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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 February 2008 (07.02.2008)

PCT

(10) International Publication Number
WO 2008/016800 A2

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

(21) International Application Number:
PCT/US2007/074130

(22) International Filing Date: 23 July 2007 (23.07.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/496,903 1 August 2006 (01.08.2006) US

(71) Applicant *(for all designated States except US)*: **CISCO TECHNOLOGY, INC.** [US/US]; 170 W. Tasman Drive, San Jose, CA 95134-1706 (US).

(72) Inventors: **CAM-WINGET, Nancy**; 325 Martens Avenue, Mountain View, CA 94040 (US). **ZHOU, Hao**; 7316 Capilano Drive, Solon, Ohio 44139 (US). **O'HARA, JR., Robert, B.**; 4233 Rivermark Parkway, Santa Clara, CA 95054 (US). **CALHOUN, Patrice, R.**; 1937 Via Di Salerno, Pleasanton, CA 94566 (US). **STIEGLITZ, Jeremy**; 10666 Laurel Street, Menlo Park, California 94025 (US).

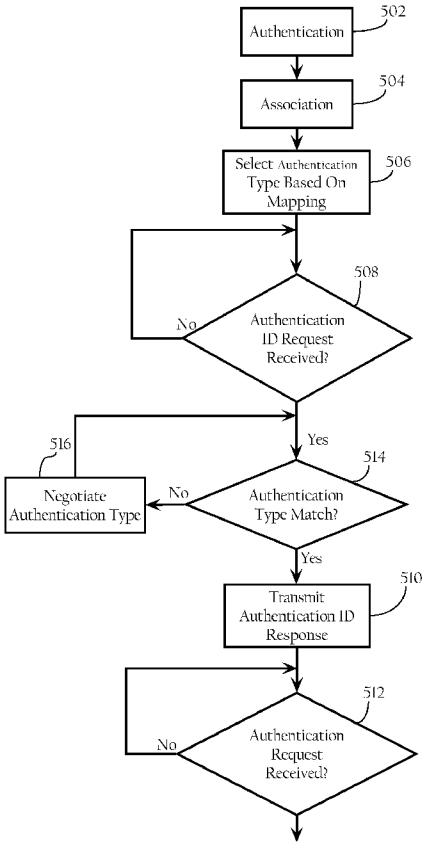
(74) Agent: **SPOLYAR, Mark, James**; Law Office of Mark J. Spolyar, 2200 Cesar Chavez Street, Suite 8, San Francisco, CA 94124 (US).

(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),

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR SELECTING AN APPROPRIATE AUTHENTICATION METHOD ON A CLIENT



(57) Abstract: In one embodiment, a method for facilitating authentication and ease the configuration of authentication includes receiving a credential type selection and selecting one or more authentication types based on the credential type selection and one or more policies set by the administrators. The policies can be pre- configured or dynamically pushed or fetched and updated to the client.

WO 2008/016800 A2

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).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *without international search report and to be republished upon receipt of that report*

PATENT APPLICATION

Method and Apparatus for Selecting an Appropriate Authentication Method on a Client

Inventor(s): Nancy Cam-Winget, a citizen of the United States, residing at 325 Martens Avenue, Mountain View, CA 94040.

Hao Zhou, a citizen of the People's Republic of China, residing at 7316 Capilano Drive, Solon, OH 44139.

Robert O'Hara, a citizen of the United States, residing at 4233 Rivermark Parkway, Santa Clara, CA 95054.

Patrice Calhoun, a citizen of the United States, residing at 3629 Huff Court, Pleasanton, CA 94588

Jeremy Stieglitz, a citizen of the United States, residing at 10666 Laurel Street, Menlo Park, CA 94025

Assignee: Cisco Technology, Inc.
170 West Tasman Dr.
San Jose, CA 95134

Prepared By: Law Office of Mark J. Spolyar
2200 Cesar Chavez Street, Suite 8
San Francisco, CA 94124
(415) 826-7966
(415) 480-1780 (fax)
Customer No.: 30505

Attorney Docket: 6571/53982PCT

Method and Apparatus for Selecting an Appropriate Authentication Method on a Client

FIELD OF THE INVENTION

[0001] The present invention relates to both wireless and wire networks and, more particularly, to methods, apparatuses, and systems directed to authenticating clients in a network before allowing access.

BACKGROUND OF THE INVENTION

[0002] In local area network (LAN) configurations, a user is typically required to select a particular Extensible Authentication Protocol (EAP) method for authentication to gain network access. A problem with EAP selection is that users typically do not have much more knowledge about enterprise information technology (IT) requirements and, in particular, the appropriate EAP method. Accordingly, the required EAP method is typically made by a network administrator of an IT department, since a given EAP method is based on various complex technical considerations and requirements. Furthermore, there are an increasing number of EAP method types, and sometimes even multiple available methods suitable for the same type of user credentials. This makes it increasingly harder for a user to select the correct EAP method. Furthermore, there is a risk that if the user picks the wrong EAP type, not only would the network connection not be established, but there would also be an increased risk of user credentials being compromised (e.g., if a weak EAP type is being negotiated with a rogue device). Furthermore, by requiring EAP type configuration on the wireless client, migration to newer EAP types becomes a burden. Using state of the art products today, such a migration would require that users manually modify their network profiles.

DESCRIPTION OF THE DRAWINGS

[0003] Figure 1A is a topological diagram of the components in a wireless local area network (WLAN) system according to one implementation of the present invention.

[0004] Figure 1B illustrates a hierarchical wireless network including a central controller, according to one implementation of the present invention.

[0005] Figure 1C illustrates for didactic purposes a hardware system, which may be used to implement a central controller.

[0006] Figure 2 illustrates for didactic purposes a hardware system, which may be used to implement an authentication server.

[0007] Figure 3 illustrates for didactic purposes a hardware system, which may be used to implement a wireless client.

[0008] Figure 4 is a flow chart illustrating a process flow, according to one implementation of the present invention, implemented at a wireless client.

[0009] Figure 5A is a flow chart illustrating a process flow, according to one implementation of the present invention, implemented by a client configuration application.

[0010] Figure 5B is a flow chart illustrating a process flow, according to another implementation of the present invention, implemented by a client configuration application.

[0011] Figure 5C is a flow chart illustrating a process flow, according to another implementation of the present invention, implemented by a client configuration application.

DESCRIPTION OF EXEMPLARY EMBODIMENTS

A. Overview

[0012] Particular embodiments of the present invention facilitate authentication of clients in a network. According to one implementation, the present invention facilitates the configuration of one or more authentication attributes associated with client-side authentication functions. In one implementation, a user need only provide the wireless network infrastructure with the type of user credentials being used (e.g., user name, password, one time password, secure token, certificate, etc.), and a client utility automatically selects the appropriate authentication method based on the user credentials of the client, minimum security requirements based on the type of network (wired, wireless, dial-up etc.), and based on policies set by the network administrator. In one implementation, the authentication type may be an Extensible Authentication Protocol (EAP) method. As described in detail below, in one implementation, a network administrator may set policies mapping authentication types with sets of user credentials and may optionally set additional policies ranking authentication types by criteria (e.g., best security, best performance, etc.). The network client utility may include such policies in a policy configuration, which the network infrastructure transmits to the client during a configuration process. Accordingly, based on the user credentials that the user provides, a client utility/application may then select an authentication type based on those user credentials, minimum security requirements based on the type of network (wired, wireless, dial-up etc.), and the policy configuration. In one implementation, if more than one authentication type is available for a given set of user credentials, the client configuration application may select multiple authentication types and an order of preferences. In one implementation, the processes described above may be extended to wireless or wired networks, or any network the EAP is being used.

B. Exemplary Wireless Network System Architecture

B.1. Network Topology

[0013] A network environment including a wireless local area network (WLAN) according to one implementation of the present invention is shown in Figure 1A. In a specific embodiment of the present invention, the system includes an authentication server 20, a local area network (LAN) 30, a router 32, and wireless access points 50a, 50b, 50c, and 50d (collectively referred to as wireless access points 50). LAN 30 is implemented by a switch (or an array of switches) and/or other network devices, such as a bridge.

[0014] As Figure 1A illustrates, these network elements are operably connected to a network 52. Network 52, in one implementation, generally refers to a computer network, such as a LAN, a WAN, etc., that includes one or more intermediate network devices (e.g., routers, switches, etc.), which allow for the transmission of messages between authentication server 20 and wireless clients via wireless access points 50. Of course, network 52 can include a variety of network segments, transmission technologies and components, such as terrestrial WAN links, satellite links, optical fiber links, and cellular links. Network 52 could also be a campus LAN. LAN 30 may be a LAN, LAN segments implemented by an Ethernet switch (not shown), or an array of switches having multiple ports to which wireless access points 50 are connected. The wireless access points 50 are typically connected to switch ports via Ethernet links; however, other link layer connection protocols or communication means can be employed. Figure 1A illustrates one possible network environment in which the invention may operate; however, other implementations are possible. For example, although WLAN management server 20 is illustrated as being on a different LAN or LAN segment, it may be co-located with wireless access points 50.

[0015] The wireless access points 50 are operative to wirelessly communicate with remote wireless client devices 60a, 60b, 60c, and 60d. In one implementation, the wireless access points 50 implement the wireless network protocol specified in the IEEE 802.11 WLAN specification. The wireless access points 50 may be autonomous or so-called "fat" wireless access points, or light-

weight wireless access points operating in connection with a wireless switch (see Figure 1B). In addition, the network infrastructure may also include a Wireless LAN Solution Engine (WLSE) offered by Cisco Systems, Inc. of San Jose, California or another wireless network management system. In some implementations, the network infrastructure may also include one or more Wireless Control System (WCS) nodes operative to manage one or more wireless switches and access points. Of course, configuration and management information can be obtained in a variety of manners without departing from the scope of the present invention.

B.2. Central Controller

[0016] Figure 1B illustrates a hierarchical wireless network including a central controller 70 according to one implementation of the present invention. In one implementation, the central controller 70 may be implemented as a wireless domain server (WDS) or, alternatively, as a wireless switch. If the central controller 70 is implemented with a WDS, the central controller 70 is operative to communicate with autonomous or so-called "fat" wireless access points. If the central controller 70 is implemented with a wireless switch, the central controller 70 is operative to communicate with light-weight wireless access points. As Figure 1B illustrates, a central controller 70 may be directly connected to one or more access points 50. Alternatively, a central controller 43 may be operably connected to one or more access points over a switched and/or routed network environment, as Figure 1A illustrates.

[0017] Figure 1C illustrates for didactic purposes a hardware system 100, which may be used to implement a central controller 70 of Figure 1B. As Figure 1C shows, in one implementation, the central control elements each comprise a switch function or fabric 102 comprising a network interface 104a (e.g., a Ethernet adapter) for connection to network 52 and network interfaces 104b, 104c, and 104d for connection to wireless access points. This switch function or fabric is implemented to facilitate connection to the access elements. Central controller 70, in one implementation, further comprises a processor 106, a memory 108, one or more software modules stored in memory 108, including

instructions for performing the functions described herein, and a system bus 110 operably connecting these components. The central control elements may optionally include an administrative network interface 112 allowing for administrative access for such purposes as configuration and diagnostic access.

B.2. Authentication Server

[0018] Figure 2 illustrates for didactic purposes a hardware system 200, which may be used to implement authentication server 20 of Figure 1A. In one implementation, hardware system 200 comprises a processor 202, a cache memory 204, and one or more software applications and drivers directed to the functions described herein. Additionally, hardware system 200 includes a high performance input/output (I/O) bus 206 and a standard I/O bus 208. A host bridge 210 couples processor 202 to high performance I/O bus 206, whereas I/O bus bridge 212 couples the two buses 206 and 208 to each other. A system memory 214 and a network/communication interface 216 couple to bus 206. Hardware system 200 may further include video memory (not shown) and a display device coupled to the video memory. Mass storage 218 and I/O ports 220 couple to bus 208. Hardware system 200 may optionally include a keyboard and pointing device (not shown) coupled to bus 208. Collectively, these elements are intended to represent a broad category of computer hardware systems, including but not limited to general purpose computer systems based on the Pentium® processor manufactured by Intel Corporation of Santa Clara, Calif., as well as any other suitable processor.

[0019] The elements of hardware system 200 are described in greater detail below. In particular, network interface 216 provides communication between hardware system 200 and any of a wide range of networks, such as an Ethernet (e.g., IEEE 802.3) network, etc. Mass storage 218 provides permanent storage for the data and programming instructions to perform the above described functions implemented in the system controller, whereas system memory 214 (e.g., DRAM) provides temporary storage for the data and programming instructions when executed by processor 202. I/O ports 220 are one or more

serial and/or parallel communication ports that provide communication between additional peripheral devices, which may be coupled to hardware system 200.

[0020] Hardware system 200 may include a variety of system architectures; and various components of hardware system 200 may be rearranged. For example, cache 204 may be on-chip with processor 202. Alternatively, cache 204 and processor 202 may be packed together as a "processor module," with processor 202 being referred to as the "processor core." Furthermore, certain implementations of the present invention may not require nor include all of the above components. For example, the peripheral devices shown coupled to standard I/O bus 208 may couple to high performance I/O bus 206. In addition, in some implementations only a single bus may exist with the components of hardware system 200 being coupled to the single bus. Furthermore, hardware system 200 may include additional components, such as additional processors, storage devices, or memories.

[0021] As discussed above, in one embodiment, the operations of the authentication server 20 described herein are implemented as a series of software routines run by hardware system 200. These software routines comprise a plurality or series of instructions to be executed by a processor in a hardware system, such as processor 202. Initially, the series of instructions are stored on a storage device, such as mass storage 218. However, the series of instructions can be stored on any suitable storage medium, such as a diskette, CD-ROM, ROM, etc. Furthermore, the series of instructions need not be stored locally, and could be received from a remote storage device, such as a server on a network, via network/communication interface 216. The instructions are copied from the storage device, such as mass storage 218, into memory 214 and then accessed and executed by processor 202.

[0022] An operating system manages and controls the operation of hardware system 200, including the input and output of data to and from software applications (not shown). The operating system provides an interface between the software applications being executed on the system and the hardware components of the system. According to one embodiment of the present invention, the operating system is the Windows® 95/98/NT/XP operating

system, available from Microsoft Corporation of Redmond, Wash. However, the present invention may be used with other suitable operating systems, such as the Apple Macintosh Operating System, available from Apple Computer Inc. of Cupertino, Calif., UNIX operating systems, LINUX operating systems, and the like.

B.3. Wireless Client

[0023] Figure 4 illustrates for didactic purposes a hardware system 400, which may be used to implement a wireless client 60 of Figure 1A. In one embodiment, hardware system 400 includes a processor 402 and a cache memory 404 coupled to each other as shown. Additionally, hardware system 400 includes a high performance input/output (I/O) bus 406 and a standard I/O bus 408. A host bridge 410 couples processor 402 to high performance I/O bus 406, whereas an I/O bus bridge 412 couples the two buses 406 and 408 to each other. A wireless network interface 424, a system memory 414, and a video memory 416 couple to bus 406. In turn, a display device 418 couples to video memory 416. A mass storage 420, a keyboard and pointing device 422, and I/O ports 426 couple to bus 408. Collectively, these elements are intended to represent a broad category of computer hardware systems, including but not limited to general purpose computer systems based on the Pentium® processor manufactured by Intel Corporation of Santa Clara, Calif., as well as any other suitable processor.

[0024] The elements of hardware system 400 are described in greater detail below. In particular, wireless network interface 424 provides communication between hardware system 400 and any of a wide range of wireless networks, such as a WLAN (i.e., IEEE 802.11), WiMax (i.e., IEEE 802.16), Cellular (e.g., GSMA), etc. Mass storage 420 provides permanent storage for the data and programming instructions to perform the above described functions implemented in the system controller, whereas system memory 414 (e.g., DRAM) is used to provide temporary storage for the data and programming instructions when executed by processor 402. I/O ports 426 are one or more

serial and/or parallel communication ports that provide communication between additional peripheral devices, which may couple to hardware system 400.

[0025] Hardware system 400 may include a variety of system architectures; and various components of hardware system 400 may be rearranged. For example, cache 404 may be on-chip with processor 402. Alternatively, cache 404 and processor 402 may be packed together as a "processor module," with processor 402 being referred to as the "processor core." Furthermore, certain implementations of the present invention may not require nor include all of the above components. For example, the peripheral devices shown coupled to standard I/O bus 408 may couple to high performance I/O bus 406. In addition, in some implementations only a single bus may exist, with the components of hardware system 400 being coupled to the single bus. Furthermore, hardware system 400 may include additional components, such as additional processors, storage devices, or memories.

[0026] In one embodiment, the operations of wireless client-side functionality are implemented as a series of software routines run by hardware system 400. These software routines, which can be embodied in a wireless network interface client utility application and/or network interface driver, comprise a plurality or series of instructions to be executed by a processor in a hardware system, such as processor 402. Initially, the series of instructions are stored on a storage device, such as mass storage 420. However, the series of instructions can be stored on any suitable storage medium, such as a diskette, CD-ROM, ROM, etc. Furthermore, the series of instructions need not be stored locally, and could be received from a remote storage device, such as a server on a network, via network/communication interface 424. The instructions are copied from the storage device, such as mass storage 420, into memory 414 and then accessed and executed by processor 402. In alternate embodiments, one or more aspects of the client side functions discussed herein can be embodied in hardware or firmware.

[0027] While Figure 4 illustrates, for didactic purposes, the hardware architecture of a wireless client according to one implementation of the present invention, the present invention, however, may be implemented on a wide

variety of computer system architectures, such as special purpose, hand-held or portable devices, Personal Digital Assistants (e.g., converged devices which support WLAN data+voice and cellular), Laptop computers, and the like. An operating system manages and controls the operation of hardware system 400, including the input and output of data to and from software applications (not shown). The operating system provides an interface, such as a graphical user interface (GUI), between the user and the software applications being executed on the system. According to one embodiment of the present invention, the operating system is the Windows® 95/98/NT/XP operating system and/or Windows® CE (WinCE) operating system, available from Microsoft Corporation of Redmond, Wash. However, the present invention may be used with other suitable operating systems, such as the Apple Macintosh Operating System, available from Apple Computer Inc. of Cupertino, Calif., UNIX operating systems, LINUX operating systems, Symbian operating systems, and the like.

C. Authentication Method Selection and Negotiation

[0028] The following describes how a wireless client and a wireless network negotiate an authentication method type according to one implementation of the invention. Figure 4 is a flow chart illustrating a process flow, according to one implementation of the present invention, implemented at a wireless client 60. As Figure 4 shows, wireless client 60 initiates a media connection operation, which, in one implementation, may include authentication (502) and association (504) processes with the wireless network infrastructure. In one implementation, the authentication and association processes are open systems authentication processes according to the IEEE 802.11 WLAN specification. Next, wireless client 60 selects an authentication type based on a mapping (506) between a user-selected credential and one or more authentication methods. In one implementation, the authentication type may be an Extensible Authentication Protocol (EAP) type.

[0029] In one embodiment, the mapping provided by the wireless network infrastructure minimizes the knowledge needed by a user to authenticate by limiting the user choices to user credentials and optionally "levels" of security

and performance (versus specific feature types). The reduction in choices reduces the dependence on the user to correctly configure the wireless client and provides more control to the network administrator. In one implementation, the mapping may be preconfigured on the wireless client (e.g., when a user gets a new wireless client or adds a new network interface to the wireless client). More specifically, for a given set of user credentials, the authentication type and order of preference may be preconfigured. As described in more detail below in connection with Figures 5A-5C, the authentication type may be based on several factors such as credential selection and applicable security policies. For example, a user-credential set including username and password could map to one authentication type (e.g., LEAP, PEAP, EAP-MD5, EAP-FAST, etc.) if the Network Admission Control (NAC) is not enabled. Or, the same user-credential set could also map to another authentication type (e.g., EAP-FAST, PEAP-MSCHAPv2, etc.) if the NAC is enabled.

[0030] Where performance versus security may be a tradeoff, the wireless network infrastructure may allow the user to provide performance and/or security choices in addition to providing user credentials. For example, performance choices may include "good," "better," "best, " etc., and security choices may include "open," "legacy," "secure," etc. In one embodiment, the network administrator may disable such choices from the user if the policy requires the fastest performance, where one authentication type (e.g. LEAP) may be the most appropriate for a given set of user credentials. Similarly, in one embodiment, the policy may require the "most secure" authentication type (e.g. EAP-FAST) for a given set of user credentials.

[0031] Accordingly, based on the user credentials, type of network access, and local client policies, only certain authentication types may be allowed or disallowed. For example, on a wireless LAN, an authentication type, referred to as EAP-MD5, would not be allowed, because it does not generate keys and does not meet the wireless network EAP method requirements.

Next, wireless client 60 determines whether an authentication ID request, identifying an EAP type, has been received from authentication server 20 (508). Based on a user credential selection and security tradeoffs, the wireless client

60, as described above, automatically selects the appropriate authentication type suitable for the type of credentials and network access (506). In one implementation, if more than one authentication type is available, wireless client 60 may select one or more of the authentication types and optionally an order of preference. If the selected EAP type matches the EAP type in the authentication ID request, the wireless client transmits an authentication ID assertion response (510). If the EAP type identified in the authentication ID request does not match the selected (or most preferred) EAP type, Wireless client 60 then transmits a negative acknowledgment proposing the selected EAP type to authentication server 20. This EAP type negotiation continues until both ends agree on an EAP type (516). Authentication server 20 then initiates an authentication process according to the authentication type. Next, wireless client 60 determines if an EAP request has been received from authentication server 20 (512), and the wireless client and the authentication server 20 complete the authentication session.

D. Client Authentication Configuration Utility

[0032] Figure 5A is a flow chart illustrating a process flow, according to one implementation of the present invention, implemented by a client configuration application. As Figure 5A shows, the client configuration application receives a policy configuration from the wireless network infrastructure (602). In one implementation, a network administrator determines the policy configuration, which is a policy or set of policies used to determine authentication types (e.g., EAP types) required for a given set of user credentials. In one implementation, the policy configuration includes security policies, which may include policies associated with Network Admission Control (NAC)/Network Admission Protocol (NAP) or Cisco Trusted Security (CTS) or any other security mechanisms. In one implementation, the policy configuration may be preloaded onto wireless client 60. In another implementation, the policy configuration may be stored and periodically updated in a configuration database accessible to the client configuration application. In one implementation, local client policies may be centrally managed by the administrator thru a standard policy management

mechanisms, such as Group Policy Objects. Among the policy items, authentication types, which allow for a particular type of credentials, as well as an order of preferences, may be included. In one implementation, any suitable network management system or tool may be used to propagate policies or default profiles to wireless clients. This also allows the wireless network infrastructure to migrate to a newer authentication type over time with no wireless client-side configuration.

[0033] Next, the client configuration application receives a user credential selection from a user (604). As described above, the user credentials may include name and password, one time password, token, certificate, etc. In one implementation, additional selected information such as a trusted anchor for the authentication server or a means for the user to aid the application in choosing the trusted anchor may also be selected. In one implementation, a trusted anchor may be a data store containing information allowing for validation of credentials. In one implementation, a trusted anchor may be a certificate authority.

[0034] Note that the client configuration application may receive the policy configuration and credential selection in any order. For example, the client configuration application may receive the credential selection before receiving the policy configuration, as described above. Conversely, the client configuration application may receive the credential selection after receiving the policy configuration. In addition, as discussed above, the policy configuration may be pre-loaded on wireless client 60.

[0035] Next, the client configuration application identifies a profile (606), which may be based on the device type, network type (e.g., service set identifier (SSID)), network identity, etc. In one implementation, a profile is a set of parameters used to configure the hardware and software of the network adapter for operation on a particular network. The parameters may include, but are not limited to, radio band selections, data rate selections, proprietary extension selections, security method selections, user identity information, authentication method selections, and network identification information.

[0036] Next, the client configuration application identifies a policy (608), which may be based on the credential selection and the identified profile, etc. Next, the client configuration application may receive the credential selection before receiving the policy configuration, may select one or more authentication types, and optionally may select the order of preference of authentication types (610). In one implementation, the order of preference may be based on the policy configuration and identified profile. Accordingly, such implementations enable wireless clients to better support servers having different authentication types.

[0037] Next, the client configuration application makes the authentication type and order accessible to the network interface driver of the wireless client (612). In one implementation, the authentication type and order may be stored in a configuration file or in a database accessible to the network interface driver.

[0038] In one implementation, once the network interface driver has access to the authentication type, the network interface driver may then negotiate with the authentication server standard authentication/EAP method procedures, as discussed above, to determine an authentication type that both the wireless client and the authentication server will use for a connection.

[0039] Note that the client configuration application may receive a credential selection, identify a profile, and identify a policy in any order. For example, while Figure 5A above illustrates one implementation where the client configuration application first receives a credential selection, then identifies a profile first, and then identifies a policy, Figure 5B below shows one implementation where the client configuration application first receives a credential selection, then identifies a policy, and then identifies a profile. Figure 5C below shows one implementation where the client configuration application first identifies a profile, then receives a credential selection, and then identifies a policy. In Figures 5A-5C, or in any other permutation, each step may be based in the previous step(s).

[0040] Figure 5B is a flow chart illustrating a process flow, according to another implementation of the present invention, implemented by a client configuration application. The process flow described in Figure 5B is similar to the process flow described above in Figure 5A except that the client configuration

application first identifies a policy (608). In one implementation, the identification of the policy may be based on the credential selection and the policy configuration. The client configuration application then identifies a profile (606). In one implementation, the identification of the profile may be based on the credential selection and the identified policy.

[0041] Figure 5C is a flow chart illustrating a process flow, according to another implementation of the present invention, implemented by a client configuration application. The process flow of Figure 5C is similar to the process flow described above in Figure 5A except that the client configuration application first identifies a profile (606). In one implementation, the identification of the policy may be a default profile based on the configuration. The client configuration application then receives a credential selection (604). The client configuration application then identifies a policy (608). In one implementation, the identification of the profile may be based on the identified policy and the credential selection.

[0042] The present invention has been explained with reference to specific embodiments. For example, while embodiments of the present invention have been described as operating in connection with IEEE 802.11 networks, the present invention can be used in connection with any suitable wireless network environment. Other embodiments will be evident to those of ordinary skill in the art. It is therefore not intended that the present invention be limited, except as indicated by the appended claims.

CLAIMS

What is claimed is:

1. Logic for facilitating authentication, the logic encoded in one or more media for execution and when executed operable to:

receive a credential type selection;

select one or more authentication types based on the credential type selection and one or more policies; and

make the one or more selected authentication types available to a network profile operated on a network interface.

2. The logic of claim 1 wherein at least one authentication type is an Extensible Authentication Protocol (EAP) type.

3. The logic of claim 1 wherein the one or more policies enable an ordering of selected authentication types.

4. The logic of claim 1 wherein the logic is further operable to:

support multiple authentication types.

5. The logic of claim 1 wherein the logic is further operable to:

authenticate to a remote node based on the one or more selected authentication types.

6. The logic of claim 1 wherein the logic is further operable to:

select a profile; and

authenticate based at least in part on the profile.

7. The logic of claim 1 wherein the network interface is a wireless network interface.

8. The logic of claim 1 wherein the network interface is a wired Ethernet network interface.

9. The logic of claim 1 wherein the one or more policies may be preconfigured on a client or dynamically provisioned.

10. A method for facilitating authentication, the method comprising:

receiving a credential type selection;

selecting one or more authentication types based on the credential type selection and one or more policies; and

making the one or more selected authentication types available to a network interface driver.

11. The method of claim 10 wherein at least one authentication type is an Extensible Authentication Protocol (EAP) type.

12. The method of claim 10 wherein the one or more policies enable an ordering of selected authentication types.

13. The method of claim 10 further comprising:

supporting multiple authentication types.

14. The method of claim 10 further comprising:

authenticating to a remote node based on the one or more selected authentication types.

15. The method of claim 10 further comprising:

selecting a profile; and

authenticating based at least in part on the profile.

16. The method of claim 10 wherein the network interface is a wireless network interface.

17. The method of claim 10 wherein the network interface is a wired Ethernet network interface.

18. The method of claim 10 wherein the one or more policies may be preconfigured on a client or dynamically provisioned.

19. Logic for facilitating authentication, the logic encoded in one or more media for execution and when executed operable to:

receive a credential type selection;

apply a policy for mapping the credential type selection with one or more authentication types;

select and configure one or more authentication types in association with a given profile and one or more policies; and

make the one or more selected authentication types available to a network interface driver.

20. The logic of claim 19 wherein the network application further comprises instructions operable to cause the one or more processors and the apparatus to:

support one or more authentication mechanisms.

21. A method for facilitating authentication, the method comprising:

receiving a credential type selection;

applying a policy for mapping the credential type selection with one or more authentication types;

selecting and configuring one or more authentication types in association with a given profile and one or more policies; and

making the one or more selected authentication types available to a network interface driver.

22. The method of claim 21 further comprising:

supporting one or more authentication mechanisms.

23. A system for facilitating authentication, the system comprising:

a client operative to receive a credential type selection, select one or more authentication types based on the credential type selection and one or more policies, and make the one or more selected authentication types available to a network interface driver; and

an apparatus operative to receive a credential type selection, identify and apply a policy for mapping the credential type selection with one or more authentication types, select and configure one or more authentication types in association with a given profile and one or more policies, and make the one or more selected authentication types available to a network interface driver.

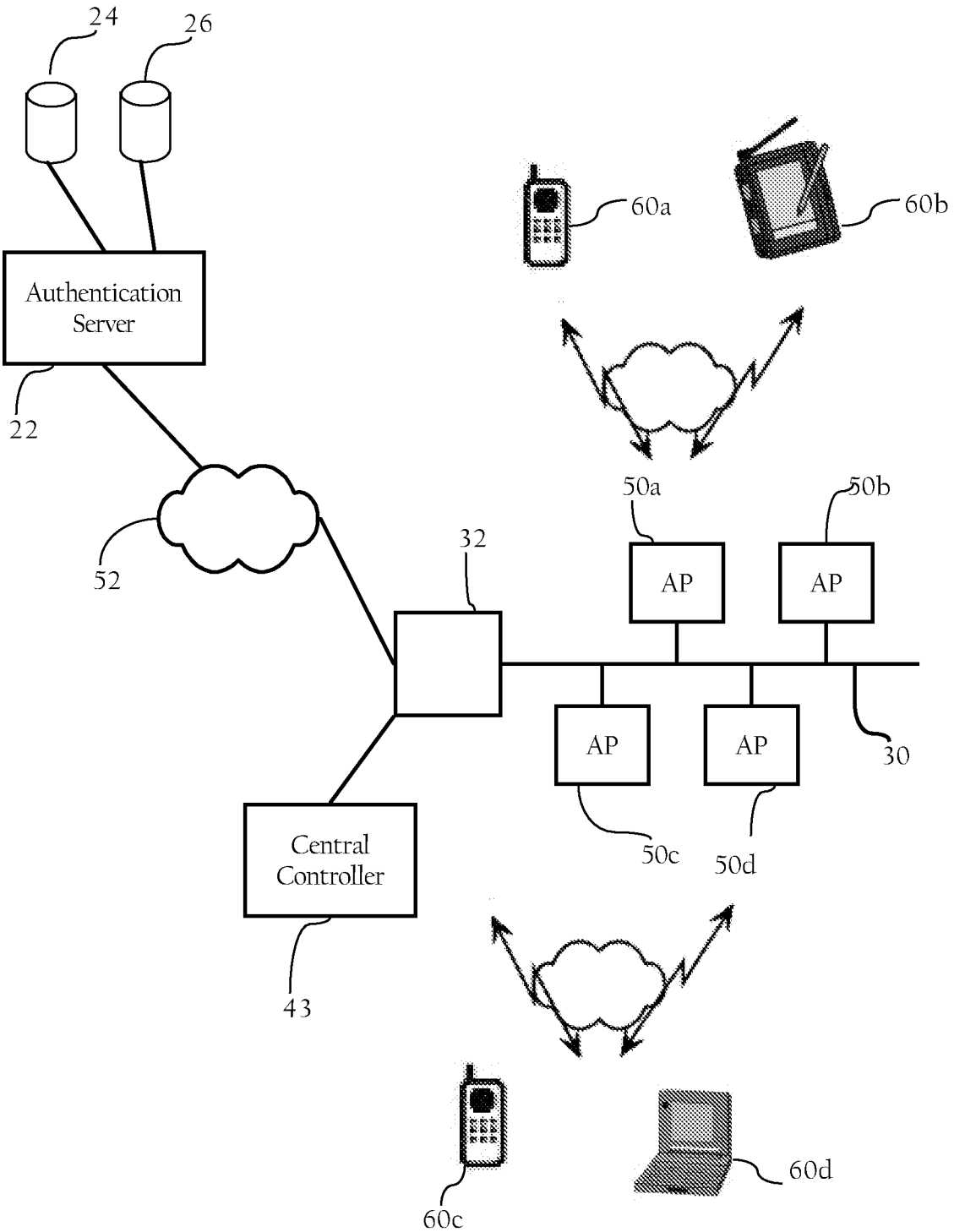

24
26
Authentication Server
22
52
32
60a
60b
50a
50b
AP
AP
AP
AP
30
50c
50d
Central Controller
43
60c
60d

Fig._1A

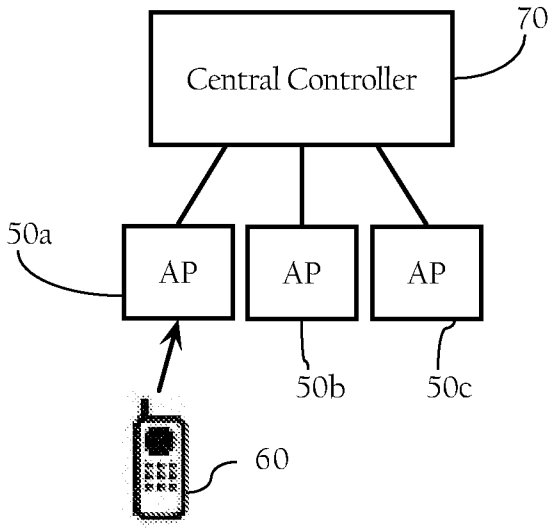

70
Central Controller
50a
AP
AP
AP
50b
50c
60

Fig._1B

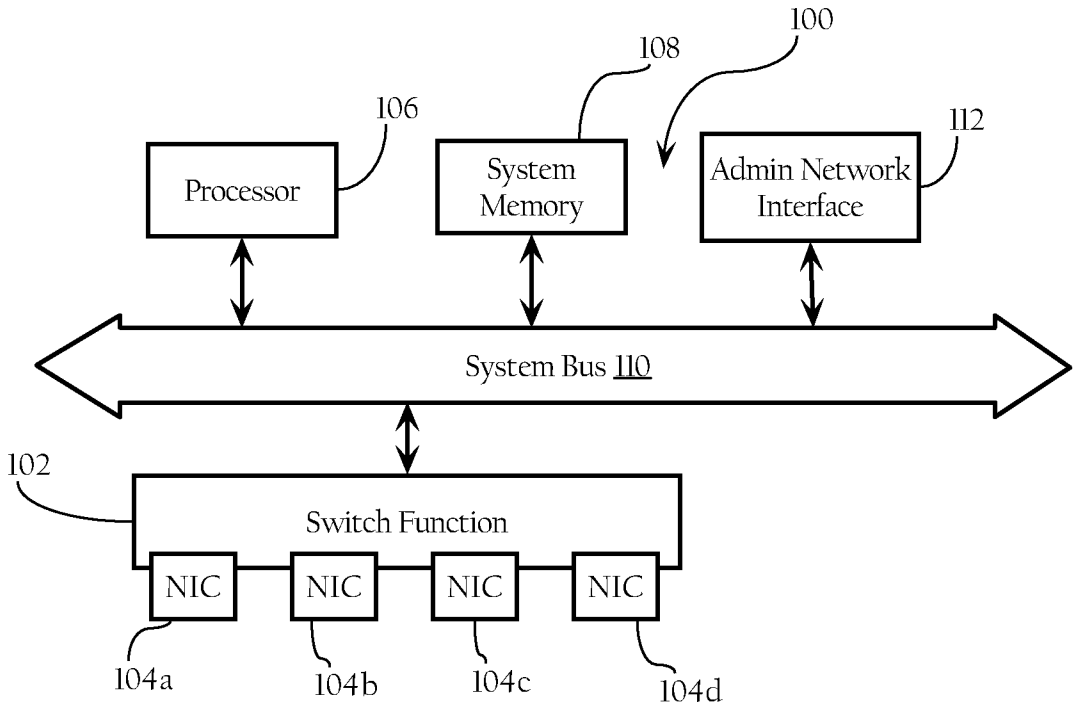

100
108
106
112
Processor
System Memory
Admin Network Interface
System Bus 110
102
Switch Function
NIC
NIC
NIC
NIC
104a
104b
104c
104d

Fig._1C

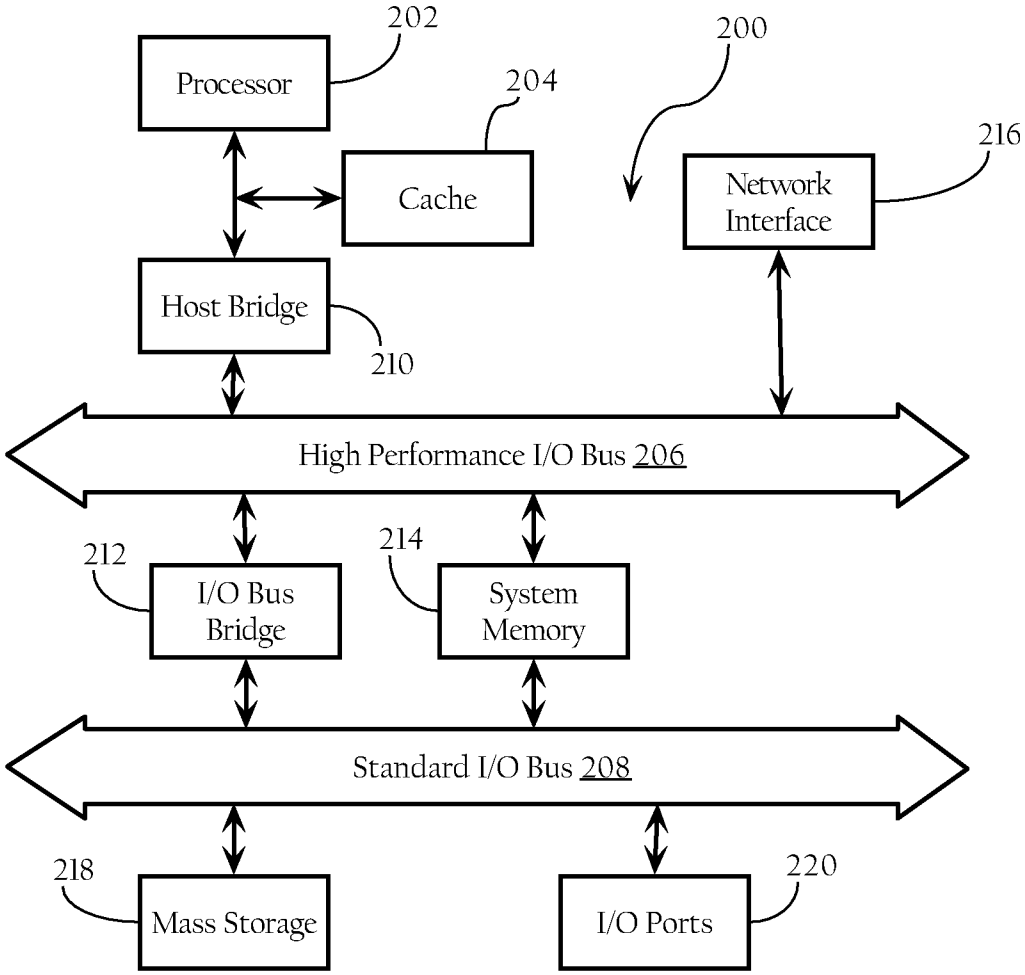

202
Processor
204
Cache
200
216
Network Interface
Host Bridge
210
High Performance I/O Bus 206
212
I/O Bus Bridge
214
System Memory
Standard I/O Bus 208
218
Mass Storage
220
I/O Ports

Fig._2

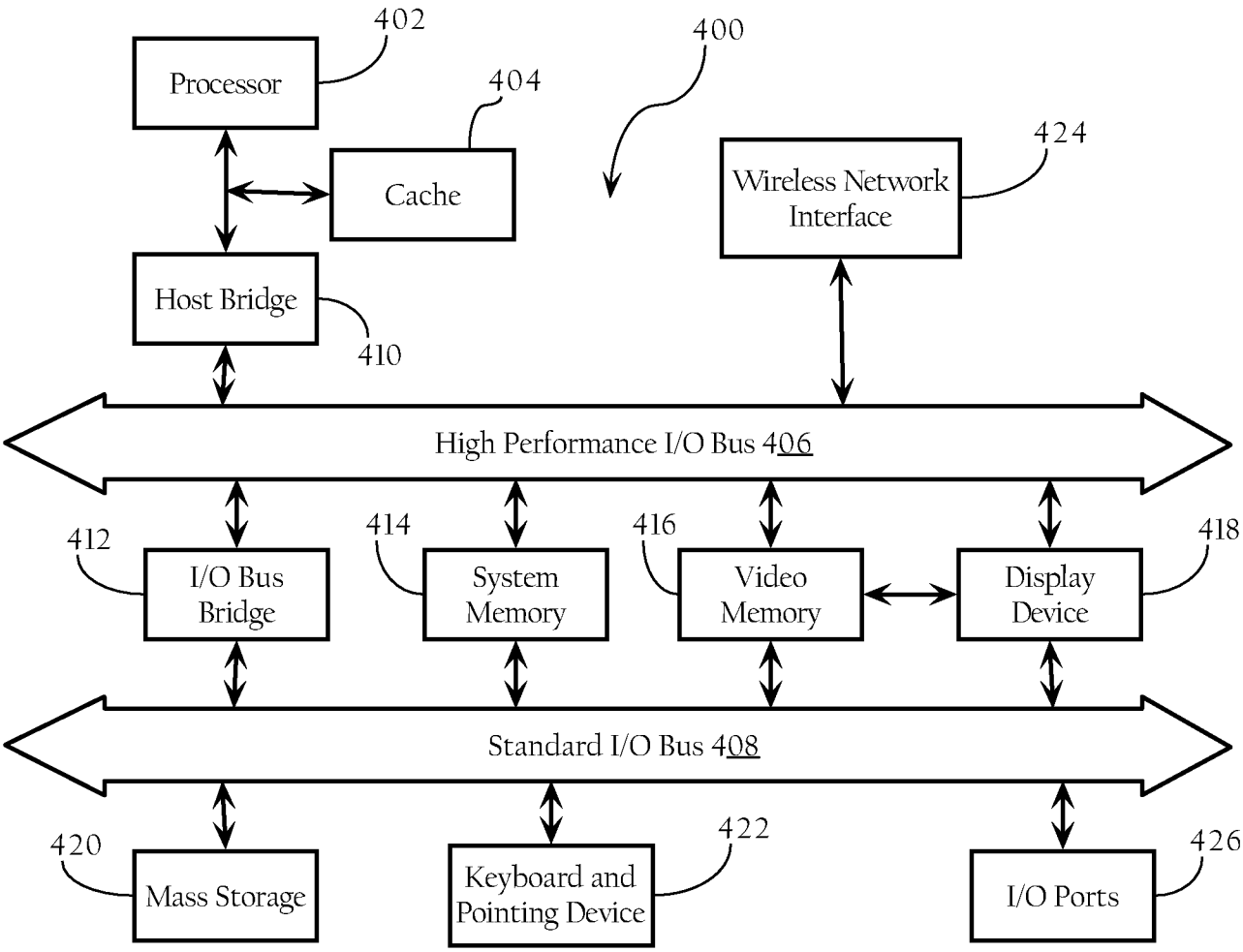

402
Processor
404
Cache
400
424
Wireless Network Interface
Host Bridge
410
High Performance I/O Bus 406
412
I/O Bus Bridge
414
System Memory
416
Video Memory
418
Display Device
Standard I/O Bus 408
420
Mass Storage
422
Keyboard and Pointing Device
426
I/O Ports

Fig._3

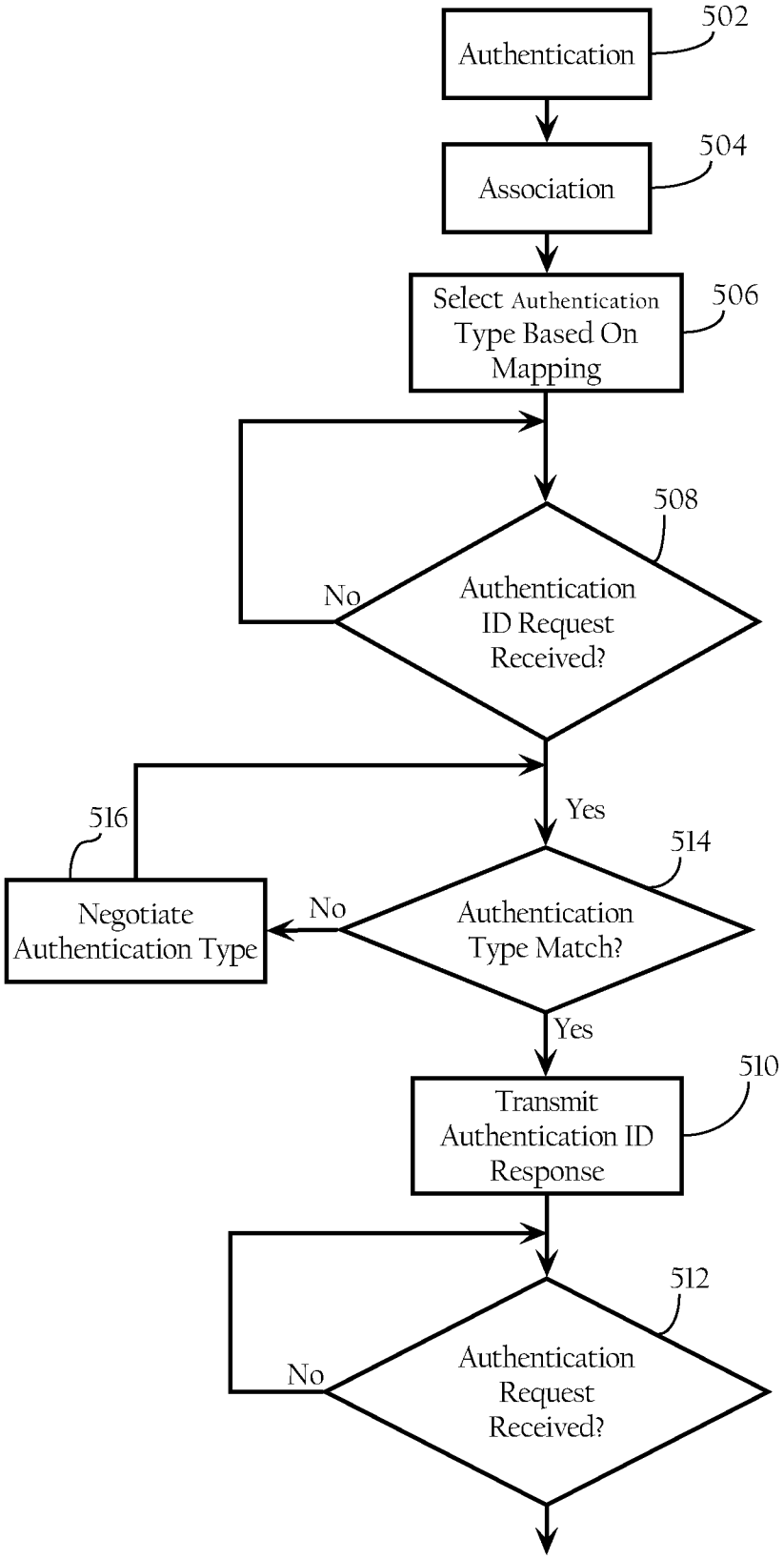

Authentication
502
Association
504
Select Authentication Type Based On Mapping
506
Authentication ID Request Received?
508
No
Yes
Authentication Type Match?
514
No
Negotiate Authentication Type
516
Yes
Transmit Authentication ID Response
510
Authentication Request Received?
512
No

Fig._4

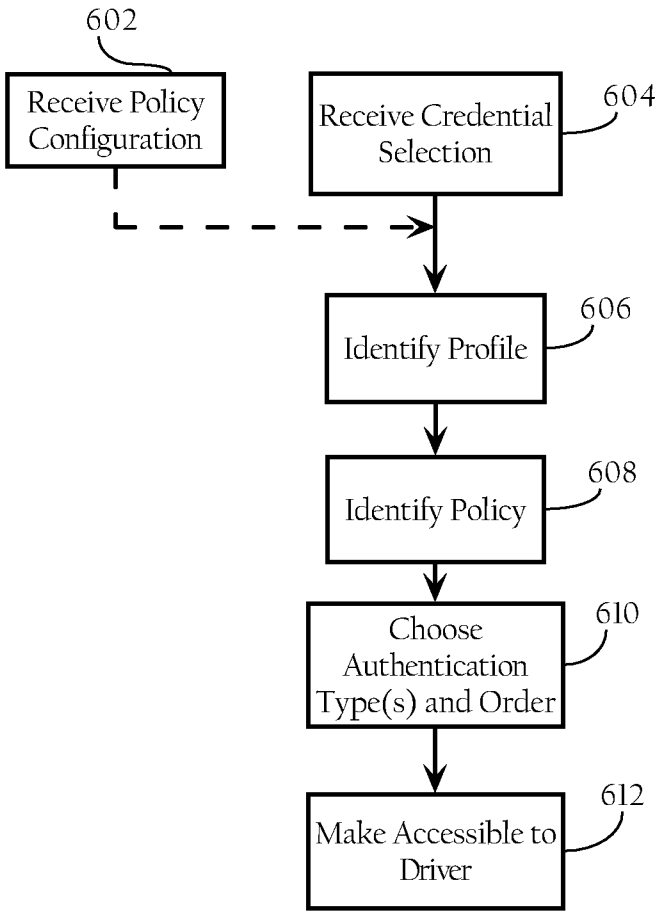

602
Receive Policy Configuration
Receive Credential Selection
604
Identify Profile
606
Identify Policy
608
Choose Authentication Type(s) and Order
610
Make Accessible to Driver
612

Fig._5A

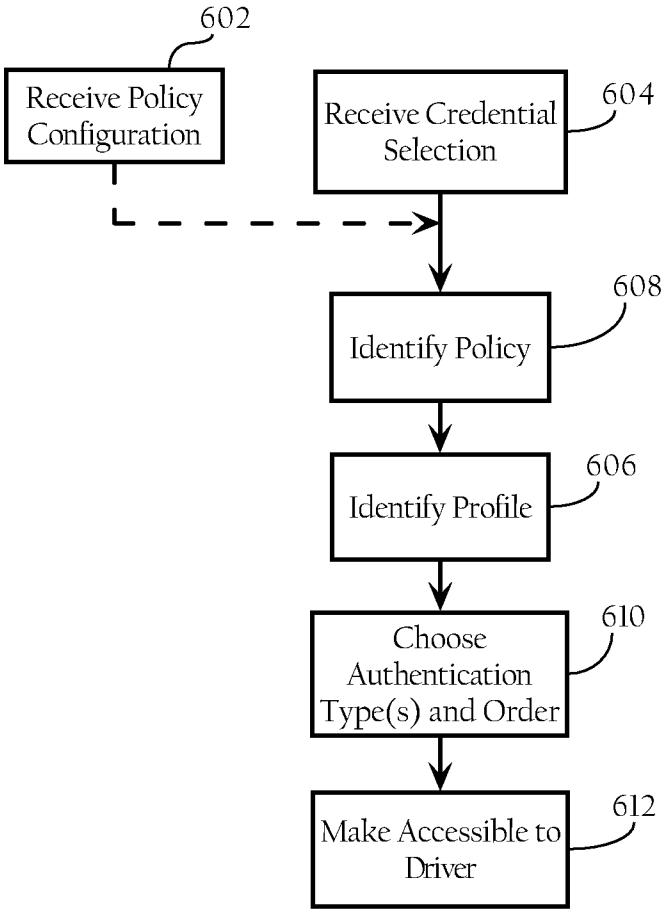

602
Receive Policy Configuration
Receive Credential Selection
604
608
Identify Policy
606
Identify Profile
610
Choose Authentication Type(s) and Order
612
Make Accessible to Driver

Fig._5B

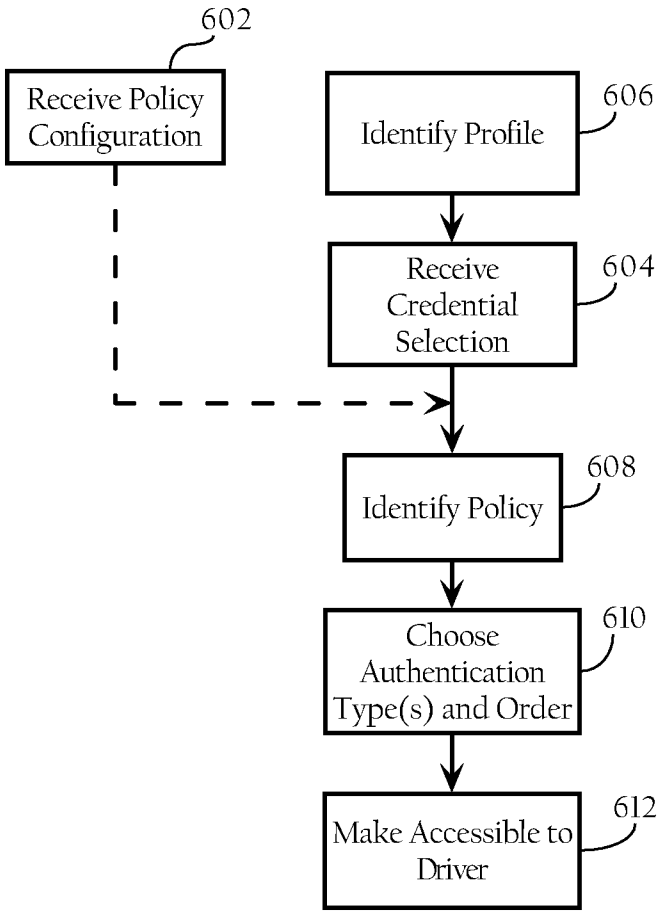

602
Receive Policy Configuration
Identify Profile
606
Receive Credential Selection
604
Identify Policy
608
Choose Authentication Type(s) and Order
610
Make Accessible to Driver
612

Fig._5C