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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 October 2008 (30.10.2008)

PCT

(10) International Publication Number
WO 2008/130440 A1

(51) International Patent Classification:
H04L 9/00 (2006.01)

(21) International Application Number:
PCT/US2007/082557

(22) International Filing Date: 25 October 2007 (25.10.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/862,965 25 October 2006 (25.10.2006) US
11/923,572 24 October 2007 (24.10.2007) US

(71) Applicant *(for all designated States except US)*: **IOVATION, INC.** [US/US]; 111 SW 5th Avenue, Suite 3200, Portland, OR 97204 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **LUNDE, Ron** [US/US]; 111 SW Fifth Avenue, Suite 3200, Portland, OR 97204 (US).

(74) Agents: **LEMOND, Kevin, T.** et al.; Schwabe, Williamson & Wyatt, Pacwest Center, Suite 1900, 1211 SW Fifth Avenue, Portland, OR 97204 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(54) **Title:** CREATING AND VERIFYING GLOBALLY UNIQUE DEVICE-SPECIFIC IDENTIFIERS

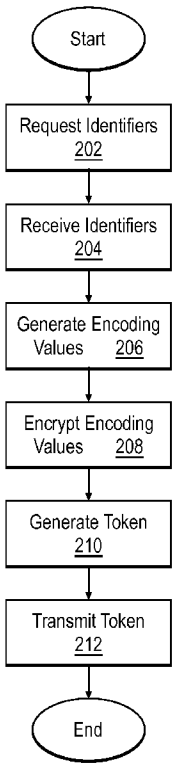

FIG. 2

(57) **Abstract:** Methods, apparatuses, and articles for receiving, by a server, a plurality of identifiers associated with a client device are described herein. The server may also encrypt a plurality of encoding values associated with the plurality of identifiers using a first key of a key pair of the server, and generate a token uniquely identifying the client device, a body of the token including the encrypted plurality of encoding values. In other embodiments, the server may receive a token along with the plurality of identifiers. In such embodiments, the server may further verify the validity of the received token, including attempting to decrypt a body of the received token with a key associated with a second server, the second server having generated the received token, and, if decryption succeeds, comparing ones of the plurality of identifiers with second identifiers found in the decrypted body to check for inconsistencies.

WO 2008/130440 A1

CREATING AND VERIFYING GLOBALLY UNIQUE DEVICE-SPECIFIC IDENTIFIERS

RELATED APPLICATIONS

The present application claims priority to U.S. Nonprovisional Application No. 11/923,572, entitled "Creating and Using Globally Unique Device-Specific Identifiers," filed on October 24, 2007, and U.S. Provisional Patent Application No. 60/862,965, entitled "Creating and Using Globally Unique Device-Specific Identifiers," filed on October 25, 2006, the entire specifications of which are hereby incorporated by reference in their entirety for all purposes, except for all those sections, if any, that are inconsistent with this specification.

FIELD OF THE INVENTION

The present invention relates to the field of data processing. More specifically, the present invention relates to the creation and verification of globally unique device-specific identifiers.

BACKGROUND OF THE INVENTION

Advances in microprocessor technologies have made computing ubiquitous. Advances in networking and telecommunication technologies have also made computing increasingly networked. Today, huge volumes of content and services are available through interconnected public and/or private networks. Ironically, the ubiquitous availability of computing has also led to abuses, such as denial of service attacks, viruses, spam, and phishing. For various on-line applications, it is increasingly desirable to uniquely identify a computing device (hereinafter, simply device).

Prior art methods of identifying devices have included the usage of device serial numbers, media access control (MAC) addresses and so forth.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be described by way of exemplary embodiments, but not limitations, illustrated in the accompanying drawings in which like

references denote similar elements, and in which:

Figure 1 illustrates an overview of various embodiments of the present invention;

Figure 2 illustrates a flowchart view of selected token creating operations, in accordance with various embodiments;

Figure 3 illustrates a flowchart view of selected token verifying operations, in accordance with various embodiments; and

Figure 4 is a block diagram illustrating an example computer system suitable for use to practice the present invention, in accordance with various embodiments.

DETAILED DESCRIPTION OF EMBODIMENTS OF THE INVENTION

Illustrative embodiments of the present invention include, but are not limited to, methods and apparatuses for receiving, by a server, a plurality of identifiers associated with a client device. The server may also encrypt a plurality of encoding values associated with the plurality of identifiers using a first key of a key pair of the server, and may generate a token uniquely identifying the client device, a body of the token including the encrypted plurality of encoding values. In other embodiments, the server may receive a token along with the plurality of identifiers. In such embodiments, the server may further verify the validity of the received token, including attempting to decrypt a body of the received token with a key associated with a second server, such as its public key, the second server having generated the received token, and, if decryption succeeds, comparing ones of the plurality of identifiers with second identifiers found in the decrypted body to check for inconsistencies.

Various aspects of the illustrative embodiments will be described using terms commonly employed by those skilled in the art to convey the substance of their work to others skilled in the art. However, it will be apparent to those skilled in the art that alternate embodiments may be practiced with only some of the described aspects. For purposes of explanation, specific numbers, materials, and configurations are set forth in order to provide a thorough understanding of the illustrative embodiments. However, it will be apparent to one skilled in the art that

alternate embodiments may be practiced without the specific details. In other instances, well-known features are omitted or simplified in order not to obscure the illustrative embodiments.

Further, various operations will be described as multiple discrete operations, in turn, in a manner that is most helpful in understanding the illustrative embodiments; however, the order of description should not be construed as to imply that these operations are necessarily order dependent. In particular, these operations need not be performed in the order of presentation.

The phrase "in one embodiment" is used repeatedly. The phrase generally does not refer to the same embodiment; however, it may. The terms "comprising," "having," and "including" are synonymous, unless the context dictates otherwise. The phrase "A/B" means "A or B". The phrase "A and/or B" means "(A), (B), or (A and B)". The phrase "at least one of A, B and C" means "(A), (B), (C), (A and B), (A and C), (B and C) or (A, B and C)". The phrase "(A) B" means "(B) or (A B)", that is, A is optional.

Figure 1 illustrates an overview of various embodiments of the present invention. As illustrated, one or more client devices 102 may receive tokens serving as globally unique device-specific identifiers of the client devices 102 from a server 108. Client devices 102 may be communicatively coupled to one or both of web server 106 and server 108 through networking fabric 104. In one embodiment, web server 106 and server 108 may actually be the same server device. In other embodiments, web server 106 may mediate communication between client devices 102 and server 108. Server 108 may in turn include token creating logic 110 and token verifying logic 112.

Server 108 may request of a client device 102 a plurality of non-unique identifiers and a token, if the client device already has a token. If the client device 102 does not have a token, it may provide only the identifiers, and server 108 may invoke creating logic 110. Creating logic 110 may receive the identifiers and encrypt a plurality of encoding values associated with the identifiers with a private key of server 108. Creating logic 110 may then generate a token having the encrypted values as the token body, and may transmit the token to the client device 102.

If, on the other hand, the client device 102 does have a token, it may provide both the token and the identifiers, and server 108 may invoke verifying logic 112. Verifying logic 112 may receive the token and identifiers and may verify the validity of the token. In some embodiments, verifying logic 112 may first attempt to decrypt the body of the token using the public key of the server that generated the token. If decryption succeeded, verifying logic 112 may then compare the received identifiers to identifiers comprising the body of the decrypted token to check for inconsistencies. In one embodiment, verifying logic 112 may reissue the token, if verification fails, based on one or more factors. In other embodiments, verifying logic 112 may periodically reissue the token regardless of the success or failure of verification.

As illustrated, client device 102, web server 106, and/or server 108 may each be one or more of any sort of computing device known in the art, except for creating logic 110, verifying logic 112, and other logic adapted to perform the operations described above and below. Client device 102, web server 106, and/or server 108 may each be a personal computer (PC), a workstation, a server, a router, a mainframe, a modular computer within a blade server or high-density server, a personal digital assistant (PDA), an entertainment center, a set-top box or a mobile device. Further, client device 102, web server 106, and/or server 108 may each be any single- or multi-processor or processor core central processing unit (CPU) computing system known in the art, except for creating logic 110, verifying logic 112, and other logic adapted to perform the operations described above and below. An exemplary single-/multi-processor or processor core client device 102, web server 106, or server 108 is illustrated by Figure 4, and will be described in greater detail herein.

In various embodiments, as previously mentioned, client devices 102 may be any end-user or other computing devices in communication with one or both of web server 106 and/or server 108. In one exemplary embodiment, a client device 102 may be a client in an ecommerce transaction, and web server 106 may require that the client device 102 provide a globally unique identifier as prior to completion of the transaction and/or as part of a log-in/authentication process.

Client devices 102 may possess client logic having been provided to client

devices 102 to enable client devices 102 to recognize and use tokens serving as the client devices' 102 globally unique identifiers. In one embodiment, a client device 102 may retrieve such client logic from web server 106, the web server 106 having received the client logic from server 108. In other embodiments, server 108 and web server 106 may be the same device, and client devices 102 may receive the client logic directly from server 108. The client logic may enable client devices 102 to perform a series of functions, including responding to requests from server 108 for tokens and non-unique identifiers. The logic may be able to determine whether or not a client device possesses a token, and if so, may be able to provide the token in response to the request. The logic may also enable a client device 102 to determine a plurality of non-unique identifiers and to provide those to the requesting server 108, with or without the token. Such non-unique identifiers may include at least one of a device serial number, a MAC address, an operating system (OS) type, and OS version, a time code, a country code, or a region code. In some embodiments, the client logic may also enable a client device 102 to receive a token from server 108 and to store that token to facilitate the client device 102 in responding to future requests from the token and identifiers.

As illustrated, client device 102, web server 106, and/or server 108 may each be communicatively connected to one or all of each other. In some embodiments, client device 102, web server 106, and/or server 108 may be connected by a networking fabric 104. Networking fabric 104 may include one or more of a LAN, a WAN, and the Internet. Networking fabric 104 may also be partially wired or wireless. In one embodiment, networking fabric 104 may be a private network connecting client devices 102 and server 108, server 108 also serving as web server 106, combining the services of those devices into one device. Communications across networking fabric 104 may be facilitated by any communication protocol known in the art, such as the Hypertext Transfer Protocol (HTTP) or the file transfer protocol (FTP), and any transport protocol known in the art, such as the Transmission Control Protocol/Internet Protocol (TCP/IP) suite of protocols. In some embodiments, client device 102, web server 106, and/or server 108 may be connected by one or more routers of the networking fabric (not

illustrated).

In various embodiments, as previously described, web server 106 may serve as an intermediary between client devices 102 and server 108. In such embodiments, some or all of the communications between client devices 102 and server 108, described more fully herein, may actually be routed through web server 106. As previously mentioned, in one exemplary embodiment, web server 106 may be a web server in an ecommerce transaction and may require end-user client devices 102 to provide a token serving as a globally-unique identifier. In such an embodiment, web server 106 may communicate with server 108 to request creation and/or verification of tokens for client devices 102. Web server 106 may also receive client logic from server 108 and may provide the client logic to new client devices 102 to enable the new client devices 102 to receive and provide tokens. In one embodiment, web server 106 may request a token from a client device 102 as part of a log-in/authentication process. In other embodiments, web server 106 may request the token at some different stage in the ecommerce transaction or as part of some other sort of transaction.

In alternate embodiments, web server 106 and server 108 may actually be the same computing device or may be two different devices of a common computing environment. In such an embodiment, the combined web server 106/server 108 may perform some or all of the operations of both devices. Such alternate embodiments may be utilized, for example, within the context of a private network where communication between a web server 106 of the private network and a server 108 not belonging to the private network might be considered undesirable.

As illustrated, and as previously mentioned, server 108 may be communicatively coupled to one or both of client devices 102 and server 106 through networking fabric 104. And as further described, server 108 may possess token creating logic 110 and token verifying logic 112 to enable server 108 to create and verify tokens that serve as globally-unique device identifiers. In various embodiments, server 108 may provide the above-mentioned client logic to web server 106 for dispersal to client devices 102, or may provide the client logic to client devices 102 directly via networking fabric 104. The client logic may have

been generated by server 108, or may have been generated by another device and provided to server 108. In some embodiments, server 108 may receive a request from web server 106, the request asking server 108 to verify or create a token for a client device 102. In response, server 108 may request the client device 102 to provide its token if it has one, as well as a plurality of non-unique identifiers associated with the client device 102. In other embodiments, server 108 may request the token and identifiers automatically, on some pre-determined basis, without first receiving a request from the web server 106. In one embodiment, server 108 may specify which non-unique identifiers the client device 102 should provide. In other embodiments, server 108 allows the client device 102 to determine which non-unique identifiers to provide. Upon receiving the client device 102's response, server 108 may determine whether the client device 102 included a token in the response. If the client device 102 did not provide a token, server 108 may invoke token creating logic 110. If client device 102 did provide a token, server 108 may invoke token verifying logic 112.

In various embodiments, creating logic 110 may receive the non-unique identifiers of the client device 102. In one embodiment, the identifiers may be received as parameters to the invocation of creating logic 110. Upon receiving the identifiers, creating logic 110 may generate a plurality of encoding values associated with the identifiers. Creating logic 110, in some embodiments, may generate the encoding values by performing a hash function on the identifiers (such as, for example, an MD5 hash of the identifiers). Creating logic may then assemble all or a portion of each encoding value (such as, for example, the low-order byte of an MD5 hash of an identifier having a length of one byte) into a token body. In various embodiments, creating logic may then encrypt the token body with a first key of a key pair, such as, for example, a private key of a public-private key pair, of server 108. The resulting encrypted token body may be, for example, a base64 string.

In some embodiments, creating logic 112 may then generate a token including the encrypted body, the token uniquely identifying the client device 102. Creating token may also include within the token a field to hold a server identifier. The server identifier may be useful if there are multiple servers 108 to identify

which of the servers 108 generated the token. Thus, as described in greater detail below, verifying logic 112 of a different server 108 may identify which key to decrypt a token body with. Each token may also include a message format version field and fields for separating characters, in some embodiments. The entire generated token may, in one embodiment, comprise an ANSI string.

In some embodiments, upon generating the token, creating logic 112 may transmit the token to the client device 102, either through web server 106 or directly.

In various embodiments, verifying logic 112 may receive the token and non-unique identifiers of the client device 102. In one embodiment, the token and identifiers may be received as parameters to the invocation of verifying logic 112. Upon receiving the token and identifiers, verifying logic 112 may verify the validity of the token, the verifying including decrypting the token's body and, if decryption succeeds, comparing the received identifiers to identifiers found in the decrypted body to check for inconsistencies. In some embodiments, verifying logic 112 may first attempt to decrypt the body of the token. Verifying logic 112 may read the server identifier of the token to determine which server 108 generated the token, and may decrypt the token body with the public key of that server 108. In some embodiments, each server 108 stores the public keys of every other server 108 in connection with every other server 108's server identifier. In other embodiments, verifying logic 112 may request the generating server 108's public key from that server 108 or from a common storage.

If decryption succeeded, verifying logic 112 may then compare the received identifiers to identifiers found in the decrypted body to check for inconsistencies. As previously mentioned, a token body may comprise a plurality of encoding values associated with a plurality of non-unique identifiers. These non-unique identifiers of the token ought to be, in some embodiments, identical to the received plurality of non-unique identifiers, as both are non-unique identifiers of the same device. In some embodiments, such as when a client device 102 has a new OS installed or a new hard drive, some of the non-unique identifiers for that client device 102 may change. Thus, in such a case, the non-unique identifiers whose encoding values form the token body may differ from the non-unique

identifiers provided. Also, if one device steals another device's token, there may likely be a number of inconsistencies between the identifiers. In one embodiment, verifying logic 112 may first decode the encoding values to retrieve the identifiers, and may then perform a comparison of the decoded identifiers with the received identifiers to determine if there are any differences.

In addition to decrypting and comparing, verifying logic 112 may also check other fields of the token, such as a "re-issue" flag field indicating that the token is a reissued token, a counter of the number of times the server 108 has seen that token, as well as a list of tokens and identifiers associated with evidence of fraud. Based on some of all of the above operations, verification logic 112 may ascertain the token's validity. The criteria used in making such judgments may vary from embodiment to embodiment. For example, in one embodiment, verifying logic 112 may consider a token to be valid as long as its body decrypts. In another embodiment, verifying logic 112 may require that decryption succeed, that identifiers be identical, and that none of the token and identifiers are present on the list of tokens and identifiers associated with evidence of fraud.

If verifying logic found the token to be valid, verifying logic 112 may inform the web server 106 and/or client devices 102 that the token is valid. If, on the other hand, verifying logic 112 found the token to be invalid, verifying logic may determine whether to reissue the token, the determining being based on one or more factors. In some embodiments, such factors may include success or failure of decryption, inconsistencies between some of the identifiers, presence of a token and/or device identifier of the client device 102 on the list of tokens and identifiers associated with evidence of fraud, a count of a number of times the token has been received by the server 108, some other association of the client device 102 to evidence of fraud, or a level of risk associated with the client device 102 (for example, a token with its reissue flag set may be deemed to be associated with a higher level of risk, in some embodiments). The number and weight of the factors may vary from embodiment to embodiment. Generally, if verifying logic 112 determines that fraud is less likely, based on the aforementioned factors, verifying logic may invoke creating logic 110 to reissue the token, passing creating logic 110 the received non-unique identifiers, as well

as indicating to creating logic 110 that a reissue flag should be present in the token body and should be set. If, on the other hand, verifying logic 112 determines that a token should not be reissued, verifying logic may simply inform the web server 106 and/or client device 102 that the token was invalid.

In various embodiments, regardless of whether the token is valid, the server 108 may periodically reissue the token.

Figure 2 illustrates a flowchart view of selected token creating operations, in accordance with various embodiments. As illustrated, a server may request of a client device a plurality of identifiers associated with the client device, block 202. In various embodiments, the plurality of identifiers may include at least one of a device serial number, a MAC address, an operating system (OS) type, and OS version, a time code, a country code, or a region code. The server may then receive the plurality of identifiers, block 204. In one embodiment, the identifiers may be received from a web server of a subscriber to services of the server.

In various embodiments, the server may then generate a plurality of encoding values associated with the plurality of identifiers by performing a hash function on the plurality of identifiers, block 206. Next, the server may encrypt the plurality of encoding values using a first key of a key pair of the server, block 208. In one embodiment, the first key of the key pair may be a private key of the server. In some embodiments, the server may then generate a token uniquely identifying the client device, a body of the token including the encrypted plurality of encoding values, block 210. In one embodiment, the token may further include a server identifier to identify the server as a generator of the token.

In some embodiments, the server may then transmit the token to the client device, block 212. In one embodiment, rather then transmitting directly to the client device, the server may transmit the token to a web server to facilitate the web server in providing the token to the client device. In another embodiment, the server may also provide client logic to the client device, directly or indirectly, to enable the client device to recognize and use the token.

Figure 3 illustrates a flowchart view of selected token verifying operations, in accordance with various embodiments. As illustrated, a first server may receive a token associated with a client device, the token acting as a unique identifier of

the client device, and a plurality of first identifiers associated with the client device, block 302. In one embodiment, the token and first identifiers may be received from a web server of a subscriber to services of the first server.

In various embodiments, the first server may then verify the validity of the received token, block 304. In some embodiments, the verifying may include attempting to decrypt a body of the token with a key associated with a second server, block 304a, the second server having generated the token. In one embodiment, the key associated with the second server may be the public key of the second server. Also, the verifying may include, if decryption succeeds, comparing ones of the plurality of first identifiers with second identifiers found in the decrypted body to check for inconsistencies, block 304b. In some embodiments, the second identifiers are identical to the first identifiers. Further, the first or second plurality of identifiers may include at least one of a device serial number, a media access control (MAC) address, an operating system (OS) type, and OS version, a time code, a country code, or a region code.

Next, the first server may, if token verification fails, determine whether to reissue the token based on one or more factors, block 306. In one embodiment, the one or more factors may include at least one of success or failure of decryption, inconsistencies between ones of the first and second identifiers, presence of a device identifier of the client device on a list of suspect devices, a count of a number of times the token has been received, association of the client device to evidence of fraud, or a level of risk associated with the client device. If the determination indicates that the token should be reissued, the first server may reissue the token, block 308.

In various embodiments, the first server may periodically reissue the token, block 308, regardless of whether the verifying indicates that the token is valid.

Figure 4 is a block diagram illustrating an example computer system suitable for use to practice the present invention, in accordance with various embodiments. As shown, computing system 400 includes one or more processors or processor cores 402, and system memory 404. For the purpose of this application, including the claims, the terms “processor” and “processor cores” may be considered synonymous, unless the context clearly requires otherwise.

Additionally, computing system 400 includes mass storage devices 406 (such as diskette, hard drive, compact disc read only memory (CDROM) and so forth), input/output devices 408 (such as keyboard, cursor control and so forth) and communication interfaces 410 (such as network interface cards, modems and so forth). The elements are coupled to each other via system bus 412, which represents one or more buses. In the case of multiple buses, they are bridged by one or more bus bridges (not illustrated).

Each of these elements performs its conventional functions known in the art. In particular, system memory 404 and mass storage 406 may be employed to store a working copy and a permanent copy of the programming instructions implementing all or a portion of earlier described server functions, herein collectively denoted as 422. The instructions 422 may be assembler instructions supported by processor(s) 402 or instructions that can be compiled from high level languages, such as C.

The permanent copy of the programming instructions may be placed into permanent storage 406 in the factory, or in the field, through, for example, a distribution medium (not shown), such as a compact disc (CD), or through communication interface 410 (from a distribution server (not shown)). That is, one or more distribution media having instructions 422 may be employed to distribute the instructions 422 and program various computing devices.

The constitution of these elements 402-412 are known, and accordingly will not be further described.

Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that a wide variety of alternate and/or equivalent implementations may be substituted for the specific embodiments shown and described, without departing from the scope of the present invention. Those skilled in the art will readily appreciate that the present invention may be implemented in a very wide variety of embodiments or extended therefrom. This application is intended to cover any adaptations or variations of the embodiments discussed herein. Therefore, it is manifestly intended that this invention be limited only by the claims and the equivalents thereof.

What is claimed is:

1. A method comprising:

receiving, by a server, a plurality of identifiers associated with a client device;

encrypting, by the server, a plurality of encoding values associated with the plurality of identifiers using a first key of a key pair of the server; and

generating, by the server, a token uniquely identifying the client device, a body of the token including the encrypted plurality of encoding values.

2. The method of claim 1, wherein the identifiers are received from a web server of a subscriber to services of the server, and the method further comprises transmitting, by the server, the token to the web server to facilitate the web server in providing the token to the client device.

3. The method of claim 1, further comprising transmitting, by the server, the token to the client device.

4. The method of claim 1, wherein the plurality of identifiers include at least one of a device serial number, a media access control (MAC) address, an operating system (OS) type, and OS version, a time code, a country code, or a region code.

5. The method of claim 1, further comprising generating, by the server, the encoding values by performing a hash function on the plurality of identifiers.

6. The method of claim 1, wherein the token further includes a server identifier to identify the server as a generator of the token.

7. The method of claim 1, further comprising requesting of the client device, by the server, the plurality of identifiers.

8. The method of claim 1, wherein the first key of the key pair is a private key of the server.

9. A method comprising:

receiving, by a first server, a token associated with a client device, the token acting as a unique identifier of the client device, and a plurality of first identifiers; and

verifying, by the first server, validity of the token, including

attempting to decrypt a body of the token with a key associated with a second server, the second server having generated the token, and

if decryption succeeds, comparing ones of the plurality of first identifiers with second identifiers found in the decrypted body to check for inconsistencies.

10. The method of claim 9, wherein the token and first identifiers are received from a web server of a subscriber to services of the first server.

11. The method of claim 9, wherein the first or second plurality of identifiers include at least one of a device serial number, a media access control (MAC) address, an operating system (OS) type, and OS version, a time code, a country code, or a region code.

12. The method of claim 9, wherein the key associated with the second server is the public key of the second server.

13. The method of claim 9, wherein the second identifiers are identical to the first identifiers.

14. The method of claim 9, further comprising, if token verification fails, determining, by the first server, whether to reissue the token based on one or more factors.

15. The method of claim 15, wherein the one or more factors include at least one of success or failure of decryption, inconsistencies between ones of the first and second identifiers, presence of a device identifier of the client device on a list of suspect devices, a count of a number of times the token has been received, association of the client device to evidence of fraud, or a level of risk associated with the client device.

16. The method of claim 9, further comprising periodically reissuing the token, by the first server, regardless of whether the verifying indicates that the token is valid.

17. A first server comprising:
a processor; and
logic to be operated by the processor to
receive a token associated with a client device, the token acting as a unique identifier of the client device, and a plurality of first identifiers, and
verify validity of the token, including
attempting to decrypt a body of the token with a key associated with a second server, the second server having generated the token, and
if decryption succeeds, comparing ones of the plurality of first identifiers with second identifiers found in the decrypted body to check for inconsistencies.

18. The first server of claim 17, wherein the logic is further to, if token verification fails, determine whether to reissue the token based on one or more rules or policies.

19. The first server of claim 18, wherein the one or more rules or policies include at least one of success or failure of decryption, inconsistencies between ones of the first and second identifiers, presence of a device identifier of the client

device on a list of suspect devices, a count of a number of times the token has been received, association of the client device to evidence of fraud, or a level of risk associated with the client device.

20. An article of manufacture comprising:

a storage medium; and

a plurality of programming instructions stored on the storage medium and configured to program a server to

receive a plurality of identifiers associated with a client device,

encrypt a plurality of encoding values associated with the plurality of identifiers using a private key of the server, and

generate a token uniquely identifying the client device, a body of the token including the encrypted plurality of encoding values.

21. The article of claim 20, wherein the programming instructions are further configured to program the server to provide client logic to the client device, directly or indirectly, to enable the client device to recognize and use the token.

22. The article of claim 21, wherein the plurality of identifiers include at least one of a device serial number, a media access control (MAC) address, an operating system (OS) type, and OS version, a time code, a country code, or a region code.

23. The article of claim 21, wherein the token further includes a server identifier to identify the server as a generator of the token.

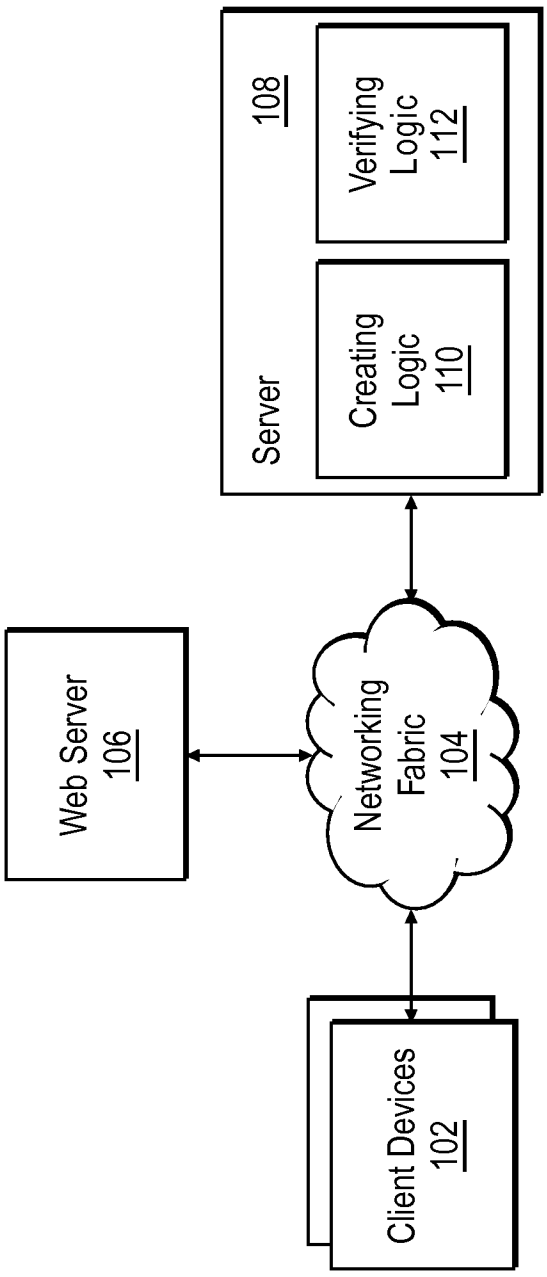
Server
108
Verifying Logic 112
Creating Logic 110
Networking Fabric 104
Web Server 106
Client Devices 102

FIG. 1

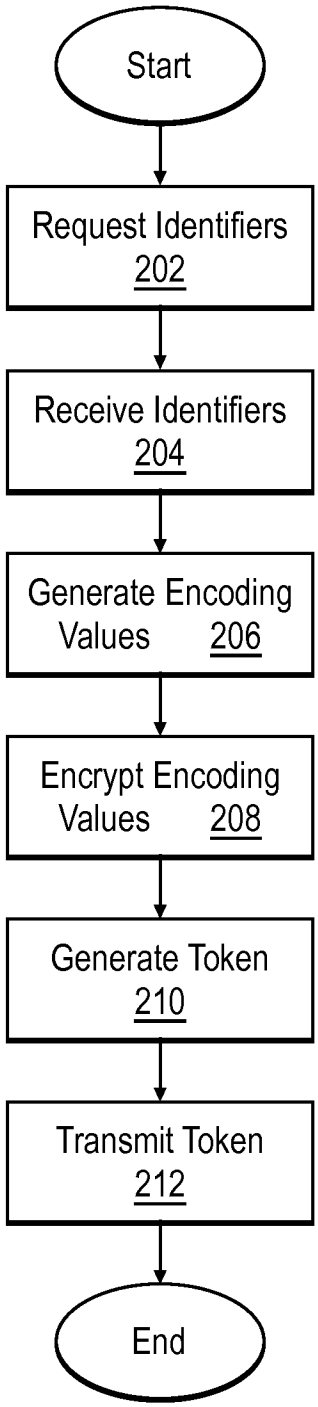
Start
Request Identifiers 202
Receive Identifiers 204
Generate Encoding Values 206
Encrypt Encoding Values 208
Generate Token 210
Transmit Token 212
End

FIG. 2

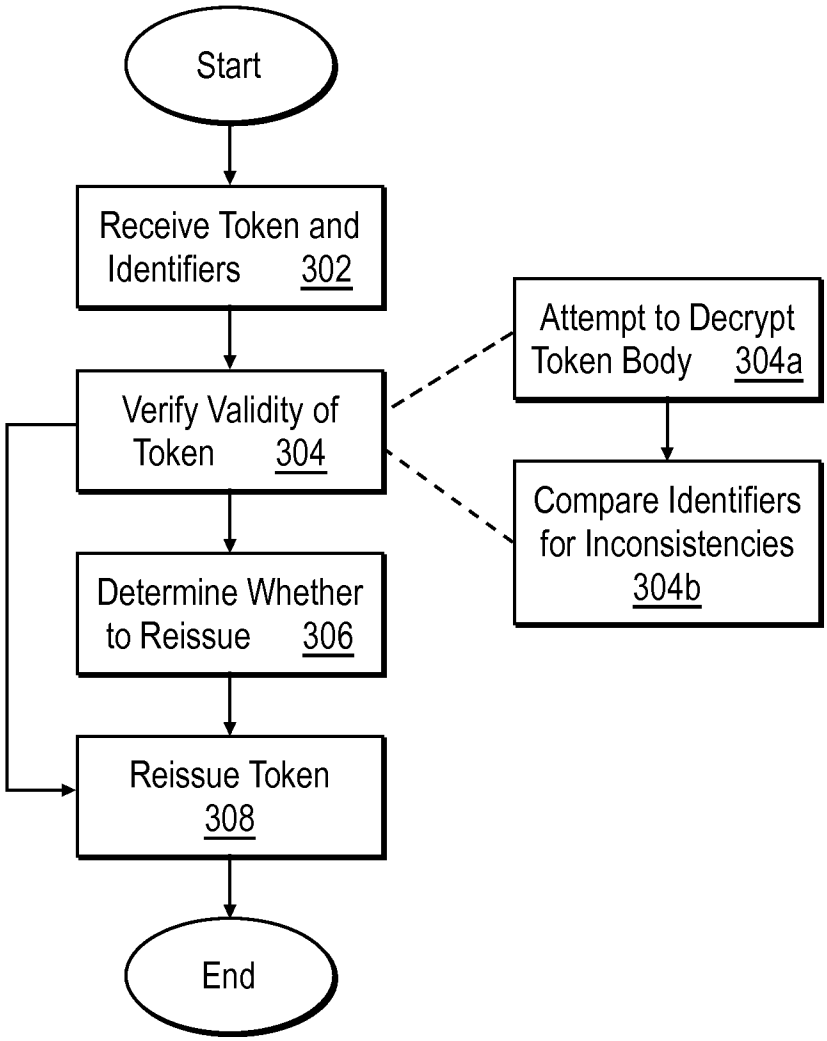
Start
Receive Token and Identifiers 302
Verify Validity of Token 304
Attempt to Decrypt Token Body 304a
Compare Identifiers for Inconsistencies 304b
Determine Whether to Reissue 306
Reissue Token 308
End

FIG. 3

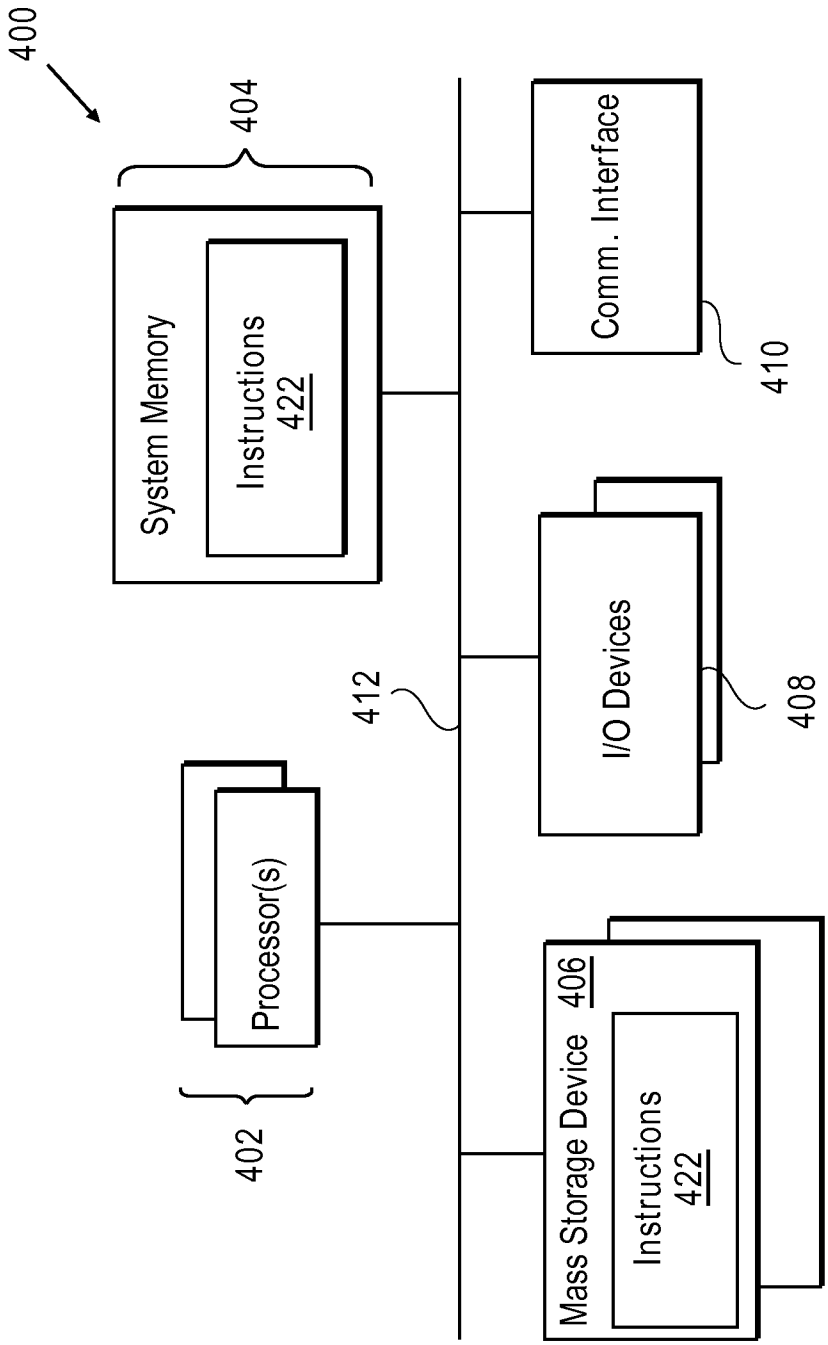
400
404
System Memory
Instructions
422
Comm. Interface
410
412
I/O Devices
408
Processor(s)
402
Mass Storage Device
406
Instructions
422

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 07/82557

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - H04L 9/00 (2008.04)
USPC - 705/64
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
USPC - 705/64
IPC(8) - H04L 9/00 (2008.04)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC - 705/50, 65; 700/1, 90

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
PubWEST (USPT, PGPB, EPAB, JPAB); DIALOG PRO (Engineering) & Google; server, identifier, client device, encrypting, encoding values, key, key pair, token, body of the token, verifying, unique identifier, validity of the token, decryption, decrypted body, inconsistencies, processor, logic, storage medium, programming...

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2006/0080536 A1 (Teppler) 13 August 2006 (13.08.2006), entire document, especially Fig. 13; Para [0004], [0015], [0021], [0030], [0032], [0039]-[0040], [0043]-[0044], [0048]-[0049], [0051]-[0053], [0056], [0070], [0074], [0079], [0107], [0114], [0120]-[0121], [0129], [0151], [0154], [0184], [0195]-[0198], [0205], [0208]-[0209], [0212]-[0213], [0215], [0221]-[0222], [0224], [0231], [0248], [0253], [0261], [0304], [0319], [0321], [0324]-[0325], [0340], [0342], [0361], [0383], [0390], [0406] and [0437].	1-23

☐ Further documents are listed in the continuation of Box C. ☐

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

Date of the actual completion of the international search	Date of mailing of the international search report
03 June 2008 (10.05.2008)	13 JUN 2008
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (April 2007)