

(19)



(11)

EP 2 130 159 B1

(12)

EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:

19.06.2019 Bulletin 2019/25

(51) Int Cl.:

G06F 21/31^(2013.01) H04L 9/32^(2006.01)

(86) International application number:

PCT/US2008/055240

(21) Application number: **08730925.8**

(22) Date of filing: **28.02.2008**

(87) International publication number:

WO 2008/118602 (02.10.2008 Gazette 2008/40)

(54) **SECURE DATA STORAGE AND RETRIEVAL INCORPORATING HUMAN PARTICIPATION**

SICHERE DATENSPEICHERUNG UND -ABRUFUNG UNTER MENSCHLICHER BETEILIGUNG

STOCKAGE DE DONNÉES SÉCURISÉES ET RÉCUPÉRATION AVEC PARTICIPATION HUMAINE

(84) Designated Contracting States:

**AT BE BG CH CY CZ DE DK EE ES FI FR GB GR
HR HU IE IS IT LI LT LU LV MC MT NL NO PL PT
RO SE SI SK TR**

(74) Representative: **Goddard, Heinz J.**

**Boehmert & Boehmert
Anwaltspartnerschaft mbB
Pettenkoferstrasse 22
80336 München (DE)**

(30) Priority: **23.03.2007 US 690685**

(43) Date of publication of application:

09.12.2009 Bulletin 2009/50

(56) References cited:

**US-A- 5 921 548 US-A1- 2003 046 572
US-A1- 2004 146 164 US-A1- 2006 080 537
US-A1- 2006 287 963 US-B1- 6 754 827
US-B1- 6 944 765**

(73) Proprietor: **Microsoft Technology Licensing, LLC
Redmond, WA 98052 (US)**

- **VON AHN L ET AL: "CAPTCHA: using hard AI problems for security", SECURITY IN COMMUNICATION NETWORKS : THIRD INTERNATIONAL CONFERENCE ; REVISED PAPERS / SCN 2002, AMALFI, ITALY, SEPTEMBER 11 - 13, 2002; [LECTURE NOTES IN COMPUTER SCIENCE , ISSN 0302-9743], SPRINGER VERLAG, DE, vol. 2656, 8 May 2003 (2003-05-08), pages 294-311, XP009091663, ISBN: 978-3-540-24128-7**

(72) Inventors:

- **BAKER, Arthur H.
Redmond, WA 98052-6399 (US)**
- **GUARRACI, Brian J.
Redmond, WA 98052-6399 (US)**
- **TUCKER, Andrew Stewart
Redmond, WA 98052-6399 (US)**
- **MEDVINSKY, Gennady
Redmond, WA 98052-6399 (US)**
- **DUTTA, Tanmoy
Redmond, WA 98052-6399 (US)**

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 2 130 159 B1

Description

TECHNICAL FIELD

[0001] The technical field relates generally to computer processing and more specifically to computer security.

BACKGROUND

[0002] Typical access-control mechanisms do not distinguish between input provided by people and input provided by software. As a result, rogue software running on a system can mimic the behavior of a human user and gain unauthorized access to protected resources. For example, a keystroke logger can capture a user's account name and password and later replay them as part of an access-verification sequence. Similarly, typical access-control mechanisms are susceptible to brute force attacks such as a dictionary attack or a "man in the middle" attack. Brute force attacks can be particularly detrimental to password protected storage of encrypted data.

[0003] US 5,921,548 A discloses puzzles characterized by one or a number of sets of visually identical, physically interchangeable, rotatable pieces that either contain magnets with varying north/south orientations embedded in their sides and/or have a mark or markings on one or more of their edges and computer versions of these puzzles. The object of the puzzles is to arrange the pieces in predetermined shapes and sequences in such a manner that the sides of the pieces which are in contact with the sides of other pieces have opposite magnetic poles facing each other so that the pieces attract, rather than repel or, if edge markings are used with or instead of magnets, so that adjacent edge markings comply with specified rules. In addition, the pieces may be all visually identical except for differences in edge markings, if any, or, alternatively, there may be several sets of visually identical pieces, e.g., pink pieces, blue pieces, yellow pieces and orange pieces, which together comprise a puzzle set. Because each solution to the puzzle is a particular sequence of the puzzle pieces, that sequence of pieces can be used in various ways to generate cryptographic keys which will enable ciphertext, which accompanies the puzzle, to be decoded. In addition, with some methods of generating cryptographic keys from the sequence of pieces in the solution to the puzzle, partial cryptographic keys are generated before the puzzle is completely solved, enabling partial solutions of the puzzle to be checked to determine if they are correct, thereby simplifying very difficult puzzles and making possible the solution of otherwise virtually impossible puzzles.

[0004] US 6,944,765 B1 discloses a method of authenticating anonymous users while reducing potential for "middleman" fraud includes the step of constructing a puzzle in response to information received from a software user. The puzzle includes the received information. The puzzle is sent to the user by a software provider. The user solves the puzzle and returns the solution to

the provider. The puzzle includes a portion of a value derived from an encrypted "cookie" and an exponentiation of the derived value. The cookie includes information about the user.

[0005] Von Ahn L et al: "CAPTCHA: using hard AI problems for security", Security in Communication Networks: Third International Conference; Revised Papers/SCN 2002, Amalfi, Italy, September 11-13, 2002; [LECTURE NOTES IN COMPUTER SCIENCE, ISSN 0302-9743], SPRINGER VERLAG, DE, vol. 2656, pages 294 - 311, discloses CAPTCHA, an automated test that humans can pass, but current computer programs can't pass. In particular, different applications of CAPTCHAs for practical security are introduced.

[0006] US 6,754,827 B1 discloses an implementation of CAPTCHA codes as an authentication means in the context of a data storage system.

SUMMARY

[0007] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description Of Illustrative Embodiments. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter. The invention is defined in the appended claims.

[0008] A verification mechanism requires that a human being participate in an access verification sequence. To access data, a puzzle is provided. Proper solution of the puzzle requires human intervention. The puzzle is chosen such that its solution is within the capabilities of a human, but beyond the current state of the art for computer systems. In an example embodiment, the verification mechanism is utilized to ensure that data encrypted under a user password requires an actual human agent to participate in every attempt to access to the underlying cleartext data.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The foregoing summary, as well as the following detailed description, is better understood when read in conjunction with the appended drawings. For the purpose of illustrating the secure data storage and retrieval incorporating human participation, there is shown in the drawings exemplary constructions thereof; however, providing functionality via the secure data storage and retrieval incorporating human participation is not limited to the specific methods and instrumentalities disclosed.

Figure 1 is a flow diagram of an example sequence of events for securely storing data.

Figure 2 is a flow diagram of an example sequence of events for securely retrieving data.

Figure 3 illustrates an example implementation of the storage sequence of the secure data storage and retrieval incorporating human participation.

Figure 4 illustrates an example implementation of the retrieval sequence of the secure data storage and retrieval incorporating human participation.

Figure 5 is a diagram of an exemplary processor for implementing the secure data storage and retrieval incorporating human participation.

Figure 6 is a depiction of an example suitable computing environment in which the secure data storage and retrieval incorporating human participation can be implemented.

DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

[0010] Figure 1 and Figure 2 depict a sequential flow diagram of an example sequence of events for implementing a secure data storage and retrieval incorporating human participation. Figure 1 depicts an example sequence of events for storing secure data and Figure 2 depicts an example sequence of events for retrieving secure data. A cryptographic protocol for secure storage and retrieval of data requires authorization from a human being in order to perform data retrieval operations. In an example configuration, two agents are utilized to implement the protocol. As depicted in Figure 1, the first agent 14, a client, manages interactions with a human user 12. The second agent 16, a secure server (referred to as the human verification service or HVS), coordinates cryptographic operations and acts as a repository for protected data. The client agent 14 and the HVS server 16 are depicted separately, for the sake of clarity. It is to be understood however, that the client agent 14 and the HVS server 16 can comprise separate processors, the same processor, or a combination thereof. For example, the client agent 14 and HVS server 16 could be running in separate processes on a single computer system, running on separate hosts in a multi-computer network, or a combination thereof.

[0011] As mentioned above, Figure 1 is a flow diagram of an example sequence of events for storing secure data. At step 18, the client agent 14 generates a cryptographic key, K . The cryptographic key can comprise any appropriate random cryptographic key. In an example embodiment, the cryptographic key, K , is a symmetric key (e.g., a symmetric key in accordance with the Advanced Encryption Standard, AES). At step 20, the plain text of the data to be stored is encrypted with the cryptographic key, K , to generate encrypted data. Another cryptographic key, K_1 , is generated by the client agent 14 at step 22. The cryptographic key, K_1 , can comprise any appropriate cryptographic key, such as a symmetric cryptographic key, for example. It is to be understood

that the sequence of events as depicted in the Figures is exemplary and thus not limited thereto. The sequence of events can be modified as appropriate. For example, step 22 could be performed before step 18 or step 20.

[0012] The client agent 14 queries the user, at step 24, for a secret, such as a PIN, a password, or the like. The user responds to the query with the user's secret, S_U , at step 26. The client agent 14 operates on the user's secret, S_U , with a cryptographic hash function, at step 28. Hash functions are known in the art. A hash function is a function that transforms a variable-size input into a fixed size value. Typically, hash functions are one way, meaning that it is impracticable or impossible to determine the input value from the output (transformed) value. Providing the same input to a hash function will provide the same output. A slight change in the input typically results in a considerable change in the output. Example hash functions include MD2, MD5, SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512.

[0013] At step 30, the cryptographic key, K , is encrypted with the cryptographic key, K_1 . At step 32, the Hash value indicative of the user's secret, $\text{Hash}(S_U)$, is encrypted with the cryptographic key, K_1 . In an example embodiment, the user's secret, S_U , is not stored in the client agent 14 after the hash function is performed. This can provide added security because the user's secret, S_U , is less susceptible to theft. At step 34, the cryptographic key, K_1 , is encrypted with the public key, of a public/private cryptographic key pair of the secure server 16. Public key cryptography, which utilizes a public/private key pair, is known in the art. One key is used to encrypt and the other is used to decrypt. Knowledge of one key does not provide knowledge of the other key. Typically one key is kept secret, and thus called the private key. The other key typically is made public. An example of public key cryptography is RSA public key cryptography. The information encrypted at steps 30, 32, and 33, referred to as a blob in Figure 1, is provided to the secure server 16, at step 36. That is, at step 36, the client agent 14 provides to the secure server 16, the cryptographic key, K , encrypted with the cryptographic key, K_1 , the hash value indicative of the user's secret, $\text{Hash}(S_U)$, encrypted with the cryptographic key, K_1 , and the cryptographic key, K_1 , encrypted with the public key, of the secure server's 16 public/private cryptographic key pair.

[0014] At step 38, the secure server 16 generates a unique tag value, referred to as a retrieval tag in Figure 1, to identify the blob received at step 36. The blob is stored in a database, memory, or the like of the secure server at step 40. The blob is indexed by the retrieval tag. Thus, the blob can be retrieved from the database, memory, or the like, via the retrieval tag. At step 42, the secure server 16 provides the retrieval tag to the client agent 14. At step 44, the client agent 14 stores the encrypted data generated at step 20 using the retrieval tag as an index. That is, the client agent 14 stores in a database, memory, or the like the plain text encrypted with the cryptographic key, K , and the stored encrypted data

is indexed utilizing the retrieval tag such that the retrieval tag can be used to retrieve the encrypted data from storage.

[0015] Figure 2 is a flow diagram of an example sequence of events for securely retrieving data. To retrieve the original plain text, the client agent 14 utilizes the retrieval tag that was generated (at step 38) during the storage process to access the secure server 16. A secure channel is established, at step 46, between the client agent 14 and the secure server 16. The secure channel can be established in accordance with any appropriate means. At step 48, the client agent 14 sends the retrieval tag and a request for a puzzle to the secure server 16.

[0016] A puzzle and the correct solution to the puzzle are generated at step 50. The puzzle can comprise any appropriate type of puzzle requiring human intervention to obtain a solution. In various embodiments, the puzzle can require a user to respond to a visual rendering, an audio rendering, or a combination thereof. For example, the puzzle could require a user to enter letters observed in a distorted image (e.g., a completely automated public Turing test, CAPTCHA). The puzzle could require a user to answer a question observed in a distorted image. The puzzle could require a user to enter letters heard in distorted audio. The puzzle could require a user to answer a question heard in distorted audio. In various embodiments, the user's response can be provided via a keyboard, a mouse, orally, or a combination thereof. The puzzle is chosen such that its solution is within the capabilities of a human, but beyond the current state of the art for computer systems. The puzzle can be generated by the secure server 16, by another processor, or a combination thereof. In an example embodiment, the puzzle is obtained via a library of pluggable puzzle generators. This embodiment allows puzzles to be replaced as the state of the art of computing technology makes some puzzles more susceptible to mechanical solution. In various embodiments, the puzzle can be generated by the secure server 16 each time a request is provided thereto, the puzzle can be a predetermined puzzle, a puzzle type (e.g., puzzle category) can be predetermined, or a combination thereof. If the puzzle is a predetermined puzzle or the puzzle type is predetermined, an indication of a puzzle or puzzle type could be provided at step 48.

[0017] At step 52, the secure server 16 generates a random cryptographic key, K_2 . The cryptographic key, K_2 , can comprise any appropriate cryptographic key, such as a symmetric cryptographic key, for example. At step 54, the correct solution to the puzzle is encrypted with the cryptographic key, K_2 . In an example embodiment, a timestamp is included with the correct solution and encrypted with the cryptographic key, K_2 . The encrypted correct solution and optional timestamp, referred to as a verification blob, and the puzzle, are provided by the secure server 16 to the client agent 14, at step 56. The puzzle is provided by the client agent 14 to the user 12 at step 58. The user 12 solves the puzzle incorporating human participation and provides a proposed solution to

the client agent 14, at step 60. The user 12 is queried for the user's secret at step 62. The user 12 responds to the query (step 62) with the user's secret, S_U , at step 64. At step 66, the client agent 14, sends to the secure server 16, the proposed solution (received at step 60), the hash value indicative of the user's secret ($\text{Hash}(S_U)$), and the verification blob (the encrypted correct solution and optional timestamp generated at step 54).

[0018] The secure server 16, at step 68, retrieves the indexed client blob utilizing the retrieval tag. That is, the secure server 16 retrieves, using the retrieval tag generated at step 38, the cryptographic key, K , encrypted with the cryptographic key, K_1 , the hash value indicative of the user's secret, $\text{Hash}(S_U)$, encrypted with the cryptographic key, K_1 , and the cryptographic key, K_1 , encrypted with the public key, of the secure server's 16 public/private cryptographic key pair, stored at step 40. At step 70, the cryptographic key, K_1 , that was encrypted (at step 34) with the public key, PK_{HVS} is decrypted with the private key, PK^{-1}_{HVS} , of the secure server's 16 public/private cryptographic key pair. At step 72, the hash value indicative of the user's secret ($\text{Hash}(S_U)$), that was encrypted (at step 32) is decrypted with cryptographic key, K_1 . At step 74, the verification blob (correct solution and optional timestamp generated) that was encrypted (at step 54) is decrypted with the cryptographic key, K_2 .

[0019] Verification is performed at step 76. If verification is successful, access to the stored data (plain text) is allowed to. If verification is unsuccessful, access to the data (plain text) is not allowed. In an example embodiment, verification fails if the hash value indicative of the user's secret provided at step 64 differs from the hash value indicative of the user's secret provided at step 26. That is, verification fails if the user's secret provided when the client agent 14 first queries the user 12 differs from the user's secret provided with the user's proposed solution to the puzzle. In an example embodiment, the user 12 is given a limited period of time to provide a proposed response to the puzzle. Accordingly, the verification fails if the current time, i.e., the time at which the secure server 16 is performing verification, differs from the time indicated by the time stamp in the verification blob (generated at step 54) by greater than a predetermined amount of time. For example, if the current time differs from the time indicated by the time stamp by more than 30 seconds, verification will fail. Verification fails if the proposed solution to the puzzle provided by the user 12 (at step 60) differs from the correct solution.

[0020] Figure 3 and Figure 4 illustrate an example implementation of the secure data storage and retrieval incorporating human participation utilizing a passive Universal Serial Bus (USB) device 90 as storage for the storage management application 86. Figure 3 illustrates an example implementation of the storage sequence of the secure data storage and retrieval incorporating human participation. The implementation comprises a storage management application 86 (e.g., the client agent), a human verification service (HVS) 88 (e.g., the secure HVS

server), the storage medium 90, and storage medium 92. The storage medium 90 can comprise any appropriate storage medium, such as a disk, semiconductor memory, a database, or the like, for example. As depicted in Figure 3, in an example embodiment, the storage medium 90 comprises a USB storage device (e.g., a key drive). The storage device 90 requires no special cryptographic capabilities. The storage medium 92 can comprise any appropriate storage medium, such as a disk, semiconductor memory, a database, or the like, for example.

[0021] To accomplish storage of secure data, the storage-management application 86 accepts a user data-key to be stored, a password, and a friendly display name from the user. A user data key can comprise any appropriate secret that guards sensitive user data. For example, a data key can comprise a password, an encryption, of the like. The display name functions as a reminder to the user of the identity of the user data-key. the display name is not be encrypted. After executing the storage sequence of the secure data storage and retrieval incorporating human participation as described above, the storage-management application 86 possesses a retrieval tag and a copy of the user-data key that has been encrypted with the cryptographic key, K. The storage management application 86 stores both the retrieval tag and the copy of the user-data key along with the data key's friendly display name on the USB device 90.

[0022] After executing the of the storage sequence, as described above, of the storage and retrieval incorporating human participation, the HVS 88 possesses a copy of the cryptographic key, K, and the hash value indicative of the user's secret, S_U , both of which are encrypted using the cryptographic key K_1 . The HVS 88 also possesses an encrypted copy of K_1 that the HVS 88 can recover by using the HVS's 88 private key. These items are stored as a group and indexed with the retrieval tag value. Taken together, this data will allow the HVS 88 to recreate the value of K and the hash value indicative of the user's secret.

[0023] Figure 4 illustrates an example implementation of the retrieval sequence of the secure data storage and retrieval incorporating human participation. When the user wants to recover the data key, the user executes the storage management application 86 again. The storage-management application 86 presents a list of friendly display names loaded from the USB device 90 to the user. The user can select one of the display names to recover the associated data key(s). During the retrieval sequence of the secure data storage and retrieval incorporating human participation, the storage management application 86 prompts the user for a password or the like and provides to the user a puzzle requiring human participation to obtain a solution. Upon receiving the user's response (e.g., the user's proposed solution to the puzzle), the storage management application 86, sends the user's response to the HVS 88. The HVS 88 evaluates the password and user's proposed puzzle solution. If the HVS determines that the password is correct and the user's

proposed puzzle solution is correct, the HVS 88 returns the original key, K, to the storage management application 88. The storage management application 86 uses cryptographic key, K, to decrypt the original user data key from the USB storage device 90. At no point during the storage sequence or the retrieval sequence, does the HVS 88 have access to the plaintext version of the user's data key.

[0024] Figure 5 is a diagram of an exemplary processor 94 for implementing the secure data storage and retrieval incorporating human participation. The processor 94 comprises a processing portion 96, a memory portion 98, and an input/output portion 100. The processing portion 96, memory portion 98, and input/output portion 100 are coupled together (coupling not shown in Figure 5) to allow communications therebetween. The input/output portion 100 is capable of providing and/or receiving components utilized to implement the secure data storage and retrieval incorporating human participation as described above. For example, the input/output portion 100 is cable of receiving and/or providing, a query for a user's secret, a user's secret, encrypted information, a cryptographic key, indexed information, a hash value, a retrieval tag, a puzzle, a proposed puzzle solution, or a combination thereof.

[0025] The processing portion 96 is capable of implementing the secure data storage and retrieval incorporating human participation as described above. For example, the processing portion 96 is capable of generating a cryptographic key, performing encryption, performing a hash function, generating a retrieval tag, storing an indexed information, establishing a secure channel, generating a puzzle requiring human participation to obtain a solution, retrieving indexed information, performing decryption, performing verification, determining if the current time differs from the time indicated by the time stamp by at least a predetermined amount of time, determining if a first user's secret differs from a second user's secret, determining if the proposed puzzle solution is correct, or a combination thereof.

[0026] The processor 94 can be implemented as a client processor and/or a server processor. In a basic configuration, the processor 94 can include at least one processing portion 96 and memory portion 98. The memory portion 98 can store any information utilized in conjunction with implementing the secure data storage and retrieval incorporating human participation. For example, the memory portion 98 can store a cryptographic key, a user's secret, a puzzle, a puzzle solution, a hash value, indexed information, a retrieval tag, or a combination thereof. Depending upon the exact configuration and type of processor, the memory portion 98 can be volatile (such as RAM) 102, non-volatile (such as ROM, flash memory, etc.) 104, or a combination thereof. The processor 94 can have additional features/functionality. For example, the processor 94 can include additional storage (removable storage 106 and/or non-removable storage 108) including, but not limited to, magnetic or optical

disks, tape, flash, smart cards or a combination thereof. Computer storage media, such as memory portion 98, 102, 104, 106, and 108, include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules, or other data. Computer storage media include, but are not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, universal serial bus (USB) compatible memory, smart cards, or any other medium which can be used to store the desired information and which can be accessed by the processor 94. Any such computer storage media can be part of the processor 94.

[0027] The processor 94 can also contain communications connection(s) 114 that allow the processor 94 to communicate with other devices, for example. Communications connection(s) 114 is an example of communication media. Communication media typically embody computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. The term computer readable media as used herein includes both storage media and communication media. The processor 94 also can have input device(s) 112 such as keyboard, mouse, pen, voice input device, touch input device, *etc.* Output device(s) 110 such as a display, speakers, printer, *etc.* also can be included.

[0028] Figure 6 and the following discussion provide a brief general description of an example suitable computing environment in which the secure data storage and retrieval incorporating human participation can be implemented. Although not required, various aspects of the secure data storage and retrieval incorporating human participation can be described in the general context of computer executable instructions, such as program modules, being executed by a computer, such as a client workstation or a server. Generally, program modules include routines, programs, objects, components, data structures and the like that perform particular tasks or implement particular abstract data types. Moreover, implementation of the secure data storage and retrieval incorporating human participation can be practiced with other computer system configurations, including hand held devices, multi processor systems, microprocessor based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like.

Further, the provision of functionality via the secure data storage and retrieval incorporating human participation also can be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules can be located in both local and remote memory storage devices.

[0029] A computer system can be roughly divided into three component groups: the hardware component, the hardware/software interface system component, and the applications programs component (also referred to as the "user component" or "software component"). In various embodiments of a computer system the hardware component may comprise the central processing unit (CPU) 521, the memory (both ROM 564 and RAM 525), the basic input/output system (BIOS) 566, and various input/output (I/O) devices such as a keyboard 540, a mouse 542, a monitor 547, and/or a printer (not shown), among other things. The hardware component comprises the basic physical infrastructure for the computer system.

[0030] The applications programs component comprises various software programs including but not limited to compilers, database systems, word processors, business programs, videogames, and so forth. Application programs provide the means by which computer resources are utilized to solve problems, provide solutions, and process data for various users (machines, other computer systems, and/or end-users). In an example embodiment, application programs perform the functions associated with the secure data storage and retrieval incorporating human participation as described above.

[0031] The hardware/software interface system component comprises (and, in some embodiments, may solely consist of) an operating system that itself comprises, in most cases, a shell and a kernel. An "operating system" (OS) is a special program that acts as an intermediary between application programs and computer hardware. The hardware/software interface system component may also comprise a virtual machine manager (VMM), a Common Language Runtime (CLR) or its functional equivalent, a Java Virtual Machine (JVM) or its functional equivalent, or other such software components in the place of or in addition to the operating system in a computer system. A purpose of a hardware/software interface system is to provide an environment in which a user can execute application programs.

[0032] The hardware/software interface system is generally loaded into a computer system at startup and thereafter manages all of the application programs in the computer system. The application programs interact with the hardware/software interface system by requesting services via an application program interface (API). Some application programs enable end-users to interact with the hardware/software interface system via a user interface such as a command language or a graphical user interface (GUI).

[0033] A hardware/software interface system traditionally performs a variety of services for applications. In a multitasking hardware/software interface system where multiple programs may be running at the same time, the hardware/software interface system determines which applications should run in what order and how much time should be allowed for each application before switching to another application for a turn. The hardware/software interface system also manages the sharing of internal memory among multiple applications, and handles input and output to and from attached hardware devices such as hard disks, printers, and dial-up ports. The hardware/software interface system also sends messages to each application (and, in certain case, to the end-user) regarding the status of operations and any errors that may have occurred. The hardware/software interface system can also offload the management of batch jobs (e.g., printing) so that the initiating application is freed from this work and can resume other processing and/or operations. On computers that can provide parallel processing, a hardware/software interface system also manages dividing a program so that it runs on more than one processor at a time.

[0034] A hardware/software interface system shell (referred to as a "shell") is an interactive end-user interface to a hardware/software interface system. (A shell may also be referred to as a "command interpreter" or, in an operating system, as an "operating system shell"). A shell is the outer layer of a hardware/software interface system that is directly accessible by application programs and/or end-users. In contrast to a shell, a kernel is a hardware/software interface system's innermost layer that interacts directly with the hardware components.

[0035] As shown in Figure 6, an exemplary general purpose computing system includes a conventional computing device 560 or the like, including a processing unit 521, a system memory 562, and a system bus 523 that couples various system components including the system memory to the processing unit 521. The system bus 523 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory includes read only memory (ROM) 564 and random access memory (RAM) 525. A basic input/output system 566 (BIOS), containing basic routines that help to transfer information between elements within the computing device 560, such as during start up, is stored in ROM 564. The computing device 560 may further include a hard disk drive 527 for reading from and writing to a hard disk (hard disk not shown), a magnetic disk drive 528 (e.g., floppy drive) for reading from or writing to a removable magnetic disk 529 (e.g., floppy disk, removal storage), and an optical disk drive 530 for reading from or writing to a removable optical disk 531 such as a CD ROM or other optical media. The hard disk drive 527, magnetic disk drive 528, and optical disk drive 530 are connected to the system bus 523 by a hard disk drive interface 532, a magnetic disk drive interface 533, and

an optical drive interface 534, respectively. The drives and their associated computer readable media provide non volatile storage of computer readable instructions, data structures, program modules and other data for the computing device 560. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 529, and a removable optical disk 531, it should be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, random access memories (RAMs), read only memories (ROMs), and the like may also be used in the exemplary operating environment. Likewise, the exemplary environment may also include many types of monitoring devices such as heat sensors and security or fire alarm systems, and other sources of information.

[0036] A number of program modules can be stored on the hard disk, magnetic disk 529, optical disk 531, ROM 564, or RAM 525, including an operating system 535, one or more application programs 536, other program modules 537, and program data 538. A user may enter commands and information into the computing device 560 through input devices such as a keyboard 540 and pointing device 542 (e.g., mouse). Other input devices (not shown) may include a microphone, joystick, game pad, satellite disk, scanner, or the like. These and other input devices are often connected to the processing unit 521 through a serial port interface 546 that is coupled to the system bus, but may be connected by other interfaces, such as a parallel port, game port, or universal serial bus (USB). A monitor 547 or other type of display device is also connected to the system bus 523 via an interface, such as a video adapter 548. In addition to the monitor 547, computing devices typically include other peripheral output devices (not shown), such as speakers and printers. The exemplary environment of Figure 6 also includes a host adapter 555, Small Computer System Interface (SCSI) bus 556, and an external storage device 562 connected to the SCSI bus 556.

[0037] The computing device 560 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 549. The remote computer 549 may be another computing device (e.g., personal computer), a server, a router, a network PC, a peer device, or other common network node, and typically includes many or all of the elements described above relative to the computing device 560, although only a memory storage device 550 (floppy drive) has been illustrated in Figure 6. The logical connections depicted in Figure 6 include a local area network (LAN) 551 and a wide area network (WAN) 552. Such networking environments are commonplace in offices, enterprise wide computer networks, intranets and the Internet.

[0038] When used in a LAN networking environment, the computing device 560 is connected to the LAN 551 through a network interface or adapter 553. When used in a WAN networking environment, the computing device

560 can include a modem 554 or other means for establishing communications over the wide area network 552, such as the Internet. The modem 554, which may be internal or external, is connected to the system bus 523 via the serial port interface 546. In a networked environment, program modules depicted relative to the computing device 560, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

[0039] While it is envisioned that numerous embodiments of the secure data storage and retrieval incorporating human participation are particularly well-suited for computerized systems, nothing in this document is intended to limit the invention to such embodiments. On the contrary, as used herein the term "computer system" is intended to encompass any and all devices capable of storing and processing information and/or capable of using the stored information to control the behavior or execution of the device itself, regardless of whether such devices are electronic, mechanical, logical, or virtual in nature.

[0040] The various techniques described herein can be implemented in connection with hardware or software or, where appropriate, with a combination of both. Thus, the methods and apparatuses for implementing the provision of functionality the secure data storage and retrieval incorporating human participation, or certain aspects or portions thereof, can take the form of program code (i.e., instructions) embodied in tangible media, such as floppy diskettes, CD-ROMs, hard drives, or any other machine-readable storage medium, wherein, when the program code is loaded into and executed by a machine, such as a computer, the machine becomes an apparatus for providing functionality via the secure data storage and retrieval incorporating human participation.

[0041] The program(s) can be implemented in assembly or machine language, if desired. In any case, the language can be a compiled or interpreted language, and combined with hardware implementations. The methods and apparatuses for implementing the secure data storage and retrieval incorporating human participation also can be practiced via communications embodied in the form of program code that is transmitted over some transmission medium, such as over electrical wiring or cabling, through fiber optics, or via any other form of transmission, wherein, when the program code is received and loaded into and executed by a machine, such as an EPROM, a gate array, a programmable logic device (PLD), a client computer, or the like. When implemented on a general-purpose processor, the program code combines with the processor to provide a unique apparatus that operates to invoke the functionality of the secure data storage and retrieval incorporating human participation. Additionally, any storage techniques used in connection with the secure data storage and retrieval incorporating human participation can invariably be a combination of hardware

and software.

[0042] While secure data storage and retrieval incorporating human participation has been described in connection with the example embodiments of the various figures, it is to be understood that other similar embodiments can be used or modifications and additions can be made to the described embodiments for performing the same functions of secure data storage and retrieval incorporating human participation without deviating therefrom. Therefore, secure data storage and retrieval incorporating human participation as described herein should not be limited to any single embodiment, but rather should be construed in breadth and scope in accordance with the appended claims.

Claims

1. A computer related security method incorporating human participation, the method comprising:

receiving a request for data stored in a storage medium;
responsive to the request, transmitting (48) from a client agent (14) to a server (16), a puzzle request and a retrieval tag, wherein the retrieval tag is suitable for enabling the retrieval of a verification blob comprising an encrypted solution to a puzzle;
receiving (56), at the client agent (14), from the server (16), the puzzle, wherein human participation is intended to obtain a solution to the puzzle, and a correct solution to the puzzle;
transmitting (58) the puzzle from the client agent (14) to a user (12);
receiving (60), from the user (12), a proposed solution to the puzzle;
transmitting (62) to the user, a query seeking identification of a stored secret;
receiving (64) from the user, a response to the query seeking identification of the stored secret;
transmitting (66) to the server, the proposed solution to the puzzle and the response to the query;
using (68) in the server, the retrieval tag received from the client agent to retrieve the verification blob;
determining if the proposed solution and the response to the query is correct; and
if the proposed solution and the response to the query is correct, allowing access to the data stored in the storage medium.

2. A method in accordance with claim 1, further comprising, if an amount of time between providing the puzzle and receiving the proposed solution is greater than a predetermined amount of time, not allowing access to the data.

3. A method in accordance with claim 1, further comprising:

prior to receiving (60) the proposed solution, requesting a first secret; 5
 receiving the first secret;
 subsequent to receiving the request for data, requesting a second secret;
 receiving the second secret; 10
 comparing the first secret with the second secret; and
 if the first secret does not match the second secret, not allowing access to the data.

4. A method in accordance with claim 1, wherein: 15

the puzzle is intended to be visually rendered; and
 a solution to the puzzle is intended to be obtained via observation of the visually rendering of the puzzle. 20

5. A method in accordance with claim 1, wherein:

the puzzle is intended to be audibly rendered; and 25
 a solution to the puzzle is intended to be obtained via observation of the audio rendering of the puzzle. 30

6. A method in accordance with claim 1, further comprising obtaining the puzzle from a library of pluggable puzzle generators.

7. A method in accordance with claim 1, further comprising: 35

encrypting the data with a first cryptographic key;
 encrypting the first cryptographic key with a second cryptographic key; 40
 storing in the storage medium, the encrypted data and the encrypted first cryptographic key; and
 if the proposed solution is correct: 45

decrypting, with the second cryptographic key, the encrypted first cryptographic key; and
 providing the decrypted first cryptographic key. 50

8. A system for implementing computer related security incorporating human participation, the system comprising: 55

an input/output portion configured to receive a request for data stored in a memory portion;
 a first processing portion of a client agent (14),

the first processing portion configured to:

responsive to the request, transmit (48) from the client agent (14) to a server (16), a puzzle request and a retrieval tag, wherein the retrieval tag is suitable for enabling the retrieval of a verification blob comprising an encrypted solution to a puzzle;
 receive (56), at the client agent (14), from a server (16), the puzzle, wherein human participation is intended to obtain a solution to the puzzle, and a correct solution to the puzzle;
 transmit (58) the puzzle from the client agent (14) to a user (12);
 and
 receive, from the user (12), a proposed solution to the puzzle;
 transmit (62) to the user, a query seeking identification of a stored secret;
 receive (64) from the user, a response to the query seeking identification of the stored secret;
 transmit (66) to the server, the proposed solution to the puzzle and the response to the query;

a second processing portion of the server, the second processing portion configured to:

using (68) in the server, the retrieval tag received from the client agent to retrieve the verification blob;
 determine if the proposed solution and the response to the query is correct; and if the proposed solution and the response to the query is correct, allow access to the data stored in the memory portion; and
 the memory portion configured to store the data.

9. A system in accordance with claim 8, wherein the processing portion is further configured to, if an amount of time between providing the puzzle and receiving the proposed solution is greater than a predetermined amount of time, not allow access to the data.

10. A system in accordance with claim 8, wherein:

the processing portion is further configured to:

prior to receiving the proposed solution, request a first secret;
 subsequent to receiving the request for data, request a second secret;
 compare the first secret with the second secret; and
 if the first secret does not match the second

- secret, not allow access to the data; and
- the input/output portion is further configured to:
- receive the first secret; and 5
receive the second secret.
11. A system in accordance with claim 8, wherein:
- the puzzle is intended to be visually rendered; 10
and
a solution to the puzzle is intended to be obtained via observation of the visually rendering of the puzzle. 15
12. A system in accordance with claim 8, wherein:
- the puzzle is intended to be audibly rendered; and
a solution to the puzzle is intended to be obtained via observation of the audio rendering of the puzzle. 20
13. A system in accordance with claim 8, the processing portion further configured to obtain the puzzle from a library of pluggable puzzle generators. 25
14. A system in accordance with claim 8, wherein:
- the processing portion is further configured to: 30
- encrypt the data with a first cryptographic key;
encrypt the first cryptographic key with a second cryptographic key; 35
store in the memory portion, the encrypted data and the encrypted first cryptographic key; and
if the proposed solution is correct, decrypt, with the second cryptographic key, the encrypted first cryptographic key; and 40
- the input/output portion is further configured to provide the decrypted first cryptographic key. 45
15. A computer-readable medium having stored thereon computer-executable instructions for performing computer related security incorporating human participation, by performing the method in accordance with one of the claims 1 to 7. 50

Patentansprüche

1. Computerbezogenes Sicherheitsverfahren mit menschlicher Beteiligung, wobei das Verfahren umfasst: 55

Empfangen einer Anforderung für Daten, die auf einem Speichermedium gespeichert sind;
als Reaktion auf die Anforderung, Übertragen (48) einer Rätselanforderung und eines Abruf-Tags von einem Client-Agenten (14) an einen Server (16), wobei das Abruf-Tag geeignet ist, das Abrufen eines Verifizierungsblobs zu ermöglichen, der eine verschlüsselte Lösung für ein Puzzle umfasst;
Empfangen (56) des Puzzles beim Client-Agenten (14) vom Server (16), wobei die menschliche Teilnahme dazu bestimmt ist, eine Lösung für das Puzzle und eine korrekte Lösung für das Puzzle zu erhalten;
Übertragen (58) des Puzzles von dem Client-Agenten (14) zu einem Benutzer (12);
Empfangen (60) einer vorgeschlagenen Lösung für das Puzzle vom Benutzer (12);
Übertragen (62) einer Anfrage an den Benutzer, die die Identifizierung eines gespeicherten Geheimnisses zum Ziel hat;
Empfangen (64) einer Antwort auf die Anfrage, die die Identifizierung des gespeicherten Geheimnisses zum Ziel hat, vom Benutzer;
Übertragen (66) an den Server, die vorgeschlagene Lösung für das Puzzle und die Antwort auf die Anfrage;
Verwenden (68) des vom Client-Agenten empfangenen Abruf-Tags im Server, um den Verifizierungs-Blob abzurufen;
Bestimmen, ob die vorgeschlagene Lösung und die Antwort auf die Anfrage korrekt sind; und
wenn die vorgeschlagene Lösung und die Antwort auf die Anfrage korrekt sind, Erlauben von Zugriff auf die auf dem Speichermedium gespeicherten Daten.

2. Verfahren nach Anspruch 1, ferner umfassend, wenn eine Zeitspanne zwischen dem Bereitstellen des Puzzles und dem Empfangen der vorgeschlagenen Lösung größer ist als eine vordefinierte Zeitspanne, nicht Erlauben von Zugriff auf die Daten.

3. Verfahren nach Anspruch 1, ferner umfassend:

vor dem Empfangen (60) der vorgeschlagenen Lösung, Anfordern eines ersten Geheimnisses;
Erhalten des ersten Geheimnisses;
nach Erhalt der Anfrage für Daten, Anfordern eines zweiten Geheimnisses;
Empfangen des zweiten Geheimnisses;
Vergleichen des ersten Geheimnisses mit dem zweiten Geheimnis; und
wenn das erste Geheimnis nicht mit dem zweiten Geheimnis übereinstimmt, nicht Erlauben von Zugriff auf die Daten.

4. Verfahren nach Anspruch 1, wobei:

das Puzzle dazu bestimmt ist, visuell wiedergegeben zu werden; und
eine Lösung für das Puzzle durch Beobachtung der visuellen Darstellung des Puzzles erhalten werden soll.

5

5. Verfahren nach Anspruch 1, wobei:

das Puzzle dazu bestimmt ist, hörbar wiedergegeben zu werden; und
eine Lösung für das Puzzle durch Beobachtung der Audiowiedergabe des Puzzles erhalten werden soll.

10

6. Verfahren nach Anspruch 1, ferner umfassend das Erhalten des Puzzles aus einer Bibliothek von steckbaren Puzzle-Generatoren.

15

7. Verfahren nach Anspruch 1, ferner umfassend:

20

Verschlüsseln der Daten mit einem ersten kryptographischen Schlüssel;
Verschlüsseln des ersten kryptografischen Schlüssels mit einem zweiten kryptografischen Schlüssel;
Speichern der verschlüsselten Daten und des verschlüsselten ersten kryptographischen Schlüssels auf dem Speichermedium; und
wenn die vorgeschlagene Lösung korrekt ist:

25

30

Entschlüsseln des verschlüsselten ersten kryptographischen Schlüssels mit dem zweiten kryptographischen Schlüssel; und
Bereitstellen des entschlüsselten ersten kryptografischen Schlüssels.

35

8. System zur Implementierung computerbezogener Sicherheit unter Einbeziehung menschlicher Partizipation, wobei das System umfasst:

40

einen Ein-/Ausgabebereich, der konfiguriert ist, um eine Anforderung für in einem Speicherabschnitt gespeicherte Daten zu empfangen;
einen ersten Verarbeitungsabschnitt eines Client-Agenten (14), wobei der erste Verarbeitungsabschnitt konfiguriert ist für:

45

als Reaktion auf die Anforderung, Übertragen (48) einer Rätselanforderung und eines Abrufkennzeichens vom Client-Agenten (14) an einen Server (16), wobei das Abrufkennzeichen geeignet ist, das Abrufen eines Verifizierungsblobs zu ermöglichen, der eine verschlüsselte Lösung für ein Puzzle umfasst;
Empfangen (56), beim Client-Agenten (14), des Rätsels von einem Server (16), wobei die menschliche Teilnahme dazu bestimmt

55

ist, eine Lösung für das Rätsel und eine korrekte Lösung für das Rätsel zu erhalten;
Übertragen (58) des Puzzles von dem Client-Agenten (14) zu einem Benutzer (12);
und

Erhalten, vom Benutzer (12), eines Lösungsvorschlags für das Puzzle;
Übertragen (62) einer Anfrage, die Identifizierung eines gespeicherten Geheimnisses zum Ziel hat, an den Benutzer;
Empfangen (64) einer Antwort auf die Anfrage, die Identifizierung des gespeicherten Geheimnisses zum Ziel hat, vom Benutzer;
Übertragen (66), an den Server, der vorgeschlagenen Lösung für das Puzzle und die Antwort auf die Anfrage;

einen zweiten Verarbeitungsabschnitt des Servers, wobei der zweite Verarbeitungsabschnitt konfiguriert ist, um:

Verwenden (68) des vom Client-Agenten empfangenen Abruf-Tags im Server, um den Verifizierungs-Blob abzurufen;

Feststellen, ob die vorgeschlagene Lösung und die Antwort auf die Anfrage korrekt sind; und
wenn die vorgeschlagene Lösung und die Antwort auf die Anfrage korrekt sind, Erlauben von Zugriff auf die im Speicherabschnitt gespeicherten Daten; und
wobei der Speicherabschnitt konfiguriert ist, um die Daten zu speichern.

9. System nach Anspruch 8, wobei der Verarbeitungsabschnitt ferner konfiguriert ist, um, wenn eine Zeitspanne zwischen dem Bereitstellen des Puzzles und dem Empfangen der vorgeschlagenen Lösung größer als eine vorbestimmte Zeitspanne ist, keinen Zugriff auf die Daten zu erlauben.

10. System nach Anspruch 8, wobei:

der Verarbeitungsabschnitt weiter konfiguriert ist für:

vor dem Erhalten der vorgeschlagene Lösung, Anfordern eines ersten Geheimnisses;
nach Erhalt der Datenanfrage Anfordern eines zweiten Geheimnisses;
Vergleichen des ersten Geheimnisses mit dem zweiten Geheimnis; und
wenn das erste Geheimnis nicht mit dem zweiten Geheimnis übereinstimmt, nicht Erlauben von Zugriff auf die Daten; und

der Ein-/Ausgangsabschnitt ist weiterhin konfiguriert für:

Empfangen des ersten Geheimnisses; und
Empfangen des zweiten Geheimnisses.

11. System nach Anspruch 8, wobei:

das Puzzle dazu bestimmt ist, visuell wiedergegeben zu werden; und
eine Lösung für das Puzzle durch Beobachtung der visuellen Darstellung des Puzzles erhalten werden soll.

12. System nach Anspruch 8, wobei:

das Puzzle dazu bestimmt ist, hörbar wiedergegeben zu werden; und
eine Lösung für das Puzzle soll durch Beobachtung der Audiowiedergabe des Puzzles erhalten werden.

13. System nach Anspruch 8, wobei der Verarbeitungsabschnitt ferner konfiguriert ist, um das Puzzle aus einer Bibliothek von steckbaren Puzzle-Generatoren zu beziehen.

14. System nach Anspruch 8, wobei:

der Verarbeitungsabschnitt weiter konfiguriert ist für:

Verschlüsseln der Daten mit einem ersten kryptographischen Schlüssel;
Verschlüsseln des ersten kryptografischen Schlüssels mit einem zweiten kryptografischen Schlüssel;
Speichern der verschlüsselten Daten und des verschlüsselten ersten kryptografischen Schlüssels im Speicherabschnitt;
und
wenn die vorgeschlagene Lösung korrekt ist, Entschlüsseln, mit dem zweiten kryptographischen Schlüssel, des verschlüsselten ersten kryptographischen Schlüssels;
und

der Ein-/Ausgabebereich ist ferner konfiguriert, um den entschlüsselten ersten kryptografischen Schlüssel bereitzustellen.

15. Computerlesbares Medium, auf dem computerausführbare Anweisungen zur Durchführung computerbezogener Sicherheit unter Einbeziehung menschlicher Beteiligung gespeichert sind, durch Durchführung des Verfahrens nach einem der Ansprüche 1 bis 7.

Revendications

1. Procédé de sécurité informatique, impliquant une participation humaine, le procédé comprenant les étapes ci-dessous consistant à :

recevoir une demande concernant des données stockées sur un support de stockage ;
en réponse à la demande, transmettre (48), d'un agent client (14) à un serveur (16), une demande de casse-tête et une étiquette de récupération, dans lequel l'étiquette de récupération est pertinente pour permettre la récupération d'un grand objet binaire de vérification comprenant une solution chiffrée à un casse-tête ;
recevoir (56), au niveau de l'agent client (14), en provenance du serveur (16), le casse-tête, dans lequel la participation humaine a pour but d'obtenir une solution au casse-tête, et une solution correcte au casse-tête ;
transmettre (58) le casse-tête, de l'agent client (14) à un utilisateur (12) ;
recevoir (60), en provenance de l'utilisateur (12), une solution proposée au casse-tête ;
transmettre (62) à l'utilisateur, une requête demandant l'identification d'un secret stocké ;
recevoir (64), en provenance de l'utilisateur, une réponse à la requête demandant l'identification du secret stocké ;
transmettre (66), au serveur, la solution proposée au casse-tête et la réponse à la requête ;
utiliser (68), dans le serveur, l'étiquette de récupération reçue en provenance de l'agent client, en vue de récupérer le grand objet binaire de vérification ;
déterminer si la solution proposée et la réponse à la requête sont correctes ; et
si la solution proposée et la réponse à la requête sont correctes, permettre l'accès aux données stockées sur le support de stockage.

2. Procédé selon la revendication 1, comprenant en outre l'étape consistant à, si une durée s'écoulant entre la fourniture du casse-tête et la réception de la solution proposée est supérieure à une durée prédéterminée, ne pas permettre l'accès aux données.

3. Procédé selon la revendication 1, comprenant en outre les étapes ci-dessous consistant à :

avant de recevoir (60) la solution proposée, demander un premier secret ;
recevoir le premier secret ;
subséquemment à la demande de données, demander un second secret ;
recevoir le second secret ;
comparer le premier secret au second secret ; et
si le premier secret ne correspond pas au se-

- cond secret, ne pas permettre l'accès aux données.
4. Procédé selon la revendication 1, dans lequel :
- le casse-tête est destiné à être rendu visuellement ; et
une solution au casse-tête est destinée à être obtenue par le biais de l'observation du rendu visuel du casse-tête.
5. Procédé selon la revendication 1, dans lequel :
- le casse-tête est destiné à être rendu de façon audible ; et
une solution au casse-tête est destinée à être obtenue par le biais de l'observation du rendu audio du casse-tête.
6. Procédé selon la revendication 1, comprenant en outre l'étape consistant à obtenir le casse-tête à partir d'une bibliothèque de générateurs de casse-têtes enfichables.
7. Procédé selon la revendication 1, comprenant en outre les étapes ci-dessous consistant à :
- chiffrer les données au moyen d'une première clé cryptographique ;
chiffrer la première clé cryptographique au moyen d'une seconde clé cryptographique ;
stocker, sur le support de stockage, les données chiffrées et la première clé cryptographique chiffrée; et
si la solution proposée est correcte :
déchiffrer, au moyen de la seconde clé cryptographique, la première clé cryptographique chiffrée ; et
fournir la première clé cryptographique déchiffrée.
8. Système destiné à mettre en oeuvre une sécurité informatique impliquant une participation humaine, le système comprenant :
- une partie d'entrée/sortie configurée de manière à recevoir une demande concernant des données stockées dans une partie de mémoire ;
une première partie de traitement d'un agent client (14), la première partie de traitement étant configurée de manière à :
en réponse à la demande, transmettre (48), d'un agent client (14) à un serveur (16), une demande de casse-tête et une étiquette de récupération, dans lequel l'étiquette de récupération est pertinente pour permettre la
- recupération d'un grand objet binaire de vérification comprenant une solution chiffrée à un casse-tête ;
recevoir (56), au niveau de l'agent client (14), en provenance du serveur (16), le casse-tête, dans lequel la participation humaine a pour but d'obtenir une solution au casse-tête, et une solution correcte au casse-tête ;
transmettre (58) le casse-tête, de l'agent client (14) à un utilisateur (12) ;
recevoir, en provenance de l'utilisateur (12), une solution proposée au casse-tête ;
transmettre (62) à l'utilisateur, une requête demandant l'identification d'un secret stocké ;
recevoir (64), en provenance de l'utilisateur, une réponse à la requête demandant l'identification du secret stocké ;
transmettre (66), au serveur, la solution proposée au casse-tête et la réponse à la requête ;
- une seconde partie de traitement du serveur, la seconde partie de traitement étant configurée de manière à
utiliser (68), dans le serveur, l'étiquette de récupération reçue en provenance de l'agent client, en vue de récupérer le grand objet binaire de vérification ;
déterminer si la solution proposée et la réponse à la requête sont correctes ; et
si la solution proposée et la réponse à la requête sont correctes, permettre l'accès aux données stockées dans la partie de mémoire ; et
- dans lequel la partie de mémoire est configurée de manière à stocker les données.
9. Système selon la revendication 8, dans lequel la partie de traitement est en outre configurée de manière à, si une durée s'écoulant entre la fourniture du casse-tête et la réception de la solution proposée est supérieure à une durée prédéterminée, ne pas permettre l'accès aux données.
10. Système selon la revendication 8, dans lequel :
- la partie de traitement est en outre configurée de manière à :
avant de recevoir la solution proposée, demander un premier secret ;
subséquemment à la réception de la demande de données, demander un second secret ;

comparer le premier secret au second secret ; et
si le premier secret ne correspond pas au second secret, ne pas permettre l'accès aux données ; et

5

la partie d'entrée/sortie est en outre configurée de manière à :

recevoir le premier secret ; et
recevoir le second secret.

10

11. Système selon la revendication 8, dans lequel :

le casse-tête est destiné à être rendu visuellement ; et
une solution au casse-tête est destinée à être obtenue par le biais de l'observation du rendu visuel du casse-tête.

15

20

12. Système selon la revendication 8, dans lequel :

le casse-tête est destiné à être rendu de façon audible ; et
une solution au casse-tête est destinée à être obtenue par le biais de l'observation du rendu audio du casse-tête.

25

13. Système selon la revendication 8, dans lequel la partie de traitement est en outre configurée de manière à obtenir le casse-tête à partir d'une bibliothèque de générateurs de casse-têtes enfichables.

30

14. Système selon la revendication 8, dans lequel :

35

la partie de traitement est en outre configurée de manière à :

chiffrer les données au moyen d'une première clé cryptographique ;
chiffrer la première clé cryptographique au moyen d'une seconde clé cryptographique ;
stocker, dans la partie de mémoire, les données chiffrées et la première clé cryptographique chiffrée ; et
si la solution proposée est correcte, déchiffrer, au moyen de la seconde clé cryptographique, la première clé cryptographique chiffrée ; et

40

45

50

la partie d'entrée/sortie est en outre configurée de manière à fournir la première clé cryptographique déchiffrée.

55

15. Support lisible par ordinateur sur lequel sont stockées des instructions exécutables par ordinateur pour mettre en oeuvre une sécurité informatique im-

pliant une participation humaine, en mettant en oeuvre le procédé selon l'une quelconque des revendications 1 à 7.

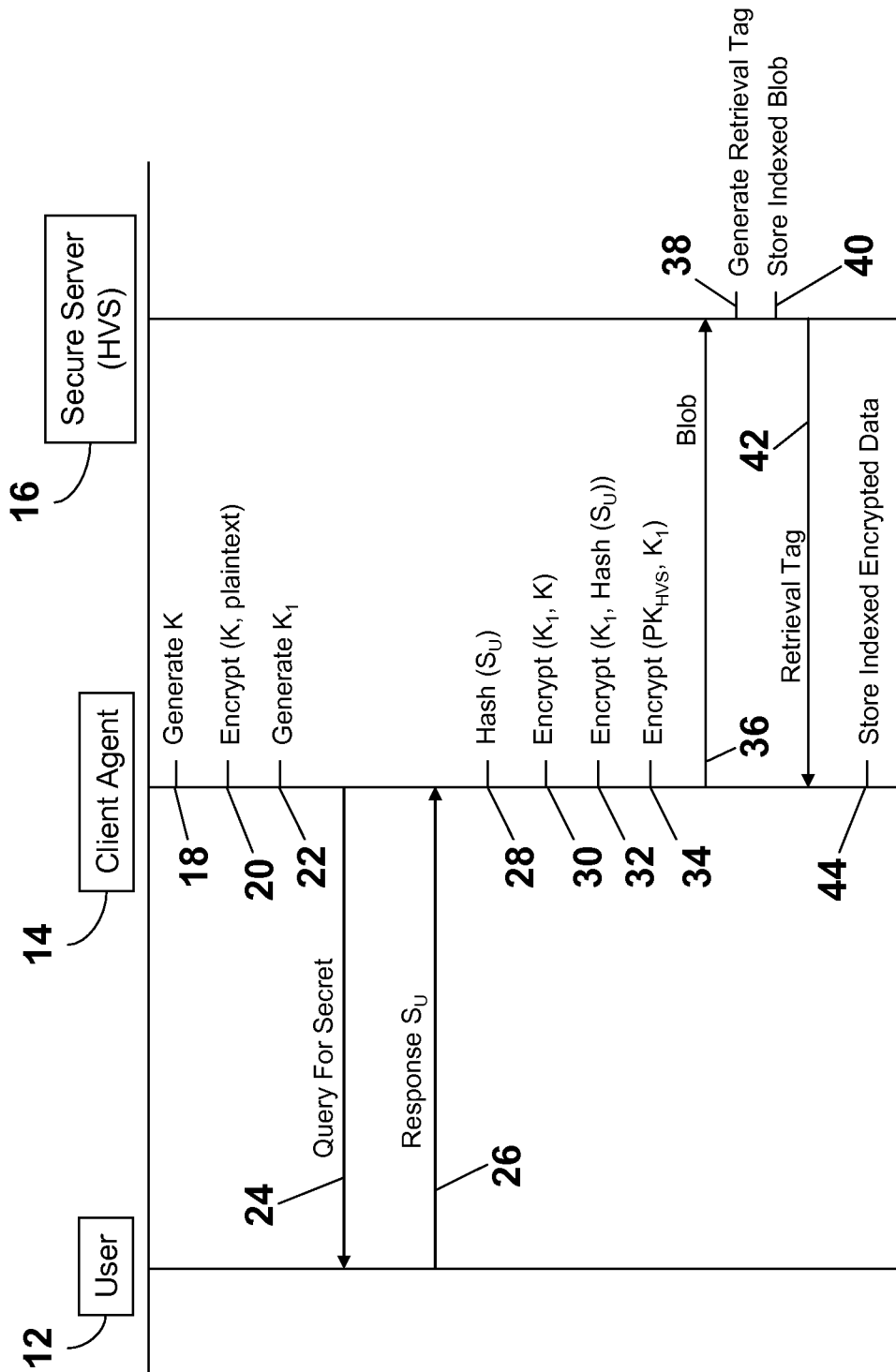


FIGURE 1

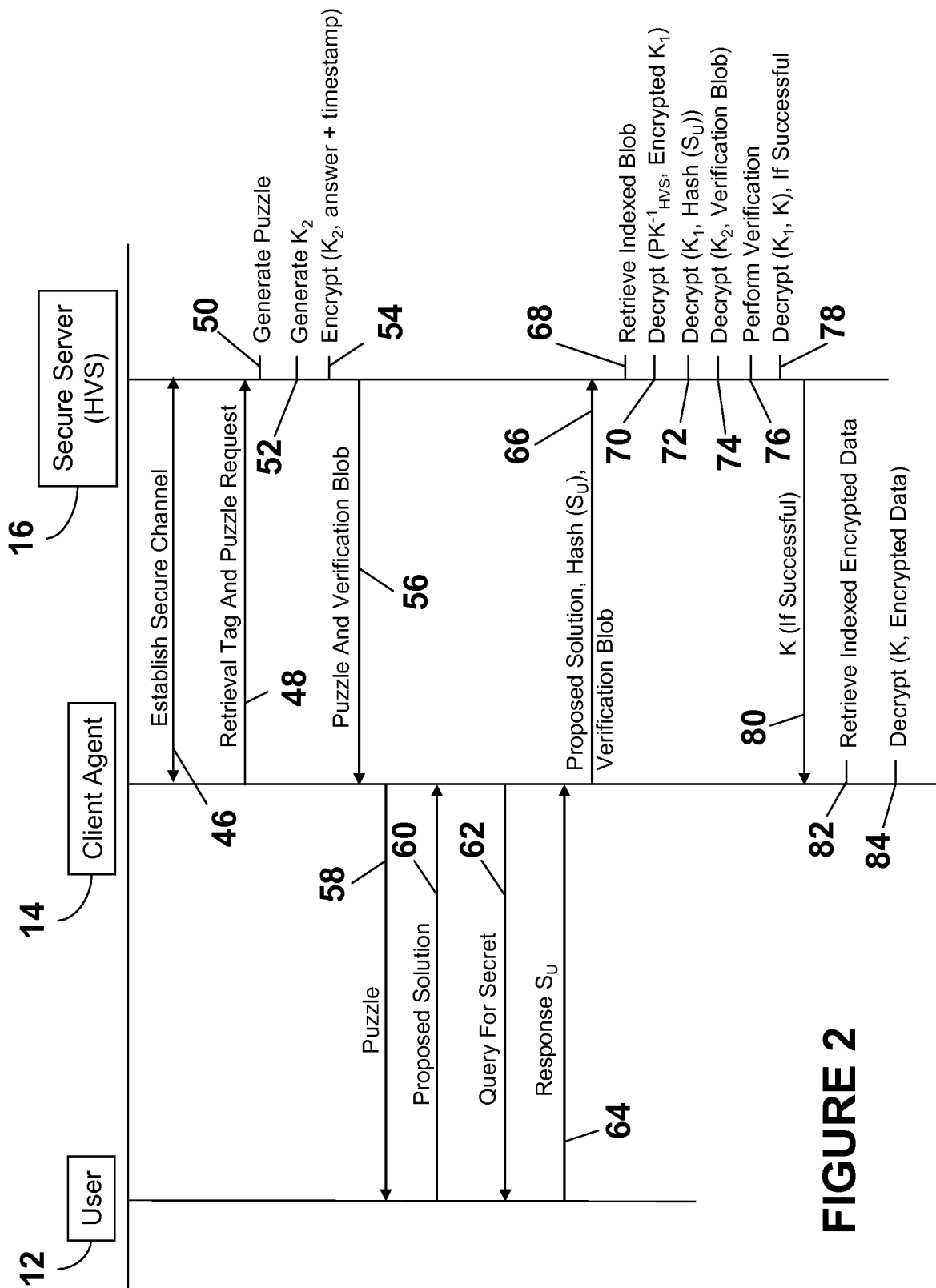


FIGURE 2

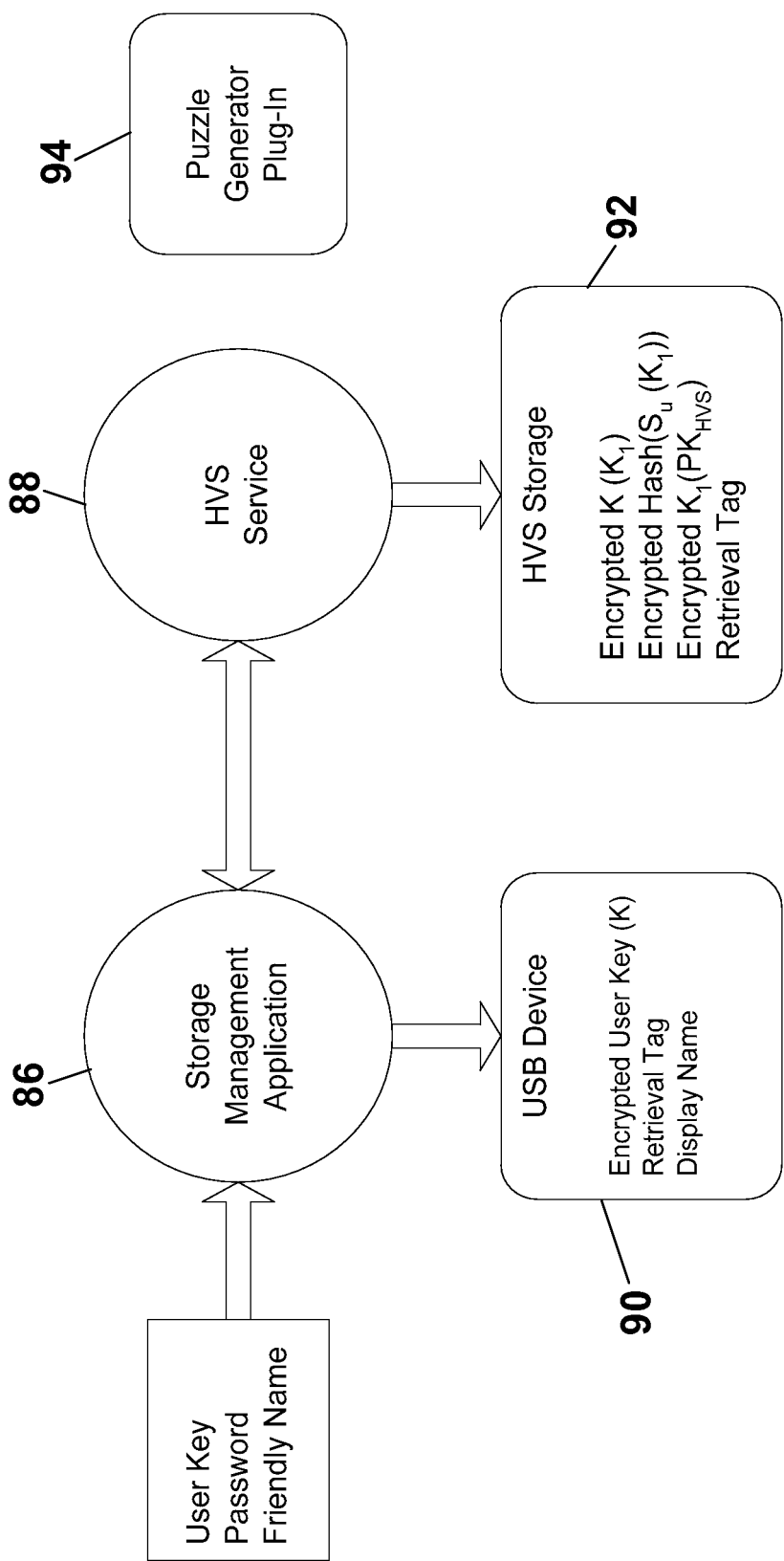


FIGURE 3

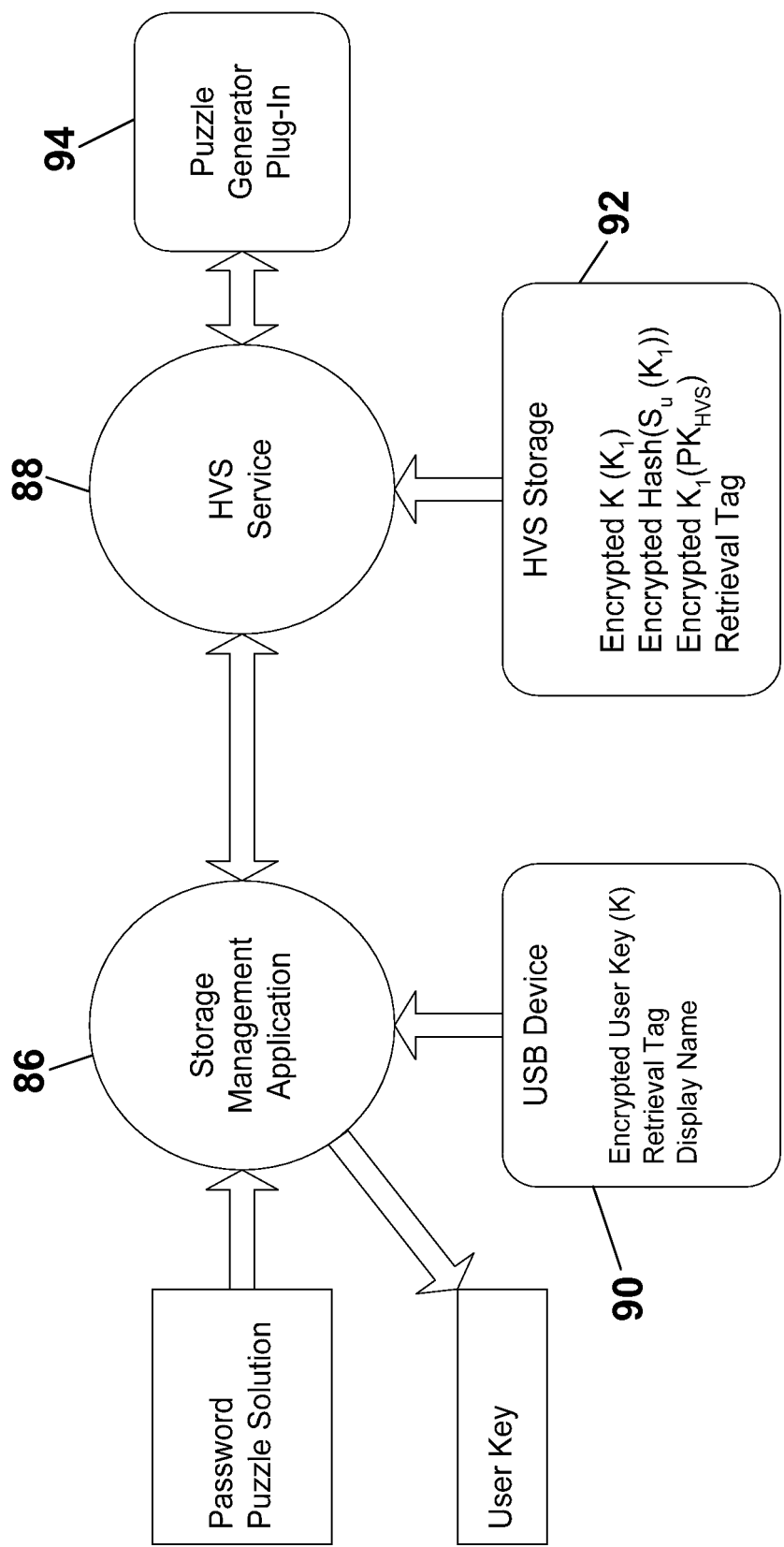


FIGURE 4

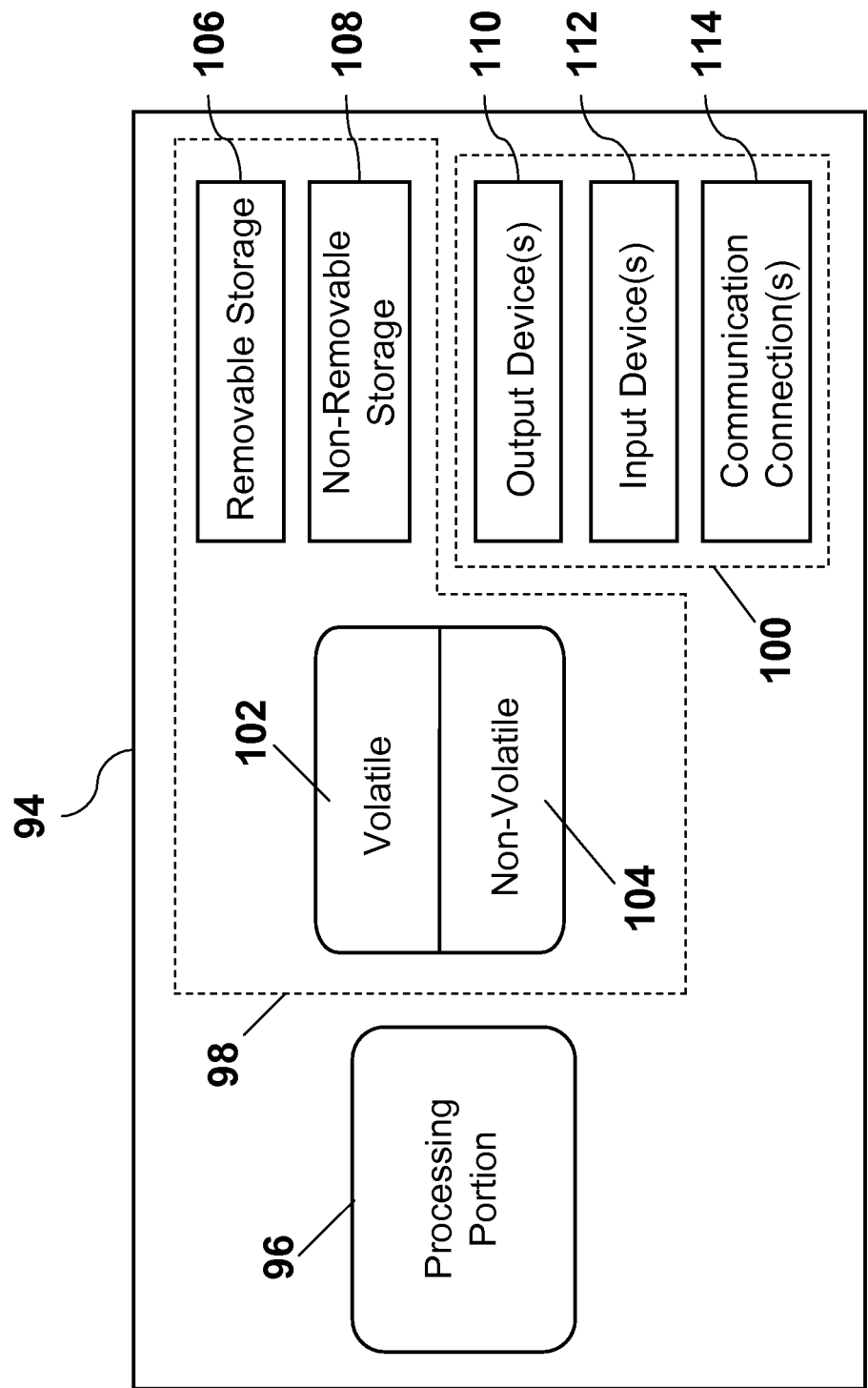


FIGURE 5

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 5921548 A [0003]
- US 6944765 B1 [0004]
- US 6754827 B1 [0006]

Non-patent literature cited in the description

- **AHN L et al.** CAPTCHA: using hard AI problems for security. *Security in Communication Networks: Third International Conference; Revised Papers/SCN 2002*, 11 September 2002 [0005]
- LECTURE NOTES IN COMPUTER SCIENCE. SPRINGER VERLAG, vol. 2656, 294-311 [0005]