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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 March 2016 (24.03.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/044373 A1

(51) International Patent Classification:
G06F 7/04 (2006.01) *H04L 29/06* (2006.01)

(21) International Application Number:
PCT/US2015/050348

(22) International Filing Date:
16 September 2015 (16.09.2015)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
14/487,992 16 September 2014 (16.09.2014) US

(71) Applicant: NOK NOK LABS, INC. [US/US]; 4151 Middlefield Road, Suite 200, Palo Alto, California 94303 (US).

(72) Inventors: WILSON, Brendon J.; 4151 Middlefield Road, Suite 200, Palo Alto, California 94303 (US). BAGHDASARYAN, Davit; 4151 Middlefield Road, Suite 200, Palo Alto, California 94303 (US).

(74) Agent: WEBSTER, Thomas; Nicholson De Vos Webster & Elliott, LLP, 217 High Street, Palo Alto, California 94301 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, 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, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, 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, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) Title: SYSTEM AND METHOD FOR INTEGRATING AN AUTHENTICATION SERVICE WITHIN A NETWORK ARCHITECTURE

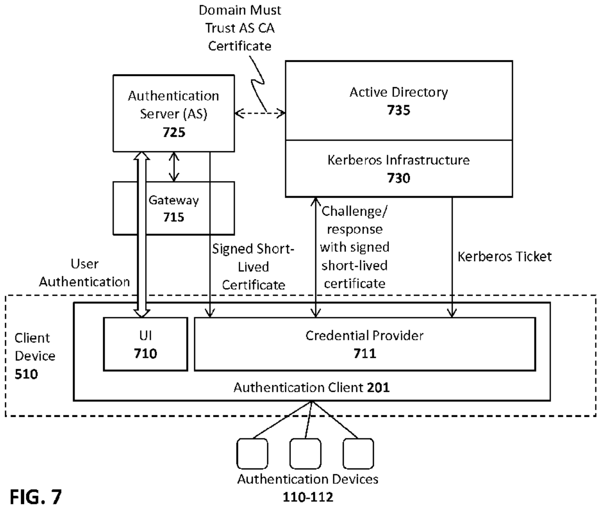
Domain Must Trust AS CA Certificate
Authentication Server (AS) 725
Active Directory 735
Kerberos Infrastructure 730
Gateway 715
Challenge/ response with signed short-lived certificate
Signed Short-Lived Certificate
User Authentication
Kerberos Ticket
Client Device 510
UI 710
Credential Provider 711
Authentication Client 201
Authentication Devices 110-112

FIG. 7

(57) Abstract: A system, apparatus, method, and machine readable medium are described for integrating an authentication service within an existing network infrastructure. One embodiment comprises: a gateway configured to restrict access to an internal network; an authentication server communicatively coupled to the gateway; a client device with an authentication client having a plurality of authentication devices coupled thereto for authenticating a user, the authentication client configured to establish a communication channel with the authentication server and to register one or more of the authentication devices with the authentication server, the authentication devices usable for performing online authentication with the authentication server following registration; the authentication client to authenticate the user with the authentication server using one or more of the registered authentication devices in response to an attempt to gain access to the internal network via the gateway.

WO 2016/044373 A1

SYSTEM AND METHOD FOR INTEGRATING AN AUTHENTICATION SERVICE WITHIN A NETWORK ARCHITECTURE

BACKGROUND

Field of the Invention

[0001] This invention relates generally to the field of data processing systems. More particularly, the invention relates to a system and method for integrating an authentication service within a network architecture.

Description of Related Art

[0002] Systems have also been designed for providing secure user authentication over a network using biometric sensors. In such systems, the a score generated by an authenticator, and/or other authentication data, may be sent over a network to authenticate the user with a remote server. For example, Patent Application No. 2011/0082801 ("'801 Application") describes a framework for user registration and authentication on a network which provides strong authentication (e.g., protection against identity theft and phishing), secure transactions (e.g., protection against "malware in the browser" and "man in the middle" attacks for transactions), and enrollment/management of client authentication tokens (e.g., fingerprint readers, facial recognition devices, smartcards, trusted platform modules, etc).

[0003] The assignee of the present application has developed a variety of improvements to the authentication framework described in the '801 application. Some of these improvements are described in the following set of US Patent Applications, which are assigned to the present assignee: Serial No. 13/730,761, Query System and Method to Determine Authentication Capabilities; Serial No. 13/730,776, System and Method for Efficiently Enrolling, Registering, and Authenticating With Multiple Authentication Devices; 13/730,780, System and Method for Processing Random Challenges Within an Authentication Framework; Serial No. 13/730,791, System and Method for Implementing Privacy Classes Within an Authentication Framework; Serial No. 13/730,795, System and Method for Implementing Transaction Signaling Within an Authentication Framework; and Serial No. 14/218,504, Advanced Authentication Techniques and Applications (hereinafter "'504 Application"). These applications are sometimes referred to herein as the ("Co-pending Applications").

[0004] Briefly, the Co-Pending applications describe authentication techniques in which a user enrolls with authentication devices (or Authenticators) such as biometric devices (e.g., fingerprint sensors) on a client device. When a user enrolls with a biometric device, biometric reference data is captured (e.g., by swiping a finger, snapping a picture, recording a voice, etc). The user may subsequently register/provision the authentication devices with one or more servers over a network (e.g., Websites or other relying parties equipped with secure transaction/authentication services as described in the Co-Pending Applications); and subsequently authenticate with those servers using data exchanged during the registration process (e.g., cryptographic keys provisioned into the authentication devices). Once authenticated, the user is permitted to perform one or more online transactions with a Website or other relying party. In the framework described in the Co-Pending Applications, sensitive information such as fingerprint data and other data which can be used to uniquely identify the user, may be retained locally on the user's authentication device to protect a user's privacy.

[0005] The '504 Application describes a variety of additional techniques including techniques for designing composite authenticators, intelligently generating authentication assurance levels, using non-intrusive user verification, transferring authentication data to new authentication devices, augmenting authentication data with client risk data, and adaptively applying authentication policies, and creating trust circles, to name just a few.

[0006] Augmenting a Relying Party's web-based or other network enabled application to leverage the remote authentication techniques described in the co-pending applications typically requires the application to integrate directly with an authentication server. This poses a barrier to the adoption of such authentication, as Relying Parties will need to expend effort to update their applications to integrate with an authentication server in order to gain the authentication flexibility provided by the techniques described in the co-pending applications.

[0007] In some cases, the Relying Party may have already integrated with federation solutions, and thus a simple integration path is to simply integrate online authentication support into the federation solution. Unfortunately, this approach does not address other legacy systems (such as VPNs, Windows Kerberos deployments) that either lack awareness of federation protocols (and thus could be front-ended by a federation server augmented with online authentication functionality), or lack sufficient extensibility to enable direct integration of online authentication functionality. Hence, a key problem that must be solved for certain Relying Party applications is finding a way to enable them to integrate online authentication systems, without requiring the code for the applications themselves to be modified.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] A better understanding of the present invention can be obtained from the following detailed description in conjunction with the following drawings, in which:

[0009] **FIGS. 1A-B** illustrate two different embodiments of a secure authentication system architecture;

[0010] **FIG. 2** is a transaction diagram showing how keys may be registered into authentication devices;

[0011] **FIG. 3** illustrates a transaction diagram showing remote authentication;

[0012] **FIG. 4** illustrates a system for connecting a user to an internal network through a secure sockets layer (SSL) virtual private network (VPN) gateway;

[0013] **FIG. 5** illustrates one embodiment of a system for integrating an authentication server within a network infrastructure;

[0014] **FIG. 6** illustrates one embodiment of a method for performing authentication using an authentication server integrated within a network infrastructure;

[0015] **FIG. 7** illustrates one embodiment of a system for integrating an authentication server within a Kerberos infrastructure;

[0016] **FIG. 8** illustrates one embodiment of a method for performing authentication using an authentication server integrated within a Kerberos infrastructure;

[0017] **FIG. 9** illustrates one embodiment of a computer architecture used for servers and/or clients; and

[0018] **FIG. 10** illustrates one embodiment of a computer architecture used for servers and/or clients.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0019] Described below are embodiments of an apparatus, method, and machine-readable medium for implementing advanced authentication techniques and associated applications. Throughout the description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without some of these specific details. In other instances, well-known structures and devices are not shown or are shown in a block diagram form to avoid obscuring the underlying principles of the present invention.

[0020] The embodiments of the invention discussed below involve authentication devices with user verification capabilities such as biometric modalities or PIN entry. These devices are sometimes referred to herein as “tokens,” “authentication devices,” or “authenticators.” While certain embodiments focus on facial recognition hardware/software (e.g., a camera and associated software for recognizing a user’s face and tracking a user’s eye movement), some embodiments may utilize additional biometric devices including, for example, fingerprint sensors, voice recognition hardware/software (e.g., a microphone and associated software for recognizing a user’s voice), and optical recognition capabilities (e.g., an optical scanner and associated software for scanning the retina of a user). The user verification capabilities may also include non-biometric modalities, like PIN entry. The authenticators might use devices like trusted platform modules (TPMs), smartcards and secure elements for cryptographic operations and key storage.

[0021] In a mobile biometric implementation, the biometric device may be remote from the relying party. As used herein, the term “remote” means that the biometric sensor is not part of the security boundary of the computer it is communicatively coupled to (e.g., it is not embedded into the same physical enclosure as the relying party computer). By way of example, the biometric device may be coupled to the relying party via a network (e.g., the Internet, a wireless network link, etc) or via a peripheral input such as a USB port. Under these conditions, there may be no way for the relying party to know if the device is one which is authorized by the relying party (e.g., one which provides an acceptable level of authentication strength and integrity protection) and/or whether a hacker has compromised or even replaced the biometric device. Confidence in the biometric device depends on the particular implementation of the device.

[0022] The term “local” is used herein to refer to the fact that the user is completing a transaction in person, at a particular location such as at an automatic teller machine (ATM) or a point of sale (POS) retail checkout location. However, as discussed below, the authentication techniques employed to authenticate the user may involve non-location components such as communication over a network with remote servers and/or other data processing devices. Moreover, while specific embodiments are described herein (such as an ATM and retail location) it should be noted that the underlying principles of the invention may be implemented within the context of any system in which a transaction is initiated locally by an end user.

[0023] The term “relying party” is sometimes used herein to refer, not merely to the entity with which a user transaction is attempted (e.g., a Website or online service performing user transactions), but also to the secure transaction servers (sometimes referred to as “au implemented on behalf of that entity which may performed the underlying authentication techniques described herein. The secure transaction servers may be owned and/or under the control of the relying party or may be under the control of a third party offering secure transaction services to the relying party as part of a business arrangement.

[0024] The term “server” is used herein to refer to software executed on a hardware platform (or across multiple hardware platforms) that receives requests over a network from a client, responsively performs one or more operations, and transmits a response to the client, typically including the results of the operations. The server responds to client requests to provide, or help to provide, a network “service” to the clients. Significantly, a server is not limited to a single computer (e.g., a single hardware device for executing the server software) and may, in fact, be spread across multiple hardware platforms, potentially at multiple geographical locations.

EXEMPLARY ONLINE AUTHENTICATION ARCHITECTURES AND TRANSACTIONS

[0025] **Figures 1A-B** illustrate two embodiments of a system architecture comprising client-side and server-side components for registering authentication devices (also sometimes referred to as “provisioning”) and authenticating a user. The embodiment shown in **Figure 1A** uses a web browser plugin-based architecture for communicating with a website while the embodiment shown in **Figure 1B** does not require a web browser. The various techniques described herein such as enrolling a user with authentication devices, registering the authentication devices with a secure server, and verifying a user may be implemented on either of these system architectures. Thus, while the architecture shown in **Figure 1A** is used to demonstrate the operation of several of the embodiments described below, the same basic principles may be easily implemented on the system shown in **Figure 1B** (e.g., by removing the browser plugin 105 as the intermediary for communication between the server 130 and the secure transaction service 101 on the client).

[0026] Turning first to **Figure 1A**, the illustrated embodiment includes a client 100 equipped with one or more authentication devices 110-112 (sometimes referred to in the art as authentication “tokens” or ”Authenticators”) for enrolling and verifying an end user. As mentioned above, the authentication devices 110-112 may include biometric device such as fingerprint sensors, voice recognition hardware/software (e.g., a microphone and associated software for recognizing a user's voice), facial recognition hardware/software (e.g., a camera and associated software for recognizing a user's face), and optical recognition capabilities (e.g., an optical scanner and associated software for scanning the retina of a user) and support for non-biometric modalities, such as PIN verification. The authentication devices might use trusted platform modules (TPMs), smartcards or secure elements for cryptographic operations and key storage.

[0027] The authentication devices 110-112 are communicatively coupled to the client through an interface 102 (e.g., an application programming interface or API) exposed by a secure transaction service 101. The secure transaction service 101 is a secure application for communicating with one or more secure transaction servers 132-133 over a network and for interfacing with a secure transaction plugin 105 executed within the context of a web browser 104. As illustrated, the Interface 102 may also provide secure access to a secure storage device 120 on the client 100 which stores information related to each of the authentication devices 110-112 such as a device identification code, user identification code, user enrollment data (e.g., scanned fingerprint or other biometric data) protected by he authentication device, and keys wrapped by the authentication device used to perform the secure authentication techniques described herein. For example, as discussed in detail below, a unique key may be stored into each of the authentication devices and used when communicating to servers 130 over a network such as the Internet.

[0028] As discussed below, certain types of network transactions are supported by the secure transaction plugin 105 such as HTTP or HTTPS transactions with websites 131 or other servers. In one embodiment, the secure transaction plugin is initiated in response to specific HTML tags inserted into the HTML code of a web page by the web server 131 within the secure enterprise or Web destination 130 (sometimes simply referred to below as "server 130"). In response to detecting such a tag, the secure transaction plugin 105 may forward transactions to the secure transaction service 101 for processing. In addition, for certain types of transactions (e.g., such as secure key exchange) the secure transaction service 101 may open a direct communication channel with the on-premises transaction server 132 (i.e., co-located with the website) or with an off-premises transaction server 133.

[0029] The secure transaction servers 132-133 are coupled to a secure transaction database 120 for storing user data, authentication device data, keys and other secure information needed to support the secure authentication transactions described below.

It should be noted, however, that the underlying principles of the invention do not require the separation of logical components within the secure enterprise or web destination 130 shown in **Figure 1A**. For example, the website 131 and the secure transaction servers 132-133 may be implemented within a single physical server or separate physical servers. Moreover, the website 131 and transaction servers 132-133 may be implemented within an integrated software module executed on one or more servers for performing the functions described below.

[0030] As mentioned above, the underlying principles of the invention are not limited to a browser-based architecture shown in **Figure 1A**. **Figure 1B** illustrates an alternate implementation in which a stand-alone application 154 utilizes the functionality provided by the secure transaction service 101 to authenticate a user over a network. In one embodiment, the application 154 is designed to establish communication sessions with one or more network services 151 which rely on the secure transaction servers 132-133 for performing the user/client authentication techniques described in detail below.

[0031] In either of the embodiments shown in **Figures 1A-B**, the secure transaction servers 132-133 may generate the keys which are then securely transmitted to the secure transaction service 101 and stored into the authentication devices within the secure storage 120. Additionally, the secure transaction servers 132-133 manage the secure transaction database 120 on the server side.

[0032] Certain basic principles associated with remotely registering authentication devices and authenticating with a relying party will be described with respect to **Figures 2-3**, followed by a detailed description of embodiments of the invention for establishing trust using secure communication protocols.

[0033] **Figure 2** illustrates a series of transactions for registering authentication devices on a client (such as devices 110-112 on client 100 in Figures 1A-B) (sometimes referred to as "provisioning" authentication devices). For simplicity, the secure transaction service 101 and interface 102 are combined together as authentication client 201 and the secure enterprise or web destination 130 including the secure transaction servers 132-133 are represented as a relying party 202.

[0034] During registration of an authenticator (e.g., a fingerprint authenticator, voice authenticator, etc), a key associated with the authenticator is shared between the authentication client 201 and the relying party 202. Referring back to **Figures 1A-B**, the key may be stored within the secure storage 120 of the client 100 and the secure transaction database 120 used by the secure transaction servers 132-133. In one embodiment, the key is a symmetric key generated by one of the secure transaction servers 132-133. However, in another embodiment discussed below, asymmetric keys are be used. In this embodiment, the public/private key pair may be generated by the secure transaction servers 132-133. The public key may then be stored by the secure transaction servers 132-133 and the related private key may be stored in the secure storage 120 on the client. In an alternate embodiment, the key(s) may be generated on the client 100 (e.g., by the authentication device or the authentication device interface rather than the secure transaction servers 132-133). The underlying principles of the invention are not limited to any particular types of keys or manner of generating the keys.

[0035] A secure key provisioning protocol is employed in one embodiment to share the key with the client over a secure communication channel. One example of a key provisioning protocol is the Dynamic Symmetric Key Provisioning Protocol (DSKPP) (see, e.g., Request for Comments (RFC) 6063). However, the underlying principles of the invention are not limited to any particular key provisioning protocol. In one particular embodiment, the client generates a public/private key pair and sends the public key to the server, which may be attested with an attestation key.

[0036] Turning to the specific details shown in **Figure 2**, to initiate the registration process, the relying party 202 generates a randomly generated challenge (e.g., a cryptographic nonce) that must be presented by the authentication client 201 during device registration. The random challenge may be valid for a limited period of time. In response, the authentication client 201 initiates an out-of-band secure connection with the relying party 202 (e.g., an out-of-band transaction) and communicates with the relying party 202 using the key provisioning protocol (e.g., the DSKPP protocol mentioned above). To initiate the secure connection, the authentication client 201 may provide the random challenge back to the relying party 202 (potentially with a signature generated over the random challenge). In addition, the authentication client 201 may transmit the identity of the user (e.g., a user ID or other code) and the identity of the authentication device(s) to be provisioned registered (e.g., using the authentication attestation ID (AAID) which uniquely identify the type of authentication device(s) being provisioned).

[0037] The relying party locates the user with the user name or ID code (e.g., in a user account database), validates the random challenge (e.g., using the signature or simply comparing the random challenge to the one that was sent), validates the authentication device's authentication code if one was sent (e.g., the AAID), and creates a new entry in a secure transaction database (e.g., database 120 in Figures 1A-B) for the user and the authentication device(s). In one embodiment, the relying party maintains a database of authentication devices which it accepts for authentication. It may query this database with the AAID (or other authentication device(s) code) to determine if the authentication device(s) being provisioned are acceptable for authentication. If so, then it will proceed with the registration process.

[0038] In one embodiment, the relying party 202 generates an authentication key for each authentication device being provisioned. It writes the key to the secure database and sends the key back to the authentication client 201 using the key provisioning protocol. Once complete, the authentication device and the relying party 202 share the same key if a symmetric key was used or different keys if asymmetric keys were used. For example, if asymmetric keys were used, then the relying party 202 may store the public key and provide the private key to the authentication client 201. Upon receipt of the private key from the relying party 202, the authentication client 201 provisions the key into the authentication device (storing it within secure storage associated with the authentication device). It may then use the key during authentication of the user (as described below). In an alternate embodiment, the key(s) are generated by the authentication client 201 and the key provisioning protocol is used to provide the key(s) to the relying party 202. In either case, once provisioning is complete, the authentication client 201 and relying party 202 each have a key and the authentication client 201 notifies the relying party of the completion.

[0039] **Figure 3** illustrates a series of transactions for user authentication with the provisioned authentication devices. Once device registration is complete (as described in Figure 2), the relying party 202 will accept an authentication response (sometimes referred to as a “token”) generated by the local authentication device on the client as a valid authentication response.

[0040] Turning to the specific details shown in **Figure 3**, in response to the user initiating a transaction with the relying party 202 which requires authentication (e.g., initiating payment from the relying party’s website, accessing private user account data, etc), the relying party 202 generates an authentication request which includes a random challenge (e.g., a cryptographic nonce). In one embodiment, the random challenge has a time limit associated with it (e.g., it is valid for a specified period of time). The relying party may also identify the authenticator to be used by the authentication client 201 for authentication. As mentioned above, the relying party may provision each authentication device available on the client and stores a public key for each provisioned authenticator. Thus, it may use the public key of an authenticator or may use an authenticator ID (e.g., AAID) to identify the authenticator to be used. Alternatively, it may provide the client with a list of authentication options from which the user may select.

[0041] In response to receipt of the authentication request, the user may be presented with a graphical user interface (GUI) requesting authentication (e.g., in the form of a web page or a GUI of an authentication application/app). The user then performs the authentication (e.g., swiping a finger on a fingerprint reader, etc). In response, the authentication client 201 generates an authentication response containing a signature over the random challenge with the private key associated with the authenticator. It may also include other relevant data such as the user ID code in the authentication response.

[0042] Upon receipt of the authentication response, the relying party may validate the signature over the random challenge (e.g., using the public key associated with the authenticator) and confirm the identity of the user. Once authentication is complete, the user is permitted to enter into secure transactions with the relying party, as illustrated.

[0043] A secure communication protocol such as Transport Layer Security (TLS) or Secure Sockets Layer (SSL) may be used to establish a secure connection between the relying party 201 and the authentication client 202 for any or all of the transactions illustrated in **Figures 2-3**.

System and Method for Integrating an Authentication Service With a Network Architecture

[0044] Many legacy systems may feature support for an authentication methods other than usernames and passwords. For example, secure sockets layer (SSL) virtual private network (VPN) systems support the use of One Time Passwords (OTPs). Systems such as Kerberos allow the user to authenticate to a network or service using a digital certificate.

[0045] The embodiments of the invention described herein leverage these features to integrate an online authentication service with such legacy systems without requiring any changes to the legacy system itself (other than configuration changes).

[0046] To augment the security of secure socket layer (SSL) virtual private networks (VPNs), enterprises deploy second factor authentication solutions based on OTP approaches. Solutions such as RSA SecurID or OATH require the user to carry an OTP generator and input the OTP generated by this generator in combination with the username and password to authenticate to VPN.

[0047] **Figure 4** illustrates an OTP validation server 425 configured to operate in combination with an SSL VPN gateway 415. In operation, the user opens a web browser 410 and navigates to the SSL VPN gateway 415 which renders an HTML-based login form 411 containing a user ID field 412 and password field 413. The user may enter a user ID in the UID field 412 and the OTP in the password field 413 (either by itself or appended to the user's static password). After entering the user name and password via the HTML form 411, the user submits the results to the SSL VPN gateway 415.

[0048] The SSL VPN gateway 415 validates the username and password against a user store 420 (e.g., verifying the user name exists and that the correct password was entered) and validates the OTP by providing the OTP entered by the user to the OTP validation server 425. If the OTP validation server 425 provides an affirmative response, validating the OTP, the SSL VPN gateway 415 grants the user access to the protected internal network 430.

[0049] As mentioned, in the above example, the SSL VPN gateway 415 may render a separate form element to enable input of the OTP while, in other cases, the SSL VPN gateway 415 may simply rely on the user appending their OTP to the password in the form's password field. In addition, the SSL VPN gateway 415 may immediately reject access if the primary username and password are not accepted by the user store 420 validation. Communication between the SSL VPN gateway 415 and the OTP validation server 425 may be facilitated by a plugin provided by either the SSL VPN gateway vendor or the OTP validation server vendor. However the majority of SSL VPN gateways support Remote Authentication Dial In User Service (RADIUS; see RFC 2865) integration. Thus, RADIUS support by the OTP solution obviates the need for the OTP server provider to develop SSL VPN gateway-specific connectors.

[0050] As illustrated in **Figure 5**, one embodiment of the invention relies on existing features of the SSL VPN gateway 515 to integrate online authentication techniques (e.g., such as those described above with respect to Figures 1A-B and 3) without altering the network infrastructure. As illustrated, this embodiment includes an authentication server 202 communicatively coupled to the SSL VPN gateway 515, potentially in the same (or a similar) manner as the OTP validation server 425 described above. The authentication server 202 is also communicatively coupled to a client device 510 with an authentication client 201 for authenticating a user using one or more authentication devices 110-112 (e.g., fingerprint authenticators, voice authenticators, retinal scanning authenticators, etc). While the authentication server 202 is coupled to the authentication client 201 via a browser in **Figure 5** (e.g., in a similar manner as the embodiment shown in Figure 1A), the underlying principles of the invention are not limited to a browser-based implementation.

[0051] In one embodiment, the interaction between the SSL VPN gateway 515, browser 510, and authentication server 202 is as follows. A user opens the web browser 510 and navigates to the SSL VPN gateway 515 which renders a web page 511 containing browser-executable code 512 such as JavaScript. In one embodiment, the browser-executable code 512 triggers authentication by establishing a communication channel with the authentication server 202 and triggering the authentication client 201 to authenticate the user. In one embodiment, the authentication server 202 and client 201 enter into a series of authentication transactions such as those described above with respect to **Figure 3**. For example, the authentication server 202 may generate an authentication request which includes a random challenge (e.g., a cryptographic nonce) and may (or may not) identify the authenticator 110-112 to be used by the authentication client 201 for authentication. In response to receipt of the authentication request, the user may be presented with a graphical user interface (GUI) requesting authentication (e.g., in the form of a web page or a GUI of an authentication application/app). The user then performs the authentication (e.g., swiping a finger on a fingerprint reader, etc). In response, the authentication client 201 generates an authentication response containing a signature over the random challenge with the private key associated with the authenticator. It may also include other relevant data such as the user ID code in the authentication response. Upon receipt of the authentication response, the authentication server 202 validates the signature over the random challenge (e.g., using the public key associated with the authenticator) and confirms the identity of the user. In one embodiment, the JavaScript or other browser executable code 512 passes the above authentication messages between the authentication server 202 and authentication client 201.

[0052] In one embodiment, in response to a successful authentication, the authentication server 202 generates and passes a cryptographic data structure, referred to herein as a "ticket," to the browser 510. In one embodiment, the ticket comprises a random string of digits or other form of one time password (OTP) capable of being submitted to the SSL VPN gateway 515 via the fields of the HTML form 511. For example, as mentioned above, a separate field may be defined in the HTML form 511 for the ticket or the ticket may be appended to the end of the user's static password. Regardless of how the ticket is entered, in one embodiment, the JavaScript or other browser executable code 512 submits ticket to the SSL VPN gateway 515. Once received, the SSL VPN gateway 515 validates the ticket via communication with the authentication server 202 (e.g., providing the ticket to the authentication server and receiving a communication indicating that the ticket is valid). For example, upon receipt of the ticket and other user data from the SSL VPN gateway 515 (e.g., the user ID or other form of identifier), the authentication server 202 may compare the ticket with the ticket provided to the browser 510. If the tickets match, then the authentication server 202 sends an “authentication success” message to the SSL VPN gateway 515. If the tickets do not match, then the authentication server sends an “authentication failure” message to the SSL VPN gateway 515. In one embodiment, the SSL VPN gateway 515 validates the ticket against the authentication server 202 using RADIUS (although the underlying principles of the invention are not limited to any specific protocol). Once validated, the SSL VPN gateway 515 grants the user access to the protected internal network 530.

[0053] Significantly, the transactions between the SSL VPN gateway 515 and authentication server 202 may be implemented in the same manner (e.g., using the same protocols and data fields) as the success/failure messages provided by the OTP validation server 425. As a result, the SSL VPN gateway 515 does not need to be reconfigured to implement the embodiments of the invention described herein, thereby simplifying the implementation and reducing the time and expense associated therewith.

[0054] In the above approach, the SSL VPN login page 511 may be customized to include custom JavaScript or other browser executable code 512 to trigger the authentication. Of course, alternate embodiments may be implemented in the event that the user does not have the authentication client 201 installed.

[0055] In addition, communication with the SSL VPN gateway 515 by the JavaScript or other browser executable code 512 may be facilitated through the same HTML form 511 that the user would normally use to authenticate to the SSL VPN gateway 515. The goal would be to pass the ticket obtained by the JavaScript or other executable code using the existing password or OTP fields in the default SSL VPN’s HTML form 511 (once again, simplifying and reducing the time and expense associated with implementing the above techniques).

[0056] Because these techniques address a well defined problem for a large number of VPN solutions without developing VPN-specific integrations, achieving this integration would require relatively little effort, and allow the authentication service provider (i.e., the entity managing the authentication server 202 and client 201) to provide a packaged solution for delivering secure remote access.

[0057] A method in accordance with one embodiment of the invention is illustrated in **Figure 6**. The method may be implemented within the context of the architecture shown in Figure 5, but is not limited to any specific system architecture.

[0058] At 601, the user opens a browser and navigates to the SSL VPN gateway. At 602, the SSL VPN gateway renders the page containing browser-executable code to trigger authentication on the client. At 603, the browser-executable code establishes a connection with an authentication server to trigger authentication of the user. At 604, the browser-executable code exchanges messages between the authentication client and authentication server to authenticate the user (see, e.g., description above with respect to Figures 1A-B, 3, and 5). Once authenticated, the authentication server returns a ticket.

[0059] At 605, the browser-executable code submits the ticket to the SSL VPN gateway and, at 606, the SSL VPN gateway validates the ticket against the authentication server. As mentioned above, this may involve the authentication server comparing the ticket to the ticket returned in operation 604 to confirm the validity of the ticket (e.g., via RADIUS). At 607, once the ticket is validated, the SSL VPN gateway grants the user access to the protected internal network.

[0060] An alternative approach to integrating with legacy systems is possible in cases where the legacy system accepts the use of digital certificates for authentication. These solutions, such as VPNs or Windows Active Directory using Kerberos, typically involve a client-side component to perform the certificate authentication.

[0061] Unlike the integration approach outlined above, where the integration on the client side was primarily browser-based (e.g., using JavaScript), in this embodiment, elements of the authentication client 201 are integrated into the legacy solution's client side software to achieve the integration; however, as before, no server-side integration is necessary.

[0062] In the specific embodiment shown in **Figure 7**, the authentication client 201 is equipped with a credential provider component 711 for managing signed certificates, which it uses to gain access to network resources via a Kerberos infrastructure 730. For example, in one embodiment, the authentication client 201 may be integrated into the Windows® operating system via the Credential Provider Framework using the credential provider component 730. It should be noted, however, that the underlying principles of the invention are not limited to a Kerberos implementation or any particular type of operating system.

[0063] This embodiment also relies on communication between the authentication server 725 and authentication client 201 which enter into a series of authentication transactions to authenticate the end user (e.g., as described above with respect to Figures 1B and 3). In one embodiment, the active directory 735 and Kerberos infrastructure 730 are configured to trust the root certificate held by the authentication server 725. Once the user is authenticated, the authentication server 725 issues a short-lived certificate comprising a cryptographic public/private key pair which it signs using a root certificate held by the authentication server 725 (e.g., signing the short-lived certificate with the private key of the root certificate). In particular, in one embodiment, the public key of the short-lived certificate is signed with the private key of the root certificate. In addition to the key pairs, the short-lived certificate may also include timestamp/timeout data indicating a length of time for which the short-lived certificate is valid (e.g., 5 minutes, 1 hour, etc).

[0064] In one embodiment, once the credential provider 711 receives the signed short-lived certificate from the authentication server, it enters into a challenge response transaction with the Kerberos infrastructure 730 involving the short-lived certificate. In particular, the Kerberos infrastructure sends a challenge (e.g., random data such as a nonce) to the credential provider 711 which then signs the challenge using the private key of the short-lived certificate. It then sends the short-lived certificate to the Kerberos infrastructure which (1) validates the signature on the short-lived certificate using the public key of the root certificate provided by the authentication server 725 (which it has been configured to trust); and (2) validates the signature over the challenge using the public key from the short-lived certificate. If both signatures are valid, then the Kerberos infrastructure issues a Kerberos ticket to the credential provider 711 which it may then use to gain access to network resources such as file servers, email accounts, etc, managed by the Kerberos infrastructure.

[0065] Using these techniques, the authentication server 725 and client 201 may be integrated without significant modification to the existing active directory 735 and Kerberos infrastructure 730. Rather, all that is required is that the active directory 735/Kerberos infrastructure are configured to trust the root certificate held by the authentication server 725.

[0066] **Figure 8** illustrates one embodiment of a method for integrating an online authentication infrastructure with a legacy system. The method may be implemented within the context of the architecture shown in **Figure 7**, but is not limited to any particular system architecture.

[0067] At 801, the user opens a device such as a Windows device and attempts to log in. At 802, an authentication client is triggered to authenticate the user. In response, the authentication client performs online authentication with an authentication server. For example, as discussed above, the authentication client may have previously registered one or more authentication devices with the server (e.g., a fingerprint authentication device, a voice authentication device, etc). It may then authenticate with the server using a series of transactions such as those described above with respect to Figures 1A-B and 3. For example, the authentication server may send the authentication client an authentication request with a random challenge, which the authentication client signs using a private key associated with the authentication device used. The authentication server may then use the public key to validate the signature.

[0068] Regardless of the specific protocol used for authentication, if authentication is successful, then at 803, the authentication server returns a short-lived digital certificate to the authentication client which is signed using a private key of a root certificate maintained by the authentication server. As mentioned, the root certificate is trusted by the active directory/Kerberos infrastructure.

[0069] At 804, the authentication client then uses the short-lived digital certificate to authenticate to the Kerberos infrastructure. For example, the Kerberos infrastructure may send a challenge (e.g., random data such as a nonce) to the authentication client which then signs the challenge using the private key of the short-lived certificate. It then sends the short-lived certificate to the Kerberos infrastructure which, at 805, validates the signature on the short-lived certificate using the public key of the root certificate provided by the authentication server (which it has been configured to trust); and validates the signature over the challenge using the public key from the short-lived certificate. If both signatures are valid, then the Kerberos infrastructure issues a Kerberos ticket to the authentication client which, at 806, it may then use to gain access to network resources such as file servers, email accounts, etc, managed by the Kerberos infrastructure.

[0070] The end result is that online authentication using an authentication server and authentication client may be used to front-end authentication for a legacy system, gaining all the flexibility of efficient online authentication, without requiring changes to the back end legacy application infrastructure.

[0071] Numerous benefits are realized through the embodiments of the invention described herein including, but not limited to:

[0072] **Reduction in initial integration effort:** Allows a Relying Party to deploy online authentication without re-writing their application to incorporate the online authentication functionality, or to enable integration with a third-party federation server.

[0073] **Simplification of policy administration:** By expressing the authentication policy outside of code, this approach allows the organization to easily update their authentication policies without requiring code changes. Changes to reflect new interpretations of regulatory mandates, or to respond to attacks on existing authentication mechanisms become a simple change in the policy, and can be effected quickly.

[0074] **Enablement of future refinement:** As new authentication devices and mechanisms become available, an organization can evaluate the appropriateness of the devices/mechanisms when addressing new or emerging risks. Integrating a newly-available authentication device only requires adding the device to a policy; no new code has to be written to deploy the new capability immediately, even to legacy applications.

[0075] **Reduction in direct token costs**: Legacy OTP approaches rely on physical hardware tokens that tend to be both relatively expensive on a per-user basis (though they are getting cheaper), and carry the problem of loss/breakage replacement costs. The online authentication approach described herein can dramatically reduce the deployment costs by leveraging capabilities already available on the end user's device, eliminating the cost of acquiring dedicated authentication hardware for each end user.

[0076] **Indirect deployment costs:** OTP approaches typically require an IT administrator to provision the end user's token with the OTP validation server; software-based desktop OTP generators still require helpdesk intervention during initial deployment. The online authentication approach can dramatically reduce the deployment costs by leveraging capabilities already available on the end user's device, and delivering a self-service enrollment model for deployment.

[0077] **Improved end user experience:** OTP approaches require the user to not only carry their OTP generator (which many forget, resulting in additional helpdesk costs to enable temporary access) but also to manually input the OTP into the application. The FIDO approach can dramatically reduce the impact of authentication on the end user by replacing user name / password and OTP entry with something simpler, like swiping a finger over a fingerprint sensor.

Exemplary Data Processing Devices

[0078] **Figure 9** is a block diagram illustrating an exemplary clients and servers which may be used in some embodiments of the invention. It should be understood that while **Figure 9** illustrates various components of a computer system, it is not intended to represent any particular architecture or manner of interconnecting the components as such details are not germane to the present invention. It will be appreciated that other computer systems that have fewer components or more components may also be used with the present invention.

[0079] As illustrated in **Figure 9**, the computer system 900, which is a form of a data processing system, includes the bus(es) 950 which is coupled with the processing system 920, power supply 925, memory 930, and the nonvolatile memory 940 (e.g., a hard drive, flash memory, Phase-Change Memory (PCM), etc.). The bus(es) 950 may be connected to each other through various bridges, controllers, and/or adapters as is well known in the art. The processing system 920 may retrieve instruction(s) from the memory 930 and/or the nonvolatile memory 940, and execute the instructions to perform operations as described above. The bus 950 interconnects the above components together and also interconnects those components to the optional dock 960, the display controller & display device 990, Input/Output devices 980 (e.g., NIC (Network Interface Card), a cursor control (e.g., mouse, touchscreen, touchpad, etc.), a keyboard, etc.), and the optional wireless transceiver(s) 990 (e.g., Bluetooth, WiFi, Infrared, etc.).

[0080] **Figure 10** is a block diagram illustrating an exemplary data processing system which may be used in some embodiments of the invention. For example, the data processing system 1000 may be a handheld computer, a personal digital assistant (PDA), a mobile telephone, a portable gaming system, a portable media player, a tablet or a handheld computing device which may include a mobile telephone, a media player, and/or a gaming system. As another example, the data processing system 1000 may be a network computer or an embedded processing device within another device.

[0081] According to one embodiment of the invention, the exemplary architecture of the data processing system 1000 may used for the mobile devices described above. The data processing system 1000 includes the processing system 1020, which may include one or more microprocessors and/or a system on an integrated circuit. The processing system 1020 is coupled with a memory 1010, a power supply 1025 (which includes one or more batteries) an audio input/output 1040, a display controller and display device 1060, optional input/output 1050, input device(s) 1070, and wireless transceiver(s) 1030. It will be appreciated that additional components, not shown in **Figure 10**, may also be a part of the data processing system 1000 in certain embodiments of the invention, and in certain embodiments of the invention fewer components than shown in **Figure 10** may be used. In addition, it will be appreciated that one or more buses, not shown in **Figure 10**, may be used to interconnect the various components as is well known in the art.

[0082] The memory 1010 may store data and/or programs for execution by the data processing system 1000. The audio input/output 1040 may include a microphone and/or a speaker to, for example, play music and/or provide telephony functionality through the speaker and microphone. The display controller and display device 1060 may include a graphical user interface (GUI). The wireless (e.g., RF) transceivers 1030 (e.g., a WiFi transceiver, an infrared transceiver, a Bluetooth transceiver, a wireless cellular telephony transceiver, etc.) may be used to communicate with other data processing systems. The one or more input devices 1070 allow a user to provide input to the system. These input devices may be a keypad, keyboard, touch panel, multi touch panel, etc. The optional other input/output 1050 may be a connector for a dock.

[0083] Embodiments of the invention may include various steps as set forth above. The steps may be embodied in machine-executable instructions which cause a general-purpose or special-purpose processor to perform certain steps. Alternatively, these steps may be performed by specific hardware components that contain hardwired logic for performing the steps, or by any combination of programmed computer components and custom hardware components.

[0084] Elements of the present invention may also be provided as a machine-readable medium for storing the machine-executable program code. The machine-readable medium may include, but is not limited to, floppy diskettes, optical disks, CD-ROMs, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, magnetic or optical cards, or other type of media/machine-readable medium suitable for storing electronic program code.

[0085] Throughout the foregoing description, for the purposes of explanation, numerous specific details were set forth in order to provide a thorough understanding of the invention. It will be apparent, however, to one skilled in the art that the invention may be practiced without some of these specific details. For example, it will be readily apparent to those of skill in the art that the functional modules and methods described herein may be implemented as software, hardware or any combination thereof. Moreover, although some embodiments of the invention are described herein within the context of a mobile computing environment, the underlying principles of the invention are not limited to a mobile computing implementation. Virtually any type of client or peer data processing devices may be used in some embodiments including, for example, desktop or workstation computers. Accordingly, the scope and spirit of the invention should be judged in terms of the claims which follow.

[0086] Embodiments of the invention may include various steps as set forth above. The steps may be embodied in machine-executable instructions which cause a general-purpose or special-purpose processor to perform certain steps. Alternatively, these steps may be performed by specific hardware components that contain hardwired logic for performing the steps, or by any combination of programmed computer components and custom hardware components.

CLAIMS

We claim:

1. A system comprising:

a gateway configured to restrict access to an internal network;

an authentication server communicatively coupled to the gateway;

a client device with an authentication client having a plurality of authentication devices coupled thereto for authenticating a user, the authentication client configured to establish a communication channel with the authentication server and to register one or more of the authentication devices with the authentication server, the authentication devices usable for performing online authentication with the authentication server following registration;

the authentication client to authenticate the user with the authentication server using one or more of the registered authentication devices in response to an attempt to gain access to the internal network via the gateway;

the authentication server to provide the client device with a cryptographic data structure in response to a successful authentication;

the client device to provide the cryptographic data structure to the gateway as proof of the successful authentication; and

the gateway to validate the cryptographic data structure with the authentication server, wherein upon receiving an indication from the gateway that the cryptographic data structure is valid, the gateway to provide access by the client device to the internal network.

2. The system as in claim 1 further comprising:

a browser configured on the client device, wherein in response to an attempt by the user to access the internal network via the browser, the gateway provides browser-executable code which, when executed by the browser, triggers the authentication by establishing a communication channel between the authentication server and the authentication client.

3. The system as in claim 2 wherein the cryptographic data structure comprises a ticket.

4. The system as in claim 3 wherein the gateway is configured to provide the browser with an Hypertext Markup Language (HTML) form which includes the browser-

executable code, the HTML form having one or more fields for entry of a user name and one or more passwords.

5. The system as in claim 4 wherein the ticket comprises a random string of digits or other form of one time password (OTP) capable of being submitted to the gateway via a field of the HTML form.

6. The system as in claim 1 wherein the authentication server validates the cryptographic data structure provided by the gateway using a key associated with the authentication device used for authentication.

7. The system as in claim 6 wherein the gateway validates the cryptographic data structure with the authentication server using a Remote Authentication Dial In User Service (RADIUS) protocol.

8. The system as in claim 1 wherein the gateway comprises a secure sockets layer (SSL) virtual private network (VPN) gateway.

9. A system comprising:

a network security infrastructure to provide network security services for an internal network;

an authentication server communicatively coupled to the existing network security infrastructure;

a client device with an authentication client having a plurality of authentication devices coupled thereto for authenticating a user, the authentication client configured to establish a communication channel with the authentication server and to register one or more of the authentication devices with the authentication server, the authentication devices usable for performing online authentication with the authentication server following registration;

the authentication client to authenticate the user with the authentication server using one or more of the registered authentication devices in response to an attempt to gain access to the internal network;

the authentication server to provide the client device with a cryptographic data structure in response to a successful authentication;

the client device to use the cryptographic data structure to authenticate with the network security infrastructure; and

the network security infrastructure to validate the cryptographic data structure based on a trust relationship established with the authentication server, the network security infrastructure to provide access by the client device to the internal network upon validation of the cryptographic data structure.

10. The system as in claim 9 wherein the cryptographic data structure comprises a digital certificate signed with a root certificate held by the authentication server, wherein the trust relationship comprises the network security infrastructure trusting signatures generated using the root certificate.

11. The system as in claim 10 wherein the digital certificate comprises a public/private key pair generated by the authentication server.

12. The system as in claim 11 wherein the digital certificate includes a timestamp or other data indicating a length of time for which the digital certificate is valid.

13. The system as in claim 12 wherein to use the digital certificate to authenticate with the network security infrastructure, the authentication client is to sign a challenge provided by the network security infrastructure.

14. The system as in claim 13 wherein the network security infrastructure is configured to validate the signature on the digital certificate using a public key of the root certificate provided by the authentication server with which it has the trust relationship; and is further configured to validate the signature over the challenge using a public key from the digital certificate.

15. The system as in claim 9 wherein the network security infrastructure comprises a Microsoft Active Directory and Kerberos infrastructure.

16. The system as in claim 9 further comprising:

a gateway coupling the authentication client to the authentication server.

17. A method comprising:

configuring a gateway to restrict access to an internal network;

communicatively coupling an authentication server to the gateway;

configuring an authentication client of a client device to establish a communication channel with the authentication server and to register one or more authentication devices with the authentication server, the authentication devices usable for performing online authentication with the authentication server following registration;

the authentication client to authenticate the user with the authentication server using one or more of the registered authentication devices in response to an attempt to gain access to the internal network via the gateway;

the authentication server to provide the client device with a cryptographic data structure in response to a successful authentication;

the client device to provide the cryptographic data structure to the gateway as proof of the successful authentication; and

the gateway to validate the cryptographic data structure with the authentication server, wherein upon receiving an indication from the gateway that the cryptographic data structure is valid, the gateway to provide access by the client device to the internal network.

18. The method as in claim 17 wherein a browser is configured on the client device, wherein in response to an attempt by the user to access the internal network via the browser, the gateway provides browser-executable code which, when executed by the browser, triggers the authentication by establishing a communication channel between the authentication server and the authentication client.

19. The method as in claim 18 wherein the cryptographic data structure comprises a ticket.

20. The method as in claim 19 wherein the gateway is configured to provide the browser with an Hypertext Markup Language (HTML) form which includes the browser-executable code, the HTML form having one or more fields for entry of a user name and one or more passwords.

21. The method as in claim 20 wherein the ticket comprises a random string of digits or other form of one time password (OTP) capable of being submitted to the gateway via a field of the HTML form.

22. The method as in claim 17 wherein the authentication server validates the cryptographic data structure provided by the gateway using a key associated with the authentication device used for authentication.

23. The method as in claim 22 wherein the gateway validates the cryptographic data structure with the authentication server using a Remote Authentication Dial In User Service (RADIUS) protocol.

24. The method as in claim 17 wherein the gateway comprises a secure sockets layer (SSL) virtual private network (VPN) gateway.

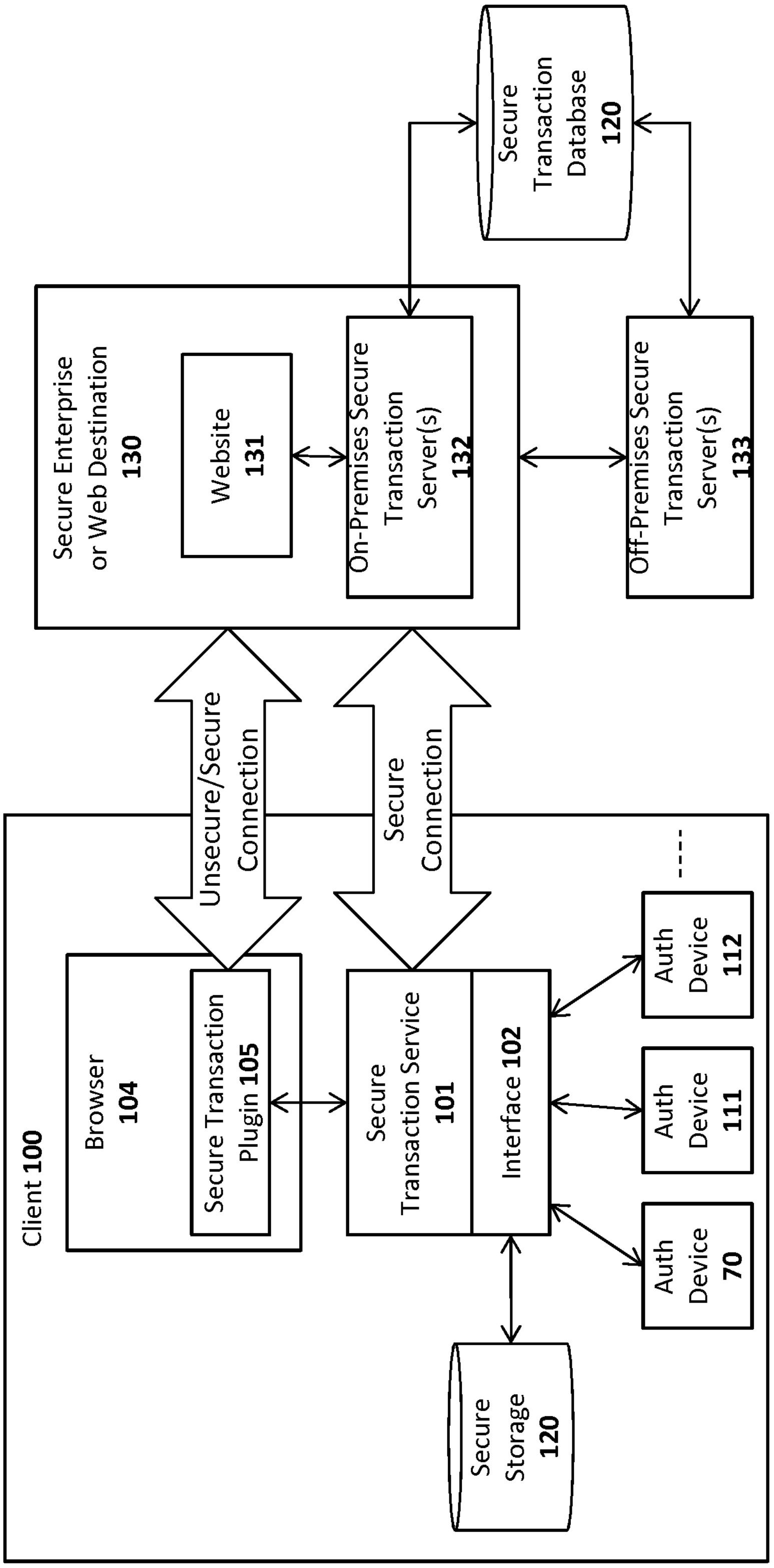
Client 100
Browser 104
Secure Transaction Plugin 105
Secure Transaction Service 101
Interface 102
Secure Storage 120
Auth Device 70
Auth Device 111
Auth Device 112
Unsecure/Secure Connection
Secure Connection
Secure Enterprise or Web Destination 130
Website 131
On-Premises Secure Transaction Server(s) 132
Off-Premises Secure Transaction Server(s) 133
Secure Transaction Database 120

FIG. 1A

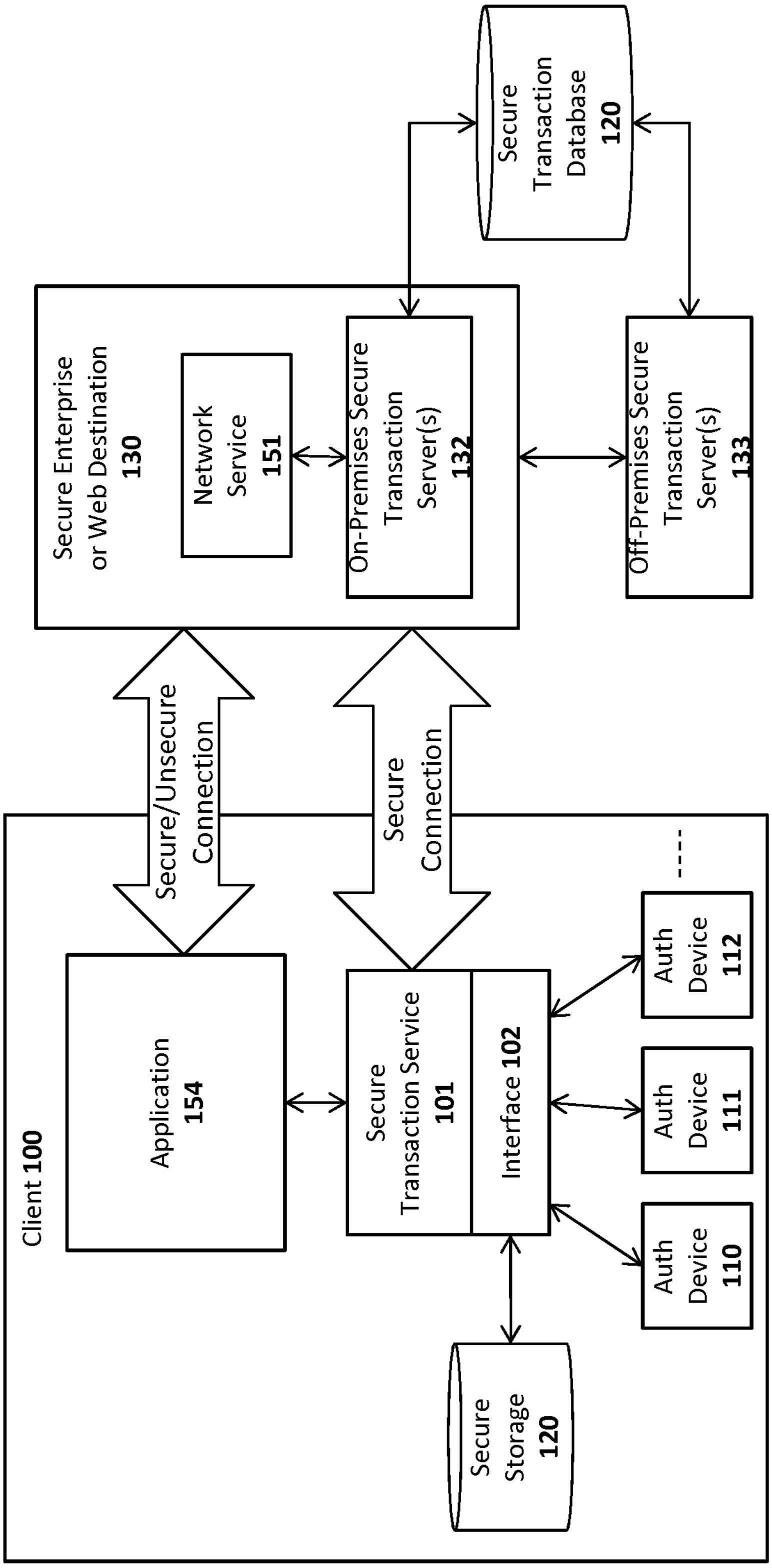
Secure Enterprise or Web Destination 130
Network Service 151
On-Premises Secure Transaction Server(s) 132
Secure Transaction Database 120
Off-Premises Secure Transaction Server(s) 133
Secure/Unsecure Connection
Secure Connection
Client 100
Application 154
Secure Transaction Service 101
Interface 102
Auth Device 110
Auth Device 111
Auth Device 112
Secure Storage 120

FIG. 1B

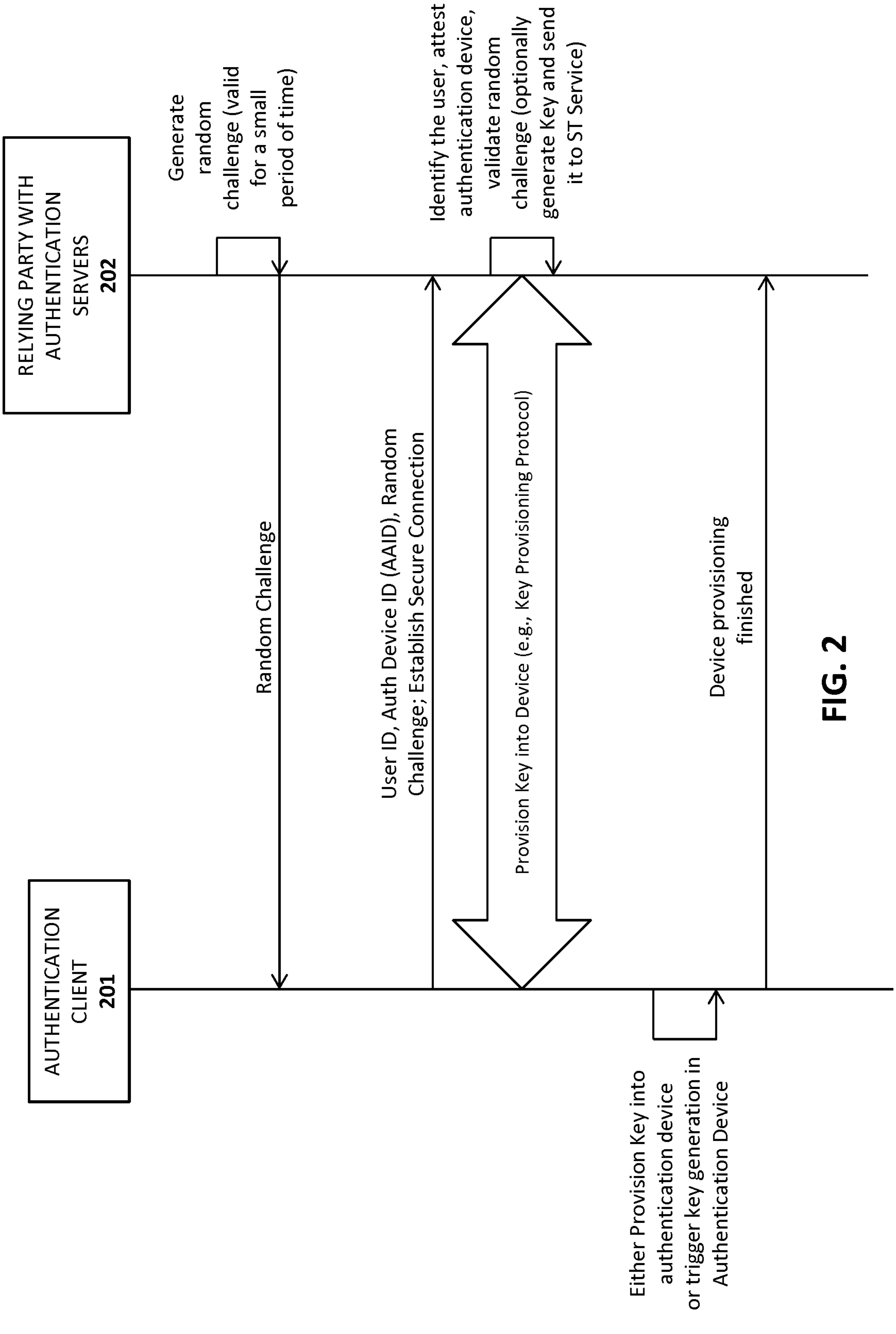
RELYING PARTY WITH AUTHENTICATION SERVERS
202
AUTHENTICATION CLIENT
201
Generate random challenge (valid for a small period of time)
Random Challenge
User ID, Auth Device ID (AAID), Random Challenge; Establish Secure Connection
Identify the user, attest authentication device, validate random challenge (optionally generate Key and send it to ST Service)
Provision Key into Device (e.g., Key Provisioning Protocol)
Either Provision Key into authentication device or trigger key generation in Authentication Device
Device provisioning finished

FIG. 2

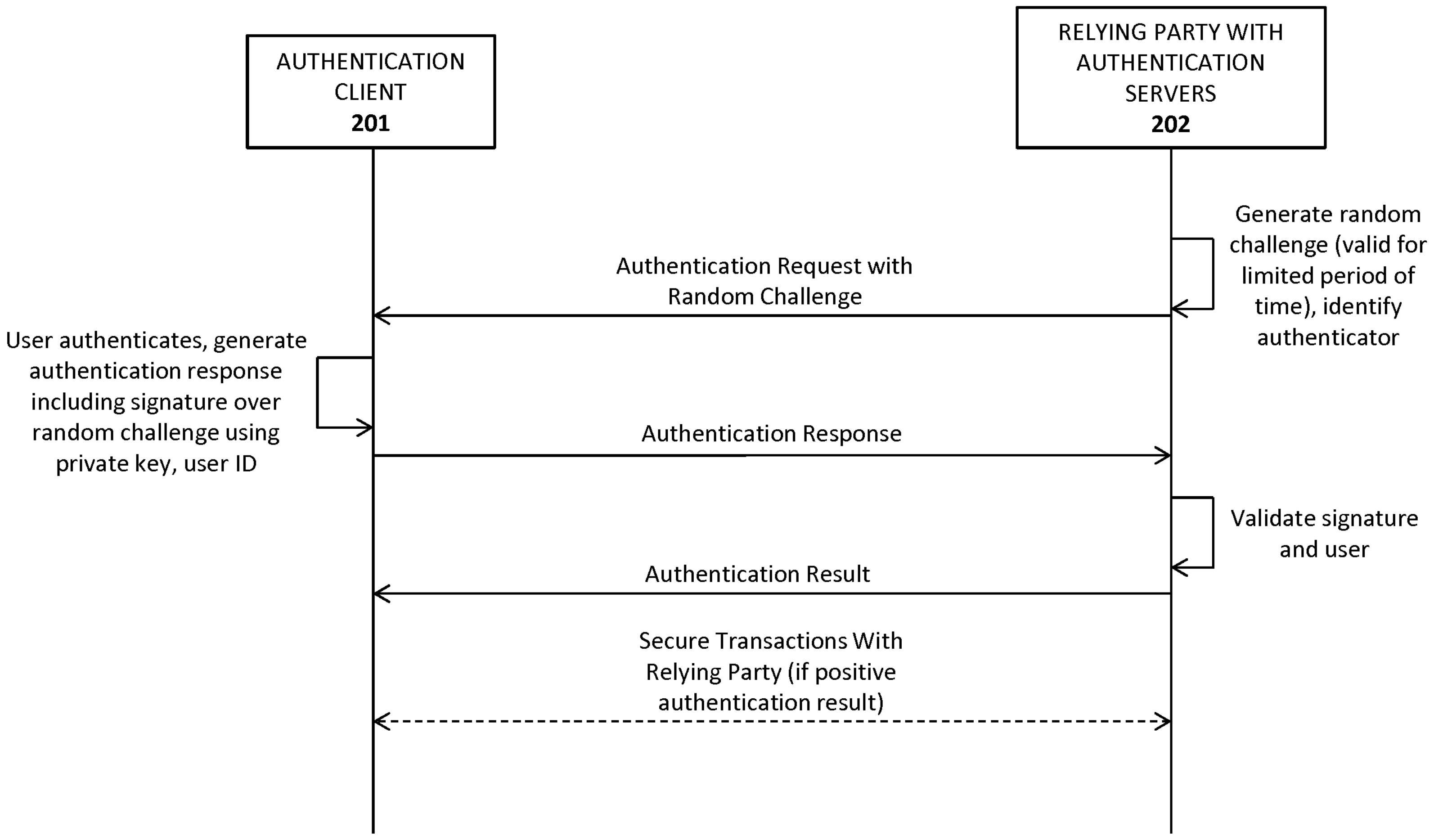
AUTHENTICATION CLIENT
201
RELYING PARTY WITH AUTHENTICATION SERVERS
202
Generate random challenge (valid for limited period of time), identify authenticator
Authentication Request with Random Challenge
User authenticates, generate authentication response including signature over random challenge using private key, user ID
Authentication Response
Validate signature and user
Authentication Result
Secure Transactions With Relying Party (if positive authentication result)

FIG. 3

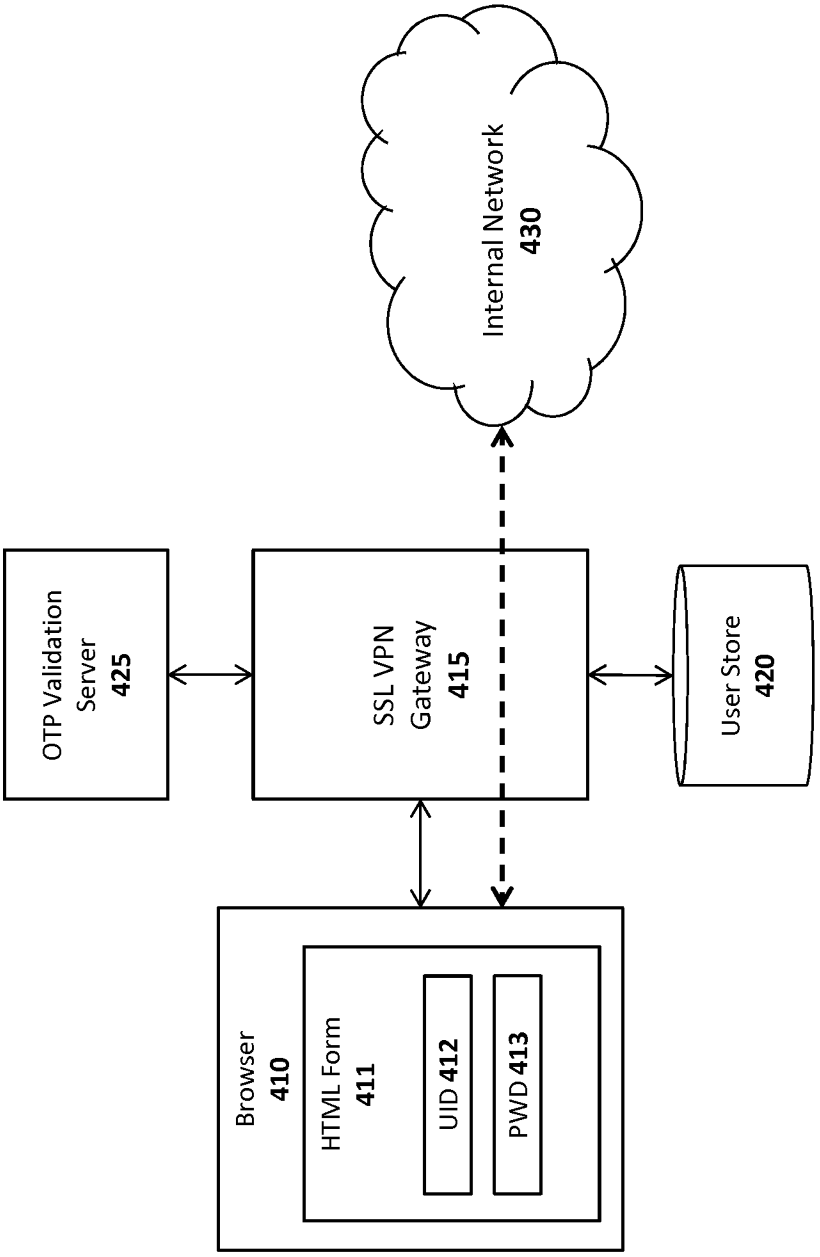
Internal Network
430
OTP Validation Server
425
SSL VPN Gateway
415
User Store
420
Browser
410
HTML Form
411
UID 412
PWD 413

FIG. 4

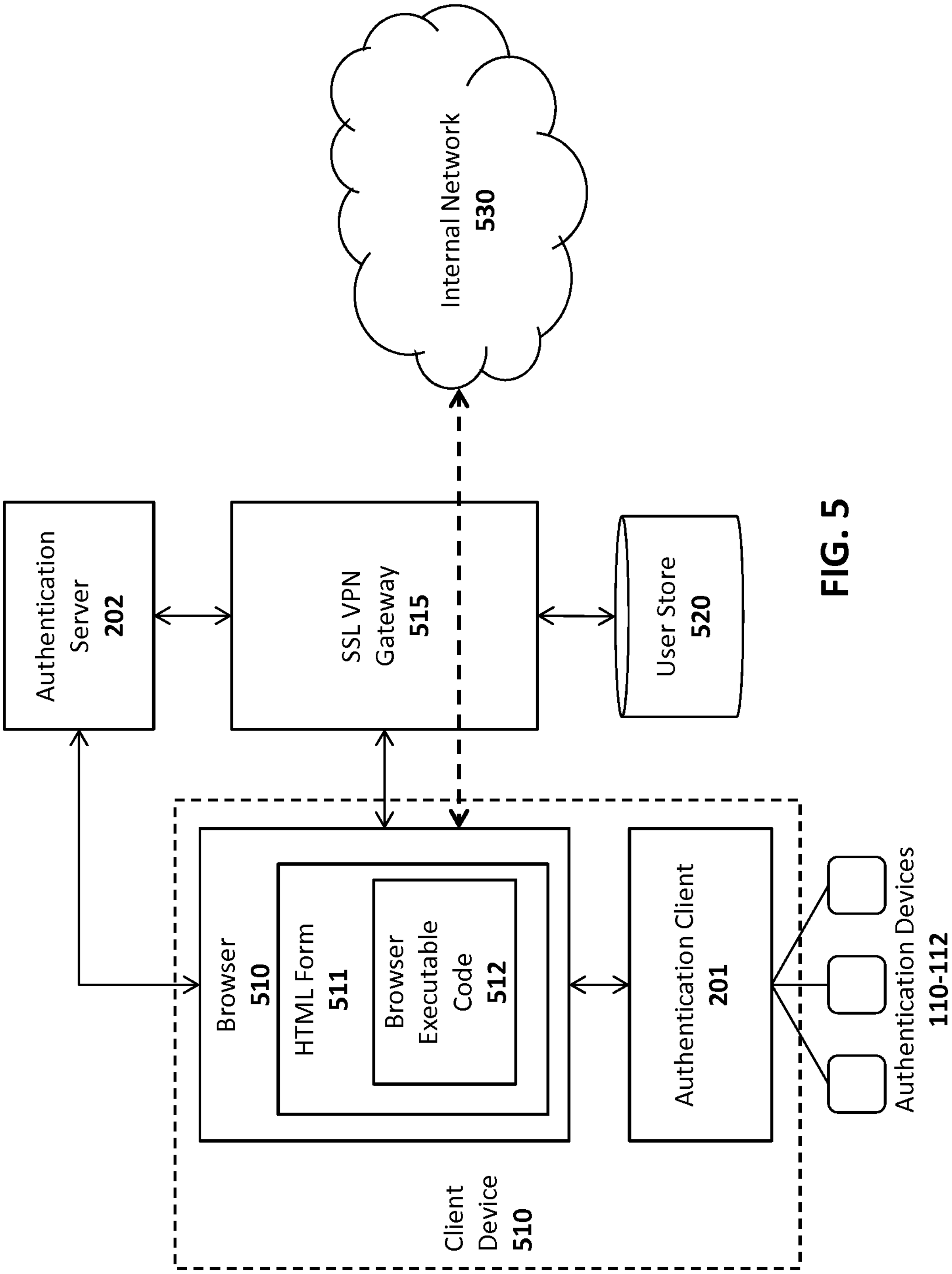
Authentication Server
202
SSL VPN Gateway
515
User Store
520
Internal Network
530
Client Device
510
Browser
510
HTML Form
511
Browser Executable Code
512
Authentication Client
201
Authentication Devices
110-112

FIG. 5

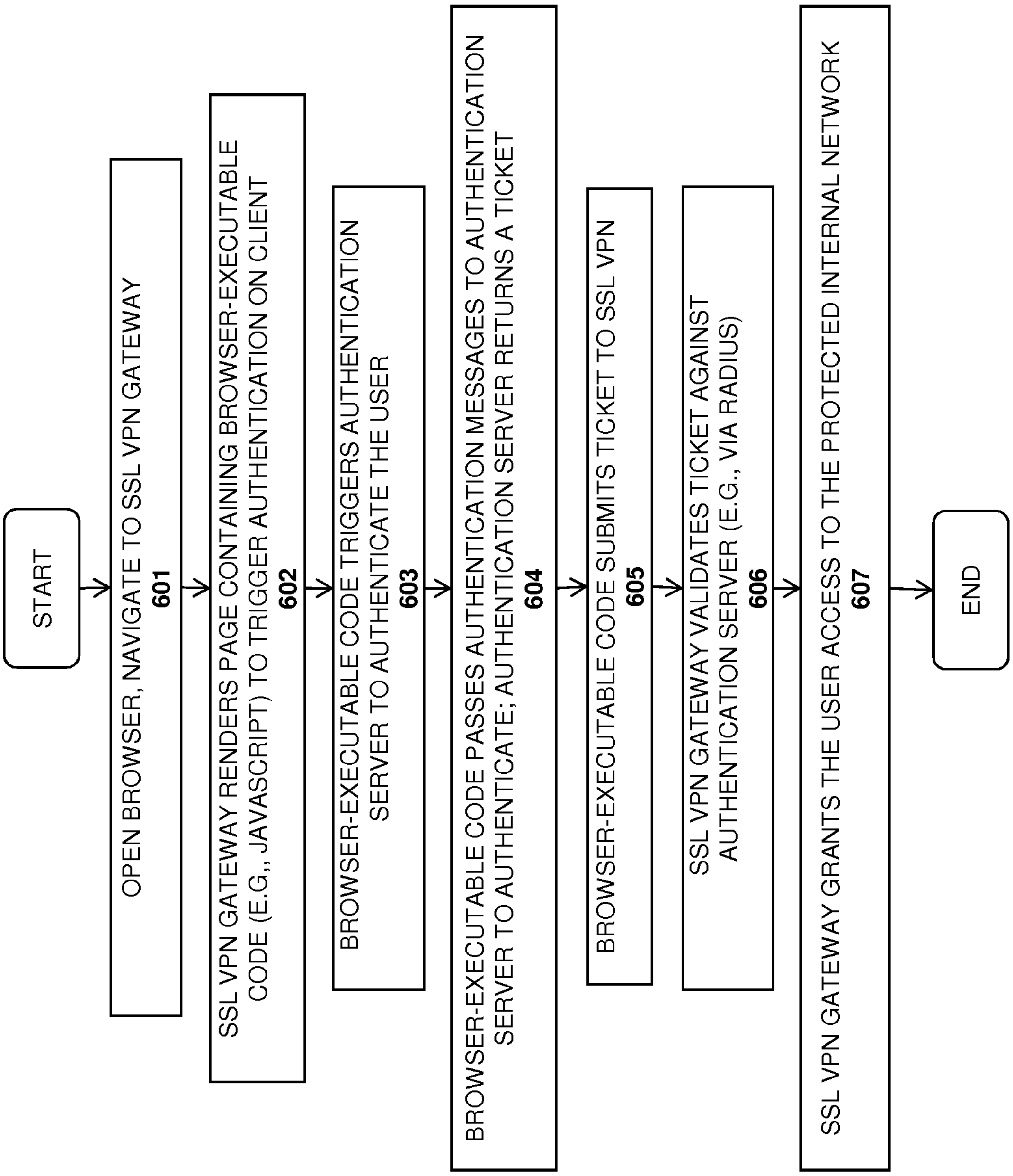
START
OPEN BROWSER, NAVIGATE TO SSL VPN GATEWAY
601
SSL VPN GATEWAY RENDERS PAGE CONTAINING BROWSER-EXECUTABLE CODE (E.G,., JAVASCRIPT) TO TRIGGER AUTHENTICATION ON CLIENT
602
BROWSER-EXECUTABLE CODE TRIGGERS AUTHENTICATION SERVER TO AUTHENTICATE THE USER
603
BROWSER-EXECUTABLE CODE PASSES AUTHENTICATION MESSAGES TO AUTHENTICATION SERVER TO AUTHENTICATE; AUTHENTICATION SERVER RETURNS A TICKET
604
BROWSER-EXECUTABLE CODE SUBMITS TICKET TO SSL VPN
605
SSL VPN GATEWAY VALIDATES TICKET AGAINST AUTHENTICATION SERVER (E.G., VIA RADIUS)
606
SSL VPN GATEWAY GRANTS THE USER ACCESS TO THE PROTECTED INTERNAL NETWORK
607
END

FIG. 6

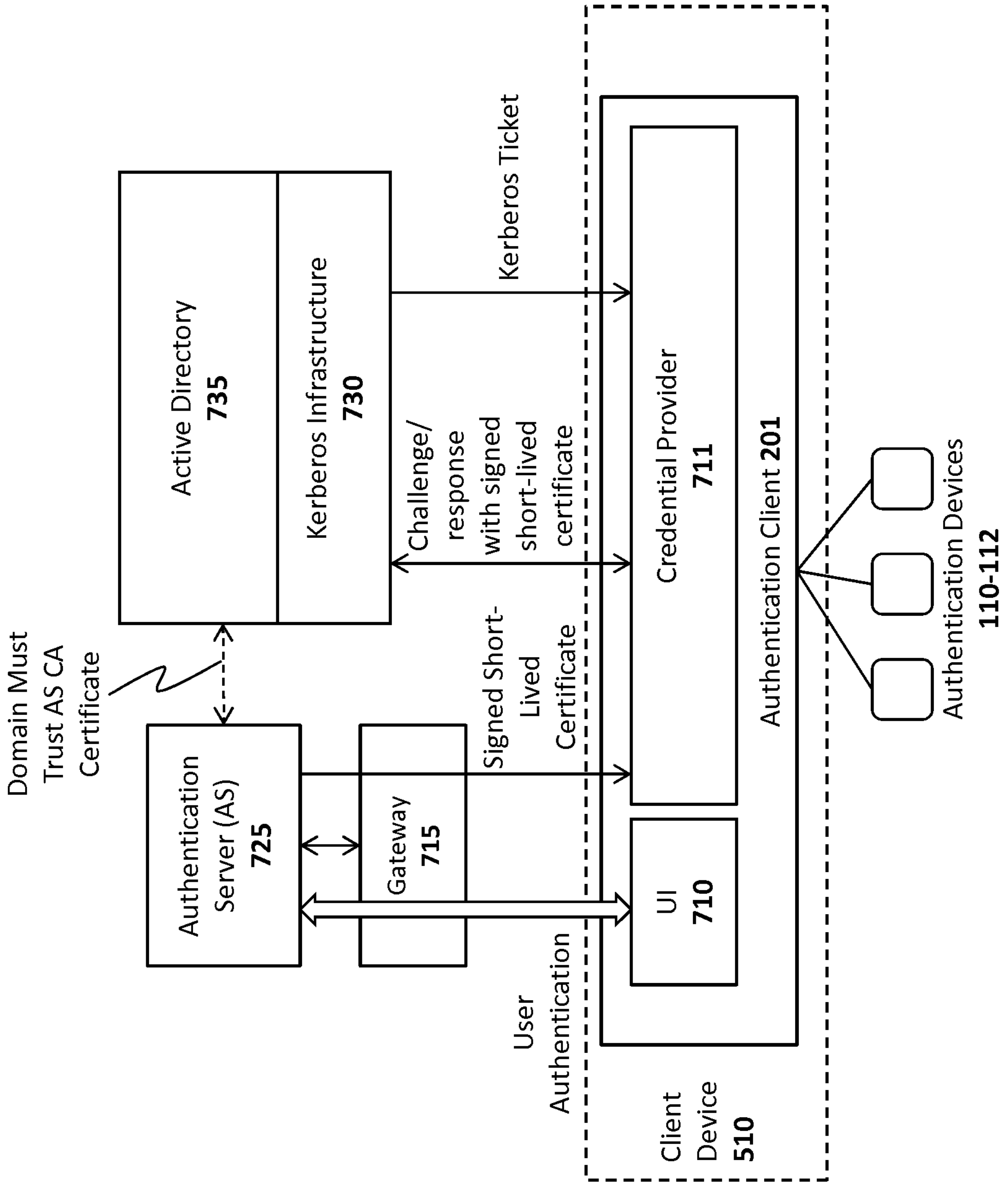
Domain Must Trust AS CA Certificate
Active Directory 735
Kerberos Infrastructure 730
Authentication Server (AS) 725
Gateway 715
Kerberos Ticket
Challenge/ response with signed short-lived certificate
Signed Short-Lived Certificate
User Authentication
Credential Provider 711
UI 710
Authentication Client 201
Client Device 510
Authentication Devices 110-112

FIG. 7

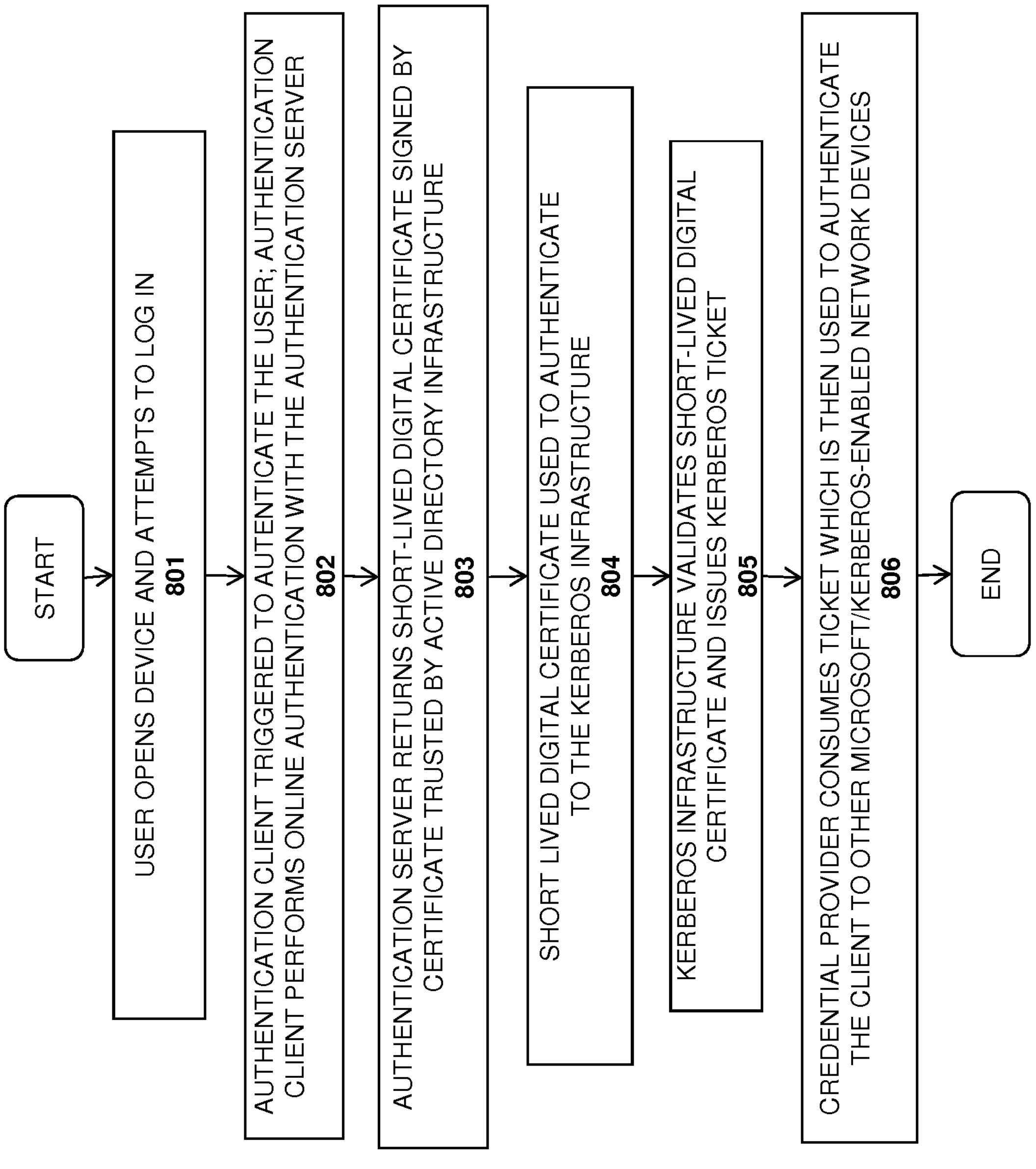
START
USER OPENS DEVICE AND ATTEMPTS TO LOG IN
801
AUTHENTICATION CLIENT TRIGGERED TO AUTENTICATE THE USER; AUTHENTICATION CLIENT PERFORMS ONLINE AUTHENTICATION WITH THE AUTHENTICATION SERVER
802
AUTHENTICATION SERVER RETURNS SHORT-LIVED DIGITAL CERTIFICATE SIGNED BY CERTIFICATE TRUSTED BY ACTIVE DIRECTORY INFRASTRUCTURE
803
SHORT LIVED DIGITAL CERTIFICATE USED TO AUTHENTICATE TO THE KERBEROS INFRASTRUCTURE
804
KERBEROS INFRASTRUCTURE VALIDATES SHORT-LIVED DIGITAL CERTIFICATE AND ISSUES KERBEROS TICKET
805
CREDENTIAL PROVIDER CONSUMES TICKET WHICH IS THEN USED TO AUTHENTICATE THE CLIENT TO OTHER MICROSOFT/KERBEROS-ENABLED NETWORK DEVICES
806
END

Fig. 8

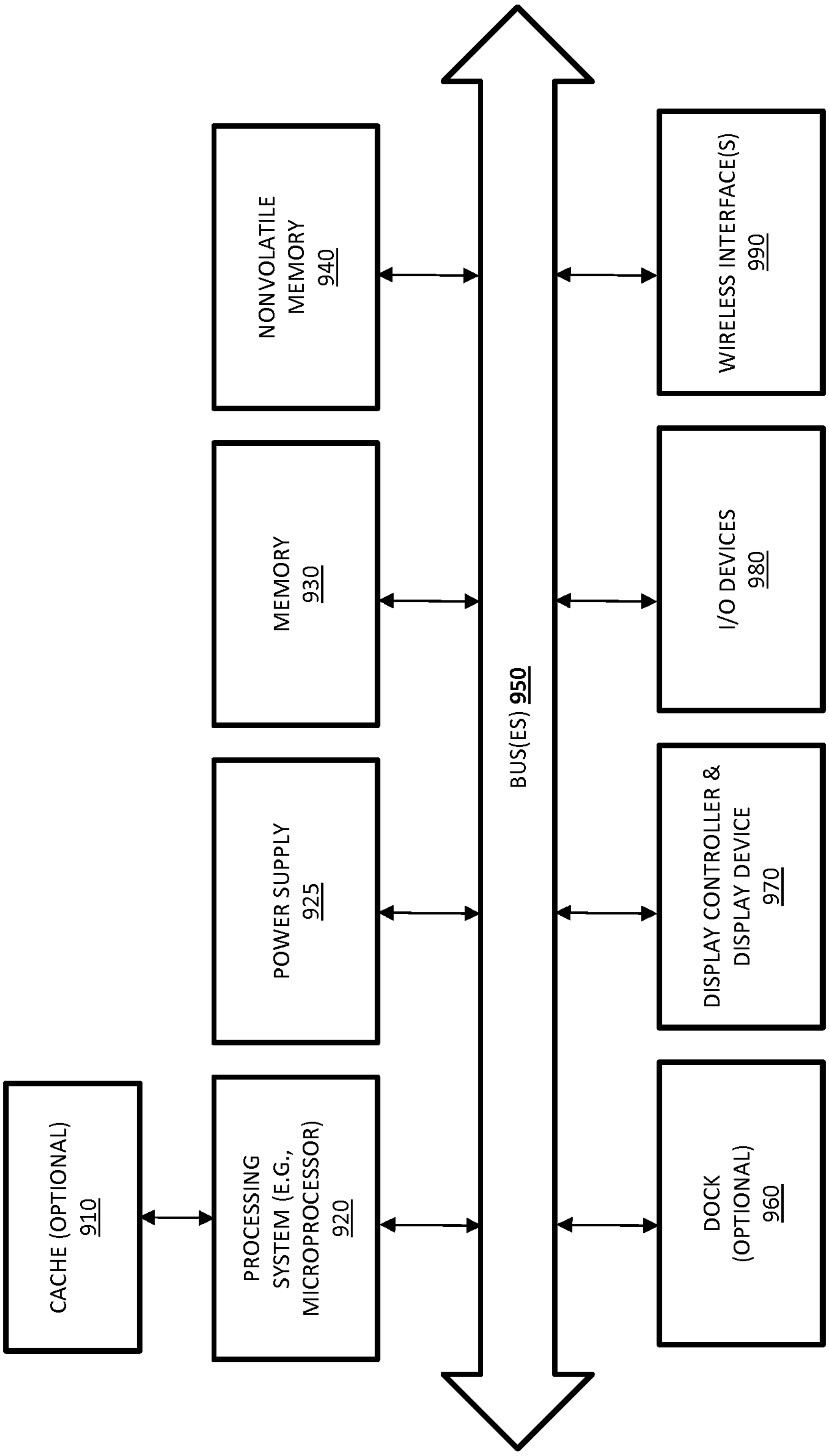
CACHE (OPTIONAL)
910
PROCESSING SYSTEM (E.G., MICROPROCESSOR)
920
POWER SUPPLY
925
MEMORY
930
NONVOLATILE MEMORY
940
BUS(ES) 950
DOCK (OPTIONAL)
960
DISPLAY CONTROLLER & DISPLAY DEVICE
970
I/O DEVICES
980
WIRELESS INTERFACE(S)
990

FIG. 9

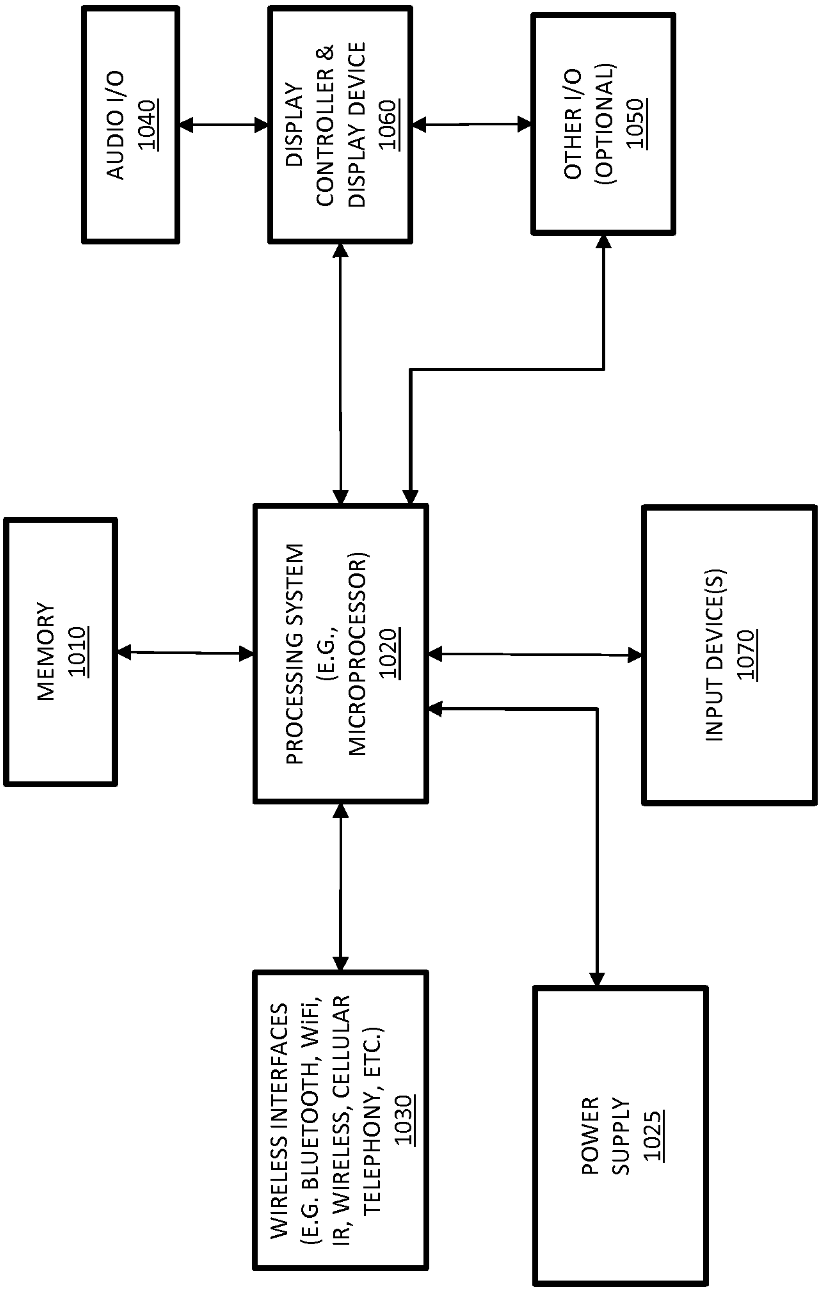
AUDIO I/O
1040
DISPLAY CONTROLLER & DISPLAY DEVICE
1060
OTHER I/O (OPTIONAL)
1050
MEMORY
1010
PROCESSING SYSTEM (E.G., MICROPROCESSOR)
1020
INPUT DEVICE(S)
1070
WIRELESS INTERFACES (E.G. BLUETOOTH, WiFi, IR, WIRELESS, CELLULAR TELEPHONY, ETC.)
1030
POWER SUPPLY
1025

FIG. 10

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 15/50348

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - G06F 7/04, H04L 29/06 (2015.01)

CPC - H04L 63/08

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)
CPC: H04L63/08; IPC(8): G06F 7/04, H04L 29/06 (2015.01); USPC: 726/4

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
CPC: H04L63/08, G06F21/6218, H04L63/102, H04L63/10, H04W12/06; USPC: 726/4, 726/2, 726/3, 726/14, 709/223, 709/224, 709/225
(keyword limited; terms below)

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
PatBase; Google Scholar
Search Terms: gateway, access, internal, network, authentication, server, client, registered, cryptographic, validate.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X -- Y	US 2014/0189808 A1 (MAHAFFEY et al.) 03 July 2014 (03.07.2014), entire document, especially abstract and para [0041]-[0043], [0045], [0047], [0050], [0055]-[0056], [0058]-[0059], [0062]-[0065], [0067]-[0068], [0073], [0075], [0089]-[0090], [0094]-[0103], [0119], [0123], [0144], Fig. 3C, Fig. 3D, Fig. 3E, Fig. 3F, Fig. 4A, Fig. 4B, Fig. 9.	1-6, 9, 16-22 -------------------------- 7-8, 10-15, 23-24
Y	US 2011/0078443 A1 (GREENSTEIN et al.) 31 March 2011 (31.03.2011), entire document, especially abstract and para [0018]-[0019], [0025].	7, 23
Y	US 2012/0084566 A1 (CHIN et al.) 05 April 2012 (05.04.2012), entire document, especially abstract and para [0102]-[0103], [0196], [0215], Fig. 19.	8, 15, 24
Y	US 2014/0258125 A1 (GERBER et al.) 11 September 2014 (11.09.2014), entire document, especially abstract and para [0056], [0058], [0060], [0075], [0130].	10-14

□ 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 27 October 2015 (27.10.2015)	Date of mailing of the international search report 22 DEC 2015
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. 571-273-8300	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774