

(19) **DANMARK**

(10) **DK/EP 2974162 T3**



Patent- og
Varemærkestyrelsen

(12) Oversættelse af
europæisk patentskrift

-
- (51) Int.Cl.: **H 04 L 9/08 (2006.01)** **H 04 L 12/58 (2006.01)** **H 04 L 29/06 (2006.01)**
- (45) Oversættelsen bekendtgjort den: **2019-11-11**
- (80) Dato for Den Europæiske Patentmyndigheds bekendtgørelse om meddelelse af patentet: **2019-08-07**
- (86) Europæisk ansøgning nr.: **14767599.5**
- (86) Europæisk indleveringsdag: **2014-03-12**
- (87) Den europæiske ansøgnings publiceringsdag: **2016-01-20**
- (86) International ansøgning nr.: **US2014025078**
- (87) Internationalt publikationsnr.: **WO2014151140**
- (30) Prioritet: **2013-03-15 US 201313844136**
- (84) Designerede stater: **AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR**
- (73) Patenthaver: **Thorpe, John R., 3902 Buffington Drive, Columbia, MO 65203, USA**
- (72) Opfinder: **Thorpe, John R., 3902 Buffington Drive, Columbia, MO 65203, USA**
- (74) Fuldmægtig i Danmark: **Ijon AB, Kärleksgatan 2A 5TR, 21145 Malmö, Sverige**
- (54) Benævnelse: **SYSTEM TIL IKKE-OPBEVAREDE BESKEDER**
- (56) Fremdragne publikationer:
EP-A1- 2 043 396
US-A1- 2003 147 536
US-A1- 2010 057 869
US-A1- 2010 250 691
US-A1- 2011 202 756
US-A1- 2012 303 612
US-B2- 7 523 314

DESCRIPTION

[0001] This application relates to electronic messaging. In particular, the specification relates to non-retained electronic messaging.

BACKGROUND

[0002] Existing products and systems, e.g., e-mail systems, involve sending messages through a complex network of servers such as SMTP, IMAP and POP servers. When messages are sent through these servers, copies of the messages are often stored and retained for the purposes of delivery. Even after the messages have been delivered, it is highly likely that numerous copies of the message are retained in the network, either as backups, cloud-based copies of emails, archives, inboxes, junk mail, trash items, etc. In many circumstances, especially where highly sensitive or confidential messages or documents are being transmitted, the goal is only to communicate to the receiving party and not have any of the information retained anywhere else throughout the system. Having messages or documents retained, many times permanently, means that the communicating parties have lost control of those messages and documents. Such a loss of control can lead to detrimental outcomes, including accidental disclosure of information, unwanted indications of communications and numerous other undesirable consequences.

[0003] US 2010/0057869 A1 refers to email revocation in which transmitted email can be recalled before a recipient is able to read the transmitted email is provided. An event server stores a transmitted email for a given time period or until being retrieved by a receiving email client. If the given time period expires or the email is recalled, the receiving email client is unable to retrieve the email.

SUMMARY

[0004] Objects of the present invention are achieved by subject matters of independent claims. Further exemplary embodiments of the present invention are presented in dependent claims.

[0005] The specification overcomes deficiencies and limitations of the prior art at least in part by providing a system and method for non-retained electronic messaging.

[0006] The specification describes a system, method and computer program product for non-retained electronic messaging according to some embodiments. In one embodiment, the system comprises a message receiver module, a message storing and identifier module, a message retrieval module and an expunging module. The message receiver module receives a message. The message storing and identifier generation module stores the message in a

non-transitory, non-persistent memory of one or more computing devices, generates a message identifier and sends the message identifier to a recipient device. The message retrieval module receives a selection of the message identifier from the recipient device, retrieves the message from the non-transitory, non-persistent memory and sends the message to the recipient device for presentation. The expunging module expunges the message from the one or more devices responsive to the message retrieval module sending the message to the recipient device for presentation.

[0007] In some embodiments, the expunging module expunges the message identifier from the one or more computing devices responsive to sending the message identifier to the recipient device. In some embodiments, the message identifier and message are sent anonymously based on a user preference associated with a sender of the message. In some embodiments, the message identifier is a URL. In some embodiments, the system lacks a writable, persistent memory. In some embodiments, the message identifier and message are sent to an e-mail client through a standard e-mail protocol.

[0008] In some embodiments, the system includes a key generation module for generating a globally unique key. In some embodiments, the message identifier is based at least in part on the globally unique key. In some embodiments, the expunging module expunges the globally unique key from the one or more computing devices responsive to sending the message identifier to the recipient device, and receiving the selection of the message identifier includes receiving the globally unique key.

[0009] In some embodiments, the system includes an index hashing module for generating a hashed index based at least in part on the globally unique key, and the message is stored in the non-transitory, non-persistent memory using the hashed index. In some embodiments, the index is hashed based at least in part on a device key, the device key associated with a computing device comprising the non-transitory, non-persistent memory on which the message is stored. In some embodiments, the expunging module expunges the hashed index from the one or more computing devices responsive to sending the message identifier to the recipient device.

[0010] In some embodiments, the system includes an index generation module for generating a globally unique index responsive to receiving the message. In some embodiments, the hashed index generated by the index hashing module is based at least in part on the globally unique index, the expunging module expunges the globally unique index from the one or more computing devices responsive to sending the message identifier to the recipient device, the message identifier is based at least in part on the globally unique index and receiving the selection of the message identifier includes receiving the globally unique index.

[0011] In some embodiments, the system includes a message encryption module for encrypting the message prior to storing the message in the non-transitory, non-persistent memory. In some embodiments, a key generation module generates a globally unique key, the message encryption module encrypts the message using an encryption key prior to storing the

message in the non-transitory, non-persistent memory, wherein the encryption key is based at least in part on the globally unique key, and decrypts the message retrieved from the non-transitory, non-persistent memory prior to sending the message to the recipient device for presentation, and the expunging module expunges the globally unique key and the encryption key from the one or more computing devices responsive to sending the message identifier to the recipient device, the message identifier based at least in part on the globally unique key, and wherein receiving the selection of the message identifier includes receiving the globally unique key.

[0012] In some embodiments, the expunging module sets a timer based on a user defined time period and expunges the message from the non-transitory, non-persistent memory of the one or more computing devices responsive to a failure to receive the selection of the message identifier from the recipient device within the user defined time period. In some embodiments, the expunging module sets a timer based on a system defined time period for the system and expunges the message from the non-transitory, non-persistent memory of the one or more computing devices responsive to a failure to receive the selection of the message identifier from the recipient device within the system defined time period.

[0013] The features and advantages described herein are not all-inclusive and many additional features and advantages will be apparent in view of the figures and description. Moreover, it should be noted that the language used in the specification has been principally selected for readability and instructional purposes, and not to limit the scope of the subject matter disclosed herein.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] The embodiments are illustrated by way of example, and not by way of limitation in the figures of the accompanying drawings in which like reference numerals are used to refer to similar elements.

Figure 1 illustrates a system for non-retained electronic messaging according to one embodiment.

Figure 2A is a block diagram illustrating a computing device for non-retained messaging according to one embodiment.

Figure 2B is a block diagram illustrating a non-retention message server according to one embodiment.

Figure 3 is a block diagram illustrating a non-retained messaging module according to one embodiment.

Figure 4 is a flow chart illustrating a method for non-retained electronic messaging according to one embodiment.

Figure 5 is a flow chart illustrating a method for non-retained electronic messaging according to another embodiment.

Figure 6A-6B is a flow chart illustrating a method for non-retained electronic messaging according to yet another embodiment.

DETAILED DESCRIPTION

[0015] A system and method for non-retained electronic messaging. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the embodiments. It will be apparent, however, that the embodiments can be practiced without these specific details. In other instances, structures and devices are shown in block diagram form in order to avoid obscuring the embodiments. For example, one embodiment is described below with reference to user interfaces and particular hardware. However, the present embodiments may apply to different types of computing device that can receive data and commands, and peripheral devices providing services.

[0016] Reference in the specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment.

[0017] Some portions of the detailed descriptions that follow are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers or the like.

[0018] It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms including, for example, "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities

within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0019] The present embodiments also relate to an apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general-purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, including, but not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, and magnetic disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, flash memories including USB keys with non-volatile memory or any type of media suitable for storing electronic instructions, each coupled to a computer system bus.

[0020] The embodiments can take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment containing both hardware and software elements. An exemplary embodiment is implemented in software, which includes but is not limited to firmware, resident software, microcode, etc.

[0021] Furthermore, the embodiments can take the form of a computer program product accessible from a computer-usable or computer-readable medium providing program code for use by or in connection with a computer or any instruction execution system. For the purposes of this description, a computer-usable or computer readable medium can be any apparatus that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device.

[0022] A data processing system suitable for storing and/or executing program code will include at least one processor coupled directly or indirectly to memory elements through a system bus. The memory elements can include local memory employed during actual execution of the program code, bulk storage, and cache memories which provide temporary storage of at least some program code in order to reduce the number of times code must be retrieved from bulk storage during execution.

[0023] Input/output or I/O devices (including but not limited to keyboards, displays, pointing devices, etc.) can be coupled to the system either directly or through intervening I/O controllers.

[0024] Network adapters may also be coupled to the system to enable the data processing system to become coupled to other data processing systems or remote printers or storage devices through intervening private or public networks. Modems, cable modem and Ethernet cards are just a few of the currently available types of network adapters.

[0025] Finally, the algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general-purpose systems may be used with

programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will appear from the description below. In addition, the present embodiments are not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the embodiments as described herein.

System Overview

[0026] Figure 1 illustrates a block diagram of a system 100 for non-retained electronic messaging. The illustrated system 100 includes client devices 115a, 115b, and 115n (also referred to collectively as client devices 115 or individually as client device 115) that are accessed by users 125a, 125b, and 125n (also referred to collectively as users 125 or individually as user 125), non-retained message (NRM) servers 101a, 101b, and 101c (also referred to collectively as NRM servers 101 or individually as NRM server 101), a non-retained message directory server 180, a third party server 190, and an authorization server 107. In the illustrated embodiment, these entities are communicatively coupled via a network 105. Although three client devices 115 are illustrated, any number of client devices 115 are available to any number of users 125.

[0027] The client devices 115 in Figure 1 are used by way of example. While Figure 1 illustrates three client devices 115, the present specification applies to any system architecture having one or more client devices 115. Furthermore, while only one network 105 is coupled to the client devices 115, the NRM servers 101 and the authorization server 107, in practice any number of networks 105 can be connected to the entities. Furthermore, while only one non-retained message directory server 180 is shown, the system 100 can include any number of non-retained message directory servers 180. Furthermore, while only one third party server 190 is shown, the system 100 can include any number of third party servers 190.

[0028] Furthermore, while only one authorization server 107 is shown, the system 100 can include any number of authorization servers 107. In one embodiment, the system 100 includes multiple authorization servers 107 addressed by a single URL, address or domain name. In one embodiment, the system 100 includes multiple authorization servers 107 fronted by a load balancer (not shown).

[0029] Furthermore, while Figure 1 illustrates three NRM servers 101, the present specification applies to any system architecture having one or more NRM servers 101. In one embodiment, the system 100 includes NRM servers 101 addressed by a single URL, address or domain name. In one embodiment, the system 100 includes multiple NRM servers 101 fronted by a load balancer.

[0030] In one embodiment, a non-retained messaging module 220a is included in the NRM server 101a and is operable on the NRM server 101a, which is connected to the network 105

via signal line 104. In another embodiment, the non-retained messaging module 220b is included in the NRM server 101b and is operable on the NRM server 101b, which is connected to the network 105 via signal line 106. In yet another embodiment, the non-retained messaging module 220c is included in the NRM server 101c and is operable on the NRM server 101c, which is connected to the network 105 via signal line 108. It will be recognized that the non-retained messaging module 220a/220b/220c (referred to generally as the non-retained messaging module 220) can be stored in any combination of one or more NRM servers 101. In some embodiments the non-retained messaging module 220 includes multiple, distributed modules that cooperate with each other to perform the functions described below. Details describing the functionality and components of the non-retained messaging module 220 are explained in further detail below with regard to Figure 3.

[0031] The network 105 enables communications between client devices 115, the NRM servers 101 and the authorization server 107. Thus, the network 105 can include links using technologies including, for example, Wi-Fi, Wi-Max, 2G, Universal Mobile Telecommunications System (UMTS), 3G, Ethernet, 802.11, integrated services digital network (ISDN), digital subscriber line (DSL), asynchronous transfer mode (ATM), InfiniBand, PCI Express Advanced Switching, etc. Similarly, the networking protocols used on the network 105 can include the transmission control protocol/Internet protocol (TCP/IP), multi-protocol label switching (MPLS), the User Datagram Protocol (UDP), the hypertext transport protocol (HTTP), the simple mail transfer protocol (SMTP), the file transfer protocol (FTP), lightweight directory access protocol (LDAP), Code Division Multiple Access (CDMA), Wideband Code Division Multiple Access (WCDMA), Global System for Mobile communications (GSM), High-Speed Downlink Packet Access (HSDPA), etc. The data exchanged over the network 105 can be represented using technologies and/or formats including the hypertext markup language (HTML), the extensible markup language (XML), JavaScript Object Notation (JSON), Comma Separated Values (CSV), etc. In addition, all or some of links can be encrypted using conventional encryption technologies, for example, the secure sockets layer (SSL), Secure HTTP (HTTPS) and/or virtual private networks (VPNs) or Internet Protocol security (IPsec). In another embodiment, the entities can use custom and/or dedicated data communications technologies instead of, or in addition to, the ones described above. Depending upon the embodiment, the network 105 can also include links to other networks.

[0032] In one embodiment, the network 105 is a partially public or a wholly public network, for example, the Internet. The network 105 can also be a private network or include one or more distinct or logical private networks (e.g., virtual private networks, Wide Area Networks ("WAN") and/or Local Area Networks ("LAN")). Additionally, the communication links to and from the network 105 can be wireline or wireless (i.e., terrestrial or satellite-based transceivers). In one embodiment, the network 105 is an IP-based wide or metropolitan area network.

[0033] In the illustrated embodiment, the client devices 115a, 115b and 115n are coupled to the network 105 via signal lines 108, 112 and 114, respectively. The user 125a can interact with the client device 115a. Similarly, the user 125b can interact with the client device 115b, and the user 125n can interact with the client device 115n. The NRM server 101a is

communicatively coupled to the network 105 via signal line 104. The NRM server 101b is communicatively coupled to the network 105 via signal line 106. The NRM server 101c is communicatively coupled to the network 105 via signal line 108. The authorization server 107 is communicatively coupled to the network 105 via signal line 116. In one embodiment, the authorization server 107 is communicatively coupled to data storage 130 via signal line 102. In one embodiment, the non-retained message directory server 180 is communicatively coupled to the network 105 via signal line 118. In one embodiment, the third party servers 190 is communicatively coupled to the network via signal line 122.

[0034] In one embodiment, the data storage 130 stores data and information of each user 125 of the system 100. In one embodiment, the stored data and information includes credentials associated with each user 125. Credentials may be based at least in part on one or more of what a user 125 knows (e.g., a password), what a user 125 is and what a user 125 possesses. Examples of credentials include but are not limited to a user name and/or password, a user alias, e-mail address, a biometric identifier, an electronic identifier or anything else capable of identifying a user 125 and/or an associated user account. In one embodiment, which is discussed below, a storage device 214 (see Figure 2) is included in the authorization server 107 (i.e. one embodiment of a computing device 200) and the storage device 214 stores the data and information of users 125 of the authorization server 107.

[0035] In one embodiment, a client device 115a/115b/115n is an electronic device having a messaging client 120a/120b/120n (also referred to collectively as messaging clients 120 or individually as messaging client) for interacting with the various servers 101, 107 and client devices 115 of the system 100 via the network 105. The client device 115 can be, for example, a laptop computer, a desktop computer, a tablet computer, a mobile telephone, a personal digital assistant (PDA), a mobile email device, a portable game player, a portable music player, a television with one or more processors embedded therein or coupled thereto, or any other electronic device capable of accessing a network. It will be recognized that other types of client devices 115 are possible. In one embodiment, the system 100 comprises a combination of different types of client devices 115. For example, a combination of a personal computer, a mobile phone and a tablet computer. In one embodiment, the system comprises a combination of different messaging clients 120. For example, messaging client 120a is Messaging Client A offered by Company A, messaging client 120b is Messaging Client B offered by Company B and messaging client 120c is Messaging Client C offered by Company C. In one embodiment, the client device includes a web browser (not shown). The user 125 is a human user of the client device 115.

[0036] In one embodiment, the non-retained message directory server 180 locates a NRM server 101 for storage and retrieval of a message by an NRM server 101. In one embodiment, the non-retained message directory server 180 communicates with the NRM servers 101 to determine which NRM servers will store redundant copies of a message for back-up. In one embodiment, the non-retained message directory server 180 is not a separate server, but incorporated into an NRM server 101. For example, the message back-up module 322, discussed below in reference to Figure 3, determines which NRM servers 101 will store

redundant copies of a message for back-up.

[0037] In one embodiment, the third party servers 190 is a server associated with a traditional messaging system (e.g. e-mail, instant message, social networks, micro-blogs, short message services (SMS), etc.) and provides traditional messaging services (e.g. e-mailing, instant messaging, social networking, micro-blogging, SMS messaging, etc.). In one embodiment, the third party server 190 is used by the non-retained messaging system 100 to send a message identifier (not the message itself) to a recipient. The message identifier is discussed below in reference to Figure 3. In one embodiment, messages storage and sending is exclusive to NRM servers 101 and a third party server 190 or other server (e.g. authorization server 107) is not used to send or store a message.

Example Computing Device 200

[0038] Figure 2A is a block diagram of a computing device 200 for non-retaining messaging according to one embodiment. As illustrated in Figure 2A, the computing device 200 includes a network adapter 202 coupled to a bus 204. According to one embodiment, also coupled to the bus 204 are at least one processor 206, memory 208, a graphics adapter 210, an input device 212, a storage device 214. The memory 208 stores one or more modules, which are executed by the processor 206. In one embodiment, the functionality of the bus 204 is provided by an interconnecting chipset. The computing device 200 also includes a display 218, which is coupled to the graphics adapter 210.

[0039] The processor 206 may be any general-purpose processor. The processor 206 comprises an arithmetic logic unit, a microprocessor, a general purpose controller or some other processor array to perform computations and execute code and routines. The processor 206 is coupled to the bus 204 for communication with the other components of the computing device 200. Processor 206 processes data signals and may comprise various computing architectures including a complex instruction set computer (CISC) architecture, a reduced instruction set computer (RISC) architecture, or an architecture implementing a combination of instruction sets. Although only a single processor is shown in Figure 2A, multiple processors may be included. The processing capability may be limited to supporting the display of images and the capture and transmission of images. The processing capability might be enough to perform more complex tasks, including various types of feature extraction and sampling. The computing device 200 also includes an operating system executable by the processor including but not limited to WINDOWS®, MacOS X, Android or UNIX® based operating systems. It will be recognized that other processors, operating systems, sensors, displays and physical configurations are possible.

[0040] The memory 208 is a non-transitory storage medium. The memory 208 holds instructions and/or data that may be executed by the processor 206. In one embodiment, the instructions and/or data stored on the memory 208 comprise code for performing any and/or all of the techniques described herein. The memory 208 may be a dynamic random access

memory (DRAM) device, a static random access memory (SRAM) device, flash memory or some other memory device. In one embodiment, the memory 208 also includes a non-volatile memory or similar permanent storage device and media, for example, a hard disk drive, a floppy disk drive, a CD-ROM device, a DVD-ROM device, a DVD-RAM device, a DVD-RW device, a flash memory device, or some other mass storage device known for storing information on a more permanent basis. In some embodiments, the memory 208 includes only volatile memory. The memory 208 is coupled by the bus 204 for communication with the other components of the computing device 200. In one embodiment, the computing device 200 is a NRM server 101 and a non-retained messaging module 220 is stored in memory 208 and executable by the processor 206. In one embodiment, the computing device 200 is an authorization module 107 and an authentication module 240 is stored in the memory 208 and executable by the processor 206. In one embodiment, the computing device 200 is a client device 115 and a messaging client 120 is stored in the memory 208 and executable by the processor 206.

[0041] In one embodiment, the computing device 200 is a NRM server 101 and includes a non-retained messaging module 220. The non-retained messaging module 220, which is occasionally referred to herein as a "NRM module 220," includes code and routines executable by the processor 206 for non-retained electronic messaging. In one embodiment, the non-retained messaging module 220 is a set of instructions executable by the processor 206. In another embodiment, the non-retained messaging module 220 is stored in the memory 208 and is accessible and executable by the processor 206. Details describing the functionality and components of the non-retained messaging module 220 are explained in further detail below in reference to Figure 3.

[0042] In one embodiment, the computing device 200 is an authorization server 107 and includes an authentication module 240. The authentication module 240 includes code and routines executable by the processor 206 for authenticating credentials and authorizing use of the non-retained messaging system 100. In one embodiment, the authentication module 240 is a set of instructions executable by the processor 206. In another embodiment, the authentication module 240 is stored in the memory 208 and is accessible and executable by the processor 206.

[0043] The authentication module 240 authenticates credentials and authorizes use of the non-retained messaging system 100. In one embodiment, the authentication module 240 compares user credentials provided by a user to those stored by the authorization server 107 (e.g. in a data store 130 or storage device 214 of the authorization server 107), and authenticates the user if there is a match. In one embodiment, user credentials include a username and password and the username and hashed password of each user is stored (e.g. as a flat file or relational database) in the data store 130 or storage device 214 of the authorization server 107. In one embodiment, the passwords are hashed to prevent illegitimate acquisition and exploitation of the passwords by a hacker or other nefarious user. In one embodiment, multiple authorization servers 107 are included in the non-retained messaging system 100 and the multiple authorization servers 107 share a common database of user

credentials. It will be recognized that other embodiments may include credentials other than, or different from, username and password.

[0044] In one embodiment, the computing device 200 is a client device 115 and includes a messaging client 120. The messaging client 120 includes code and routines executable by the processor 206 for sending and receiving messages over the non-retained electronic messaging system 100. In one embodiment, the messaging client 120 is a set of instructions executable by the processor 206. In another embodiment, the messaging client 120 is stored in the memory 208 and is accessible and executable by the processor 206.

[0045] A messaging client 120 may include one or more of an e-mail client, instant messaging client, or any other messaging client. For the purposes of clarity and simplification, many of the examples contained herein assume the messaging client 120 is an e-mail client. However, it will be recognized that the description may be applied to other types of messaging clients 120 as well.

[0046] In one embodiment, the user configures the messaging client 120 in much the same way as the user would for a typical messaging service. For example, in one embodiment, the sender adds an e-mail server account to the e-mail client in the same manner as any other e-mail account except the outgoing mail server for the account is the address, or domain name, of the NRM servers 101.

[0047] In one embodiment, the messaging client 120 allows the user to compose a message (e.g., including one or more of a subject, text, audio, video, images, files, attachments, etc.), identify a recipient and send the message. In one embodiment, the user interfaces for composing a message to be sent using the non-retained messaging system 100 may be identical, or nearly identical, to those for sending a traditional message using the messaging client 120. In one embodiment, the messaging client 120 formats the message the same as a message to be sent on a traditional messaging system (e.g. e-mail, instant message, etc.). For example, assume the messaging client 120 is an e-mail client; in one embodiment, the e-mail client formats the message using a standard e-mail protocol (e.g. SMTP) for sending via the non-retained messaging system 100. It will be recognized that the preceding is merely an example of a format and that others exist.

[0048] In one embodiment, the messaging client 120 receives and stores user preferences locally on the client device 115. Examples of user preferences include, but are not limited to, one or more of whether the sender of a message is identified to the recipient, a user defined time period defining a message's lifespan on NRM server(s) 101 and event from which the lifespan is measured. Some of these examples are discussed further below. It will be recognized that the preceding are merely examples and other examples of user preferences exist. In one embodiment, the messaging client 120 allows a recipient user to locally save or print a message sent via the non-retained message system 100. In one embodiment, assuming a user decides not to locally save or print a message delivered via the system 100, that message is permanently lost and unrecoverable, because messages are automatically

expunged from the system 100 after retrieval/delivery.

[0049] The storage device 214 is any device capable of holding data, like a hard drive, compact disk read-only memory (CD-ROM), DVD, or a solid-state memory device. The storage device 214 is a non-volatile memory device or similar permanent storage device and media. The storage device 214 stores data and instructions for processor 206 and comprises one or more devices including a hard disk drive, a floppy disk drive, a CD-ROM device, a DVD-ROM device, a DVD-RAM device, a DVD-RW device, a flash memory device, or some other mass storage device. In one embodiment, the storage device 214 stores data and information of a user 125. For example, in one embodiment, the computing device 200 is an authorization server 107 and the storage device 214 stores the user data and information discussed above in reference to data storage 130 (e.g. credentials). In another example, in one embodiment, the computing device 200 is a client device 115 and the storage device 214 stores messages.

[0050] The input device 212 may include a mouse, track ball, or other type of pointing device to input data into the computing device 200. The input device 212 may also include a keyboard, for example, a QWERTY keyboard, a graphical code scanner or any other physical or soft keyboard in any language. The input device 212 may also include a microphone, a web camera or similar audio or video capture device. The graphics adapter 210 displays images and other information on the display 218. The display 218 is a conventional type, for example, a liquid crystal display (LCD) or any other similarly equipped display device, screen, touchscreen or monitor. The display 218 represents any device equipped to display electronic images and data as described herein. The network adapter 202 couples the computing device 200 to a local or wide area network.

[0051] As is known in the art, a computing device 200 can have different and/or other components than those shown in Figure 2A. For example, the computing device 200 can have speakers or another form of audio output. In addition, the computing device 200 can lack certain illustrated components. For example, in one embodiment, the computing device 200 is an authorization server 107 and lacks an input device 212, graphics adapter 210 and/or display 218. Moreover, the storage device 214 can be local and/or remote from the computing device 200 (e.g., a storage area network (SAN)).

[0052] Now referring to Figure 2B, which illustrates a block diagram of a NRM server 101 according to one embodiment. In one example, the computing device 200 is an NRM server 101 and according to the illustrated one embodiment lacks an input device 212, storage device 214, graphics adapter 210 and a display 218. Furthermore, according to one embodiment, a NRM server 101 includes a non-persistent memory 207 and a persistent memory 205. The memories 205, 207 are coupled by the bus 204 for communication with the other components of the NRM server 101.

[0053] In one embodiment, the non-persistent memory 207 stores a message 230a, 230n sent using the non-retained messaging system 100 pending delivery to the recipient. In one embodiment, the non-persistent memory 207 is volatile memory. Examples of volatile memory

include, but are not limited to, dynamic random access memory (DRAM) device, a static random access memory (SRAM) device, a processor cache, etc.

[0054] In one embodiment, the NRM server 101 includes persistent memory 205 for storing the non-retained messaging module 220. Examples of persistent memory include non-volatile memory or similar permanent storage devices and media, for example, a hard disk drive, a floppy disk drive, a CD-ROM device, a DVD-ROM device, a DVD-RAM device, a DVD-RW device, a flash memory device, or some other mass storage device for storing information on a more permanent basis. In an exemplary embodiment, the persistent memory 205 is a read only memory (ROM) and incapable of storing messages sent using the non-retained messaging system 100. In one embodiment, the computing device 200 is a NRM server 101 and a non-retained messaging module 220 is stored in the persistent memory 205 and executable by the processor 206. Since non-persistent memory 207 (e.g. RAM) is not permanent and is generally more expensive and provides less capacity than persistent memory 205 (e.g. a hard disk drive), embodiments in which the NRM server 101 lacks a writable, persistent memory or persistent memory entirely may decrease the chances and disincentivize retaining messages indefinitely on the non-retained messaging system 100.

[0055] As is known in the art, the computing device 200 is adapted to execute computer program modules for providing the functionality described herein. As used herein, the term "module" refers to computer program logic utilized to provide the specified functionality. Thus, a module can be implemented in hardware, firmware, and/or software. In one embodiment, program modules are executed by the processor 206.

[0056] Embodiments of the entities described herein can include other and/or different modules than the ones described here. In addition, the functionality attributed to the modules can be performed by other or different modules in other embodiments. Moreover, this description occasionally omits the term "module" for purposes of clarity and convenience.

Example Non-Retained Messaging Module 220

[0057] Referring now to Figure 3, the non-retained messaging module 220 is shown in more detail according to one embodiment. Figure 3 is a block diagram of the non-retained messaging module 220 included in a NRM server 101.

[0058] In one embodiment, the non-retained messaging module 220 comprises a communications interface 302, a message receiver module 304, a message storing and identifier generation module 318, a message retrieval module 322 and an expunging module 324. In some embodiments, the non-retained messaging module 220 optionally also includes one or more of an authentication request module 306, a key generation module 308, an index generation module 310, an index hashing module 312, an encryption key generation module 314, a message encryption module 316 and a message back-up module 320.

[0059] It will be recognized that the modules 302, 304, 306, 308, 310, 312, 314, 316, 318, 320, 322, 324 comprised in the non-retained messaging module 220 are not necessarily all on the same NRM server 101. In one embodiment, the modules 302, 304, 306, 308, 310, 312, 314, 316, 318, 320, 322, 324 are distributed across multiple NRM servers 101. For example, in one embodiment, the message back-up module 316 is included in NRM server 101a and the other modules 302, 304, 306, 308, 310, 312, 314, 318, 320, 322 and 324 are included in NRM server 101b. It will be recognized that the preceding is merely an example of distributing modules across multiple NRM servers 101 and that other examples exist.

[0060] The communication interface 302 includes code and routines for handling communications between the message receiver module 304, the authentication request module 306 (depending on the embodiment), the key generation module 308 (depending on the embodiment), the index generation module 310 (depending on the embodiment), the index hashing module 312 (depending on the embodiment), the encryption key generation module 314 (depending on the embodiment), the message encryption module 316 (depending on the embodiment), the message storing and identifier generation module 318, the message back-up module 320 (depending on the embodiment), the message retrieval module 322, the expunging module 324 and other components of the NRM server 101. In one embodiment, the communication interface 302 is a set of instructions executable by the processor 206. In another embodiment, the communication interface 302 is stored in the persistent memory 205 and is accessible and executable by the processor 206. In either embodiment, the communication interface 302 is adapted for cooperation and communication with the processor 206, other components of the NRM server 101 and other components of the non-retained messaging module 220.

[0061] The communication interface 302 handles communications between the message receiver module 304, the authentication request module 306 (depending on the embodiment), the key generation module 308 (depending on the embodiment), the index generation module 310 (depending on the embodiment), the index hashing module 312 (depending on the embodiment), the encryption key generation module 314 (depending on the embodiment), the message encryption module 316 (depending on the embodiment), the message storing and identifier generation module 318, the message back-up module 320 (depending on the embodiment), the message retrieval module 322, the expunging module 324 and other components of the NRM server 101. For example, in one embodiment, the communication interface 202 communicates with the key generation module 308 and the index hashing module 312 to pass the output of the key generation module 308 (i.e. a globally unique key) to the index hashing module 312. However, this description may occasionally omit mention of the communication interface 302 for purposes of clarity and convenience. For example, for purposes of clarity and convenience, the above scenario may be described as the key generation module 308 passing the globally unique key to the index hashing module 312.

[0062] The message receiver module 304 includes code and routines for receiving a message. In one embodiment, the message receiver module 304 is a set of instructions executable by the processor 206. In another embodiment, the message receiver module 304 is stored in the

persistent memory 205 and is accessible and executable by the processor 206. In either embodiment, the message receiver module 304 is adapted for cooperation and communication with the processor 206, other components of the NRM server 101 and other components of the non-retained messaging module 220.

[0063] The message receiver module 304 receives a message. In one embodiment, the message receiver module 304 receives a message from a sending user's messaging client 120. For simplicity and clarity, a user 125 sending a message is occasionally referred to as a "sender." For example, the message receiver module 304 is communicatively coupled to receive the message from the messaging client 120 of a sender's client device 115 via the network 105.

[0064] A messaging client 120 may include one or more of an e-mail client, instant messaging client or any other messaging client. In one embodiment, the message receiver module 304 receives a message from a messaging client 120 with little-to-no modification to the messaging client 120. For example, in one embodiment, the message receiver module 304 receives messages from an e-mail client. Examples of little modification include installation of a plug-in, add-on, expansion pack, etc. It will be recognized that the preceding examples are merely examples of existing, commercially available e-mail clients and that other examples of messaging clients and e-mail clients exist.

[0065] In one embodiment, the message receiver module 304 receives a message including a recipient identifier and a message corpus. The recipient identifier is a unique identifier associated with the intended recipient of the sender's message. Examples of a recipient identifier include, but are not limited to, e-mail addresses, phone numbers, user names or any other identifier associated with a user and unique within the non-retained messaging system 100. The corpus of a message includes the content, which the sender wishes to communicate to the recipient. The message corpus may include, e.g., one or more of text, audio, video, images, files, attachments, etc.

[0066] In one embodiment, the received message has a format identical to that of a message sent using a traditional messaging system. For example, assume the messaging client 120 is an e-mail client; in one embodiment, the message receiver module 304 receives a message formatted using a standard e-mail protocol (e.g. SMTP). It will be recognized that the preceding is merely an example of a format and that others exist.

[0067] In one embodiment, the message receiver module 304 passes the received message to the message storing and identifier generation module 318. For example, the message receiver module 304 is communicatively coupled to the message storing and identifier generation module 318 to send the received message to the message storing and identifier generation module 318. In another embodiment, the message receiver module 304 passes the received message to the message encryption module 316. For example, the message receiver module 304 is communicatively coupled to the message encryption module 316 to send the received message to the message encryption module 316.

[0068] In some embodiments, it may be desirable to authenticate users. For example, it may be desirable to authenticate a user in order for the user to access the system 100 and/or a feature or functionality thereof. For example, it may be desirable to authenticate the user prior to one or more of composing a message, sending a message, sending a message identifier, etc. In one such embodiment, the non-retained messaging module 220 includes an optional authentication request module 306.

[0069] The authentication request module 306 includes code and routines for requesting user authentication. In one embodiment, the authentication request module 306 is a set of instructions executable by the processor 206. In another embodiment, the authentication request module 306 is stored in the persistent memory 205 and is accessible and executable by the processor 206. In either embodiment, the authentication request module 306 is adapted for cooperation and communication with the processor 206, other components of the NRM server 101 and other components of the non-retained messaging module 220.

[0070] The authentication request module 306 requests user authentication. In some embodiments, user authentication is based on credentials. In one embodiment, the authentication request module 306 requests user authentication from an authorization server 107. For example, assume that the NRM server 101 does not store credentials, because, e.g., the NRM server 101 lacks a storage device 214 and writeable persistent memory 205.

[0071] In one embodiment, the authentication request module 306 requests the user's credentials and passes the credentials, received from the user 125, to the authorization server 107 as part of the request for user authentication. In another embodiment, the authentication request module 306 passes a request for user authentication to the authorization server 107, and the authorization server 107 requests and receives the user's credentials. In either embodiment, the authorization server 107 determines whether the user is authorized based at least in part on the credentials and notifies the authentication request module 306. For example, the authorization server 107 determines whether the user is authorized based on whether a username and password provided by the user match a username and associated password stored by the authorization server 107 and notifies the authentication request module 306 whether the user is authenticated or not.

[0072] In some embodiments, one or more of the modules of the non-retained messaging module 220 execute subject to user authentication. For example, in one embodiment, the message receiver module 304 executes pending user authentication of the sending user. In another example, in one embodiment, the message storing and identifier generation module 318 executes pending user authentication of the sending user.

[0073] In one embodiment, the authentication request module 306 passes the user authentication to one or more of the other modules of the non-retained messaging module 220. For example, the authentication request module 306 is communicatively coupled to one or more of the other modules of the non-retained messaging module 220 to send the user

authentication to one or more of the other modules of the non-retained messaging module 220.

[0074] The optional key generation module 308 includes code and routines for generating a globally unique key for each message. In one embodiment, the key generation module 308 is a set of instructions executable by the processor 206. In another embodiment, the key generation module 308 is stored in the persistent memory 205 and is accessible and executable by the processor 206. In either embodiment, the key generation module 308 is adapted for cooperation and communication with the processor 206, other components of the NRM server 101 and other components of the non-retained messaging module 220.

[0075] The key generation module 308 generates a globally unique key for each message. A globally unique key is a single unique object that is unique in the world across all computing devices. For example, in one embodiment, the globally unique key is a random 128 bit number, which has 2^{128} possibilities (approximately 3.48×10^{38}) and, therefore, extremely unlikely have conflicts or be guessed. In another example, the globally unique key is generated similar to a Globally Unique Identifier (GUID).

[0076] In one embodiment, the key generation module 308 also generates a device key. A device key is a globally unique key. In one embodiment, the device key is extremely large so that the device key is virtually impossible to be guessed or figured out. For example, in one embodiment, the device key is a random 128 bit number, which has 2^{128} possibilities (approximately 3.48×10^{38}) and, therefore, extremely unlikely have conflicts or be guessed. In another example, the device key is generated similar to a Globally Unique Identifier (GUID). In one embodiment, the device key is known only to the NRM server 101 associated with the device key. For example, in one embodiment, the key generation module 308 of NRM server 101a generates a device key associated with and known only by NRM server 101a, and the key generation module 308 of NRM server 101b generates a device key associated with and known only by NRM server 101b. In one embodiment, the device key is associated with a NRM server 101, but known by at least one other NRM server 101.

[0077] In one embodiment, the device key is dynamic. For example, in some embodiments, the key generation module 308 generates a new device key each time at start-up of the NRM server 101 or after detecting an (un)authorized access and expunging the non-persistent memory of all messages, keys, indexes, etc. In an alternative embodiment, the device key may be a static, unique key assigned by the manufacturer. Regardless of whether the device key is static or dynamic, in some embodiments, each copy of a message that may exist on multiple NRM servers 101 (e.g. for back-up) may have a different hashed index and encryption key for each copy of the same message on the various NRM servers 101, because each NRM server 101 is associated with a different device key.

[0078] In one embodiment, the key generation module 308 passes the globally unique key to one or more of the index hashing module 312, the encryption key generation module 314 and

the message storing and identifier generation module 318. For example, the key generation module 308 is communicatively coupled to one or more of the index hashing module 312, the encryption key generation module 314 and the message storing and identifier generation module 318 to send the globally unique key to one or more of the index hashing module 312, the encryption key generation module 314 and the message storing and identifier generation module 318.

[0079] In one embodiment, the key generation module 308 passes the device key to one or more of the index hashing module 312, the encryption key generation module 314 and the message storing and identifier generation module 318. For example, the key generation module 308 is communicatively coupled to one or more of the index hashing module 312, the encryption key generation module 314 and the message storing and identifier generation module 318 to send the device key to one or more of the index hashing module 312, the encryption key generation module 314 and the message storing and identifier generation module 318.

[0080] The optional index generation module 310 includes code and routines for generating a globally unique index. In one embodiment, the index generation module 310 is a set of instructions executable by the processor 206. In another embodiment, the index generation module 310 is stored in the persistent memory 205 and is accessible and executable by the processor 206. In either embodiment, the index generation module 310 is adapted for cooperation and communication with the processor 206, other components of the NRM server 101 and other components of the non-retained messaging module 220.

[0081] The optional index generation module 310 generates a globally unique index for each message. Generating a globally unique index is optional and the non-retained message system 100 works and is secure without a globally unique index. However, in one embodiment, generating a globally unique index may increase the amount of effort necessary to locate and decrypt a message thereby adding further security to the system.

[0082] In one embodiment, the index generation module 310 passes the globally unique index to the index hashing module 312. For example, the index generation module 310 is communicatively coupled to the index hashing module 312 to send the globally unique index to the index hashing module 312.

[0083] The optional index hashing module 312 includes code and routines for generating a hashed index. In one embodiment, the index hashing module 312 is a set of instructions executable by the processor 206. In another embodiment, the index hashing module 312 is stored in the memory 208 and is accessible and executable by the processor 206. In either embodiment, the index hashing module 312 is adapted for cooperation and communication with the processor 206, other components of the NRM server 101 and other components of the non-retained messaging module 220.

[0084] The index hashing module 312 generates a hashed index. In one embodiment, the

index hashing module 312 generates a hashed index based on a globally unique key. For example, in one embodiment, the index hashing module 312 generates a hashed index by hashing the globally unique key. In one embodiment, the index hashing module 312 generates a hashed index based on a globally unique key and a device key. For example, in one embodiment, the index hashing module 312 generates a hashed index by hashing the globally unique key as the salt and the device key.

[0085] In one embodiment, the index hashing module 312 generates a hashed index based on the globally unique key received from the key generation module 308 and the globally unique index received from the index generation module 310. For example, in one embodiment, the index hashing module 312 generates a hashed index by hashing the globally unique key as the salt and the globally unique index. For example, in another embodiment, the index hashing module 312 generates a hashed index by hashing the globally unique key as the salt in combination with the globally unique index and device key.

[0086] In one embodiment, the index hashing module 312 passes the hashed index to the message storing and identifier generation module 318. For example, the index hashing module 312 is communicatively coupled to the message storing and identifier generation module 318 to send the hashed index to the message storing and identifier generation module 318.

[0087] The encryption key generation module 314 includes code and routines for generating an encryption key. In one embodiment, the encryption key generation module 314 is a set of instructions executable by the processor 206. In another embodiment, the encryption key generation module 314 is stored in the persistent memory 205 and is accessible and executable by the processor 206. In either embodiment, the encryption key generation module 314 is adapted for cooperation and communication with the processor 206, other components of the NRM server 101 and other components of the non-retained messaging module 220.

[0088] The encryption key generation module 314 generates an encryption key. In some embodiments, the encryption key generation module 314 generates an encryption key for a message based on the globally unique key associated with that message. Therefore, in some embodiments, the encryption key is unique for each message.

[0089] In one embodiment, the encryption key generation module 314 generates an encryption key based on the globally unique key. For example, in one embodiment, the encryption key generation module 314 generates an encryption key using the globally unique key. In one embodiment, the encryption key generation module 314 generates an encryption key based on the globally unique key and the device key. For example, in one embodiment, the encryption key generation module 314 generates an encryption key by combining the globally unique key and the device key, or using the device key as the encryption key and the globally unique key as the initialization vector for the encryption.

[0090] In some embodiments, which include both the index hashing module 312 and the encryption key generation module 314, the encryption key generation module 314 generates

an encryption key using a process different from that the index hashing module 312 uses to generate the hashed index. For example, in one embodiment, the encryption key generation module 314 generates the encryption key using the globally unique key in combination with the device key and the index hashing module 312 generates a hashed index by hashing the globally unique key as the salt combined with the globally unique index and device key.

In one embodiment, the encryption key generation module 314 passes the encryption key to the message encryption module 316. For example, the encryption key generation module 314 is communicatively coupled to the message encryption module 316 to send the encryption key to the message encryption module 316.

[0091] The optional message encryption module 316 includes code and routines for encrypting a message. In one embodiment, the message encryption module 316 is a set of instructions executable by the processor 206. In another embodiment, the message encryption module 316 is stored in the persistent memory 205 and is accessible and executable by the processor 206. In either embodiment, the message encryption module 316 is adapted for cooperation and communication with the processor 206, other components of the NRM server 101 and other components of the non-retained messaging module 220.

[0092] The message encryption module 316 optionally encrypts the message received by the message receiver module 304. In one embodiment, the encryption module 316 encrypts the message received by the message receiver module 304 using the encryption key generated by, and received from, the encryption key generation module 314. In another embodiment, encryption module 316 encrypts the message using a different encryption key.

In one embodiment, the unencrypted message is deleted from the non-persistent memory 207 responsive to encryption. For example, in one embodiment, the unencrypted message is expunged by the expunging module 324 responsive to encryption. In one embodiment, the encryption module 316 decrypts a message retrieved by the message retrieval module 322.

[0093] In one embodiment, the message encryption module 316 passes the encrypted message to the message storing and identifier generation module 318 for storage in the non-persistent memory. For example, the message encryption module 316 is communicatively coupled to the message storing and identifier generation module 318 to send the encrypted message to the message storing and identifier generation module 318.

[0094] The message storing and identifier generation module 318 includes code and routines for storing a message, generating an identifier and sending the identifier to a recipient. In one embodiment, the message storing and identifier generation module 318 is a set of instructions executable by the processor 206. In another embodiment, the message storing and identifier generation module 318 is stored in the persistent memory 205 and is accessible and executable by the processor 206. In either embodiment, the message storing and identifier generation module 318 is adapted for cooperation and communication with the processor 206, other components of the NRM server 101 and other components of the non-retained messaging module 220.

[0095] The message storing and identifier generation module 318 stores the message. In one embodiment, the message storing and identifier generation module 318 is in the non-persistent memory 207 of an NRM server 101. In one embodiment, the message storing and identifier generation module 318 receives the hashed index generated by the index hashing module 312 and stores the message using the hashed index as a handle for storing and retrieving the message. Such an embodiment beneficially provides an obfuscated index for storing the message. In one embodiment, the message stored by the message storing and identifier generation module 318 is an encrypted version of the message.

[0096] The message storing and identifier generation module 318 generates a message identifier. The message identifier is a unique identifier having an enormous number of potential values so that it is virtually impossible to guess or iterate through to discover a valid identifier especially since a message is not retained indefinitely in the system 100. The message identifier is uniquely associated with a message stored in the non-persistent memory 207 of at least one NRM server 101. In one embodiment, the message identifier is a URL to the non-retained messaging system 100.

[0097] In embodiments where a globally unique key was generated by the key generation module 308 and used by the index hashing module 312 to generate a hashed index and/or by the encryption key generation module 314 to generate an encryption key, the message identifier includes the globally unique key. For example, the message storing and identifier generation module 318 generates a URL containing the globally unique key.

[0098] In embodiments where a globally unique index was generated by the index generation module 310 and used by the index hashing module 312 to generate a hashed index, the message identifier includes the globally unique index. For example, the message storing and identifier generation module 318 generates a URL containing the globally unique key and optionally a globally unique index. In one embodiment, the URL is a non-descript HTTP URL. In one embodiment, the URL is a non-descript HTTPS URL, which may beneficially provide greater security than a HTTP URL. It will be recognized that a URL is merely one example of a message identifier and other message identifiers exist.

[0099] The message storing and identifier generation module 318 sends the message identifier to the recipient. In one embodiment, the message is not sent using a third party server 190 (e.g. those of traditional message services such as e-mail, which retains copies of the message). Instead, the message storing and identifier generation module 318 sends the message identifier using a third party server. For example, the message storing and identifier generation module 318 sends the message identifier through a standard e-mail service hosted by third party server 190. In one embodiment, the message storing and identifier generation module 318 uses a gateway service, for example, an e-mail gateway service to avoid issues with spam filters and/or to balance network load.

[0100] In some embodiments, responsive to sending the identifier, information is expunged from the non-persistent memory of the NRM server(s) 101. In some embodiments, the

information expunged from the NRM server(s) 101 ensures that the NRM server(s) do not have all the information to independently identify, locate and decrypt the message. Such embodiments may beneficially prevent a message from being accessed by anyone other than the recipient. Examples of information that may be expunged include one or more of the globally unique key, the globally unique index, the hashed index, the encryption key and the message identifier. For example, in one embodiment, the globally unique key, the globally unique index, the hashed index, the encryption key and the message identifier are expunged from the NRM server(s) 101. In some embodiments, the information expunged from the NRM server(s) 101 and the messaging identifier ensure that neither the NRM server(s) 101 nor the recipient of the messaging identifier have all the information to independently identify, locate and decrypt the message. For example, the identifier includes the globally unique key, but not the device key and the NRM server 101 does not have the globally unique key, but has the device key.

[0101] The information expunged after the identifier is sent depends on the embodiment and what information exists. For example, a globally unique index is not expunged when one was not generated (e.g. the non-retained messaging module 220 did not include the optional index generation module 310). In one embodiment, the information is expunged by the expunging module 324 discussed below.

[0102] In some embodiments, the identity of the sender may not be shared with the recipient. For example, the e-mail including the message identifier does not identify the sending user. In another example, the message retrieved by the message retrieval module 322 and presented to the recipient user does not identify the sending user. In one embodiment, whether the sending user is identified to the recipient and/or at what point is determined based on a user preference of the sender. In one embodiment, if the sending user is not identified the sender identified is the NRM server 101 containing the URL for the message.

[0103] In one embodiment, the message storing and identifier generation module 318 passes the message identifier to a third party server 190. For example, the message storing and identifier generation module 318 is communicatively coupled to the third party server 190 to send the message identifier to the recipient via the third party server 190.

[0104] The optional message back-up module 320 includes code and routines for providing redundancy. In one embodiment, the message back-up module 320 is a set of instructions executable by the processor 206. In another embodiment, the message back-up module 320 is stored in the persistent memory 205 and is accessible and executable by the processor 206. In either embodiment, the message back-up module 320 is adapted for cooperation and communication with the processor 206, other components of the NRM server 101 and other components of the non-retained messaging module 220.

[0105] In some embodiments, the configuration of the NRM server 101 makes it more likely for a message to be permanently lost prior to delivery than in a traditional messaging system (e.g. e-mail). For example, in some embodiments, the NRM server 101 lacks persistent, writable

storage and messages are stored by non-persistent memory; therefore, a disruption in power to the NRM server 101 (e.g. power outage or natural disaster) may expunge undelivered messages on that NRM server 101. In another example, in some embodiments, the NRM server 101 is configured to actively expunge all memory if the NRM server 101 is logged into in order to enhance security. Under such circumstances, the undelivered messages would also be permanently lost.

[0106] In one embodiment, the message back-up module 320 provides redundancy by sending back-up information to at least one additional NRM server 101. Such an embodiment beneficially increases the chances the message is deliverable even if a NRM server's memory is expunged. In one embodiment, back-up information includes the message received from the sender's messaging client 120. For example, the message receiver module 304 of NRM server 101a receives a message and the message back-up module 320 automatically forwards a copy of the received message to NRM server 101b where the message receiver module 304 of NRM server 101b receives the copy.

[0107] In some embodiments, when a globally unique key associated with a received message is generated by the key generation module 308, that globally unique key is back-up information and is sent by the back-up module 320 to at least one other NRM server 101. For example, in one embodiment, the message receiver module 304 of NRM server 101a receives a message, the key generation module 308 generates a globally unique key for that message and the message back-up module 320 automatically forwards a copy of the received message and the globally unique key to NRM server 101b.

[0108] In some embodiment, when a globally unique index associated with a received message is generated by the index generation module 310 and associated with a received message, that globally unique index is back-up information and is sent by the back-up module 320 to at least one other NRM server 101. For example, in one embodiment, the message receiver module 304 of NRM server 101a receives a message, the key generation module 308 generates a globally unique key for that message, the index generation module 310 generates a globally unique index for the message and the message back-up module 320 automatically forwards a copy of the received message, the globally unique key and the globally unique index to NRM server 101b.

[0109] In some embodiments, any hashed index or encryption key generated for the at least one other NRM server 101 (e.g. NRM server 101b) will be different from the hashed index or encryption key for the NRM server 101 that originally received the message (e.g. NRM server 101a) regardless of whether the same globally unique key and/or globally unique index is forwarded and used, because each NRM server 101 is associated with a different device key.

[0110] Unlike traditional messaging systems, such as e-mail, any redundant messages, also occasionally referred to herein as "back-ups," "copies" or the like are expunged from the non-retained messaging system 101 when, depending on the embodiment, the message is retrieved by the message retrieval module 322, the message is delivered for presentation to

the recipient or the lifespan of the message expires.

[0111] In one embodiment, the message back-up module 320 passes back-up information to at least one other NRM server 101. For example, the message back-up module 320 is communicatively coupled to at least one other NRM server 101 to send the back-up information to at least one other NRM server 101.

[0112] The message retrieval module 322 includes code and routines for retrieving a message. In one embodiment, the message retrieval module 322 is a set of instructions executable by the processor 206. In another embodiment, the message retrieval module 322 is stored in the persistent memory 205 and is accessible and executable by the processor 206. In either embodiment, the message retrieval module 322 is adapted for cooperation and communication with the processor 206, other components of the NRM server 101 and other components of the non-retained messaging module 220.

[0113] The message retrieval module 322 retrieves a message using the identifier. In one embodiment, the message retrieval module 322 retrieves a message using the identifier responsive to the selection of the identifier. For example, assume the message identifier is a HTTPS URL which was sent to the recipient via e-mail. In one embodiment, the recipient receives the e-mail, opens the e-mail and selects the HTTPS URL, the message retrieval module 322 receives the HTTPS URL responsive to the selection and retrieves the associated message and sends that message for presentation to the user (e.g. in a messaging client 120 or web browser (not shown) window).

[0114] Since many modules of the non-retained messaging module 220 are optional, many combinations of modules and, therefore, embodiments exist. The steps the message retrieval module 322 takes to retrieve a message vary depending on the embodiment and which, if any, optional modules (e.g. 308, 310, 312, 314 and 316) are included in the non-retained messaging module 200. For example, assume that the non-retained messaging module 220 includes an index hashing module 312; in one embodiment, the message retrieval module 322 retrieves a message using a globally unique key included in the message identifier to obtain the hashed index for retrieving the message from the non-persistent memory. In another example, assume that the non-retained messaging module 220 includes an encryption module 316; in one embodiment, the message retrieval module 322 retrieves an encrypted version of the message and must obtain a decrypted version prior to sending the message for presentation to the user.

[0115] In one embodiment, the message retrieval module 322 retrieves a message using the identifier in combination with a device key. For example, in one embodiment, the message retrieval module 322 passes the globally unique key (and, depending on the embodiment, globally unique index) from the URL to the index hashing module 312 which retrieves the device key associated with the NRM server 101 and generates the hashed index that was used to store the message. The message retrieval module 322 retrieves the message using the hashed index as a handle.

[0116] Depending on the embodiment, the message the message retrieval module 322 retrieves is encrypted and needs to be decrypted. In one embodiment, the message retrieval module 322 passes the globally unique key to the encryption key generation module 314 which retrieves the device key associated with the NRM server 101 and generates the encryption key used to decrypt the message. In one embodiment, the message retrieval module 322 decrypts the message itself. For example, the message retrieval module 322 receives the encryption key from the encryption key module 314 and decrypts the message. In another embodiment, the message encryption module 316 receives the encryption key and decrypts the message.

[0117] The message retrieval module 322 sends the message for presentation to the user based on the identifier. For example, assume the message identifier is a URL; in one embodiment, the message retrieval module 322 sends the message to the location associated with the URL for presentation to the user. In one embodiment, when the message is presented to the user, the message has a similar visual format of an e-mail. For example, the message is presented via the messaging client 120 or web browser with a subject line, message body and attachments.

[0118] In one embodiment, the message retrieval module 322 passes information included in the message identifier (e.g. a globally unique key) received from a user to one or more of the other modules (e.g. 312, 314, 316) of the non-retained messaging module 220 in order to retrieve the message and send the message for presentation. For example, the message retrieval module 322 is communicatively coupled to the index hashing module 312 to pass the received globally unique key to the index hashing module 312 in order to obtain the handle for retrieving the message (i.e. the hashed index).

[0119] In one embodiment, the message retrieval module 322 passes a message for presentation to a user. For example, the message retrieval module 322 is communicatively coupled to the messaging client 120, or web browser, of the client device 115 of the recipient to send the message to the messaging client 120, or web browser, of the client device 115 of the recipient. In one embodiment, the message retrieval module 322 passes an indication that the message has been retrieved to the expunging module 324. For example, the message retrieval module 322 is communicatively coupled to the expunging module 324 to send the indication that the message has been retrieved to the expunging module 324.

[0120] The expunging module 324 includes code and routines for expunging messages from a NRM server 101. In one embodiment, the expunging module 324 is a set of instructions executable by the processor 206. In another embodiment, the expunging module 324 is stored in the persistent memory 205 and is accessible and executable by the processor 206. In either embodiment, the expunging module 324 is adapted for cooperation and communication with the processor 206, other components of the NRM server 101 and other components of the non-retained messaging module 220.

[0121] The expunging module 324 expunges messages from a NRM server 101. In one

embodiment, the expunging module 324 expunges a message from an NRM server 101 responsive to the retrieval of the message. For example, assume the expunging module 324 receives an indication from the message retrieval module 322 that the message has been retrieved for delivery or the expunging module 324 itself detects that the message retrieval module 322 detects retrieval of the message for delivery; in one embodiment, the expunging module 324 expunges the message from the NRM server(s) 101 storing that message.

[0122] In one embodiment, the expunging module 324 expunges a message from an NRM server 101 responsive to the delivery of the message. For example, assume the expunging module 324 receives an indication from the message messaging client 120, or web browser, that the message has been received; in one embodiment, the expunging module 324 expunges the message from the NRM server(s) 101 storing that message. In one embodiment, expunging the message includes expunging sender and receiver information responsive to retrieval or delivery. In other words, in one embodiment, the non-retained messaging system 100 does not retain any sender or receiver information including logs of who sent whom a message.

[0123] In one embodiment, the expunging module 324 expunges a message from an NRM server 101 responsive to an expiration of a time period associated with the message. The expiration of a time period associated with the message is occasionally referred to herein as the "message exceeding its lifespan" or the like. In one embodiment, the time period, which is occasionally referred to herein as a message's "lifespan," is user defined. For example, assume the user specifies a time period using the messaging client 120, and the time period is stored on the client device 115 (e.g. as a user preference) and sent with each outgoing message sent using that messaging client 120; in one embodiment, the expunging module 324 receives the user defined time period and sets a timer accordingly. When the timer expires (i.e. the user defined time period has passed), the expunging module 324 expunges the message from the NRM server(s) 101 assuming the message has not already been expunged (e.g., the message was retrieved and expunged from the NRM server(s) 101 responsive to retrieval and prior to the expiration of the timer). Depending on the embodiment, the user may define a time period for each individual message or define a time period to be used for all outgoing messages unless redefined. Embodiments which provide for message expungement after a user defined time beneficially allow a user to ensure that a message is not available on the NRM server(s) 101 when the user no longer wants the message available.

[0124] In one embodiment, the time period is system defined. In one embodiment, the system defined time period includes a default used when a user defined time period has not been set. For example, does not define a message lifespan; in one embodiment, the expunging module 324 sets a default timer that is system defined. When the default timer expires, the expunging module 324 expunges the message from the NRM server(s) 101 assuming the message has not already been expunged.

[0125] In one embodiment, the system defined time period defines a maximum message lifespan. For example, in one embodiment, the expunging module 324 sets a timer that is

system defined, and when the system defined timer expires, the expunging module 324 expunges the message from the NRM server(s) 101 assuming the message has not already been expunged and regardless of whether the user defined timer has expired. Embodiments which provide for message expungement after a system defined maximum time period beneficially reduce the costs of running the NRMS system 100. For example, non-persistent memory 207 is often more expensive per byte of capacity than persistent storage; therefore, a higher memory turn-over rate is desirable, because removing messages that have not been retrieved after a certain period of time so that the non-persistent memory 207 may be used by other messages may avoid the cost of adding additional NRM servers 101 and/or non-persistent memory 207 to accommodate messages which may never be retrieved. Embodiments which provide for message expungement after a system defined maximum time period may also provide additional security to the NRMS system 100 by limiting the amount of time a hacker or other nefarious entity could potentially access the message en route from the sender to the recipient.

[0126] A time period, regardless of whether the time period is user defined or system defined, may be measured from one of a plurality of events. Examples of events include, but are not limited to receipt of the message, sending of the identifier associated with the message to the recipient, retrieval of the message and delivery of the message. Embodiments in which the time period is measured from the retrieval or delivery of the message may potentially allow a recipient another opportunity to receive the message should an error occur during retrieval or delivery of the message.

[0127] In one embodiment, the expunging module 324 expunges other information from the NRM server 101 in addition to messages. Examples of other information include, but are not limited to one or more of the globally unique key, index and message identifier, the encryption key, un-encrypted message, the sender, the recipient. For example, in one embodiment, responsive to sending the message identifier associated with a message, the expunging module 324 expunges the globally unique key and message identifier associated with that message from the NRM server 101 ensuring the NRM server lacks the necessary information to independently identify and locate the message.

[0128] In one embodiment, the expunging module 324 expunges everything from memory responsive to detecting an unauthorized access of the NRM server 101. For example, assume the NRM server 101 detects predetermined number of failed login attempts using a system administrator's username; in one embodiment, the NRM server 101 expunges everything from memory. In one embodiment, the expunging module 324 expunges everything from memory responsive to detecting an access of the NRM server 101 regardless of whether the access is authorized or unauthorized. For example, assume the NRM server 101 detects a successful system administrator login; in one embodiment, the NRM server 101 expunges everything from memory responsive to detecting the login.

[0129] The expungement impedes access to the expunged data. The expungement the expunging module 324 performs may vary depending on the embodiment. Examples of

expungement include, but are not limited to, removing handles (e.g. pointers) to the expunged data, overwriting the expunged data with new data (e.g. a new message or writing to zero) or any other method of wiping data from memory, which allows the memory to be reused.

Example Processes

[0130] Figures 4, 5 and 6A-B depict various methods 400, 500, 600 performed by the system described above in reference to Figures 1-3.

[0131] Figure 4 is a flow chart illustrating a method 400 for non-retained electronic messaging according to one embodiment. At block 402, the message receiver module 304 of the non-retained messaging module 220 receives a message from a sender's messaging client 120. At block 410, the message encryption module 316 optionally encrypts the message received at block 402. At block 412, the message storing and identifier generation module 318 stores the message in non-persistent memory 207. At block 414, the message storing and identifier generation module 318 generates and sends a message identifier associated with the message stored at step 412. At block 418, the message retrieval module 322 receives selection of the message identifier. Responsive to receiving the selection of the message identifier at block 418, the message retrieval module 322, at block 420, retrieves the message, decrypts the message if encrypted at block 410, and sends the message for presentation. At block 422, the expunging module 324 expunges the message from the non-persistent memory 207.

[0132] Figure 5 is a flow chart illustrating a method 500 for non-retained electronic messaging according to another embodiment. At block 502, the message receiver module 304 of the non-retained messaging module 220 receives a message from a sender's messaging client 120. At block 504, the key generation module 308 generates a globally unique key. At block 506, the index generation module 310 optionally generates a globally unique index. At block 508, the index hashing module generates a hashed index based at least in part on the globally unique key generated at block 504 and the globally unique index if generated at block 506. At block 510, the message encryption module 316 encrypts the message using an encryption key based at least in part on the globally unique key generated at block 504. At block 512, the message storing and identifier generation module 318 stores the encrypted message in non-persistent memory according to the hashed index generated at block 508. At block 514, the message storing and identifier generation module 318 generates and sends a message identifier which includes the globally unique key generated at block 504. When a globally unique index is generated at block 506 and used to generate the hashed index at block 508, the message identifier generated at block 514 also includes that globally unique index. At block 516, information (e.g., the globally unique key generated at block 504, the globally unique index optionally generated at block 506, the hashed index generated at block 508 and the message identifier generated at block 514) is expunged from the non-persistent memory 207 by the expunging module 324. At block 518, the message retrieval module 322 receives selection of the message identifier sent at block 514. Responsive to receiving the selection of

the message identifier, at block 518, the message retrieval module 322 retrieves, at block 520, the message and sends the message for presentation. At block 522, the expunging module 324 expunges the message from the non-persistent memory 207.

[0133] Figures 6A and 6B are flow charts illustrating a method 600 for non-retained electronic messaging according to yet another embodiment.

[0134] At block 602, the message receiver module 304 of the non-retained messaging module 220 receives a message from a sender's messaging client 120. At block 604, the authentication request module 306 requests and receives authentication of the sender from an authorization server 107. At block 606, responsive to authentication at block 604, the non-retained messaging module 220 retrieves sender preferences including a message lifespan preference and sender identification preference. At block 608, the key generation module 308 generates a globally unique key. At block 610, the index generation module 310 optionally generates a globally unique index. At block 612, the index hashing module 312 generates a hashed index based at least in part on the globally unique key generated at block 608 and the globally unique index if generated at block 610. At block 614, the message encryption module 316 encrypts the message using an encryption key based at least in part on the globally unique key generated at block 608. At block 616, the message storing and identifier generation module 318 stores the encrypted message in non-persistent memory according to the hashed index. At block 618, the expunging module 324 sets a timer associated with the message. At block 620, the message storing and identifier generation module 318 generates and sends a message identifier which includes the globally unique key generated at block 608. When a globally unique index is generated at block 610 and used to generate the hashed index at block 612, the message identifier generated at block 620 also includes that globally unique index.

[0135] Referring now to Figure 6B, at block 622, information (e.g., the globally unique key generated at block 608, the globally unique index optionally generated at block 610, the hashed index generated at block 612, the encryption key used at block 614 and the message identifier generated at block 620) is expunged from the non-persistent memory 207 by the expunging module 324.

[0136] At block 624, the message retrieval module 322 determines whether a selection of the message identifier has been received. If the message retrieval module 322 determines that a selection of the message identifier has been received (624-Yes), the method 600 continues at block 630. At block 630, the message retrieval module 322 retrieves the message and sends the message for presentation to the user. The expunging module 324 expunges, at block 632, the message from the non-persistent memory 207, and the method 600 ends.

[0137] If the message retrieval module 322 determines that a selection of the message identifier has not been received (624-No), the method 600 continues at block 626. At block 626, the expunging module 324 determines whether the user defined message lifespan has been met or exceeded. If the expunging module 324 determines that the user defined

message lifespan has been met or exceeded (626-Yes), the method 600 continues at block 632. If the expunging module 324 determines that the user defined message lifespan has not been met or exceeded (626-No), the method 600 continues at block 628.

[0138] At block 628, the expunging module 324 determines whether the system defined message lifespan has been met or exceeded. If the expunging module 324 determines that the system defined message lifespan has been met or exceeded (628-Yes), the method 600 continues at block 632. If the expunging module 324 determines that the system defined message lifespan has not been met or exceeded (628-No), the method 600 continues at block 624. At block 632, the expunging module 324 expunges the message from the non-persistent memory 207, and the method 600 ends.

[0139] The foregoing description of the embodiments has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the present embodiments to the precise forms disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the present embodiments be limited not by this detailed description, but rather by the claims of this application. As will be understood by those familiar with the art, the present embodiments may take other specific forms without departing from the scope or essential characteristics thereof. Likewise, the particular naming and division of the modules, routines, features, attributes, methodologies and other aspects are not mandatory or significant, and the mechanisms that implement one embodiment or its features may have different names, divisions and/or formats. Furthermore, as will be apparent, the modules, routines, features, attributes, methodologies and other aspects of the embodiments can be implemented as software, hardware, firmware or any combination of the three. Also, wherever a component, an example of which is a module, is implemented as software, the component can be implemented as a standalone program, as part of a larger program, as a plurality of separate programs, as a statically or dynamically linked library, as a kernel loadable module, as a device driver, and/or in every and any other way known now or in the future. Additionally, the embodiments are in no way limited to implementation in any specific programming language, or for any specific operating system or environment. Accordingly, the disclosure is intended to be illustrative, but not limiting, of the scope, which is set forth in the following claims.

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

- US20100057869A1 [0003]

Patentkrav

1. Fremgangsmåde (500, 600), der er konfigureret til at udføre en ikke-opbevaret
5 beskedanordning (101a, 220a, 101b, 220b, 101c, 220c, 101, 220), der omfatter:
at modtage (502, 602), ved anvendelse af en eller flere computeranordninger for
den ikke-opbevarede beskedanordning (101a, 220a, 101b, 220b, 101c, 220c, 101,
220), en besked fra en beskedklient (120a, 120b, 120c) fra en sendende bruger;
at generere (504, 608), ved anvendelse af den ene eller de flere
10 computeranordninger, en global unik nøgle;
at generere (508, 612), ved anvendelse af den ene eller de flere
computeranordninger, et hashed-indeks, der er baseret i det mindste delvist på den
globale unikke nøgle og en anordningsnøgle;
at generere, ved anvendelse af den ene eller de flere computeranordninger, en
15 krypteringsnøgle, der er baseret i det mindste delvist på den globale unikke nøgle;
at kryptere (510, 614), ved anvendelse af den ene eller de flere
computeranordninger, beskeden ved anvendelse af krypteringsnøglen;
at lagre (512, 616), ved anvendelse af den ene eller de flere computeranordninger,
den krypterede besked i en ikke-midlertidig, ikke-persistent hukommelse for den
20 ene eller de flere computeranordninger ved anvendelse af hash-indekset;
at generere (514, 620), ved anvendelse af den ene eller de flere
computeranordninger, en beskedidentifikator, der er baseret i det mindste delvist
på den globale unikke nøgle;
at sende (514, 620), ved anvendelse af den ene eller de flere computeranordninger,
25 beskedidentifikatoren til en modtageanordning;
at slette (516, 622), ved anvendelse af den ene eller de flere computeranordninger,
ved at fjerne den globale unikke nøgle, hash-indekset og beskedidentifikatoren fra
den ene eller de flere computeranordninger som reaktion på at sende
beskedidentifikatoren til modtageanordningen;
30 at bestemme, ved anvendelse af den ene eller de flere computeranordninger,
hvorvidt et udvalg af beskedidentifikatoren modtages;
som reaktion på at bestemme at udvalget af beskedidentifikatoren er blevet
modtaget;

- 2 -

- (a) at bestemme, ved anvendelse af den ene eller de flere computeranordninger, hash-indekset fra den globale unikke nøgle, der er modtaget som reaktion på udvælgelse af beskedidentifikatoren;
- (b) at hente, ved anvendelse af den ene eller de flere computeranordninger, den
- 5 krypterede besked fra den ikke-midlertidige, ikke-persistente hukommelse ved anvendelse af hash-indekset;
- (c) at bestemme, ved anvendelse af den ene eller de flere computeranordninger, krypteringsnøglen ved anvendelse af den globale unikke nøgle og dekryptering af den krypterede besked ved anvendelse af krypteringsnøglen;
- 10 (d) at sende beskeden til modtageanordningen til præsentation; og
- (e) at slette (522, 632), ved anvendelse af den ene eller de flere computeranordninger, ved at fjerne den krypterede besked fra den ene eller de flere anordninger som reaktion på at sende beskeden til modtageanordningen til præsentation;
- 15 som reaktion på at bestemme at udvalget af beskedidentifikatoren ikke er blevet modtaget:
- (a) at bestemme, ved anvendelse af en eller flere computeranordninger, hvorvidt en brugerdefineret beskedlevetid eller en systemdefineret beskedlevetid er overskredet; og
- 20 (b) som reaktion på at bestemme, at i det mindste en af den brugerdefinerede beskedlevetid eller den systemdefinerede beskedlevetid er overskredet, at slette (522, 632), ved anvendelse af den ene eller de flere computeranordninger, ved at fjerne den krypterede besked fra den ene eller de flere anordninger.
- 25 2. Fremgangsmåde (500, 600) ifølge krav 1, der yderligere omfatter:
- at indstille, ved anvendelse af den ene eller de flere computeranordninger, en timer, der er baseret på den brugerdefinerede tidsperiode; og
- at slette, ved anvendelse af den ene eller de flere computeranordninger, beskeden fra den ikke-midlertidige, ikke-persistente hukommelse for den ene eller de flere
- 30 computeranordninger som reaktion på en manglende modtagelse af udvælgelsen af beskedidentifikatoren fra modtageanordningen inden for den brugerdefinerede tidsperiode, der er baseret på timeren.

3. Fremgangsmåde (500, 600) ifølge krav 1, der yderligere omfatter:
at indstille, ved anvendelse af den ene eller de flere computeranordninger, en timer, der er baseret på den systemdefinerede tidsperiode; og
at slette, ved anvendelse af den ene eller de flere computeranordninger, beskeden
- 5 fra den ikke-midlertidige, ikke-persistente hukommelse for den ene eller de flere computeranordninger som reaktion på en manglende modtagelse af udvælgelsen af beskedidentifikatoren fra modtageanordningen inden for den systemdefinerede tidsperiode, der er baseret på timeren.
- 10 4. Fremgangsmåde (500, 600) ifølge krav 1, hvor beskedidentifikatoren og beskeden sendes anonymt baseret på en brugerpræference, der er forbundet med en sender af beskeden.
5. Fremgangsmåde (500, 600) ifølge krav 1, hvor beskedidentifikatoren er en
- 15 uniform resource locator, URL.
6. Fremgangsmåde (500, 600) ifølge krav 1, hvor beskedidentifikatoren og beskeden sendes til en e-mail-klient gennem en standard e-mail-protokol.
- 20 7. Fremgangsmåde (500, 600) ifølge krav 1, hvor den ene eller de flere computeranordninger mangler en skrivbar, persistent hukommelse.
8. Ikke-opbevaret beskedanordning (101a, 220a, 101b, 220b, 101c, 220c, 101, 220) der omfatter:
- 25 et beskedmodtagermodul (304) til at modtage en besked fra en beskedklient fra en sendende bruger;
et nøglegenereringsmodul (308) til at generere en global unik nøgle;
et indeks-hashing modul (312) til at generere et hashed-indeks, der er baseret i det mindste delvist på den globale unikke nøgle og en anordningsnøgle, hvor beskeden
- 30 er lagret i en ikke-midlertidig, ikke-persistent hukommelse ved anvendelse af hash-indekset;
et beskedkrypteringsmodul (316) til at generere en krypteringsnøgle, der er baseret i det mindste delvist på den globale unikke nøgle og til at kryptere beskeden ved

anvendelse af krypteringsnøglen forud for lagring af beskeden i den ikke-midlertidige, ikke-persistente hukommelse;

et beskedlagrings- og identificeringsgenereringsmodul (318) til at lagre den krypterede besked i den ikke-midlertidige, ikke-persistente hukommelse af en eller
5 flere computeranordninger, til at generere en beskedidentifikator, der er baseret i det mindste delvist på den globale unikke nøgle og til at sende beskedidentifikatoren til en modtageanordning;

et beskedhentemodul (322) til at modtage en udvælgelse af beskedidentifikatoren fra modtageanordningen, til at hente beskeden fra den ikke-midlertidige, ikke-
10 persistente hukommelse og til at sende beskeden til modtageanordningen til præsentation; og

et sletningsmodus (324) til at slette, ved at fjerne (1) den globale unikke nøgle, hash-indekset og beskedidentifikatoren fra den ene eller de flere computeranordninger som reaktion på at sende beskedidentifikatoren til
15 modtageanordningen, og til at slette, ved at fjerne (2), beskeden fra den ene eller de flere anordninger som reaktion på at sende beskeden til modtageanordningen til præsentation, hvor sletningen af beskeden sker som reaktion på en af:

(a) at bestemme, ved anvendelse af den ene eller de flere computeranordninger, hash-indekset fra den globale unikke nøgle, der er modtaget som reaktion på
20 udvælgelse af beskedidentifikatoren; at hente, ved anvendelse af den ene eller de flere computeranordninger, den krypterede besked fra den ikke-midlertidige, ikke-persistente hukommelse ved anvendelse af hash-indekset; at bestemme, ved anvendelse af den ene eller de flere computeranordninger, krypteringsnøglen ved anvendelse af den globale unikke nøgle og dekryptering af den krypterede besked
25 ved anvendelse af krypteringsnøglen; og at sende beskeden til modtageanordningen til præsentation; og

(b) at bestemme, ved anvendelse af en eller flere computeranordninger, hvorvidt en brugerdefineret beskedlevetid eller en systemdefineret beskedlevetid er overskredet.

30

9. Ikke-opbevaret beskedanordning (101a, 220a, 101b, 220b, 101c, 220c, 101, 220) ifølge krav 8, der yderligere omfatter:

en sletningsmodus (324) til at indstille en timer, der er baseret på den brugerdefinerede tidsperiode og til at slette beskeden fra den ikke-midlertidige, ikke-persistente hukommelse for den ene eller de flere computeranordninger som reaktion på en manglende modtagelse af udvælgelsen af beskedidentifikatoren fra
5 modtageanordningen inden for den brugerdefinerede tidsperiode, der er baseret på timeren.

10. Ikke-opbevaret beskedanordning (101a, 220a, 101b, 220b, 101c, 220c, 101, 220) ifølge krav 8, der yderligere omfatter:
10 en sletningsmodus (324) til at indstille en timer, der er baseret på den systemdefinerede tidsperiode for systemet og til at slette beskeden fra den ikke-midlertidige, ikke-persistente hukommelse for den ene eller de flere computeranordninger som reaktion på en manglende modtagelse af udvælgelsen af beskedidentifikatoren fra modtageanordningen inden for den systemdefinerede
15 tidsperiode, der er baseret på timeren.

11. Ikke-opbevaret beskedanordning (101a, 220a, 101b, 220b, 101c, 220c, 101, 220) ifølge krav 8, hvor beskedidentifikatoren og beskeden sendes anonymt baseret på en brugerpræference, der er forbundet med en sender af beskeden.
20

12. Ikke-opbevaret beskedanordning (101a, 220a, 101b, 220b, 101c, 220c, 101, 220) ifølge krav 8, hvor beskedidentifikatoren er en URL.

13. Ikke-opbevaret beskedanordning (101a, 220a, 101b, 220b, 101c, 220c, 101, 220) ifølge krav 8, hvor systemet mangler en skrivbar, persistent hukommelse.
25

14. Ikke-opbevaret beskedanordning (101a, 220a, 101b, 220b, 101c, 220c, 101, 220) ifølge krav 8, hvor beskedidentifikatoren og beskeden sendes til en e-mail-klient gennem en standard e-mail-protokol.

DRAWINGS

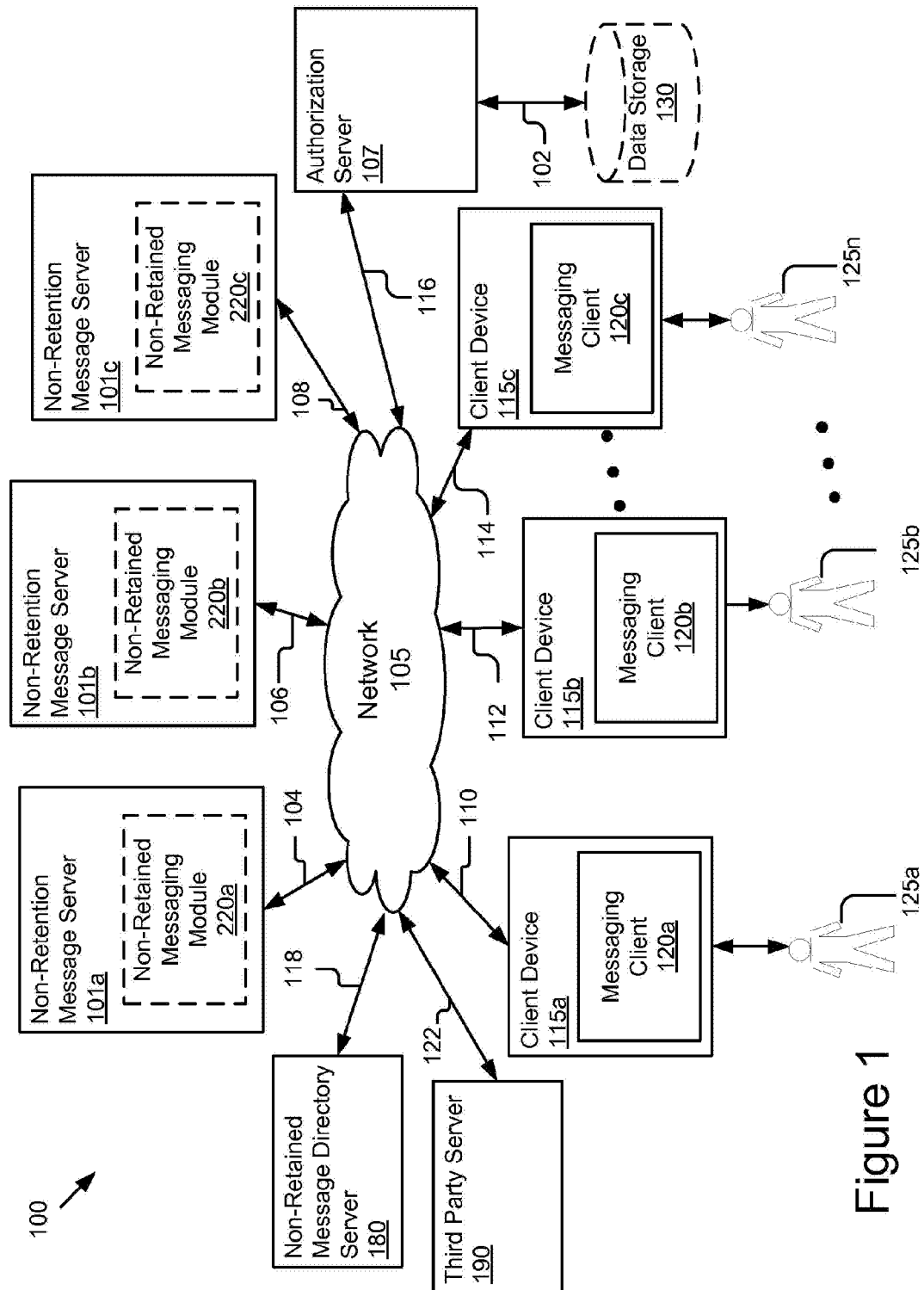


Figure 1

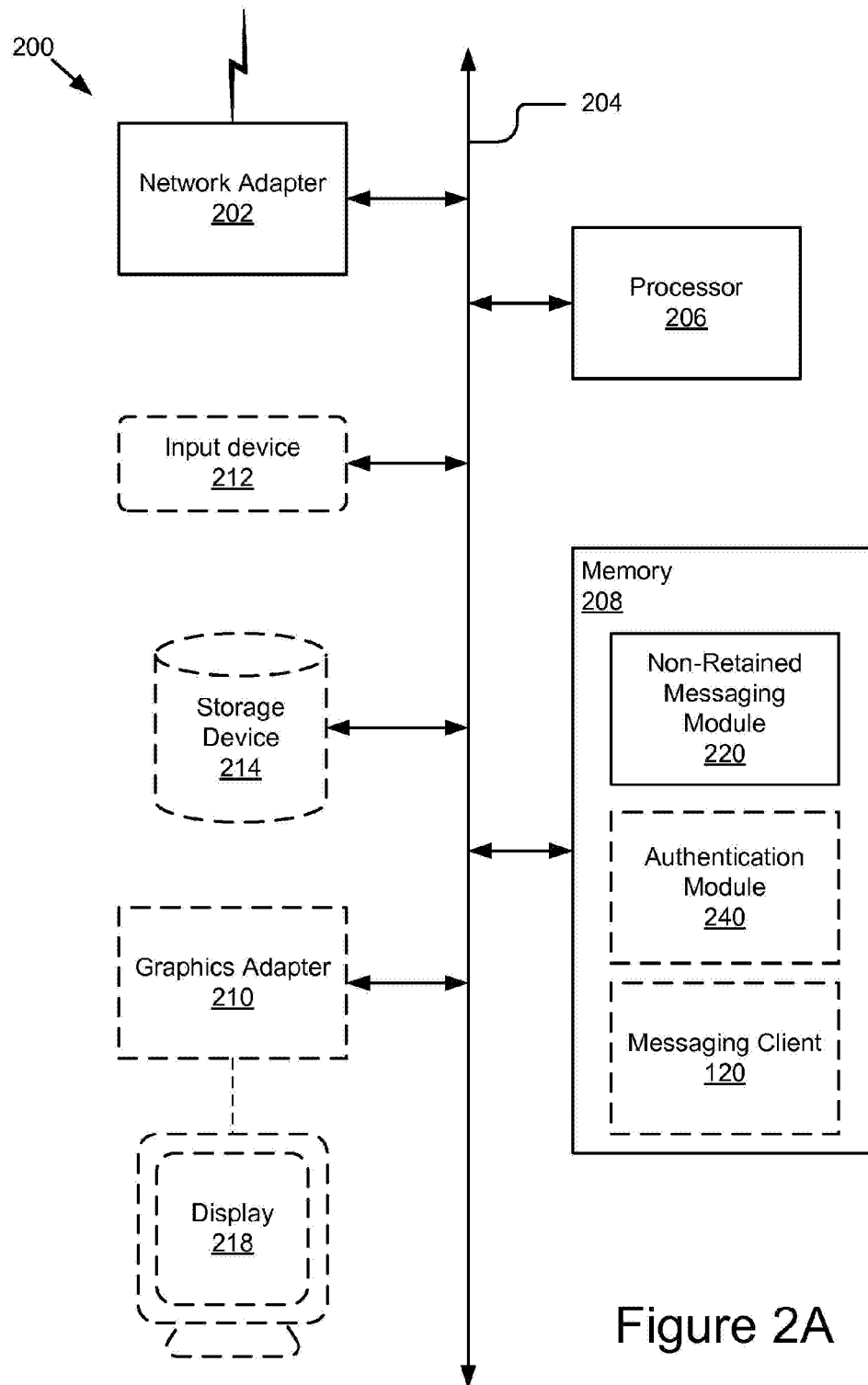
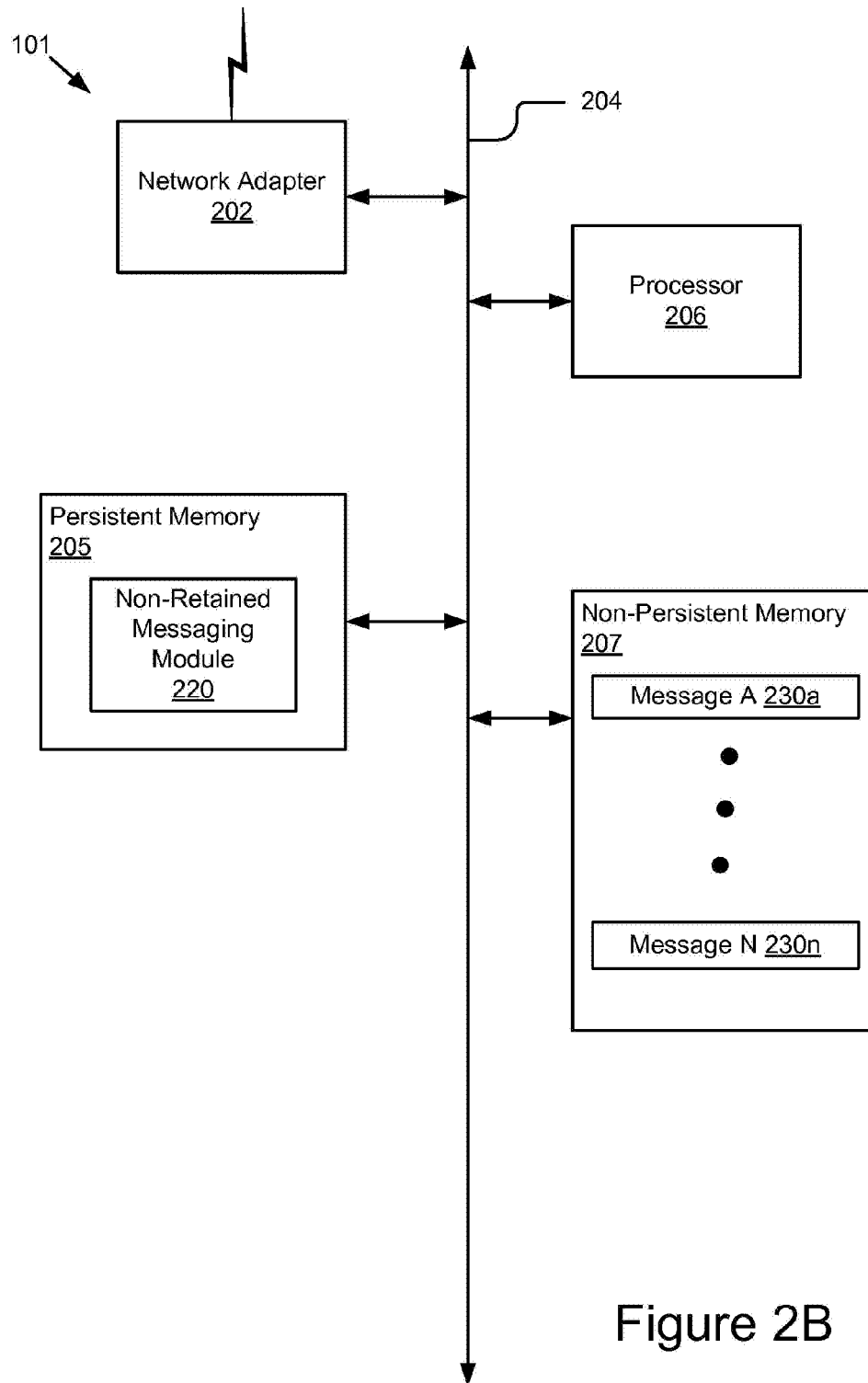


Figure 2A



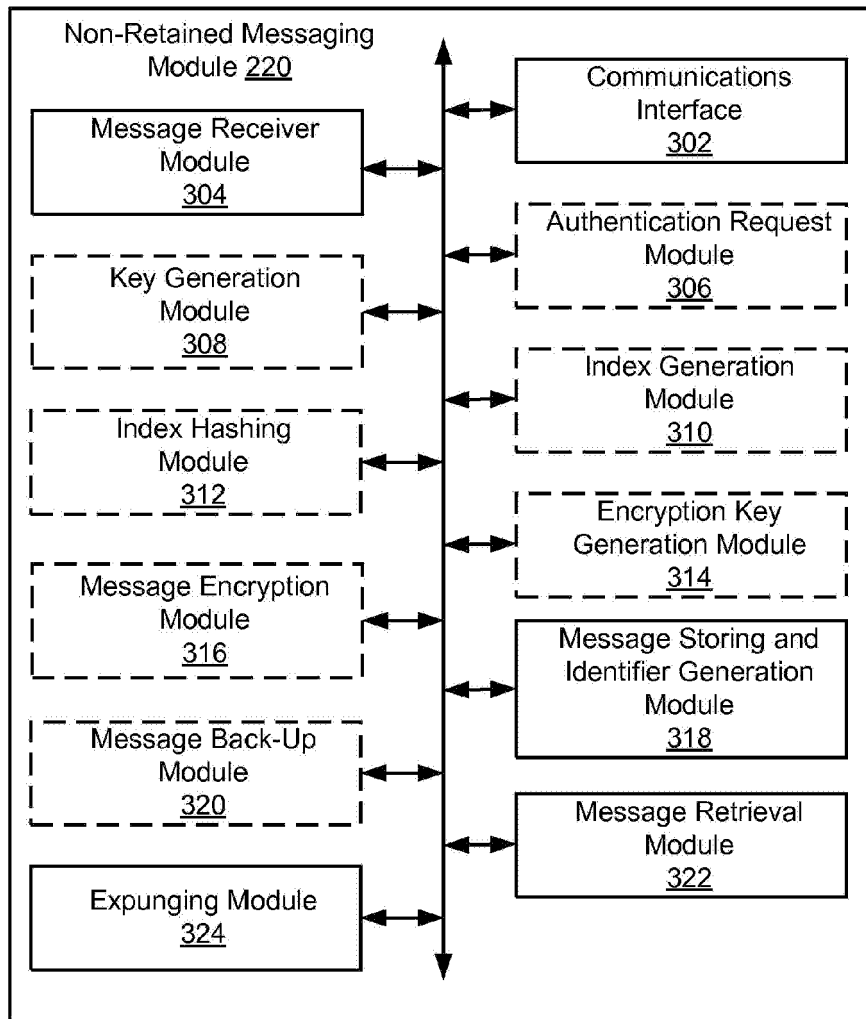


Figure 3

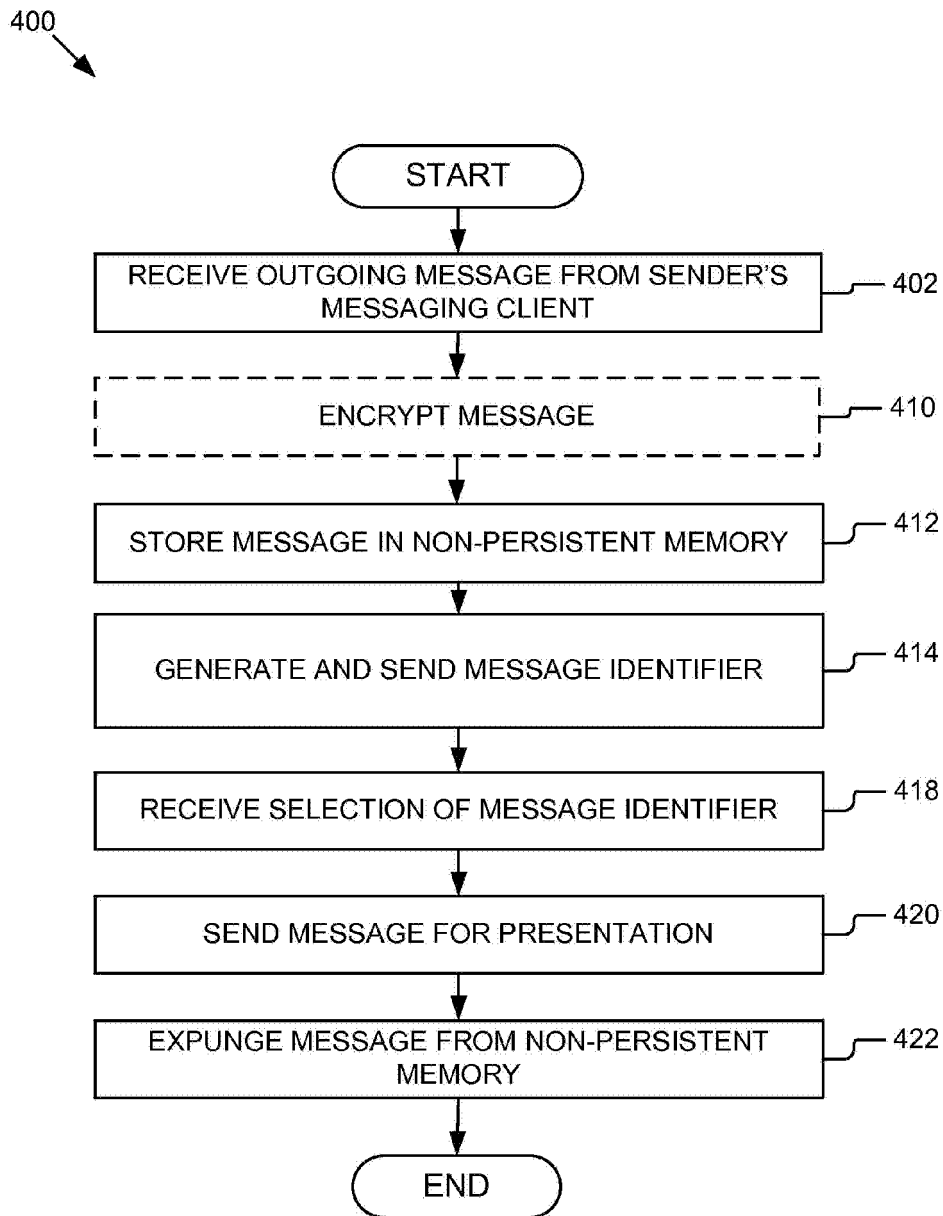


Figure 4

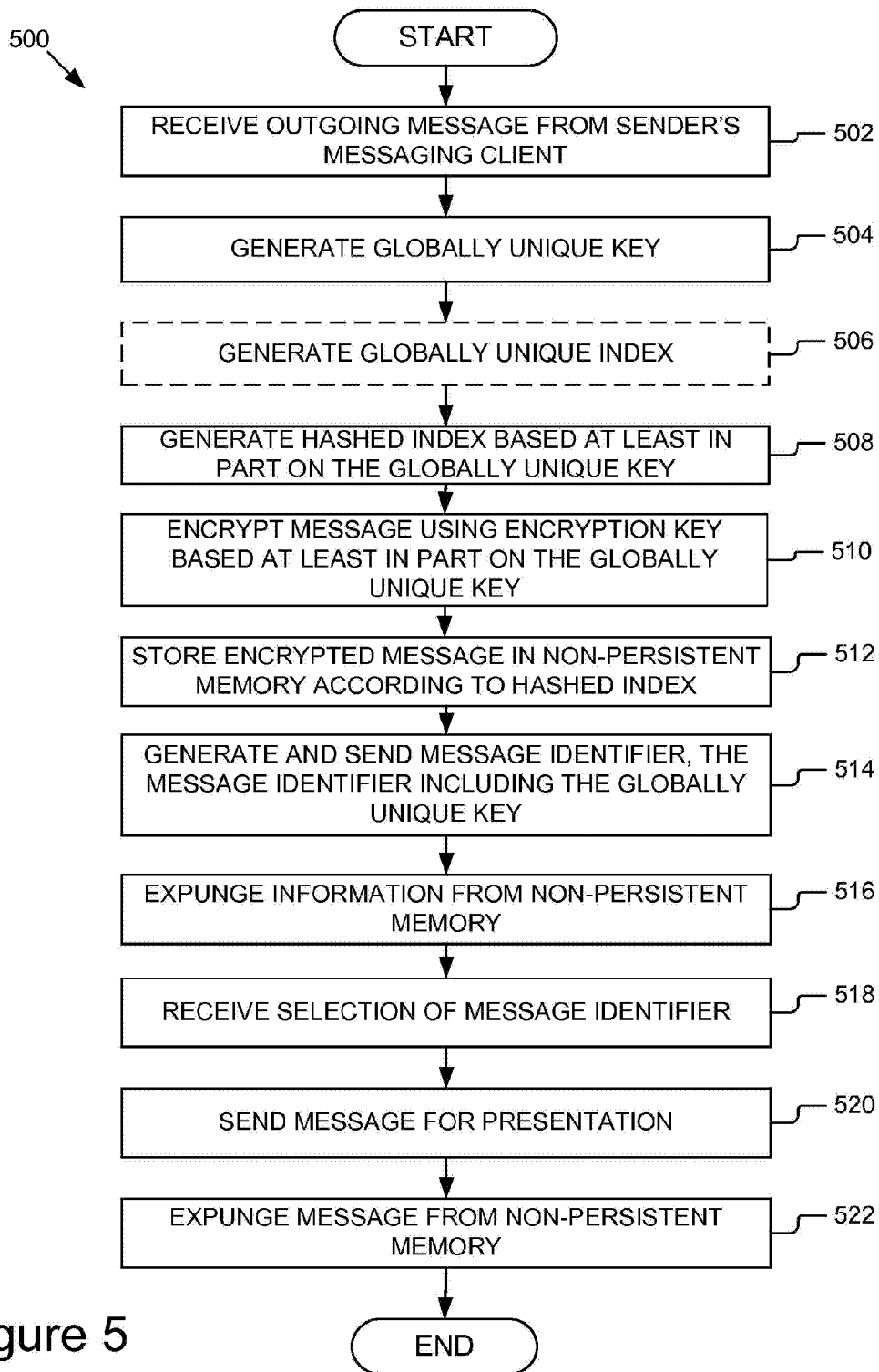
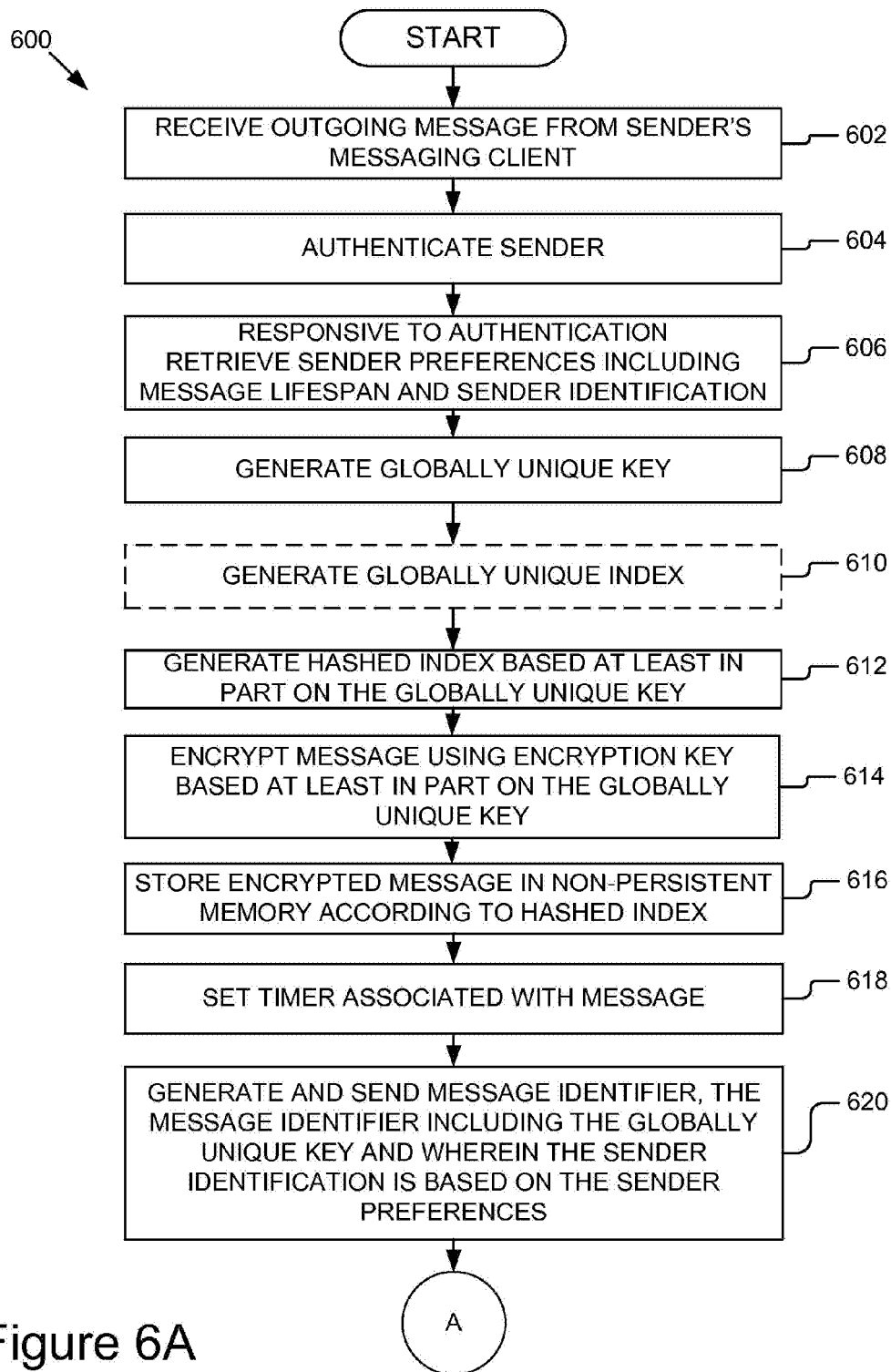


Figure 5



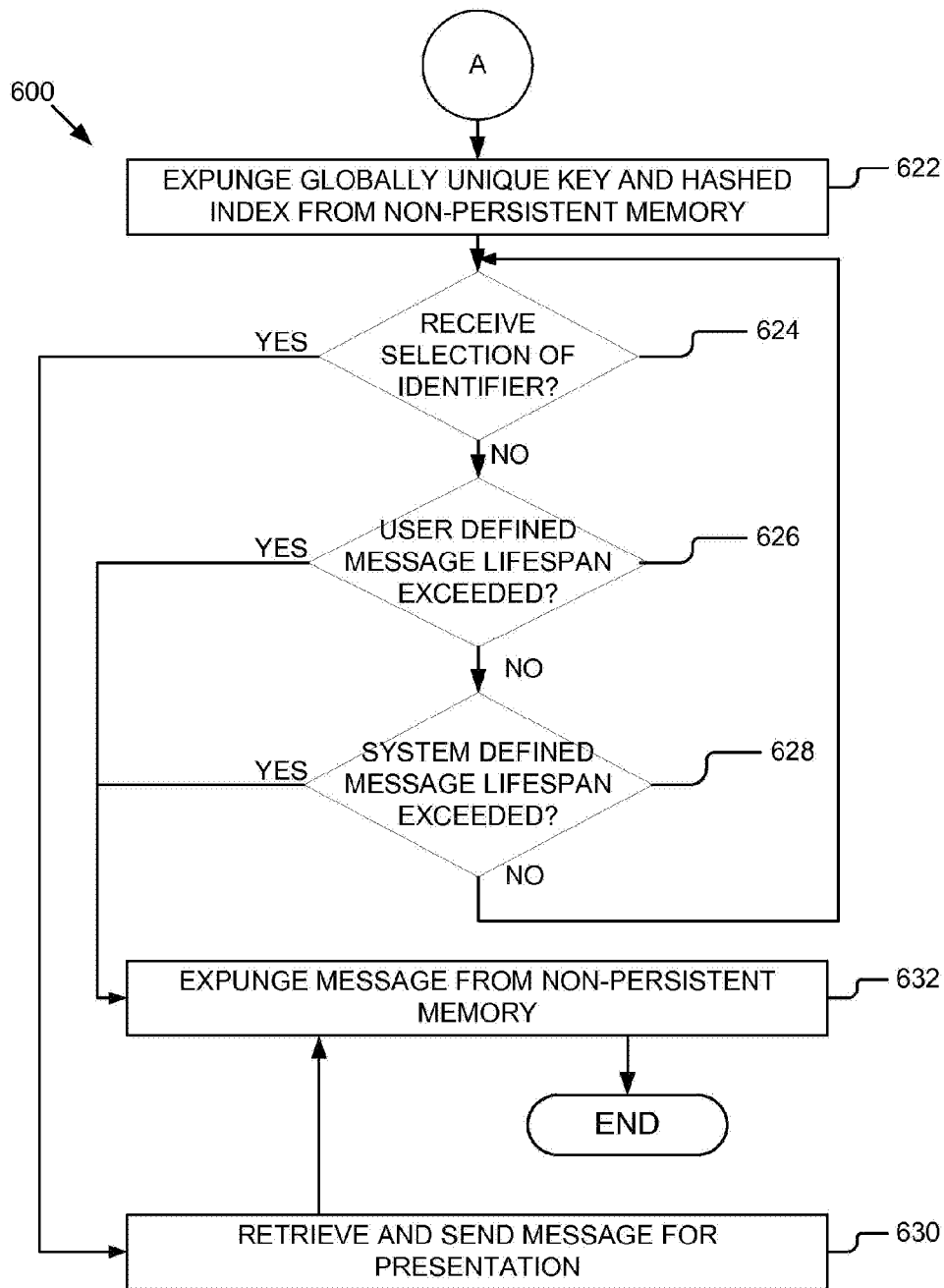


Figure 6B