



US006341164B1

(12) **United States Patent**
Dilkie et al.

(10) **Patent No.:** **US 6,341,164 B1**
(45) **Date of Patent:** **Jan. 22, 2002**

(54) **METHOD AND APPARATUS FOR CORRECTING IMPROPER ENCRYPTION AND/OR FOR REDUCING MEMORY STORAGE**

(75) Inventors: **Lee Dilkie; Paul C. Van Oorschot,**
both of Ottawa (CA)

(73) Assignee: **Entrust Technologies Limited,** Ottawa
(CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/120,716**

(22) Filed: **Jul. 22, 1998**

(51) **Int. Cl.⁷** **H04K 9/00**

(52) **U.S. Cl.** **380/278**

(58) **Field of Search** 713/153; 380/278

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,228,083 A * 7/1993 Lozowick et al. 380/9
5,680,551 A * 10/1997 Martino 395/200

OTHER PUBLICATIONS

Bruce Schneier, Applied Cryptology, pp. 40, 461, 473–474, 476–478, 480–481, Oct. 1995.*

* cited by examiner

Primary Examiner—Gail Hayes

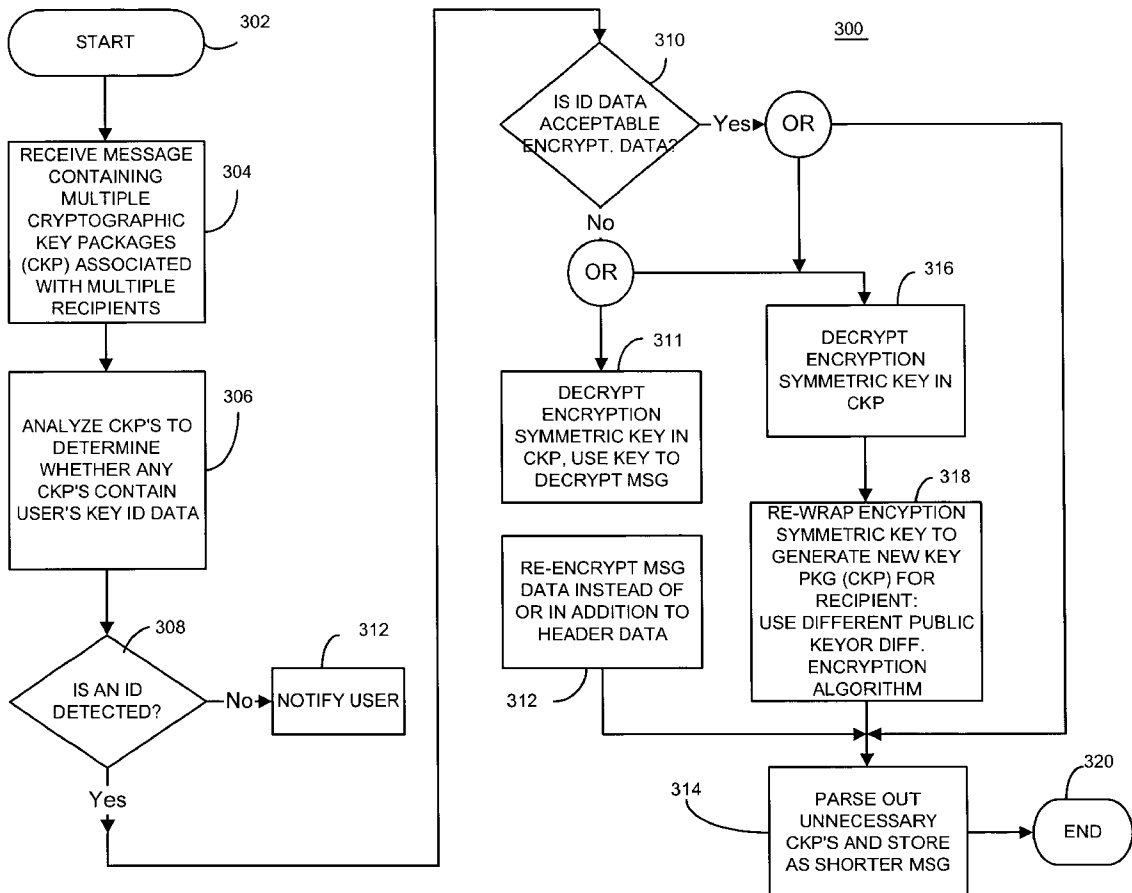
Assistant Examiner—James Seal

(74) *Attorney, Agent, or Firm*—Vedder, Price, Kaufman & Kammholz

(57) **ABSTRACT**

An apparatus and methods for facilitating an encryption process for use in systems employing cryptography based security, removes unnecessary data relating to encryption keys prior to storing the data after receipt of the encrypted information from a sender. Encrypted data, such as message data for multiple recipients, is analyzed to determine whether encryption related data for other recipients may be removed and/or whether a preferred encrypting process was used. In one embodiment, the apparatus and method also determines whether a non-preferred encryption process was used to encrypt encrypted data and re-encrypts the encrypted data with a different encryption process in response to detected non-preferred encryption key usage.

38 Claims, 4 Drawing Sheets



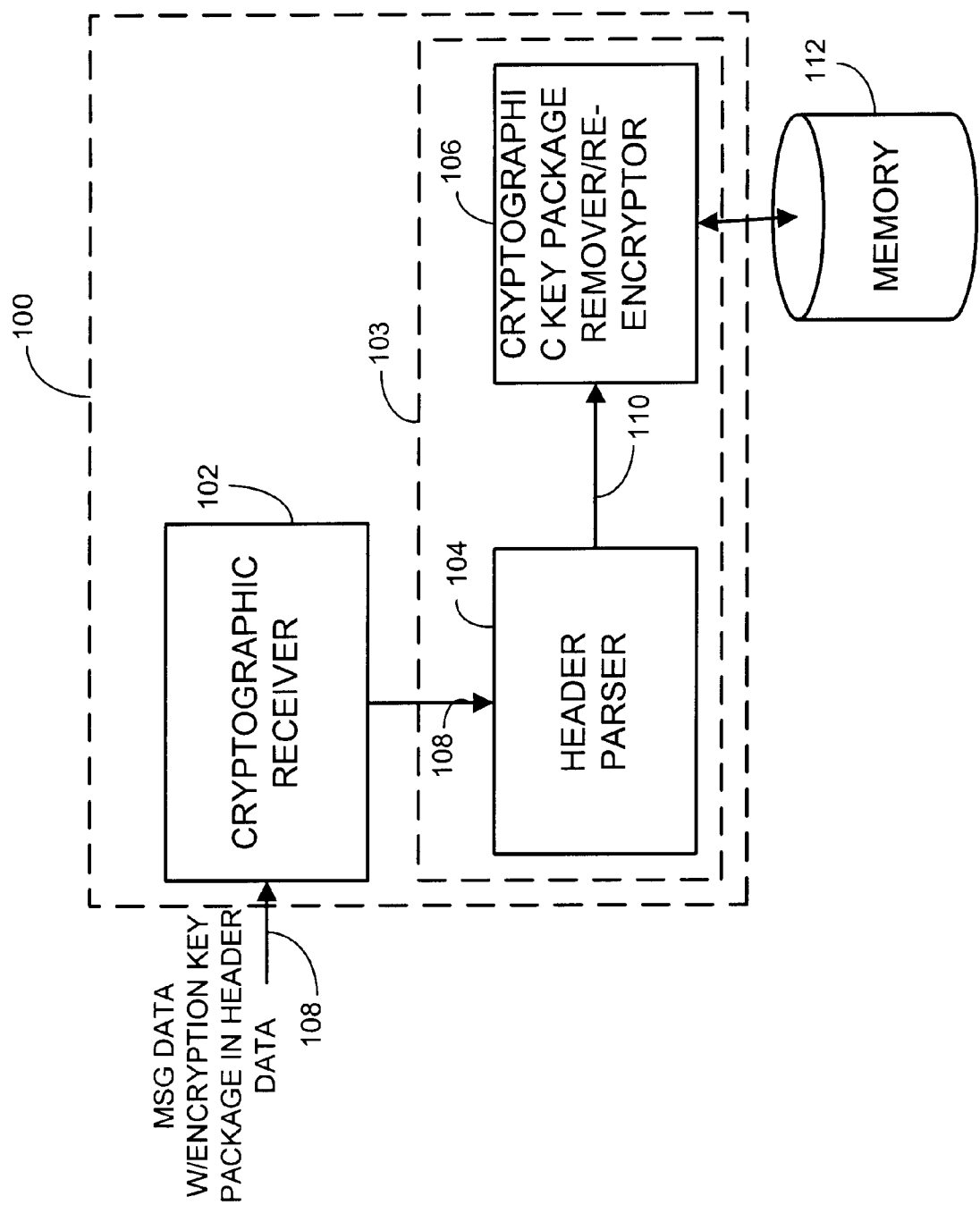


FIG. 1

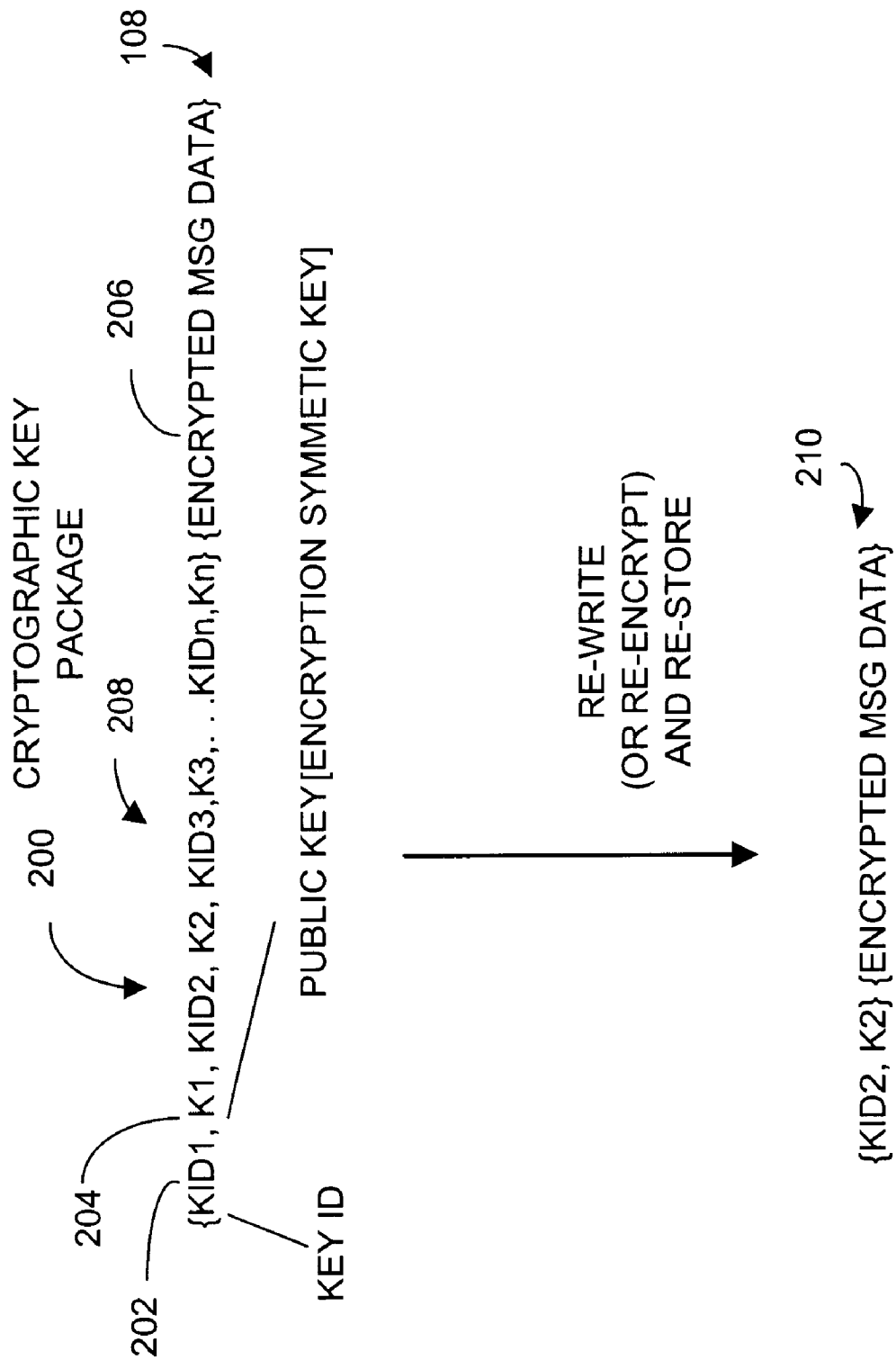


FIG. 2

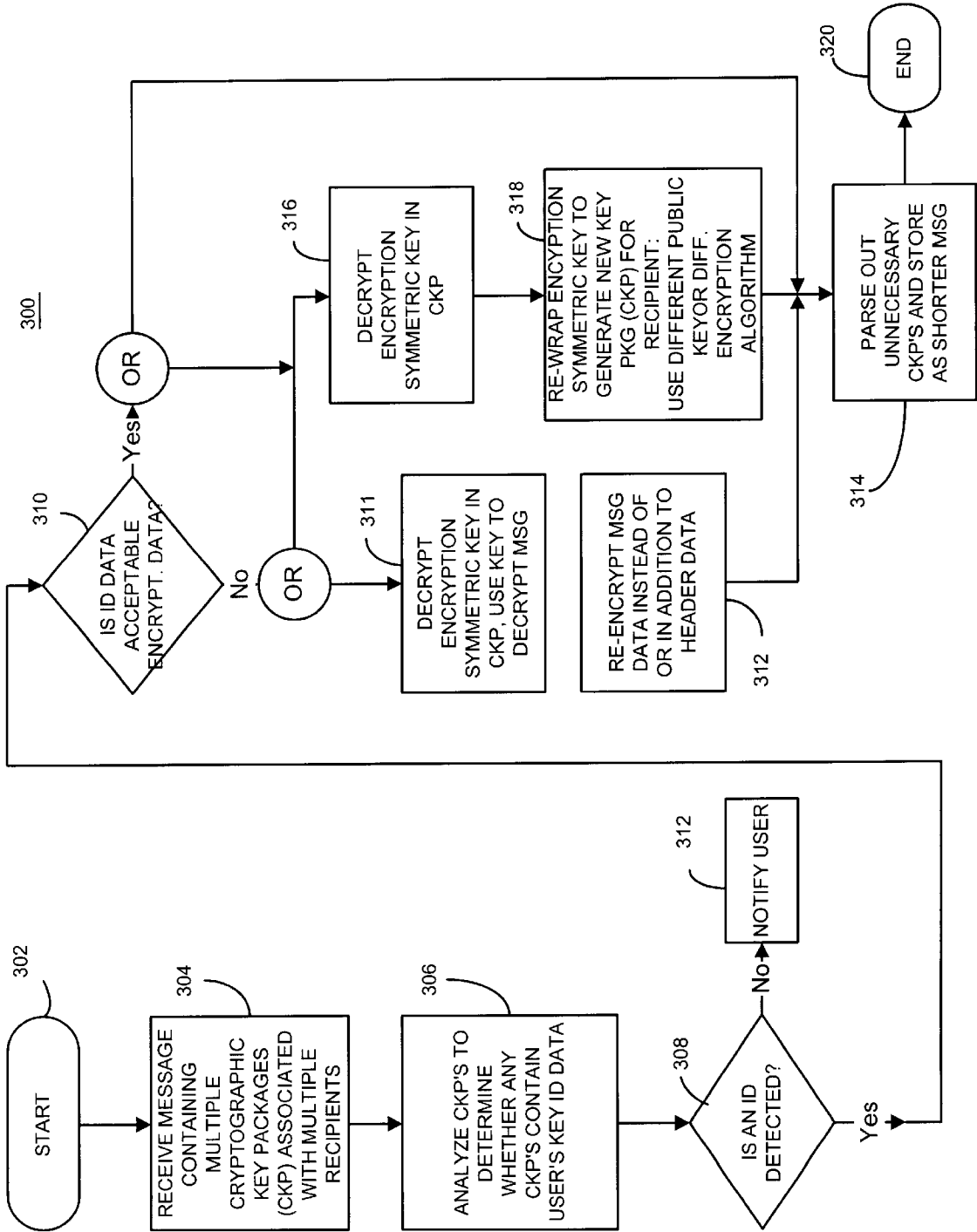
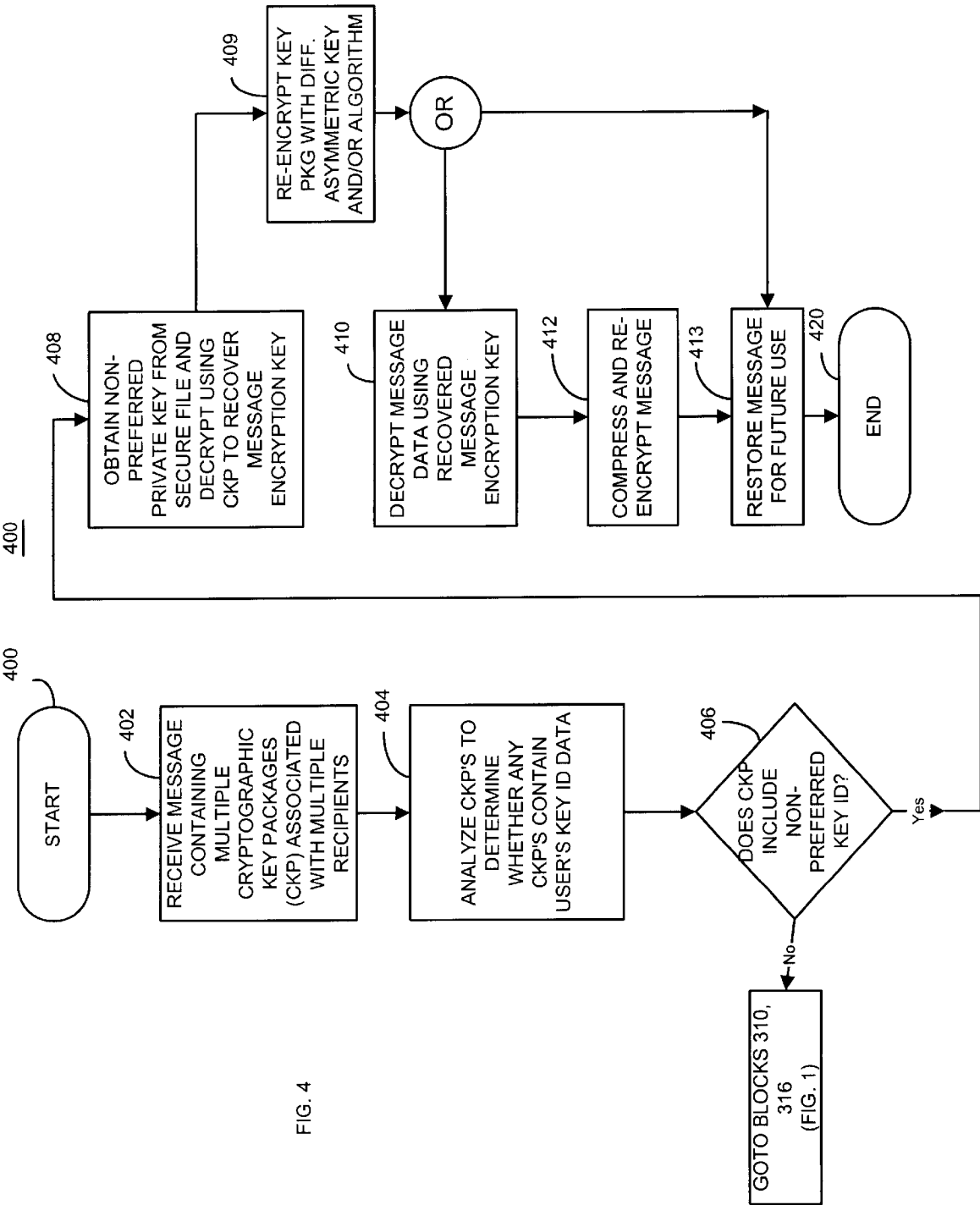


FIG. 3



**METHOD AND APPARATUS FOR
CORRECTING IMPROPER ENCRYPTION
AND/OR FOR REDUCING MEMORY
STORAGE**

BACKGROUND OF THE INVENTION

The invention relates generally to systems employing cryptography based security, and more particularly to systems employing public key encryption.

Communication systems such as computer networks, telecommunication systems, and other systems are increasingly using cryptography to encrypt information sent electronically. With the increase in electronic commerce, electronic mail communication, and other information that subscribers may wish to keep secret, public key cryptography systems employ an encryption key pair, such as a decryption private key and an encryption public key to decrypt and encrypt data. The decryption private keys are secret keys that are protected by the use of encryption techniques and other techniques to help ensure that an attacker cannot readily obtain this critical key.

Also, communication systems are known that additionally employ digital signing key pairs, such as a signing private key and a signing public key to allow the use of digital signatures so that e-mail, electronic contracts and other data may be uniquely and securely signed by an individual. Such systems are sometimes referred to as dual public key systems. Some dual public key systems generate and use a separate digital signature certificate and a separate encryption certificate that is sent with encrypted data or digitally signed data. The separate certificates contain the respective public key data along with certificate expiry data, and other data, as known in the art. Other dual public key systems use a single certificate that contains both the signing public key and the encryption public key.

When a subscriber's certificates expire, the cryptographic system typically generates new certificates. For encryption certificates, it is desirable to keep a history of all decryption private keys so that a subscriber can decrypt old e-mail messages even though the encryption certificates have expired.

Some standards exist that set forth particular requirements so that dual and single certificate systems may communicate with one other, such as the revised versions of the Secure Multipurpose Internet Mail Extensions (S/MIME) standard. With such standards, header data is included in an encrypted message that typically includes certificate key usage data indicating the data to use in the attached certificate, and whether the attached certificates are dual certificates or a single certificate.

However, an interoperability problem can still arise when a dual certificate system and a single certificate system communicate with one another if a sending unit or receiving subscriber unit misinterprets or ignores the key usage data. For example, a single certificate sending unit, sending data to a dual certificate subscriber unit, may encrypt data using the subscriber's public signing key since it may use the first public key listed in its single certificate which may be the digital signing certificate. For example, this may happen with RSA algorithm keys, which may be used for both encryption and signatures. The message is encrypted using the signing key data. However, for encrypting data, the single certificate system should have used the public encryption key from the encryption certificate. When a receiving unit receives an encrypted message, the unit typically searches a user profile containing the securely stored match-

ing private decryption key to decrypt the message. The matching key is typically found by using a key identifier or by other mechanisms such as inclusion of the public key certificate in the message header. However, when the signing certificate data is used for encryption, a receiving subscriber who expects an encryption key to have been used is typically unable to locate the proper or matching private decryption key. Hence the message cannot be decrypted by the receiving unit. Hence when some systems send an encrypted message, the system does not send the encryption key identifier or certificate to allow decryption by a receiving user, but instead sends the signing key identifier or certificate.

A system that sends or encrypts a message using a signing public key from the signing certificate and sends the signing key identifier or certificate to a receiving unit prevents secure communication among users expecting to communicate. Hence conventional cryptographic systems may not be capable of communicating with systems that may improperly encrypt message data. Accordingly, it would be desirable to have a cryptography based system that facilitates detection of an improper encryption process yet allows suitable decryption so that a user may still communicate with a sender that is improperly encrypting information.

Also, where a sender may wish to send an encrypted message to one hundred recipients, typically an encryption key package is sent for each recipient as part of message header data. An encryption key package may include, for example, key identification data, such as a certificate serial number or unique key identifier data, along with an encrypted public encryption key separately for each of the one hundred recipients. One copy of the encrypted message data is also sent with the encryption key packages so that a recipient typically needs to store all one hundred key identification data entries corresponding encrypted public encryption key for each recipient. This results in the unnecessary use of memory since each recipient typically stores data associated with the other ninety nine recipients.

Typically when a recipient receives an encrypted message, the message is decrypted and read and then restored in an encrypted manner so that an adversary cannot readily obtain the information off of a memory storage unit such as a computer's hard drive. The required storage increases dramatically when a recipient receives hundreds of e-mail messages on a daily basis. Accordingly, it would be desirable to have a system that facilitates memory storage reduction where message data includes encryption key packages associated with multiple recipients.

Also, some cryptographic systems are known that may re-encrypt entire messages to facilitate a higher level of security, or to conform with security policy data at a recipient's site. For example a system employing a single DES algorithm may also employ a triple DES algorithm to provide additional security through the use of a longer encryption key. However, the storage of the entire message requires a substantial amount of storage space in a receiving unit's memory whether or not the message is re-encrypted.

Consequently, there exists a need for a system that facilitates detection of improper encryption and can correct the improper encryption and/or a system that facilitates a reduction of memory storage for encrypted data for public key cryptography based security systems and other data security systems.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram generally depicting a system for facilitating an encryption process in accordance with one embodiment of the invention.

3

FIG. 2 is a diagram illustrating header data reduction to facilitate memory storage reduction in accordance with one embodiment of the invention.

FIG. 3 is a flowchart illustrating an example of one embodiment of facilitating an encryption process in accordance with the invention.

FIG. 4 is a flowchart illustrating another example of an embodiment of memory storage reduction in accordance with the invention.

DETAILED DESCRIPTION OF THE INVENTION

An apparatus and methods for facilitating an encryption process for use in systems employing cryptography based security, removes unnecessary data relating to encryption keys prior to storing the data after receipt of the encrypted information from a sender. Encrypted data, such as message data for multiple recipients, is analyzed to determine whether encryption related data for other recipients may be removed and/or whether a proper encrypting process was used.

In one embodiment, the apparatus and one method determines whether an improper or undesirable encryption key was used to encrypt encrypted data and re-encrypts the encrypted data with a different encryption process in response to detected improper encryption key usage. As one example, involving the RSA algorithm, received cryptographic key packages are analyzed to determine whether digital signing key information was used to encrypt the message.

In another embodiment, the apparatus and method receives header data containing cryptographic key packages associated with multiple recipients, then re-encrypts ("re-wraps") selected header data that corresponds to a predetermined recipient and stores the re-encrypted message data for later use by the recipient. In another embodiment, no new encryption is performed, but a deletion of unnecessary header data is performed by parsing the header data to determine whether a cryptographic key package corresponds to key identification data associated with a predetermined recipient. Cryptographic key packages not corresponding to the key identification data for that recipient are removed so that the pertinent cryptographic key package data is stored with corresponding encrypted message data for the recipient for later use.

FIG. 1 shows an example of one embodiment of a memory storage reduction apparatus 100 having a cryptographic receiver 102, an encrypted message data analyzer 103 having a header data parser 104, and a cryptographic key package removal stage 106. The apparatus 100 may be a suitably programmed computer that receives encrypted message data with cryptographic key packages 108. The apparatus 100 may be in each communication unit in a communication system, a subset of units, a gateway computer to which recipients are connected, or any other suitable device. The message data with cryptographic key packages 108 may be an encrypted message, containing any suitable information, sent by a sender to many recipients, such as an e-mail message over a worldwide computer network or an internal company computer network or over any other suitable communication system. The cryptographic key package information is preferably contained as header data, that includes a plurality of encryption key packages associated with multiple recipients. Header data as referred to herein is not limited to data preceding other data, but may be located at any location in a data stream or storage unit as

4

long as it is associated with some data. A cryptographic key package may include, for example, a symmetric encryption key wrapped, or encrypted, with an asymmetric encryption key, such as a recipient's public key, as known in the art (this is represented as K1, K2 . . . Kn in FIG. 2). However, any suitable encryption information may be used.

The cryptographic receiver 102, receives the encrypted message data with the header data and if desired, temporarily stores this data. The cryptographic receiver 102 communicates the received message data with cryptographic key package data 108 to encrypted message data analyzer 103.

Depending upon how the encrypted message data analyzer 103 is configured, the analyzer 103 may remove detected unnecessary cryptographic key package data, detect whether an improper encryption key was used, re-encrypt message data using a preferred encryption key or any suitable combination of these tasks. The header data parser 104 parses the received cryptographic key packages into separate key packages by standard techniques, such as relying on known data formatting templates and/or field identifiers or markers to distinguish separate packages. The header data parser 104 then communicates the parsed information 110 to the cryptographic key package removal stage 106.

The cryptographic key package removal stage 106 analyzes each cryptographic key package to determine whether at least one of the cryptographic key packages associated with the message data corresponds to key identification data for a predetermined recipient. The cryptographic key package removal stage 106 removes the cryptographic key packages that do not correspond to the key identification data associated with the predetermined recipient. The resulting data is preferably only data representing the cryptographic key package (or packages) associated with the predetermined recipient based on the encryption identification data along with the corresponding encrypted message data which is then re-stored in memory 112 for later use.

FIG. 2 shows by way of example, data that is received by the cryptographic receiver 102 such as message data with cryptographic key packages 200 in the header 108. Key identification data 202 (KID1-KIDn) is included for each recipient of the message. In a public key cryptography based system, this may be for example the serial number of a public key certificate containing the public key used to create the key package, a hashed version of such a public key, a unique key identifier, or other information which may be used to associate a key package with a particular end-user, end-user public key, or public key certificate. Also, corresponding key data 204 (K1-Kn) is included with the key identification data 202 in the cryptographic key package 200. The cryptographic key packages 200 are sent with the encrypted message data 206, as known in the art. The memory storage reduction apparatus 100 in one embodiment, parses the header data containing the cryptographic key packages 200 and removes the cryptographic key package data that does not correspond to key identification data associated with a recipient, or memory storage reduction apparatus analyzing the received information. For example, if a recipient is assigned key data K2, such as encryption public key data, which is given key identification data KID2, all other cryptographic key packages are removed prior to being re-stored in memory 112. The re-stored information by the cryptographic key package removal stage 106 is represented generally as stored data 210. Hence if a recipient is one of one hundred recipients to an encrypted e-mail message or other encrypted data, upon receipt of the e-mail message, the memory storage reduction

5

apparatus **100** may delete ninety nine of the one hundred cryptographic key packages (wrapped keys **K1** and **K3-K100** and associated KID data) in the header saving only the wrapped key, **K2** and **KID2**, for the one recipient before restoring the wrapped key with the encrypted message for future use.

Referring back to FIG. 1, the header data parser **104**, if desired, also analyzes the header data **108** to determine whether any of the cryptographic key packages includes key identification data corresponding to a signature key for a given recipient. As such, the system detects for conditions where an improper encryption key was used for example, in the case of the RSA algorithm for where a signing key or signing identification data has been used instead of a preferred key. As such, the header parser **104** determines whether an encryption key used to encrypt encrypted data was an improper or undesirable key. If desired, the data parser **104** may instead look for any other defined public key or other data to determine if it was used as the encryption key.

The cryptographic key package removal stage **106** may also include a data re-encryptor, such as any suitable encryption process, that allows re-encryption of either message data and/or header data using a different encryption process, symmetric or asymmetric respectively. The encryption process may include for example the use of a longer encryption key (for either the encryption of the message body, or both), or the use of a different encryption algorithm than that used to originally encrypt the received information, such as an RSA algorithm or other suitable algorithm, typically a public key algorithm. For example, the header parser could be used to convert from one-key-wrapping scheme to another. For example, if a received wrapped key has been encrypted using static Diffie-Hellman algorithm, the parser may unwrap the file key and re-wrap it using a different scheme, such as an RSA scheme for long term storage. This would allow for substitution of a more desirable or stronger scheme for long-term storage (e.g., 1024-bit RSA, whereas the received wrapped key may have been a 512-bit RSA or Diffie-Hellman key transfer). Other types of substitutions could alternatively be made, for example the use of RSA for key wrapping may be replaced by an Elliptic Curve Diffie-Hellman key wrapping in order to save storage space. For example, if the result of the latter algorithm results in a space savings. Substitute or replacement algorithms could thus be used for any number of reasons, including to improve security, achieve space savings, to conform with local security policy, to allow algorithm migrations or upgrades, to change between different formatting of the same algorithm (for example, RSA formatting per the PKCS No. 1 format versus the ANSI X9.31 format), and so on. As such, an improperly encrypted message is re-encrypted using a proper or preferred encryption public key and re-stored for the recipient's later use. Also, the data re-encryptor may increase the security strength of a given message by applying a stronger symmetric encryption process such as a triple DES process to message data encrypted using a less secure algorithm such as a single DES process. Since unnecessary header data is removed, the re-encryption is applied to only the header data for the given recipient instead of the entire received message and header data. Storage costs may be further reduced by decrypting the encrypted message data, compressing the decrypted data, and then re-encrypting the same data with either the same key and/or algorithm as before, or with a new key and/or algorithm. In the case that the same key and algorithm are re-used, the relevant key package in the header need not be modified (re-wrapped).

6

FIG. 3 shows by way of example, methods for facilitating an encryption process **300** which may start, for example, in block **302** when the memory storage reduction apparatus **100** is awaiting receipt of an encrypted message. As shown in block **304**, the cryptographic receiver **102** receives an encrypted message containing multiple cryptographic key packages associated with multiple recipients. The header parser **104** analyzes the cryptographic key packages to determine whether any cryptographic key packages contain a predefined recipient's identification data, as shown in block **306**. As previously noted, when the header data parser **104** searches and analyzes the cryptographic key packages for key identification data or identification data corresponding to a signature key, it facilitates detection of the use of an improper or non-preferred encryption key, such as where a system uses signing certificate data instead of encryption certificate data.

As shown in block **308**, if identification data corresponding to the predefined recipient is detected in the key packages, the header parser **104** determines whether the identification data represents encryption data or digital signing data as shown in block **310**. It does this by, for example, using the identification data to locate corresponding keying material such as a public key pair or public key certificate, and then examining information associated with this keying material which indicates its intended usage. Such information may for example be specified in associated data records or in the key usage field in an X.509 version 3 certificate. If no corresponding identification data is detected, the user is notified, as shown in block **312**, that the message does not concern the recipient. If the identification data does not indicate an acceptable encryption process, for example if signing key data has been used as encryption data, one of several methods may be used. In this context, "acceptable" may be determined by any of a number of criteria based on the type of process expected or preferred locally.

In one method, as shown in block **312**, the cryptographic key package removal stage, which may include the re-encryptor, re-encrypts the message data instead of, or in addition to the header data, using a proper or preferred encryption key. In the case that a new asymmetric encryption process is required, the public encryption key may be obtained by the memory reduction apparatus **100** from a stored location based on the key identification data or other data in the message. For symmetric re-encryption, the symmetric key previously used may be re-used, or preferably a new random symmetric key may be obtained from a random number generator. For example this re-encryption may be based on the public encryption key obtained from an encryption public key certificate directory or locally stored certificates for the given recipient so that the message is properly encrypted, it may be based on a new message encryption key, it may be based on a different encryption algorithm to facilitate for example a higher level of security for the encrypted message, or to facilitate reduced storage or any other suitable re-encryption of the message data. In the case where message data is re-encrypted, the existing received encrypted message data must first be decrypted to obtain the plaintext. As shown in block **314**, the header parser **104** parses out the unnecessary cryptographic key packages and stores the message data preferably with only the cryptographic key package for the given recipient.

Alternatively, if digital signing data is detected as being used as an encryption key, the memory reduction apparatus **100** decrypts the encryption symmetric key in the cryptographic key package containing the signing key data and re-wraps (encrypts) the encryption symmetric key to gener-

7

ate a new cryptographic key package for the recipient. This is shown in blocks 316 and 318. One way of detecting whether a key is preferred or improper is to check data fields in the certificate which indicate allowed usage. For example, this is commonly indicated in X.509 certificates by "key usage" bits. Also, the step of parsing can be done before or after re-encryption of the message.

Referring back to block 310, if the identification data is determined to be encryption data, again several methods may be used. One method includes decrypting the encryption symmetric key in the cryptographic key package containing the encryption data and re-encrypting the encryption symmetric key using a different encryption process to generate a new cryptographic key package for the recipient as shown in block 318. The different encryption process may involve either a different public key, or a different public key algorithm as well. If desired, as shown in FIG. 3., the header data parser 104 and data removal stage may parse and remove unnecessary cryptographic key packages and store the shorter message data as shown in block 314. A combination of removing unnecessary header data and re-encrypting the remaining header data and/or message data may also be performed. The method may end as shown in block 320 where the memory reduction apparatus waits for another message to be received.

FIG. 4 shows in more detail a method 400 that re-encrypts message data after detecting non-preferred encryption key usage. The first three blocks, 400, 402 and 404 are the same as blocks 302-306 as shown in FIG. 3. In block 406, the analyzer 103 (FIG. 1.) analyzes the received data to determine whether any of cryptographic key packages include digital signing key identification data (or other non-encryption key based data) associated with the given recipient. If no digital signing key identification data is found, the method analyzes the cryptographic key packages to determine whether acceptable encryption data was used, as shown for example in FIG. 3, block 310. The process may then continue through blocks 316, 318, and 314-320 as previously described or, if desired, once proper encryption data has been detected, the cryptographic key packages may be parsed and unnecessary cryptographic key packages are removed as shown in block 314.

Referring back to block 406, if the analyzer 103 detects that a parsed cryptographic key package includes non-preferred (for example, signing key) identification data associated with the recipient, the analyzer 103 obtains a corresponding private key from a secure file as shown in block 408. The corresponding private key (for example, signing key) is used to unwrap the cryptographic key package to recover a message encryption key as known in the art. The system may re-encrypt the key package with a different asymmetric key and/or algorithm as shown in block 409. The analyzer 103 may then decrypt the message data in any suitable manner using the message encryption key as shown in block 410. As shown in block 412, the message data is then compressed and re-encrypted by choosing any suitable encryption process. For example, re-encryption may be based on a new random message encryption key and a different symmetric encryption algorithm, by encrypting the message using the preferred encryption key, or creating an encryption header containing the cryptographic key package by encrypting the message encryption key with the public encryption key from a stored recipient public key certificate corresponding to a preferred public key algorithm. The message may then be restored for future use as indicated in block 413.

As an alternative, the step of block 409 may occur after blocks 410, 412 in the case that the process involves blocks 409, 410 and 412.

8

If desired, compression of the message can be done even if no changes are made to either the public key algorithm (wrapping) in the header, or to the symmetric encryption algorithm/key for the message (body). In this instance, the process may delete the unnecessary header cryptographic key packages and compress the message data which includes unwrapping the key package to obtain the decryption key, decrypting the encrypted message data, compressing the recovered plaintext data, then re-encrypting the compressed data.

For example, where an incoming message is encrypted under algorithm X with symmetric key Y wrapped (encrypted) with asymmetric key Z, the system may decrypt asymmetrically to recover the symmetric key Y, and re-encrypt the symmetric key Y with a different asymmetric key Z' and replace the previous cryptographic key package with the new re-encrypted key data forming a new cryptographic key package in the header. The message data with the new cryptographic key package may then be stored. This is to accomplish re-encrypting of the header without re-encrypting the file itself, thereby only changing the wrapping on the header key. The public key algorithms corresponding to the keys Z and Z' may be the same algorithm such as an RSA encryption algorithm or one key may correspond to an RSA encryption algorithm, and the other may be a different algorithm such as ElGamal encryption, a store-and-forward version of a Diffie-Hellman based encryption process, elliptic curve Diffie-Hellman encryption process or any other suitable encryption process.

Also, the step of re-encrypting recipient encryption keys may include generating a cryptographic key package by encrypting a message encryption key with a public encryption key from a public key certificate. The symmetric encryption may be performed using different encryption algorithms, for example, DES, Triple DES, and any other suitable encryption algorithm.

In addition, the re-encrypted message may be compressed using a data compressor such as any suitable compression algorithm and restored in memory for future use. Compression of message data is done after the plaintext is recovered (following decryption), and prior to re-encryption. That is, the plaintext data is always compressed, rather than ciphertext. Compressing the data reduces the amount of data to process and/or store and therefore reduces memory storage requirements, bandwidth requirements and subsequent processing load. The method ends as shown block 420 where the system waits for another message to arrive.

In another embodiment, the storage reduction device and methods may be included and performed by a gateway server that receives multiple wrapped keys and message data for a number of recipients prior to being distributed to any recipients. In this way, the gateway computer or other computer performs the necessary operations to reduce the memory storage required by a given recipient by removing unnecessary key packages associated with other recipients. In addition, it may be desirable remove all wrapped keys for several different recipients or remove some of the cryptographic key packages for selected recipients so that each recipient need not analyze all wrapped keys. This may balance overhead requirements among the recipients and a gateway server, for example.

Based on the foregoing, the disclosed systems and methods may provide detection and removal of unnecessary data relating to encryption keys prior to storing the data after receipt of the encrypted information from a sender and/or detection of whether a proper encrypting process was used.

In one embodiment, the invention determines whether an improper encryption key was used to encrypt encrypted data and re-encrypts the encrypted data with a different encryption process in response to detected improper encryption key usage.

It should be understood that the implementation of other variations and modifications of the invention in its various aspects will be apparent to those of ordinary skill in the art, and that the invention is not limited by the specific embodiments described. It is therefore contemplated to cover by the present invention, any and all modifications, variations, or equivalents that fall within the spirit and scope of the basic underlying principles disclosed and claimed herein.

What is claimed is:

1. A method for facilitating an encryption process comprising the steps of:

receiving data containing header data wherein the header data includes a plurality of cryptographic key packages associated with multiple recipients;

parsing the header data to determine whether at least one of the plurality of cryptographic key packages corresponds to key identification data for a predetermined recipient; and

removing at least some cryptographic key packages not corresponding to the key identification data for the predetermined recipient.

2. The method of claim 1 wherein at least one cryptographic key package includes at least one symmetric encryption key wrapped with an asymmetric encryption key.

3. The method of claim 1 wherein the data also contains message data and wherein the method further includes the step of re-storing the parsed header data with encrypted message data for future use wherein the re-stored parsed header data omits the removed cryptographic key packages.

4. The method of claim 1 wherein the step of removing at least some cryptographic key packages includes removing all cryptographic key packages except the cryptographic key package corresponding to the predetermined recipient.

5. The method of claim 2 including the step of re-encrypting the symmetric encryption key with a different asymmetric encryption key to generate a second and different cryptographic key package.

6. The method of claim 5 including the step of parsing the header data to determine whether at least one of the plurality of cryptographic key packages includes identification data corresponding to digital signing data for a predetermined recipient.

7. The method of claim 2 including the step of re-encrypting the symmetric encryption key using a different asymmetric encryption process than that used to originally wrap the symmetric encryption key.

8. The method of claim 7 wherein the different asymmetric encryption process is from the group of: elliptic curve, RSA, ElGamal and Diffie-Hellman encryption processes.

9. The method of claim 2 including the steps of:

retrieving a digital signing key based on identification data corresponding to digital signing data in the key identification data;

decrypting the header data to recover a message encryption key;

re-encrypting the message encryption key using a different asymmetric encryption key;

replacing key identification data with new key identification data corresponding to a new asymmetric encryption key; and

re-storing the new key identification data and the corresponding message encryption key as part of a new

cryptographic key package to replace a current cryptographic key package, for subsequent use.

10. The method of claim 1 including the step of decrypting, compressing, and re-encrypting the message data prior to restoring the encrypted message.

11. A method for facilitating an encryption process comprising the steps of:

receiving data containing header data wherein the header data includes a plurality of cryptographic key packages associated with multiple recipients;

parsing the header data to determine whether at least one of the plurality of cryptographic key packages corresponds to key identification data for a predetermined recipient; and

removing at least some cryptographic key packages not corresponding to the predetermined recipient to obtain a first remaining cryptographic key package corresponding to the predetermined recipient;

decrypting an encryption symmetric key from the first remaining cryptographic key package to obtain a recipient encryption key;

re-encrypting the recipient encryption key to generate a second cryptographic key package; and

storing the second cryptographic key package for later use by the recipient.

12. The method of claim 11 wherein the step of re-encrypting the recipient encryption key includes using at least one of elliptic curve, RSA, ElGamal and Diffie-Hellman encryption processes as a different asymmetric encryption process to generate the second cryptographic key package.

13. The method of claim 11 wherein the step of re-encrypting the recipient encryption key includes generating the second cryptographic key package by encrypting a message encryption key with a public encryption key from a public key certificate.

14. A method for facilitating an encryption process comprising the steps of:

receiving data containing header data wherein the header data includes a plurality of cryptographic key packages associated with multiple recipients;

re-encrypting the header data leaving only header data corresponding to the predetermined recipient; and

storing the re-encrypted header data for later use by the recipient.

15. The method of claim 14 wherein the step of re-encrypting includes upgrading recipient asymmetric key strength by re-encrypting a symmetric key from an associated recipient cryptographic key package with a different asymmetric key.

16. The method of claim 14 including the step of re-encrypting message data with a different symmetric encryption process, involving at least one of a different symmetric encryption key and a different symmetric encryption algorithm.

17. The method of claim 16 wherein the step of re-encrypting the message data includes:

parsing the header data and extracting a specific recipient cryptographic key package;

using a symmetric encryption key recovered from the cryptographic key package to decrypt message data;

selecting a different symmetric encryption key;

compressing the decrypted message data;

re-encrypting the message data with the different symmetric encryption process; and

11

re-wrapping the different symmetric encryption key with a public encryption key.

18. A method for facilitating an encryption process comprising the steps of:

- receiving encrypted data;
- determining an encryption key used to encrypt the encrypted data to determine whether an improper encryption key was used to encrypt the encrypted data;
- re-wrapping the encrypted data with a different encryption process in response to a detected improper encryption key usage.

19. The method of claim 18 wherein the encrypted data includes header data containing at least one cryptographic key package having a wrapped message encryption key and key identification data associated with at least one recipient and used to encrypt message data.

20. The method of claim 19 wherein key identification data is used to determine improper encryption key usage.

21. The method of claim 18 wherein the step of determining an encryption key used to encrypt the encrypted data includes determining whether a digital signing key was used as an encryption key.

22. The method of claim 20 wherein the step of re-wrapping the encrypted data with a different encryption process includes re-wrapping using a different encryption key or algorithm from the group of: RSA, elliptic curve, ElGamal and Diffie-Hellman based encryption algorithms.

23. The method of claim 21 wherein the step of re-wrapping the encrypted data with a different encryption process includes re-encrypting using a different encryption key.

24. The method of claim 19 including the steps of:

- parsing the header data to determine whether at least one of a plurality of cryptographic key packages corresponds to key identification data for a predetermined recipient; and
- removing at least some cryptographic key packages not corresponding to the key identification data.

25. The method of claim 18 where the encrypted data includes header data and message data, including the step of re-encrypting message data with a different symmetric key.

26. The method of claim 25 wherein the step of re-encrypting the message data includes:

- parsing the header data and extracting a specific recipient's cryptographic key package;
- using a symmetric encryption key recovered from the cryptographic key package to decrypt message data;
- selecting a different symmetric encryption key;
- compressing the decrypted message data;
- re-encrypting the message data with the different symmetric encryption key; and
- re-wrapping the different symmetric encryption key with a public encryption key.

27. An apparatus for facilitating an encryption process comprising:

- means for receiving data containing header data wherein the header data includes a plurality of cryptographic key packages associated with multiple recipients;
- means, operatively coupled to the receiving means, for parsing the header data to determine whether at least one of the plurality of cryptographic key packages corresponds to key identification data for a predetermined recipient; and
- means, operatively coupled to the parsing means, for removing at least some cryptographic key packages not corresponding to the key identification data.

12

28. The apparatus of claim 27 wherein at least one cryptographic key package includes at least one symmetric encryption key wrapped using an asymmetric encryption key.

29. The apparatus of claim 27 wherein the data also contains message data and wherein the system includes means for re-storing the parsed header data with encrypted message data for future use.

30. The apparatus of claim 27 wherein the means for removing at least some cryptographic key packages removes all cryptographic key packages except the cryptographic key package corresponding to the predetermined recipient.

31. The apparatus of claim 27 including means for parsing the header data to determine whether at least one of the plurality of cryptographic key packages includes digital signing key data for a predetermined recipient.

32. The apparatus of claim 28 including means for re-encrypting the symmetric encryption key with at least one of a different asymmetric encryption key and asymmetric encryption process to generate a second cryptographic key package.

33. The apparatus of claim 32 wherein the means for removing includes means for retrieving a decryption key based on key identification data, decrypting a cryptographic key package to recover a symmetric encryption key, then re-encrypting the symmetric encryption key.

34. The apparatus of claim 27 including means for compressing message data prior to restoring encrypted message data.

35. An apparatus for facilitating an encryption process comprising:

- means for receiving data containing header data wherein the header data includes a plurality of cryptographic key packages associated with multiple recipients;
- a re-encryptor, operatively responsive to the received data, that re-encrypts the header data with only header data corresponding to the predetermined recipient; and
- means for storing the re-encrypted header data for later use by the recipient.

36. The apparatus of claim 35 wherein the re-encryptor uses a preferred recipient asymmetric key by re-encrypting a symmetric key recovered from an associated recipient cryptographic key package with a different asymmetric key.

37. The apparatus of claim 35 including a message data re-encryptor that operates by:

- parsing the header data and extracting a specific recipient's cryptographic key package;
- using a symmetric encryption key recovered from the cryptographic key package to decrypt message data;
- selecting a different symmetric encryption key;
- compressing the decrypted message data;
- re-encrypting the message data with the different symmetric encryption key; and
- re-wrapping the different symmetric encryption key with a public encryption key.

38. The apparatus of claim 35 including means for determining an encryption key used to encrypt the encrypted data to determine whether a non-preferred encryption key was used to encrypt the encrypted data and for re-wrapping the encrypted data with a different encryption process in response to a detected non-preferred encryption key usage.

* * * * *