificate from the certificate authority.

[0157] According to another embodiment, the trust engine 110 may advantageously include one or more certificate issuing modules capable of issuing one or more certificate types. According to this embodiment, the certificate issuing module may provide the foregoing certificate. When the cryptographic handling module 625 acquires the certificate, the interoperability process 970 proceeds to step 976, and performs the cryptographic action or function using the public key, private key, or both corresponding to the acquired certificate.

[0158] When the user, in step 982, has not provided the trust engine 110 with enrollment authentication data meeting the requirements of chosen certificate authority, the cryptographic handling module 625 determines, in step 986 whether there are other certificate authorities that have different authentication requirements. For example, the cryptographic handling module 625 may look for certificate authorities having lower authentication

requirements, but still issue the chosen certificates, or cross-certifications thereof.

[0159] When the foregoing certificate authority having lower requirements exists, the interoperability process 970 proceeds to step 980 and chooses that certificate authority. Alternatively, when no such certificate authority exists, in step 988, the trust engine 110 may request additional authentication tokens from the user. For example, the trust engine 110 may request new enrollment authentication data comprising, for example, biometric data. Also, the trust engine 110 may request the user appear before a trusted third party and provide appropriate authenticating credentials, such as, for example, appearing before a notary with a drivers license, social security card, bank card, birth certificate, military ID, or the like. When the trust engine 110 receives updated authentication data, the interoperability process 970 proceeds to step 984 and acquires the foregoing chosen certificate.

[0160] Through the foregoing interoperability process 970, the cryptographic handling module 625 advantageously provides seamless, transparent, translations and conversions between differing cryptographic systems. A skilled artisan will recognize from the disclosure herein, a wide number of advantages and implementations of the foregoing interoperable system. For example, the foregoing step 986 of the interoperability process 970 may advantageously include aspects of trust arbitrage, discussed in further detail below, where the certificate authority may under special circumstances accept lower levels of cross-certification. In

addition, the interoperability process 970 may include ensuring interoperability between and employment of standard certificate revocations, such as employing certificate revocation lists (CRL), online certificate status protocols (OCSP), or the like.

[0161] FIGURE 10 illustrates a data flow of an authentication process 1000 according to aspects of an embodiment of the invention. According to one embodiment, the authentication process 1000 includes gathering current authentication data from a user and comparing that to the enrollment authentication data of the user. For example, the authentication process 1000 begins at step 1005 where a user desires to perform a transaction with, for example, a vendor. Such transactions may include, for example, selecting a purchase option, requesting access to a restricted area or device of the vendor system 120, or the like. At step 1010, a vendor provides the user with a transaction ID and an authentication request. The transaction ID may advantageously include a 192 bit quantity having a 32 bit timestamp concatenated with a 128 bit random quantity, or a "nonce," concatenated with a 32 bit vendor specific constant. Such a transaction ID uniquely identifies the transaction such that copycat transactions can be refused by the trust engine 110.

[0162] The authentication request may advantageously include what level of authentication is needed for a particular transaction. For example, the vendor may specify a particular level of confidence that is required for the transaction at issue. If authentication cannot be made to this level of confidence, as will be discussed below, the transaction

will not occur without either further authentication by the user to raise the level of confidence, or a change in the terms of the authentication between the vendor and the server. These issues are discussed more completely below.

[0163] According to one embodiment, the transaction ID and the authentication request may be advantageously generated by a vendor-side applet or other software program. In addition, the transmission of the transaction ID and authentication data may include one or more XML documents encrypted using conventional SSL technology, such as, for example, ½ SSL, or, in other words vendor-side authenticated SSL.

[0164] After the user system 105 receives the transaction ID and authentication request, the user system 105 gathers the current authentication data, potentially including current biometric information, from the user. The user system 105, at step 1015, encrypts at least the current authentication data "B" and the transaction ID, with the public key of the authentication engine 215, and transfers that data to the trust engine 110. The transmission preferably comprises XML documents encrypted with at least conventional ½ SSL technology. In step 1020, the transaction engine 205 receives the transmission, preferably recognizes the data format or request in the URL or URI, and forwards the transmission to the authentication engine 215.

[0165] During steps 1015 and 1020, the vendor system 120, at step 1025, forwards the transaction ID and the authentication request to the trust engine 110, using the preferred FULL SSL technology. This communication may also include a vendor ID, although vendor

identification may also be communicated through a non-random portion of the transaction ID. At steps 1030 and 1035, the transaction engine 205 receives the communication, creates a record in the audit trail, and generates a request for the user's enrollment authentication data to be assembled from the data storage facilities D1 through D4. At step 1040, the depository system 700 transfers the portions of the enrollment authentication data corresponding to the user to the authentication engine 215. At step 1045, the authentication engine 215 decrypts the transmission using its private key and compares the enrollment authentication data to the current authentication data provided by the user.

[0166] The comparison of step 1045 may advantageously apply heuristical context sensitive authentication, as referred to in the forgoing, and discussed in further detail below. For example, if the biometric information received does not match perfectly, a lower confidence match results. In particular embodiments, the level of confidence of the authentication is balanced against the nature of the transaction and the desires of both the user and the vendor. Again, this is discussed in greater detail below.

[0167] At step 1050, the authentication engine 215 fills in the authentication request with the result of the comparison of step 1045. According to one embodiment of the invention, the authentication request is filled with a YES/NO or TRUE/FALSE result of the authentication process 1000. In step 1055 the filled-in authentication request is returned to the vendor for the vendor to act upon, for example,

allowing the user to complete the transaction that initiated the authentication request. According to one embodiment, a confirmation message is passed to the user.

[0168] Based on the foregoing, the authentication process 1000 advantageously keeps sensitive data secure and produces results configured to maintain the integrity of the sensitive data. For example, the sensitive data is assembled only inside the authentication engine 215. For example, the enrollment authentication data is undecipherable until it is assembled in the authentication engine 215 by the data assembling module, and the current authentication data is undecipherable until it is unwrapped by the conventional SSL technology and the private key of the authentication engine 215. Moreover, the authentication result transmitted to the vendor does not include the sensitive data, and the user may not even know whether he or she produced valid authentication data.

[0169] Although the authentication process 1000 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the authentication process 1000. For example, the vendor may advantageously be replaced by almost any requesting application, even those residing with the user system 105. For example, a client application, such as Microsoft Word, may use an application program interface (API) or a cryptographic API (CAPI) to request authentication before unlocking a document. Alternatively, a mail server, a network, a

cellular phone, a personal or mobile computing device, a workstation, or the like, may all make authentication requests that can be filled by the authentication process 1000. In fact, after providing the foregoing trusted authentication process 1000, the requesting application or device may provide access to or use of a wide number of electronic or computer devices or systems.

[0170] Moreover, the authentication process 1000 may employ a wide number of alternative procedures in the event of authentication failure. For example, authentication failure may maintain the same transaction ID and request that the user reenter his or her current authentication data. As mentioned in the foregoing, use of the same transaction ID allows the comparator of the authentication engine 215 to monitor and limit the number of authentication attempts for a particular transaction, thereby creating a more secure cryptographic system 100.

[0171] In addition, the authentication process 1000 may be advantageously be employed to develop elegant single sign-on solutions, such as, unlocking a sensitive data vault. For example, successful or positive authentication may provide the authenticated user the ability to automatically access any number of passwords for an almost limitless number of systems and applications. For example, authentication of a user may provide the user access to password, login, financial credentials, or the like, associated with multiple online vendors, a local area network, various personal computing devices, Internet service providers, auction providers, investment brokerages, or the like. By employing a sensitive data vault, users may choose

truly large and random passwords because they no longer need to remember them through association. Rather, the authentication process 1000 provides access thereto. For example, a user may choose a random alphanumeric string that is twenty plus digits in length rather than something associated with a memorable data, name, etc.

[0172] According to one embodiment, a sensitive data vault associated with a given user may advantageously be stored in the data storage facilities of the depository 210, or split and stored in the depository system 700. According to this embodiment, after positive user authentication, the trust engine 110 serves the requested sensitive data, such as, for example, to the appropriate password to the requesting application. According to another embodiment, the trust engine 110 may include a separate system for storing the sensitive data vault. For example, the trust engine 110 may include a stand-alone software engine implementing the data vault functionality and figuratively residing "behind" the foregoing front-end security system of the trust engine 110. According to this embodiment, the software engine serves the requested sensitive data after the software engine receives a signal indicating positive user authentication from the trust engine 110.

[0173] In yet another embodiment, the data vault may be implemented by a third-party system. Similar to the software engine embodiment, the third-party system may advantageously serve the requested sensitive data after the third-party system receives a signal indicating positive user authentication from the trust engine 110. According to yet another embodiment, the data vault may be implemented on the user system 105. A user-side

software engine may advantageously serve the foregoing data after receiving a signal indicating positive user authentication from the trust engine 110.

[0174] Although the foregoing data vaults are disclosed with reference to alternative embodiments, a skilled artisan will recognize from the disclosure herein, a wide number of additional implementations thereof. For example, a particular data vault may include aspects from some or all of the foregoing embodiments. In addition, any of the foregoing data vaults may employ one or more authentication requests at varying times. For example, any of the data vaults may require authentication every one or more transactions, periodically, every one or more sessions, every access to one or more Webpages or Websites, at one or more other specified intervals, or the like.

[0175] FIGURE 11 illustrates a data flow of a signing process 1100 according to aspects of an embodiment of the invention. As shown in FIGURE 11, the signing process 1100 includes steps similar to those of the authentication process 1000 described in the foregoing with reference to FIGURE 10. According to one embodiment of the invention, the signing process 1100 first authenticates the user and then performs one or more of several digital signing functions as will be discussed in further detail below. According to another embodiment, the signing process 1100 may advantageously store data related thereto, such as hashes of messages or documents, or the like. This data may advantageously be used in an audit or any other event, such as for example, when a participating party attempts to repudiate a transaction.

[0176] As shown in FIGURE 11, during the authentication steps, the user and vendor may advantageously agree on a message, such as, for example, a contract. During signing, the signing process 1100 advantageously ensures that the contract signed by the user is identical to the contract supplied by the vendor. Therefore, according to one embodiment, during authentication, the vendor and the user include a hash of their respective copies of the message or contract, in the data transmitted to the authentication engine 215. By employing only a hash of a message or contract, the trust engine 110 may advantageously store a significantly reduced amount of data, providing for a more efficient and cost effective cryptographic system. In addition, the stored hash may be advantageously compared to a hash of a document in question to determine whether the document in question matches one signed by any of the parties. The ability to determine whether the document is identical to one relating to a transaction provides for additional evidence that can be used against a claim for repudiation by a party to a transaction.

[0177] In step 1103, the authentication engine 215 assembles the enrollment authentication data and compares it to the current authentication data provided by the user. When the comparator of the authentication engine 215 indicates that the enrollment authentication data matches the current authentication data, the comparator of the authentication engine 215 also compares the hash of the message supplied by the vendor to the hash of the message supplied by the user. Thus, the authentication engine 215 advantageously ensures

that the message agreed to by the user is identical to that agreed to by the vendor.

[0178] In step 1105, the authentication engine 215 transmits a digital signature request to the cryptographic engine 220. According to one embodiment of the invention, the request includes a hash of the message or contract. However, a skilled artisan will recognize from the disclosure herein that the cryptographic engine 220 may encrypt virtually any type of data, including, but not limited to, video, audio, biometrics, images or text to form the desired digital signature. Returning to step 1105, the digital signature request preferably comprises an XML document communicated through conventional SSL technologies.

[0179] In step 1110, the authentication engine 215 transmits a request to each of the data storage facilities D1 through D4, such that each of the data storage facilities D1 through D4 transmit their respective portion of the cryptographic key or keys corresponding to a signing party. According to another embodiment, the cryptographic engine 220 employs some or all of the steps of the interoperability process 970 discussed in the foregoing, such that the cryptographic engine 220 first determines the appropriate key or keys to request from the depository 210 or the depository system 700 for the signing party, and takes actions to provide appropriate matching keys. According to still another embodiment, the authentication engine 215 or the cryptographic engine 220 may advantageously request one or more of the keys associated with the signing party and stored in the depository 210 or depository system 700.

[0180] According to one embodiment, the signing party includes one or both the user and the vendor. In such case, the authentication engine 215 advantageously requests the cryptographic keys corresponding to the user and/or the vendor. According to another embodiment, the signing party includes the trust engine 110. In this embodiment, the trust engine 110 is certifying that the authentication process 1000 properly authenticated the user, vendor, or both. Therefore, the authentication engine 215 requests the cryptographic key of the trust engine 110, such as, for example, the key belonging to the cryptographic engine 220, to perform the digital signature. According to another embodiment, the trust engine 110 performs a digital notary-like function. In this embodiment, the signing party includes the user, vendor, or both, along with the trust engine 110. Thus, the trust engine 110 provides the digital signature of the user and/or vendor, and then indicates with its own digital signature that the user and/or vendor were properly authenticated. In this embodiment, the authentication engine 215 may advantageously request assembly of the cryptographic keys corresponding to the user, the vendor, or both. According to another embodiment, the authentication engine 215 may advantageously request assembly of the cryptographic keys corresponding to the trust engine 110.

[0181] According to another embodiment, the trust engine 110 performs power of attorney-like functions. For example, the trust engine 110 may digitally sign the message on behalf of a third party. In such case, the authentication engine 215 requests the cryptographic keys associated with the third party.

According to this embodiment, the signing process 1100 may advantageously include authentication of the third party, before allowing power of attorney-like functions. In addition, the authentication process 1000 may include a check for third party constraints, such as, for example, business logic or the like dictating when and in what circumstances a particular third-party's signature may be used.

[0182] Based on the foregoing, in step 1110, the authentication engine requested the cryptographic keys from the data storage facilities D1 through D4 corresponding to the signing party. In step 1115, the data storage facilities D1 through D4 transmit their respective portions of the cryptographic key corresponding to the signing party to the cryptographic engine 220. According to one embodiment, the foregoing transmissions include SSL technologies. According to another embodiment, the foregoing transmissions may advantageously be super-encrypted with the public key of the cryptographic engine 220.

[0183] In step 1120, the cryptographic engine 220 assembles the foregoing cryptographic keys of the signing party and encrypts the message therewith, thereby forming the digital signature(s). In step 1125 of the signing process 1100, the cryptographic engine 220 transmits the digital signature(s) to the authentication engine 215. In step 1130, the authentication engine 215 transmits the filled-in authentication request along with a copy of the hashed message and the digital signature(s) to the transaction engine 205. In step 1135, the transaction engine 205 transmits a receipt comprising the transaction ID, an indication of whether the authentication was

successful, and the digital signature(s), to the vendor. According to one embodiment, the foregoing transmission may advantageously include the digital signature of the trust engine 110. For example, the trust engine 110 may encrypt the hash of the receipt with its private key, thereby forming a digital signature to be attached to the transmission to the vendor.

[0184] According to one embodiment, the transaction engine 205 also transmits a confirmation message to the user. Although the signing process 1100 is disclosed with reference to its preferred and alternative embodiments, the invention is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the signing process 1100. For example, the vendor may be replaced with a user application, such as an email application. For example, the user may wish to digitally sign a particular email with his or her digital signature. In such an embodiment, the transmission throughout the signing process 1100 may advantageously include only one copy of a hash of the message. Moreover, a skilled artisan will recognize from the disclosure herein that a wide number of client applications may request digital signatures. For example, the client applications may comprise word processors, spreadsheets, emails, voicemail, access to restricted system areas, or the like.

[0185] In addition, a skilled artisan will recognize from the disclosure herein that steps 1105 through 1120 of the signing process 1100 may advantageously employ some or all of the steps of the interoperability

process 970 of FIGURE 9B, thereby providing interoperability between differing cryptographic systems that may, for example, need to process the digital signature under differing signature types.

[0186] FIGURE 12 illustrates a data flow of an encryption/decryption process 1200 according to aspects of an embodiment of the invention. As shown in FIGURE 12, the decryption process 1200 begins by authenticating the user using the authentication process 1000. According to one embodiment, the authentication process 1000 includes in the authentication request, a synchronous session key. For example, in conventional PKI technologies, it is understood by skilled artisans that encrypting or decrypting data using public and private keys is mathematically intensive and may require significant system resources. However, in symmetric key cryptographic systems, or systems where the sender and receiver of a message share a single common key that is used to encrypt and decrypt a message, the mathematical operations are significantly simpler and faster. Thus, in the conventional PKI technologies, the sender of a message will generate synchronous session key, and encrypt the message using the simpler, faster symmetric key system. Then, the sender will encrypt the session key with the public key of the receiver. The encrypted session key will be attached to the synchronously encrypted message and both data are sent to the receiver. The receiver uses his or her private key to decrypt the session key, and then uses the session key to decrypt the message. Based on the foregoing, the simpler and faster symmetric key system is used for the majority of the encryption/decryption processing.

Thus, in the decryption process 1200, the decryption advantageously assumes that a synchronous key has been encrypted with the public key of the user. Thus, as mentioned in the foregoing, the encrypted session key is included in the authentication request.

[0187] Returning to the decryption process 1200, after the user has been authenticated in step 1205, the authentication engine 215 forwards the encrypted session key to the cryptographic engine 220. In step 1210, the authentication engine 215 forwards a request to each of the data storage facilities, D1 through D4, requesting the cryptographic key data of the user. In step 1215, each data storage facility, D1 through D4, transmits their respective portion of the cryptographic key to the cryptographic engine 220. According to one embodiment, the foregoing transmission is encrypted with the public key of the cryptographic engine 220.

[0188] In step 1220 of the decryption process 1200, the cryptographic engine 220 assembles the cryptographic key and decrypts the session key therewith. In step 1225, the cryptographic engine forwards the session key to the authentication engine 215. In step 1227, the authentication engine 215 fills in the authentication request including the decrypted session key, and transmits the filled-in authentication request to the transaction engine 205. In step 1230, the transaction engine 205 forwards the authentication request along with the session key to the requesting application or vendor. Then, according to one embodiment, the requesting application or vendor uses the session key to decrypt the encrypted message.

[0189] Although the decryption process 1200 is disclosed with reference to its preferred and

alternative embodiments, a skilled artisan will recognize from the disclosure herein, a wide number of alternatives for the decryption process 1200. For example, the decryption process 1200 may forego synchronous key encryption and rely on full public-key technology. In such an embodiment, the requesting application may transmit the entire message to the cryptographic engine 220, or, may employ some type of compression or reversible hash in order to transmit the message to the cryptographic engine 220. A skilled artisan will also recognize from the disclosure herein that the foregoing communications may advantageously include XML documents wrapped in SSL technology.

[0190] The encryption/decryption process 1200 also provides for encryption of documents or other data. Thus, in step 1235, a requesting application or vendor may advantageously transmit to the transaction engine 205 of the trust engine 110, a request for the public key of the user. The requesting application or vendor makes this request because the requesting application or vendor uses the public key of the user, for example, to encrypt the session key that will be used to encrypt the document or message. As mentioned in the enrollment process 900, the transaction engine 205 stores a copy of the digital certificate of the user, for example, in the mass storage 225. Thus, in step 1240 of the encryption process 1200, the transaction engine 205 requests the digital certificate of the user from the mass storage 225. In step 1245, the mass storage 225 transmits the digital certificate corresponding to the user, to the transaction engine 205. In step 1250, the transaction engine 205 transmits the digital certificate to the requesting

application or vendor. According to one embodiment, the encryption portion of the encryption process 1200 does not include the authentication of a user. This is because the requesting vendor needs only the public key of the user, and is not requesting any sensitive data.

[0191] A skilled artisan will recognize from the disclosure herein that if a particular user does not have a digital certificate, the trust engine 110 may employ some or all of the enrollment process 900 in order to generate a digital certificate for that particular user. Then, the trust engine 110 may initiate the encryption/decryption process 1200 and thereby provide the appropriate digital certificate. In addition, a skilled artisan will recognize from the disclosure herein that steps 1220 and 1235 through 1250 of the encryption/decryption process 1200 may advantageously employ some or all of the steps of the interoperability process of FIGURE 9B, thereby providing interoperability between differing cryptographic systems that may, for example, need to process the encryption.

[0192] FIGURE 13 illustrates a simplified block diagram of a trust engine system 1300 according to aspects of yet another embodiment of the invention. As shown in FIGURE 13, the trust engine system 1300 comprises a plurality of distinct trust engines 1305, 1310, 1315, and 1320, respectively. To facilitate a more complete understanding of the invention, FIGURE 13 illustrates each trust engine, 1305, 1310, 1315, and 1320 as having a transaction engine, a depository, and an authentication engine. However, a skilled artisan will recognize that each transaction engine may advantageously comprise some, a combination, or all of

the elements and communication channels disclosed with reference to FIGURES 1-8. For example, one embodiment may advantageously include trust engines having one or more transaction engines, depositories, and cryptographic servers or any combinations thereof.

[0193] According to one embodiment of the invention, each of the trust engines 1305, 1310, 1315 and 1320 are geographically separated, such that, for example, the trust engine 1305 may reside in a first location, the trust engine 1310 may reside in a second location, the trust engine 1315 may reside in a third location, and the trust engine 1320 may reside in a fourth location. The foregoing geographic separation advantageously decreases system response time while increasing the security of the overall trust engine system 1300.

[0194] For example, when a user logs onto the cryptographic system 100, the user may be nearest the first location and may desire to be authenticated. As described with reference to FIGURE 10, to be authenticated, the user provides current authentication data, such as a biometric or the like, and the current authentication data is compared to that user's enrollment authentication data. Therefore, according to one example, the user advantageously provides current authentication data to the geographically nearest trust engine 1305. The transaction engine 1321 of the trust engine 1305 then forwards the current authentication data to the authentication engine 1322 also residing at the first location. According to another embodiment, the transaction engine 1321 forwards the current authentication data to one or more of the authentication engines of the trust engines 1310, 1315, or 1320.

[0195] The transaction engine 1321 also requests the assembly of the enrollment authentication data from the depositories of, for example, each of the trust engines, 1305 through 1320. According to this embodiment, each depository provides its portion of the enrollment authentication data to the authentication engine 1322 of the trust engine 1305. The authentication engine 1322 then employs the encrypted data portions from, for example, the first two depositories to respond, and assembles the enrollment authentication data into deciphered form. The authentication engine 1322 compares the enrollment authentication data with the current authentication data and returns an authentication result to the transaction engine 1321 of the trust engine 1305.

[0196] Based on the above, the trust engine system 1300 employs the nearest one of a plurality of geographically separated trust engines, 1305 through 1320, to perform the authentication process. According to one embodiment of the invention, the routing of information to the nearest transaction engine may advantageously be performed at client-side applets executing on one or more of the user system 105, vendor system 120, or certificate authority 115. According to an alternative embodiment, a more sophisticated decision process may be employed to select from the trust engines 1305 through 1320. For example, the decision may be based on the availability, operability, speed of connections, load, performance, geographic proximity, or a combination thereof, of a given trust engine.

[0197] In this way, the trust engine system 1300 lowers its response time while maintaining the security

advantages associated with geographically remote data storage facilities, such as those discussed with reference to FIGURE 7 where each data storage facility stores randomized portions of sensitive data. For example, a security compromise at, for example, the depository 1325 of the trust engine 1315 does not necessarily compromise the sensitive data of the trust engine system 1300. This is because the depository 1325 contains only non-decipherable randomized data that, without more, is entirely useless.

[0198] According to another embodiment, the trust engine system 1300 may advantageously include multiple cryptographic engines arranged similar to the authentication engines. The cryptographic engines may advantageously perform cryptographic functions such as those disclosed with reference to FIGURES 1-8. According to yet another embodiment, the trust engine system 1300 may advantageously replace the multiple authentication engines with multiple cryptographic engines, thereby performing cryptographic functions such as those disclosed with reference to FIGURES 1-8. According to yet another embodiment of the invention, the trust engine system 1300 may replace each multiple authentication engine with an engine having some or all of the functionality of the authentication engines, cryptographic engines, or both, as disclosed in the foregoing,

[0199] Although the trust engine system 1300 is disclosed with reference to its preferred and alternative embodiments, a skilled artisan will recognize that the trust engine system 1300 may comprise portions of trust engines 1305 through 1320. For example, the trust engine system 1300 may include

one or more transaction engines, one or more depositories, one or more authentication engines, or one or more cryptographic engines or combinations thereof.

[0200] FIGURE 14 illustrates a simplified block diagram of a trust engine System 1400 according to aspects of yet another embodiment of the invention. As shown in FIGURE 14, the trust engine system 1400 includes multiple trust engines 1405, 1410, 1415 and 1420. According to one embodiment, each of the trust engines 1405, 1410, 1415 and 1420, comprise some or all of the elements of trust engine 110 disclosed with reference to FIGURES 1-8. According to this embodiment, when the client side applets of the user system 105, the vendor system 120, or the certificate authority 115, communicate with the trust engine system 1400, those communications are sent to the IP address of each of the trust engines 1405 through 1420. Further, each transaction engine of each of the trust engines, 1405, 1410, 1415, and 1420, behaves similar to the transaction engine 1321 of the trust engine 1305 disclosed with reference to FIGURE 13. For example, during an authentication process, each transaction engine of each of the trust engines 1405, 1410, 1415, and 1420 transmits the current authentication data to their respective authentication engines and transmits a request to assemble the randomized data stored in each of the depositories of each of the trust engines 1405 through 1420. FIGURE 14 does not illustrate all of these communications; as such illustration would become overly complex. Continuing with the authentication process, each of the depositories then communicates its portion of the randomized data to each of the

authentication engines of the each of the trust engines 1405 through 1420. Each of the authentication engines of the each of the trust engines employs its comparator to determine whether the current authentication data matches the enrollment authentication data provided by the depositories of each of the trust engines 1405 through 1420. According to this embodiment, the result of the comparison by each of the authentication engines is then transmitted to a redundancy module of the other three trust engines. For example, the result of the authentication engine from the trust engine 1405 is transmitted to the redundancy modules of the trust engines 1410, 1415, and 1420. Thus, the redundancy module of the trust engine 1405 likewise receives the result of the authentication engines from the trust engines 1410, 1415, and 1420.

[0201] FIGURE 15 illustrates a block diagram of the redundancy module of FIGURE 14. The redundancy module comprises a comparator configured to receive the authentication result from three authentication engines and transmit that result to the transaction engine of the fourth trust engine. The comparator compares the authentication result form the three authentication engines, and if two of the results agree, the comparator concludes that the authentication result should match that of the two agreeing authentication engines. This result is then transmitted back to the transaction engine corresponding to the trust engine not associated with the three authentication engines.

[0202] Based on the foregoing, the redundancy module determines an authentication result from data received from authentication engines that are preferably geographically remote from the trust engine of that the

redundancy module. By providing such redundancy functionality, the trust engine system 1400 ensures that a compromise of the authentication engine of one of the trust engines 1405 through 1420, is insufficient to compromise the authentication result of the redundancy module of that particular trust engine. A skilled artisan will recognize that redundancy module functionality of the trust engine system 1400 may also be applied to the cryptographic engine of each of the trust engines 1405 through 1420. However, such cryptographic engine communication was not shown in FIGURE 14 to avoid complexity. Moreover, a skilled artisan will recognize a wide number of alternative authentication result conflict resolution algorithms for the comparator of FIGURE 15 are suitable for use in the present invention.

[0203] According to yet another embodiment of the invention, the trust engine system 1400 may advantageously employ the redundancy module during cryptographic comparison steps. For example, some or all of the foregoing redundancy module disclosure with reference to FIGURES 14 and 15 may advantageously be implemented during a hash comparison of documents provided by one or more parties during a particular transaction.

[0204] Although the foregoing invention has been described in terms of certain preferred and alternative embodiments, other embodiments will be apparent to those of ordinary skill in the art from the disclosure herein. For example, the trust engine 110 may issue short-term certificates, where the private cryptographic key is released to the user for a predetermined period of time. For example, current

certificate standards include a validity field that can be set to expire after a predetermined amount of time. Thus, the trust engine 110 may release a private key to a user where the private key would be valid for, for example, 24 hours. According to such an embodiment, the trust engine 110 may advantageously issue a new cryptographic key pair to be associated with a particular user and then release the private key of the new cryptographic key pair. Then, once the private cryptographic key is released, the trust engine 110 immediately expires any internal valid use of such private key, as it is no longer securable by the trust engine 110.

[0205] In addition, a skilled artisan will recognize that the cryptographic system 100 or the trust engine 110 may include the ability to recognize any type of devices, such as, but not limited to, a laptop, a cell phone, a network, a biometric device or the like. According to one embodiment, such recognition may come from data supplied in the request for a particular service, such as, a request for authentication leading to access or use, a request for cryptographic functionality, or the like. According to one embodiment, the foregoing request may include a unique device identifier, such as, for example, a processor ID. Alternatively, the request may include data in a particular recognizable data format. For example, mobile and satellite phones often do not include the processing power for full X509.v3 heavy encryption certificates, and therefore do not request them. According to this embodiment, the trust engine 110 may recognize the type of data format presented, and respond only in kind.

[0206] In an additional aspect of the system described above, context sensitive authentication can be provided using various techniques as will be described below. Context sensitive authentication, for example as shown in FIGURE 16, provides the possibility of evaluating not only the actual data which is sent by the user when attempting to authenticate himself, but also the circumstances surrounding the generation and delivery of that data. Such techniques may also support transaction specific trust arbitrage between the user and trust engine 110 or between the vendor and trust engine 110, as will be described below.

[0207] As discussed above, authentication is the process of proving that a user is who he says he is. Generally, authentication requires demonstrating some fact to an authentication authority. The trust engine 110 of the present invention represents the authority to which a user must authenticate himself. The user must demonstrate to the trust engine 110 that he is who he says he is by either: knowing something that only the user should know (knowledge-based authentication), having something that only the user should have (token-based authentication), or by being something that only the user should be (biometric-based authentication).

[0208] Examples of knowledge-based authentication include without limitation a password, PIN number, or lock combination. Examples of token-based authentication include without limitation a house key, a physical credit card, a driver's license, or a particular phone number. Examples of biometric-based authentication include without limitation a fingerprint, handwriting analysis, facial scan, hand

scan, ear scan, iris scan, vascular pattern, DNA, a voice analysis, or a retinal scan.

[0209] Each type of authentication has particular advantages and disadvantages, and each provides a different level of security. For example, it is generally harder to create a false fingerprint that matches someone else's than it is to overhear someone's password and repeat it. Each type of authentication also requires a different type of data to be known to the authenticating authority in order to verify someone using that form of authentication.

[0210] As used herein, "authentication" will refer broadly to the overall process of verifying someone's identity to be who he says he is. An "authentication technique" will refer to a particular type of authentication based upon a particular piece of knowledge, physical token, or biometric reading. "Authentication data" refers to information which is sent to or otherwise demonstrated to an authentication authority in order to establish identity. "Enrollment data" will refer to the data which is initially submitted to an authentication authority in order to establish a baseline for comparison with authentication data. An "authentication instance" will refer to the data associated with an attempt to authenticate by an authentication technique.

[0211] The internal protocols and communications involved in the process of authenticating a user is described with reference to FIGURE 10 above. The part of this process within which the context sensitive authentication takes place occurs within the comparison step shown as step 1045 of FIGURE 10. This step takes place within the authentication engine 215 and involves

assembling the enrollment data 410 retrieved from the depository 210 and comparing the authentication data provided by the user to it. One particular embodiment of this process is shown in FIGURE 16 and described below.

[0212] The current authentication data provided by the user and the enrollment data retrieved from the depository 210 are received by the authentication engine 215 in step 1600 of FIGURE 16. Both of these sets of data may contain data which is related to separate techniques of authentication. The authentication engine 215 separates the authentication data associated with each individual authentication instance in step 1605. This is necessary so that the authentication data is compared with the appropriate subset of the enrollment data for the user (e.g. fingerprint authentication data should be compared with fingerprint enrollment data, rather than password enrollment data).

[0213] Generally, authenticating a user involves one or more individual authentication instances, depending on which authentication techniques are available to the user. These methods are limited by the enrollment data which were provided by the user during his enrollment process (if the user did not provide a retinal scan when enrolling, he will not be able to authenticate himself using a retinal scan), as well as the means which may be currently available to the user (e.g. if the user does not have a fingerprint reader at his current location, fingerprint authentication will not be practical). In some cases, a single authentication instance may be sufficient to authenticate a user; however, in certain circumstances a combination of

multiple authentication instances may be used in order to more confidently authenticate a user for a particular transaction.

[0214] Each authentication instance consists of data related to a particular authentication technique (e.g. fingerprint, password, smart card, etc.) and the circumstances which surround the capture and delivery of the data for that particular technique. For example, a particular instance of attempting to authenticate via password will generate not only the data related to the password itself, but also circumstantial data, known as "metadata", related to that password attempt. This circumstantial data includes information such as: the time at which the particular authentication instance took place, the network address from which the authentication information was delivered, as well as any other information as is known to those of skill in the art which may be determined about the origin of the authentication data (the type of connection, the processor serial number, etc.).

[0215] In many cases, only a small amount of circumstantial metadata will be available. For example, if the user is located on a network which uses proxies or network address translation or another technique which masks the address of the originating computer, only the address of the proxy or router may be determined. Similarly, in many cases information such as the processor serial number will not be available because of either limitations of the hardware or operating system being used, disabling of such features by the operator of the system, or other

limitations of the connection between the user's system and the trust engine 110.

[0216] As shown in FIGURE 16, once the individual authentication instances represented within the authentication data are extracted and separated in step 1605, the authentication engine 215 evaluates each instance for its reliability in indicating that the user is who he claims to be. The reliability for a single authentication instance will generally be determined based on several factors. These may be grouped as factors relating to the reliability associated with the authentication technique, which are evaluated in step 1610, and factors relating to the reliability of the particular authentication data provided, which are evaluated in step 1815. The first group includes without limitation the inherent reliability of the authentication technique being used, and the reliability of the enrollment data being used with that method. The second group includes without limitation the degree of match between the enrollment data and the data provided with the authentication instance, and the metadata associated with that authentication instance. Each of these factors may vary independently of the others.

[0217] The inherent reliability of an authentication technique is based on how hard it is for an imposter to provide someone else's correct data, as well as the overall error rates for the authentication technique. For passwords and knowledge based authentication methods, this reliability is often fairly low because there is nothing that prevents someone from revealing their password to another person and for that second person to use that password. Even a more complex

knowledge based system may have only moderate reliability since knowledge may be transferred from person to person fairly easily. Token based authentication, such as having a proper smart card or using a particular terminal to perform the authentication, is similarly of low reliability used by itself, since there is no guarantee that the right person is in possession of the proper token.

[0218] However, biometric techniques are more inherently reliable because it is generally difficult to provide someone else with the ability to use your fingerprints in a convenient manner, even intentionally. Because subverting biometric authentication techniques is more difficult, the inherent reliability of biometric methods is generally higher than that of purely knowledge or token based authentication techniques. However, even biometric techniques may have some occasions in which a false acceptance or false rejection is generated. These occurrences may be reflected by differing reliabilities for different implementations of the same biometric technique. For example, a fingerprint matching system provided by one company may provide a higher reliability than one provided by a different company because one uses higher quality optics or a better scanning resolution or some other improvement which reduces the occurrence of false acceptances or false rejections.

[0219] Note that this reliability may be expressed in different manners. The reliability is desirably expressed in some metric which can be used by the heuristics 530 and algorithms of the authentication engine 215 to calculate the confidence level of each

authentication. One preferred mode of expressing these reliabilities is as a percentage or fraction. For instance, fingerprints might be assigned an inherent reliability of 97%, while passwords might only be assigned an inherent reliability of 50%. Those of skill in the art will recognize that these particular values are merely exemplary and may vary between specific implementations.

[0220] The second factor for which reliability must be assessed is the reliability of the enrollment. This is part of the "graded enrollment" process referred to above. This reliability factor reflects the reliability of the identification provided during the initial enrollment process. For instance, if the individual initially enrolls in a manner where they physically produce evidence of their identity to a notary or other public official, and enrollment data is recorded at that time and notarized, the data will be more reliable than data which is provided over a network during enrollment and only vouched for by a digital signature or other information which is not truly tied to the individual.

[0221] Other enrollment techniques with varying levels of reliability include without limitation: enrollment at a physical office of the trust engine 110 operator; enrollment at a user's place of employment; enrollment at a post office or passport office; enrollment through an affiliated or trusted party to the trust engine 110 operator; anonymous or pseudonymous enrollment in which the enrolled identity is not yet identified with a particular real individual, as well as such other means as are known in the art.

[0222] These factors reflect the trust between the trust engine 110 and the source of identification provided during the enrollment process. For instance, if enrollment is performed in association with an employer during the initial process of providing evidence of identity, this information may be considered extremely reliable for purposes within the company, but may be trusted to a lesser degree by a government agency, or by a competitor. Therefore, trust engines operated by each of these other organizations may assign different levels of reliability to this enrollment.

[0223] Similarly, additional data which is submitted across a network, but which is authenticated by other trusted data provided during a previous enrollment with the same trust engine 110 may be considered as reliable as the original enrollment data was, even though the latter data were submitted across an open network. In such circumstances, a subsequent notarization will effectively increase the level of reliability associated with the original enrollment data. In this way for example, an anonymous or pseudonymous enrollment may then be raised to a full enrollment by demonstrating to some enrollment official the identity of the individual matching the enrolled data.

[0224] The reliability factors discussed above are generally values which may be determined in advance of any particular authentication instance. This is because they are based upon the enrollment and the technique, rather than the actual authentication. In one embodiment, the step of generating reliability based upon these factors involves looking up previously determined values for this particular authentication

technique and the enrollment data of the user. In a further aspect of an advantageous embodiment of the present invention, such reliabilities may be included with the enrollment data itself. In this way, these factors are automatically delivered to the authentication engine 215 along with the enrollment data sent from the depository 210.

[0225] While these factors may generally be determined in advance of any individual authentication instance, they still have an effect on each authentication instance which uses that particular technique of authentication for that user. Furthermore, although the values may change over time (e.g. if the user re-enrolls in a more reliable fashion), they are not dependent on the authentication data itself. By contrast, the reliability factors associated with a single specific instance's data may vary on each occasion. These factors, as discussed below, must be evaluated for each new authentication in order to generate reliability scores in step 1815.

[0226] The reliability of the authentication data reflects the match between the data provided by the user in a particular authentication instance and the data provided during the authentication enrollment. This is the fundamental question of whether the authentication data matches the enrollment data for the individual the user is claiming to be. Normally, when the data do not match, the user is considered to not be successfully authenticated, and the authentication fails. The manner in which this is evaluated may change depending on the authentication technique used. The comparison of such data is performed by the

comparator 515 function of the authentication engine 215 as shown in FIGURE 5.

[0227] For instance, matches of passwords are generally evaluated in a binary fashion. In other words, a password is either a perfect match, or a failed match. It is usually not desirable to accept as even a partial match a password which is close to the correct password if it is not exactly correct. Therefore, when evaluating a password authentication, the reliability of the authentication returned by the comparator 515 is typically either 100% (correct) or 0% (wrong), with no possibility of intermediate values.

[0228] Similar rules to those for passwords are generally applied to token based authentication methods, such as smart cards. This is because having a smart card which has a similar identifier or which is similar to the correct one, is still just as wrong as having any other incorrect token. Therefore tokens tend also to be binary authenticators: a user either has the right token, or he doesn't.

[0229] However, certain types of authentication data, such as questionnaires and biometrics, are generally not binary authenticators. For example, a fingerprint may match a reference fingerprint to varying degrees. To some extent, this may be due to variations in the quality of the data captured either during the initial enrollment or in subsequent authentications. (A fingerprint may be smudged or a person may have a still healing scar or burn on a particular finger.) In other instances the data may match less than perfectly because the information itself is somewhat variable and based upon pattern matching. (A voice analysis may seem close but not

quite right because of background noise, or the acoustics of the environment in which the voice is recorded, or because the person has a cold.) Finally, in situations where large amounts of data are being compared, it may simply be the case that much of the data matches well, but some doesn't. (A ten-question questionnaire may have resulted in eight correct answers to personal questions, but two incorrect answers.) For any of these reasons, the match between the enrollment data and the data for a particular authentication instance may be desirably assigned a partial match value by the comparator 515. In this way, the fingerprint might be said to be a 85% match, the voice print a 65% match, and the questionnaire an 80% match, for example.

[0230] This measure (degree of match) produced by the comparator 515 is the factor representing the basic issue of whether an authentication is correct or not. However, as discussed above, this is only one of the factors which may be used in determining the reliability of a given authentication instance. Note also that even though a match to some partial degree may be determined, that ultimately, it may be desirable to provide a binary result based upon a partial match. In an alternate mode of operation, it is also possible to treat partial matches as binary, i.e. either perfect (100%) or failed (0%) matches, based upon whether or not the degree of match passes a particular threshold level of match. Such a process may be used to provide a simple pass/fail level of matching for systems which would otherwise produce partial matches.

[0231] Another factor to be considered in evaluating the reliability of a given authentication instance

concerns the circumstances under which the authentication data for this particular instance are provided. As discussed above, the circumstances refer to the metadata associated with a particular authentication instance. This may include without limitation such information as: the network address of the authenticator, to the extent that it can be determined; the time of the authentication; the mode of transmission of the authentication data (phone line, cellular, network, etc.); and the serial number of the system of the authenticator.

[0232] These factors can be used to produce a profile of the type of authentication that is normally requested by the user. Then, this information can be used to assess reliability in at least two manners. One manner is to consider whether the user is requesting authentication in a manner which is consistent with the normal profile of authentication by this user. If the user normally makes authentication requests from one network address during business days (when she is at work) and from a different network address during evenings or weekends (when she is at home), an authentication which occurs from the home address during the business day is less reliable because it is outside the normal authentication profile. Similarly, if the user normally authenticates using a fingerprint biometric and in the evenings, an authentication which originates during the day using only a password is less reliable.

[0233] An additional way in which the circumstantial metadata can be used to evaluate the reliability of an instance of authentication is to determine how much corroboration the circumstance provides that the

authenticator is the individual he claims to be. For instance, if the authentication comes from a system with a serial number known to be associated with the user, this is a good circumstantial indicator that the user is who they claim to be. Conversely, if the authentication is coming from a network address which is known to be in Los Angeles when the user is known to reside in London, this is an indication that this authentication is less reliable based on its circumstances.

[0234] It is also possible that a cookie or other electronic data may be placed upon the system being used by a user when they interact with a vendor system or with the trust engine 110. This data is written to the storage of the system of the user and may contain an identification which may be read by a Web browser or other software on the user system. If this data is allowed to reside on the user system between sessions (a "persistent cookie"), it may be sent with the authentication data as further evidence of the past use of this system during authentication of a particular user. In effect, the metadata of a given instance, particularly a persistent cookie, may form a sort of token based authenticator itself.

[0235] Once the appropriate reliability factors based on the technique and data of the authentication instance are generated as described above in steps 1610 and 1615 respectively, they are used to produce an overall reliability for the authentication instance provided in step 1620. One means of doing this is simply to express each reliability as a percentage and then to multiply them together.

[0236] For example, suppose the authentication data is being sent in from a network address known to be the user's home computer completely in accordance with the user's past authentication profile (100%), and the technique being used is fingerprint identification (97%), and the initial finger print data was roistered through the user's employer with the trust engine 110 (90%), and the match between the authentication data and the original fingerprint template in the enrollment data is very good (99%). The overall reliability of this authentication instance could then be calculated as the product of these reliabilities: 100% * 97% * 90% * 99% - 86.4% reliability.

[0237] This calculated reliability represents the reliability of one single instance of authentication. The overall reliability of a single authentication instance may also be calculated using techniques which treat the different reliability factors differently, for example by using formulas where different weights are assigned to each reliability factor. Furthermore, those of skill in the art will recognize that the actual values used may represent values other than percentages and may use non-arithmetic systems. One embodiment may include a module used by an authentication requestor to set the weights for each factor and the algorithms used in establishing the overall reliability of the authentication instance.

[0238] The authentication engine 215 may use the above techniques and variations thereof to determine the reliability of a single authentication instance, indicated as step 1620. However, it may be useful in many authentication situations for multiple authentication instances to be provided at the same

time. For example, while attempting to authenticate himself using the system of the present invention, a user may provide a user identification, fingerprint authentication data, a smart card, and a password. In such a case, three independent authentication instances are being provided to the trust engine 110 for evaluation. Proceeding to step 1625, if the authentication engine 215 determines that the data provided by the user includes more than one authentication instance, then each instance in turn will be selected as shown in step 1630 and evaluated as described above in steps 1610, 1615 and 1620.

[0239] Note that many of the reliability factors discussed may vary from one of these instances to another. For instance, the inherent reliability of these techniques is likely to be different, as well as the degree of match provided between the authentication data and the enrollment data. Furthermore, the user may have provided enrollment data at different times and under different circumstances for each of these techniques, providing different enrollment reliabilities for each of these instances as well. Finally, even though the circumstances under which the data for each of these instances is being submitted is the same, the use of such techniques may each fit the profile of the user differently, and so may be assigned different circumstantial reliabilities. (For example, the user may normally use their password and fingerprint, but not their smart card.)

[0240] As a result, the final reliability for each of these authentication instances may be different from One another. However, by using multiple instances

together, the overall confidence level for the authentication will tend to increase.

[0241] Once the authentication engine has performed steps 1610 through 1620 for all of the authentication instances provided in the authentication data, the reliability of each instance is used in step 1635 to evaluate the overall authentication confidence level. This process of combining the individual authentication instance reliabilities into the authentication confidence level may be modeled by various methods relating the individual reliabilities produced, and may also address the particular interaction between some of these authentication techniques. (For example, multiple knowledge-based systems such as passwords may produce less confidence than a single password and even a fairly weak biometric, such as a basic voice analysis.)

[0242] One means in which the authentication engine 215 may combine the reliabilities of multiple concurrent authentication instances to generate a final confidence level is to multiply the unreliability of each instance to arrive at a total unreliability. The unreliability is generally the complementary percentage of the reliability. For example, a technique which is 84% reliable is 16% unreliable. The three authentication instances described above (fingerprint, smart card, password)which produce reliabilities of 86%, 75%, and 72% would have corresponding unreliabilities of (100- 86)%, (100- 75)% and (100- 72)%, or 14%, 25%, and 28%, respectively. By multiplying these unreliabilities, we get a cumulative unreliability of 14% * 25% * 28% - .98% unreliability, which corresponds to a reliability of 99.02%.

[0243] In an additional mode of operation, additional factors and heuristics 530 may be applied within the authentication engine 215 to account for the interdependence of various authentication techniques. For example, if someone has unauthorized access to a particular home computer, they probably have access to the phone line at that address as well. Therefore, authenticating based on an originating phone number as well as upon the serial number of the authenticating system does not add much to the overall confidence in the authentication. However, knowledge based authentication is largely independent of token based authentication (i.e. if someone steals your cellular phone or keys, they are no more likely to know your PIN or password than if they hadn't).

[0244] Furthermore, different vendors or other authentication requestors may wish to weigh different aspects of the authentication differently. This may include the use of separate weighing factors or algorithms used in calculating the reliability of individual instances as well as the use of different means to evaluate authentication events with multiple instances.

[0245] For instance, vendors for certain types of transactions, for instance corporate email systems, may desire to authenticate primarily based upon heuristics and other circumstantial data by default. Therefore, they may apply high weights to factors related to the metadata and other profile related information associated with the circumstances surrounding authentication events. This arrangement could be used to ease the burden on users during normal operating hours, by not requiring more from the user than that he

be logged on to the correct machine during business hours. However, another vendor may weigh authentications coming from a particular technique most heavily, for instance fingerprint matching, because of a policy decision that such a technique is most suited to authentication for the particular vendor's purposes.

[0246] Such varying weights may be defined by the authentication requestor in generating the authentication request and sent to the trust engine 110 with the authentication request in one mode of operation. Such options could also be set as preferences during an initial enrollment process for the authentication requestor and stored within the authentication engine in another mode of operation.

[0247] Once the authentication engine 215 produces an authentication confidence level for the authentication data provided, this confidence level is used to complete the authentication request in step 1640, and this information is forwarded from the authentication engine 215 to the transaction engine 205 for inclusion in a message to the authentication requestor.

[0248] The process described above is merely exemplary, and those of skill in the art will recognize that the steps need not be performed in the order shown or that only certain of the steps are desired to be performed, or that a variety of combinations of steps may be desired. Furthermore, certain steps, such as the evaluation of the reliability of each authentication instance provided, may be carried out in parallel with one another if circumstances permit.

[0249] In a further aspect of this invention, a method is provided to accommodate conditions when the

authentication confidence level produced by the process described above fails to meet the required trust level of the vendor or other party requiring the authentication. In circumstances such as these where a gap exists between the level of confidence provided and the level of trust desired, the operator of the trust engine 110 is in a position to provide opportunities for one or both parties to provide alternate data or requirements in order to close this trust gap. This process will be referred to as "trust arbitrage" herein.

[0250] Trust arbitrage may take place within a framework of cryptographic authentication as described above with reference to FIGURES 10 and 11. As shown therein, a vendor or other party will request authentication of a particular user in association with a particular transaction. In one circumstance, the vendor simply requests an authentication, either positive or negative, and after receiving appropriate data from the user, the trust engine 110 will provide such a binary authentication. In circumstances such as these, the degree of confidence required in order to secure a positive authentication is determined based upon preferences set within the trust engine 110.

[0251] However, it is also possible that the vendor may request a particular level of trust in order to complete a particular transaction. This required level may be included with the authentication request (e.g. authenticate this user to 98% confidence) or may be determined by the trust engine 110 based on other factors associated with the transaction (i.e. authenticate this user as appropriate for this transaction). One such factor might be the economic

value of the transaction. For transactions which have greater economic value, a higher degree of trust may be required. Similarly, for transactions with high degrees of risk a high degree of trust may be required. Conversely, for transactions which are either of low risk or of low value, lower trust levels may be required by the vendor or other authentication requestor.

[0252] The process of trust arbitrage occurs between the steps of the trust engine 110 receiving the authentication data in step 1050 of FIGURE 10 and the return of an authentication result to the vendor in step 1055 of FIGURE 10. Between these steps, the process which leads to the evaluation of trust levels and the potential trust arbitrage occurs as shown in FIGURE 17. In circumstances where simple binary authentication is performed, the process shown in FIGURE 17 reduces to having the transaction engine 205 directly compare the authentication data provided with the enrollment data for the identified user as discussed above with reference to FIGURE 10, flagging any difference as a negative authentication.

[0253] As shown in FIGURE 17, the first step after receiving the data in step 1050 is for the transaction engine 205 to determine the trust level which is required for a positive authentication for this particular transaction in step 1710. This step may be performed by one of several different methods. The required trust level may be specified to the trust engine 110 by the authentication requestor at the time when the authentication request is made. The authentication requestor may also set a preference in advance which is stored within the depository 210 or

other storage which is accessible by the transaction engine 205. This preference may then be read and used each time an authentication request is made by this authentication requestor. The preference may also be associated with a particular user as a security measure such that a particular level of trust is always required in order to authenticate that user, the user preference being stored in the depository 210 or other storage media accessible by the transaction engine 205. The required level may also be derived by the transaction engine 205 or authentication engine 215 based upon information provided in the authentication request, such as the value and risk level of the transaction to be authenticated.

[0254] In one mode of operation, a policy management module or other software which is used when generating the authentication request is used to specify the required degree of trust for the authentication of the transaction. This may be used to provide a series of rules to follow when assigning the required level of trust based upon the policies which are specified within the policy management module. One advantageous mode of operation is for such a module to be incorporated with the web server of a vendor in order to appropriately determine required level of trust for transactions initiated with the vendor's web server. In this way, transaction requests from users may be assigned a required trust level in accordance with the policies of the vendor and such information may be forwarded to the trust engine 110 along with the authentication request.

[0255] This required trust level correlates with the degree of certainty that the vendor wants to have that

the individual authenticating is in fact who he identifies himself as. For example, if the transaction is one where the vendor wants a fair degree of certainty because goods are changing hands, the vendor may require a trust level of 85%. For situation where the vendor is merely authenticating the user to allow him to view members only content or exercise privileges on a chat room, the downside risk may be small enough that the vendor requires only a 60% trust level. However, to enter into a production contract with a value of tens of thousands of dollars, the vendor may require a trust level of 99% or more.

[0256] This required trust level represents a metric to which the user must authenticate himself in order to complete the transaction. If the required trust level is 85% for example, the user must provide authentication to the trust engine 110 sufficient for the trust engine 110 to say with 85% confidence that the user is who they say they are. It is the balance between this required trust level and the authentication confidence level which produces either a positive authentication (to the satisfaction of the vendor) or a possibility of trust arbitrage.

[0257] As shown in FIGURE 17, after the transaction engine 205 receives the required trust level, it compares in step 1720 the required trust level to the authentication confidence level which the authentication engine 215 calculated for the current authentication (as discussed with reference to FIGURE 16). If the authentication confidence level is higher than the required trust level for the transaction in step 1730, then the process moves to step 1740 where a positive authentication for this transaction is

produced by the transaction engine 205. A message to this effect will then be inserted into the authentication results and returned to the vendor by the transaction engine 205 as shown in step 1055 (see FIGURE 10).

[0258] However, if the authentication confidence level does not fulfill the required trust level in step 1730, then a confidence gap exists for the current authentication, and trust arbitrage is conducted in step 1750. Trust arbitrage is described more completely with reference to FIGURE 18 below. This process as described below takes place within the transaction engine 205 of the trust engine 110. Because no authentication or other cryptographic operations are needed to execute trust arbitrage (other than those required for the SSL communication between the transaction engine 205 and other components), the process may be performed outside the authentication engine 215. However, as will be discussed below, any reevaluation of authentication data or other cryptographic or authentication events will require the transaction engine 205 to resubmit the appropriate data to the authentication engine 215. Those of skill in the art will recognize that the trust arbitrage process could alternately be structured to take place partially or entirely within the authentication engine 215 itself.

[0259] As mentioned above, trust arbitrage is a process where the trust engine 110 mediates a negotiation between the vendor and user in an attempt to secure a positive authentication where appropriate. As shown in step 1805, the transaction engine 205 first determines whether or not the current situation is

appropriate for trust arbitrage. This may be determined based upon the circumstances of the authentication, e.g. whether this authentication has already been through multiple cycles of arbitrage, as well as upon the preferences of either the vendor or user, as will be discussed further below.

[0260] In such circumstances where arbitrage is not possible, the process proceeds to step 1810 where the transaction engine 205 generates a negative authentication and then inserts it into the authentication results which are sent to the vendor in step 1055 (see FIGURE 10). One limit which may be advantageously used to prevent authentications from pending indefinitely is to set a time-out period from the initial authentication request. In this way, any transaction which is not positively authenticated within the time limit is denied further arbitrage and negatively authenticated. Those of skill in the art will recognize that such a time limit may vary depending upon the circumstances of the transaction and the desires of the user and vendor. Limitations may also be placed upon the number of attempts that may be made at providing a successful authentication. Such limitations may be handled by an attempt limiter 535 as shown in FIGURE 5.

[0261] If arbitrage is not prohibited in step 1805, the transaction engine 205 will then engage in negotiation with one or both of the transacting parties. The transaction engine 205 may send a message to the user requesting some form of additional authentication in order to boost the authentication confidence level produced as shown in step 1820. In the simplest form, this may simply indicates that

authentication was insufficient. A request to produce one or more additional authentication instances to improve the overall confidence level of the authentication may also be sent.

[0262] If the user provides some additional authentication instances in step 1825, then the transaction engine 205 adds these authentication instances to the authentication data for the transaction and forwards it to the authentication engine 215 as shown in step 1015 (see FIGURE 10), and the authentication is reevaluated based upon both the pre-existing authentication instances for this transaction and the newly provided authentication instances.

[0263] An additional type of authentication may be a request from the trust engine 110 to make some form of person-to-person contact between the trust engine 110 operator (or a trusted associate) and the user, for example, by phone call. This phone call or other non-computer authentication can be used to provide personal contact with the individual and also to conduct some form of questionnaire based authentication. This also may give the opportunity to verify an originating telephone number and potentially a voice analysis of the user when he calls in. Even if no additional authentication data can be provided, the additional context associated with the user's phone number may improve the reliability of the authentication context. Any revised data or circumstances based upon this phone call are fed into the trust engine 110 for use in consideration of the authentication request.

[0264] Additionally, in step 1820 the trust engine 110 may provide an opportunity for the user to purchase insurance, effectively buying a more confident authentication. The operator of the trust engine 110 may, at times, only want to make such an option available if the confidence level of the authentication is above a certain threshold to begin with. In effect, this user side insurance is a way for the trust engine 110 to vouch for the user when the authentication meets the normal required trust level of the trust engine 110 for authentication, but does not meet the required trust level of the vendor for this transaction. In this way, the user may still successfully authenticate to a very high level as may be required by the vendor, even though he only has authentication instances which produce confidence sufficient for the trust engine 110.

[0265] This function of the trust engine 110 allows the trust engine 110 to vouch for someone who is authenticated to the satisfaction of the trust engine 110, but not of the vendor. This is analogous to the function performed by a notary in adding his signature to a document in order to indicate to someone reading the document at a later time that the person whose signature appears on the document is in fact the person who signed it. The signature of the notary testifies to the act of signing by the user. In the same way, the trust engine is providing an indication that the person transacting is who they say they are.

[0266] However, because the trust engine 110 is artificially boosting the level of confidence provided by the user, there is a greater risk to the trust engine 110 operator, since the user is not actually meeting the required trust level of the vendor. The

cost of the insurance is designed to offset the risk of a false positive authentication to the trust engine 110 (who may be effectively notarizing the authentications of the user). The user pays the trust engine 110 operator to take the risk of authenticating to a higher level of confidence than has actually been provided.

[0267] Because such an insurance system allows someone to effectively buy a higher confidence rating from the trust engine 110, both vendors and users may wish to prevent the use of user side insurance in certain transactions. Vendors may wish to limit positive authentications to circumstances where they know that actual authentication data supports the degree of confidence which they require and so may indicate to the trust engine 110 that user side insurance is not to be allowed. Similarly, to protect his online identity, a user may wish to prevent the use of user side insurance on his account, or may wish to limit its use to situations where the authentication confidence level without the insurance is higher than a certain limit. This may be used as a security measure to prevent someone from overhearing a password or stealing a smart card and using them to falsely authenticate to a low level of confidence, and then purchasing insurance to produce a very high level of (false) confidence. These factors may be evaluated in determining whether user side insurance is allowed.

[0268] If user purchases insurance in step 1840, then the authentication confidence level is adjusted based upon the insurance purchased in step 1845, and the authentication confidence level and required trust level are again compared in step 1730 (see FIGURE 17). The process continues from there, and may lead to

either a positive authentication in step 1740 (see FIGURE 17), or back into the trust arbitrage process in step 1750 for either further arbitrage (if allowed) or a negative authentication in step 1810 if further arbitrage is prohibited.

[0269] In addition to sending a message to the user in step 1820, the transaction engine 205 may also send a message to the vendor in step 1830 which indicates that a pending authentication is currently below the required trust level. The message may also offer various options on how to proceed to the vendor. One of these Options is to simply inform the vendor of what the current authentication confidence level is and ask if the vendor wishes to maintain their current unfulfilled required trust level. This may be beneficial because in some cases, the vendor may have independent means for authenticating the transaction or may have been using a default set of requirements which generally result in a higher required level being initially specified than is actually needed for the particular transaction at hand.

[0270] For instance, it may be standard practice that all incoming purchase order transactions with the vendor are expected to meet a 98% trust level. However, if an order was recently discussed by phone between the vendor and a long-standing customer, and immediately thereafter the transaction is authenticated, but only to a 93% confidence level, the vendor may wish to simply lower the acceptance threshold for this transaction, because the phone call effectively provides additional authentication to the vendor. In certain circumstances, the vendor may be willing to lower their required trust level, but not

all the way to the level of the current authentication confidence. For instance, the vendor in the above example might consider that the phone call prior to the order might merit a 4% reduction in the degree of trust needed; however, this is still greater than the 93% confidence produced by the user.

[0271] If the vendor does adjust their required trust level in step 1835, then the authentication confidence level produced by the authentication and the required trust level are compared in step 1730 (see FIGURE 17). If the confidence level now exceeds the required trust level, a positive authentication may be generated in the transaction engine 205 in step 1740 (see FIGURE 17). If not, further arbitrage may be attempted as discussed above if it is permitted.

[0272] In addition to requesting an adjustment to the required trust level, the transaction engine 205 may also offer vendor side insurance to the vendor requesting the authentication. This insurance serves a similar purpose to that described above for the user side insurance. Here, however, rather than the cost corresponding to the risk being taken by the trust engine 110 in authenticating above the actual authentication confidence level produced, the cost of the insurance corresponds to the risk being taken by the vendor in accepting a lower trust level in the authentication.

[0273] Instead of just lowering their actual required trust level, the vendor has the option of purchasing insurance to protect itself from the additional risk associated with a lower level of trust in the authentication of the user. As described above, it may be advantageous for the vendor to only consider

purchasing such insurance to cover the trust gap in conditions where the existing authentication is already above a certain threshold.

[0274] The availability of such vendor side insurance allows the vendor the option to either: lower his trust requirement directly at no additional cost to himself, bearing the risk of a false authentication himself (based on the lower trust level required); or, buying insurance for the trust gap between the authentication confidence level and his requirement, with the trust engine 110 operator bearing the risk of the lower confidence level which has been provided. By purchasing the insurance, the vendor effectively keeps his high trust level requirement; because the risk of a false authentication is shifted to the trust engine 110 operator.

[0275] If the vendor purchases insurance in step 1840, the authentication confidence level and required trust levels are compared in step 1730 (see FIGURE 17), and the process continues as described above.

[0276] Note that it is also possible that both the user and the vendor respond to messages from the trust engine 110. Those of skill in the art will recognize that there are multiple ways in which such situations can be handled. One advantageous mode of handling the possibility of multiple responses is simply to treat the responses in a first-come, first-served manner. For example, if the vendor responds with a lowered required trust level and immediately thereafter the user also purchases insurance to raise his authentication level, the authentication is first reevaluated based upon the lowered trust requirement from the vendor. If the authentication is now

positive, the user's insurance purchase is ignored. In another advantageous mode of operation, the user might only be charged for the level of insurance required to meet the new, lowered trust requirement of the vendor (if a trust gap remained even with the lowered vendor trust requirement).

[0277] If no response from either party is received during the trust arbitrage process at step 1850 within the time limit set for the authentication, the arbitrage is reevaluated in step 1805. This effectively begins the arbitrage process again. If the time limit was final or other circumstances prevent further arbitrage in step 1805, a negative authentication is generated by the transaction engine 205 in step 1810 and returned to the vendor in step 1055 (see FIGURE 10). If not, new messages may be sent to the user and vendor, and the process may be repeated as desired.

[0278] Note that for certain types of transactions, for instance, digitally signing documents which are not part of a transaction, there may not necessarily be a vendor or other third party; therefore the transaction is primarily between the user and the trust engine 110. In circumstances such as these, the trust engine 110 will have its own required trust level which must be satisfied in order to generate a positive authentication. However, in such circumstances, it will often not be desirable for the trust engine 110 to offer insurance to the user in order for him to raise the confidence of his own signature.

[0279] The process described above and shown in FIGURES 16-18 may be carried out using various communications modes as described above with reference

to the trust engine 110. For instance, the messages may be web-based and sent using SSL connections between the trust engine 110 and applets downloaded in real time to browsers running on the user or vendor systems. In an alternate mode of operation, certain dedicated applications may be in use by the user and vendor which facilitate such arbitrage and insurance transactions. In another alternate mode of operation, secure email operations may be used to mediate the arbitrage described above, thereby allowing deferred evaluations and batch processing of authentications. Those of skill in the art will recognize that different communications modes may be used as are appropriate for the circumstances and authentication requirements of the vendor.

[0280] The following description with reference to FIGURE 19 describes a sample transaction which integrates the various aspects of the present invention as described above. This example illustrates the overall process between a user and a vendor as mediates by the trust engine 110. Although the various steps and components as described in detail above may be used to carry out the following transaction, the process illustrated focuses on the interaction between the trust engine 110, user and vendor.

[0281] The transaction begins when the user, while viewing web pages online, fills out an order form on the web site of the vendor in step 1900. The user wishes to submit this order form to the vendor, signed with his digital signature. In order to do this, the user submits the order form with his request for a signature to the trust engine 110 in step 1905. The user will also provide authentication data which will

be used as described above to authenticate his identity.

[0282] In step 1910 the authentication data is compared to the enrollment data by the trust engine 110 as discussed above, and if a positive authentication is produced, the hash of the order form, signed with the private key of the user, is forwarded to the vendor along with the order form itself.

[0283] The vendor receives the signed form in step 1915, and then the vendor will generate an invoice or other contract related to the purchase to be made in step 1920. This contract is sent back to the user with a request for a signature in step 1925. The vendor also sends an authentication request for this contract transaction to the trust engine 110 in step 1930 including a hash of the contract which will be signed by both parties. To allow the contract to be digitally signed by both parties, the vendor also includes authentication data for itself so that the vendor's signature upon the contract can later be verified if necessary.

[0284] As discussed above, the trust engine 110 then verifies the authentication data provided by the vendor to confirm the vendor's identity, and if the data produces a positive authentication in step 1935, continues with step 1955 when the data is received from the user. If the vendor's authentication data does not match the enrollment data of the vendor to the desired degree, a message is returned to the vendor requesting further authentication. Trust arbitrage may be performed here if necessary, as described above, in order for the vendor to successfully authenticate itself to the trust engine 110.

[0285] When the user receives the contract in step 1940, he reviews it, generates authentication data to sign it if it is acceptable in step 1945, and then sends a hash of the contract and his authentication data to the trust engine 110 in step 1950. The trust engine 110 verifies the authentication data in step 1955 and if the authentication is good, proceeds to process the contract as described below. As discussed above with reference to FIGURES 17 and 18, trust arbitrage may be performed as appropriate to close any trust gap which exists between the authentication confidence level and the required authentication level for the transaction.

[0286] The trust engine 110 signs the hash of the contract with the user's private key, and sends this signed hash to the vendor in step 1960, signing the complete message on its own behalf, i.e., including a hash of the complete message (including the user's signature) encrypted with the private key 510 of the trust engine 110. This message is received by the vendor in step 1965. The message represents a signed contract (hash of contract encrypted using user's private key) and a receipt from the trust engine 110 (the hash of the message including the signed contract, encrypted using the trust engine 110's private key).

[0287] The trust engine 110 similarly prepares a hash of the contract with the vendor's private key in step 1970, and forwards this to the user, signed by the trust engine 110. In this way, the user also receives a copy of the contract, signed by the vendor, as well as a receipt, signed by the trust engine 110, for delivery of the signed contract in step 1975.

[0288] In addition to the foregoing, an additional aspect of the invention provides a cryptographic Service Provider Module (SPM) which may be available to a client side application as a means to access functions provided by the trust engine 110 described above. One advantageous way to provide such a service is for the cryptographic SPM is to mediate communications between a third party Application Programming Interface (API) and a trust engine 110 which is accessible via a network or other remote connection. A sample cryptographic SPM is described below with reference to FIGURE 20.

[0289] For example, on a typical system, a number of API's are available to programmers. Each API provides a set of function calls which may be made by an application 2000 running upon the system. Examples of API's which provide programming interfaces suitable for cryptographic functions, authentication functions, and other security function include the Cryptographic API (CAPI) 2010 provided by Microsoft with its Windows operating systems, and the Common Data Security Architecture (CDSA), sponsored by IBM, Intel and other members of the Open Group. CAPI will be used as an exemplary security API in the discussion that follows. However, the cryptographic SPM described could be used with CDSA or other security API's as are known in the art.

[0290] This API is used by a user system 105 or vendor system 120 when a call is made for a cryptographic function. Included among these functions may be requests associated with performing various cryptographic operations, such as encrypting a document with a particular key, signing a document, requesting a

digital certificate, verifying a signature upon a signed document, and such other cryptographic functions as are described herein or known to those of skill in the art.

[0291] Such cryptographic functions are normally performed locally to the system upon which CAPI 2010 is located. This is because generally the functions called require the use of either resources of the local user system 105, such as a fingerprint reader, or software functions which are programmed using libraries which are executed on the local machine. Access to these local resources is normally provided by one or more Service Provider Modules (SPM's) 2015, 2020 as referred to above which provide resources with which the cryptographic functions are carried out. Such SPM's may include software libraries 2015 to perform encrypting or decrypting operations, or drivers and applications 2020 which are capable of accessing specialized hardware 2025, such as biometric scanning devices. In much the way that CAPI 2010 provides functions which may be used by an application 2000 of the system 105, the SPM's 2015, 2020 provide CAPI with access to the lower level functions and resources associated with the available services upon the system.

[0292] In accordance with the invention, it is possible to provide a cryptographic SPM 2030 which is capable of accessing the cryptographic functions provided by the trust engine 110 and making these functions available to an application 2000 through CAPI 2010. Unlike embodiments where CAPI 2010 is only able to access resources which are locally available through SPM's 2015, 2020, a cryptographic SPM 2030 as described herein would be able to submit requests for

cryptographic operations to a remotely-located, network-accessible trust engine 110 in order to perform the operations desired.

[0293] For instance, if an application 2000 has a need for a cryptographic operation, such as signing a document, the application 2000 makes a function call to the appropriate CAPI 2010 function. CAPI 2010 in turn will execute this function, making use of the resources which are made available to it by the SPM's 2015, 2020 and the cryptographic SPM 2030. In the case of a digital signature function, the cryptographic SPM 2030 will generate an appropriate request which will be sent to the trust engine 110 across the communication link 125.

[0294] The operations which occur between the cryptographic SPM 2030 and the trust engine 110 are the same operations that would be possible between any other system and the trust engine 110. However, these functions are effectively made available to a user system 105 through CAPI 2010 such that they appear to be locally available upon the user system 105 itself. However, unlike ordinary SPM's 2015, 2020, the functions are being carried out on the remote trust engine 110 and the results relayed to the cryptographic SPM 2030 in response to appropriate requests across the communication link 125.

[0295] This cryptographic SPM 2030 makes a number of operations available to the user system 105 or a vendor system 120 which might not otherwise be available. These functions include without limitation: encryption and decryption of documents; issuance of digital certificates; digital signing of documents; verification of digital signatures; and such other

operations as will be apparent to those of skill in the art.

[0296] In a separate embodiment, the present invention comprises a complete system for performing the data securing methods of the present invention on any data set. The computer system of this embodiment comprises a data splitting module that comprises the functionality shown in FIGURE 8 and described herein. In one embodiment of the present invention, the data splitting module, sometimes referred to herein as a secure data parser, comprises a parser program or software suite which comprises data splitting, encryption and decryption, reconstitution or reassembly functionality. This embodiment may further comprise a data storage facility or multiple data storage facilities, as well. The data splitting module, or secure data parser, comprises a cross-platform software module suite which integrates within an electronic infrastructure, or as an add-on to any application which requires the ultimate security of its data elements. This parsing process operates on any type of data set, and on any and all file types, or in a database on any row, column or cell of data in that database.

[0297] The parsing process of the present invention may, in one embodiment, be designed in a modular tiered fashion, and any encryption process is suitable for use in the process of the present invention. The modular tiers of the parsing and splitting process of the present invention may include, but are not limited to, 1) cryptographic split, dispersed and securely stored in multiple locations; 2) encrypt, cryptographically split, dispersed and securely stored in multiple

locations; 3) encrypt, cryptographically split, encrypt each share, then dispersed and securely stored in multiple locations; and 4) encrypt, cryptographically split, encrypt each share with a different type of encryption than was used in the first step, then dispersed and securely stored in multiple locations.

[0298] The process comprises, in one embodiment, splitting of the data according to the contents of a generated random number, or key and performing the same cryptographic splitting of the key used in the encryption of splitting of the data to be secured into two or more portions, or shares, of parsed and split data, and in one embodiment, preferably into four or more portions of parsed and split data, encrypting all of the portions, then scattering and storing these portions back into the database, or relocating them to any named device, fixed or removable, depending on the requestor's need for privacy and security. Alternatively, in another embodiment, encryption may occur prior to the splitting of the data set by the splitting module or secure data parser. The original data processed as described in this embodiment is encrypted and obfuscated and is secured. The dispersion of the encrypted elements, if desired, can be virtually anywhere, including, but not limited to, a single server or data storage device, or among separate data storage facilities or devices. Encryption key management in one embodiment may be included within the software suite, or in another embodiment may be integrated into an existing infrastructure or any other desired location.

[0299] A cryptographic split (cryptosplit) partitions the data into N number of shares. The

partitioning can be on any size unit of data, including an individual bit, bits, bytes, kilobytes, megabytes, or larger units, as well as any pattern or combination of data unit sizes whether predetermined or randomly generated. The units can also be of different sized, based on either a random or predetermined set of values. This means the data can be viewed as a sequence of these units. In this manner the size of the data units themselves may render the data more secure, for example by using one or more predetermined or randomly generated pattern, sequence or combination of data unit sizes. The units are then distributed (either randomly or by a predetermined set of values) into the N shares. This distribution could also involve a shuffling of the order of the units in the shares. It is readily apparent to those of ordinary skill in the art that the distribution of the data units into the shares may be performed according to a wide variety of possible selections, including but not limited to size-fixed, predetermined sizes, or one or more combination, pattern or sequence of data unit sizes that are predetermined or randomly generated.

[0300] One example of this cryptographic split process, or cryptosplit, would be to consider the data to be 23 bytes in size, with the data unit size chosen to be one byte, and with the number of shares selected to be 4. Each byte would be distributed into one of the 4 shares. Assuming a random distribution, a key would be obtained to create a sequence of 23 random numbers (r1, r2, r3 through r23), each with a value between 1 and 4 corresponding to the four shares. Each of the units of data (in this example 23 individual bytes of data) is associated with one of the 23 random

numbers corresponding to one of the four shares. The distribution of the bytes of data into the four shares would occur by placing the first byte of the data into share number r1, byte two into share r2, byte three into share r3, through the 23rd byte of data into share r23. It is readily apparent to those of ordinary skill in the art that a wide variety of other possible steps or combination or sequence of steps, including the size of the data units, may be used in the cryptosplit process of the present invention, and the above example is a non-limiting description of one process for cryptosplitting data. To recreate the original data, the reverse operation would be performed.

[0301] In another embodiment of the cryptosplit process of the present invention, an option for the cryptosplitting process is to provide sufficient redundancy in the shares such that only a subset of the shares are needed to reassemble or restore the data to its original or useable form. As a non-limiting example, the cryptosplit may be done as a "3 of 4" cryptosplit such that only three of the four shares are necessary to reassemble or restore the data to its original or useable form. This is also referred to as a "M of N cryptosplit" wherein N is the total number of shares, and M is at least one less than N. It is readily apparent to those of ordinary skill in the art that there are many possibilities for creating this redundancy in the cryptosplitting process of the present invention.

[0302] In one embodiment of the cryptosplitting process of the present invention, each unit of data is stored in two shares, the primary share and the backup share. Using the "3 of 4" cryptosplitting process

described above, any one share can be missing, and this is sufficient to reassemble or restore the original data with no missing data units since only three of the total four shares are required. As described herein, a random number is generated that corresponds to one of the shares. The random number is associated with a data unit, and stored in the corresponding share, based on a key. One key is used, in this embodiment, to generate the primary and backup share random number. As described herein for the cryptosplitting process of the present invention, a set of random numbers (also referred to as primary share numbers) from 0 to 3 are generated equal to the number of data units. Then another set of random numbers is generated (also referred to as backup share numbers) from 1 to 3 equal to the number of data units. Each unit of data is then associated with a primary share number and a backup share number. Alternatively, a set of random numbers may be generated that is fewer than the number of data units, and repeating the random number set, but this may reduce the security of the sensitive data. The primary share number is used to determine into which share the data unit is stored. The backup share number is combined with the primary share number to create a third share number between 0 and 3, and this number is used to determine into which share the data unit is stored. In this example, the equation to determine the third share number is:

(primary share number + backup share number) MOD 4 = third share number.

[0303] In the embodiment described above where the primary share number is between 0 and 3, and the backup share number is between 1 and 3 ensures that the third

share number is different from the primary share number. This results in the data unit being stored in two different shares. It is readily apparent to those of ordinary skill in the art that there are many ways of performing redundant cryptosplitting and non-redundant cryptosplitting in addition to the embodiments disclosed herein. For example, the data units in each share could be shuffled utilizing a different algorithm. This data unit shuffling may be performed as the original data is split into the data units, or after the data units are placed into the shares, or after the share is full, for example.

[0304] The various cryptosplitting processes and data shuffling processes described herein, and all other embodiments of the cryptosplitting and data shuffling methods of the present invention may be performed on data units of any size, including but not limited to, as small as an individual bit, bits, bytes, kilobytes, megabytes or larger.

[0305] An example of one embodiment of source code that would perform the cryptosplitting process described herein is:

```
DATA [1:24] - array of bytes with the data to be split
SHARES[0:3; 1:24] - 2-dimensionalarray with each row
representing one of the shares
RANDOM[1:24] - array random numbers in the range of
0..3
S1 = 1;
S2 = 1;
S3 = 1;
S4 = 1;

For J = 1 to 24 do
```

```
Begin
IF RANDOM[J[ ==0 then
     Begin
     SHARES[1,S1] = DATA [J];
     S1 = S1 + 1;
     End
ELSE IF RANDOM[J[ ==1 then
     Begin
     SHARES[2,S2] = DATA [J];
     S2 = S2 + 1;
     END
ELSE IF RANDOM[J[ ==2 then
     Begin
     Shares[3,S3] = data [J];
     S3 = S3 + 1;
     End
Else   begin
     Shares[4,S4] = data [J];
     S4 = S4 + 1;
     End;
END;
```

[0306] An example of one embodiment of source code that would perform the cryptosplitting RAID process described herein is:

[0307] Generate two sets of numbers, PrimaryShare is 0 to 3, BackupShare is 1 to 3. Then put each data unit into share[primaryshare[1]] and share[(primaryshare[1]+backupshare[1]) mod 4, with the same process as in cryptosplitting described above. This method will be scalable to any size N, where only N-1 shares are necessary to restore the data.

[0308] The retrieval, recombining, reassembly or reconstituting of the encrypted data elements may

utilize any number of authentication techniques, including, but not limited to, biometrics, such as fingerprint recognition, facial scan, hand scan, iris scan, retinal scan, ear scan, vascular pattern recognition or DNA analysis. The data splitting and/or parser modules of the present invention may be integrated into a wide variety of infrastructure products or applications as desired.

[0309] Traditional encryption technologies known in the art rely on one or more key used to encrypt the data and render it unusable without the key. The data, however, remains whole and intact and subject to attack. The secure data parser of the present invention, in one embodiment, addresses this problem by performing a cryptographic parsing and splitting of the encrypted file into two or more portions or shares, and in another embodiment, preferably four or more shares, adding another layer of encryption to each share of the data, then storing the shares in different physical and/or logical locations. When one or more data shares are physically removed from the system, either by using a removable device, such as a data storage device, or by placing the share under another party's control, any possibility of compromise of secured data is effectively removed.

[0310] An example of one embodiment of the secure data parser of the present invention and an example of how it may be utilized is shown in FIGURE 21 and described below. However, it is readily apparent to those of ordinary skill in the art that the secure data parser of the present invention may be utilized in a wide variety of ways in addition to the non-limiting example below. As a deployment option, and in one

embodiment, the secure data parser may be implemented with external session key management or secure internal storage of session keys. Upon implementation, a Parser Master Key will be generated which will be used for securing the application and for encryption purposes. It should be also noted that the incorporation of the Parser Master key in the resulting secured data allows for a flexibility of sharing of secured data by individuals within a workgroup, enterprise or extended audience.

[0311] As shown in Figure 21, this embodiment of the present invention shows the steps of the process performed by the secure data parser on data to store the session master key with the parsed data:

[0312] 1. Generating a session master key and encrypt the data using RS1 stream cipher.

[0313] 2. Separating the resulting encrypted data into four shares or portions of parsed data according to the pattern of the session master key.

[0314] 3. In this embodiment of the method, the session master key will be stored along with the secured data shares in a data depository. Separating the session master key according to the pattern of the Parser Master Key and append the key data to the encrypted parsed data.

[0315] 4. The resulting four shares of data will contain encrypted portions of the original data and portions of the session master key. Generate a stream cipher key for each of the four data shares.

[0316] 5. Encrypting each share, then store the encryption keys in different locations from the encrypted data portions or shares: Share 1 gets Key 4,

Share 2 gets Key 1, Share 3 gets Key 2, Share 4 gets Key 3.

[0317] To restore the original data format, the steps are reversed.

[0318] It is readily apparent to those of ordinary skill in the art that certain steps of the methods described herein may be performed in different order, or repeated multiple times, as desired. It is also readily apparent to those skilled in the art that the portions of the data may be handled differently from one another. For example, multiple parsing steps may be performed on only one portion of the parsed data. Each portion of parsed data may be uniquely secured in any desirable way provided only that the data may be reassembled, reconstituted, reformed, decrypted or restored to its original or other usable form.

[0319] As shown in FIGURE 22 and described herein, another embodiment of the present invention comprises the steps of the process performed by the secure data parser on data to store the session master key data in one or more separate key management table:

[0320] 1. Generating a session master key and encrypt the data using RS1 stream cipher.

[0321] 2. Separating the resulting encrypted data into four shares or portions of parsed data according to the pattern of the session master key.

[0322] 3. In this embodiment of the method of the present invention, the session master key will be stored in a separate key management table in a data depository. Generating a unique transaction ID for this transaction. Storing the transaction ID and session master key in a separate key management table. Separating the transaction ID according to the pattern

of the Parser Master Key and append the data to the encrypted parsed or separated data.

[0323] 4. The resulting four shares of data will contain encrypted portions of the original data and portions of the transaction ID.

[0324] 5. Generating a stream cipher key for each of the four data shares.

[0325] 6. Encrypting each share, then store the encryption keys in different locations from the encrypted data portions or shares: Share 1 gets Key 4, Share 2 gets Key 1, Share 3 gets Key 2, Share 4 gets Key 3.

[0326] To restore the original data format, the steps are reversed.

[0327] It is readily apparent to those of ordinary skill in the art that certain steps of the method described herein may be performed in different order, or repeated multiple times, as desired. It is also readily apparent to those skilled in the art that the portions of the data may be handled differently from one another. For example, multiple separating or parsing steps may be performed on only one portion of the parsed data. Each portion of parsed data may be uniquely secured in any desirable way provided only that the data may be reassembled, reconstituted, reformed, decrypted or restored to its original or other usable form.

[0328] As shown in Figure 23, this embodiment of the present invention shows the steps of the process performed by the secure data parser on data to store the session master key with the parsed data:

[0329] 1. Accessing the parser master key associated with the authenticated user

[0330] 2. Generating a unique Session Master key

[0331] 3. Derive an Intermediary Key from an exclusive OR function of the Parser Master Key and Session Master key

[0332] 4. Optional encryption of the data using an existing or new encryption algorithm keyed with the Intermediary Key.

[0333] 5. Separating the resulting optionally encrypted data into four shares or portions of parsed data according to the pattern of the Intermediary key.

[0334] 6. In this embodiment of the method, the session master key will be stored along with the secured data shares in a data depository. Separating the session master key according to the pattern of the Parser Master Key and append the key data to the optionally encrypted parsed data shares.

[0335] 7. The resulting multiple shares of data will contain optionally encrypted portions of the original data and portions of the session master key.

[0336] 8. Optionally generate an encryption key for each of the four data shares.

[0337] 9. Optionally encrypting each share with an existing or new encryption algorithm, then store the encryption keys in different locations from the encrypted data portions or shares: for example, Share 1 gets Key 4, Share 2 gets Key 1, Share 3 gets Key 2, Share 4 gets Key 3.

[0338] To restore the original data format, the steps are reversed.

[0339] It is readily apparent to those of ordinary skill in the art that certain steps of the methods described herein may be performed in different order, or repeated multiple times, as desired. It is also

readily apparent to those skilled in the art that the portions of the data may be handled differently from one another. For example, multiple parsing steps may be performed on only one portion of the parsed data. Each portion of parsed data may be uniquely secured in any desirable way provided only that the data may be reassembled, reconstituted, reformed, decrypted or restored to its original or other usable form.

[0340] As shown in FIGURE 24 and described herein, another embodiment of the present invention comprises the steps of the process performed by the secure data parser on data to store the session master key data in one or more separate key management table:

[0341] 1. Accessing the Parser Master Key associated with the authenticated user

[0342] 2. Generating a unique Session Master Key

[0343] 3. Derive an Intermediary Key from an exclusive OR function of the Parser Master Key and Session Master key

[0344] 4. Optionally encrypt the data using an existing or new encryption algorithm keyed with the Intermediary Key.

[0345] 5. Separating the resulting optionally encrypted data into four shares or portions of parsed data according to the pattern of the Intermediary Key.

[0346] 6. In this embodiment of the method of the present invention, the session master key will be stored in a separate key management table in a data depository. Generating a unique transaction ID for this transaction. Storing the transaction ID and session master key in a separate key management table or passing the Session Master Key and transaction ID back to the calling program for external management.

Separating the transaction ID according to the pattern of the Parser Master Key and append the data to the optionally encrypted parsed or separated data.

[0347] 7. The resulting four shares of data will contain optionally encrypted portions of the original data and portions of the transaction ID.

[0348] 8. Optionally generate an encryption key for each of the four data shares.

[0349] 9. Optionally encrypting each share, then store the encryption keys in different locations from the encrypted data portions or shares. For example: Share 1 gets Key 4, Share 2 gets Key 1, Share 3 gets Key 2, Share 4 gets Key 3.

[0350] To restore the original data format, the steps are reversed.

[0351] It is readily apparent to those of ordinary skill in the art that certain steps of the method described herein may be performed in different order, or repeated multiple times, as desired. It is also readily apparent to those skilled in the art that the portions of the data may be handled differently from one another. For example, multiple separating or parsing steps may be performed on only one portion of the parsed data. Each portion of parsed data may be uniquely secured in any desirable way provided only that the data may be reassembled, reconstituted, reformed, decrypted or restored to its original or other usable form.

[0352] A wide variety of encryption methodologies are suitable for use in the methods of the present invention, as is readily apparent to those skilled in the art. The One Time Pad algorithm, is often considered one of the most secure encryption methods,

and is suitable for use in the method of the present invention. Using the One Time Pad algorithm requires that a key be generated which is as long as the data to be secured. The use of this method may be less desirable in certain circumstances such as those resulting in the generation and management of very long keys because of the size of the data set to be secured. In the One-Time Pad (OTP) algorithm, the simple exclusive-or function, XOR, is used. For two binary streams x and y of the same length, x XOR y means the bitwise exclusive-or of x and y.

[0353] At the bit level is generated:

0 XOR 0 = 0

0 XOR 1 = 1

1 XOR 0 = 1

1 XOR 1 = 0

[0354] An example of this process is described herein for an n-byte secret, s, (or data set) to be split. The process will generate an n-byte random value, a, and then set:

b = a XOR s.

[0355] Note that one can derive "s" via the equation:

s = a XOR b.

[0356] The values a and b are referred to as shares or portions and are placed in separate depositories. Once the secret s is split into two or more shares, it is discarded in a secure manner.

[0357] The secure data parser of the present invention may utilize this function, performing multiple XOR functions incorporating multiple distinct secret key values: K1, K2, K3, Kn, K5. At the beginning of the operation, the data to be secured is

passed through the first encryption operation, secure data = data XOR secret key 5:

S = D XOR K5

[0358] In order to securely store the resulting encrypted data in, for example, four shares, S1, S2, S3, Sn, the data is parsed and split into "n" segments, or shares, according to the value of K5. This operation results in "n" pseudorandom shares of the original encrypted data. Subsequent XOR functions may then be performed on each share with the remaining secret key values, for example: Secure data segment 1 = encrypted data share 1 XOR secret key 1:

SD1 = S1 XOR K1

SD2 = S2 XOR K2

SD3 = S3 XOR K3

SDn = Sn XOR Kn.

[0359] In one embodiment, it may not be desired to have any one depository contain enough information to decrypt the information held there, so the key required to decrypt the share is stored in a different data depository:

Depository 1: SD1, Kn

Depository 2: SD2, K1

Depository 3: SD3, K2

Depository n: SDn, K3.

[0360] Additionally, appended to each share may be the information required to retrieve the original session encryption key, K5. Therefore, in the key management example described herein, the original session master key is referenced by a transaction ID split into "n" shares according to the contents of the installation dependant Parser Master Key (TID1, TID2, TID3, TIDn):

Depository 1: SD1, Kn, TID1
Depository 2: SD2, K1, TID2
Depository 3: SD3, K2, TID3
Depository n: SDn, K3, TIDn.

[0361] In the incorporated session key example described herein, the session master key is split into "n" shares according to the contents of the installation dependant Parser Master Key (SK1, SK2, SK3, SKn):

Depository 1: SD1, Kn, SK1
Depository 2: SD2, K1, SK2
Depository 3: SD3, K2, SK3
Depository n: SDn, K3, SKn.

[0362] Unless all four shares are retrieved, the data cannot be reassembled according to this example. Even if all four shares are captured, there is no possibility of reassembling or restoring the original information without access to the session master key and the Parser Master Key.

[0363] This example has described an embodiment of the method of the present invention, and also describes, in another embodiment, the algorithm used to place shares into depositories so that shares from all depositories can be combined to form the secret authentication material. The computations needed are very simple and fast. However, with the One Time Pad (OTP) algorithm there may be circumstances that cause it to be less desirable, such as a large data set to be secured, because the key size is the same size as the data to be stored. Therefore, there would be a need to store and transmit about twice the amount of the original data which may be less desirable under certain circumstances.

Stream Cipher RS1

[0364] The stream cipher RS1 splitting technique is very similar to the OTP splitting technique described herein. Instead of an n-byte random value, an n' = min(n, 16)-byte random value is generated and used to key the RS1 Stream Cipher algorithm. The advantage of the RS1 Stream Cipher algorithm is that a pseudorandom key is generated from a much smaller seed number. The speed of execution of the RS1 Stream Cipher encryption is also rated at approximately 10 times the speed of the well known in the art Triple DES encryption without compromising security. The RS1 Stream Cipher algorithm is well known in the art, and may be used to generate the keys used in the XOR function. The RS1 Stream Cipher algorithm is interoperable with other commercially available stream cipher algorithms, such as the RC4™ stream cipher algorithm of RSA Security, Inc and is suitable for use in the methods of the present invention.

[0365] Using the key notation above, K1 thru K5 are now an n' byte random values and we set:

SD1 = S1 XOR E(K1)

SD2 = S2 XOR E(K2)

SD3 = S3 XOR E(K3)

SDn = Sn XOR E(Kn)

where E(K1) thru E(Kn) are the first n' bytes of output from the RS1 Stream Cipher algorithm keyed by K1 thru Kn. The shares are now placed into data depositories as described herein.

[0366] In this stream cipher RS1 algorithm, the required computations needed are nearly as simple and fast as the OTP algorithm. The benefit in this example

using the RS1 Stream Cipher is that the system needs to store and transmit on average only about 16 bytes more than the size of the original data to be secured per share. When the size of the original data is more than 16 bytes, this RS1 algorithm is more efficient than the OTP algorithm because it is simply shorter. It is readily apparent to those of ordinary skill in the art that a wide variety of encryption methods or algorithms are suitable for use in the present invention, including, but not limited to RS1, OTP, RC4™, Triple DES and AES.

[0367] There are major advantages provided by the data security methods and computer systems of the present invention over traditional encryption methods. One advantage is the security gained from moving shares of the data to different locations on one or more data depositories or storage devices, that may be in different logical, physical or geographical locations. When the shares of data are split physically and under the control of different personnel, for example, the possibility of compromising the data is greatly reduced.

[0368] Another advantage provided by the methods and system of the present invention is the combination of the steps of the method of the present invention for securing data to provide a comprehensive process of maintaining security of sensitive data. The data is encrypted with a secure key and split into one or more shares, and in one embodiment, four shares, according to the secure key. The secure key is stored safely with a reference pointer which is secured into four shares according to a secure key. The data shares are then encrypted individually and the keys are stored

safely with different encrypted shares. When combined, the entire process for securing data according to the methods disclosed herein becomes a comprehensive package for data security.

[0369] The data secured according to the methods of the present invention is readily retrievable and restored, reconstituted, reassembled, decrypted, or otherwise returned into its original or other suitable form for use. In order to restore the original data, the following items may be utilized:

[0370] 1. All shares or portions of the data set.

[0371] 2. Knowledge of and ability to reproduce the process flow of the method used to secure the data.

[0372] 3. Access to the session master key.

[0373] 4. Access to the Parser Master Key.

[0374] Therefore, it may be desirable to plan a secure installation wherein at least one of the above elements may be physically separated from the remaining components of the system (under the control of a different system administrator for example).

[0375] Protection against a rogue application invoking the data securing methods application may be enforced by use of the Parser Master Key. A mutual authentication handshake between the secure data parser and the application may be required in this embodiment of the present invention prior to any action taken.

[0376] The security of the system dictates that there be no "backdoor" method for recreation of the original data. For installations where data recovery issues may arise, the secure data parser can be enhanced to provide a mirror of the four shares and session master key depository. Hardware options such as RAID (redundant array of inexpensive disks, used to

spread information over several disks) and software options such as replication can assist as well in the data recovery planning.

Key Management

[0377] In one embodiment of the present invention, the data securing method uses three sets of keys for an encryption operation. Each set of keys may have individual key storage, retrieval, security and recovery options, based on the installation. The keys that may be used, include, but are not limited to:

The Parser Master Key

[0378] This key is an individual key associated with the installation of the secure data parser. It is installed on the server on which the secure data parser has been deployed. There are a variety of options suitable for securing this key including, but not limited to, a smart card, separate hardware key store, standard key stores, custom key stores or within a secured database table, for example.

The Session Master Key

[0379] A Session Master Key may be generated each time data is secured. The Session Master Key is used to encrypt the data prior to the parsing and splitting operations. It may also be incorporated (if the Session Master Key is not integrated into the parsed data) as a means of parsing the encrypted data. The Session Master Key may be secured in a variety of manners, including, but not limited to, a standard key store, custom key store, separate database table, or secured within the encrypted shares, for example.

The Share Encryption Keys

[0380] For each share or portions of a data set that is created, an individual Share Encryption Key may be generated to further encrypt the shares. The Share Encryption Keys may be stored in different shares than the share that was encrypted.

[0381] It is readily apparent to those of ordinary skill in the art that the data securing methods and computer system of the present invention are widely applicable to any type of data in any setting or environment. In addition to commercial applications conducted over the Internet or between customers and vendors, the data securing methods and computer systems of the present invention are highly applicable to non-commercial or private settings or environments. Any data set that is desired to be kept secure from any unauthorized user may be secured using the methods and systems described herein. For example, access to a particular database within a company or organization may be advantageously restricted to only selected users by employing the methods and systems of the present invention for securing data. Another example is the generation, modification or access to documents wherein it is desired to restrict access or prevent unauthorized or accidental access or disclosure outside a group of selected individuals, computers or workstations. These and other examples of the ways in which the methods and systems of data securing of the present invention are applicable to any non-commercial or commercial environment or setting for any setting, including, but not limited to any organization, government agency or corporation.

[0382] In another embodiment of the present invention, the data securing method uses three sets of

keys for an encryption operation. Each set of keys may have individual key storage, retrieval, security and recovery options, based on the installation. The keys that may be used, include, but are not limited to:

1. The Parser Master Key

[0383] This key is an individual key associated with the installation of the secure data parser. It is installed on the server on which the secure data parser has been deployed. There are a variety of options suitable for securing this key including, but not limited to, a smart card, separate hardware key store, standard key stores, custom key stores or within a secured database table, for example.

2. The Session Master Key

[0384] A Session Master Key may be generated each time data is secured. The Session Master Key is used in conjunction with the Parser Master key to derive the Intermediary Key. The Session Master Key may be secured in a variety of manners, including, but not limited to, a standard key store, custom key store, separate database table, or secured within the encrypted shares, for example.

3. The Intermediary Key

[0385] An Intermediary Key may be generated each time data is secured. The Intermediary Key is used to encrypt the data prior to the parsing and splitting operation. It may also be incorporated as a means of parsing the encrypted data.

4. The Share Encryption Keys

[0386] For each share or portions of a data set that is created, an individual Share Encryption Key may be generated to further encrypt the shares. The Share

Encryption Keys may be stored in different shares than the share that was encrypted.

[0387] It is readily apparent to those of ordinary skill in the art that the data securing methods and computer system of the present invention are widely applicable to any type of data in any setting or environment. In addition to commercial applications conducted over the Internet or between customers and vendors, the data securing methods and computer systems of the present invention are highly applicable to non-commercial or private settings or environments. Any data set that is desired to be kept secure from any unauthorized user may be secured using the methods and systems described herein. For example, access to a particular database within a company or organization may be advantageously restricted to only selected users by employing the methods and systems of the present invention for securing data. Another example is the generation, modification or access to documents wherein it is desired to restrict access or prevent unauthorized or accidental access or disclosure outside a group of selected individuals, computers or workstations. These and other examples of the ways in which the methods and systems of data securing of the present invention are applicable to any non-commercial or commercial environment or setting for any setting, including, but not limited to any organization, government agency or corporation.

Workgroup, Project, Individual PC/Laptop or Cross Platform Data Security

[0388] The data securing methods and computer systems of the present invention are also useful in securing data by workgroup, project, individual

PC/Laptop and any other platform that is in use in, for example, businesses, offices, government agencies, or any setting in which sensitive data is created, handled or stored. The present invention provides methods and computer systems to secure data that is known to be sought after by organizations, such as the U.S. Government, for implementation across the entire government organization or between governments at a state or federal level.

[0389] The data securing methods and computer systems of the present invention provide the ability to not only parse and split flat files but also data fields, sets and or table of any type. Additionally, all forms of data are capable of being secured under this process, including, but not limited to, text, video, images, biometrics and voice data. Scalability, speed and data throughput of the methods of securing data of the present invention are only limited to the hardware the user has at their disposal.

[0390] In one embodiment of the present invention, the data securing methods are utilized as described below in a workgroup environment. In one embodiment, as shown in FIGURE 23 and described below, the Workgroup Scale data securing method of the present invention uses the private key management functionality of the TrustEngine to store the user/group relationships and the associated private keys (Parser Group Master Keys) necessary for a group of users to share secure data. The method of the present invention has the capability to secure data for an enterprise, workgroup, or individual user, depending on how the Parser Master Key was deployed.

[0391] In one embodiment, additional key management and user/group management programs may be provided, enabling wide scale workgroup implementation with a single point of administration and key management. Key generation, management and revocation are handled by the single maintenance program, which all become especially important as the number of users increase. In another embodiment, key management may also be set up across one or several different system administrators, which may not allow any one person or group to control data as needed. This allows for the management of secured data to be obtained by roles, responsibilities, membership, rights, etc., as defined by an organization, and the access to secured data can be limited to just those who are permitted or required to have access only to the portion they are working on, while others, such as managers or executives, may have access to all of the secured data. This embodiment allows for the sharing of secured data among different groups within a company or organization while at the same time only allowing certain selected individuals, such as those with the authorized and predetermined roles and responsibilities, to observe the data as a whole. In addition, this embodiment of the methods and systems of the present invention also allows for the sharing of data among, for example, separate companies, or separate departments or divisions of companies, or any separate organization departments, groups, agencies, or offices, or the like, of any government or organization or any kind, where some sharing is required, but not any one party may be permitted to have access to all the data. Particularly apparent examples of the need and utility for such a method and

system of the present invention are to allow sharing, but maintain security, in between government areas, agencies and offices, and between different divisions, departments or offices of a large company, or any other organization, for example.

[0392] An example of the applicability of the methods of the present invention on a smaller scale is as follows. A Parser Master key is used as a serialization or branding of the secure data parser to an organization. As the scale of use of the Parser Master key is reduced from the whole enterprise to a smaller workgroup, the data securing methods described herein are used to share files within groups of users.

[0393] In the example shown in FIGURE 25 and described below, there are six users defined along with their title or role within the organization. The side bar represents five possible groups that the users can belong to according to their role. The arrow represents membership by the user in one or more of the groups.

[0394] When configuring the secure data parser for use in this example, the system administrator accesses the user and group information from the operating system by a maintenance program. This maintenance program generates and assigns Parser Group Master Keys to users based on their membership in groups.

[0395] In this example, there are three members in the Senior Staff group. For this group, the actions would be:

[0396] 1. Access Parser Group Master Key for the Senior Staff group (generate a key if not available);

[0397] 2. Generate a digital certificate associating CEO with the Senior Staff group;

[0398] 3. Generate a digital certificate associating CFO with the Senior Staff group;

[0399] 4. Generate a digital certificate associating Vice President, Marketing with the Senior Staff group.

[0400] The same set of actions would be done for each group, and each member within each group. When the maintenance program is complete, the Parser Group Master Key becomes a shared credential for each member of the group. Revocation of the assigned digital certificate may be done automatically when a user is removed from a group through the maintenance program without affecting the remaining members of the group.

[0401] Once the shared credentials have been defined, the parsing and splitting process remains the same. When a file, document or data element is to be secured, the user is prompted for the target group to be used when securing the data. The resulting secured data is only accessible by other members of the target group. This functionality of the methods and systems of the present invention may be used with any other computer system or software platform, any may be, for example, integrated into existing application programs or used standalone for file security.

[0402] It is readily apparent to those of ordinary skill in the art that any one or combination of encryption algorithms are suitable for use in the methods and systems of the present invention. For example, the encryption steps may, in one embodiment, be repeated to produce a multi-layered encryption scheme. In addition, a different encryption algorithm, or combination of encryption algorithms, may be used in repeat encryption steps such that different encryption

algorithms are applied to the different layers of the multi-layered encryption scheme. As such, the encryption scheme itself may become a component of the methods of the present invention for securing sensitive data from unauthorized use or access.

[0403] The secure data parser may include as an internal component, as an external component, or as both an error-checking component. For example, in one suitable approach, as portions of data are created using the secure data parser in accordance with the present invention, to assure the integrity of the data within a portion, a hash value is taken at preset intervals within the portion and is appended to the end of the interval. The hash value is a predictable and reproducible numeric representation of the data. If any bit within the data changes, the hash value would be different. A scanning module (either as a stand-alone component external to the secure data parser or as an internal component) may then scan the portions of data generated by the secure data parser. Each portion of data (or alternatively, less than all portions of data according to some interval or by a random or pseudo-random sampling) is compared to the appended hash value or values and an action may be taken. This action may include a report of values that match and do not match, an alert for values that do not match, or invoking of some external or internal program to trigger a recovery of the data. For example, recovery of the data could be performed by invoking a recovery module based on the concept that fewer than all portions may be needed to generate original data in accordance with the present invention.

[0404] Any other suitable integrity checking may be implemented using any suitable integrity information appended anywhere in all or a subset of data portions. Integrity information may include any suitable information that can be used to determine the integrity of data portions. Examples of integrity information may include hash values computed based on any suitable parameter (e.g., based on respective data portions), digital signature information, message authentication code (MAC) information, any other suitable information, or any combination thereof.

[0405] The secure data parser of the present invention may be used in any suitable application. Namely, the secure data parser described herein has a variety of applications in different areas of computing and technology. Several such areas are discussed below. It will be understood that these are merely illustrative in nature and that any other suitable applications may make use of the secure data parser. It will further be understood that the examples described are merely illustrative embodiments that may be modified in any suitable way in order to satisfy any suitable desires. For example, parsing and splitting may be based on any suitable units, such as by bits, by bytes, by kilobytes, by megabytes, by any combination thereof, or by any other suitable unit.

[0406] The secure data parser of the present invention may be used to implement secure physical tokens, whereby data stored in a physical token may be required in order to access additional data stored in another storage area. In one suitable approach, a physical token, such as a compact USB flash drive, a floppy disk, an optical disk, a smart card, or any

other suitable physical token, may be used to store one of at least two portions of parsed data in accordance with the present invention. In order to access the original data, the USB flash drive would need to be accessed. Thus, a personal computer holding one portion of parsed data would need to have the USB flash drive, having the other portion of parsed data, attached before the original data can be accessed. FIGURE 26 illustrates this application. Storage area 2500 includes a portion of parsed data 2502. Physical token 2504, having a portion of parsed data 2506 would need to be coupled to storage area 2500 using any suitable communications interface 2508 (e.g., USB, serial, parallel, Bluetooth, IR, IEEE 1394, Ethernet, or any other suitable communications interface) in order to access the original data. This is useful in a situation where, for example, sensitive data on a computer is left alone and subject to unauthorized access attempts. By removing the physical token (e.g., the USB flash drive), the sensitive data is inaccessible. It will be understood that any other suitable approach for using physical tokens may be used.

[0407] The secure data parser of the present invention may be used to implement a secure authentication system whereby user enrollment data (e.g., passwords, private encryption keys, fingerprint templates, biometric data or any other suitable user enrollment data) is parsed and split using the secure data parser. The user enrollment data may be parsed and split whereby one or more portions are stored on a smart card, a government Common Access Card, any suitable physical storage device (e.g., magnetic or

optical disk, USB key drive, etc.), or any other suitable device. One or more other portions of the parsed user enrollment data may be stored in the system performing the authentication. This provides an added level of security to the authentication process (e.g., in addition to the biometric authentication information obtained from the biometric source, the user enrollment data must also be obtained via the appropriate parsed and split data portion).

[0408] The secure data parser of the present invention may be integrated into any suitable existing system in order to provide the use of its functionality in each system's respective environment. FIGURE 27 shows a block diagram of an illustrative system 2600, which may include software, hardware, or both for implementing any suitable application. System 2600 may be an existing system in which secure data parser 2602 may be retrofitted as an integrated component. Alternatively, secure data parser 2602 may be integrated into any suitable system 2600 from, for example, its earliest design stage. Secure data parser 2600 may be integrated at any suitable level of system 2600. For example, secure data parser 2602 may be integrated into system 2600 at a sufficiently back-end level such that the presence of secure data parser 2602 may be substantially transparent to an end user of system 2600. Secure data parser 2602 may be used for parsing and splitting data among one or more storage devices 2604 in accordance with the present invention. Some illustrative examples of systems having the secure data parser integrated therein are discussed below.

[0409] The secure data parser of the present invention may be integrated into an operating system

kernel (e.g., Linux, Unix, or any other suitable commercial or proprietary operating system). This integration may be used to protect data at the device level whereby, for example, data that would ordinarily be stored in one or more devices is separated into a certain number of portions by the secure data parser integrated into the operating system and stored among the one or more devices. When original data is attempted to be accessed, the appropriate software, also integrated into the operating system, may recombine the parsed data portions into the original data in a way that may be transparent to the end user.

[0410] The secure data parser of the present invention may be integrated into a volume manager or any other suitable component of a storage system to protect local and networked data storage across any or all supported platforms. For example, with the secure data parser integrated, a storage system may make use of the redundancy offered by the secure data parser (i.e., which is used to implement the feature of needing fewer than all separated portions of data in order to reconstruct the original data) to protect against data loss. The secure data parser also allows all data written to storage devices, whether using redundancy or not, to be in the form of multiple portions that are generated according to the parsing of the present invention. When original data is attempted to be accessed, the appropriate software, also integrated into the volume manager or other suitable component of the storage system, may recombine the parsed data portions into the original data in a way that may be transparent to the end user.

[0411] In one suitable approach, the secure data parser of the present invention may be integrated into a RAID controller (as either hardware or software). This allows for the secure storage of data to multiple drives while maintaining fault tolerance in case of drive failure.

[0412] The secure data parser of the present invention may be integrated into a database in order to, for example, protect sensitive table information. For example, in one suitable approach, data associated with particular cells of a database table (e.g., individual cells, one or more particular columns, one or more particular rows, any combination thereof, or an entire database table) may be parsed and separated according to the present invention (e.g., where the different portions are stored on one or more storage devices at one or more locations or on a single storage device). Access to recombine the portions in order to view the original data may be granted by traditional authentication methods (e.g., username and password query).

[0413] The secure parser of the present invention may be integrated in any suitable system that involves data in motion (i.e., transfer of data from one location to another). Such systems include, for example, email, streaming data broadcasts, and wireless (e.g., WiFi) communications. With respect to email, in one suitable approach, the secure parser may be used to parse outgoing messages (i.e., containing text, binary data, or both (e.g., files attached to an email message)) and sending the different portions of the parsed data along different paths thus creating multiple streams of data. If any one of these streams

of data is compromised, the original message remains secure because the system may require that more than one of the portions be combined, in accordance with the present invention, in order to generate the original data. In another suitable approach, the different portions of data may be communicated along one path sequentially so that if one portion is obtained, it may not be sufficient to generate the original data. The different portions arrive at the intended recipient's location and may be combined to generate the original data in accordance with the present invention.

[0414] FIGURES 28 and 29 are illustrative block diagrams of such email systems. FIGURE 28 shows a sender system 2700, which may include any suitable hardware, such as a computer terminal, personal computer, handheld device (e.g., PDA, Blackberry), cellular telephone, computer network, any other suitable hardware, or any combination thereof. Sender system 2700 is used to generate and/or store a message 2704, which may be, for example, an email message, a binary data file (e.g., graphics, voice, video, etc.), or both. Message 2704 is parsed and split by secure data parser 2702 in accordance with the present invention. The resultant data portions may be communicated across one or more separate communications paths 2706 over network 2708 (e.g., the Internet, an intranet, a LAN, WiFi, Bluetooth, any other suitable hard-wired or wireless communications means, or any combination thereof) to recipient system 2710. The data portions may be communicated parallel in time or alternatively, according to any suitable time delay between the communication of the different data portions. Recipient system 2710 may be any suitable

hardware as described above with respect to sender system 2700. The separate data portions carried along communications paths 2706 are recombined at recipient system 2710 to generate the original message or data in accordance with the present invention.

[0415] FIGURE 29 shows a sender system 2800, which may include any suitable hardware, such as a computer terminal, personal computer, handheld device (e.g., PDA), cellular telephone, computer network, any other suitable hardware, or any combination thereof. Sender system 2800 is used to generate and/or store a message 2804, which may be, for example, an email message, a binary data file (e.g., graphics, voice, video, etc.), or both. Message 2804 is parsed and split by secure data parser 2802 in accordance with the present invention. The resultant data portions may be communicated across a single communications paths 2806 over network 2808 (e.g., the Internet, an intranet, a LAN, WiFi, Bluetooth, any other suitable communications means, or any combination thereof) to recipient system 2810. The data portions may be communicated serially across communications path 2806 with respect to one another. Recipient system 2810 may be any suitable hardware as described above with respect to sender system 2800. The separate data portions carried along communications path 2806 are recombined at recipient system 2810 to generate the original message or data in accordance with the present invention.

[0416] It will be understood that the arrangement of FIGURES 28 and 29 are merely illustrative. Any other suitable arrangement may be used. For example, in another suitable approach, the features of the systems of FIGURES 28 and 29 may be combined whereby the multi-

path approach of FIGURE 28 is used and in which one or more of communications paths 2706 are used to carry more than one portion of data as communications path 2806 does in the context of FIGURE 29.

[0417] The secure data parser may be integrated at any suitable level of a data-in motion system. For example, in the context of an email system, the secure parser may be integrated at the user-interface level (e.g., into Microsoft® Outlook), in which case the user may have control over the use of the secure data parser features when using email. Alternatively, the secure parser may be implemented in a back-end component such as at the exchange server, in which case messages may be automatically parsed, split, and communicated along different paths in accordance with the present invention without any user intervention.

[0418] Similarly, in the case of streaming broadcasts of data (e.g., audio, video), the outgoing data may be parsed and separated into multiple streams each containing a portion of the parsed data. The multiple streams may be transmitted along one or more paths and recombined at the recipient's location in accordance with the present invention. One of the benefits of this approach is that it avoids the relatively large overhead associated with traditional encryption of data followed by transmission of the encrypted data over a single communications channel. The secure parser of the present invention allows data in motion to be sent in multiple parallel streams, increasing speed and efficiency.

[0419] It will be understand that the secure data parser may be integrated for protection of and fault tolerance of any type of data in motion through any

transport medium, including, for example, wired, wireless, or physical. For example, voice over Internet protocol (VoIP) applications may make use of the secure data parser of the present invention. Wireless or wired data transport from or to any suitable personal digital assistant (PDA) devices such as Blackberries and SmartPhones may be secured using the secure data parser of the present invention. Communications using wireless 802.11 protocols for peer to peer and hub based wireless networks, satellite communications, point to point wireless communications, Internet client/server communications, or any other suitable communications may involve the data in motion capabilities of the secure data parser in accordance with the present invention. Data communication between computer peripheral device (e.g., printer, scanner, monitor, keyboard, network router, biometric authentication device (e.g., fingerprint scanner), or any other suitable peripheral device) between a computer and a computer peripheral device, between a computer peripheral device and any other suitable device, or any combination thereof may make use of the data in motion features of the present invention.

[0420] The data in motion features of the present invention may also apply to physical transportation of secure shares using for example, separate routes, vehicles, methods, any other suitable physical transportation, or any combination thereof. For example, physical transportation of data may take place on digital/magnetic tapes, floppy disks, optical disks, physical tokens, USB drives, removable hard drives, consumer electronic devices with flash memory (e.g., Apple IPODs or other MP3 players), flash memory, any

other suitable medium used for transporting data, or any combination thereof.

[0421] The secure data parser of the present invention may provide security with the ability for disaster recovery. According to the present invention, fewer than all portions of the separated data generated by the secure data parser may be necessary in order to retrieve the original data. That is, out of m portions stored, n may be the minimum number of these m portions necessary to retrieve the original data, where n <= m. For example, if each of four portions is stored in a different physical location relative to the other three portions, then, if n=2 in this example, two of the locations may be compromised whereby data is destroyed or inaccessible, and the original data may still be retrieved from the portions in the other two locations. Any suitable value for n or m may be used.

[0422] In addition, the n of m feature of the present invention may be used to create a "two man rule" whereby in order to avoid entrusting a single individual or any other entity with full access to what may be sensitive data, two or more distinct entities, each with a portion of the separated data parsed by the secure parser of the present invention may need to agree to put their portions together in order to retrieve the original data.

[0423] The secure data parser of the present invention may be used to provide a group of entities with a group-wide key that allows the group members to access particular information authorized to be accessed by that particular group. The group key may be one of the data portions generated by the secure parser in accordance with the present invention that may be

required to be combined with another portion centrally stored, for example in order to retrieve the information sought. This feature allows for, for example, secure collaboration among a group. It may be applied in for example, dedicated networks, virtual private networks, intranets, or any other suitable network.

[0424] Specific applications of this use of the secure parser include, for example, coalition information sharing in which, for example, multi-national friendly government forces are given the capability to communicate operational and otherwise sensitive data on a security level authorized to each respective country over a single network or a dual network (i.e., as compared to the many networks involving relatively substantial manual processes currently used). This capability is also applicable for companies or other organizations in which information needed to be known by one or more specific individuals (within the organization or without) may be communicated over a single network without the need to worry about unauthorized individuals viewing the information.

[0425] Another specific application includes a multi-level security hierarchy for government systems. That is, the secure parser of the present invention may provide for the ability to operate a government system at different levels of classified information (e.g., unclassified, classified, secret, top secret) using a single network. If desired, more networks may be used (e.g., a separate network for top secret), but the present invention allows for substantially fewer than

current arrangement in which a separate network is used for each level of classification.

[0426] It will be understood that any combination of the above described applications of the secure parser of the present invention may be used. For example, the group key application can be used together with the data in motion security application (i.e., whereby data that is communicated over a network can only be accessed by a member of the respective group and where, while the data is in motion, it is split among multiple paths (or sent in sequential portions) in accordance with the present invention).

[0427] The secure data parser of the present invention may be integrated into any middleware application to enable applications to securely store data to different database products or to different devices without modification to either the applications or the database. Middleware is a general term for any product that allows two separate and already existing programs to communicate. For example, in one suitable approach, middleware having the secure data parser integrated, may be used to allow programs written for a particular database to communicate with other databases without custom coding.

[0428] The secure data parser of the present invention may be implemented having any combination of any suitable capabilities, such as those discussed herein. In some embodiments of the present invention, for example, the secure data parser may be implemented having only certain capabilities whereas other capabilities may be obtained through the use of external software, hardware, or both interfaced either directly or indirectly with the secure data parser.

[0429] FIGURE 30, for example, shows an illustrative implementation of the secure data parser as secure data parser 3000. Secure data parser 3000 may be implemented with very few built-in capabilities. As illustrated, secure data parser 3000 may include built-in capabilities for parsing and splitting data into portions (also referred to herein as shares) of data using module 3002 in accordance with the present invention. Secure data parser 3000 may also include built in capabilities for performing redundancy in order to be able to implement, for example, the m of n feature described above (i.e., recreating the original data using fewer than all shares of parsed and split data) using module 3004. Secure data parser 3000 may also include share distribution capabilities using module 3006 for placing the shares of data into buffers from which they are sent for communication to a remote location, for storage, etc. in accordance with the present invention. It will be understood that any other suitable capabilities may be built into secure data parser 3000.

[0430] Assembled data buffer 3008 may be any suitable memory used to store the original data (although not necessarily in its original form) that will be parsed and split by secure data parser 3000. In a splitting operation, assembled data buffer 3008 provides input to secure data parser 3008. In a restore operation, assembled data buffer 3008 may be used to store the output of secure data parser 3000.

[0431] Split shares buffers 3010 may be one or more memory modules that may be used to store the multiple shares of data that resulted from the parsing and splitting of original data. In a splitting operation,

split shares buffers 3010 hold the output of the secure data parser. In a restore operation, split shares buffers hold the input to secure data parser 3000.

[0432] It will be understood that any other suitable arrangement of capabilities may be built-in for secure data parser 3000. Any additional features may be built-in and any of the features illustrated may be removed, made more robust, made less robust, or may otherwise be modified in any suitable way. Buffers 3008 and 3010 are likewise merely illustrative and may be modified, removed, or added to in any suitable way.

[0433] Any suitable modules implemented in software, hardware or both may be called by or may call to secure data parser 3000. If desired, even capabilities that are built into secure data parser 3000 may be replaced by one or more external modules. As illustrated, some external modules include random number generator 3012, cipher feedback key generator 3014, hash algorithm 3016, any one or more types of encryption 3018, and key management 3020. It will be understood that these are merely illustrative external modules. Any other suitable modules may be used in addition to or in place of those illustrated.

[0434] Cipher feedback key generator 3014 may, externally to secure data parser 3000, generate for each secure data parser operation, a unique key, or random number (using, for example, random number generator 3012), to be used as a seed value for an operation that extends an original session key size (e.g., a value of 128, 256, 512, or 1024 bits) into a value equal to the length of the data to be parsed and split. Any suitable algorithm may be used for the

cipher feedback key generation, including, for example, the AES cipher feedback key generation algorithm.

[0435] In order to facilitate integration of secure data parser 3000 and its external modules (i.e., secure data parser layer 3026) into an application layer 3024 (e.g., email application, database application, etc.), a wrapping layer that may make use of, for example, API function calls may be used. Any other suitable arrangement for facilitating integration of secure data parser layer 3026 into application layer 3024 may be used.

[0436] FIGURE 31 illustratively shows how the arrangement of FIGURE 30 may be used when a write (e.g., to a storage device), insert (e.g., in a database field), or transmit (e.g., across a network) command is issued in application layer 3024. At step 3100 data to be secured is identified and a call is made to the secure data parser. The call is passed through wrapper layer 3022 where at step 3102, wrapper layer 3022 streams the input data identified at step 3100 into assembled data buffer 3008. Also at step 3102, any suitable share information, filenames, any other suitable information, or any combination thereof may be stored (e.g., as information 3106 at wrapper layer 3022). Secure data processor 3000 then parses and splits the data it takes as input from assembled data buffer 3008 in accordance with the present invention. It outputs the data shares into split shares buffers 3010. At step 3104, wrapper layer 3022 obtains from stored information 3106 any suitable share information (i.e., stored by wrapper 3022 at step 3102) and share location(s) (e.g., from one or more configuration files). Wrapper layer 3022 then writes

the output shares (obtained from split shares buffers 3010) appropriately (e.g., written to one or more storage devices, communicated onto a network, etc.).

[0437] FIGURE 32 illustratively shows how the arrangement of FIGURE 30 may be used when a read (e.g., from a storage device), select (e.g., from a database field), or receive (e.g., from a network) occurs. At step 3200, data to be restored is identified and a call to secure data parser 3000 is made from application layer 3024. At step 3202, from wrapper layer 3022, any suitable share information is obtained and share location is determined. Wrapper layer 3022 loads the portions of data identified at step 3200 into split shares buffers 3010. Secure data parser 3000 then processes these shares in accordance with the present invention (e.g., if only three of four shares are available, then the redundancy capabilities of secure data parser 3000 may be used to restore the original data using only the three shares). The restored data is then stored in assembled data buffer 3008. At step 3204, application layer 3022 converts the data stored in assembled data buffer 3008 into its original data format (if necessary) and provides the original data in its original format to application layer 3024.

[0438] It will be understood that the parsing and splitting of original data illustrated in FIGURE 31 and the restoring of portions of data into original data illustrated in FIGURE 32 is merely illustrative. Any other suitable processes, components, or both may be used in addition to or in place of those illustrated.

[0439] FIGURE 33 is a block diagram of an illustrative process flow for parsing and splitting original data into two or more portions of data in

accordance with one embodiment of the present invention. As illustrated, the original data desired to be parsed and split is plain text 3306 (i.e., the word "SUMMIT" is used as an example). It will be understood that any other type of data may be parsed and split in accordance with the present invention. A session key 3300 is generated. If the length of session key 3300 is not compatible with the length of original data 3306, then cipher feedback session key 3304 may be generated.

[0440] In one suitable approach, original data 3306 may be encrypted prior to parsing, splitting, or both. For example, as FIGURE 33 illustrates, original data 3306 may be XORed with any suitable value (e.g., with cipher feedback session key 3304, or with any other suitable value). It will be understood that any other suitable encryption technique may be used in place of or in addition to the XOR technique illustrate. It will further be understood that although FIGURE 33 is illustrated in terms of byte by byte operations, the operation may take place at the bit level or at any other suitable level. It will further be understood that, if desired, there need not be any encryption whatsoever of original data 3306.

[0441] The resultant encrypted data (or original data if no encryption took place) is then hashed to determine how to split the encrypted (or original) data among the output buckets (e.g., of which there are four in the illustrated example). In the illustrated example, the hashing takes place by bytes and is a function of cipher feedback session key 3304. It will be understood that this is merely illustrative. The hashing may be performed at the bit level, if desired.

The hashing may be a function of any other suitable value besides cipher feedback session key 3304. In another suitable approach, hashing need not be used. Rather, any other suitable technique for splitting data may be employed.

[0442] FIGURE 34 is a block diagram of an illustrative process flow for restoring original data 3306 from two or more parsed and split portions of original data 3306 in accordance with one embodiment of the present invention. The process involves hashing the portions in reverse (i.e., to the process of FIGURE 33) as a function of cipher feedback session key 3304 to restore the encrypted original data (or original data if there was no encryption prior to the parsing and splitting). The encryption key may then be used to restore the original data (i.e., in the illustrated example, cipher feedback session key 3304 is used to decrypt the XOR encryption by XORing it with the encrypted data). This the restores original data 3306.

[0443] FIGURE 35 shows how bit-splitting may be implemented in the example of FIGURES 33 and 34. A hash may be used (e.g., as a function of the cipher feedback session key, as a function of any other suitable value) to determine a bit value at which to split each byte of data. It will be understood that this is merely one illustrative way in which to implement splitting at the bit level. Any other suitable technique may be used.

[0444] It will be understood that any reference to hash functionality made herein may be made with respect to any suitable hash algorithm. These include for example, MD5 and SHA-1. Different hash algorithms may

be used at different times and by different components of the present invention.

[0445] After a split point has been determined in accordance with the above illustrative procedure or through any other procedure or algorithm, a determination may be made with regard to which data portions to append each of the left and right segments. Any suitable algorithm may be used for making this determination. For example, in one suitable approach, a table of all possible distributions (e.g., in the form of pairings of destinations for the left segment and for the right segment) may be created, whereby a destination share value for each of the left and right segment may be determined by using any suitable hash function on corresponding data in the session key, cipher feedback session key, or any other suitable random or pseudo-random value, which may be generated and extended to the size of the original data. For example, a hash function of a corresponding byte in the random or pseudo-random value may be made. The output of the hash function is used to determine which pairing of destinations (i.e., one for the left segment and one for the right segment) to select from the table of all the destination combinations. Based on this result, each segment of the split data unit is appended to the respective two shares indicated by the table value selected as a result of the hash function.

[0446] Redundancy information may be appended to the data portions in accordance with the present invention to allow for the restoration of the original data using fewer than all the data portions. For example, if two out of four portions are desired to be sufficient for restoration of data, then additional data from the

shares may be accordingly appended to each share in, for example, a round-robin manner (e.g., where the size of the original data is 4MB, then share 1 gets its own shares as well as those of shares 2 and 3; share 2 gets its own share as well as those of shares 3 and 4; share 3 gets its own share as well as those of shares 4 and 1; and share 4 gets its own shares as well as those of shares 1 and 2). Any such suitable redundancy may be used in accordance with the present invention.

[0447] It will be understood that any other suitable parsing and splitting approach may be used to generate portions of data from an original data set in accordance with the present invention. For example, parsing and splitting may be randomly or pseudo-randomly processed on a bit by bit basis. A random or pseudo-random value may be used (e.g., session key, cipher feedback session key, etc.) whereby for each bit in the original data, the result of a hash function on corresponding data in the random or pseudo-random value may indicate to which share to append the respective bit. In one suitable approach the random or pseudo-random value may be generated as, or extended to, 8 times the size of the original data so that the hash function may be performed on a corresponding byte of the random or pseudo-random value with respect to each bit of the original data. Any other suitable algorithm for parsing and splitting data on a bit by bit level may be used in accordance with the present invention. It will further be appreciated that redundancy data may be appended to the data shares such as, for example, in the manner described immediately above in accordance with the present invention.

[0448] In one suitable approach, parsing and splitting need not be random or pseudo-random. Rather, any suitable deterministic algorithm for parsing and splitting data may be used. For example, breaking up the original data into sequential shares may be employed as a parsing and splitting algorithm. Another example is to parse and split the original data bit by bit, appending each respective bit to the data shares sequentially in a round-robin manner. It will further be appreciated that redundancy data may be appended to the data shares such as, for example, in the manner described above in accordance with the present invention.

[0449] In one embodiment of the present invention, after the secure data parser generates a number of portions of original data, in order to restore the original data, certain one or more of the generated portions may be mandatory. For example, if one of the portions is used as an authentication share (e.g., saved on a physical token device), and if the fault tolerance feature of the secure data parser is being used (i.e., where fewer than all portions are necessary to restore the original data), then even though the secure data parser may have access to a sufficient number of portions of the original data in order to restore the original data, it may require the authentication share stored on the physical token device before it restores the original data. It will be understood that any number and types of particular shares may be required based on, for example, application, type of data, user, any other suitable factors, or any combination thereof.

[0450] In one suitable approach, the secure data parser or some external component to the secure data parser may encrypt one or more portions of the original data. The encrypted portions may be required to be provided and decrypted in order to restore the original data. The different encrypted portions may be encrypted with different encryption keys. For example, this feature may be used to implement a more secure "two man rule" whereby a first user would need to have a particular share encrypted using a first encryption and a second user would need to have a particular share encrypted using a second encryption key. In order to access the original data, both users would need to have their respective encryption keys and provide their respective portions of the original data. In one suitable approach, a public key may be used to encrypt one or more data portions that may be a mandatory share required to restore the original data. A private key may then be used to decrypt the share in order to be used to restore to the original data.

[0451] Any such suitable paradigm may be used that makes use of mandatory shares where fewer than all shares are needed to restore original data.

[0452] In one suitable embodiment of the present invention, distribution of data into a finite number of shares of data may be processed randomly or pseudo-randomly such that from a statistical perspective, the probability that any particular share of data receives a particular unit of data is equal to the probability that any one of the remaining shares will receive the unit of data. As a result, each share of data will have an approximately equal amount of data bits.

[0453] According to another embodiment of the present invention, each of the finite number of shares of data need not have an equal probability of receiving units of data from the parsing and splitting of the original data. Rather certain one or more shares may have a higher or lower probability than the remaining shares. As a result, certain shares may be larger or smaller in terms of bit size relative to other shares. For example, in a two-share scenario, one share may have a 1% probability of receiving a unit of data whereas the second share has a 99% probability. It should follow, therefore that once the data units have been distributed by the secure data parser among the two share, the first share should have approximately 1% of the data and the second share 99%. Any suitable probabilities may be used in accordance with the present invention.

[0454] It will be understood that the secure data parser may be programmed to distribute data to shares according to an exact (or near exact) percentage as well. For example, the secure data parser may be programmed to distribute 80% of data to a first share and the remaining 20% of data to a second share.

[0455] According to another embodiment of the present invention, the secure data parser may generate data shares, one or more of which have predefined sizes. For example, the secure data parser may split original data into data portions where one of the portions is exactly 256 bits. In one suitable approach, if it is not possible to generate a data portion having the requisite size, then the secure data parser may pad the portion to make it the correct size. Any suitable size may be used.

[0456] In one suitable approach, the size of a data portion may be the size of an encryption key, a splitting key, any other suitable key, or any other suitable data element.

[0457] As previously discussed, the secure data parser may use keys in the parsing and splitting of data. For purposes of clarity and brevity, these keys shall be referred to herein as "splitting keys." For example, the Session Master Key, previously introduced, is one type of splitting key. Also, as previously discussed, splitting keys may be secured within shares of data generated by the secure data parser. Any suitable algorithms for securing splitting keys may be used to secure them among the shares of data. For example, the Shamir algorithm may be used to secure the splitting keys whereby information that may be used to reconstruct a splitting key is generated and appended to the shares of data. Any other such suitable algorithm may be used in accordance with the present invention.

[0458] Similarly, any suitable encryption keys may be secured within one or more shares of data according to any suitable algorithm such as the Shamir algorithm. For example, encryption keys used to encrypt a data set prior to parsing and splitting, encryption keys used to encrypt a data portions after parsing and splitting, or both may be secured using, for example, the Shamir algorithm or any other suitable algorithm.

[0459] According to one embodiment of the present invention, an All or Nothing Transform (AoNT), such as a Full Package Transform, may be used to further secure data by transforming splitting keys, encryption keys, any other suitable data elements, or any combination

thereof. For example, an encryption key used to encrypt a data set prior to parsing and splitting in accordance with the present invention may be transformed by an AoNT algorithm. The transformed encryption key may then be distributed among the data shares according to, for example, the Shamir algorithm or any other suitable algorithm. In order to reconstruct the encryption key, the encrypted data set must be restored (e.g., not necessarily using all the data shares if redundancy was used in accordance with the present invention) in order to access the necessary information regarding the transformation in accordance with AoNTs as is well known by one skilled in the art. When the original encryption key is retrieved, it may be used to decrypt the encrypted data set to retrieve the original data set. It will be understood that the fault tolerance features of the present invention may be used in conjunction with the AoNT feature. Namely, redundancy data may be included in the data portions such that fewer than all data portions are necessary to restore the encrypted data set.

[0460] It will be understood that the AoNT may be applied to encryption keys used to encrypt the data portions following parsing and splitting either in place of or in addition to the encryption and AoNT of the respective encryption key corresponding to the data set prior to parsing and splitting. Likewise, AoNT may be applied to splitting keys.

[0461] In one embodiment of the present invention, encryption keys, splitting keys, or both as used in accordance with the present invention may be further encrypted using, for example, a workgroup key in order

to provide an extra level of security to a secured data set.

[0462] In one embodiment of the present invention, an audit module may be provided that tracks whenever the secure data parser is invoked to split data.

[0463] FIGURE 36 illustrates possible options 3600 for using the components of the secure data parser in accordance with the invention. Each combination of options is outlined below and labeled with the appropriate step numbers from FIGURE 36. The secure data parser may be modular in nature, allowing for any known algorithm to be used within each of the function blocks shown in FIGURE 36. For example, other key splitting (e.g., secret sharing) algorithms such as Blakely may be used in place of Shamir, or the AES encryption could be replaced by other known encryption algorithms such as Triple DES. The labels shown in the example of FIGURE 36 merely depict one possible combination of algorithms for use in one embodiment of the invention. It should be understood that any suitable algorithm or combination of algorithms may be used in place of the labeled algorithms.

[0464] 1) 3610, 3612, 3614, 3615, 3616, 3617, 3618, 3619

[0465] Using previously encrypted data at step 3610, the data may be eventually split into a predefined number of shares. If the split algorithm requires a key, a split encryption key may be generated at step 3612 using a cryptographically secure pseudo-random number generator. The split encryption key may optionally be transformed using an All or Nothing Transform (AoNT) into a transform split key at step 3614 before being key split to the predefined number of

shares with fault tolerance at step 3615. The data may then be split into the predefined number of shares at step 3616. A fault tolerant scheme may be used at step 3617 to allow for regeneration of the data from less than the total number of shares. Once the shares are created, authentication/integrity information may be embedded into the shares at step 3618. Each share may be optionally post-encrypted at step 3619.

[0466] 2) 3111, 3612, 3614, 3615, 3616, 3617, 3618, 3619

[0467] In some embodiments, the input data may be encrypted using an encryption key provided by a user or an external system. The external key is provided at step 3611. For example, the key may be provided from an external key store. If the split algorithm requires a key, the split encryption key may be generated using a cryptographically secure pseudo-random number generator at step 3612. The split key may optionally be transformed using an All or Nothing Transform (AoNT) into a transform split encryption key at step 3614 before being key split to the predefined number of shares with fault tolerance at step 3615. The data is then split to a predefined number of shares at step 3616. A fault tolerant scheme may be used at step 3617 to allow for regeneration of the data from less than the total number of shares. Once the shares are created, authentication/integrity information may be embedded into the shares at step 3618. Each share may be optionally post-encrypted at step 3619.

[0468] 3) 3612, 3613, 3614, 3615, 3612, 3614, 3615, 3616, 3617, 3618, 3619

[0469] In some embodiments, an encryption key may be generated using a cryptographically secure pseudo-

random number generator at step 3612 to transform the data. Encryption of the data using the generated encryption key may occur at step 3613. The encryption key may optionally be transformed using an All or Nothing Transform (AoNT) into a transform encryption key at step 3614. The transform encryption key and/or generated encryption key may then be split into the predefined number of shares with fault tolerance at step 3615. If the split algorithm requires a key, generation of the split encryption key using a cryptographically secure pseudo-random number generator may occur at step 3612. The split key may optionally be transformed using an All or Nothing Transform (AoNT) into a transform split encryption key at step 3614 before being key split to the predefined number of shares with fault tolerance at step 3615. The data may then be split into a predefined number of shares at step 3616. A fault tolerant scheme may be used at step 3617 to allow for regeneration of the data from less than the total number of shares. Once the shares are created, authentication/integrity information will be embedded into the shares at step 3618. Each share may then be optionally post-encrypted at step 3619.

[0470] 4) 3612, 3614, 3615, 3616, 3617, 3618, 3619

[0471] In some embodiments, the data may be split into a predefined number of shares. If the split algorithm requires a key, generation of the split encryption key using a cryptographically secure pseudo-random number generator may occur at step 3612. The split key may optionally be transformed using an All or Nothing Transform (AoNT) into a transformed split key at step 3614 before being key split into the predefined number of shares with fault tolerance at step 3615.

The data may then be split at step 3616. A fault tolerant scheme may be used at step 3617 to allow for regeneration of the data from less than the total number of shares. Once the shares are created, authentication/integrity information may be embedded into the shares at step 3618. Each share may be optionally post-encrypted at step 3619.

[0472] Although the above four combinations of options are preferably used in some embodiments of the invention, any other suitable combinations of features, steps, or options may be used with the secure data parser in other embodiments.

[0473] The secure data parser may offer flexible data protection by facilitating physical separation. Data may be first encrypted, then split into shares with "m of n" fault tolerance. This allows for regeneration of the original information when less than the total number of shares is available. For example, some shares may be lost or corrupted in transmission. The lost or corrupted shares may be recreated from fault tolerance or integrity information appended to the shares, as discussed in more detail below.

[0474] In order to create the shares, a number of keys are optionally utilized by the secure data parser. These keys may include one or more of the following:

[0475] Pre-encryption key: When pre-encryption of the shares is selected, an external key may be passed to the secure data parser. This key may be generated and stored externally in a key store (or other location) and may be used to optionally encrypt data prior to data splitting.

[0476] Split encryption key: This key may be generated internally and used by the secure data parser

to encrypt the data prior to splitting. This key may then be stored securely within the shares using a key split algorithm.

[0477] Split session key: This key is not used with an encryption algorithm; rather, it may be used to key the data partitioning algorithms when random splitting is selected. When a random split is used, a split session key may be generated internally and used by the secure data parser to partition the data into shares. This key may be stored securely within the shares using a key splitting algorithm.

[0478] Post encryption key: When post encryption of the shares is selected, an external key may be passed to the secure data parser and used to post encrypt the individual shares. This key may be generated and stored externally in a key store or other suitable location.

[0479] In some embodiments, when data is secured using the secure data parser in this way, the information may only be reassembled provided that all of the required shares and external encryption keys are present.

[0480] FIGURE 37 shows illustrative overview process 3700 for using the secure data parser of the present invention in some embodiments. As described above, two well-suited functions for secure data parser 3706 may include encryption 3702 and backup 3704. As such, secure data parser 3706 may be integrated with a RAID or backup system or a hardware or software encryption engine in some embodiments.

[0481] The primary key processes associated with secure data parser 3706 may include one or more of pre-encryption process 3708, encrypt/transform process

3710, key secure process 3712, parse/distribute process 3714, fault tolerance process 3716, share authentication process 3716, and post-encryption process 3720. These processes may be executed in several suitable orders or combinations, as detailed in FIGURE 36. The combination and order of processes used may depend on the particular application or use, the level of security desired, whether optional pre-encryption, post-encryption, or both, are desired, the redundancy desired, the capabilities or performance of an underlying or integrated system, or any other suitable factor or combination of factors.

[0482] The output of illustrative process 3700 may be two or more shares 3722. As described above, data may be distributed to each of these shares randomly (or pseudo-randomly) in some embodiments. In other embodiments, a deterministic algorithm (or some suitable combination of random, pseudo-random, and deterministic algorithms) may be used.

[0483] In addition to the individual protection of information assets, there is sometimes a requirement to share information among different groups of users or communities of interest. It may then be necessary to either control access to the individual shares within that group of users or to share credentials among those users that would only allow members of the group to reassemble the shares. To this end, a workgroup key may be deployed to group members in some embodiments of the invention. The workgroup key should be protected and kept confidential, as compromise of the workgroup key may potentially allow those outside the group to access information. Some systems and methods for workgroup key deployment and protection are discussed

below.

[0484] The workgroup key concept allows for enhanced protection of information assets by encrypting key information stored within the shares. Once this operation is performed, even if all required shares and external keys are discovered, an attacker has no hope of recreating the information without access to the workgroup key.

[0485] FIGURE 38 shows illustrative block diagram 3800 for storing key and data components within the shares. In the example of diagram 3800, the optional pre-encrypt and post-encrypt steps are omitted, although these steps may be included in other embodiments.

[0486] The simplified process to split the data includes encrypting the data using encryption key 3804 at encryption stage 3802. Portions of encryption key 3804 may then be split and stored within shares 3810 in accordance with the present invention. Portions of split encryption key 3806 may also be stored within shares 3810. Using the split encryption key, data 3808 is then split and stored in shares 3810.

[0487] In order to restore the data, split encryption key 3806 may be retrieved and restored in accordance with the present invention. The split operation may then be reversed to restore the ciphertext. Encryption key 3804 may also be retrieved and restored, and the ciphertext may then be decrypted using the encryption key.

[0488] When a workgroup key is utilized, the above process may be changed slightly to protect the encryption key with the workgroup key. The encryption key may then be encrypted with the workgroup key prior

to being stored within the shares. The modified steps are shown in illustrative block diagram 3900 of FIGURE 39.

[0489] The simplified process to split the data using a workgroup key includes first encrypting the data using the encryption key at stage 3902. The encryption key may then be encrypted with the workgroup key at stage 3904. The encryption key encrypted with the workgroup key may then be split into portions and stored with shares 3912. Split key 3908 may also be split and stored in shares 3912. Finally, portions of data 3910 are split and stored in shares 3912 using split key 3908.

[0490] In order to restore the data, the split key may be retrieved and restored in accordance with the present invention. The split operation may then be reversed to restore the ciphertext in accordance with the present invention. The encryption key (which was encrypted with the workgroup key) may be retrieved and restored. The encryption key may then be decrypted using the workgroup key. Finally, the ciphertext may be decrypted using the encryption key.

[0491] There are several secure methods for deploying and protecting workgroup keys. The selection of which method to use for a particular application depends on a number of factors. These factors may include security level required, cost, convenience, and the number of users in the workgroup. Some commonly used techniques used in some embodiments are provided below:

[0492] Hardware-based Key Storage

Hardware-based solutions generally provide the strongest guarantees for the security of encryption/decryption keys in an encryption system.

Examples of hardware-based storage solutions include tamper-resistant key token devices which store keys in a portable device (e.g., smartcard/dongle), or non-portable key storage peripherals. These devices are designed to prevent easy duplication of key material by unauthorized parties. Keys may be generated by a trusted authority and distributed to users, or generated within the hardware. Additionally, many key storage systems provide for multi-factor authentication, where use of the keys requires access both a physical object (token) and a passphrase or biometric.

[0493] Software-based Key Storage

While dedicated hardware-based storage may be desirable for high-security deployments or applications, other deployments may elect to store keys directly on local hardware (e.g., disks, RAM or non-volatile RAM stores such as USB drives). This provides a lower level of protection against insider attacks, or in instances where an attacker is able to directly access the encryption machine.

[0494] To secure keys on disk, software-based key management often protects keys by storing them in encrypted form under a key derived from a combination of other authentication metrics, including: passwords and passphrases, presence of other keys (e.g., from a hardware-based solution), biometrics, or any suitable combination of the foregoing. The level of security provided by such techniques may range from the relatively weak key protection mechanisms provided by some operating systems (e.g., MS Windows and Linux), to more robust solutions implemented using multi-factor authentication.

[0495] The secure data parser of the present invention may be advantageously used in a number of applications and technologies. For example, email system, RAID systems, video broadcasting systems, database systems, or any other suitable system may have the secure data parser integrated at any suitable level. As previously discussed, it will be understand that the secure data parser may also be integrated for protection and fault tolerance of any type of data in motion through any transport medium, including, for example, wired, wireless, or physical transport mediums. As one example, voice over Internet protocol (VoIP) applications may make use of the secure data parser of the present invention to solve problems relating to echoes and delays that are commonly found in VoIP. The need for network retry on dropped packets may be eliminated by using fault tolerance, which guarantees packet delivery even with the loss of a predetermined number of shares. Packets of data (e.g., network packets) may also be efficiently split and restored "on-the-fly" with minimal delay and buffering, resulting in a comprehensive solution for various types of data in motion. The secure data parser may act on network data packets, network voice packets, file system data blocks, or any other suitable unit of information. In addition to being integrated with a VoIP application, the secure data parser may be integrated with a file-sharing application (e.g., a peer-to-peer file-sharing application), a video broadcasting application, an electronic voting or polling application (which may implement an electronic voting protocol and blind signatures, such as the Sensus protocol), an email application, or any other

network application that may require or desire secure communication.

[0496] In some embodiments, support for network data in motion may be provided by the secure data parser of the present invention in two distinct phases -- a header generation phase and a data partitioning phase. Simplified header generation process 4000 and simplified data partitioning process 4010 are shown in FIGURES 40A and 40B, respectively. One or both of these processes may be performed on network packets, file system blocks, or any other suitable information.

[0497] In some embodiments, header generation process 4000 may be performed one time at the initiation of a network packet stream. At step 4002, a random (or pseudo-random) split encryption key, K, may be generated. The split encryption key, K, may then be optionally encrypted (e.g., using the workgroup key described above) at AES key wrap step 4004. Although an AES key wrap may be used in some embodiments, any suitable key encryption or key wrap algorithm may be used in other embodiments. AES key wrap step 4004 may operate on the entire split encryption key, K, or the split encryption key may be parsed into several blocks (e.g., 64-bit blocks). AES key wrap step 4004 may then operate on blocks of the split encryption key, if desired.

[0498] At step 4006, a secret sharing algorithm (e.g., Shamir) may be used to split the split encryption key, K, into key shares. Each key share may then be embedded into one of the output shares (e.g., in the share headers). Finally, a share integrity block and (optionally) a post-authentication tag (e.g., MAC) may be appended to the header block of each share.

Each header block may be designed to fit within a single data packet.

[0499] After header generation is complete (e.g., using simplified header generation process 4000), the secure data parser may enter the data partitioning phase using simplified data splitting process 4010. Each incoming data packet or data block in the stream is encrypted using the split encryption key, K, at step 4012. At step 4014, share integrity information (e.g., a hash H) may be computed on the resulting ciphertext from step 4012. For example, a SHA-256 hash may be computed. At step 4106, the data packet or data block may then be partitioned into two or more data shares using one of the data splitting algorithms described above in accordance with the present invention. In some embodiments, the data packet or data block may be split so that each data share contains a substantially random distribution of the encrypted data packet or data block. The integrity information (e.g., hash H) may then be appended to each data share. An optional post-authentication tag (e.g., MAC) may also be computed and appended to each data share in some embodiments.

[0500] Each data share may include metadata, which may be necessary to permit correct reconstruction of the data blocks or data packets. This information may be included in the share header. The metadata may include such information as cryptographic key shares, key identities, share nonces, signatures/MAC values, and integrity blocks. In order to maximize bandwidth efficiency, the metadata may be stored in a compact binary format.

[0501] For example, in some embodiments, the share header includes a cleartext header chunk, which is not encrypted and may include such elements as the Shamir key share, per-session nonce, per-share nonce, key identifiers (e.g., a workgroup key identifier and a post-authentication key identifier). The share header may also include an encrypted header chunk, which is encrypted with the split encryption key. An integrity header chunk, which may include integrity checks for any number of the previous blocks (e.g., the previous two blocks) may also be included in the header. Any other suitable values or information may also be included in the share header.

[0502] As shown in illustrative share format 4100 of FIGURE 41, header block 4102 may be associated with two or more output blocks 4104. Each header block, such as header block 4102, may be designed to fit within a single network data packet. In some embodiments, after header block 4102 is transmitted from a first location to a second location, the output blocks may then be transmitted. Alternatively, header block 4102 and output blocks 4104 may be transmitted at the same time in parallel. The transmission may occur over one or more similar or dissimilar communications paths.

[0503] Each output block may include data portion 4106 and integrity/authenticity portion 4108. As described above, each data share may be secured using a share integrity portion including share integrity information (e.g., a SHA-256 hash) of the encrypted, pre-partitioned data. To verify the integrity of the outputs blocks at recovery time, the secure data parser may compare the share integrity blocks of each share and then invert the split algorithm. The hash of the

recovered data may then be verified against the share hash.

[0504] Although some common applications of the secure data parser are described above, it should be clearly understood that the present invention may be integrated with any network application in order to increase security, fault-tolerance, anonymity, or any suitable combination of the foregoing.

[0505] Another aspect of the present invention is a common interface (for example, a common application programming interface ("API")) for use in securely creating, storing, and managing cryptographic keys across a plurality of platforms, interfaces, or both platforms and interfaces. The common interface may be provided by hardware, software, firmware, or any combination of hardware, software, and firmware. For example, in some embodiments, a key provider application, utility, or mechanism may provide the common interface between those systems that securely store or manage cryptographic keys and the secure data parser. The key provider application, utility, or mechanism may also permit interoperability between various key providers and key stores.

[0506] Several designs goals may be achieved in connection with the common interface, including provable security and reliance on standard cryptography. For example, cryptographic keys may be handled and transferred securely using proven industry standard mechanisms for key management. The integrity of any key provider application or mechanism may also be maintained by not permitting the common interface to interfere with the features or functionality of the underlying key provider application, utility, or

mechanism. In some embodiments, a single interface may be used for all key providers. Changes to the underlying key provider programs or the secure data parser may not impact the common interface in some embodiments. Interface changes in the key provider or secure data parser engine may have only a minimal impact on the common interface. For example, the design of the common interface (or the application, utility, or mechanism that provides the common interface) may be highly modular. Both the composition of the individual provider binaries and the mechanisms that the common interface uses to interface to these binaries may be highly modular in nature.

[0507] In some embodiments, the common interface includes functions that generate, store, retrieve, and pass cryptographic keys between applications or other interfaces or platforms. These functions may include one or more of the following illustrative functions:

[0508] OPEN - The OPEN function may be used to initiate communication with a target application or interface through the application interface binary associated with the application or interface.

[0509] CLOSE - The CLOSE function may be used to halt communication with the target application or interface.

[0510] GENERATE KEY - The GENERATE KEY function may be used to create a cryptographic key.

[0511] RETRIEVE KEY - The RETRIEVE KEY function may be used to access a key to be used in a split or restore function of a secure data parser engine.

[0512] STORE KEY - The STORE KEY function may be used to place a key within the secure data parser engine's key store (or on other target media, for

example, removable media, such as a USB flash drive or smart card).

[0513] DELETE KEY - The DELETE KEY function may be used to remove a key from the secure data parser engine's key store.

[0514] In some embodiments, each application or interface may be configured to be used by the key provider application, utility, or mechanism using an individual application interface binary. This binary (e.g., program) may translate requests from the common interface to the individual requirements of the requesting application or interface. As such, the key provider application, utility, or mechanism may seamlessly (and securely) support cryptographic key creation, deletion, storage, and management across a wide variety of platforms and interfaces.

[0515] In order for a cryptographic key to be accessible to the secure data parser engine, the key may first be placed in the secure parser engine's key store and identified for use. Each application or interface may have individualized requirements specified in the application or interface's application interface binary. As each application is added to the key provider application, utility, or mechanism, it may be supplied with a translation module that details the translation between the common interface and the unique application interface requirements.

[0516] FIGURE 42 shows illustrative process 4200 for managing a cryptographic key using a common interface. At step 4202, a request may be received from a first mechanism or interface to manage a cryptographic key. For example, a request to generate, retrieve, store, or delete a cryptographic key may be received in a first

interface format. The request may be received, for example, over a secure communications channel or via a secured communications session. If the request is received over a network, such as the Internet, an network security protocol (such as SSL, TLS, SSH, or IPsec) may be implemented.

[0517] At step 4204, the received request may be translated to a common interface request. For example, the request may be translated from a first interface format to one or more of the common interface functions described above. At step 4206, the common interface request may be authenticated. For example, the user or originating network address may be cryptographically authenticated using any authentication protocol or cryptographic handshake. The request may also be authenticated by verifying that the request is in a valid common interface format or originated from an authorized source. For example, valid requests may be cryptographically signed in some embodiments, in which case step 4206 may include verifying a cryptographic signature associated with the request. This may help prevent third-party attacks where an invalidly formed common interface request may attempt to bypass authentication. The secure data parser (or the system or mechanism executing the secure data parser) may also maintain a table of authorized clients or requesting interfaces. Clients or requesting interfaces may be authenticated using any suitable authentication protocol.

[0518] At step 4208, a determination may be made whether the common interface request has been authenticated. For example, a authentication token may be provided after a successful authentication in some

embodiments. On subsequent requests from the same interface or requesting client, the authentication token may be presented in lieu of initiating a new authentication session. The authentication token may also be associated with some expiration date, expiration time, or both. As such, the authentication token may expire after some predetermined length of time after issuance. The key provider application, utility, or mechanism may enforce all expiration dates and times associated with authentication tokens by accessing the current time from a system clock or timer. In other embodiments, each request may be authenticated regardless of the presentation of a valid and unexpired authentication token. If the request has not been authenticated, illustrative process 4200 may return to step 4202.

[0519] If, at step 4208, a determination is made that the request has been authenticated, then at step 4210 the secure data parser's cryptographic keys may be accessed. As described above, cryptographic keys may be stored or secured in a variety of ways. For example, all or some of the secure data parser's keys may be secured in a standard key store, a separate hardware key store, a smart card, a custom key store, a separate database table, or secured within one or more encrypted shares.

[0520] After accessing the secure data parser's cryptographic keys, at step 4212 the common interface request may be executed. For example, one or more of the common interface functions described above may be executed. These functions may, *inter alia*, generate a new key, retrieve an existing key, store a key, or delete an existing key from the set of the secure data

parser's cryptographic keys. Any other suitable command or action may also be executed at step 4212. At step 4214, a determination is made whether the common interface request includes any return arguments. For example, some requests may return a cryptographic key, a memory address or pointer to a cryptographic key, or any other suitable information (e.g., a success or failure return code). If no arguments are to be returned at step 4214, illustrative process 4200 may return to step 4202 and await a new request.

[0521] If one or more arguments are to be returned, at step 4216 the return arguments may be translated to a format compatible with the first interface. For example, an application interface binary may convert the return argument from the common interface format to the interface format of the requesting client. At step 4218, the return arguments may be transmitted to the first interface over a secure communications path or communications session. If the request was received over a network, such as the Internet, the return argument or arguments may be transmitted over a network connection implementing a network security protocol (such as SSL, TLS, SSH, or IPsec).

[0522] Additionally, other combinations, additions, substitutions and modifications will be apparent to the skilled artisan in view of the disclosure herein. Accordingly, the present invention is not intended to be limited by the reaction of the preferred embodiments but is to be defined by a reference to the appended claims.

What is Claimed is:

1. A method for managing cryptographic keys, the method comprising:

receiving, from a first interface, a request to manage at least one cryptographic key stored remote from the first interface;

translating the request from a first interface format to a common interface format;

authenticating the request by at least verifying that the request originated from an authorized source; and

executing the translated request in the common interface format.

2. The method of claim 1 wherein executing the translated request comprises retrieving the at least one cryptographic key.

3. The method of claim 1 wherein executing the translated request comprises generating the at least one cryptographic key.

4. The method of claim 1 wherein executing the translated request comprises deleting the at least one cryptographic key.

5. The method of claim 1 wherein executing the translated request comprises storing the at least one cryptographic key in a key store.

6. The method of claim 1 wherein executing the translated request comprises storing the at least one cryptographic key on removable media.

7. The method of claim 1 further comprising securing a data set using the at least one cryptographic key, wherein securing the data set comprises:

encrypting the data set using the at least one cryptographic key;

generating a random or pseudo-random value;

distributing, based, at least in part, on the random or pseudorandom value, encrypted data in the data set into two or more shares; and

storing the two or more shares separately on at least one data depository.

8. The method of claim 7 wherein storing the two or more shares separately on at least one data depository comprises storing the two or more shares on at least two geographically separated data depositories.

9. The method of claim 1 further comprising translating at least one return argument of the executed request from a common interface format to a first interface format.

10. The method of claim 9 wherein the at least one return argument of the executed request comprises at least one cryptographic key.

11. The method of claim 9 further comprising transmitting the at least one return argument to the first interface over a secure communications path.

12. The method of claim 1 wherein authenticating the request comprises implementing an authentication protocol or cryptographic handshake.

13. The method of claim 1 wherein authenticating the request comprises verifying a cryptographic signature associated with the request.

14. The method of claim 1 wherein authenticating the request comprises validating an authentication token.

15. The method of claim 14 wherein validating an authentication token comprises enforcing an expiration date or expiration time associated with the authentication token.

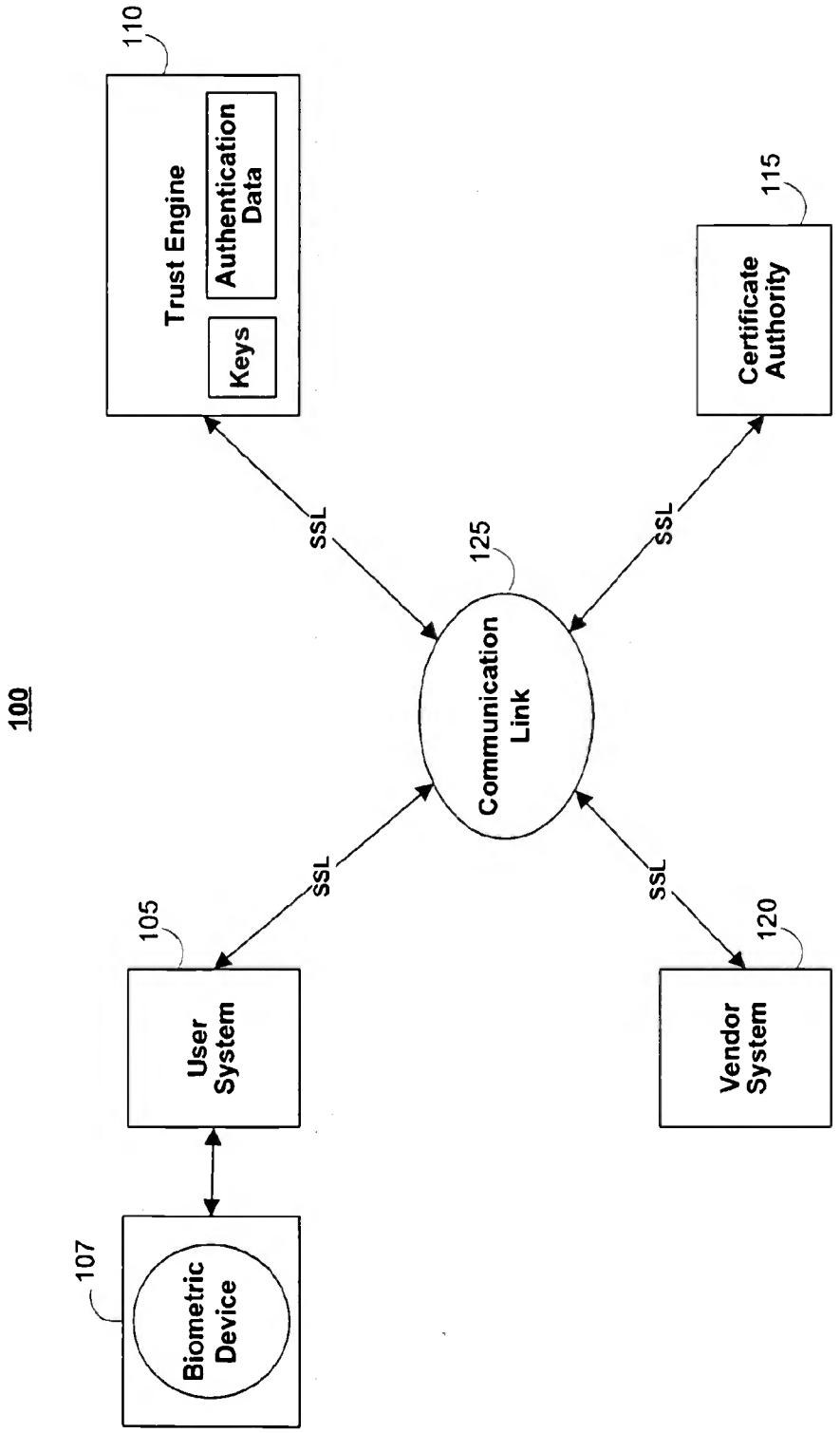

100
110
Trust Engine
Authentication Data
Keys
115
Certificate Authority
SSL
SSL
125
Communication Link
SSL
SSL
105
User System
120
Vendor System
107
Biometric Device

FIG. 1

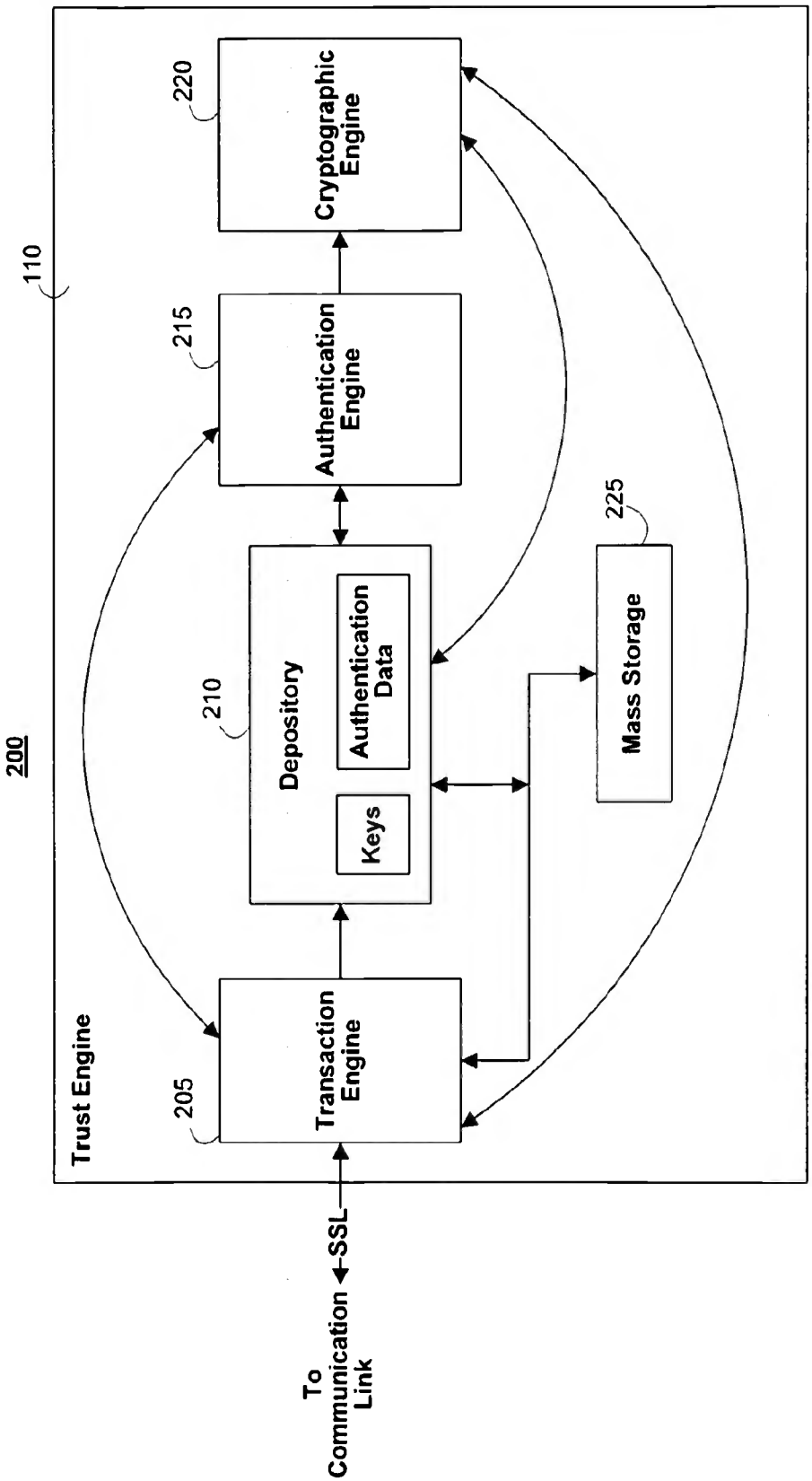

200
110
Trust Engine
220
Cryptographic Engine
215
Authentication Engine
210
Depository
Authentication Data
Keys
225
Mass Storage
205
Transaction Engine
SSL
To Communication Link

FIG. 2

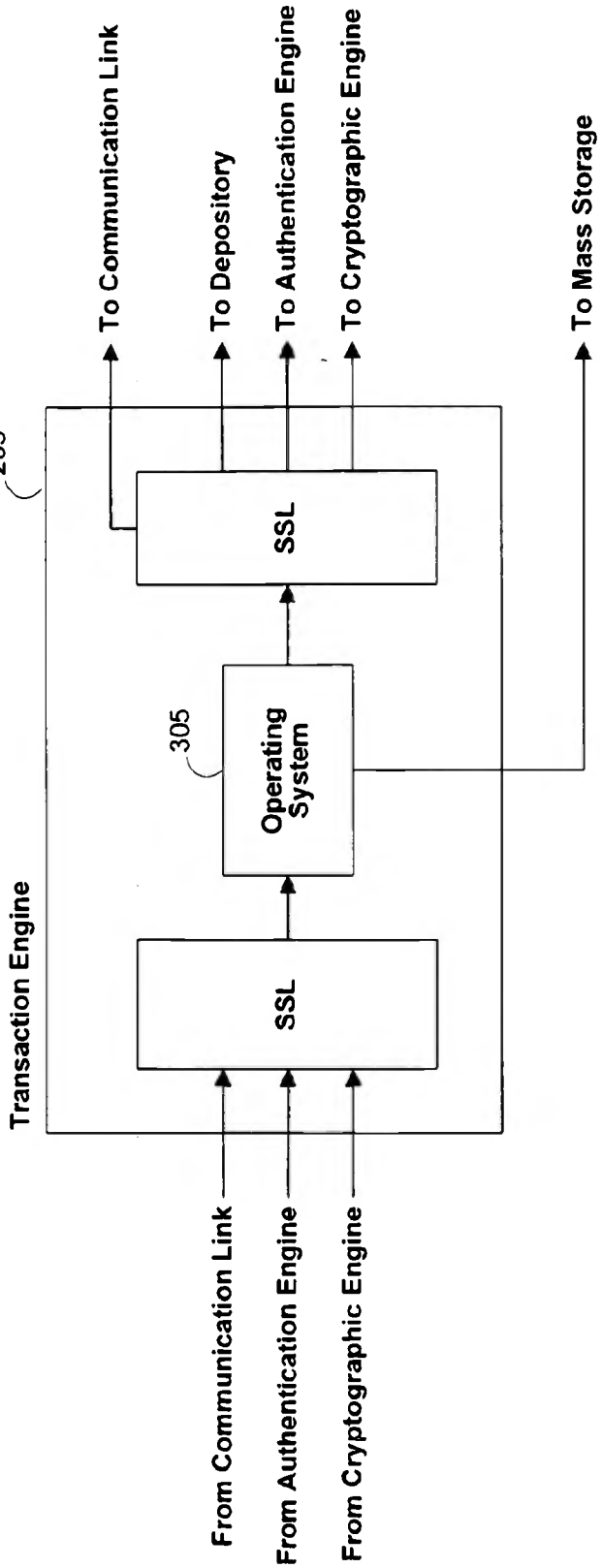

To Communication Link
To Depository
To Authentication Engine
To Cryptographic Engine
To Mass Storage
205
SSL
305
Operating System
SSL
Transaction Engine
From Communication Link
From Authentication Engine
From Cryptographic Engine

FIG. 3

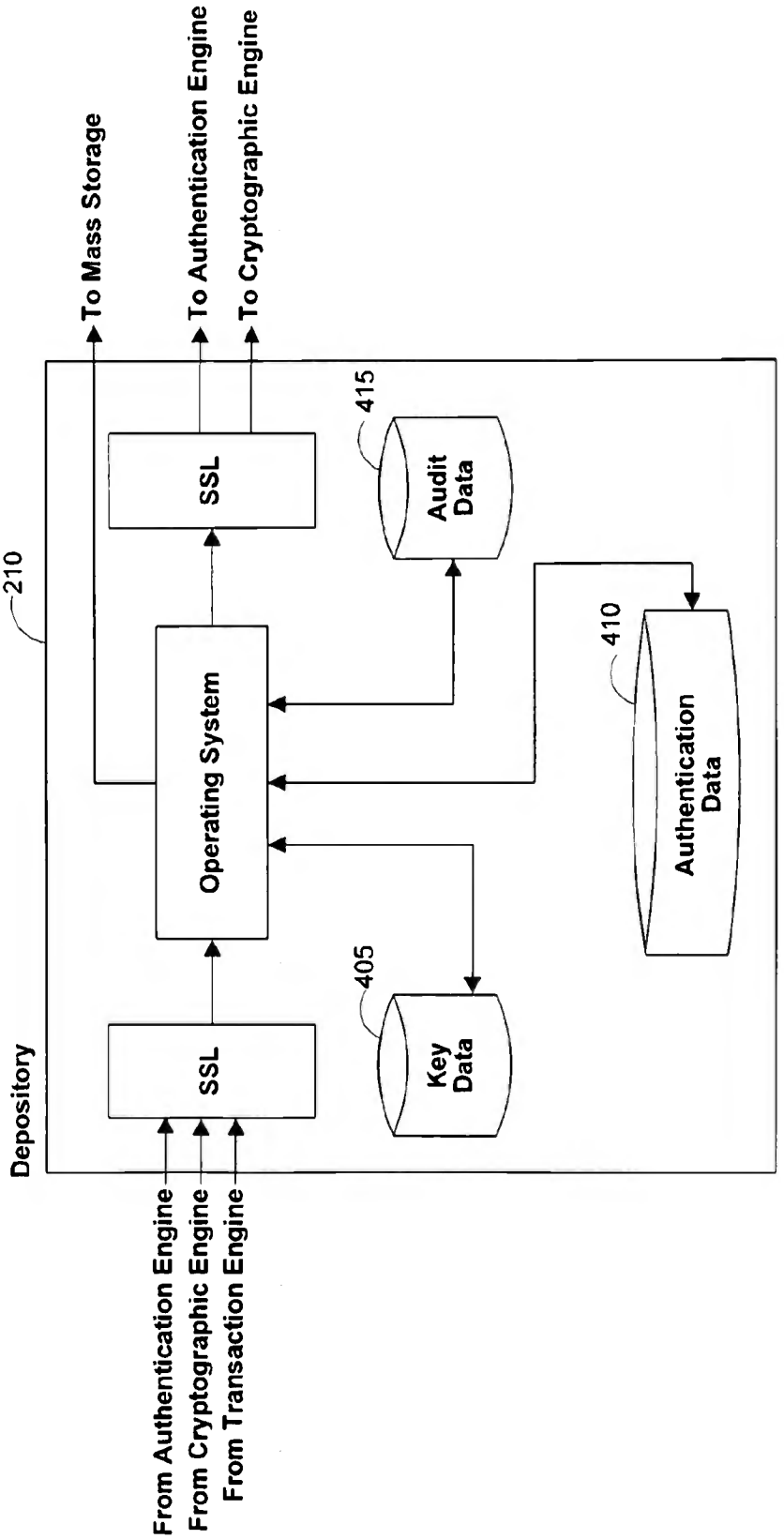

To Mass Storage
To Authentication Engine
To Cryptographic Engine
210
SSL
415
Audit Data
Operating System
410
Authentication Data
405
Key Data
SSL
Depository
From Authentication Engine
From Cryptographic Engine
From Transaction Engine

FIG. 4

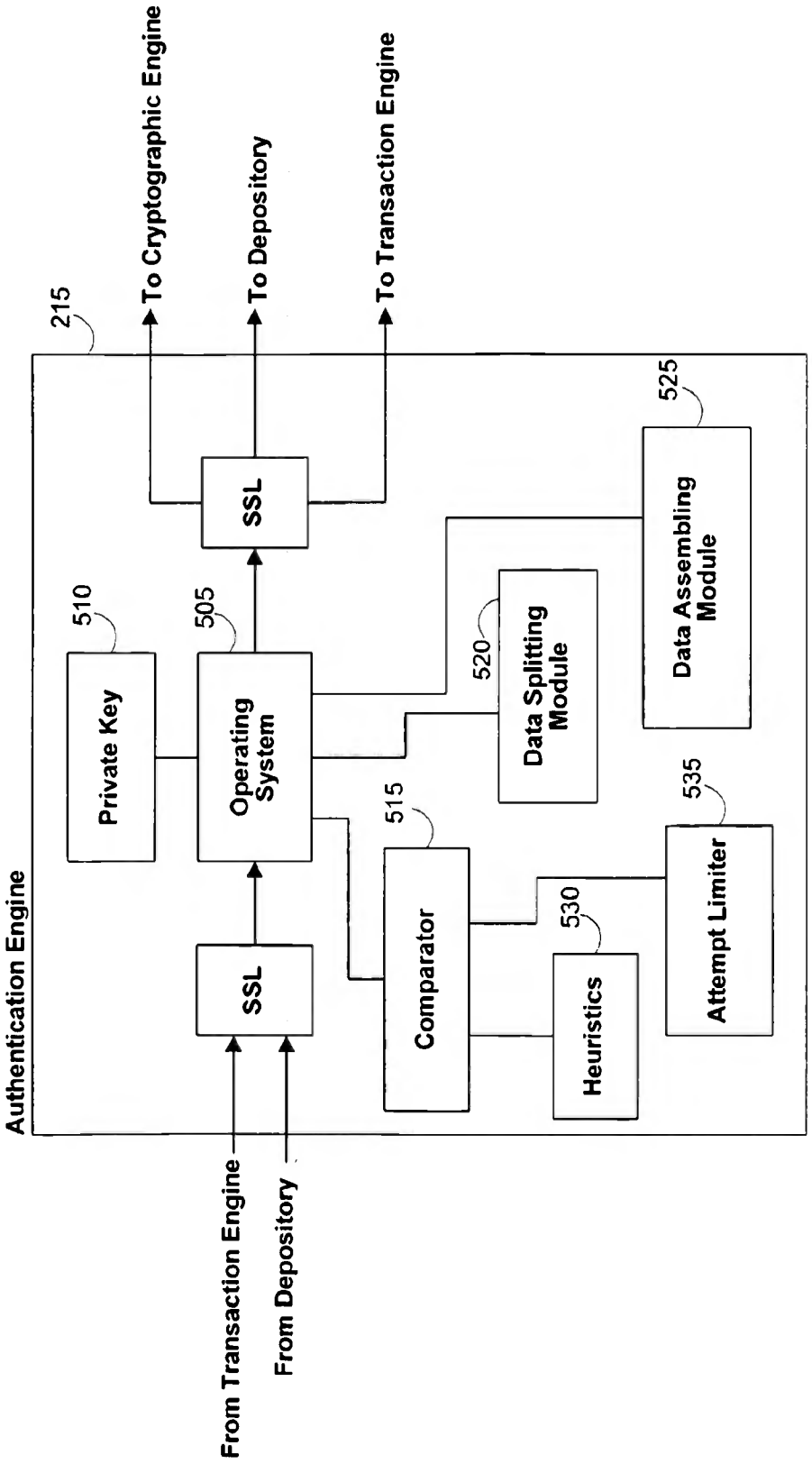

To Cryptographic Engine
To Depository
To Transaction Engine
215
Authentication Engine
SSL
Private Key
510
Operating System
505
Data Splitting Module
520
Data Assembling Module
525
SSL
Comparator
515
Heuristics
530
Attempt Limiter
535
From Transaction Engine
From Depository

FIG. 5

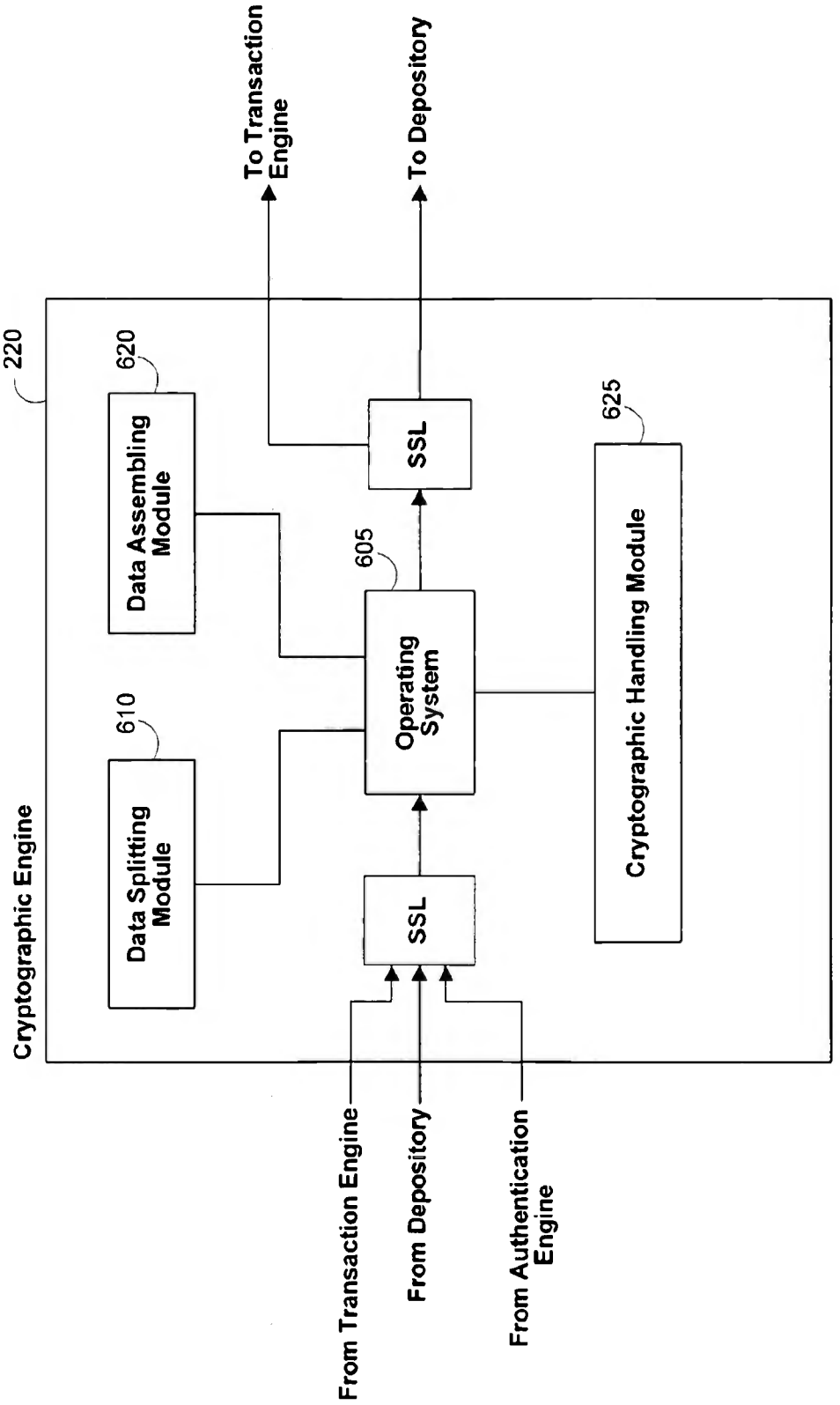

To Transaction Engine
To Depository
220
620
Data Assembling Module
SSL
625
605
Operating System
Cryptographic Handling Module
610
Data Splitting Module
SSL
Cryptographic Engine
From Transaction Engine
From Depository
From Authentication Engine

FIG. 6

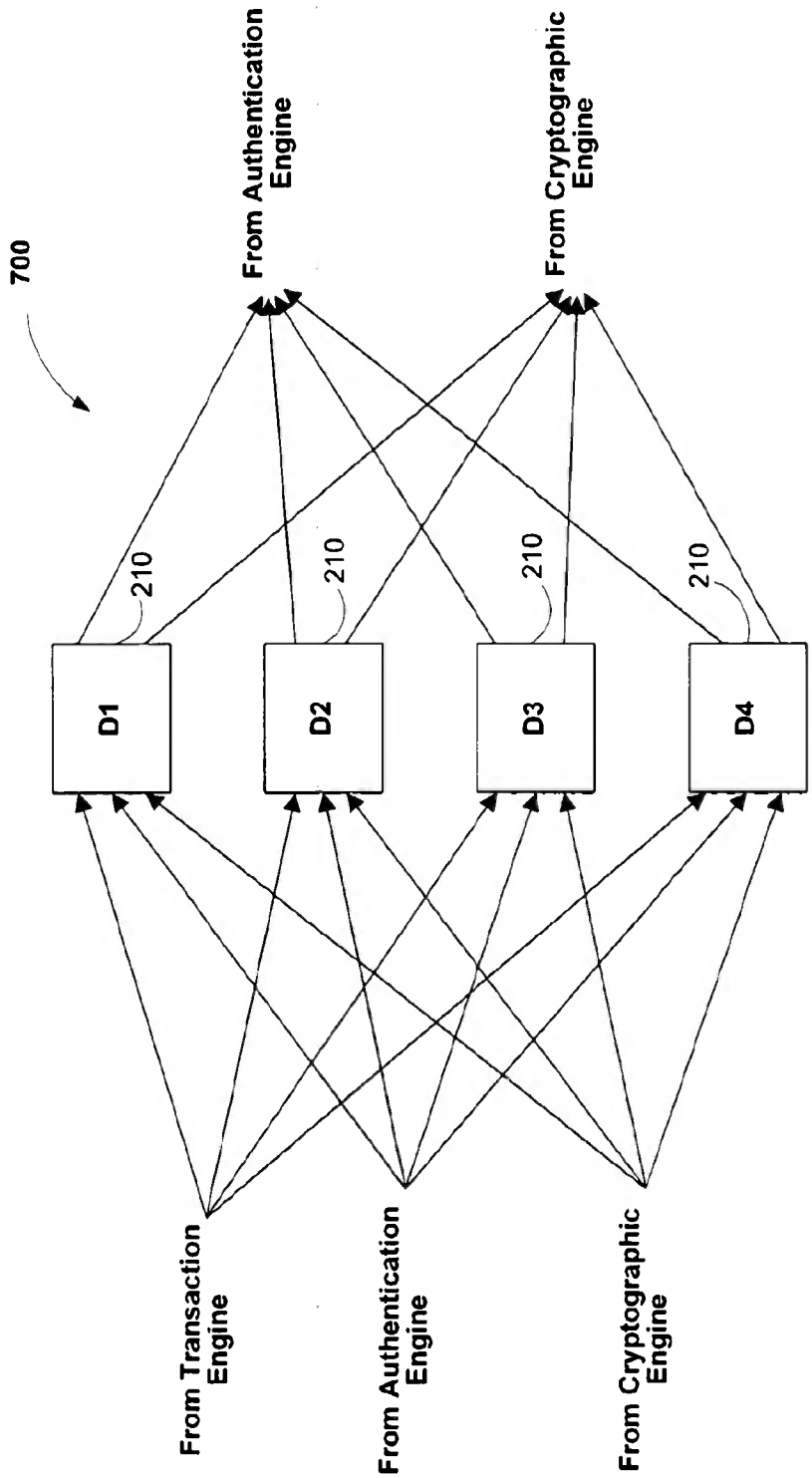

700
From Authentication Engine
From Cryptographic Engine
210
210
210
210
D1
D2
D3
D4
From Transaction Engine
From Authentication Engine
From Cryptographic Engine

FIG. 7

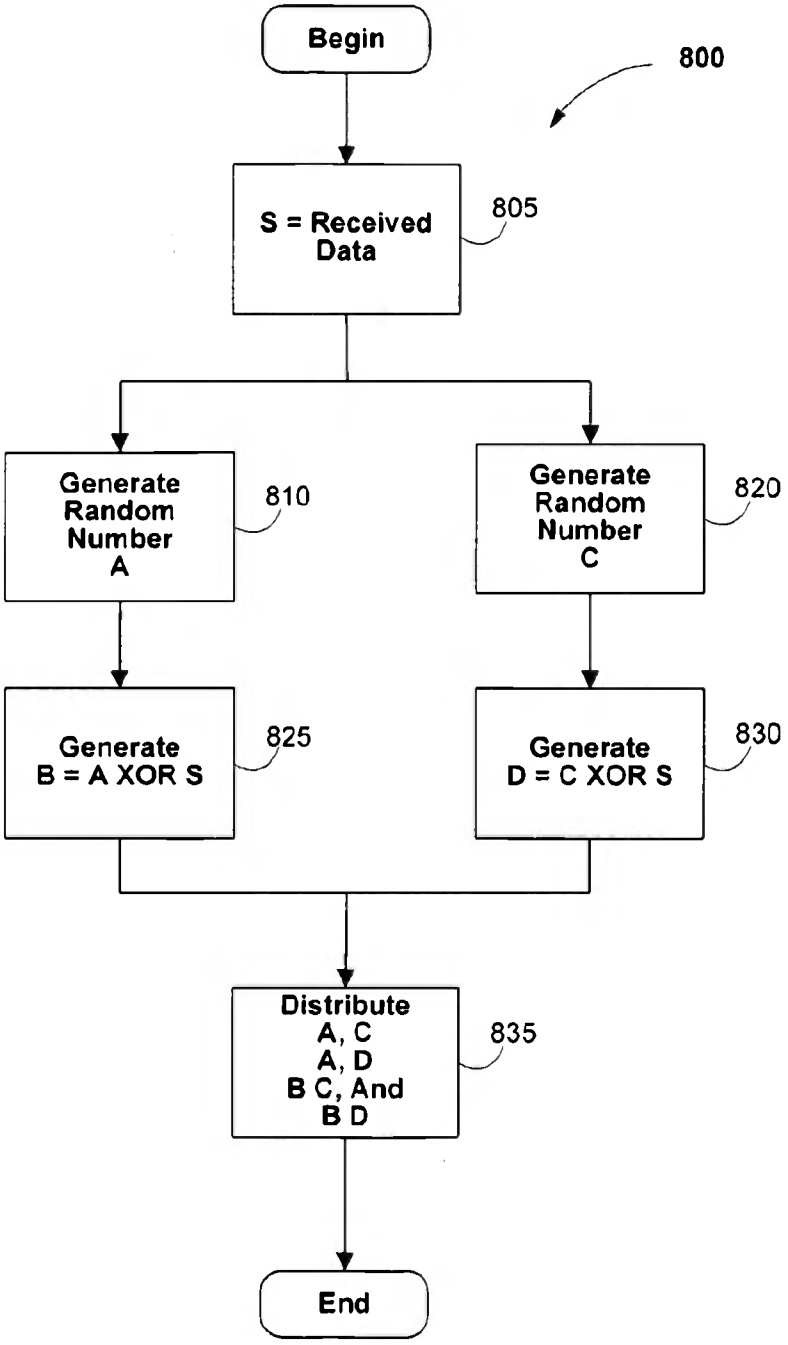
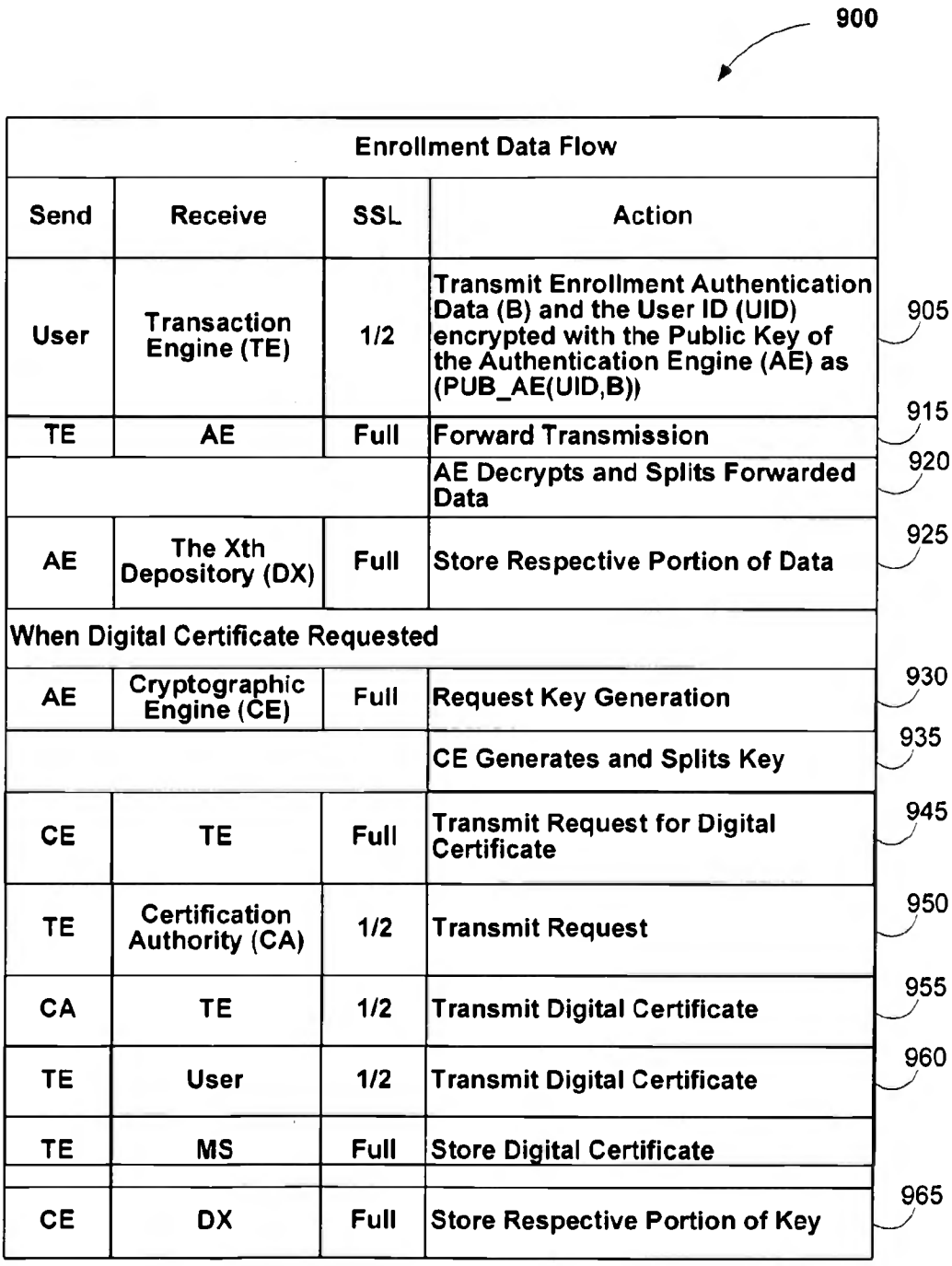

Begin
800
S = Received Data
805
Generate Random Number A
810
Generate Random Number C
820
Generate B = A XOR S
825
Generate D = C XOR S
830
Distribute A, C A, D B C, And B D
835
End

FIG. 8



Enrollment Data Flow			
Send	Receive	SSL	Action
User	Transaction Engine (TE)	1/2	Transmit Enrollment Authentication Data (B) and the User ID (UID) encrypted with the Public Key of the Authentication Engine (AE) as (PUB_AE(UID,B))
TE	AE	Full	Forward Transmission
			AE Decrypts and Splits Forwarded Data
AE	The Xth Depository (DX)	Full	Store Respective Portion of Data
When Digital Certificate Requested			
AE	Cryptographic Engine (CE)	Full	Request Key Generation
			CE Generates and Splits Key
CE	TE	Full	Transmit Request for Digital Certificate
TE	Certification Authority (CA)	1/2	Transmit Request
CA	TE	1/2	Transmit Digital Certificate
TE	User	1/2	Transmit Digital Certificate
TE	MS	Full	Store Digital Certificate
CE	DX	Full	Store Respective Portion of Key

FIG. 9, Panel A

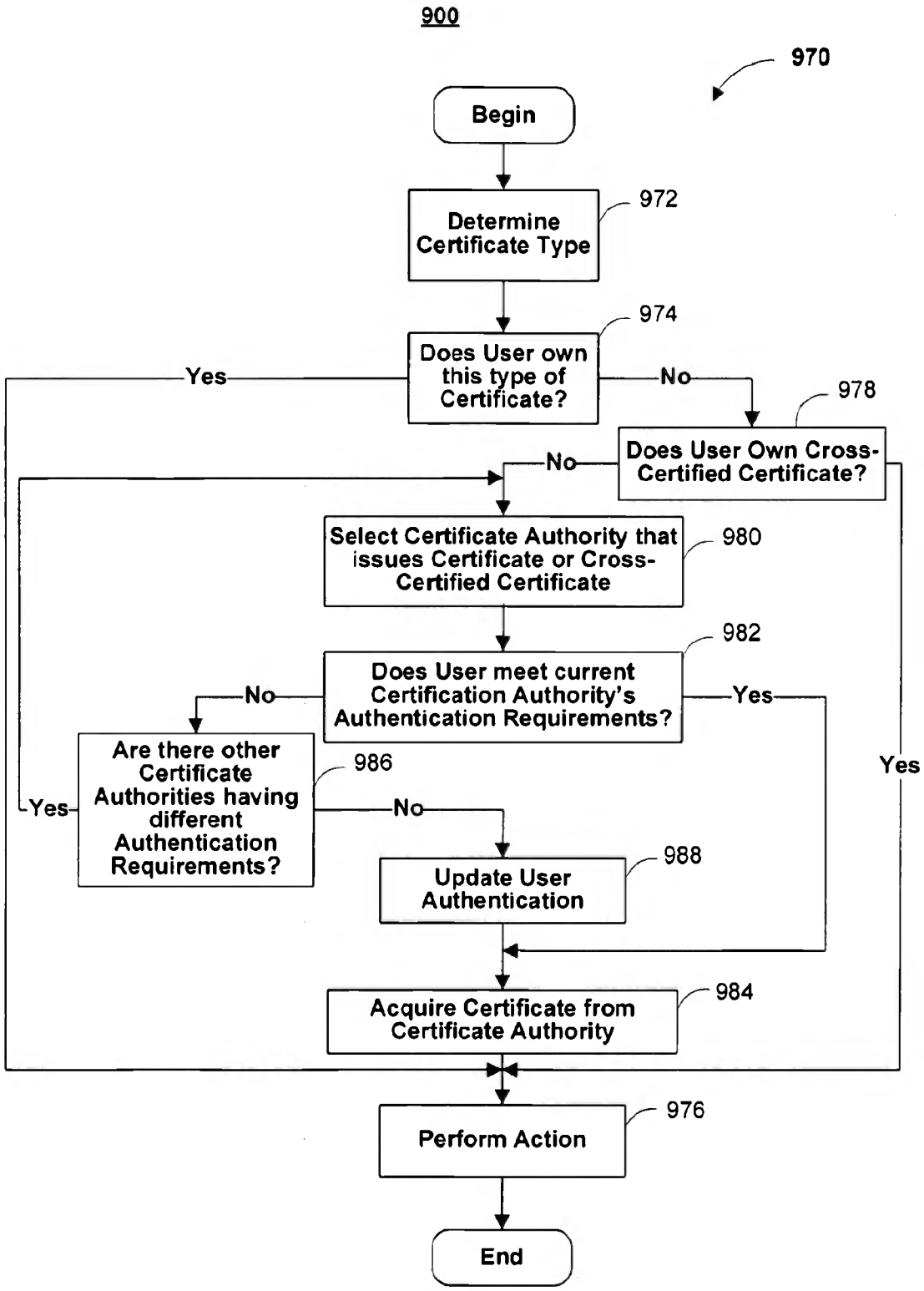

900
970
Begin
Determine Certificate Type
972
Does User own this type of Certificate?
974
Yes
No
Does User Own Cross-Certified Certificate?
978
No
Yes
Select Certificate Authority that issues Certificate or Cross-Certified Certificate
980
Does User meet current Certification Authority's Authentication Requirements?
982
No
Yes
Are there other Certificate Authorities having different Authentication Requirements?
986
Yes
No
Update User Authentication
988
Acquire Certificate from Certificate Authority
984
Perform Action
976
End

FIG. 9, Panel B



	Authentication Data Flow			
	SEND	RECEIVE	SSL	ACTION
1005	User	Vendor	1/2	Transaction occurs, such as selecting purchase
1010	Vendor	User	1/2	Transmit transaction ID (TID) and authentication request (AR)
				Authentication data (B') is gathered from User
1015	User	TE	1/2	Transmit TID and B' wrapped in the Public Key of the Authentication Engine (AE), as (PUB_AE(TID, B'))
1020	TE	AE	Full	Forward transmission
				Enrollment authentication data (B) is requested and gathered
1025	Vendor	Transaction Engine (TE)	Full	Transmits TID, AR
1030	TE	Mass Storage (MS)	Full	Create Record in database
1035	TE	The Xth Depository (DX)	Full	UID, TID
1040	DX	AE	Full	Transmit the TID and the portion of the authentication data stored at enrollment (BX) as (PUB_AE(TID, BX))
1045				AE assembles B and compares to B'
1050	AE	TE	Full	TID, the filled in AR
1055	TE	Vendor	Full	TID, Yes/No
	TE	User	1/2	TID, confirmation message

FIG. 10

1100

Signing Data Flow			
SEND	RECEIVE	SSL	ACTION
User	Vendor	1/2	Transaction occurs, such as agreeing on a deal
Vendor	User	1/2	Transmit transaction identification number (TID), authentication request (AR), and agreement or message (M)
			Current authentication data (B') and a hash of the message received by the User (h(M')) is is gathered from User
User	TE	1/2	Transmit TID, B', AR, and h(M') wrapped in the Public Key of the Authentication Engine (AE), as (PUB_AE(TID, B', h(M'))
TE	AE	Full	Forward transmission
			Gather enrollment authentication data
Vendor	Transaction Engine (TE)	Full	Transmits UID, TID, AR, and a hash of the message (h(M')).
TE	Mass Storage (MS)	Full	Create Record in database
TE	The Xth Depository (DX)	Full	UID, TID
DX	AE	Full	Transmit the TID and the portion of the authentication data stored at Enrollment (BX), as (PUB_AE(TID, BX))
			The original vendor message is transmitted to the AE
TE	AE	Full	Transmit h(M)
1103			AE assembles B, compares to B' and compares h(M) to h(M')
1105 AE	Cryptographic Engine (CE)	Full	Request for digital signature and a message to be signed, for example, the hashed message
1110 AE	DX	Full	TID, signing UID
1115 DX	CE	Full	Transmit the portion of the Crypto-graphic Key corresponding to the signing party
1120			CE assembles key and signs
1125 CE	AE	Full	Transmit the digital signature (S) of signing party
1130 AE	TE	Full	TID, the filled in AR, h(M), and S
1135 TE	Vendor	Full	TID, a receipt=(TID, Yes/No, and S), and the digital signature of the trust engine, for example, a hash of the receipt encrypted with the trust engine's Private Key (Priv_TE(h(receipt)))
1140 TE	User	1/2	TID, confirmation message

FIG. 11

1200

Encryption/Decryption Data Flow				
Send	Receive	SSL	Action	
Decryption				
			Perform Authentication Data Process 1000, include the Session Key (sync) in the AR, where the sync has been encrypted with the Public Key of the User as PUB_USER(SYNC)	
			Authenticate the User	1205
AE	CE	Full	Forward PUB_USER(SYNC) to CE	1210
AE	DX	Full	UID, TID	1215
DX	CE	Full	Transmit the TID and the portion of the Private Key as (PUB_AE(TID, KEY_USER))	1220
			CE assembles the Cryptographic Key and decrypts the sync	1225
CE	AE	Full	TID, the filled in AR including decrypted sync	1230
AE	TE	Full	Forward to TE	
TE	Requesting APP/Vendor	1/2	TID, Yes/No, Sync	
Encryption				
Requesting APP/ Vendor	TE	1/2	Request for Public Key of User	1235
TE	MS	Full	Request Digital Certificate	1240
MS	TE	Full	Transmit Digital Certificate	1245
TE	Requesting APP/Vendor	1/2	Transmit Digital Certificate	1250

FIG. 12

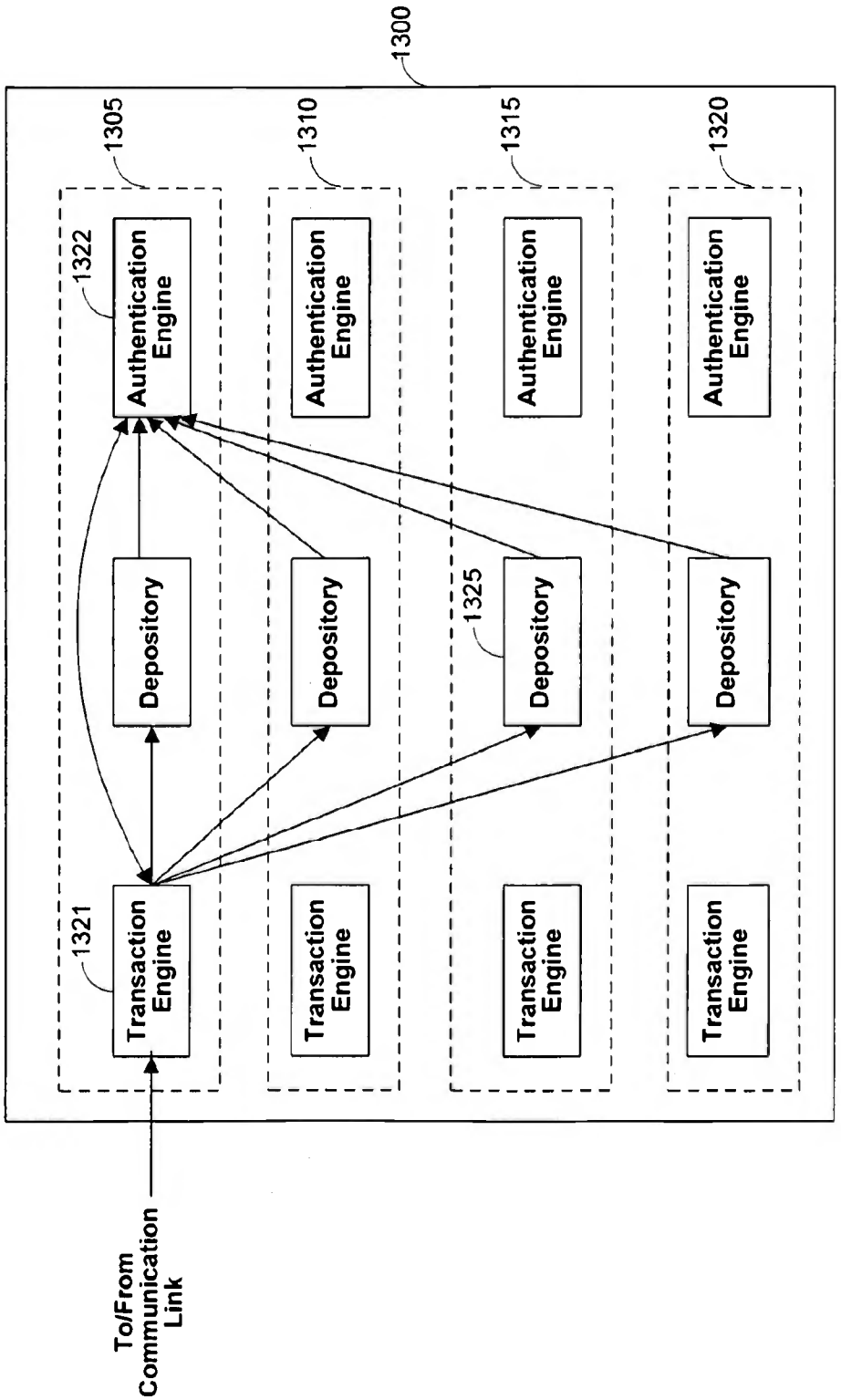

1300
1305
1310
1315
1320
1322
Authentication Engine
Depository
1325
1321
Transaction Engine
To/From Communication Link

FIG. 13

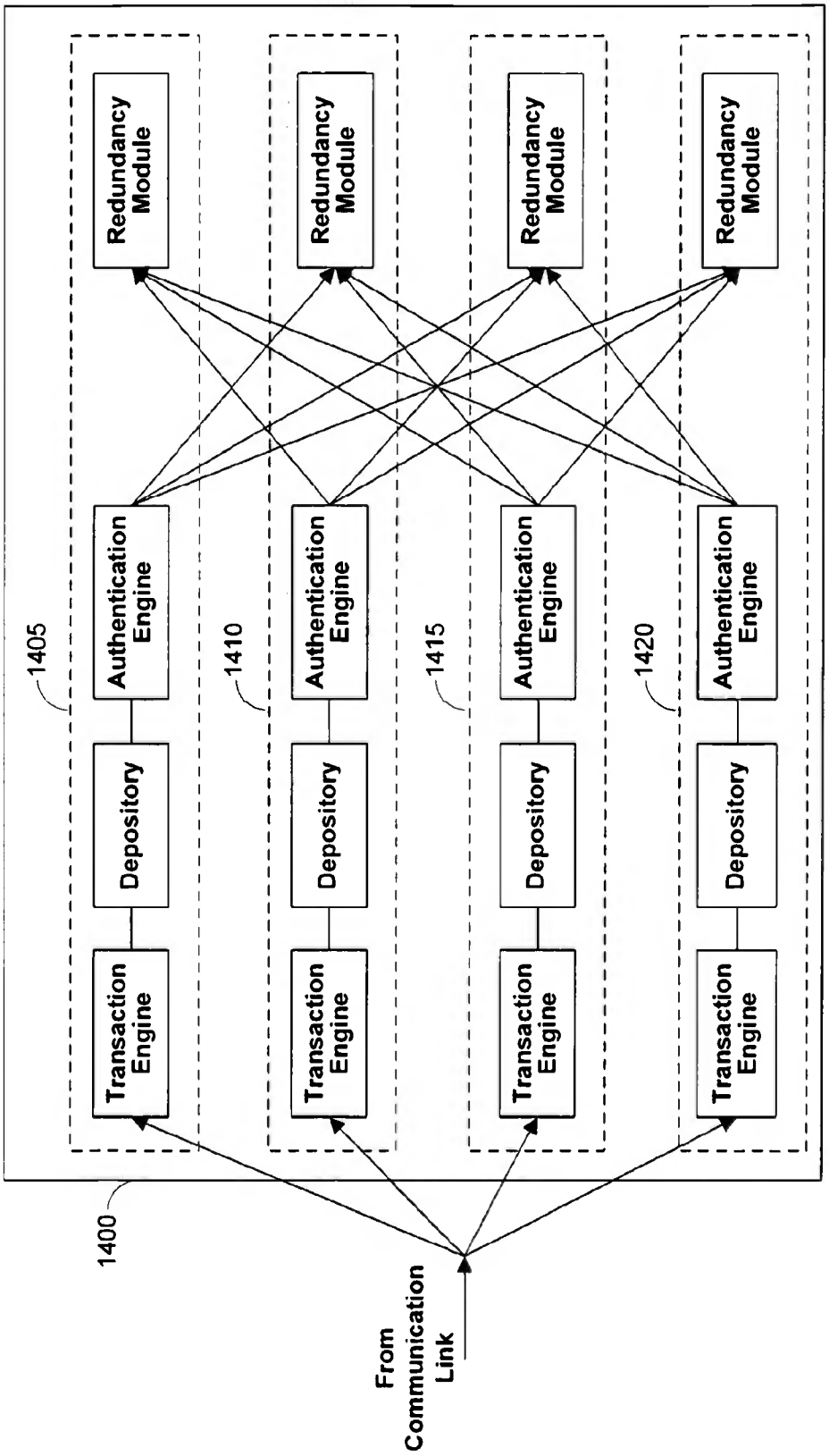

Redundancy Module
Redundancy Module
Redundancy Module
Redundancy Module
Authentication Engine
Authentication Engine
Authentication Engine
Authentication Engine
Depository
Depository
Depository
Depository
Transaction Engine
Transaction Engine
Transaction Engine
Transaction Engine
1405
1410
1415
1420
1400
From Communication Link

FIG. 14

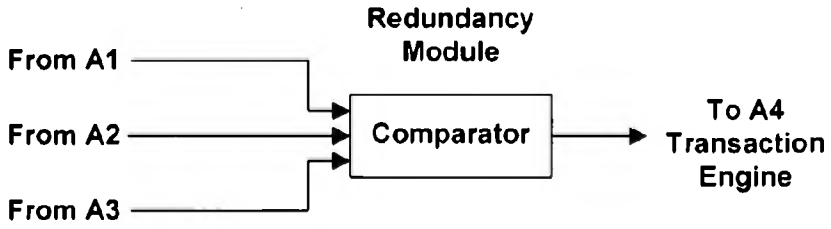

Redundancy
Module
From A1
From A2
From A3
Comparator
To A4
Transaction
Engine

FIG. 15

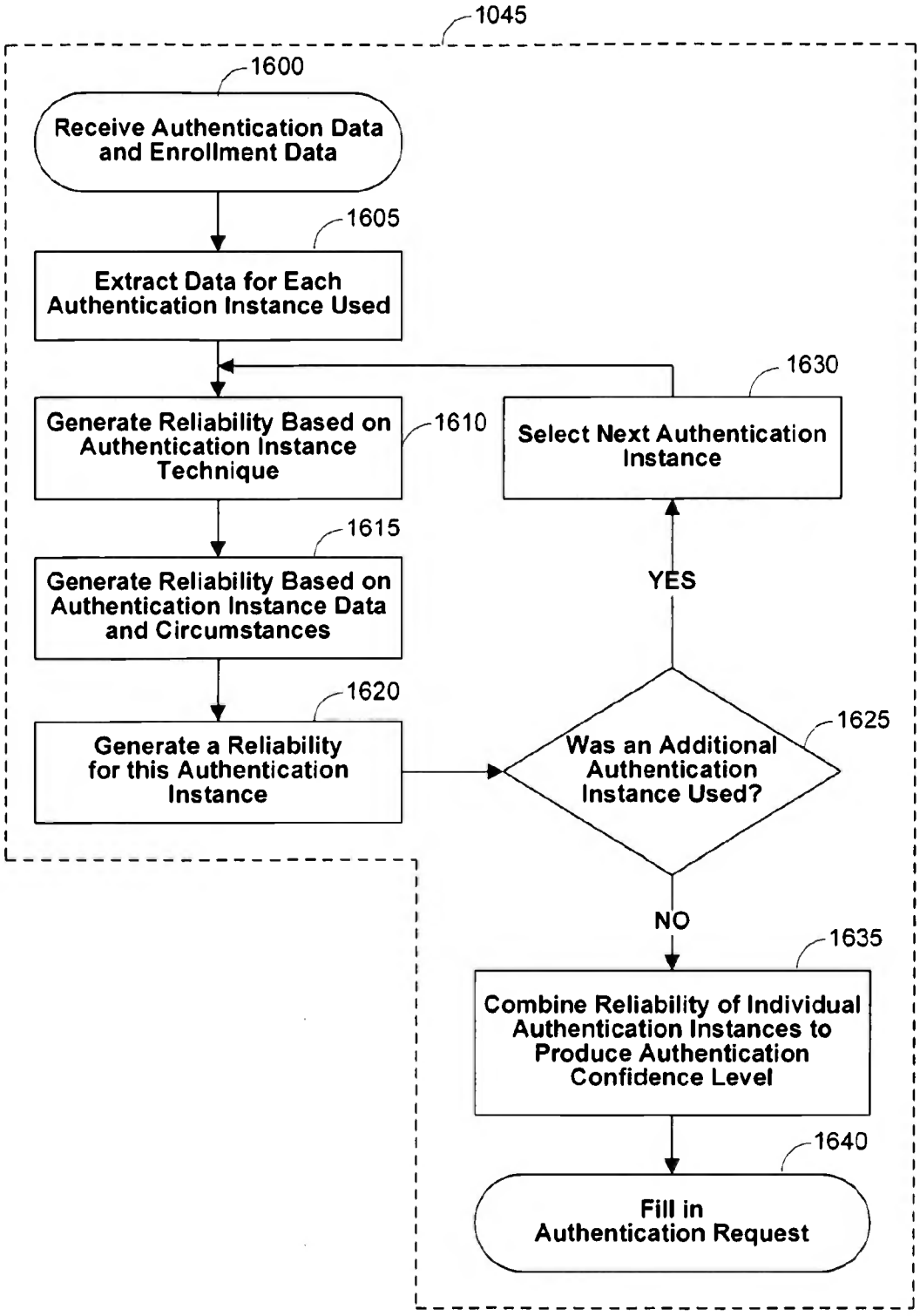

1045
1600
Receive Authentication Data and Enrollment Data
1605
Extract Data for Each Authentication Instance Used
Generate Reliability Based on Authentication Instance Technique
1610
Generate Reliability Based on Authentication Instance Data and Circumstances
1615
Generate a Reliability for this Authentication Instance
1620
Was an Additional Authentication Instance Used?
1625
YES
Select Next Authentication Instance
1630
NO
Combine Reliability of Individual Authentication Instances to Produce Authentication Confidence Level
1635
Fill in Authentication Request
1640

FIG. 16

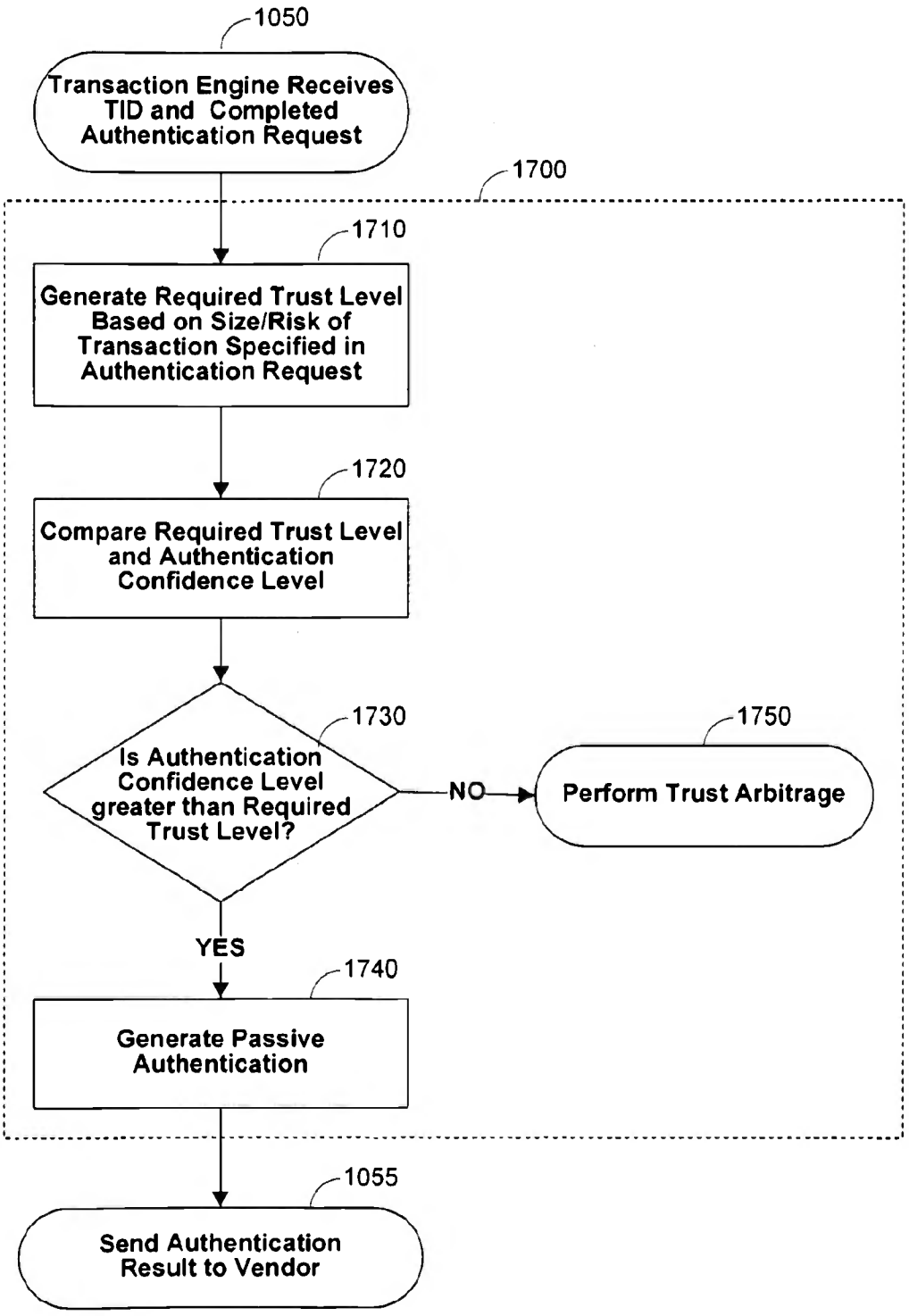

1050
Transaction Engine Receives TID and Completed Authentication Request
1700
1710
Generate Required Trust Level Based on Size/Risk of Transaction Specified in Authentication Request
1720
Compare Required Trust Level and Authentication Confidence Level
1730
Is Authentication Confidence Level greater than Required Trust Level?
NO
1750
Perform Trust Arbitrage
YES
1740
Generate Passive Authentication
1055
Send Authentication Result to Vendor

FIG. 17

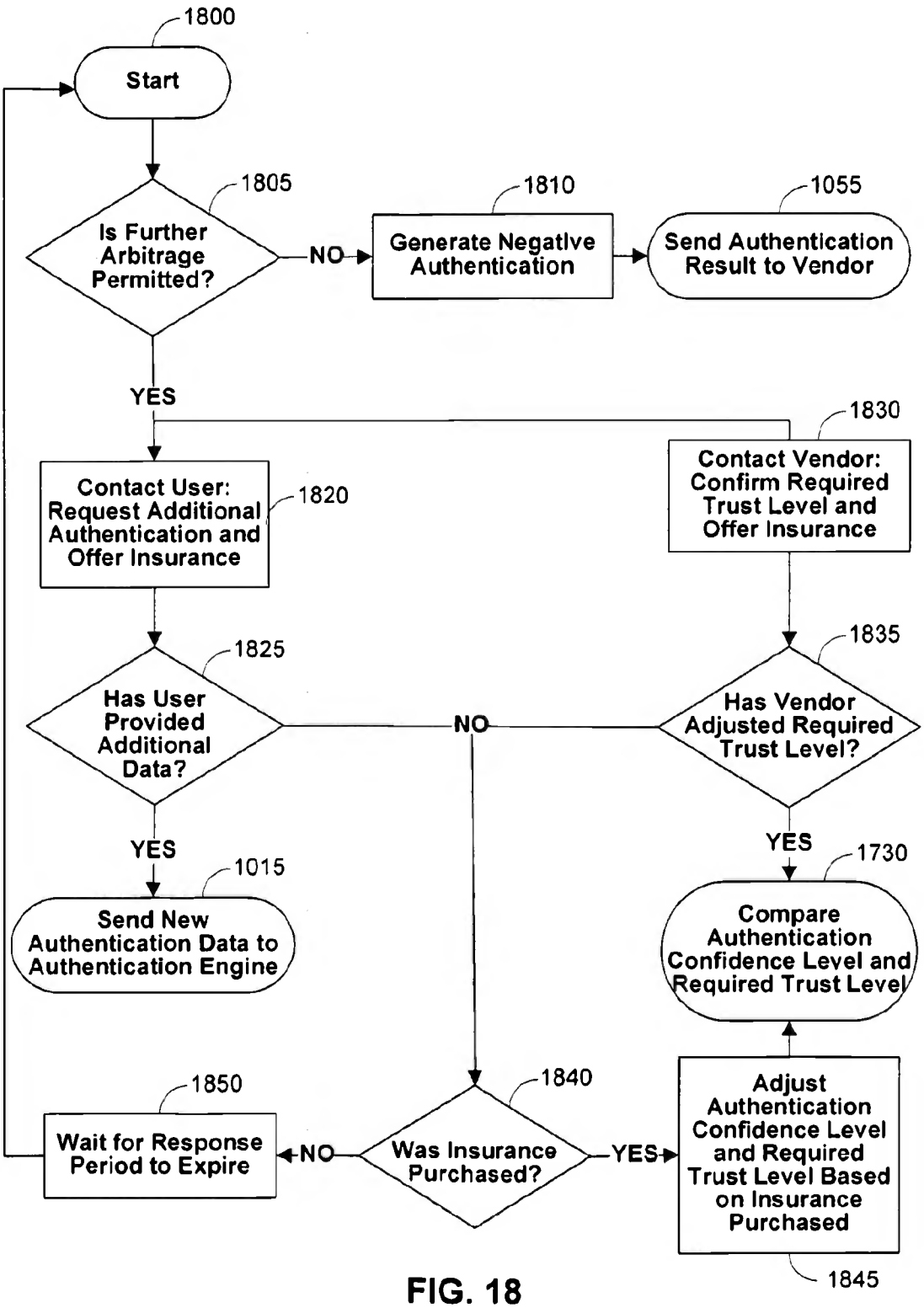

1800
Start
1805
Is Further Arbitrage Permitted?
NO
1810
Generate Negative Authentication
1055
Send Authentication Result to Vendor
YES
1830
Contact Vendor: Confirm Required Trust Level and Offer Insurance
Contact User: Request Additional Authentication and Offer Insurance
1820
1825
Has User Provided Additional Data?
NO
1835
Has Vendor Adjusted Required Trust Level?
YES
1015
Send New Authentication Data to Authentication Engine
YES
1730
Compare Authentication Confidence Level and Required Trust Level
1850
Wait for Response Period to Expire
NO
1840
Was Insurance Purchased?
YES
Adjust Authentication Confidence Level and Required Trust Level Based on Insurance Purchased
1845

FIG. 18

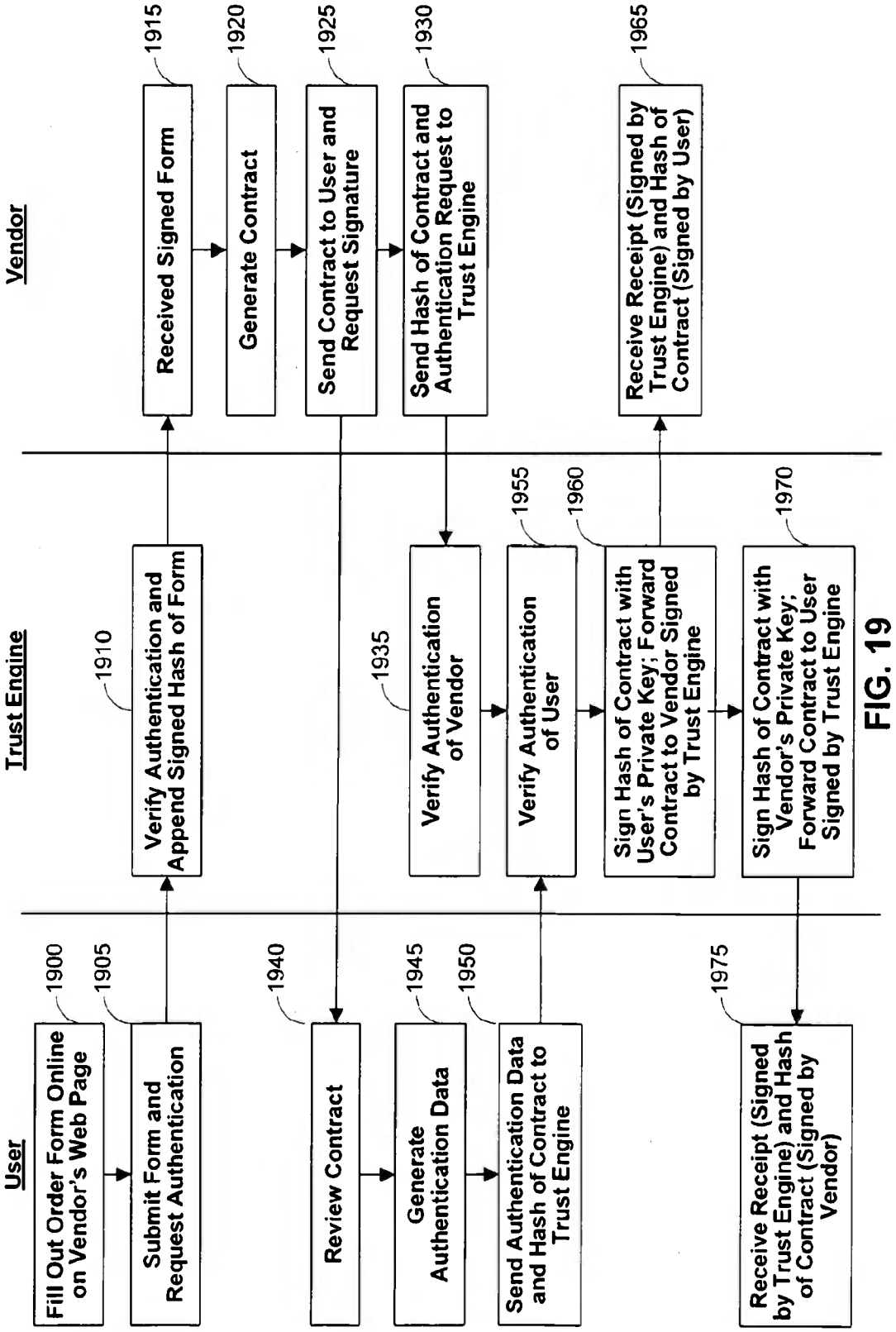

User
Trust Engine
Vendor
Fill Out Order Form Online on Vendor's Web Page
1900
Submit Form and Request Authentication
1905
Verify Authentication and Append Signed Hash of Form
1910
Received Signed Form
1915
Generate Contract
1920
Send Contract to User and Request Signature
1925
Send Hash of Contract and Authentication Request to Trust Engine
1930
Review Contract
1940
Generate Authentication Data
1945
Send Authentication Data and Hash of Contract to Trust Engine
1950
Verify Authentication of Vendor
1935
Verify Authentication of User
1955
Sign Hash of Contract with User's Private Key; Forward Contract to Vendor Signed by Trust Engine
1960
Receive Receipt (Signed by Trust Engine) and Hash of Contract (Signed by User)
1965
Sign Hash of Contract with Vendor's Private Key; Forward Contract to User Signed by Trust Engine
1970
Receive Receipt (Signed by Trust Engine) and Hash of Contract (Signed by Vendor)
1975

FIG. 19

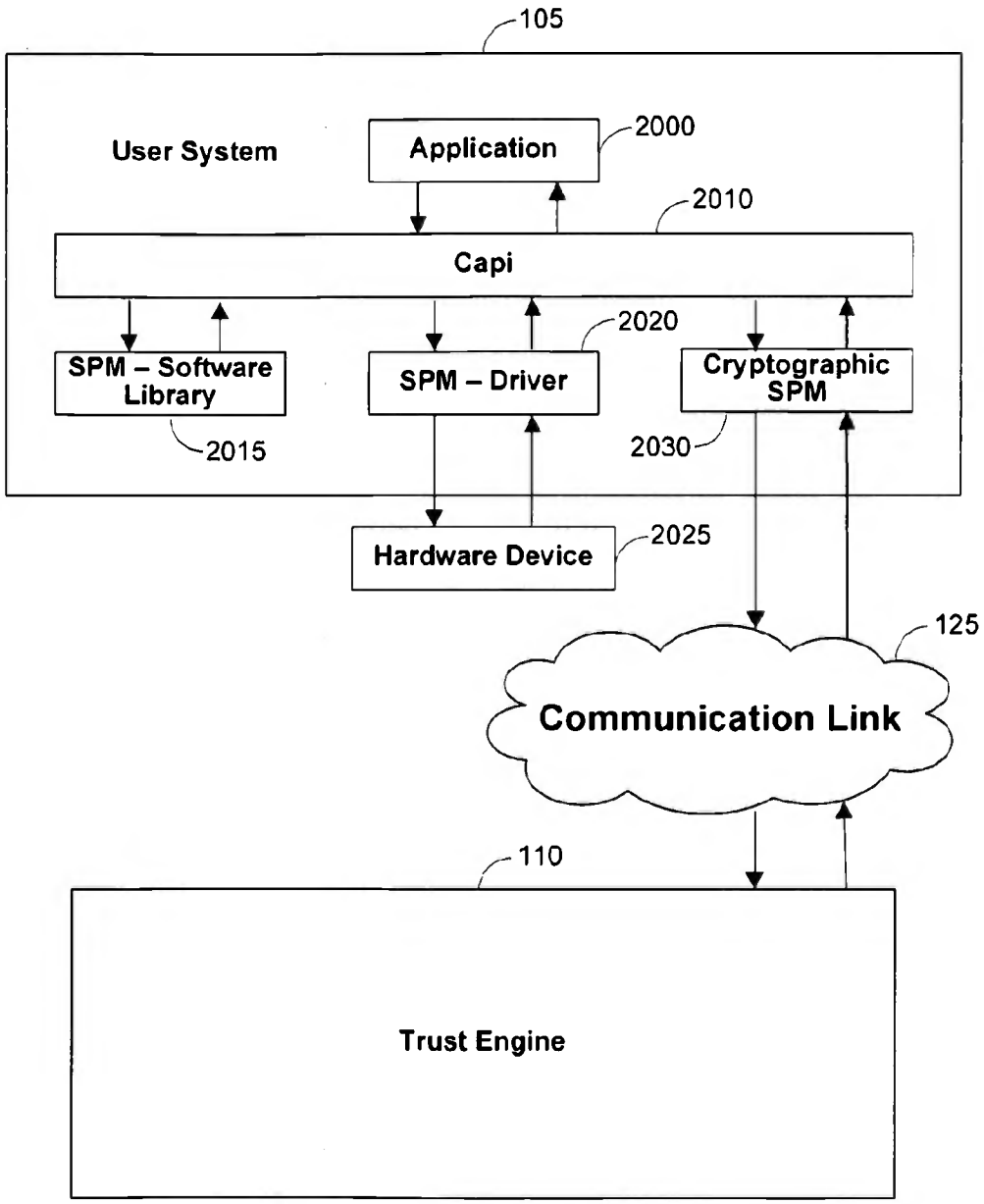

105
User System
Application
2000
2010
Capi
2020
SPM – Software Library
2015
SPM – Driver
Cryptographic SPM
2030
Hardware Device
2025
125
Communication Link
110
Trust Engine

FIG. 20

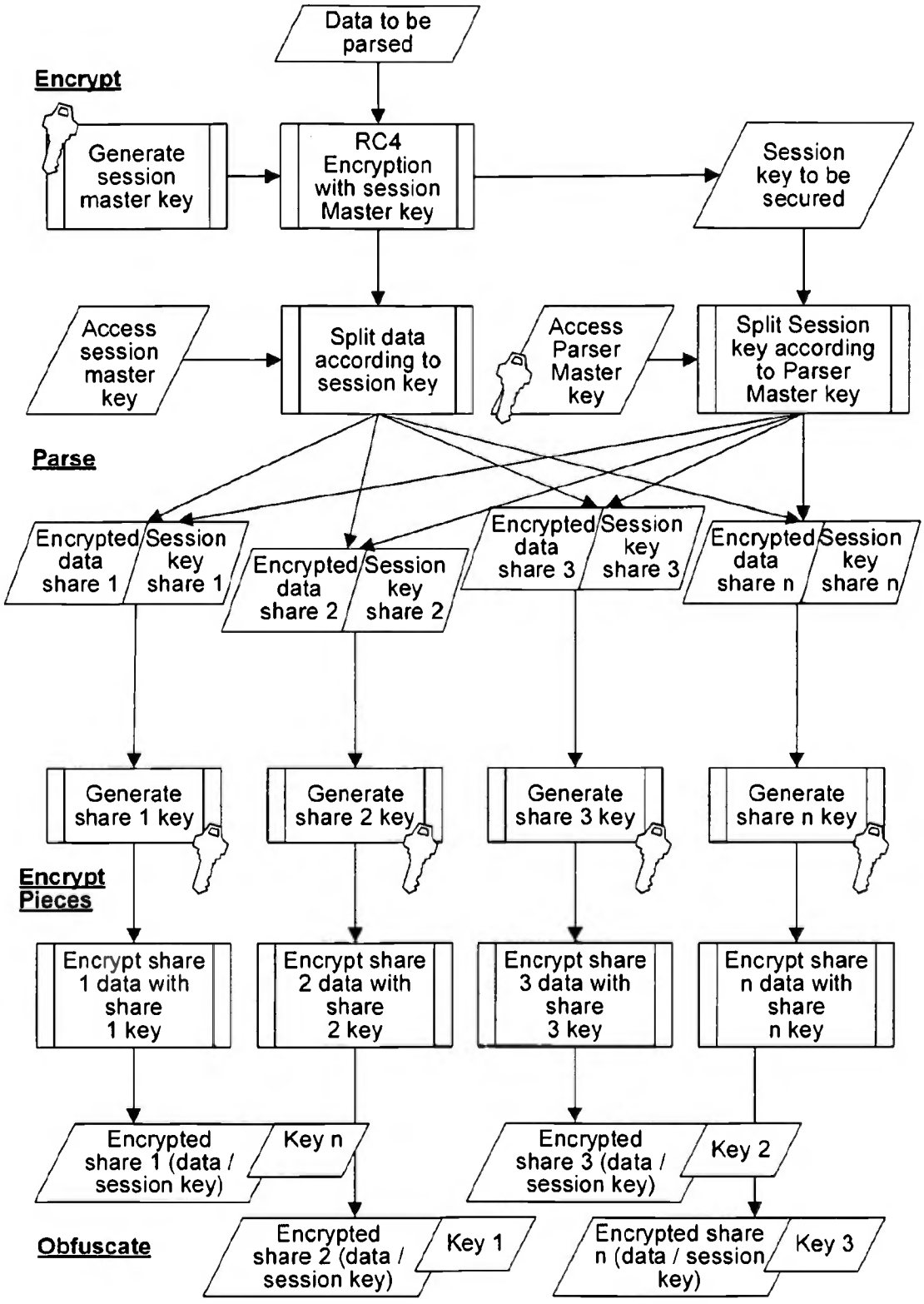

Data to be parsed
Encrypt
Generate session master key
RC4 Encryption with session Master key
Session key to be secured
Access session master key
Split data according to session key
Access Parser Master key
Split Session key according to Parser Master key
Parse
Encrypted data share 1
Session key share 1
Encrypted data share 2
Session key share 2
Encrypted data share 3
Session key share 3
Encrypted data share n
Session key share n
Generate share 1 key
Generate share 2 key
Generate share 3 key
Generate share n key
Encrypt Pieces
Encrypt share 1 data with share 1 key
Encrypt share 2 data with share 2 key
Encrypt share 3 data with share 3 key
Encrypt share n data with share n key
Encrypted share 1 (data / session key)
Key n
Encrypted share 3 (data / session key)
Key 2
Obfuscate
Encrypted share 2 (data / session key)
Key 1
Encrypted share n (data / session key)
Key 3

FIG. 21

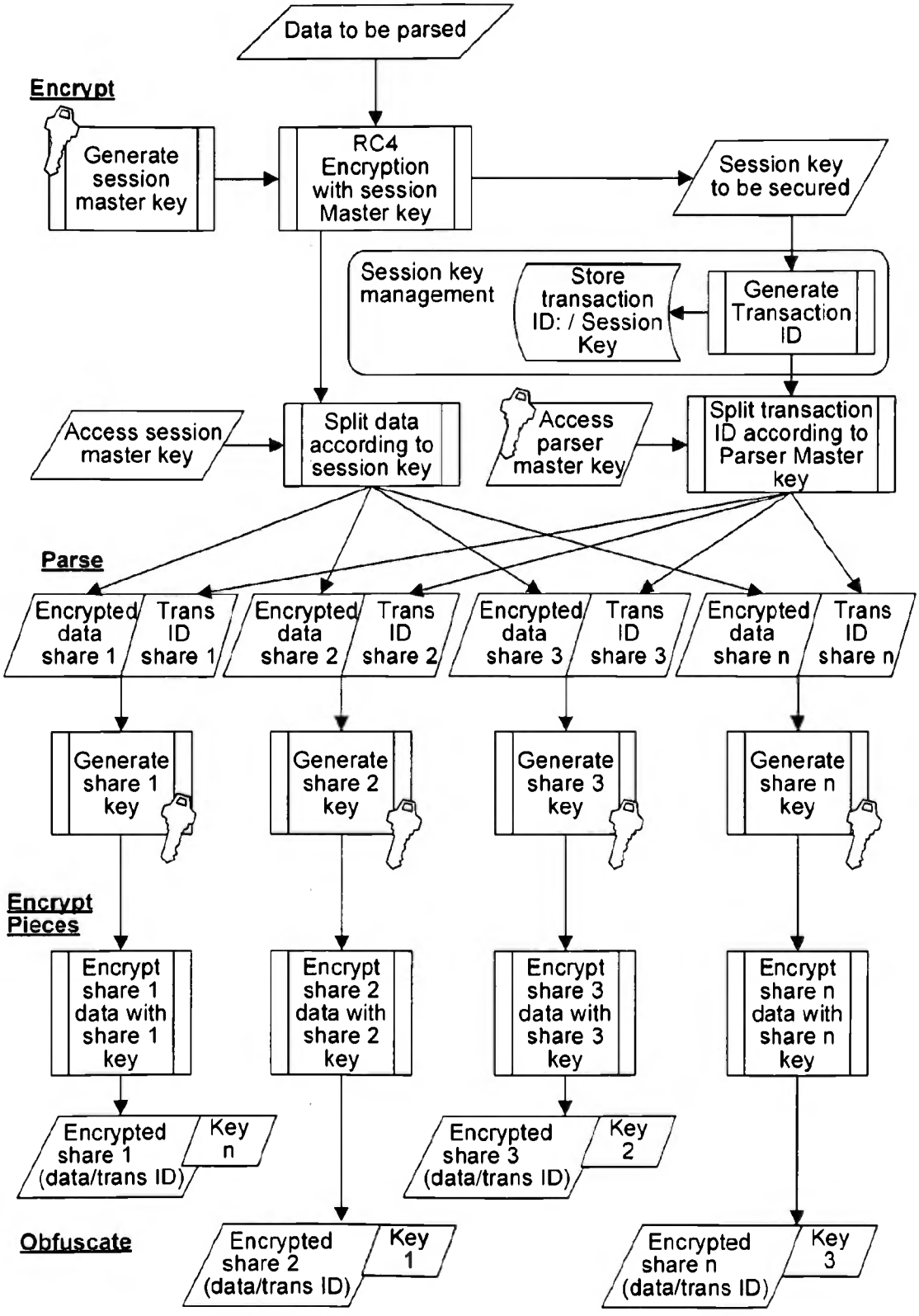

Data to be parsed
Encrypt
Generate session master key
RC4 Encryption with session Master key
Session key to be secured
Session key management
Store transaction ID: / Session Key
Generate Transaction ID
Access session master key
Split data according to session key
Access parser master key
Split transaction ID according to Parser Master key
Parse
Encrypted data share 1
Trans ID share 1
Encrypted data share 2
Trans ID share 2
Encrypted data share 3
Trans ID share 3
Encrypted data share n
Trans ID share n
Generate share 1 key
Generate share 2 key
Generate share 3 key
Generate share n key
Encrypt Pieces
Encrypt share 1 data with share 1 key
Encrypt share 2 data with share 2 key
Encrypt share 3 data with share 3 key
Encrypt share n data with share n key
Encrypted share 1 (data/trans ID)
Key n
Encrypted share 3 (data/trans ID)
Key 2
Obfuscate
Encrypted share 2 (data/trans ID)
Key 1
Encrypted share n (data/trans ID)
Key 3

FIG. 22

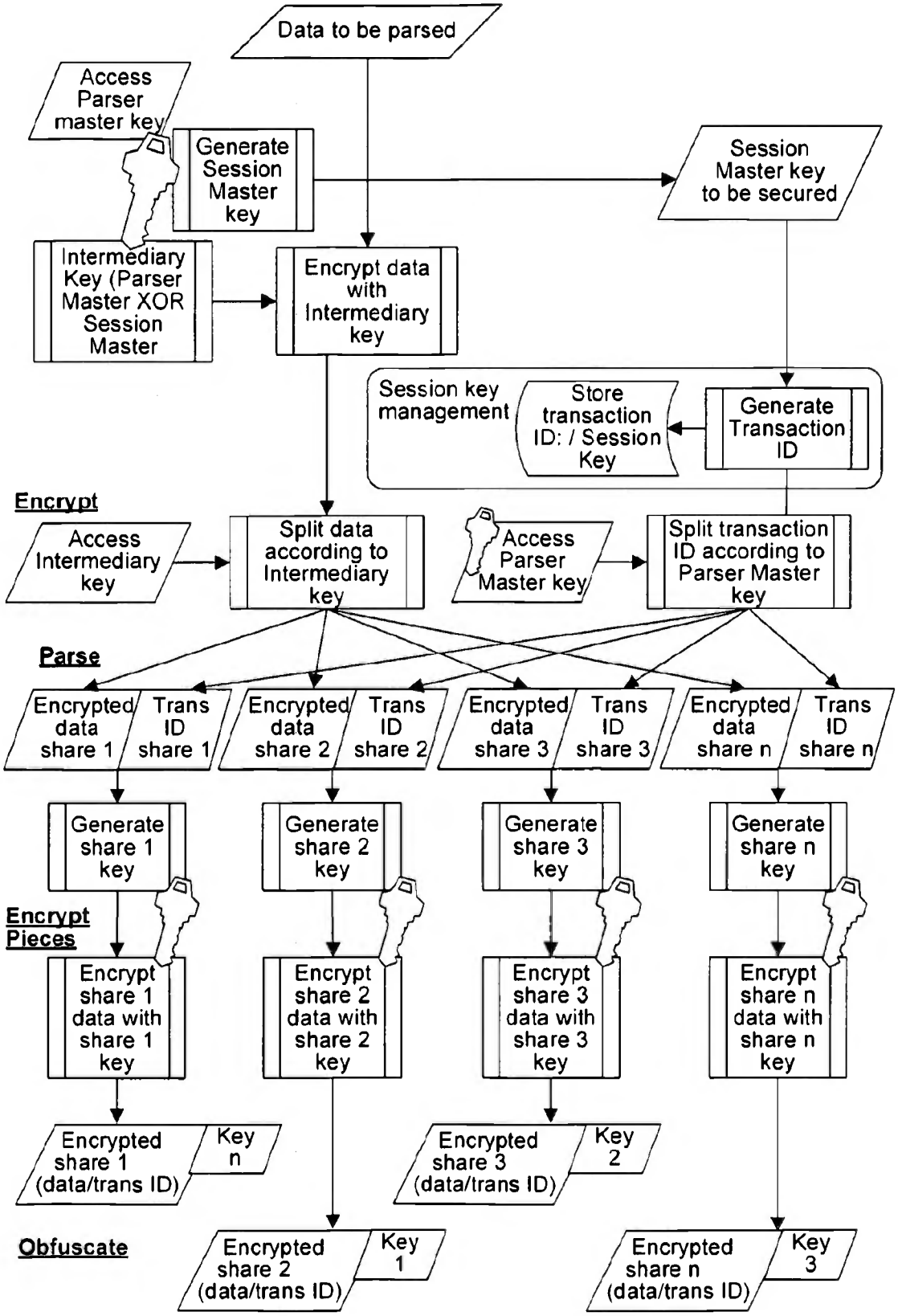

Data to be parsed
Access Parser master key
Generate Session Master key
Session Master key to be secured
Intermediary Key (Parser Master XOR Session Master
Encrypt data with Intermediary key
Session key management
Store transaction ID: / Session Key
Generate Transaction ID
Encrypt
Access Intermediary key
Split data according to Intermediary key
Access Parser Master key
Split transaction ID according to Parser Master key
Parse
Encrypted data share 1
Trans ID share 1
Encrypted data share 2
Trans ID share 2
Encrypted data share 3
Trans ID share 3
Encrypted data share n
Trans ID share n
Generate share 1 key
Generate share 2 key
Generate share 3 key
Generate share n key
Encrypt Pieces
Encrypt share 1 data with share 1 key
Encrypt share 2 data with share 2 key
Encrypt share 3 data with share 3 key
Encrypt share n data with share n key
Encrypted share 1 (data/trans ID)
Key n
Encrypted share 3 (data/trans ID)
Key 2
Obfuscate
Encrypted share 2 (data/trans ID)
Key 1
Encrypted share n (data/trans ID)
Key 3

FIG. 23

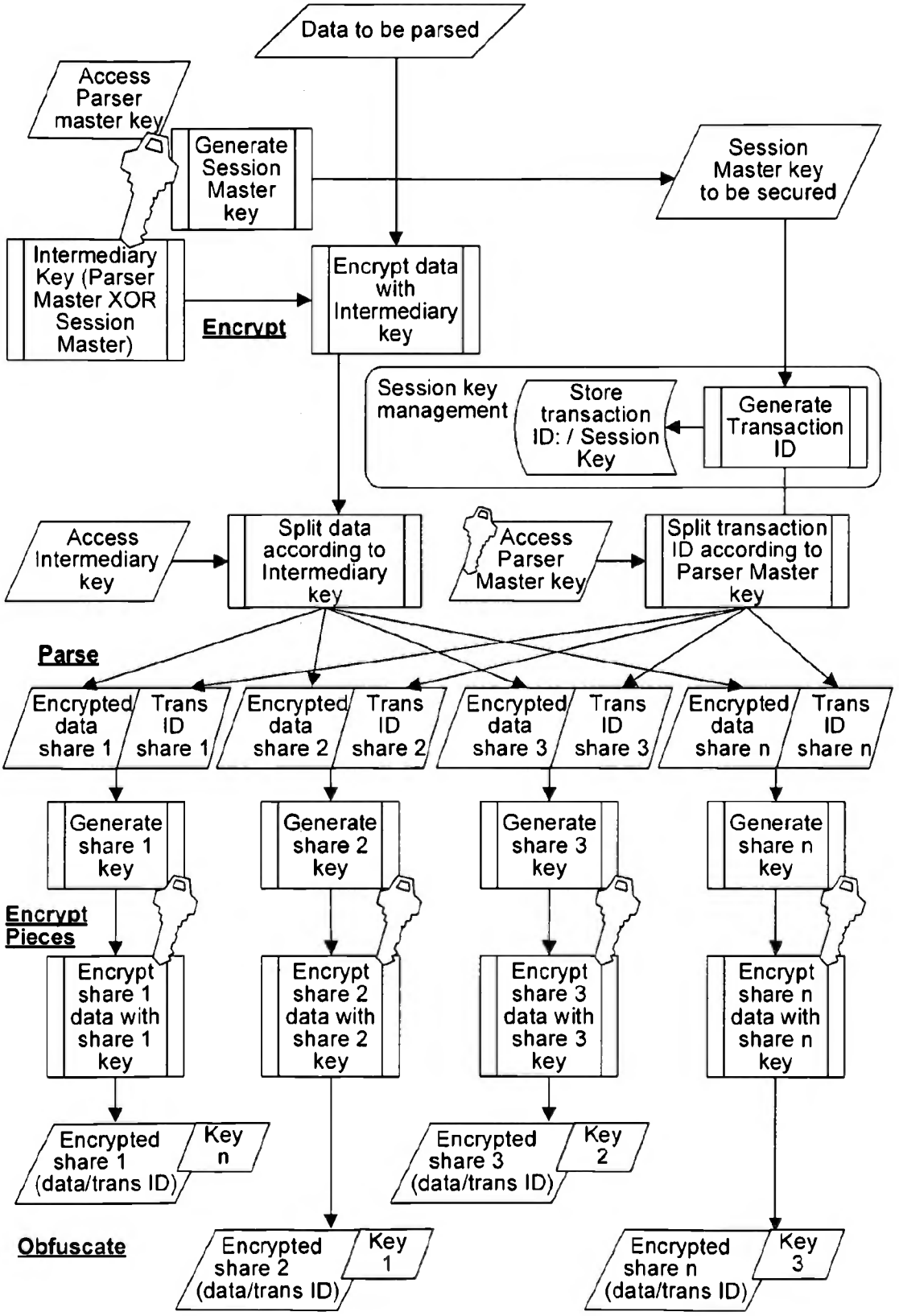

Data to be parsed
Access Parser master key
Generate Session Master key
Session Master key to be secured
Intermediary Key (Parser Master XOR Session Master)
Encrypt
Encrypt data with Intermediary key
Session key management
Store transaction ID: / Session Key
Generate Transaction ID
Access Intermediary key
Split data according to Intermediary key
Access Parser Master key
Split transaction ID according to Parser Master key
Parse
Encrypted data share 1
Trans ID share 1
Encrypted data share 2
Trans ID share 2
Encrypted data share 3
Trans ID share 3
Encrypted data share n
Trans ID share n
Generate share 1 key
Generate share 2 key
Generate share 3 key
Generate share n key
Encrypt Pieces
Encrypt share 1 data with share 1 key
Encrypt share 2 data with share 2 key
Encrypt share 3 data with share 3 key
Encrypt share n data with share n key
Encrypted share 1 (data/trans ID)
Key n
Encrypted share 3 (data/trans ID)
Key 2
Obfuscate
Encrypted share 2 (data/trans ID)
Key 1
Encrypted share n (data/trans ID)
Key 3

FIG. 24

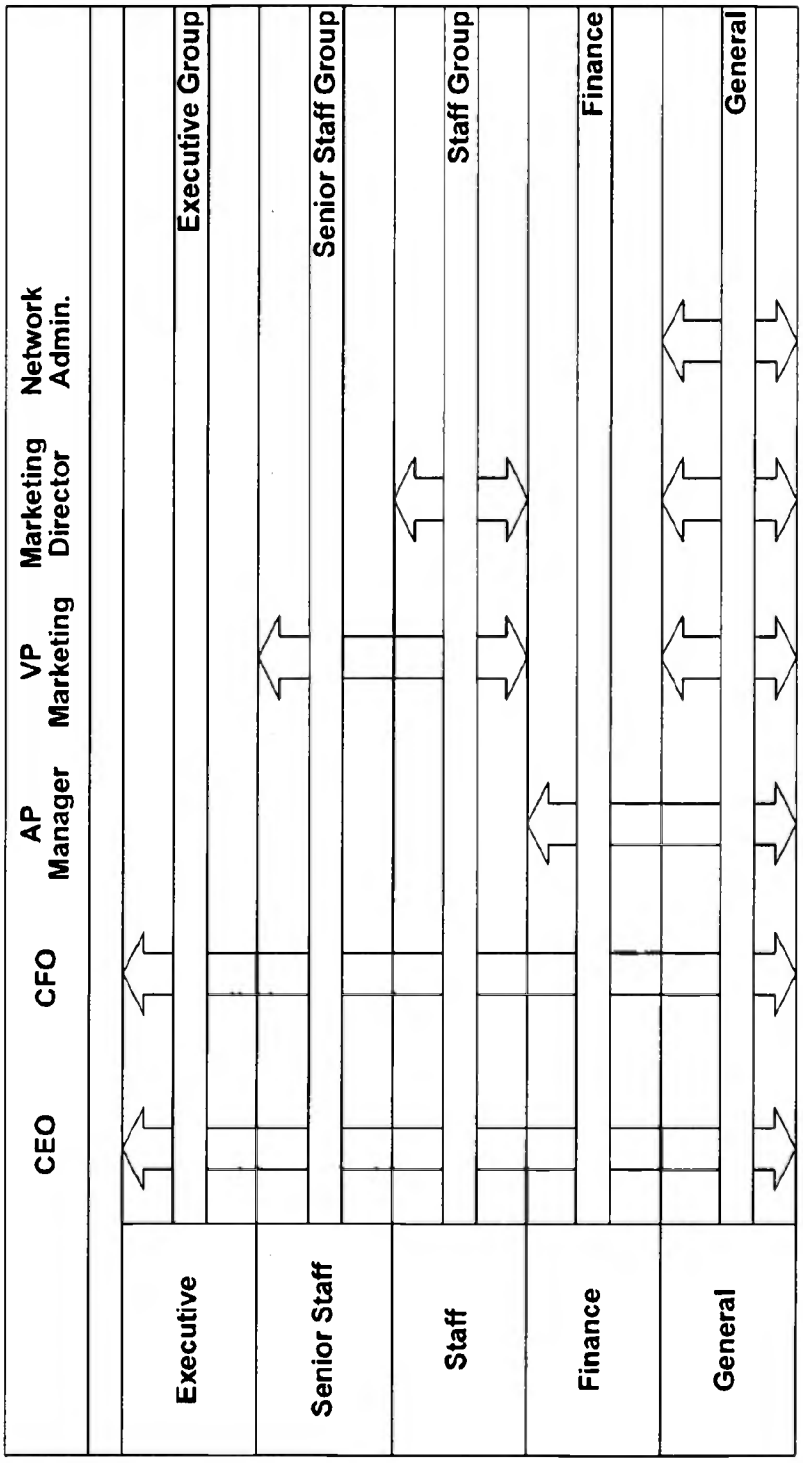

CEO
CFO
AP Manager
VP Marketing
Marketing Director
Network Admin.
Executive Group
Senior Staff Group
Staff Group
Finance
General
Executive
Senior Staff
Staff
Finance
General

FIG. 25

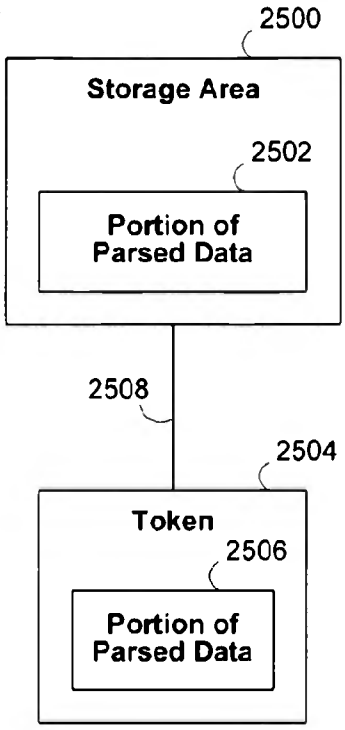

2500
Storage Area
2502
Portion of Parsed Data
2508
2504
Token
2506
Portion of Parsed Data

FIG. 26

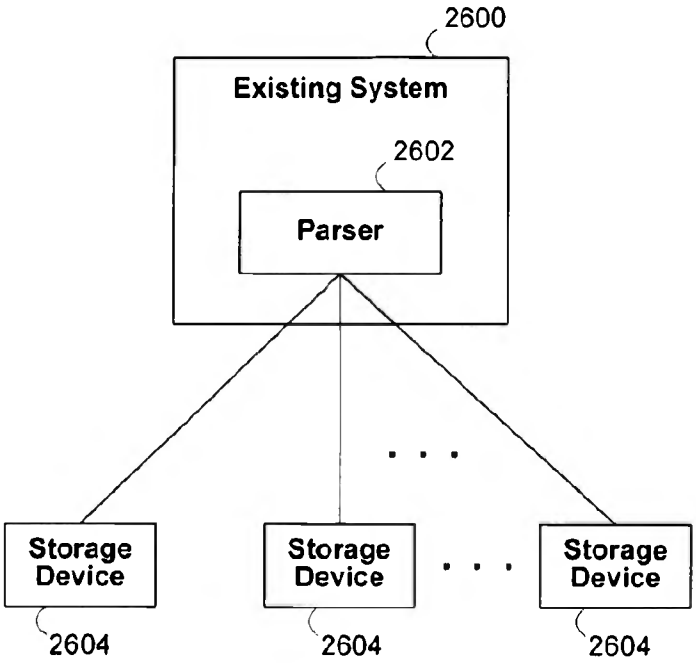

2600
Existing System
2602
Parser
Storage Device
2604
Storage Device
2604
Storage Device
2604

FIG. 27

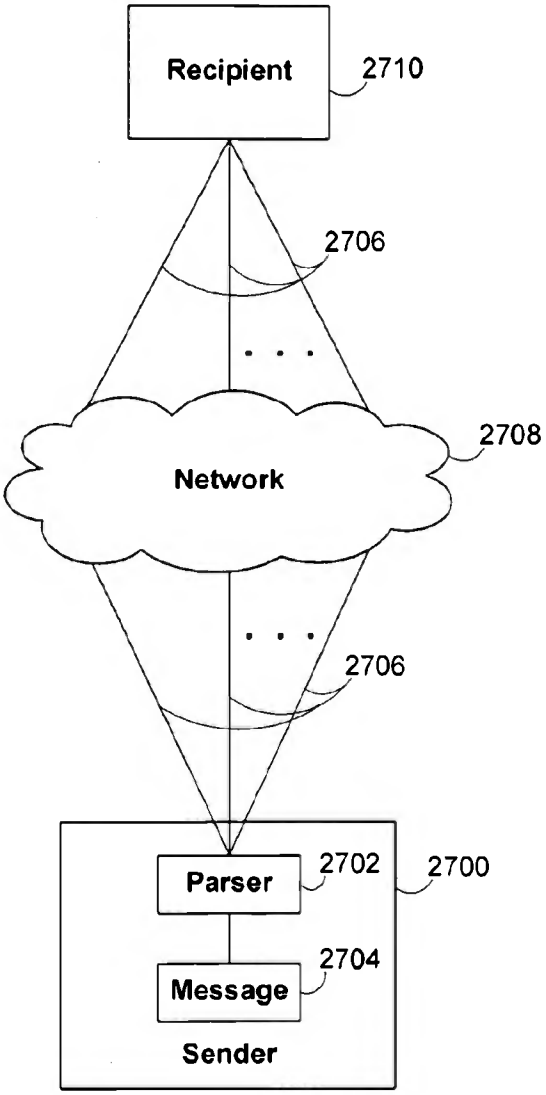

Recipient
2710
2706
2708
Network
2706
Parser
2702
2700
2704
Message
Sender

FIG. 28

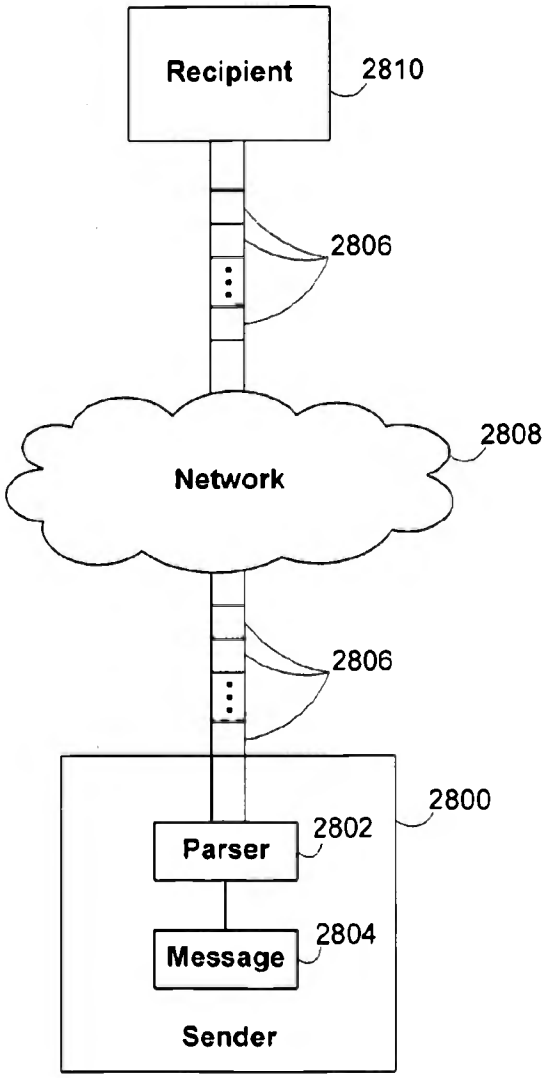

Recipient
2810
2806
2808
Network
2806
2800
Parser
2802
Message
2804
Sender

FIG. 29

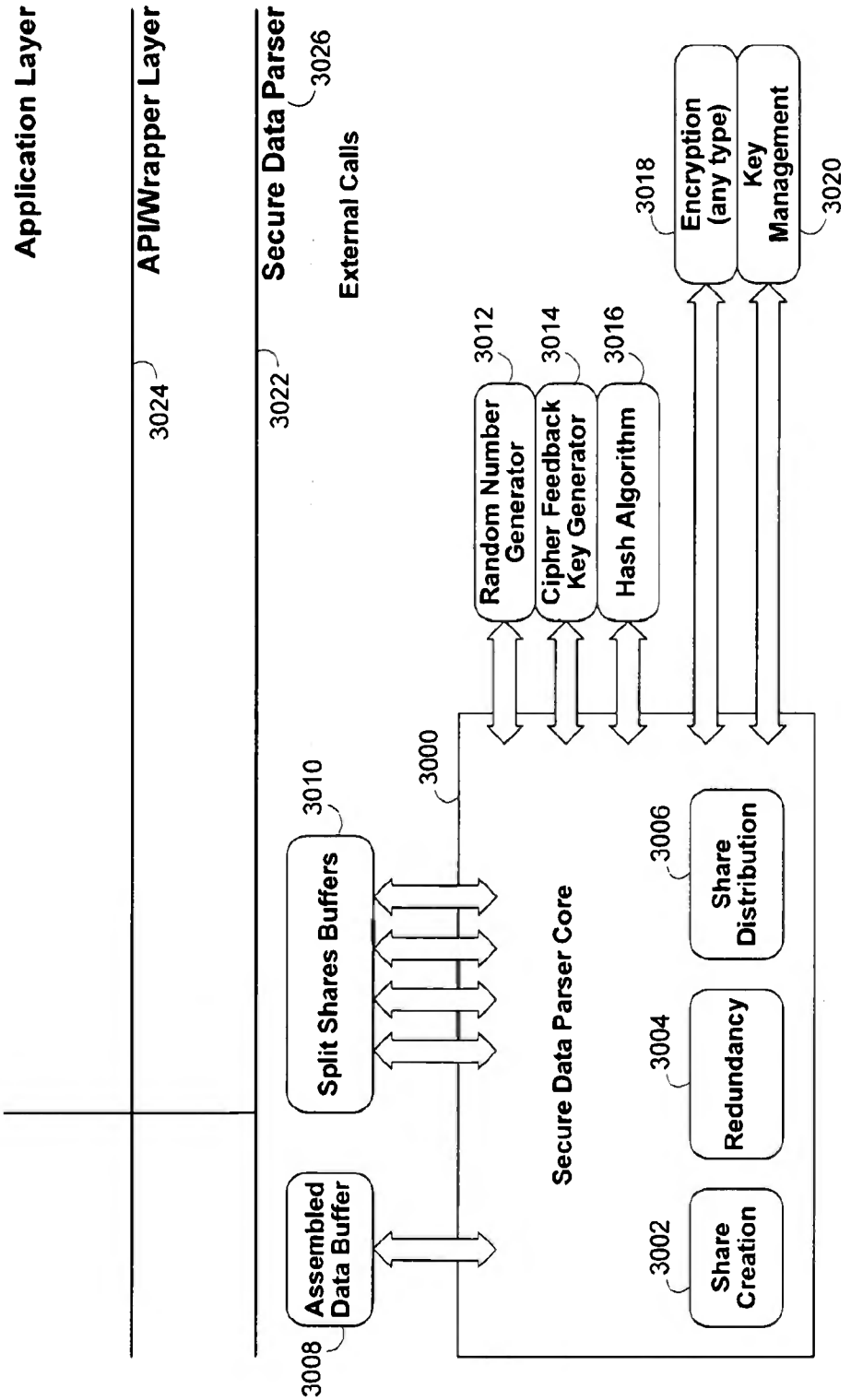

Application Layer
API/Wrapper Layer
Secure Data Parser
3026
External Calls
3024
3022
3012
3014
3016
3018
3020
Random Number Generator
Cipher Feedback Key Generator
Hash Algorithm
Encryption (any type)
Key Management
3000
3010
Split Shares Buffers
Secure Data Parser Core
3006
Share Distribution
3004
Redundancy
3002
Share Creation
Assembled Data Buffer
3008

FIG. 30

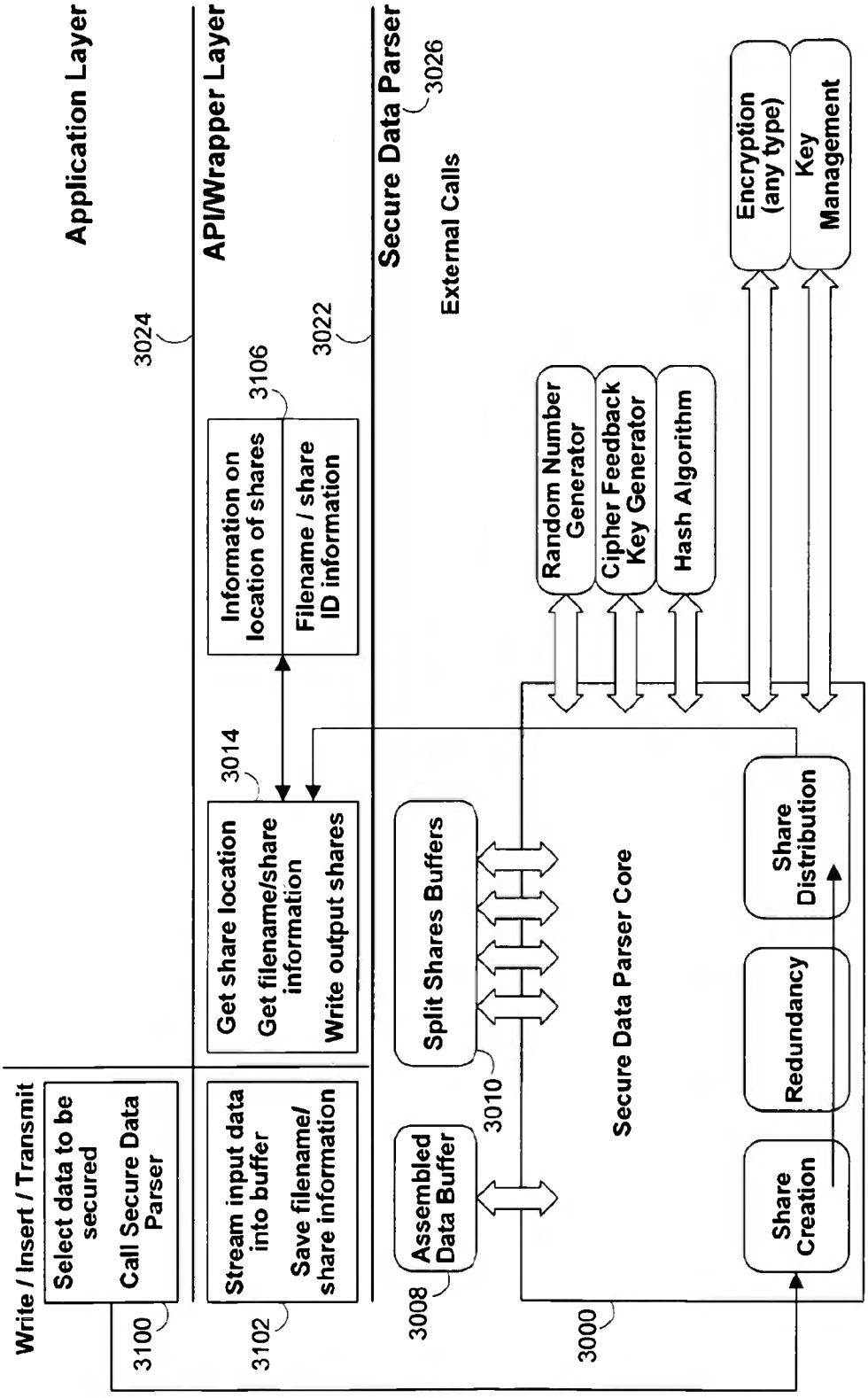

Application Layer
API/Wrapper Layer
Secure Data Parser
3026
External Calls
3024
3022
3106
Information on location of shares
Filename / share ID information
3014
Get share location
Get filename/share information
Write output shares
Write / Insert / Transmit
Select data to be secured
Call Secure Data Parser
3100
Stream input data into buffer
Save filename/ share information
3102
Assembled Data Buffer
3008
Split Shares Buffers
3010
Secure Data Parser Core
3000
Share Creation
Redundancy
Share Distribution
Random Number Generator
Cipher Feedback Key Generator
Hash Algorithm
Encryption (any type)
Key Management

FIG. 31

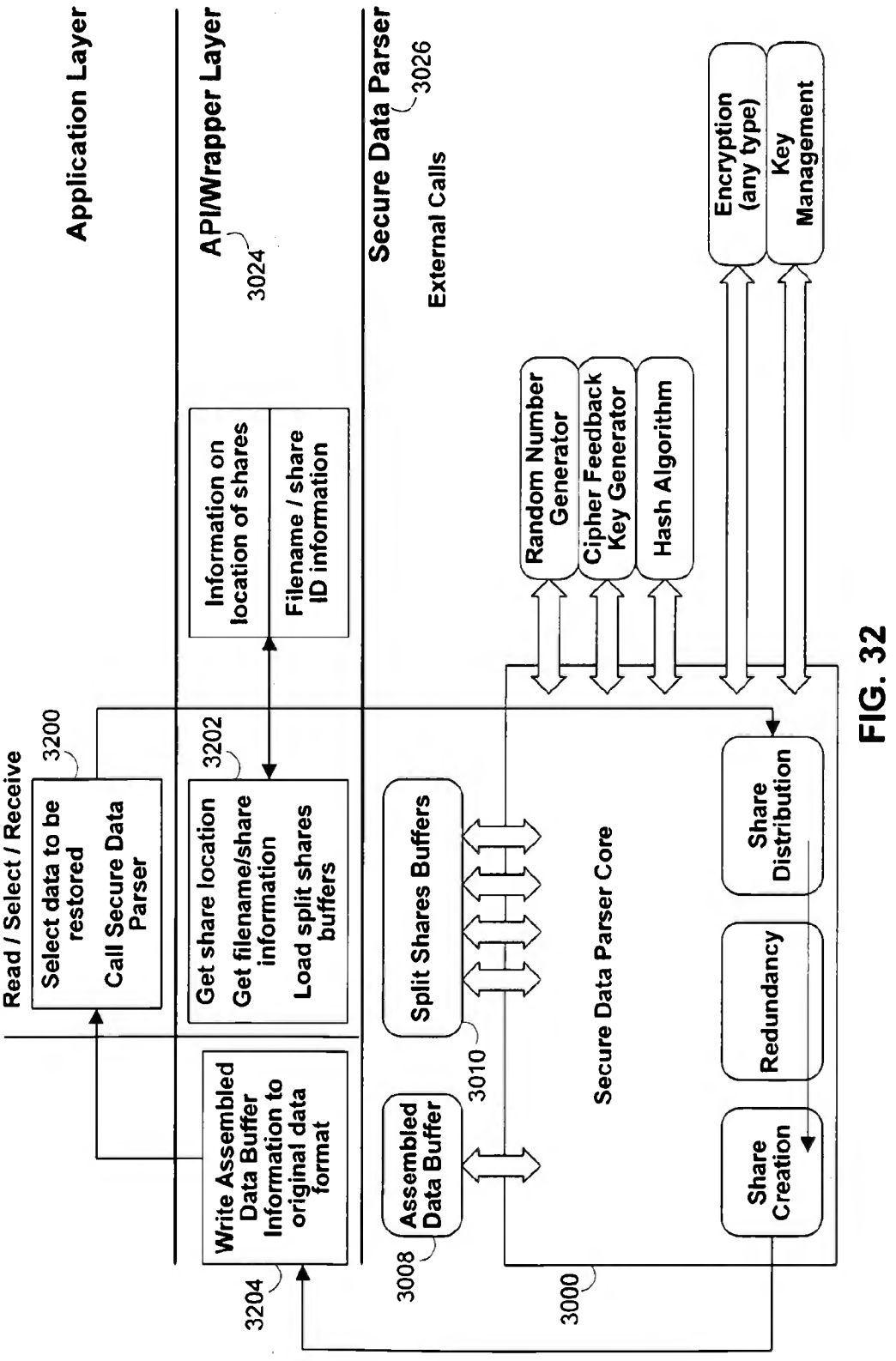

Application Layer
API/Wrapper Layer
3024
Secure Data Parser
3026
External Calls
Read / Select / Receive
3200
Select data to be restored
Call Secure Data Parser
3202
Get share location
Get filename/share information
Load split shares buffers
Information on location of shares
Filename / share ID information
Write Assembled Data Buffer Information to original data format
3204
Split Shares Buffers
3010
Assembled Data Buffer
3008
Secure Data Parser Core
3000
Random Number Generator
Cipher Feedback Key Generator
Hash Algorithm
Encryption (any type)
Key Management
Share Distribution
Redundancy
Share Creation

FIG. 32

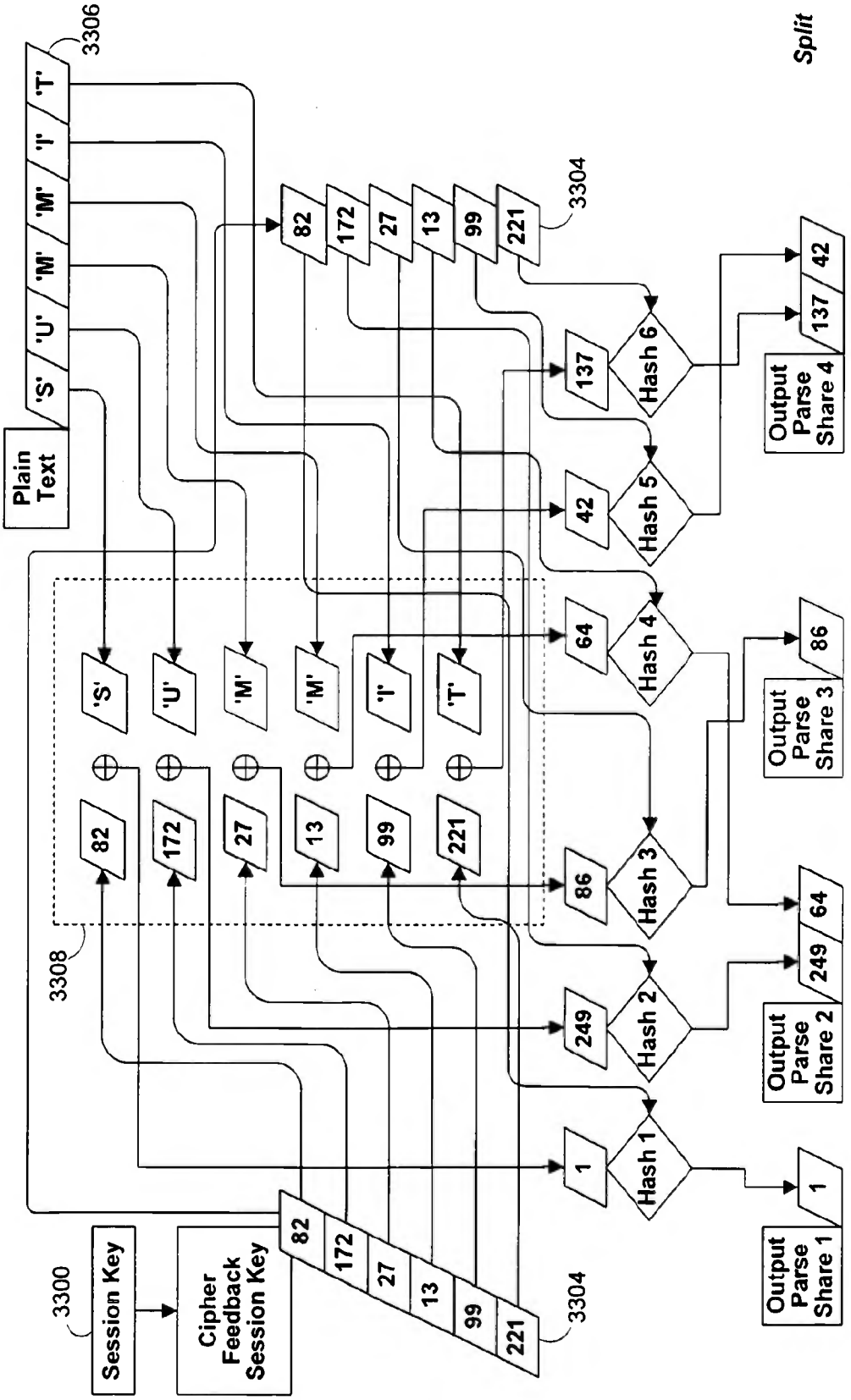

Split
3306
Plain Text
'S' 'U' 'M' 'M' 'I' 'T'
3304
82 172 27 13 99 221
3308
Session Key
3300
Cipher Feedback Session Key
Hash 1
Hash 2
Hash 3
Hash 4
Hash 5
Hash 6
Output Parse Share 1
Output Parse Share 2
Output Parse Share 3
Output Parse Share 4

FIG. 33

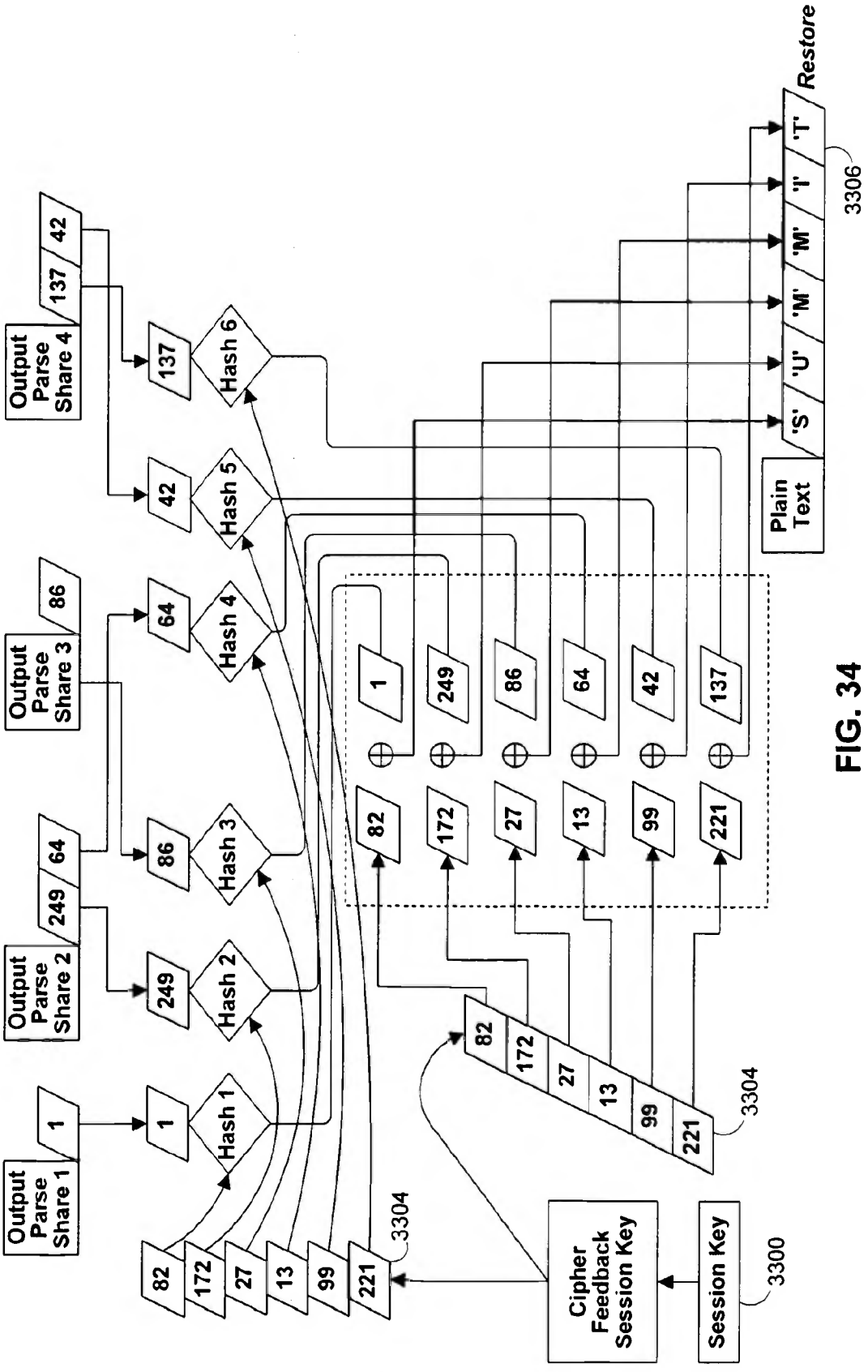

Output Parse Share 1
Output Parse Share 2
Output Parse Share 3
Output Parse Share 4
Hash 1
Hash 2
Hash 3
Hash 4
Hash 5
Hash 6
Plain Text
Restore
'S'
'U'
'M'
'M'
'I'
'T'
3306
3304
3300
Cipher Feedback Session Key
Session Key

FIG. 34

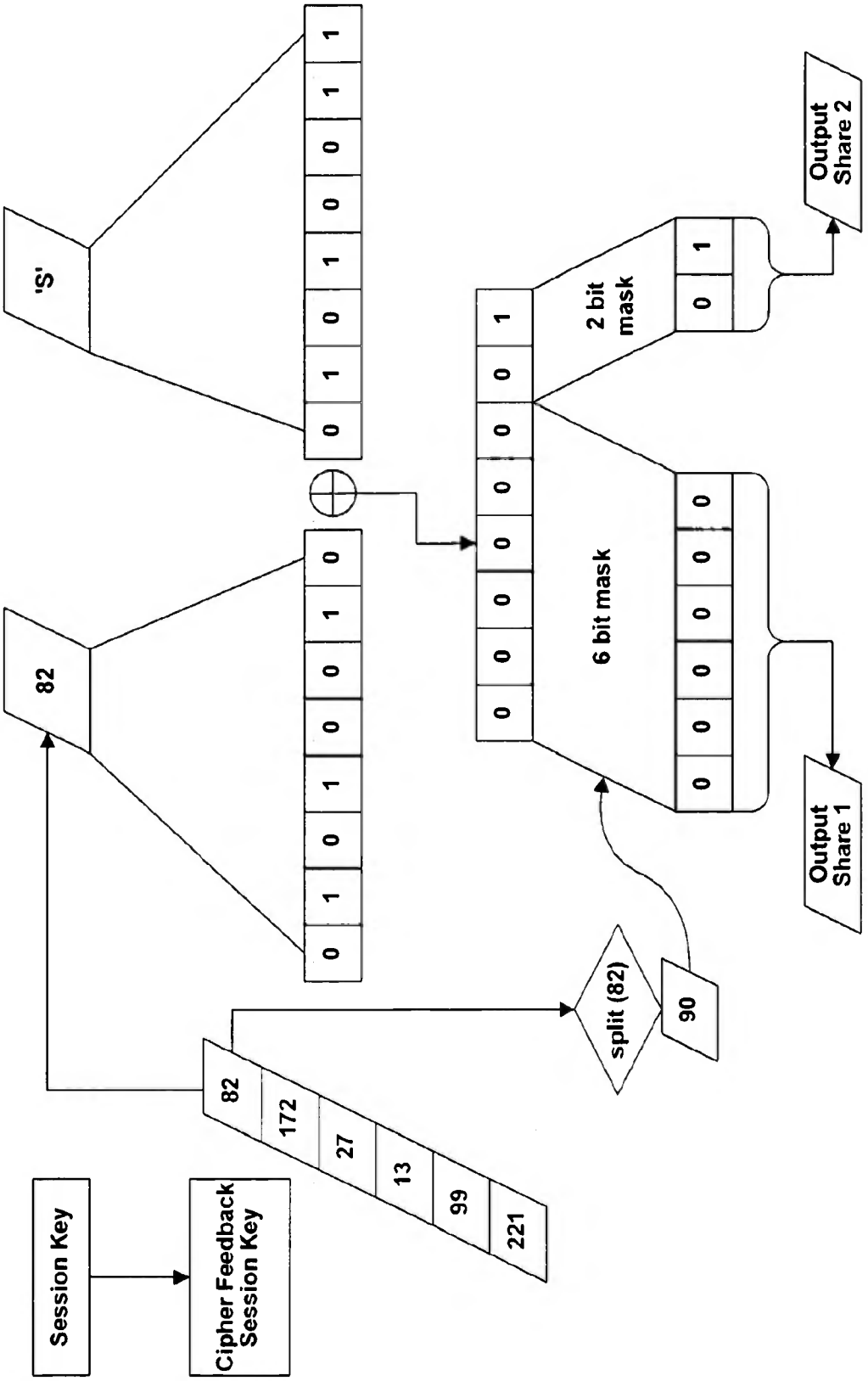

Session Key
Cipher Feedback Session Key
82
172
27
13
99
221
split (82)
90
'S'
82
6 bit mask
2 bit mask
Output Share 1
Output Share 2

FIG. 35

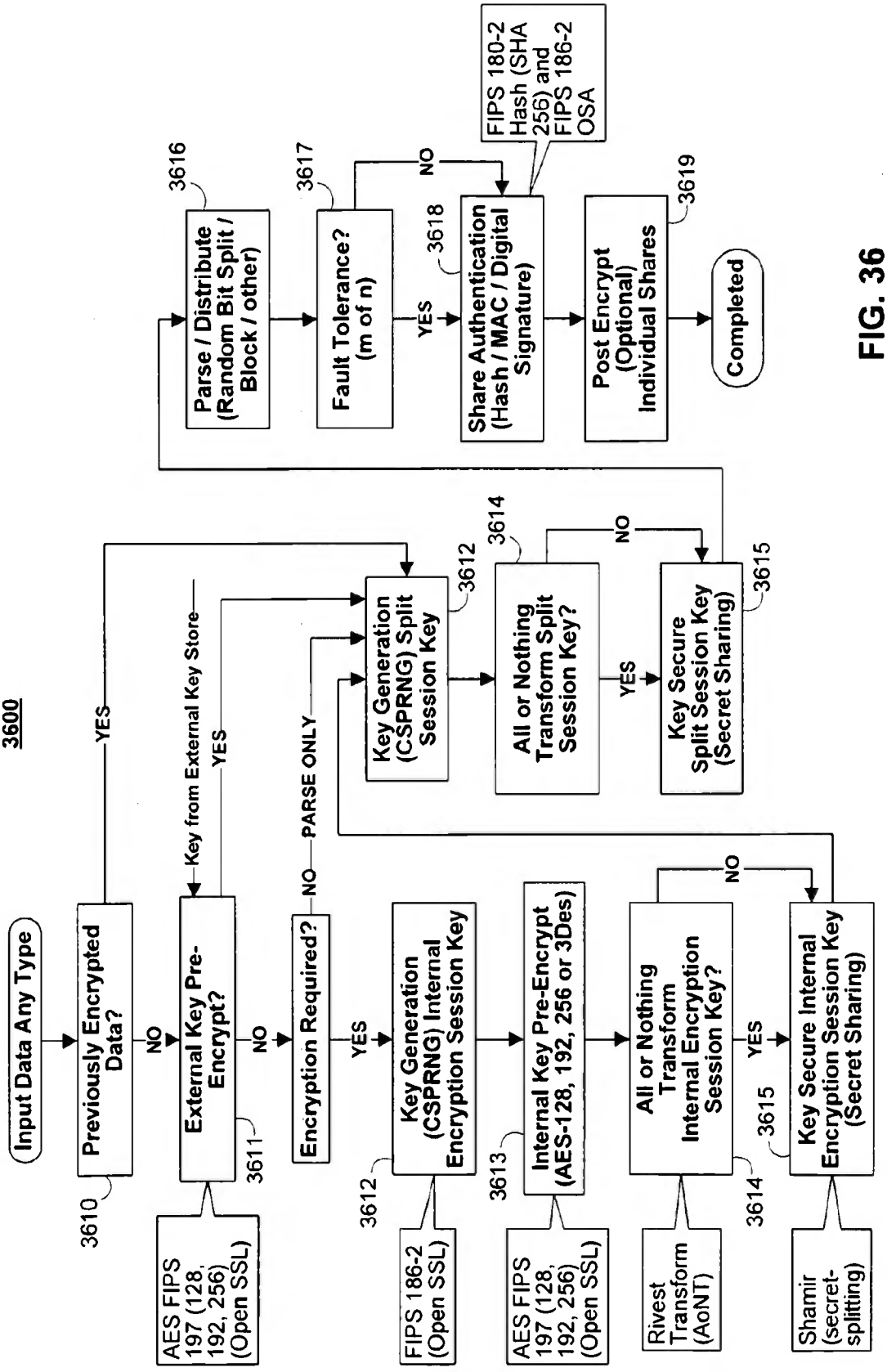

3600
Input Data Any Type
Previously Encrypted Data?
YES
NO
3610
External Key Pre-Encrypt?
Key from External Key Store
YES
NO
3611
AES FIPS 197 (128, 192, 256) (Open SSL)
Encryption Required?
PARSE ONLY
NO
YES
Key Generation (CSPRNG) Split Session Key
3612
FIPS 186-2 (Open SSL)
All or Nothing Transform Split Session Key?
3614
YES
NO
Key Secure Split Session Key (Secret Sharing)
3615
Key Generation (CSPRNG) Internal Encryption Session Key
3612
Internal Key Pre-Encrypt (AES-128, 192, 256 or 3Des)
3613
AES FIPS 197 (128, 192, 256) (Open SSL)
All or Nothing Transform Internal Encryption Session Key?
3614
NO
YES
3615
Rivest Transform (AoNT)
Key Secure Internal Encryption Session Key (Secret Sharing)
Shamir (secret-splitting)
Parse / Distribute (Random Bit Split / Block / other)
3616
Fault Tolerance? (m of n)
3617
YES
NO
3618
FIPS 180-2 Hash (SHA 256) and FIPS 186-2 OSA
Share Authentication (Hash / MAC / Digital Signature)
Post Encrypt (Optional) Individual Shares
3619
Completed

FIG. 36

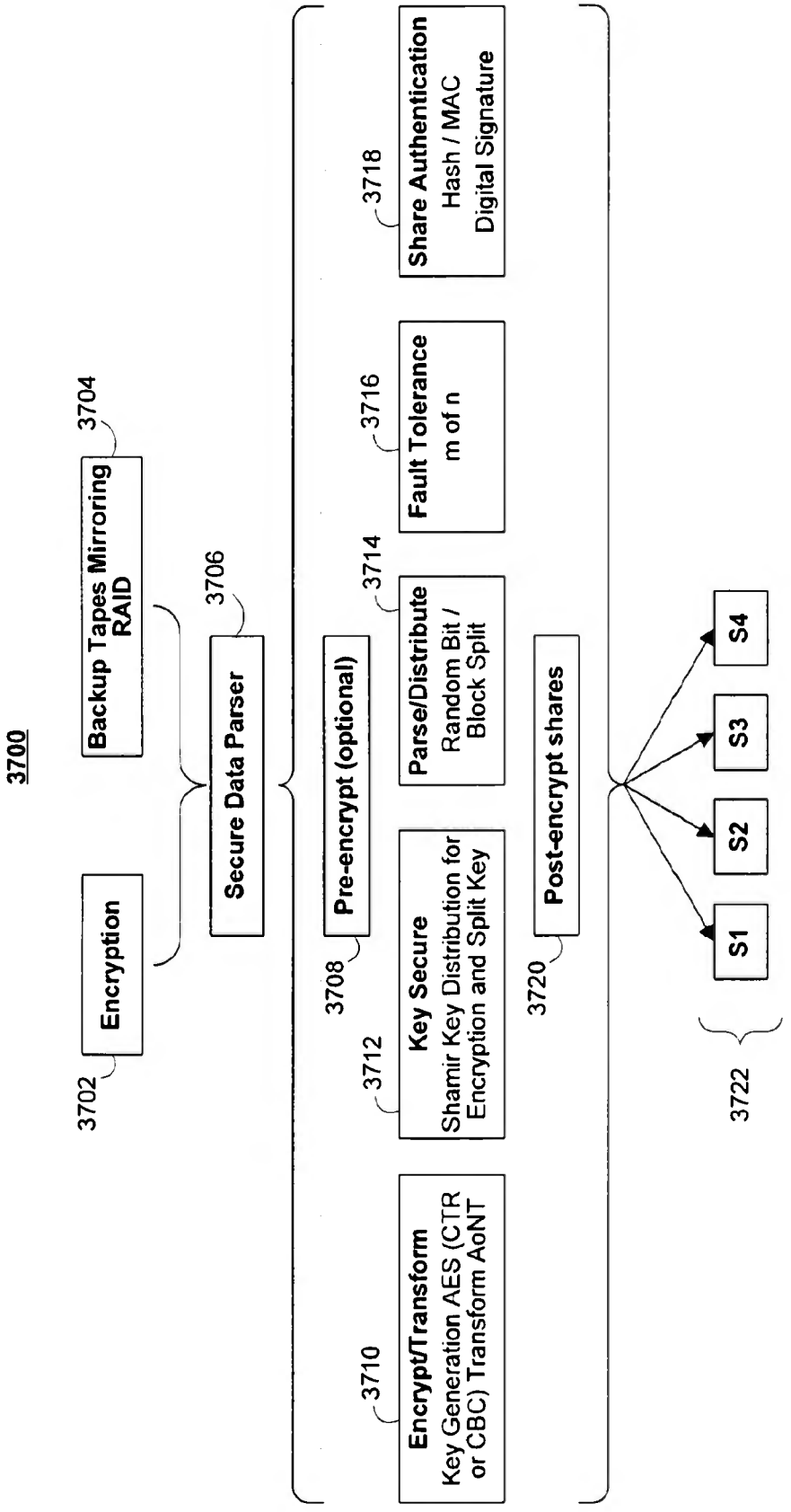

3700
Encryption
3702
Backup Tapes Mirroring RAID
3704
Secure Data Parser
3706
Pre-encrypt (optional)
3708
Key Secure
Shamir Key Distribution for Encryption and Split Key
3712
Encrypt/Transform
Key Generation AES (CTR or CBC) Transform AoNT
3710
Parse/Distribute
Random Bit / Block Split
3714
Fault Tolerance
m of n
3716
Share Authentication
Hash / MAC
Digital Signature
3718
Post-encrypt shares
3720
S1
S2
S3
S4
3722

FIG. 37

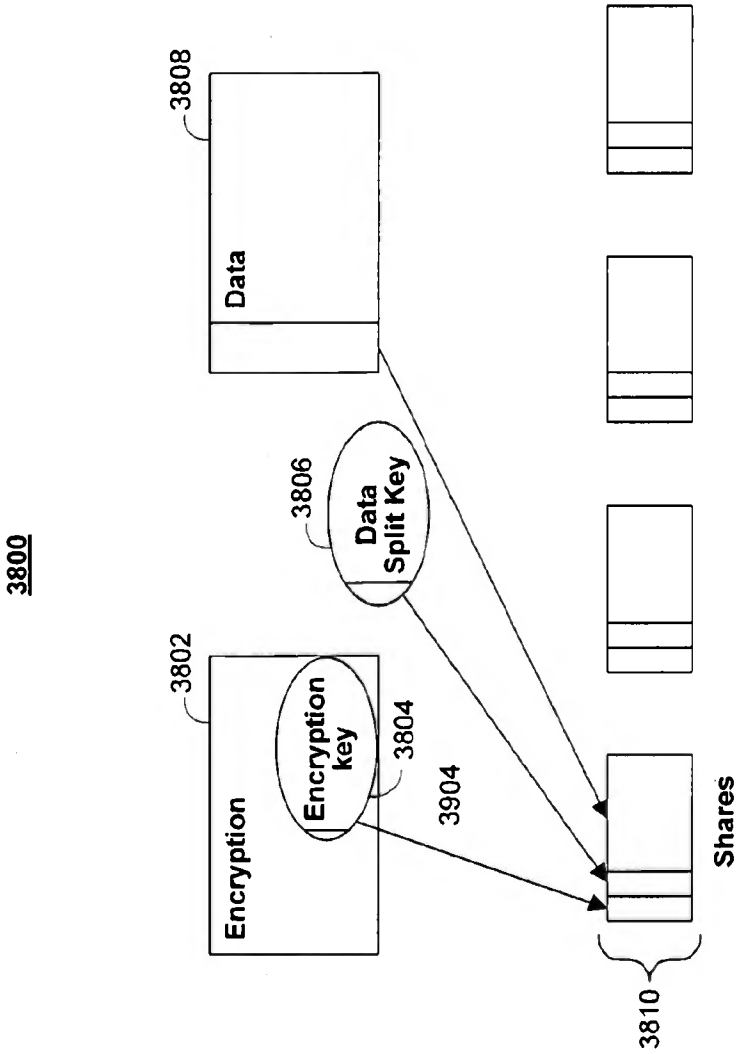

3800
3808
Data
3806
Data Split Key
3802
Encryption
Encryption key
3804
3904
Shares
3810

FIG. 38

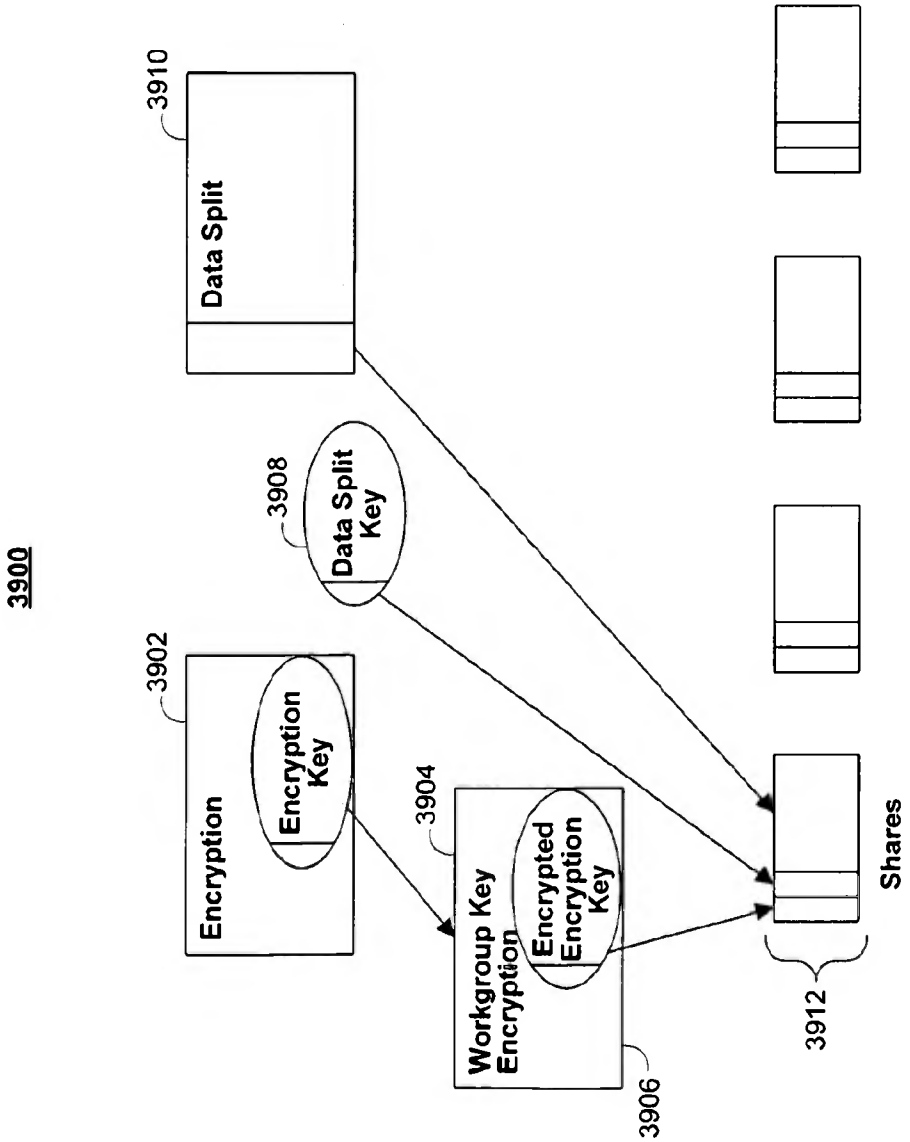

3900
3910
Data Split
3908
Data Split Key
3902
Encryption
Encryption Key
3904
Workgroup Key Encryption
Encrypted Encryption Key
3906
Shares
3912

FIG. 39

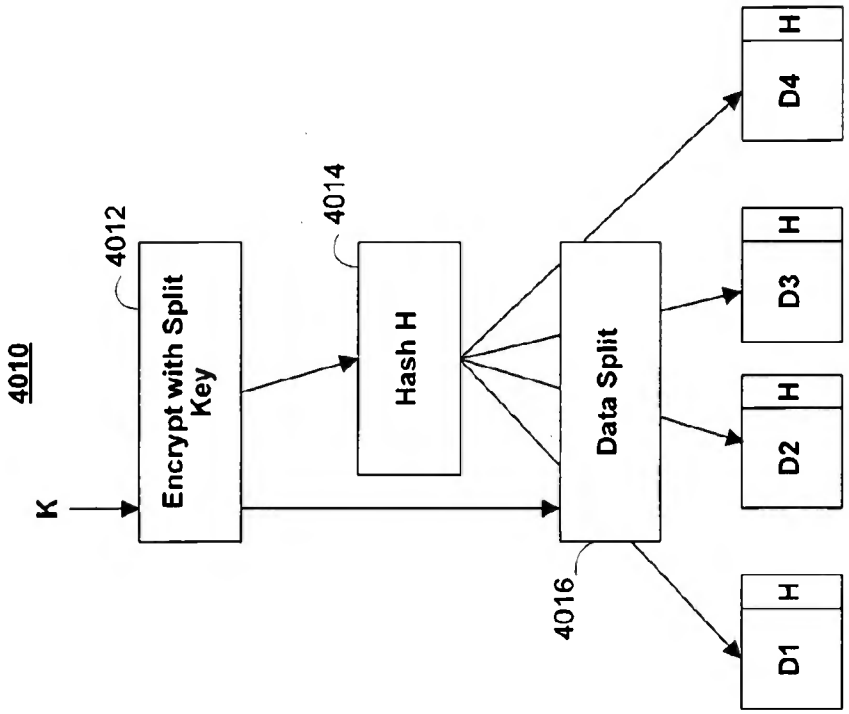
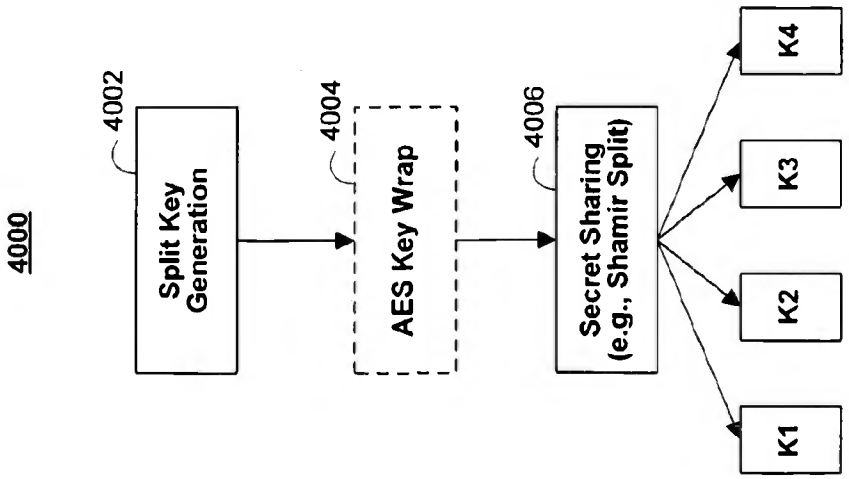

4010
K
4012
Encrypt with Split Key
4014
Hash H
Data Split
4016
D1
H
D2
H
D3
H
D4
H

FIG. 40A


4000
4002
Split Key Generation
4004
AES Key Wrap
4006
Secret Sharing (e.g., Shamir Split)
K1
K2
K3
K4

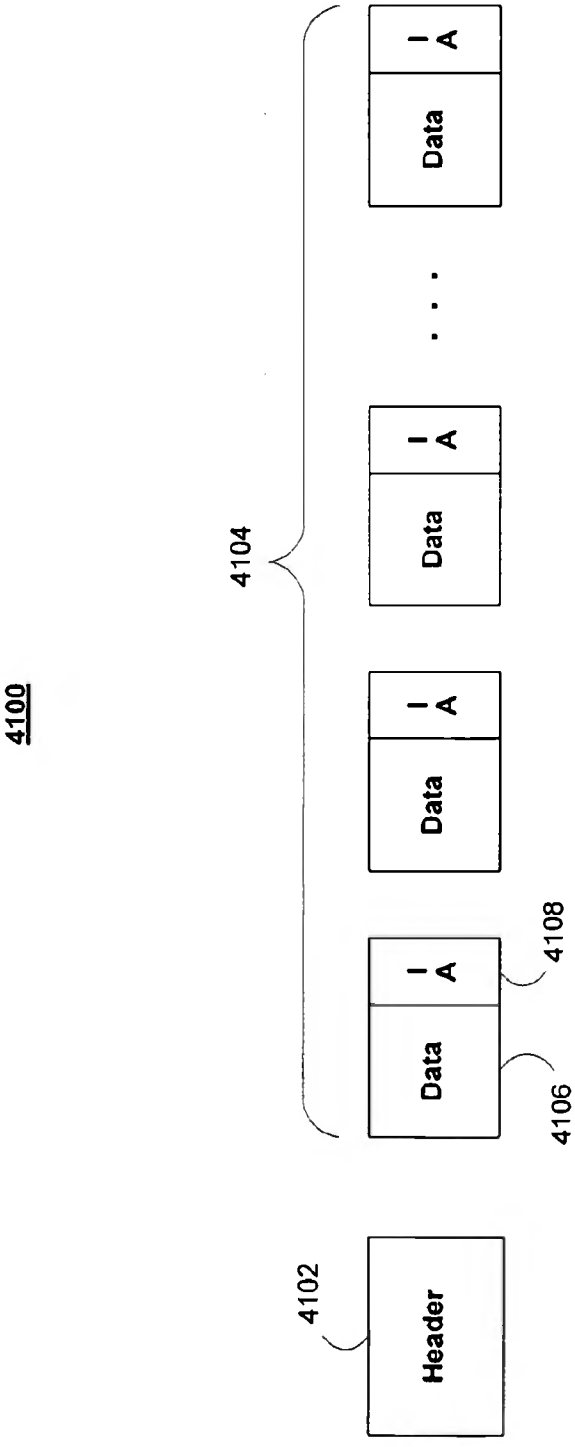


FIG. 41

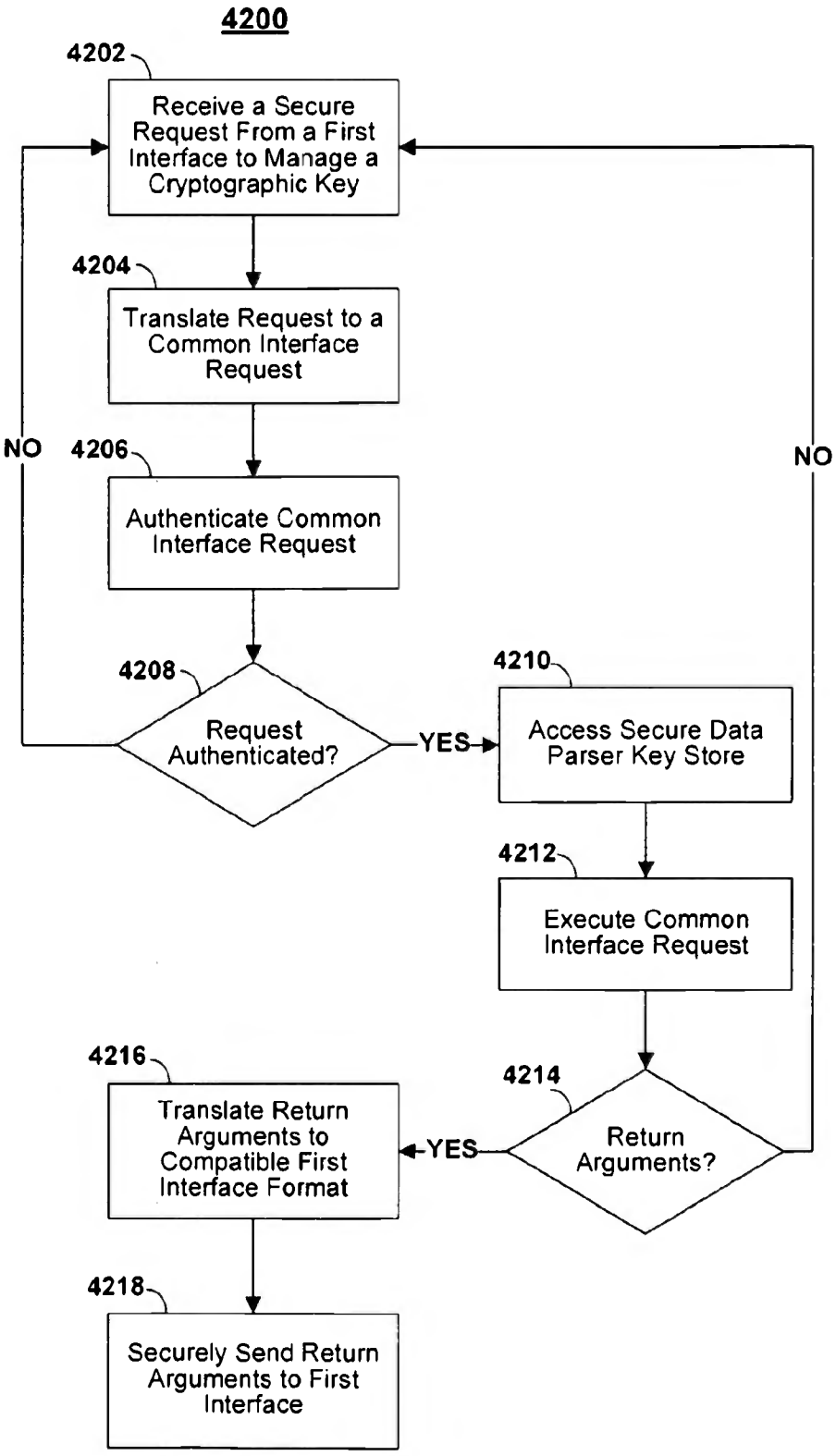

4200
4202
Receive a Secure Request From a First Interface to Manage a Cryptographic Key
4204
Translate Request to a Common Interface Request
NO
4206
Authenticate Common Interface Request
NO
4208
Request Authenticated?
YES
4210
Access Secure Data Parser Key Store
4212
Execute Common Interface Request
4214
Return Arguments?
YES
4216
Translate Return Arguments to Compatible First Interface Format
4218
Securely Send Return Arguments to First Interface

Figure 42

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 08/10677

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - H04L 9/00 (2008.04)
USPC - 713/171, 726/5
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
USPC: 713/171, 726/5

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC: 713/150, 168, 171; 726/2, 3, 4, 5, 26; 705/50, 76, 72, 64

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
PubWEST(USPT,PGPB,EPAB,JPAB); DialogPRO; WIPO, EPO, CITESEER, Google patents, Google scholar Search Terms Used: cryptographic key, first interface format, first format, common interface format, common format, second interface format, second format, format, cryptographic key management, managing cryptographic Key

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2003/0058274 A1 (HILL et al.) 27 March 2003 (27.03.2003) entire document especially abstract, para [0002], para [0004]-[0008], para [0016]-[0020], para [0025], para [0067]-[0069].	1-15
Y	US 2007/0160198 A1 (ORSINI et al.) 12 July 2007 (12.07.2007) entire document especially abstract, para [0151], para [0201], para [0424]-[0425], para [0501], para [0521].	1-15
A	US 2002/0091640 A1 (GUPTA) 11 July 2002 (11.07.2002) entire document	1-15

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 November 2008 (11.11.2008)	21 NOV 2008

Name and mailing address of the ISA/US	Authorized officer:
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (April 2007)

PATENTS
SFC-2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION

Applicants	:	Rick L. Orsini et al.
Application No.	:	11/258,839 Confirmation No. 2420
Filed	:	October 25, 2005
For	:	SECURE DATA PARSER METHOD AND SYSTEM
Art Unit	:	2432
Examiner	:	Samson B. Lemma

New York, NY 10036
June 22, 2009

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

SECOND SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

Sir:

Pursuant to 37 C.F.R. §§ 1.56 and 1.97, applicants hereby make the documents listed on the accompanying Form PTO/SB/08A of record in the above-identified patent application. Pursuant to 37 C.F.R. § 1.98(a)(2)(ii) no copies of the "U.S. Patent Documents" and "U.S. Patent Publications" are provided. Copies of the “Foreign” and "Non-Patent Literature” documents are submitted herewith. Applicants respectfully request that these documents be printed on any patent which may issue on this application. Applicants request that a copy of Form PTO/SB/08A, as considered and

initialed by the Examiner, be returned with the next communication.

Pursuant to 37 C.F.R. §1.97(b)(3), this Supplemental Information Disclosure Statement is filed before the mailing of a first Office action on the merits. Accordingly, no fee is believed to be due in connection with this submission. However, if for any reason a fee is due, the Director is hereby authorized to charge payment of any fee required in connection with this Statement to Deposit Account No. 06-1075. Consideration of the foregoing in relation to this patent application is respectfully requested.

Applicants reserve the right to establish the patentability of the claimed invention over any of the information provided herewith, and/or to prove that this information may not be prior art, and/or to prove that this information may not be enabling for the teachings purportedly offered.

Respectfully submitted,

/Brian E. Mack/

Brian E. Mack
Registration No. 57,189
Attorney for Applicants
Customer No. 1473
(212) 596-9000

PTO/SB/08a (01-08)
Approved for use through 07/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet	1	of	7

Complete if Known	
Application Number	11/258,839 (Conf. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	~~2132~~ 2432
Examiner Name	~~Gilberto Barron, Jr.~~ Samson Lemma
Attorney Docket Number	SFC-2

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
/S.L./		4,453,074	06-05-1984	Weinstein	
		4,924,513	05-08-1990	Herbison et al.	
		4,932,057	06-05-1990	Kolbert	
		5,010,572	04-23-1991	Bathrick et al.	
		5,016,274	05-14-1991	Micali et al.	
		5,051,745	09-24-1991	Katz	
		5,268,963	12-07-1993	Monroe et al.	
		5,375,244	12-20-1994	McNair	
		5,386,104	01-31-1995	Sime	
		5,485,474	01-16-1996	Rabin	
		5,524,073	06-04-1996	Stambler	
		5,615,269	03-25-1997	Micali	
		5,642,508	06-24-1997	Miyazawa	
		5,666,414	09-09-1997	Micali	
		5,666,416	09-09-1997	Micali	
		5,682,425	10-28-1997	Enari	
		5,703,907	12-30-1997	James	
		5,717,758	02-10-1998	Micali	
		5,748,735	05-05-1998	Ganesan	
		5,761,306	06-02-1998	Lewis	
		5,768,382	06-16-1998	Schneier et al.	
		5,768,519	06-16-1998	Swift et al.	
		5,790,677	08-04-1998	Fox et al.	
		5,823,948	10-20-1998	Ross et al.	
		5,903,882	05-11-1999	Asay et al.	
		5,903,652	05-11-1999	Mital	
		5,910,987	06-08-1999	Ginter et al.	
/S.L./		5,940,507	08-17-1999	Cane et al.	

Examiner Signature	/Samson Lemma/	Date Considered	12/20/2008

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08a (01-08)
Approved for use through 07/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 2 of 7

Complete if Known	
Application Number	11/258,839 (Conf. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2132
Examiner Name	Gilberto Barron, Jr.
Attorney Docket Number	SFC-2

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
/S.L./		5,960,083	09-28-1999	Micali	
		5,966,448	10-12-1999	Namba et al.	
		5,966,444	10-12-1999	Yuan et al.	
		5,991,414	11-23-1999	Garay et al.	
		6,009,177	12-28-1999	Sudia	
		6,023,508	02-08-2000	Bombard et al.	
		6,026,163	02-15-2000	Micali	
		6,073,237	06-06-2000	Ellison	
		6,092,201	07-18-2000	Turnbull et al.	
		6,094,485	07-25-2000	Weinstein et al.	
		6,134,550	10-17-2000	Van Oorschot et al.	
		6,229,894	05-08-2001	Van Oorschot et al.	
		6,240,187	05-29-2001	Lewis	
		6,240,183 B1	05-29-2001	Marchant	
		6,289,509	09-11-2001	Kryloff	
		6,301,659 B1	10-09-2001	Micali	
		6,314,409	11-06-2001	Schneck et al.	
		6,324,650	11-27-2001	Ogilvie	
		6,336,186	01-01-2002	Dyksterhouse et al.	
		6,345,314	02-05-2002	Cole et al.	
		6,345,101	02-05-2002	Shukla	
		6,356,941	03-12-2002	Cohen	
		6,363,425	03-26-2002	Hook et al.	
		6,386,451	05-14-2002	Sehr	
		6,424,718	07-23-2002	Holloway	
		6,438,690	08-20-2002	Petal et al.	
		6,449,730	09-10-2002	Mann et al.	
/S.L./		6,483,921 B1	11-19-2002	Harkins	

Examiner Signature	/Samson Lemma/	Date Considered	12/20/2008

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08a (01-08)
Approved for use through 07/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 3 of 7

Complete if Known	
Application Number	11/258,839 (Conf. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2132
Examiner Name	Gilberto Barron, Jr.
Attorney Docket Number	SFC-2

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		6,553,493	04-22-2003	Okumura et al.	
		6,615,347	09-02-2003	de Silva et al.	
		6,625,734	09-23-2003	Marvit et al.	
		6,978,367	12-20-2005	Hind et al.	

U.S. PATENT PUBLICATIONS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
/S.L./		2001/0001876 A1	05-24-2001	Morgan et al.	
		2001/0051902 A1	12-13-2001	Messner	
		2002/0010679 A1	01-24-2002	Felsher	
		2002/0032663 A1	03-14-2002	Messner	
		2002/0046359 A1	04-18-2002	Boden	
		2002/0071566 A1	06-13-2002	Kurn	
		2002/0129235 A1	09-12-2002	Okamoto et al.	
		2002/0162047	10-31-2002	Peters et al.	
		2003/0051054 A1	03-13-2003	Redlich et al.	
		2003/0070077 A1	04-10-2003	Redlich et al.	
		2003/0167408	09-04-2003	Fitzpatrick	
/S.L./		2004/0122863 A1	06-24-2004	Sidman	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3] "Number[4] "Kind Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]
/S.L./		EP 346180 B1	12-13-1989	France Telecom		√
/S.L./		EP 354774 B1	02-14-1990	International Business Machines Corporation		
/S.L./		GB 2237670	05-08-1991	Midland Bank PLC		

Examiner Signature	/Samson Lemma/	Date Considered	12/20/2008

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08a (01-08)
Approved for use through 07/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 4 of 7

Complete if Known	
Application Number	11/258,839 (Conf. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2132
Examiner Name	Gilberto Barron, Jr.
Attorney Docket Number	SFC-2

/S.L./		EP 0485090	05-13-1992	Visa International Service Association		
/S.L./		JP 04297157	10-21-1992	Mitsubishi Electric Corporation		√
/S.L./		EP 636259 B1	02-01-1995	Secure Computing Corporation		
/S.L./		EP 793367 A2	09-03-1997	Oki Electric Industry Co., Ltd.		
/S.L./		EP 0821504 A2	01-28-1998	Nippon Telegraph and Telephone Corporation		
/S.L./		EP 0862301 A2	09-02-1998	Fujitsu Limited		
/S.L./		RU 2124814 C1	01-10-1999	A.A. Moldovjan		√
/S.L./		EP 1011222 A1	06-21-2000	Fujitsu Limited		
/S.L./		EP 1239384	09-11-2002	PKWARE Inc.		

FOREIGN PATENT PUBLICATIONS

/S.L./		WO 98/47091	10-22-1998	Transactor Networks, Inc.		
		WO 99/19845	04-24-1999	David P. Maher		
		WO 99/46720	09-16-1999	Cha! Technologies Services, Inc.		
		WO 99/65207	12-16-1999	Goertzel et al.		
		WO 00/79367	12-28-2000	The Brodia Group		
		WO 01/22650	03-29-2001	Ethentica, Inc.		
		WO 01/22322	03-29-2001	Ethentica, Inc.		
		WO 01/22319	03-29-2001	Ethentica Inc.		
		WO 01/22201	03-29-2001	Ethentica Inc.		
		WO 01/22651	03-29-2001	Ethentica, Inc.		
		WO 02/21761	03-14-2002	Widevine Technologies, Inc.		
/S.L./		WO 02/21283	03-14-2002	Sanctuary Systems		

Examiner Signature	/Samson Lemma/	Date Considered	12/20/2008

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08a (01-08)
Approved for use through 07/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 5 of 7

Complete if Known	
Application Number	11/258,839 (Conf. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2132
Examiner Name	Gilberto Barron, Jr.
Attorney Docket Number	SFC-2

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
/S.L./		Myers et al "A secure, publisher-centric Web caching infrastructure" In: INFOCOM 2001 Proceedings. IEEE Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies [online], vol. 3 p. 1235-1243. Published 04/22/2001. [retrieved on 7-8-2008]. Retrieved from the interenet <URL: http://people.ischool.berkeley.edu/-chuang/pubs/gemini.pdf>	
		RSA SureFile: Software Powered by PKZIP...**BSSF DS 0103** Authorized Reseller: Technical Specifications Platforms Microsoft ® Windows ® 98 Second Edition ME NT 4.0 Workstation SP6A 2000 Professional SP2... www.rsasecurity.com/products/bsafe/datasheets/BSSF_DS_103.pdf	
		Nightingale: The New Secret-Splitting Technology from RSA... **NGBK DS 0403** http://developer.rsasecurity.com/labs/nightingale/developer.rsasecurity.com/labs/nightingale/files/nightingale-brochure.pdf	
		Waldman, et al., "Publius: A robust, tamper-evident, censorship-resistant web publishing sytem," Proceedings of the 9th USENIX Security Symposum, August 2000	
		Hugo Krawczyk, "Distributed Fingerprints and Secure Information Dispersal," 12th ACM, Symposium on Principles on Distributed Computing, Ithaca, NY, ACM 0-89191-613-1/93/0008/0207, 1993, pages 207-218	
		Michael O. Rabin, "Efficient Dispersal of Information for Security, Load Balancing and Fault Tolerance," Journal of the Association for Computing Machinery, Vol. 36, No. 2, pp. 335-348, April 1989	
		Adi Shamir, "How to Share a Secret", Communications of the ACM, Vol. 22, No. 11, pp. 612613, November 1979	
		Garay, et al., "Secure distributed storage and retrieval," Theoretical Comput. Sci., 243(1--2):363—389, July 2000	
		B. Hunter, "Simplfying PKI Usage Through a Client-Server Architecture and Dynamic Propagation of Certificate Paths and Repository Addresses", Proceedings 13th International Workshop on Database and Expert Systems Applications (IEEE Computer Soc., Los Alamitos, CA), 2-6 September 2002, pg. 505-510.	
/S.L./		Kin-Ching Chan and Chan, S.-H. G., "Distributed Servers Approach for Large-Scale Multicase", IEEE Journal on Selected Areas in Communications (IEEE, Piscataway, NJ) October 2002, 20(8):1500-1510.	

Examiner Signature	/Samson Lemma/	Date Considered	12/20/2008

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08a (01-08)
Approved for use through 07/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 6 of 7

Complete if Known	
Application Number	11/258,839 (Conf. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2132
Examiner Name	Gilberto Barron, Jr.
Attorney Docket Number	SFC-2

/S.L./		Kin-Ching Chan and Chan, S.-H. G., "Distributed Server Networks for Secure Multicast", GLOBCOM'01: IEEE Global Telecommunications Conference (IEEE, Piscataway, NJ) 3:1974-1978 (2001).	
		S.Y. Shin, Jung-Yeop Kim, R.E. Gantenbein and C.M. Lundquist, "Design a Working Model of Secure Data Transfer Using a Data Mart", Proceedings of the ISCA 14th International Conference Computer Applications in Industry and Engineering (ISCA, Cary, NC) 27-29 November 2001, pg. 66-69.	
		"Lancope Announces Stealthwatch 3.0 for Enhanced Enterprise-Wide Security and Improved Manageability", Business Wire (Newswire Association, Inc.), 14 April 2003.	
		"Decru Unveils Security Appliances for Storage Networks; Decru DataFort (TM) Security Alliances Protect SAN and NAS Environments with Wire-Speed Encryption and Transparent Deployment", PR Newswire (PR Newswire Association. Inc.), 14 October 2002.	
		Lawrence Grant and Fleming B., "Secret Sharing and Splitting", (White Paper) Notre Dame, Indiana, 16 December 2002.	
		Joel McNamara, "Strong Crypto Freeware" (Secret Sharer Version 1.0) July 11, 1995.	
		Dennis Fisher, "RSA Looks to Lock Down Personal Data", eWeek – Enterprise News & Reviews, 14 April 2003.	
		Demir Barlas, "RSA's Security Showcase", Line56.com – The E-Business Executive Daily, 15 April 2003.	
		Marcia Savage, "RSA Unveils Nightingale Technology", CRN.com, 14 April 2003.	
		John K. Waters, "RSA Integrates ID Management; discloses 'Nightingale' ", ADTmag.com, 21 April 2003.	
		Jaikurmar Vijayan, "RSA unveils Management, Encryption Products", Computerworld, 15 April 2003.	
		Eric Doyle, "RSA Splits Data to Stop Hackers", vnunet.com, 16 April 2003.	
		Stan Gibson, "Opinion", eWeek – Enterprise News & Reviews, 14 April 2003.	
		"Trustengine ™ White Paper – Enthentication Services, Secure Storage and Authentication Solutions", Ethentica, Inc. by Security First Corporation, June 2002.	
		"Tactilesense ™ White Paper – A Breakthrough in Fingerprint Authentication", Ethentica, Inc. by Security First Corporation, January, 2003.	
/S.L./		International Search Report – International Application No. PCT/US/06/45066, dated July 17, 2008	

Examiner Signature	/Samson Lemma/	Date Considered	12/20/2008

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08a (01-08)
Approved for use through 07/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet	7	of	7

Complete if Known	
Application Number	11/258,839 (Conf. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2132
Examiner Name	Gilberto Barron, Jr.
Attorney Docket Number	SFC-2

/S.L./		M. Loutrel, et al. "An EAP-BT Smartcard for Authentication in the Next Generation of Wireless Communications", Conference on Network Control and Engineering for QoS Security and Mobility (Kluwer Academic Publishers, Norwell, MA) October 23-25, 2002, pgs. 1-4-114.	

Examiner Signature	/Samson Lemma/	Date Considered	12/20/2008

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08a (01-08)
Approved for use through 04/31/2009. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 2

Complete if Known	
Application Number	11/258,839 (Conf. No. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2432
Examiner Name	Samson B. Lemma
Attorney Docket Number	SFC-2

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
/S.L./		7,069,451 B1	06-27-2006	Ginter, et al.	
/S.L./		7,391,865	06-24-2008	Orsini et al.	

U.S. PATENT PUBLICATIONS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
/S.L./		2002/0091640 A1	07-11-2002	Gupta	
/S.L./		2003/0058274 A1	03-27-2003	Hill et al.	
/S.L./		2006/0282681 A1	12-14-2006	Scheidt et al.	
/S.L./		2007/0160198	07-12-2007	Orsini et al.	
/S.L./		2008/0137857 A1	06-12-2008	Bellare et al.	
/S.L./		2008/0183992 A1	07-31-2008	Martin et al.	
/S.L./		2008/0244277 A1	10-02-2008	Orsini et al.	
/S.L./		2009/0097661 A1	04-16-2009	Orsini et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3] "Number[4] "Kind Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]
/S.L./		WO 2009/035674 A1	03-19-2009	Security First Corporation		

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
/S.L./		Crescenzo, G. et al., "Non-Interactive and Non-Malleable Commitment," Proceedings of the 30th Annual ACM Symposium on Theory of Computing. Dallas, TX, May 23-26, 1998, [Proceedings of the 30th Annual ACM Symposium on Theory of Computing], New York, NY: ACM, US, PAGES 141-150; XP000970902; ISBN: 978-0-89791-962-3.	

Examiner Signature	/Samson Lemma/	Date Considered	01/03/2010

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08a (01-08)
Approved for use through 04/31/2009. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 2 of 2

Complete if Known	
Application Number	11/258,839 (Conf. No. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2432
Examiner Name	Samson B. Lemma
Attorney Docket Number	SFC-2

/S.L./		Christian Cachin, "On-Line secret Sharing," Cryptography and Coding. IMA Conference, Proceedings, XX, XX, December 18, 1995, pages 190-198, XP002137681.	
/S.L./		Mayer, A. et al., "Generalized Secret Sharing and Group-Key Distribution Using Short Keys," Compression and Complexity of Sequences 1997, Proceedings Salerno, Italy, June 11-13, 1997, Los Alamitos, CA, USA, IEEE Comput,. SOC, US, June 11, 1997, pages 30-44, XP010274905, ISBN: 978-0-8186-8132-5.	

Examiner Signature	/Samson Lemma/	Date Considered	01/03/2010

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 2420

SERIAL NUMBER	FILING or 371(c) DATE RULE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
11/258,839	10/25/2005	713	2432	SFC-2

APPLICANTS

Rick L. Orsini, Flower Mound, TX;
Mark S. O'Hare, Coto De Caza, CA;
Roger Davenport, Campbell, TX;
Steven Winick, Roslyn Heights, NY;

**** CONTINUING DATA ***************************

This appln claims benefit of 60/622,146 10/25/2004
and claims benefit of 60/718,185 09/16/2005

**** FOREIGN APPLICATIONS ***************************

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED **** ** SMALL ENTITY **
01/17/2006

Foreign Priority claimed ☐ Yes ☑ No 35 USC 119(a-d) conditions met ☐ Yes ☑ No Verified and Acknowledged /SAMSON B LEMMA/ Examiner's Signature	☐ Met after Allowance Initials	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
		TX	36	85	21

ADDRESS

ROPES & GRAY LLP
PATENT DOCKETING 39/361
1211 AVENUE OF THE AMERICAS
NEW YORK, NY 10036-8704
UNITED STATES

TITLE

Secure data parser method and system

FILING FEE RECEIVED		
4365	FEES: Authority has been given in Paper No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit

BIB (Rev. 05/07)

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11258839	ORSINI ET AL.
	Examiner	**Art Unit**
	Samson B Lemma	2432

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	12/29/2009								
	1	÷								
	2	÷								
	3	÷								
	4	÷								
	5	÷								
	6	÷								
	7	÷								
	8	÷								
	9	÷								
	10	÷								
	11	÷								
	12	÷								
	13	÷								
	14	÷								
	15	÷								
	16	÷								
	17	÷								
	18	÷								
	19	÷								
	20	÷								
	21	÷								
	22	÷								
	23	÷								
	24	÷								
	25	÷								
	26	÷								
	27	÷								
	28	÷								
	29	÷								
	30	÷								
	31	÷								
	32	÷								
	33	÷								
	34	÷								
	35	÷								
	36	÷								

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11258839	ORSINI ET AL.
	Examiner	Art Unit
	Samson B Lemma	2432

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	12/29/2009								
	37	÷								
	38	÷								
	39	÷								
	40	÷								
	41	÷								
	42	÷								
	43	÷								
	44	÷								
	45	÷								
	46	÷								
	47	÷								
	48	÷								
	49	÷								
	50	÷								
	51	÷								
	52	÷								
	53	÷								
	54	÷								
	55	÷								
	56	÷								
	57	÷								
	58	÷								
	59	÷								
	60	÷								
	61	÷								
	62	÷								
	63	÷								
	64	÷								
	65	÷								
	66	÷								
	67	÷								
	68	÷								
	69	÷								
	70	÷								
	71	÷								
	72	÷								

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11258839	ORSINI ET AL.
	Examiner	Art Unit
	Samson B Lemma	2432

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	12/29/2009								
	73	÷								
	74	÷								
	75	÷								
	76	÷								
	77	÷								
	78	÷								
	79	÷								
	80	÷								
	81	÷								
	82	÷								
	83	÷								
	84	÷								
	85	÷								

PTO/SB/08a (01-08)
Approved for use through 04/31/2009. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 1

Complete if Known	
Application Number	11/258,839 (Conf. No. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2432
Examiner Name	Samson B. Lemma
Attorney Docket Number	SFC-2

U.S. PATENT PUBLICATIONS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
/S.L./		2003/0051159 A2	03-13-2003	McGown et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3] "Number[4] "Kind Code[5] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]
/S.L./		WO 2009/035674 A1	03-19-2009	Security First Corporation		
/S.L./		WO 00/76118	12-14-2000	Gen Instrument Corp.		
/S.L./		WO 00/36786	06-22-2000	Philips Electronics		

NON PATENT LITERATURE DOCUMENTS

Examiner Initials	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
/S.L./		Menezes "Handbook of Applied Cryptography" 1997, CRC Press LLC, XP001525007	
/S.L./		Brainard "A New Approach for Authentication with Short Secrets" RSA Laboratories, Bedford, MA (13 pages)	

Examiner Signature	/Samson Lemma/	Date Considered	01/03/2010

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/258,839	10/25/2005	Rick L. Orsini	SFC-2	2420

1473 7590 01/07/2010

ROPES & GRAY LLP
PATENT DOCKETING 39/361
1211 AVENUE OF THE AMERICAS
NEW YORK, NY 10036-8704

EXAMINER	
LEMMA, SAMSON B	

ART UNIT	PAPER NUMBER
2432	

MAIL DATE	DELIVERY MODE
01/07/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	11/258,839	ORSINI ET AL.	
	Examiner	Art Unit	
	Samson B. Lemma	2432	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *1* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 October 2005*.

2a)☐ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-85* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☐ Claim(s) _____ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☒ Claim(s) *1-85* are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in reply to an amendment filed on 09/03/2009. Claims 1-5, 23-24, 36-43 and 45 are canceled. Thus claims 6-22, 25-35, 44, 46-49 are pending.

Election/Restrictions

2. Restriction to one of the following inventions is required under 35 U.S.C. 121:

I. **Claims 1-36, 41-54 and 69-85 are drawn to** a method for securing communication of a data set from a first location to a second location, the method comprising: generating at the first location at least two portions of data from the data set, wherein each of the at least two portions of data respectively contains a substantially random distribution of a respective subset of the data set; communicating from the first location to the second location the at least two portions of data over at least one communications path, wherein the at least two portions of data are communicated separately from each other; and restoring at the second location the data set from at least a subset of the at least two portions of data, classified in **class 380 subclass 33;**

II. Claims 37-40 and 55-66 are drawn to a method for securing a data set, the method comprising: encrypting the data set using an **encryption key; transforming the encryption key using an All or**

Nothing Transform into a transformed encryption key; generating at least two portions of data from the encrypted data set, wherein each of the at least two portions of data respectively contains a substantially random distribution of a respective subset of the encrypted data set; **generating transformed encryption key information that is capable of being used to generate the transformed encryption key, wherein the transformed encryption key information is distributed among the** at least two portions of data; storing each of the at least two portions of data separately; and restoring the encrypted data set from at least a subset of the at least two portions of data, **classified in class 380, subclass 268.**

III. Claims 67-68 are drawn to a method for securing a data set, the method comprising: **determining a number of portions of data into which to distribute the data set; sequentially distributing the data set, one bit at a time, into each of the data portions, respectively, in a round-robin manner until all bits of the data set have been distributed; appending redundancy data to each of the portions of data;** storing each of the data portions separately; and restoring the data set from a subset of the data portions **classified in class 380, subclass 286.**

3. The inventions are distinct, each from the other because of the following reasons:

The inventions **in Group I, II and III,** are distinct from each other, since Group I **is mainly directed to a method of securing communication of the data by** transmitting or communicating two portion of the data communicated separately or over different communications links.

However Group II is directed to are drawn to a method for securing a data set by encrypting the data set using an encryption key; transforming the encryption key using an All or Nothing Transform into a transformed encryption key and finally Group III is mainly directed to a method for securing a data set and restoring the data set from a subset of the data portions.

Note: Group I is drawn to the **specific way of** securing communication of a data set from a first location to a second location while the limitation recited in (Group II) specifically contains a specific way securing a data set by encrypting the data set using an encryption key; transforming the encryption key using an All or Nothing Transform into a transformed encryption key; and finally the specific method recited in Group III focus on a method for securing a data set and restoring the data set from a subset of the data portions. The specific sequence of steps used in Group I, II and III is therefore distinct from one another.

4. Because these inventions are distinct for the reasons given above and have acquired a separate status in the art as shown by their different

classification, restriction for examination purposes as indicated is proper.

5. Applicants are advised that the replay to this requirement to be completed must include an election of the invention to be examined even though the requirement be traversed (37 CFR 1.143).

6. Applicants is reminded that upon the cancellation of the claims to a non-elected invention, the inventorship must be amended in compliance with 37 CFR 1.48(b) if one or more of the currently named inventors is no longer an inventor of at least one claim remaining in the application. Any amendment of inventorship must be accompanied by a petition under 37 CR 1.48(b) and by the fee required under 37 CFR 1.17(i).

CONTACT INFORMATION

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Samson B Lemma whose telephone number is 571-272-3806. The examiner can normally be reached on Monday-Friday (8:00 am---4: 30 pm).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, BARRON JR GILBERTO can be reached on 571-272-3799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov.

Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Samson B Lemma/
Examiner, Art Unit 2432

/Jung Kim/
Primary Examiner, AU 2432

Electronic Acknowledgement Receipt

EFS ID:	6966419
Application Number:	11258839
International Application Number:	
Confirmation Number:	2420
Title of Invention:	Secure data parser method and system
First Named Inventor/Applicant Name:	Rick L. Orsini
Customer Number:	01473
Filer:	Michael Jeremy Chasan
Filer Authorized By:	
Attorney Docket Number:	SFC-2
Receipt Date:	08-FEB-2010
Filing Date:	25-OCT-2005
Time Stamp:	16:56:36
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		Response_to_Restriction.pdf	11992 9c6f24f454b51dafb7c045a558aecbeefca080fc	yes	3

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Response to Election / Restriction Filed	1	1
	Applicant Arguments/Remarks Made in an Amendment	2	3

Warnings:	
Information:	
Total Files Size (in bytes):	11992

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number	Filing Date	☐ To be Mailed
11/258,839	10/25/2005	

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY ☒ RATE ($)	SMALL ENTITY ☒ FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT	02/08/2010	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 85	Minus	** 85	= 0	X $26 =	0	OR	X $ =	
	Independent (37 CFR 1.16(h))	* 21	Minus	***21	= 0	X $110 =	0	OR	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE	0	OR	TOTAL ADD'L FEE	

AMENDMENT		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/VANESSA BARBER/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

REMARKS

Summary of Office Action

Claims 1-85 are pending in the above-identified patent application.*

The Examiner has found that there are allegedly three distinct inventions in the above-identified patent application and required restriction to one of those inventions, as follows:

Group I: Claims 1-36, 41-54, and 69-85;
Group II: Claims 37-40 and 55-66; and
Group III: Claims 67 and 68

Applicants' Election

Applicants hereby elect without traverse, for examination in this application, the invention of Group I (claims 1-36, 41-54, and 69-85). Applicants hereby expressly reserve the right to pursue the nonelected inventions in one or more divisional applications.

---

* Page 2 of the Office Action mistakenly states that this Office Action is responsive to an amendment filed on September 3, 2009 and that claims 1-5, 23, 24, 36-43, and 45 have been cancelled. No amendments have been filed in this application and no claims have been cancelled. Originally filed claims 1-85 are all still pending in this application. The Office Action includes all of the originally filed claims in the restriction requirement, therefore applicants assume that this statement is an error.

An early and favorable action is respectfully requested.

Respectfully submitted,

/Michael J. Chasan/

Michael J. Chasan
Registration No. 54,026
Attorney for Applicants
ROPES & GRAY LLP
Customer No. 1473

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION

Applicants	:	Rick L. Orsini et al.
Application No.	:	11/258,839 Confirmation No. 2420
Filed	:	October 25, 2005
For	:	SECURE DATA PARSER METHOD AND SYSTEM
Art Unit	:	2432
Examiner	:	Samson B. Lemma

New York, NY 10036
February 8, 2010

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

RESPONSE TO RESTRICTION REQUIREMENT

Sir:

Applicants hereby reply to the Office Action dated January 7, 2010, which did not address the merits of the above-identified patent application, but required restriction of the application.

Remarks begin on page 2 of this Reply to Office Action.

PTO/SB/08a (01-08)
Approved for use through 04/31/2009. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 1

Complete if Known	
Application Number	11/258,839 (Conf. No. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2432
Examiner Name	Samson B. Lemma
Attorney Docket Number	SFC-2

U.S. PATENT PUBLICATIONS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		2003/0051159 A2	03-13-2003	McGown et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3] "Number[4] "Kind Code[5] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]
		WO 2009/035674 A1	03-19-2009	Security First Corporation		
		WO 00/76118	12-14-2000	Gen Instrument Corp.		
		WO 00/36786	06-22-2000	Philips Electronics		

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
		Menezes "Handbook of Applied Cryptography" 1997, CRC Press LLC, XP001525007	
		Brainard "A New Approach for Authentication with Short Secrets" RSA Laboratories, Bedford, MA (13 pages)	

Examiner Signature	/Samson Lemma/	Date Considered	04/24/2010

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08a (01-08)
Approved for use through 07/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet	1	of	7

Complete if Known	
Application Number	11/258,839 (Conf. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	~~2432~~ 2432
Examiner Name	~~Gilberto Barron, Jr.~~ Samson Lemma
Attorney Docket Number	SFC-2

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		4,453,074	06-05-1984	Weinstein	
		4,924,513	05-08-1990	Herbison et al.	
		4,932,057	06-05-1990	Kolbert	
		5,010,572	04-23-1991	Bathrick et al.	
		5,016,274	05-14-1991	Micali et al.	
		5,051,745	09-24-1991	Katz	
		5,268,963	12-07-1993	Monroe et al.	
		5,375,244	12-20-1994	McNair	
		5,386,104	01-31-1995	Sime	
		5,485,474	01-16-1996	Rabin	
		5,524,073	06-04-1996	Stambler	
		5,615,269	03-25-1997	Micali	
		5,642,508	06-24-1997	Miyazawa	
		5,666,414	09-09-1997	Micali	
		5,666,416	09-09-1997	Micali	
		5,682,425	10-28-1997	Enari	
		5,703,907	12-30-1997	James	
		5,717,758	02-10-1998	Micali	
		5,748,735	05-05-1998	Ganesan	
		5,761,306	06-02-1998	Lewis	
		5,768,382	06-16-1998	Schneier et al.	
		5,768,519	06-16-1998	Swift et al.	
		5,790,677	08-04-1998	Fox et al.	
		5,823,948	10-20-1998	Ross et al.	
		5,903,882	05-11-1999	Asay et al.	
		5,903,652	05-11-1999	Mital	
		5,910,987	06-08-1999	Ginter et al.	
		5,940,507	08-17-1999	Cane et al.	

Examiner Signature	/Samson Lemma/	Date Considered	04/24/2010

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08a (01-08)
Approved for use through 07/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 2 of 7

Complete if Known	
Application Number	11/258,839 (Conf. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2132
Examiner Name	Gilberto Barron, Jr.
Attorney Docket Number	SFC-2

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		5,960,083	09-28-1999	Micali	
		5,966,448	10-12-1999	Namba et al.	
		5,966,444	10-12-1999	Yuan et al.	
		5,991,414	11-23-1999	Garay et al.	
		6,009,177	12-28-1999	Sudia	
		6,023,508	02-08-2000	Bombard et al.	
		6,026,163	02-15-2000	Micali	
		6,073,237	06-06-2000	Ellison	
		6,092,201	07-18-2000	Turnbull et al.	
		6,094,485	07-25-2000	Weinstein et al.	
		6,134,550	10-17-2000	Van Oorschot et al.	
		6,229,894	05-08-2001	Van Oorschot et al.	
		6,240,187	05-29-2001	Lewis	
		6,240,183 B1	05-29-2001	Marchant	
		6,289,509	09-11-2001	Kryloff	
		6,301,659 B1	10-09-2001	Micali	
		6,314,409	11-06-2001	Schneck et al.	
		6,324,650	11-27-2001	Ogilvie	
		6,336,186	01-01-2002	Dyksterhouse et al.	
		6,345,314	02-05-2002	Cole et al.	
		6,345,101	02-05-2002	Shukla	
		6,356,941	03-12-2002	Cohen	
		6,363,425	03-26-2002	Hook et al.	
		6,386,451	05-14-2002	Sehr	
		6,424,718	07-23-2002	Holloway	
		6,438,690	08-20-2002	Petal et al.	
		6,449,730	09-10-2002	Mann et al.	
		6,483,921 B1	11-19-2002	Harkins	

Examiner Signature	/Samson Lemma/	Date Considered	04/24/2010

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08a (01-08)
Approved for use through 07/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 3 of 7

Complete if Known	
Application Number	11/258,839 (Conf. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2132
Examiner Name	Gilberto Barron, Jr.
Attorney Docket Number	SFC-2

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		6,553,493	04-22-2003	Okumura et al.	
		6,615,347	09-02-2003	de Silva et al.	
		6,625,734	09-23-2003	Marvit et al.	
		6,978,367	12-20-2005	Hind et al.	

U.S. PATENT PUBLICATIONS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		2001/0001876 A1	05-24-2001	Morgan et al.	
		2001/0051902 A1	12-13-2001	Messner	
		2002/0010679 A1	01-24-2002	Felsher	
		2002/0032663 A1	03-14-2002	Messner	
		2002/0046359 A1	04-18-2002	Boden	
		2002/0071566 A1	06-13-2002	Kurn	
		2002/0129235 A1	09-12-2002	Okamoto et al.	
		2002/0162047	10-31-2002	Peters et al.	
		2003/0051054 A1	03-13-2003	Redlich et al.	
		2003/0070077 A1	04-10-2003	Redlich et al.	
		2003/0167408	09-04-2003	Fitzpatrick	
		2004/0122863 A1	06-24-2004	Sidman	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3] "Number[4] "Kind Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]
		EP 346180 B1	12-13-1989	France Telecom		√
		EP 354774 B1	02-14-1990	International Business Machines Corporation		
		GB 2237670	05-08-1991	Midland Bank PLC		

Examiner Signature	/Samson Lemma/	Date Considered	04/24/2010

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08a (01-08)
Approved for use through 07/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 4 of 7

Complete if Known	
Application Number	11/258,839 (Conf. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2132
Examiner Name	Gilberto Barron, Jr.
Attorney Docket Number	SFC-2

		EP 0485090	05-13-1992	Visa International Service Association		
		JP 04297157	10-21-1992	Mitsubishi Electric Corporation		√
		EP 636259 B1	02-01-1995	Secure Computing Corporation		
		EP 793367 A2	09-03-1997	Oki Electric Industry Co., Ltd.		
		EP 0821504 A2	01-28-1998	Nippon Telegraph and Telephone Corporation		
		EP 0862301 A2	09-02-1998	Fujitsu Limited		
		RU 2124814 C1	01-10-1999	A.A. Moldovjan		√
		EP 1011222 A1	06-21-2000	Fujitsu Limited		
		EP 1239384	09-11-2002	PKWARE Inc.		

FOREIGN PATENT PUBLICATIONS

		WO 98/47091	10-22-1998	Transactor Networks, Inc.		
		WO 99/19845	04-24-1999	David P. Maher		
		WO 99/46720	09-16-1999	Cha! Technologies Services, Inc.		
		WO 99/65207	12-16-1999	Goertzel et al.		
		WO 00/79367	12-28-2000	The Brodia Group		
		WO 01/22650	03-29-2001	Ethentica, Inc.		
		WO 01/22322	03-29-2001	Ethentica, Inc.		
		WO 01/22319	03-29-2001	Ethentica Inc.		
		WO 01/22201	03-29-2001	Ethentica Inc.		
		WO 01/22651	03-29-2001	Ethentica, Inc.		
		WO 02/21761	03-14-2002	Widevine Technologies, Inc.		
		WO 02/21283	03-14-2002	Sanctuary Systems		

Examiner Signature	/Samson Lemma/	Date Considered	04/24/2010

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08a (01-08)
Approved for use through 07/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet	5	of	7

Complete if Known	
Application Number	11/258,839 (Conf. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2132
Examiner Name	Gilberto Barron, Jr.
Attorney Docket Number	SFC-2

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
		Myers et al "A secure, publisher-centric Web caching infrastructure" In: INFOCOM 2001 Proceedings. IEEE Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies [online], vol. 3 p. 1235-1243. Published 04/22/2001. [retrieved on 7-8-2008]. Retrieved from the interenet <URL: http://people.ischool.berkeley.edu/-chuang/pubs/gemini.pdf>	
		RSA SureFile: Software Powered by PKZIP...**BSSF DS 0103** Authorized Reseller: Technical Specifications Platforms Microsoft ® Windows ® 98 Second Edition ME NT 4.0 Workstation SP6A 2000 Professional SP2... www.rsasecurity.com/products/bsafe/datasheets/BSSF_DS_103.pdf	
		Nightingale: The New Secret-Splitting Technology from RSA... **NGBK DS 0403** http://developer.rsasecurity.com/labs/nightingale/developer.rsasecurity.com/labs/nightingale/files/nightingale-brochure.pdf	
		Waldman, et al., "Publius: A robust, tamper-evident, censorship-resistant web publishing sytem," Proceedings of the 9th USENIX Security Symposum, August 2000	
		Hugo Krawczyk, "Distributed Fingerprints and Secure Information Dispersal," 12th ACM, Symposium on Principles on Distributed Computing, Ithaca, NY, ACM 0-89191-613-1/93/0008/0207, 1993, pages 207-218	
		Michael O. Rabin, "Efficient Dispersal of Information for Security, Load Balancing and Fault Tolerance," Journal of the Association for Computing Machinery, Vol. 36, No. 2, pp. 335-348, April 1989	
		Adi Shamir, "How to Share a Secret", Communications of the ACM, Vol. 22, No. 11, pp. 612613, November 1979	
		Garay, et al., "Secure distributed storage and retrieval," Theoretical Comput. Sci., 243(1--2):363—389, July 2000	
		B. Hunter, "Simplfying PKI Usage Through a Client-Server Architecture and Dynamic Propagation of Certificate Paths and Repository Addresses", Proceedings 13th International Workshop on Database and Expert Systems Applications (IEEE Computer Soc., Los Alamitos, CA), 2-6 September 2002, pg. 505-510.	
		Kin-Ching Chan and Chan, S.-H. G., "Distributed Servers Approach for Large-Scale Multicase", IEEE Journal on Selected Areas in Communications (IEEE, Piscataway, NJ) October 2002, 20(8):1500-1510.	

Examiner Signature	/Samson Lemma/	Date Considered	04/24/2010

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08a (01-08)
Approved for use through 07/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 6 of 7

Complete if Known	
Application Number	11/258,839 (Conf. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2132
Examiner Name	Gilberto Barron, Jr.
Attorney Docket Number	SFC-2

		Kin-Ching Chan and Chan, S.-H. G., "Distributed Server Networks for Secure Multicast", GLOBCOM'01: IEEE Global Telecommunications Conference (IEEE, Piscataway, NJ) 3:1974-1978 (2001).	
		S.Y. Shin, Jung-Yeop Kim, R.E. Gantenbein and C.M. Lundquist, "Design a Working Model of Secure Data Transfer Using a Data Mart", Proceedings of the ISCA 14th International Conference Computer Applications in Industry and Engineering (ISCA, Cary, NC) 27-29 November 2001, pg. 66-69.	
		"Lancope Announces Stealthwatch 3.0 for Enhanced Enterprise-Wide Security and Improved Manageability", Business Wire (Newswire Association, Inc.), 14 April 2003.	
		"Decru Unveils Security Appliances for Storage Networks; Decru DataFort (TM) Security Alliances Protect SAN and NAS Environments with Wire-Speed Encryption and Transparent Deployment", PR Newswire (PR Newswire Association. Inc.), 14 October 2002.	
		Lawrence Grant and Fleming B., "Secret Sharing and Splitting", (White Paper) Notre Dame, Indiana, 16 December 2002.	
		Joel McNamara, "Strong Crypto Freeware" (Secret Sharer Version 1.0) July 11, 1995.	
		Dennis Fisher, "RSA Looks to Lock Down Personal Data", eWeek – Enterprise News & Reviews, 14 April 2003.	
		Demir Barlas, "RSA's Security Showcase", Line56.com – The E-Business Executive Daily, 15 April 2003.	
		Marcia Savage, "RSA Unveils Nightingale Technology", CRN.com, 14 April 2003.	
		John K. Waters, "RSA Integrates ID Management; discloses 'Nightingale' ", ADTmag.com, 21 April 2003.	
		Jaikurmar Vijayan, "RSA unveils Management, Encryption Products", Computerworld, 15 April 2003.	
		Eric Doyle, "RSA Splits Data to Stop Hackers", vnunet.com, 16 April 2003.	
		Stan Gibson, "Opinion", eWeek – Enterprise News & Reviews, 14 April 2003.	
		"Trustengine ™ White Paper – Enthentication Services, Secure Storage and Authentication Solutions", Ethentica, Inc. by Security First Corporation, June 2002.	
		"Tactilesense ™ White Paper – A Breakthrough in Fingerprint Authentication", Ethentica, Inc. by Security First Corporation, January, 2003.	
		International Search Report – International Application No. PCT/US/06/45066, dated July 17, 2008	

Examiner Signature	/Samson Lemma/	Date Considered	04/24/2010

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08a (01-08)
Approved for use through 07/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet	7	of	7

Complete if Known	
Application Number	11/258,839 (Conf. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2132
Examiner Name	Gilberto Barron, Jr.
Attorney Docket Number	SFC-2

		M. Loutrel, et al. "An EAP-BT Smartcard for Authentication in the Next Generation of Wireless Communications", Conference on Network Control and Engineering for QoS Security and Mobility (Kluwer Academic Publishers, Norwell, MA) October 23-25, 2002, pgs. 1-4-114.	

Examiner Signature	/Samson Lemma/	Date Considered	04/24/2010

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/258,839	10/25/2005	Rick L. Orsini	SFC-2	2420

1473 7590 05/26/2010

ROPES & GRAY LLP
PATENT DOCKETING 39/361
1211 AVENUE OF THE AMERICAS
NEW YORK, NY 10036-8704

EXAMINER	
LEMMA, SAMSON B	
ART UNIT	PAPER NUMBER
2432	

MAIL DATE	DELIVERY MODE
05/26/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	11/258,839	ORSINI ET AL.	
	Examiner	Art Unit	
	Samson B. Lemma	2432	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 October 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-36,41-54 and 69-85* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-36, 41-54 and 69-85* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This is in reply to election to restriction filed on 02/08/2010. Applicants' election of Group I (claims 1-36, 41-54, and 69-85) without traverse is acknowledged.

2. Claims **1, 18, 22, 26, 30, 31, 34, 35, 41, 47 and 69** are independent claims.

Priority

3. This application claims priority from provisional application 60/622,146 filed on 10/25/2004. Therefore, the effective filling date for the subject matter defined in the pending claims of this application is **October 25, 2004.**

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5. **Claims 1, 18, 20-36, 41-54** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

6. **Claims 1, 18, 20-36, 41-54** are rejected under 35 U.S.C. 101. Based on Supreme Court precedent and recent Federal Circuit decisions, a 35 U.S.C § 101 process must (1) be tied to a particular machine or (2) transform underlying subject matter (such as an article or materials) to a different state or thing. **In re Bilski et al,** 88 USPQ 2d 1385 CAFC

(2008); Diamond v. Diehr, 450 U.S. 175, 184 (1981); Parker v. Flook, 437 U.S. 584, 588 n.9 (1978); Gottschalk v. Benson, 409 U.S. 63, 70 (1972); Cochrane v. Deener, 94 U.S. 780,787-88 (1876).

An example of a method claim that would not qualify as a statutory process would be a claim that recited purely mental steps. Thus, to qualify as a § 101 statutory process, the claim should positively recite the particular machine to which it is tied, for example by identifying the apparatus that accomplishes the method steps, or positively recite the subject matter that is being transformed, for example by identifying the material that is being changed to a different state.

Here, applicant's method steps: recited in either independent claims 1, 22, 26, 30, 31, 34-35, 41 and 47 or even independent claim 22, are broad enough that the steps could be done or completely performed mentally, verbally or without a machine nor is any transformation apparent. Thus, the claims are non-statutory.

For instance independent claims 1, 22, 26, 30, 31, 34-35, 41 and 47, either recite communicating or storing steps which does not necessarily tied the claims to the hardware/machine/device/microprocessor. Furthermore independent claim 18 further recites a "storing portion of data in depositories" however this does not necessarily tied the claims to the hardware/machine/device/microprocessor either, since the depositories could be "software depositories".

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

8. **Claims 1-36, 41-54, and 69-85** are rejected under 35 U.S.C. 102(a) as being anticipated by Orsini et al (hereinafter referred as **Orsini**) (US Patent Publication 2004/0049687 A1) (Date of Publication 03/11/2004)

9. **As per independent claim 1, Orsini discloses, a method for securing communication of a data set from a first location to a second location, the method comprising: generating at the first location at least two portions of data from the data *set*** *[paragraph 0110] (On paragraph 0110, the following has been disclosed, "For example, according to one embodiment, the authentication engine 215 and the cryptographic engine 220 may advantageously employ their respective data splitting modules to divide sensitive data into undecipherable portions, and then transmit one or more undecipherable portions of the sensitive data to a particular data storage facility.),* **wherein each of the at least two portions of data respectively contains a substantially random distribution of a respective subset of the data set** *[paragraph 0113-0114, "FIG. 8 illustrates a flowchart of a data splitting process 800*

performed by the data splitting module according to aspects of an embodiment of the invention. As shown in FIG. 8, the data splitting process 800 begins at step 805 when sensitive data "S" is received by the data splitting module of the authentication engine 215 or the cryptographic engine 220. Preferably, in step 810, the data splitting module then generates a substantially random number, value, or string or set of bits, "A." For example, the random number A may be generated in a wide number of varying conventional techniques available to one of ordinary skill in the art, for producing high quality random numbers suitable for use in cryptographic applications. In addition, according to one embodiment, the random number A comprises a bit length which may be any suitable length, such as shorter, longer or equal to the bit length of the sensitive data, S. Furthermore on paragraph 0114, the following has been disclosed.In addition, in step 820 the data splitting process 800 generates another statistically random number "C." According to the preferred embodiment, the generation of the statistically random numbers A and C may advantageously be done in parallel);

communicating from the first location to the second location the at least two portions of data over at least one communications path, wherein the at least two portions of data are communicated separately from each other *[See paragraph 0114 see for instance the following which is disclosed on paragraph 0114, "According to one embodiment, each of the foregoing pairings is distributed to one of the depositories D1 through D4 of FIG. 7. According to another embodiment,*

each of the foregoing pairings is randomly distributed to one of the depositories D1 through D4".]; **and restoring at the second location the data set from at least a subset of the at least two portions of data** [*See at least paragraph 0117, "Thus, the data assembling module 525 may assemble the sensitive data S, when, for example, it receives data portions from at least the first two of the data storage facilities D1 through D4 to respond to an assemble request by the trust engine 110.")* .

10. **As per independent claims 18, 22, 26, 30, 31, 34, 35, 41, 47 and 69, independent claims 18, 22, 26, 30, 31, 34, 35, 41, 47 and 69 are rejected for the same reasons as that of the independent claim 1. Additional features which are added in some of the above independent claims are also disclosed by Orsini. See for instance,** *[paragraph 110-paragraph 117, paragraph 78, paragraph 110-paragraph 117, paragraph 160, 166, 189 and see finally paragraph 278- paragraph 423]*

11. **As per dependent claims 2-17, 19-21, 23-25, 27-29, 32-33, 36, 42-46, 48-54 and 70-85, Orsini discloses, a method as applied to claims above. Furthermore Orsini discloses the method wherein the communicating comprises communicating the at least two portions of data on separate communications paths in parallel.** *[See page 11, left side column line 3 and also see page 10, paragraph 110-page 11,*

paragraph 117 and see also paragraph 78, paragraph 160, 166, 189 and see finally paragraph 278- paragraph 423]

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Samson B Lemma whose telephone number is 571-272-3806. The examiner can normally be reached on Monday-Friday (8:00 am---4: 30 pm).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, BARRON JR GILBERTO can be reached on 571-272-3799. The fax phone number for the organization where this application or proceeding is assigned is 703-873-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Samson B Lemma/
Examiner, Art Unit 2432

/Jung Kim/
Primary Examiner, AU 2432

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 2420

SERIAL NUMBER	FILING or 371(c) DATE RULE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
11/258,839	10/25/2005	713	2432	SFC-2

APPLICANTS

Rick L. Orsini, Flower Mound, TX;
Mark S. O'Hare, Coto De Caza, CA;
Roger Davenport, Campbell, TX;
Steven Winick, Roslyn Heights, NY;

**** CONTINUING DATA ***************************

This appln claims benefit of 60/622,146 10/25/2004
and claims benefit of 60/718,185 09/16/2005

**** FOREIGN APPLICATIONS ***************************

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED **** ** SMALL ENTITY **
01/17/2006

Foreign Priority claimed ☐ Yes ☑ No 35 USC 119(a-d) conditions met ☐ Yes ☑ No Verified and Acknowledged /SAMSON B LEMMA/ Examiner's Signature	☐ Met after Allowance Initials	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
		TX	36	85	21

ADDRESS

ROPES & GRAY LLP
PATENT DOCKETING 39/361
1211 AVENUE OF THE AMERICAS
NEW YORK, NY 10036-8704
UNITED STATES

TITLE

Secure data parser method and system

FILING FEE RECEIVED		
4365	FEES: Authority has been given in Paper No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit

BIB (Rev. 05/07)

PTO/SB/08a (01-08)
Approved for use through 04/31/2009. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 2

Complete if Known	
Application Number	11/258,839 (Conf. No. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2432
Examiner Name	Samson B. Lemma
Attorney Docket Number	SFC-2

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		7,069,451 B1	06-27-2006	Ginter, et al.	
		7,391,865	06-24-2008	Orsini et al.	

U.S. PATENT PUBLICATIONS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		2002/0091640 A1	07-11-2002	Gupta	
		2003/0058274 A1	03-27-2003	Hill et al.	
		2006/0282681 A1	12-14-2006	Scheidt et al.	
		2007/0160198	07-12-2007	Orsini et al.	
		2008/0137857 A1	06-12-2008	Bellare et al.	
		2008/0183992 A1	07-31-2008	Martin et al.	
		2008/0244277 A1	10-02-2008	Orsini et al.	
		2009/0097661 A1	04-16-2009	Orsini et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3] "Number[4] "Kind Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]
		WO 2009/035674 A1	03-19-2009	Security First Corporation		

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
		Crescenzo, G. et al., "Non-Interactive and Non-Malleable Commitment," Proceedings of the 30th Annual ACM Symposium on Theory of Computing. Dallas, TX, May 23-26, 1998, [Proceedings of the 30th Annual ACM Symposium on Theory of Computing], New York, NY: ACM, US, PAGES 141-150; XP000970902; ISBN: 978-0-89791-962-3.	

Examiner Signature	/Samson Lemma/	Date Considered	04/24/2010

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08a (01-08)
Approved for use through 04/31/2009. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no Persons are required to respond to a collection of information unless it contains a valid OMR control number.

Substitute for form 1449A/PTO

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 2 of 2

Complete if Known	
Application Number	11/258,839 (Conf. No. 2420)
Filing Date	October 25, 2005
First Named Inventor	Rick L. Orsini
Art Unit	2432
Examiner Name	Samson B. Lemma
Attorney Docket Number	SFC-2

		Christian Cachin, "On-Line secret Sharing," Cryptography and Coding. IMA Conference, Proceedings, XX, XX, December 18, 1995, pages 190-198, XP002137681.	
		Mayer, A. et al., "Generalized Secret Sharing and Group-Key Distribution Using Short Keys," Compression and Complexity of Sequences 1997, Proceedings Salerno, Italy, June 11-13, 1997, Los Alamitos, CA, USA, IEEE Comput,. SOC, US, June 11, 1997, pages 30-44, XP010274905, ISBN: 978-0-8186-8132-5.	

Examiner Signature	/Samson Lemma/	Date Considered	04/24/2010

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1]Applicant's unique citation designation number (optional). [2]See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3]Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4]For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5]Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6]Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.L

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Notice of References Cited	Application/Control No. 11/258,839	Applicant(s)/Patent Under Reexamination ORSINI ET AL.	
	Examiner Samson B. Lemma	Art Unit 2432	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-2004/0049687	03-2004	Orsini et al.	713/189
*	B	US-5,966,448	10-1999	Namba et al.	380/33
*	C	US-2003/0167408	09-2003	Fitzpatrick et al.	713/201
*	D	US-2002/0162047	10-2002	Peters et al.	714/5
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11258839	ORSINI ET AL.
	Examiner	Art Unit
	Samson B Lemma	2432

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	12/29/2009	04/15/2010							
	1	÷	✓							
	2	÷	✓							
	3	÷	✓							
	4	÷	✓							
	5	÷	✓							
	6	÷	✓							
	7	÷	✓							
	8	÷	✓							
	9	÷	✓							
	10	÷	✓							
	11	÷	✓							
	12	÷	✓							
	13	÷	✓							
	14	÷	✓							
	15	÷	✓							
	16	÷	✓							
	17	÷	✓							
	18	÷	✓							
	19	÷	✓							
	20	÷	✓							
	21	÷	✓							
	22	÷	✓							
	23	÷	✓							
	24	÷	✓							
	25	÷	✓							
	26	÷	✓							
	27	÷	✓							
	28	÷	✓							
	29	÷	✓							
	30	÷	✓							
	31	÷	✓							
	32	÷	✓							
	33	÷	✓							
	34	÷	✓							
	35	÷	✓							
	36	÷	✓							

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11258839	ORSINI ET AL.
	Examiner	Art Unit
	Samson B Lemma	2432

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	12/29/2009	04/15/2010							
	37	÷	-							
	38	÷	-							
	39	÷	-							
	40	÷	-							
	41	÷	✓							
	42	÷	✓							
	43	÷	✓							
	44	÷	✓							
	45	÷	✓							
	46	÷	✓							
	47	÷	✓							
	48	÷	✓							
	49	÷	✓							
	50	÷	✓							
	51	÷	✓							
	52	÷	✓							
	53	÷	✓							
	54	÷	✓							
	55	÷	-							
	56	÷	-							
	57	÷	-							
	58	÷	-							
	59	÷	-							
	60	÷	-							
	61	÷	-							
	62	÷	-							
	63	÷	-							
	64	÷	-							
	65	÷	-							
	66	÷	-							
	67	÷	-							
	68	÷	-							
	69	÷	✓							
	70	÷	✓							
	71	÷	✓							
	72	÷	✓							

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11258839	ORSINI ET AL.
	Examiner	Art Unit
	Samson B Lemma	2432

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	12/29/2009	04/15/2010							
	73	÷	✓							
	74	÷	✓							
	75	÷	✓							
	76	÷	✓							
	77	÷	✓							
	78	÷	✓							
	79	÷	✓							
	80	÷	✓							
	81	÷	✓							
	82	÷	✓							
	83	÷	✓							
	84	÷	✓							
	85	÷	✓							

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11258839	ORSINI ET AL.
	Examiner	Art Unit
	Samson B Lemma	2432

SEARCHED

Class	Subclass	Date	Examiner
713	176, 186,156		
380	33, 268, 270, 247, 286		

SEARCH NOTES

Search Notes	Date	Examiner
713/$, 380/$ (With text search)	05/10/2010	SL
EAST (See attached)	05/10/2010	SL
NPL (IEEE, ACM DIGITAL library, google, citeseer,...)	05/10/2010	SL
Inventor's name search	05/10/2010	SL

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner

PTO/SB/22 (07-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARMENT OF COMMERCE
Under the paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a) **FY 2009** *(Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).)*	Docket Number (Optional) SFC-2
Application Number 11/258,839	Filed October 25, 2005
For Secure Data Parser Method and System	
Art Unit 2432	Examiner Samson B. Lemma

This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above identified application.

The requested extension and fee are as follows (check time period desired and enter the appropriate fee below):

		Fee	Small Entity Fee	
☐	One month (37 CFR 1.17(a)(1))	$130	$65	$
☑	Two months (37 CFR 1.17(a)(2))	$490	$245	$ 490.00
☐	Three months (37 CFR 1.17(a)(3))	$1110	$555	$
☐	Four months (37 CFR 1.17(a)(4))	$1730	$865	$
☐	Five months (37 CFR 1.17(a)(5))	$2350	$1175	$

☐ Applicant claims small entity status. See 37 CFR 1.27.

☐ A check in the amount of the fee is enclosed.

☐ Payment by credit card. Form PTO-2038 is attached.

☑ The Director has already been authorized to charge fees in this application to a Deposit Account.

☑ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number 06-1075.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

I am the

☐ applicant/inventor.

☐ assignee of record of the entire interest. See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed (Form PTO/SB/96).

☐ attorney or agent of record. Registration Number _____

☑ attorney or agent under 37 CFR 1.34. Registration number if acting under 37 CFR 1.34 54,026

/Michael J. Chasan/	October 26, 2010
Signature	Date
Michael J. Chasan	(212) 596-9000
Typed or printed name	Telephone Number

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☑ Total of 1 forms are submitted.

This collection of information is required by 37 CFR 1.136(a). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 6 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number	Filing Date	☐ To be Mailed
11/258,839	10/25/2005	

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY ☒ RATE ($)	SMALL ENTITY ☒ FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT	10/26/2010	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 66	Minus	** 85	= 0	X $26 =	0	OR	X $ =	
	Independent (37 CFR 1.16(h))	* 11	Minus	***21	= 0	X $110 =	0	OR	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE	0	OR	TOTAL ADD'L FEE	

AMENDMENT		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)	OR	RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/eugenia v. hardy/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PATENTS
SFC-2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION

Applicants : Rick Orsini et al.

Application No. : 11/258,839 Confirmation No. : 2420

Filed : October 25, 2005

For : SECURE DATA PARSER METHOD AND SYSTEM

Art Unit : 2432

Examiner : Samson B. Lemma

New York, New York 10036
October 26, 2010

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

REPLY TO OFFICE ACTION

Sir:

In reply to the Office Action dated May 26, 2010, applicants hereby amend the above-identified patent application as follows:

Amendments to the Claims are reflected in the listing beginning on page 2 of this Reply to Office Action.

Remarks begin on page 18 of this Reply to Office Action.

REMARKS

Summary of Office Action

Claims 1-36, 41-54, and 69-85 were pending in the above-identified patent application. Claims 37-40 and 55-68 were withdrawn in response to the Restriction Requirement mailed January 7, 2010.

The Examiner rejected claims 1, 18, 20-36, and 41-54 under 35 U.S.C. § 101 as being directed to non-statutory subject matter.

The Examiner rejected claims 1-36, 41-54, and 69-85 under 35 U.S.C. § 102(a) as being anticipated by Orsini et al. U.S. Patent Publication 2004/0049687 (hereinafter "Orsini").

Summary of Applicants' Reply

Applicants have amended claims 1-3, 6-13, 17, 18, 22, 26, 30, 31, 34, 35, 41, 47, and 69-71 in order to more particularly define the invention. These amendments add no new matter and are fully supported by applicants' originally filed specification. Support for these claim amendments can be found, for example, within paragraphs **[0409]** and **[0427]** and FIGS. 31 and 32. Applicants have cancelled withdrawn claims 37-40 and 55-68, without prejudice. Applicants have amended independent claim 41 to incorporate the subject matter of dependent claim 42 and have cancelled claim 42, without prejudice.

Applicants' Reply to the
35 U.S.C. § 101 Rejections

Claims 1, 18, 20-36, and 41-54 were rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. In the interest of expediting the prosecution of the present application, and without conceding the issue of patentability, applicants have amended independent claims 1, 18, 22, 26, 30, 31, 34, 35, 41, and 47 to more clearly recite particular apparatus elements that accomplish the claimed method steps. Applicants respectfully submit that the amended independent claims positively recite "processing circuitry" that accomplishes the claimed method steps and respectfully submit that the amended independent falls under one of the statutory categories of invention. Accordingly, applicants respectfully request that the rejection under 35 U.S.C. § 101 of claims 1, 18, 20-36, and 41-54 be withdrawn.

Applicants' Reply to the
Prior Art Rejections

Claims 1-36, 41-54, and 69-85 are rejected under 35 U.S.C. § 102(a) as being anticipated by Orsini.

I. Claims 1-17 and 69-85

Applicants have amended independent claims 1 and 69 to specify applicants' claimed approach for securing ***streaming*** of a data set from a first location to a second location. In particular, applicants' amended claims recite: "***streaming*** from the first location to the second location the at least two portions of data over at least one communications path," and "restoring at the second location data from at least a subset of the at least portions of data, ***wherein the data is restored***

as the at least two streaming portions of data are being received at the second location. (emphasis added)

Applicants' amended claims represent a patentable improvement over Orsini. In particular, streams of data (e.g., audio, video, etc.) may be secured by splitting them into multiple portions that may be streamed from a first location to a second location. If any one of these multiple streams of data is compromised, the original data may remain secure. The system may require that more than one of the streaming portions be combined in order to restore the original data. Disclosure cannot be found in Orsini for this claimed feature of securing streams of data. Disclosure also cannot be found in Orsini for restoring data as the at least two streaming portions of data are received at the second location. For at least this reason, applicants submit that amended independent claims 1 and 69 and dependent claims 2-17 and 70-85 are not anticipated by Orsini.

II. Claims 18-21

Applicants' independent claim 18 is directed to a method for securing a data set by generating at least two portions of data from the data set. In particular, independent claim 18 recites "using at least one of the at least two portions of data . . . ***as an authentication key stored in an authentication depository in order to restore the data set.***" (emphasis added) This feature of applicants' independent claim 18 represents a patentable improvement over Orsini. In particular, disclosure cannot be found in Orsini for using a portion of data "as an authentication key stored in an authentication depository." Furthermore, the Examiner has not pointed to any portion of Orsini as allegedly showing

this feature. For at least this reason, applicants submit that independent claim 18 and dependent claims 19-21 are not anticipated by Orsini.

III. Claims 22-25

Applicants' independent claim 22 is directed to a method for securing a data set by generating at least two portions of data from the data set. In particular, independent claim 22 recites "generating respective ***integrity information*** for each of the at least two portions of data," and "writing ***the respective integrity information*** to each of the at least two portions of data." (emphasis added). These features of applicants' independent claim 22 represent a patentable improvement over Orsini. In particular, disclosure cannot be found in Orsini for generating integrity information for the portions of data and/or writing the integrity information to the portions of data. Furthermore, the Examiner has not pointed to any portion of Orsini as allegedly showing this feature. For at least this reason, applicants submit that independent claim 22 and dependent claims 23-25 are not anticipated by Orsini.

IV. Claims 26-29

Applicants' independent claim 26 is directed to a method for securing a data set by generating at least two portions of data from the data set. In particular, independent claim 26 recites "wherein ***at least one particular portion of data is required*** in order to restore the data set." (emphasis added). These features of applicants' independent claim 26 represent a patentable improvement over Orsini. In particular, disclosure cannot be found in Orsini for requiring

at least one particular portion of data to restore the data set. Furthermore, the Examiner has not pointed to any portion of Orsini as allegedly showing this feature. For at least this reason, applicants submit that independent claim 26 and dependent claims 27-29 are not anticipated by Orsini.

V. Claim 30

Applicants' independent claim 30 is directed to a method for securing a data set by generating at least two portions of data from the data set. In particular, independent claim 30 recites "***wherein the probability that data from the data set is distributed to at least one of the at least two portions of data is less than the probability that data from the data set is distributed to a remainder of the at least two portions of data***." (emphasis added). This feature of applicants' independent claim 30 represents a patentable improvement over Orsini. In particular, disclosure cannot be found in Orsini for probabilities of distributing data to particular portions of data. Furthermore, the Examiner has not pointed to any portion of Orsini as allegedly showing this feature. For at least this reason, applicants submit that independent claim 30 is not anticipated by Orsini.

VI. Claims 31-33

Applicants' independent claim 31 is directed to a method for securing a data set by generating at least two portions of data from the data set. In particular, independent claim 31 recites "wherein at least one of the at least two portions of data has a ***predetermined size***." (emphasis added). This feature of applicants' independent claim 31 represents a patentable improvement over Orsini. In

particular, disclosure cannot be found in Orsini for a portion of data having a predetermined size. Furthermore, the Examiner has not pointed to any portion of Orsini as allegedly showing this feature. For at least this reason, applicants submit that independent claim 31 and dependent claims 32 and 33 are not anticipated by Orsini.

VII. Claim 34

Applicants' independent claim 34 is directed to a method for securing a data set by generating at least two portions of data from the data set. In particular, independent claim 34 recites "***wherein the probability that data from the data set is distributed to each of the at least two portions of data is substantially equal***." (emphasis added). This feature of applicants' independent claim 34 represents a patentable improvement over Orsini. In particular, disclosure cannot be found in Orsini for probabilities of distributing data to particular portions of data. Furthermore, the Examiner has not pointed to any portion of Orsini as allegedly showing this feature. For at least this reason, applicants submit that independent claim 34 is not anticipated by Orsini.

VIII. Claims 35 and 36

Applicants' independent claim 35 is directed to a method for securing a data set by generating at least two portions of data from the data set. In particular, independent claim 35 recites "generating, ***according to a Shamir algorithm***, splitting key information that is capable of being used to generate the splitting key, wherein the splitting key information is distributed among the at least

two portions of data." (emphasis added). This feature of applicants' independent claim 35 represents a patentable improvement over Orsini. In particular, disclosure cannot be found in Orsini for generating splitting key information according to a Shamir algorithm. Furthermore, the Examiner has not pointed to any portion of Orsini as allegedly showing this feature. For at least this reason, applicants submit that independent claim 35 and dependent claim 36 are not anticipated by Orsini.

IX. Claims 41 and 43-46

Applicants' amended independent claim 41 is directed to a method for securing a data set by determining into which of the portions of data to distribute each unit of data from the data set based on corresponding data from a substantially random number and distributing accordingly. In particular, amended independent claim 41 recites "wherein the determining into which of the portions of data to distribute each unit of data from the data set comprises determining into which of the portions of data to distribute each unit of data from the data ***set based on a hash function of corresponding data from the substantially random number***." (emphasis added). This feature of applicants' amended independent claim 41 represents a patentable improvement over Orsini. In particular, disclosure cannot be found in Orsini for determining into which of the portions of data to distribute each unit of data based on a hash function. Furthermore, the Examiner has not pointed to any portion of Orsini as allegedly showing this feature. For at least this reason, applicants submit that amended independent claim 41 and dependent claims 43-46 are not anticipated by Orsini.

X. Claims 47-54

Applicants' independent claim 47 is directed to a method for securing a data set by distributing data from the data set into portions of data. In particular, independent claim 47 recites "for each unit of data of the data set, ***splitting the unit of data into a left segment and a right segment by determining, based on corresponding data from the substantially random number, a splitting location*** wherein bits to the left of the splitting location represent the left segment and bits to the right of the splitting location represent the right segment," and "determining into which of the portions of data to distribute ***each of the left segment and the right segment of each unit of data*** from the data set based on corresponding data from the substantially random number and distributing accordingly." (emphasis added). These features of applicants' independent claim 31 represent a patentable improvement over Orsini. In particular, disclosure cannot be found in Orsini for splitting units of data into left and right segments and determining into which portions of data to distribute each of the segments. Furthermore, the Examiner has not pointed to any portion of Orsini as allegedly showing this feature. For at least this reason, applicants submit that independent claim 47 and dependent claims 47-54 are not anticipated by Orsini.

Conclusion

For at least the reasons set forth above, applicants respectfully submit that this application, as amended, is patentable. Reconsideration and prompt allowance of this application are respectfully requested.

Respectfully submitted,

/Michael J. Chasan/

Michael J. Chasan
Reg. No. 54,026
Attorney for Applicants
ROPES & GRAY LLP
Customer No. 1473
Tel.: (212) 596-9000

Amendments to the Claims

The following listing of claims will replace all prior versions, and listings, of claims in the above-identified patent application.

Listing of Claims

1. (Currently Amended) A method for securing ~~communication~~ streaming of a data set from a first location to a second location, the method comprising:

generating, using processing circuitry at the first location, at least two portions of data from the data set, wherein each of the at least two portions of data respectively contains a substantially random distribution of a respective subset of the data set;

~~communicating~~ streaming from the first location to the second location the at least two portions of data over at least one communications path, wherein the at least two portions of data are communicated separately from each other; and

restoring, using processing circuitry at the second location, ~~the~~ data ~~set~~ from at least a subset of the at least two portions of data, wherein the data is restored as the at least two streaming portions of data are being received at the second location.

2. (Currently Amended) The method of claim 1 wherein the ~~communicating~~ streaming comprises ~~communicating~~ streaming the at least two portions of data on separate communications paths in parallel.

3. (Currently Amended) The method of claim 1 wherein the ~~communicating~~ streaming comprises ~~communicating~~ streaming that at least two portions of data on a single communications path in a serial transmission.

4. (Original) The method of claim 1 wherein the generating is accomplished using an email application.

5. (Original) The method of claim 1 wherein the generating is accomplished using a video broadcasting application.

6. (Currently Amended) The method of claim 1 wherein the ~~communicating~~ streaming over the at least one communications path comprises ~~communicating~~ streaming over a wireless communications path.

7. (Currently Amended) The method of claim 6, wherein the ~~communicating~~ streaming over the wireless communications path comprises ~~communicating~~ streaming over the wireless communications path using an 802.11 protocol.

8. (Currently Amended) The method of claim 6, wherein the ~~communicating~~ streaming over the wireless communications path comprises ~~communicating~~ streaming over the wireless communications path using Bluetooth.

9. (Currently Amended) The method of claim 6, wherein the ~~communicating~~ streaming over the wireless

communications path comprises ~~communicating~~ streaming over a wireless LAN.

10. (Currently Amended) The method of claim 6, wherein the ~~communicating~~ streaming over the wireless communications path comprises ~~communicating~~ streaming over a cellular network.

11. (Currently Amended) The method of claim 6, wherein the ~~communicating~~ streaming over the wireless communications path comprises ~~communicating~~ streaming over a peer to peer wireless network.

12. (Currently Amended) The method of claim 6, wherein the ~~communicating~~ streaming over the wireless communications path comprises ~~communicating~~ streaming over a hub based wireless network.

13. (Currently Amended) The method of claim 6, wherein the ~~communicating~~ streaming over the wireless communications path comprises ~~communicating~~ streaming over a satellite transmission.

14. (Original) The method of claim 1 wherein at least one of the first location and the second location is a computer peripheral device.

15. (Original) The method of claim 1 wherein at least one of the first location and the second location is a PDA device.

16. (Original) The method of claim 1 wherein at least one of the first location and the second location is a storage device.

17. (Currently Amended) The method of claim 1 wherein the ~~communicating~~ streaming over the at least one communications path comprises ~~communicating~~ streaming over a wired communications path.

18. (Currently Amended) A method for securing a data set, the method comprising:

generating, using processing circuitry, at least two portions of data from the data set, wherein each of the at least two portions of data respectively contains a substantially random distribution of a respective subset of the data set;

storing the at least two portions of data on at least two data depositories, wherein the at least two data depositories are physically separate from each other; and

using at least one of the at least two portions of data stored on at least one of the at least two data depositories as an authentication key stored in an authentication depository in order to restore the data set.

19. (Original) The method of claim 18 wherein the authentication depository is a physical token device.

20. (Original) The method of claim 18 further comprising restoring the data set when the authentication depository is coupled to at least a subset of the at least two data depositories.

21. (Original) The method of claim 20 further comprising removing access to the data set when the authentication depository is no longer coupled to the at least a subset of the at least two data depositories.

22. (Currently Amended) A method for securing a data set, the method comprising:

generating, using processing circuitry, at least two portions of data from the data set, wherein each of the at least two portions of data respectively contains a substantially random distribution of a respective subset of the data set;

generating respective integrity information for each of the at least two portions of data;

writing the respective integrity information to each of the at least two portions of data;

storing each of the at least two portions of data separately; and

restoring the data set from at least a subset of the at least two portions of data when the respective integrity information of the at least a subset of the at least two portions of data has been verified.

23. (Original) The method of claim 22, wherein the integrity information is generated using a hash algorithm.

24. (Original) The method of claim 22, wherein the integrity information is generated using a MAC signature.

25. (Original) The method of claim 22, wherein the integrity information is generated using a digital signature.

26. (Currently Amended) A method for securing a data set, the method comprising:

generating, using processing circuitry, at least two portions of data from the data set, wherein each of the at least two portions of data respectively contains a substantially random distribution of a respective subset of the data set;

storing each of the at least two portions of data separately; and

restoring the data set from at least a subset of the at least two portions of data, wherein at least one particular portion of data is required in order to restore the data set.

27. (Original) The method of claim 26 further comprising encrypting at least one of the at least two portions of data, the at least one of the at least two portions of data comprising the at least one particular portion of data.

28. (Original) The method of claim 27 wherein the encrypting comprises encrypting at least one of the at least two portions of data according to a first encryption key and at least one other of the at two least portions of data according to a second encryption key.

29. (Original) The method of claim 27 wherein the encrypting comprises encrypting at least one of the at least

two portion of data using a public key that may be decrypted using a private key.

30. (Currently Amended) A method for securing a data set, the method comprising:

generating, using processing circuitry, at least two portions of data from the data set, wherein each of the at least two portions of data respectively contains a substantially random distribution of a respective subset of the data set, and wherein the probability that data from the data set is distributed to at least one of the at least two portions of data is less than the probability that data from the data set is distributed to a remainder of the at least two portions of data;

storing each of the at least two portions of data separately; and

restoring the data set from at least a subset of the at least two portions of data.

31. (Currently Amended) A method for securing a data set, the method comprising:

generating, using processing circuitry, at least two portions of data from the data set, wherein each of the at least two portions of data respectively contains a substantially random distribution of a respective subset of the data set, and wherein at least one of the at least two portions of data has a predetermined size;

storing each of the at least two portions of data separately; and

restoring the data set from at least a subset of the at least two portions of data.

32. (Original) The method of claim 31 wherein the generating the at least two portions of data comprises using a splitting key having a splitting key size and wherein the predetermined size is the splitting key size.

33. (Original) The method of claim 31 further comprising encrypting the data set with an encryption key having an encryption key size prior to the generating the at least two portions of data and wherein the predetermined size is the encryption key size.

34. (Currently Amended) A method for securing a data set, the method comprising:

generating, using processing circuitry, at least two portions of data from the data set, wherein each of the at least two portions of data respectively contains a substantially random distribution of a respective subset of the data set, and wherein the probability that data from the data set is distributed to each of the at least two portions of data is substantially equal;

storing each of the at least two portions of data separately; and

restoring the data set from at least a subset of the at least two portions of data.

35. (Currently Amended) A method for securing a data set, the method comprising:

generating, using processing circuitry, according to a splitting key, at least two portions of data from the data set, wherein each of the at least two portions

of data respectively contains a substantially random distribution of a respective subset of the data set;

generating, according to a Shamir algorithm, splitting key information that is capable of being used to generate the splitting key, wherein the splitting key information is distributed among the at least two portions of data;

storing each of the at least two portions of data separately; and

restoring the data set from at least a subset of the at least two portions of data.

36. (Original) The method of claim 35 wherein restoring the data set comprises restoring the splitting key from the splitting key information from at least a subset of the at least two portions of data.

37-40. (Cancelled)

41. (Currently Amended) A method for securing a data set, the method comprising:

determining a number of portions of data into which to distribute the data set;

generating a substantially random number;

determining, using processing circuitry, into which of the portions of data to distribute each unit of data from the data set based on corresponding data from the substantially random number and distributing accordingly;

storing the portions of data separately; and

restoring the data set from at least a subset of the portions of data, wherein the determining into which of

the portions of data to distribute each unit of data from the data set comprises determining into which of the portions of data to distribute each unit of data from the data set based on a hash function of corresponding data from the substantially random number.

42. (Cancelled)

43. (Original) The method of claim 41 wherein the unit of data is a bit.

44. (Original) The method of claim 41 wherein the size of the substantially random number is a multiple, M, of the size of the data set and wherein the determining into which of the portions of data to distribute each unit of data from the data set comprises determining into which of the portions of data to distribute each unit of data from the data set based on a hash function of a corresponding M bits from the substantially random number.

45. (Original) The method of claim 44 wherein the multiple, M, is 8 and wherein the determining into which of the portions of data to distribute each unit of data from the data set based on a hash function of a corresponding M bits from the substantially random number comprises determining into which of the portions of data to distribute each bit of data from the data set based on a hash function of a corresponding byte from the substantially random number.

46. (Original) The method of claim 41 further comprising appending redundancy data to each of the portions of data.

47. (Currently Amended) A method for securing a data set, the method comprising:

determining a number of portions of data into which to distribute the data set;

generating a substantially random number;

for each unit of data of the data set, splitting the unit of data into a left segment and a right segment by determining, based on corresponding data from the substantially random number, a splitting location wherein bits to the left of the splitting location represent the left segment and bits to the right of the splitting location represent the right segment;

determining, using processing circuitry, into which of the portions of data to distribute each of the left segment and the right segment of each unit of data from the data set based on corresponding data from the substantially random number and distributing accordingly;

storing the portions of data separately; and

restoring the data set from at least a subset of the portions of data.

48. (Original) The method of claim 47 wherein the splitting the unit of data into the left segment and the right segment comprises splitting the unit of data into the left segment and the right segment based on a hash function of a corresponding data from the substantially random number.

49. (Original) The method of claim 47 wherein the unit of data is a byte.

50. (Original) The method of claim 47 wherein generating the substantially random number comprises generating the substantially random number of size equal to the size of the data set.

51. (Original) The method of claim 50 wherein the unit of data is a byte and wherein the splitting the unit of data into the left segment and the right segment comprises splitting the unit of data into the left segment and the right segment based on a hash function of a corresponding unit of data from the substantially random number.

52. (Original) The method of claim 47 wherein the determining into which of the portions of data to distribute each of the left segment and the right segment comprises determining into which of the portions of data to distribute each of the left segment and the right segment of each unit of data from the data set based on corresponding data from the substantially random number and based on a table containing an array of all possible distributions of the right segment and the left.

53. (Original) The method of 47 wherein the determining into which of the portions of data to distribute each of the left segment and the right segment comprises determining into which of the portions of data to distribute each of the left segment and the right segment of each unit of data from the data set based on a hash function of

corresponding data from the substantially random number and based on a table containing an array of all possible distributions of the right segment and the left segment, wherein the hash function of the corresponding data from the substantially random number indicates which entry in the table to use, the entry indicating a destination portion for the left segment and a destination portion for the right segment.

54. (Original) The method of claim 53 wherein:

generating the substantially random number comprises generating the substantially random number of size equal to the size of the data set;

the unit of data is a byte; and

the corresponding data is a corresponding unit of data.

55-68. (Cancelled)

69. (Currently Amended) A system for securing ~~communication~~ streaming of a data set from a first location to a second location, the system comprising:

a first processor for generating at the first location at least two portions of data from the data set, wherein each of the at least two portions of data respectively contains a substantially random distribution of a respective subset of the data set;

at least one communications path over which are ~~communicated~~ streamed the at least two portions of data from the first location to the second location, wherein the at least two portions of data are communicated separately from each other; and

a second processor for restoring at the second location ~~the~~ data ~~set~~ from at least a subset of the at least two streaming portions of data, wherein the data is restored as the at least two streaming portions of data are being received at the second location.

70. (Currently Amended) The system of claim 69 wherein the at least two streaming portions of data are ~~communicated~~ streamed on separate communications paths in parallel.

71. (Currently Amended) The system of claim 69 wherein the at least two streaming portions of data are ~~communicated~~ streamed on a single communications path in a serial transmission.

72. (Original) The system of claim 69 wherein the first processor is used in conjunction with an email application.

73. (Original) The system of claim 69 wherein the first processor is used in conjunction with a video broadcasting application.

74. (Original) The system of claim 69 wherein the at least one communications path comprises a wireless communications path.

75. (Original) The system of claim 74, wherein the wireless communications path comprises an 802.11 protocol communications path.

76. (Original) The system of claim 74, wherein the wireless communications path comprises a Bluetooth communications path.

77. (Original) The system of claim 74, wherein the wireless communications path comprises a wireless LAN.

78. (Original) The system of claim 74, wherein the wireless communications path comprises a cellular network.

79. (Original) The system of claim 74, wherein the wireless communications path comprises a peer to peer wireless network.

80. (Original) The system of claim 74, wherein the wireless communications path comprises a hub based wireless network.

81. (Original) The system of claim 74, wherein wireless communications path comprises a satellite communications path.

82. (Original) The system of claim 69 wherein at least one of the first location and the second location is a computer peripheral device.

83. (Original) The system of claim 69 wherein at least one of the first location and the second location is a PDA device.

84. (Original) The system of claim 69 wherein at least one of the first location and the second location is a storage device.

85. (Original) The system of claim 69 wherein the at least one communications path comprises a wired communications path.

Electronic Patent Application Fee Transmittal

Application Number:	11258839
Filing Date:	25-Oct-2005
Title of Invention:	Secure data parser method and system
First Named Inventor/Applicant Name:	Rick L. Orsini
Filer:	Michael Jeremy Chasan
Attorney Docket Number:	SFC-2

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Extension - 2 months with $0 paid	1252	1	490	490

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
			Total in USD ($)	**490**

Electronic Acknowledgement Receipt

EFS ID:	8703594
Application Number:	11258839
International Application Number:	
Confirmation Number:	2420
Title of Invention:	Secure data parser method and system
First Named Inventor/Applicant Name:	Rick L. Orsini
Customer Number:	01473
Filer:	Michael Jeremy Chasan
Filer Authorized By:	
Attorney Docket Number:	SFC-2
Receipt Date:	26-OCT-2010
Filing Date:	25-OCT-2005
Time Stamp:	18:48:52
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$490
RAM confirmation Number	5555
Deposit Account	061075
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Extension of Time	SFC_2_EOT.pdf	50441 b366618271223235e9f69a828fde6ef3036acb5d	no	2
Warnings:					
Information:					
2		SFC_2_Reply.pdf	55856 28ef1d38d4ed9372c185aebf58eedd3e10b0d136	yes	26

Multipart Description/PDF files in .zip description

Document Description	Start	End
Amendment/Req. Reconsideration-After Non-Final Reject	1	1
Claims	2	17
Applicant Arguments/Remarks Made in an Amendment	18	26

Warnings:					
Information:					
3	Fee Worksheet (PTO-875)	fee-info.pdf	29897 2b57e2c9719aaede86e20ed4a45cc29a0eeef3ea	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	136194		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
11258839 9	11258839	ORSINI ET AL.
	Examiner	**Art Unit**
	Samson B Lemma	2432

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	12/29/2009	04/15/2010	12/20/2010						
	1	÷	✓	✓						
	2	÷	✓	✓						
	3	÷	✓	✓						
	4	÷	✓	✓						
	5	÷	✓	✓						
	6	÷	✓	✓						
	7	÷	✓	✓						
	8	÷	✓	✓						
	9	÷	✓	✓						
	10	÷	✓	✓						
	11	÷	✓	✓						
	12	÷	✓	✓						
	13	÷	✓	✓						
	14	÷	✓	✓						
	15	÷	✓	✓						
	16	÷	✓	✓						
	17	÷	✓	✓						
	18	÷	✓	✓						
	19	÷	✓	✓						
	20	÷	✓	✓						
	21	÷	✓	✓						
	22	÷	✓	✓						
	23	÷	✓	✓						
	24	÷	✓	✓						
	25	÷	✓	✓						
	26	÷	✓	✓						
	27	÷	✓	✓						
	28	÷	✓	✓						
	29	÷	✓	✓						
	30	÷	✓	✓						
	31	÷	✓	✓						
	32	÷	✓	✓						
	33	÷	✓	✓						

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
11258839 9	11258839	ORSINI ET AL.
	Examiner	**Art Unit**
	Samson B Lemma	2432

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	12/29/2009	04/15/2010	12/20/2010						
	34	÷	✓	✓						
	35	÷	✓	✓						
	36	÷	✓	✓						
	37	÷	-	-						
	38	÷	-	-						
	39	÷	-	-						
	40	÷	-	-						
	41	÷	✓	✓						
	42	÷	✓	-						
	43	÷	✓	✓						
	44	÷	✓	✓						
	45	÷	✓	✓						
	46	÷	✓	✓						
	47	÷	✓	✓						
	48	÷	✓	✓						
	49	÷	✓	✓						
	50	÷	✓	✓						
	51	÷	✓	✓						
	52	÷	✓	✓						
	53	÷	✓	✓						
	54	÷	✓	✓						
	55	÷	-	-						
	56	÷	-	-						
	57	÷	-	-						
	58	÷	-	-						
	59	÷	-	-						
	60	÷	-	-						
	61	÷	-	-						
	62	÷	-	-						
	63	÷	-	-						
	64	÷	-	-						
	65	÷	-	-						
	66	÷	-	-						

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
11258839 9	11258839	ORSINI ET AL.
	Examiner	**Art Unit**
	Samson B Lemma	2432

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	12/29/2009	04/15/2010	12/20/2010						
	67	÷	-	-						
	68	÷	-	-						
	69	÷	✓	✓						
	70	÷	✓	✓						
	71	÷	✓	✓						
	72	÷	✓	✓						
	73	÷	✓	✓						
	74	÷	✓	✓						
	75	÷	✓	✓						
	76	÷	✓	✓						
	77	÷	✓	✓						
	78	÷	✓	✓						
	79	÷	✓	✓						
	80	÷	✓	✓						
	81	÷	✓	✓						
	82	÷	✓	✓						
	83	÷	✓	✓						
	84	÷	✓	✓						
	85	÷	✓	✓						

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
11258839	11258839	ORSINI ET AL.
	Examiner	**Art Unit**
	Samson B Lemma	2432

SEARCHED			
Class	**Subclass**	**Date**	**Examiner**
713	176, 186,156	12/20/2010	SL
380	33, 268, 270, 247, 286	12/20/2010	SL

SEARCH NOTES		
Search Notes	**Date**	**Examiner**
713/$, 380/$ (With text search)	12/20/2010	SL
EAST (See attached)	12/20/2010	SL
NPL (IEEE, ACM DIGITAL library, google, citeseer,...)	12/20/2010	SL
Inventor's name search	12/20/2010	SL

INTERFERENCE SEARCH			
Class	**Subclass**	**Date**	**Examiner**

Notice of References Cited	Application/Control No. 11/258,839	Applicant(s)/Patent Under Reexamination ORSINI ET AL.	
	Examiner SAMSON LEMMA	Art Unit 2432	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-5,910,987	06-1999	Ginter et al.	705/52
*	B	US-6,789,198	09-2004	Chan, Hark	713/193
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/258,839	10/25/2005	Rick L. Orsini	SFC-2	2420

1473 7590 01/06/2011

ROPES & GRAY LLP
PATENT DOCKETING 39/361
1211 AVENUE OF THE AMERICAS
NEW YORK, NY 10036-8704

EXAMINER
LEMMA, SAMSON B

ART UNIT	PAPER NUMBER
2432	

NOTIFICATION DATE	DELIVERY MODE
01/06/2011	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

USPatentMail@ropesgray.com
USPatentMail2@ropesgray.com

Office Action Summary	Application No.	Applicant(s)	
	11/258,839	ORSINI ET AL.	
	Examiner	Art Unit	
	SAMSON LEMMA	2432	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 October 2010*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-36,41,43-54 and 69-85* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-36,41,43-54 and 69-85* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This office action is in reply to an amendment filed on October 26, 2010. Claims 37-40, 42 and 55-68 are canceled. Thus claims 1-36, 41, 43-54 and 69-85 are pending of which claims 1, 18, 22, 26, 30, 31, 34, 35, 41, 47 and 69 are independent. Claims 1-3, 6-13, 17, 18, 22, 26, 30, 31, 34, 35, 41, 47 and 69-71 are amended.

2. The amendment made to independent claims 1, 18, 22, 26, 30, 31, 34, 35, 41 and 47 overcomes the 101 rejection set forth in the previous office action. Thus this particular 101 rejection is withdrawn.

Priority

3. This application claims priority from provisional application 60/622,146 filed on 10/25/2004. Therefore, the effective filling date for the subject matter defined in the pending claims of this application is October 25, 2004.

Response to Arguments

4. Applicant's arguments filed on 10/26/2010 have been fully considered but are moot in view of the new ground(s) of rejection.

Double Patenting

5. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re*

Longi, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).
A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).
Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

6. Claims 1-36, 41, 43-54 and 69-85 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over **claims 36-46** of the copending Application No. 12/148,365 (hereinafter refereed as **'365 application**). Although the conflicting claims are not identical, they are not patentably distinct from each other.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented

- **As per claims 1-36, 41, 43-54 and 69-85, claims 1-36, 41, 43-54 and 69-85** of the instant application and **claims 36-46, of '365 application** are not patentably distinct from each other because not only **the difference between the two set of claims merely is rephrasing of the claim languages, but also** all elements/limitations of the instant claims **1-36, 41, 43-54 and 69-85** are **recited by the elements/limitations of the claims 36-46 of the '365**

application. Thus claims 1-36, 41, 43-54 and 69-85 of the instant application would have been obvious over at least claims **36-46 of the '365 application** because all **elements/limitation of claims 1-36, 41, 43-54 and 69-85 of the instant application is anticipated by the claims 36-46 of the '365 application.**

7. Claims 1-36, 41, 43-54 and 69-85 are also rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over **claims 50-57** of the patent, 7,391,865 (hereinafter refereed as **'865 patent**). Although the conflicting claims are not identical, they are not patentably distinct from each other.

- **As per claims 1-36, 41, 43-54 and 69-85, claims 1-36, 41, 43-54 and 69-85** of the instant application and **claims 50-57, of '865 patent** are not patentably distinct from each other because not only **the difference between the two set of claims merely is rephrasing of the claim languages, but also** all elements/limitations of the instant claims **1-36, 41, 43-54 and 69-85** are **recited by the elements/limitations of the claims 50-57 of the '865 patent. Thus claims 1-36, 41, 43-54 and 69-85 of the instant application** would have been obvious over at least claims **50-57 of the '865 application** because all **elements/limitation of claims 1-36, 41, 43-54 and 69-85 of the instant application is anticipated by the claims 50-57 of the '865 patent.**

Claim Objections

8. Independent claim 1 is objected to because of the following informalities:

On line 12, the claim recites the following limitation,...portions of the data are "communicated". It is not clear whether or not this is trying to indicate the feature of the amended limitation recited as "streaming" on line 9. Thus in order to avoid this ambiguity examiner suggests replacing the limitation, "communicated" recited on line 12 with the limitation "streamed".

Appropriate correction is required.

Claim Rejections - 35 USC § 112

9. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

10. Independent claim 1 on [line 17], just before the last line, recites the following limitation, "the at least two streaming portions of data". There is insufficient antecedent basis for this limitation in the claim. Furthermore, it is not only ambiguous but also not clear whether or not this limitation is referring to or is

one and the same with the limitation, "data from at least a subset of the at least two portions of the data" recited on line 15-16.

Appropriate correction is required.

11. Dependent claims 2-17 fully incorporating the deficiencies of the above independent claim 1 are likewise rejected.

12. Independent claim 69 on lines 15-16, recites the following limitation, a subset of "the at least two streaming portions of data". There is insufficient antecedent basis for this limitation in the claim.

Appropriate correction is required.

13. Dependent claims 70-85 fully incorporating the deficiencies of the above independent claim 69 are likewise rejected.

Conclusion

14. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. (See PTO-Form 892).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Samson B Lemma whose telephone number is 571-272-3806. The examiner can normally be reached on Monday-Friday (8:00 am---4: 30 pm).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, BARRON JR GILBERTO can be reached on 571-272-3799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov.

Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Samson B Lemma/

Examiner, Art Unit 2432

REMARKS

Summary of Office Action

Claims 1-36, 41, 43-54 and 69-85 are pending in the above-identified patent application.

In the Office Action of January 6, 2011, claims 1-36, 41, 43-54 and 69-85 were provisionally rejected for obviousness-type double patenting in view of co-pending U.S. Patent Application No. 12/148,365. Claims 1-36, 41, 43-54 and 69-85 were rejected for obviousness-type double patenting in view of U.S. Patent 7,391,865. Claim 1 was objected to because of informalities. Claims 1-17 and 69-85 were rejected under 35 U.S.C. § 112, second paragraph, as lacking sufficient antecedent basis.

Applicants respectfully traverse these rejections.

Summary of Applicants' Reply

Applicants have amended claims 1, 3, and 69. No new matter is added.

Response to Double Patenting Rejections

Applicants submit herewith two Terminal Disclaimers under 37 C.F.R. § 1.321(b,c), disclaiming the terminal part of the term of any patent issued from the above-identified, present patent application that would extend beyond the expiration date of either U.S. Patent No. 7,391,865 (currently September 20, 2020*) or any patent to issue from U.S. Patent Application No. 12/148,365, and requiring that any patent to issue from the present patent

* It is Applicants' and Assignee's intention that if any change in the patent statutes delays the expiration date of U.S. Patent No. 7,391,865 or any patent to issue from U.S. Patent Application No. 12/148,365, or if any of these patents is hereafter awarded patent term adjustment under 35 U.S.C. § 154(b), then the date beyond which the term of any patent to issue on this application is disclaimed shall change along with the expiration date of the patent whose term was adjusted.

application remain commonly-owned with U.S. Patent No. 7,391,865 and any patent to issue from U.S. Patent Application No. 12/148,365. Accordingly, Applicants respectfully request that the double-patenting rejections be withdrawn.

Response to Objection and Rejection under 35 U.S.C. § 112

Applicants believe that the present amendments address the Examiner's formalities objection and rejections under 35 U.S.C. § 112, and respectfully request withdrawal of the objection and rejections.

Conclusion

Applicants respectfully submit that this application, as amended, is in condition for allowance. Reconsideration and prompt allowance of this application are respectfully requested.

Respectfully submitted,

/Laura Zager/

Laura Zager
Reg. No. 64,813
Attorney for Applicants
ROPES & GRAY LLP
Customer No. 1473
Tel.: (617) 951-7069

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION

Applicants	:	Rick Orsini et al.
Application No.	:	11/258,839 Confirmation No. : 2420
Filed	:	October 25, 2005
For	:	SECURE DATA PARSER METHOD AND SYSTEM
Art Unit	:	2432
Examiner	:	Samson B. Lemma

Boston, Massachusetts
June 30, 2011

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

REPLY TO OFFICE ACTION

Sir:

In reply to the Office Action dated January 6, 2011, Applicants hereby amend the above-identified patent application as follows. The time for response is extended to July 6, 2011, by the accompanying petition for a three-month extension of time.

Amendments to the Claims are reflected in the listing beginning on page 2 of this Reply.

Remarks begin on page 16 of this Reply.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
8 May 2008 (08.05.2008)

PCT

(10) International Publication Number
WO 2008/054406 A2

(51) International Patent Classification:
H04L 9/28 (2006.01)

(21) International Application Number:
PCT/US2006/045066

(22) International Filing Date:
30 November 2006 (30.11.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/738,231 18 November 2005 (18.11.2005) US

(71) Applicants and
(72) Inventors: **ORSINI, Rick L.** [US/US]; 2100 Kings Forest Lane, Flower Mound, TX 75028 (US). **O'HARE, Mark S.** [US/US]; 8 Kennedy Court, Coto De Caza, CA 92679 (US). **DAVENPORT, Roger** [US/US]; 3204 Cr 4119, Campbell, TX 75422 (US). **WINICK, Steven** [US/US]; 3 Expressway Plaza, Roslyn Heights, NY 11577 (US).

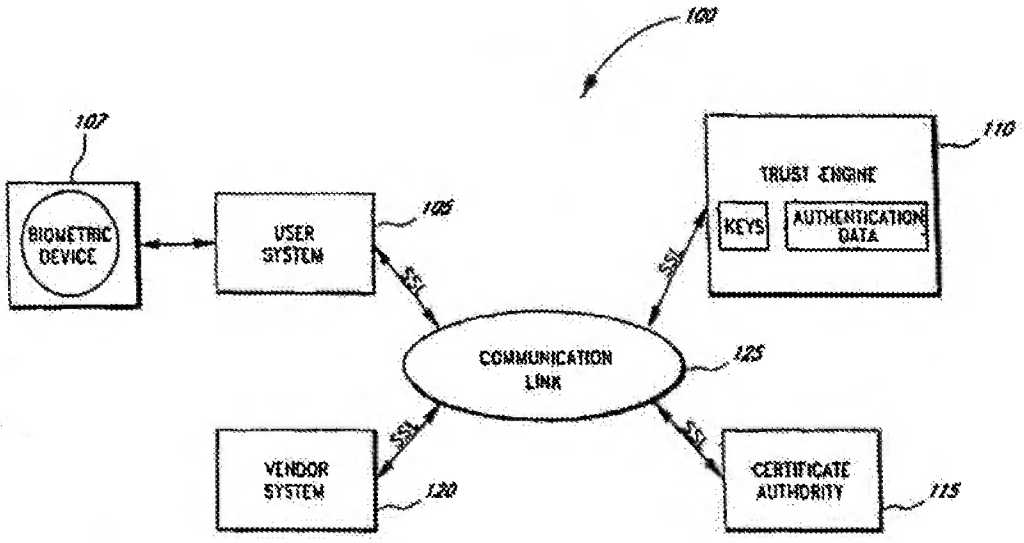
(74) Agents: **INGERMAN, Jeffrey H.** et al.; Fish & Neave IP Group, Ropes & Gray LLP, 1251 Avenue of the Americas, New York, NY 10020 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

(54) Title: SECURE DATA PARSER METHOD AND SYSTEM



(57) Abstract: A secure data parser is provided that may be integrated into any suitable system for securely storing and communicating data. The secure data parser parses data and then splits the data into multiple portions that are stored or communicated distinctly. Encryption of the original data, the portions of data, or both may be employed for additional security. The secure data parser may be used to protect data in motion by splitting original data into portions of data, that may be communicated using multiple communications paths.

WO 2008/054406 A2

SECURE DATA PARSER METHOD AND SYSTEM

Cross-reference to Related Applications

[0001] This application claims the benefit of U.S. provisional application No. 60/738,231, filed on November 18, 2005, which is hereby incorporated by reference herein in its entirety.

Field of the Invention

[0002] The present invention relates in general to a system for securing data from unauthorized access or use.

Background of the Invention

[0003] In today's society, individuals and businesses conduct an ever-increasing amount of activities on and over computer systems. These computer systems, including proprietary and non-proprietary computer networks, are often storing, archiving, and transmitting all types of sensitive information. Thus, an ever-increasing need exists for ensuring data stored and transmitted over these systems cannot be read or otherwise compromised.

[0004] One common solution for securing computer systems is to provide login and password functionality. However, password management has proven to be quite costly with a large percentage of help desk calls

EX1010, Page 2040

relating to password issues. Moreover, passwords provide little security in that they are generally stored in a file susceptible to inappropriate access, through, for example, brute-force attacks.

[0005] Another solution for securing computer systems is to provide cryptographic infrastructures. Cryptography, in general, refers to protecting data by transforming, or encrypting, it into an unreadable format. Only those who possess the key(s) to the encryption can decrypt the data into a useable format. Cryptography is used to identify users, e.g., authentication, to allow access privileges, e.g., authorization, to create digital certificates and signatures, and the like. One popular cryptography system is a public key system that uses two keys, a public key known to everyone and a private key known only to the individual or business owner thereof. Generally, the data encrypted with one key is decrypted with the other and neither key is recreatable from the other.

[0006] Unfortunately, even the foregoing typical public-key cryptographic systems are still highly reliant on the user for security. For example, cryptographic systems issue the private key to the user, for example, through the user's browser. Unsophisticated users then generally store the private key on a hard drive accessible to others through an open computer system, such as, for example, the Internet. On the other hand, users may choose poor names for files containing their private key, such as, for example, "key." The result of the foregoing and other acts is to allow the key or keys to be susceptible to compromise.

EX1010, Page 2041

[0007] In addition to the foregoing compromises, a user may save his or her private key on a computer system configured with an archiving or backup system, potentially resulting in copies of the private key traveling through multiple computer storage devices or other systems. This security breach is often referred to as "key migration." Similar to key migration, many applications provide access to a user's private key through, at most, simple login and password access. As mentioned in the foregoing, login and password access often does not provide adequate security.

[0008] One solution for increasing the security of the foregoing cryptographic systems is to include biometrics as part of the authentication or authorization. Biometrics generally include measurable physical characteristics, such as, for example, finger prints or speech that can be checked by an automated system, such as, for example, pattern matching or recognition of finger print patterns or speech patterns. In such systems, a user's biometric and/or keys may be stored on mobile computing devices, such as, for example, a smartcard, laptop, personal digital assistant, or mobile phone, thereby allowing the biometric or keys to be usable in a mobile environment.

[0009] The foregoing mobile biometric cryptographic system still suffers from a variety of drawbacks. For example, the mobile user may lose or break the smartcard or portable computing device, thereby having his or her access to potentially important data entirely cut-off. Alternatively, a malicious person may steal the mobile user's smartcard or portable computing device and use it to effectively steal the mobile user's digital credentials. On the other hand,

the portable-computing device may be connected to an open system, such as the Internet, and, like passwords, the file where the biometric is stored may be susceptible to compromise through user inattentiveness to security or malicious intruders.

Summary of the Invention

[0010] Based on the foregoing, a need exists to provide a cryptographic system whose security is user-independent while still supporting mobile users.

[0011] Accordingly, one aspect of the present invention is to provide a method for securing virtually any type of data from unauthorized access or use. The method comprises one or more steps of parsing, splitting and/or separating the data to be secured into two or more parts or portions. The method also comprises encrypting the data to be secured. Encryption of the data may be performed prior to or after the first parsing, splitting and/or separating of the data. In addition, the encrypting step may be repeated for one or more portions of the data. Similarly, the parsing, splitting and/or separating steps may be repeated for one or more portions of the data. The method also optionally comprises storing the parsed, split and/or separated data that has been encrypted in one location or in multiple locations. This method also optionally comprises reconstituting or re-assembling the secured data into its original form for authorized access or use. This method may be incorporated into the operations of any computer, server, engine or the like, that is capable of executing the desired steps of the method.

[0012] Another aspect of the present invention provides a system for securing virtually any type of data from unauthorized access or use. This system comprises a data splitting module, a cryptographic handling module, and, optionally, a data assembly module. The system may, in one embodiment, further comprise one or more data storage facilities where secure data may be stored.

[0013] Accordingly, one aspect of the invention is to provide a secure server, or trust engine, having server-centric keys, or in other words, storing cryptographic keys and user authentication data on a server. According to this embodiment, a user accesses the trust engine in order to perform authentication and cryptographic functions, such as, but not limited to, for example, authentication, authorization, digital signing and generation, storage, and retrieval of certificates, encryption, notary-like and power-of-attorney-like actions, and the like.

[0014] Another aspect of the invention is to provide a reliable, or trusted, authentication process. Moreover, subsequent to a trustworthy positive authentication, a wide number of differing actions may be taken, from providing cryptographic technology, to system or device authorization and access, to permitting use or control of one or a wide number of electronic devices.

[0015] Another aspect of the invention is to provide cryptographic keys and authentication data in an environment where they are not lost, stolen, or compromised, thereby advantageously avoiding a need to continually reissue and manage new keys and authentication data. According to another aspect of

the invention, the trust engine allows a user to use one key pair for multiple activities, vendors, and/or authentication requests. According to yet another aspect of the invention, the trust engine performs at least one step of cryptographic processing, such as, but not limited to, encrypting, authenticating, or signing, on the server side, thereby allowing clients or users to possess only minimal computing resources.

[0016] According to yet another aspect of the invention, the trust engine includes one or multiple depositories for storing portions of each cryptographic key and authentication data. The portions are created through a data splitting process that prohibits reconstruction without a predetermined portion from more than one location in one depository or from multiple depositories. According to another embodiment, the multiple depositories may be geographically remote such that a rogue employee or otherwise compromised system at one depository will not provide access to a user's key or authentication data.

[0017] According to yet another embodiment, the authentication process advantageously allows the trust engine to process multiple authentication activities in parallel. According to yet another embodiment, the trust engine may advantageously track failed access attempts and thereby limit the number of times malicious intruders may attempt to subvert the system.

[0018] According to yet another embodiment, the trust engine may include multiple instantiations where each trust engine may predict and share processing loads with the others. According to yet another embodiment, the trust engine may include a redundancy module for polling a plurality of authentication

results to ensure that more than one system authenticates the user.

[0019] Therefore, one aspect of the invention includes a secure cryptographic system, which may be remotely accessible, for storing data of any type, including, but not limited to, a plurality of private cryptographic keys to be associated with a plurality of users. The cryptographic system associates each of the plurality of users with one or more different keys from the plurality of private cryptographic keys and performs cryptographic functions for each user using the associated one or more different keys without releasing the plurality of private cryptographic keys to the users. The cryptographic system comprises a depository system having at least one server which stores the data to be secured, such as a plurality of private cryptographic keys and a plurality of enrollment authentication data. Each enrollment authentication data identifies one of multiple users and each of the multiple users is associated with one or more different keys from the plurality of private cryptographic keys. The cryptographic system also may comprise an authentication engine which compares authentication data received by one of the multiple users to enrollment authentication data corresponding to the one of multiple users and received from the depository system, thereby producing an authentication result. The cryptographic system also may comprise a cryptographic engine which, when the authentication result indicates proper identification of the one of the multiple users, performs cryptographic functions on behalf of the one of the multiple users using the associated one or more different keys received from the

depository system. The cryptographic system also may comprise a transaction engine connected to route data from the multiple users to the depository server system, the authentication engine, and the cryptographic engine.

[0020] Another aspect of the invention includes a secure cryptographic system that is optionally remotely accessible. The cryptographic system comprises a depository system having at least one server which stores at least one private key and any other data, such as, but not limited to, a plurality of enrollment authentication data, wherein each enrollment authentication data identifies one of possibly multiple users. The cryptographic system may also optionally comprise an authentication engine which compares authentication data received by users to enrollment authentication data corresponding to the user and received from the depository system, thereby producing an authentication result. The cryptographic system also comprises a cryptographic engine which, when the authentication result indicates proper identification of the user, performs cryptographic functions on behalf of the user using at least said private key, which may be received from the depository system. The cryptographic system may also optionally comprise a transaction engine connected to route data from the users to other engines or systems such as, but not limited to, the depository server system, the authentication engine, and the cryptographic engine.

[0021] Another aspect of the invention includes a method of facilitating cryptographic functions. The method comprises associating a user from multiple users with one or more keys from a plurality of private

cryptographic keys stored on a secure location, such as a secure server. The method also comprises receiving authentication data from the user, and comparing the authentication data to authentication data corresponding to the user, thereby verifying the identity of the user. The method also comprises utilizing the one or more keys to perform cryptographic functions without releasing the one or more keys to the user.

[0022] Another aspect of the invention includes an authentication system for uniquely identifying a user through secure storage of the user's enrollment authentication data. The authentication system comprises one or more data storage facilities, wherein each data storage facility includes a computer accessible storage medium which stores at least one of portions of enrollment authentication data. The authentication system also comprises an authentication engine which communicates with the data storage facility or facilities. The authentication engine comprises a data splitting module which operates on the enrollment authentication data to create portions, a data assembling module which processes the portions from at least one of the data storage facilities to assemble the enrollment authentication data, and a data comparator module which receives current authentication data from a user and compares the current authentication data with the assembled enrollment authentication data to determine whether the user has been uniquely identified.

[0023] Another aspect of the invention includes a cryptographic system. The cryptographic system comprises one or more data storage facilities, wherein

each data storage facility includes a computer accessible storage medium which stores at least one portion of one ore more cryptographic keys. The cryptographic system also comprises a cryptographic engine which communicates with the data storage facilities. The cryptographic engine also comprises a data splitting module which operate on the cryptographic keys to create portions, a data assembling module which processes the portions from at least one of the data storage facilities to assemble the cryptographic keys, and a cryptographic handling module which receives the assembled cryptographic keys and performs cryptographic functions therewith.

[0024] Another aspect of the invention includes a method of storing any type of data, including, but not limited to, authentication data in geographically remote secure data storage facilities thereby protecting the data against composition of any individual data storage facility. The method comprises receiving data at a trust engine, combining at the trust engine the data with a first substantially random value to form a first combined value, and combining the data with a second substantially random value to form a second combined value. The method comprises creating a first pairing of the first substantially random value with the second combined value, creating a second pairing of the first substantially random value with the second substantially random value, and storing the first pairing in a first secure data storage facility. The method comprises storing the second pairing in a second secure data storage facility remote from the first secure data storage facility.

EX1010, Page 2049

[0025] Another aspect of the invention includes a method of storing any type of data, including, but not limited to, authentication data comprising receiving data, combining the data with a first set of bits to form a second set of bits, and combining the data with a third set of bits to form a fourth set of bits. The method also comprises creating a first pairing of the first set of bits with the third set of bits. The method also comprises creating a second pairing of the first set of bits with the fourth set of bits, and storing one of the first and second pairings in a first computer accessible storage medium. The method also comprises storing the other of the first and second pairings in a second computer accessible storage medium.

[0026] Another aspect of the invention includes a method of storing cryptographic data in geographically remote secure data storage facilities thereby protecting the cryptographic data against comprise of any individual data storage facility. The method comprises receiving cryptographic data at a trust engine, combining at the trust engine the cryptographic data with a first substantially random value to form a first combined value, and combining the cryptographic data with a second substantially random value to form a second combined value. The method also comprises creating a first pairing of the first substantially random value with the second combined value, creating a second pairing of the first substantially random value with the second substantially random value, and storing the first pairing in a first secure data storage facility. The method also comprises storing the second

EX1010, Page 2050