using a key exchange algorithm 319a as depicted in Figure 4b. Initiator 102* can decrypt the ciphertext in message 123 using the second mutually derived symmetric key and a decryption step 320b in Figure 4b. Initiator 102* can derive a third symmetric key using at least the first mutually derived symmetric key and the second mutually derived symmetric key. Initiator 102* can use the third symmetric key to decrypt the encrypted responder authentication value using a decryption step 328 in Figure 4c. Initiator 102* can calculate a value for the responder authentication, and if the calculated value for the responder authentication matches the received value for the responder authentication, then responder 101x can be considered authenticated.

Although not depicted in Figure 1b, after authenticating responder 101x, initiator 102* can send an authentication confirm message, which could comprise a DPP authentication confirm message according to section 6.2.4 of the DPP specification version 1.0. The authentication confirm message can also include an initiator authentication value, which could be encrypted with the third symmetric key. The responder 101x can receive the encrypted authentication confirm value and decrypt the value and also internally calculate the initiator authentication value. If the received initiator authentication value and the calculated authentication confirm values are equal, then the initiator is authenticated with the responder 101x. Note that for a responder authentication 141 in Figure 1b, the authentication of initiator 102* by responder 101x is weak and only comprises verifying that initiator 102* records the bootstrap public key Br 101a for responder 101x. Also, although not depicted in Figure 1b, responder 101x could have the capabilities of an enrollee and send a DPP configuration request message to initiator 102*, which could take the role of a configurator. The DPP configuration request message can be according to section 6.3.2 of the DPP specification version 1.0, and can include an enrollee nonce, which could be a random number generated by the enrollee.

As depicted in Figure 1b, initiator 102* can then send a configuration object in a ciphertext in message 124. Message 124 could comprise a DPP configuration response message according to section 6.3.3 of the DPP specification version 1.0. Initiator 102* can record the configuration object before sending message 124. In the present invention, the configuration object comprises the network credentials 109 depicted in Figure 1a above. In other words, a configuration object can be configuration information for a device 101 with responder 101x to use in order to connect with other devices via a wireless network. Configuration object could contain a list of network identifiers, device identifiers, RF band and channel information, configuration parameters, pre-share keys, PKI keys, names, passwords, group temporal keys, a shared secret 198 for a PKEX key exchange, etc. An exemplary list of possible values for a configuration object is depicted in table 14 of DPP specification version 1.0. In message 124, the network credentials 109 as a configuration object can be sent in a ciphertext using the third mutually derived symmetric key. As depicted in Figure 1b, the ciphertext can include both the enrollee nonce and the network credentials. Responder 101x can

Petitioner Samsung Ex-1004, 1055

receive message 124, decrypt the ciphertext using the third mutually derived symmetric key, and record the network credentials 109. Device 101 can then apply the network credentials 109 and communicate with access point 105.

Figure 1c

Figure 1c is a graphical illustration of a device provisioning protocol for (i) authentication and configuration of a responder and (ii) authentication of an initiator, in accordance with conventional technology. Figure 1c depicts a summary of the WiFi Device Provisioning Protocol (DPP) specification, version 1.0 which was published on April 9, 2018, supporting a mutual authentication 142 by both initiator 102* and responder 101x. The summary depicted in Figure 1c highlights recorded bootstrap PKI keys, derived ephemeral PKI keys, and messages transmitted and received between an initiator 102* and a responder 101x. Many of (i) the PKI keys for initiator 102* and responder 101x, and (ii) the messages transmitted between the nodes are equivalent to those depicted and described in connection with Figure 1b. This description of Figure 1c herein focuses upon the differences from Figure 1b in order for initiator 102* and responder 101x to mutually authenticate.

In order to support a mutual authentication between initiator 102* and responder 101x, initiator 102* can record an initiator bootstrap public key Bi 104a and an initiator bootstrap private key bi 104b. Although bootstrap public key Bi 104a is depicted within initiator 102*, another entity besides initiator 102* could record the public key and make it available to other nodes such as a responder 101x. As contemplated by the DPP Specification, the initiator bootstrap keys can be recorded with initiator 102* before conducting a DPP 191 authentication and configuration with responder 101x. For example, the bootstrap PKI keys for the initiator could be recorded during manufacturing of the initiator, or loaded by an initiator user 102u before conducting the depicted communications with responder 101x. Bootstrap public keys for the initiator can support a selected set of cryptographic parameters 199a, where the selected set of cryptographic parameters 199a can match the parameters used by keys in responder 101x. The DPP specification version 1.0 identifies the minimal set of cryptographic parameters in order to ensure compatibility, which can comprise the "set A" for a set of cryptographic parameters 199 depicted in Figure 2b below.

In order to conduct a mutual authentication 142 in addition to the responder authentication 141 depicted in Figure 1c, a responder 101x should have access to the initiator bootstrap public key Bi 104a and may also record the initiator bootstrap public key Bi 104a. In message 126, a responder 101x can receive the initiator bootstrap public key Bi 104a for initiator 102*. The "out of band" transfer of Bi 104a in message 126 could comprise several different possible methods for

Petitioner Samsung Ex-1004, 1056

responder 101x to receive Bi 104a, which are also described in the DPP specification 1.0, such as via reading a QR code, reading a near-field communications tag, using a Bluetooth wireless connection to read Bi 104a from initiator 102*, and other possibilities exist as well, including using WiFi direct to first establish a "peer-to-peer" WiFi connection. A DPP 191 also contemplates that associated information for responder initiator 102* could be transferred in message 126, such as a device identity for initiator 102*, MAC.I 102n, in addition to Bi 104a. As depicted in Figure 1c, the responder 101x can also record the initiator bootstrap public key Bi 104a, which can be used to authenticate the initiator 102* as described below.

For a mutual authentication 142 between device 102* and responder 101x, the two nodes can exchange the same or equivalent messages as message 122 and message 123, which were also described above in connection with Figure 1b. For mutual authentication, the third shared encryption key ke in DPP specification version 1.0 can be derived by a responder 101x using a key exchange step 321 and the received initiator bootstrap public key Bi 104a as depicted and described in connection with Figure 4c below. The use of public key Bi 104a for the key exchange step 321 is described in section 6.2.3 of the DPP Specification version 1.0. The difference between the third mutually derived symmetric key ke for a mutual authentication 142 and a responder authentication 141 is the inclusion of initiator bootstrap public key Bi 104a in the key exchange algorithm 401c and key derivation function 402c in Figure 4c. The first mutually derived symmetric key can comprise k1 and the second mutually derived symmetric key can comprise k2. The third mutually derived symmetric key ke is used by the responder 101x to encrypt the responder authentication value as depicted in an encryption step 322 in Figure 4c. The responder authentication value can include a random number "initiator nonce" or I-nonce from the initiator 102* in message 122. The responder authentication value can comprise the value "R-auth" in step 3 of section 6.2.3 of DPP specification version 1.0.

Initiator 102* can also mutually derive the third symmetric encryption key ke using the initiator bootstrap private key bi 104b and key exchange step 327 depicted and described below in connection with Figure 4c. Initiator 102* can subsequently decrypt the responder authentication value using the mutually derived third symmetric encryption key ke and a decryption step 328 as depicted and described in connection with Figure 4c below. Initiator 102* can calculate the same value for the responder authentication value as described in section 6.2.4 of DPP specification version 1.0. If the received, decrypted responder authentication value matches the calculated responder authentication value, then the responder 101x is authenticated. The initiator 102* can calculate an initiator authentication value using the received responder nonce in message 123 and encrypt the initiator authentication value using the mutually derived third symmetric key ke (from a key exchange step 327) and send the initiator authentication value in a message (not shown in Figure 1c). The responder 101x can (i) decrypt the encrypted initiator authentication value using the third

Petitioner Samsung Ex-1004, 1057

mutually derived symmetric key ke (from a key exchange step 321) and (ii) also calculate the initiator authentication value. If the calculated initiator authentication value by the responder 101x matches the received, decrypted initiator authentication value then the initiator 102* is authenticated with the responder 101x for a mutual authentication. The responder 101x can then receive a configuration object in a message 124. Message 124 can include network credentials 109 encrypted with the third mutually derived symmetric key ke. The responder 101x can pass the decrypted network credentials 109 to device 101, which can then apply the network credentials 109 in order to communicate with network access point 105.

Figure 1d

Figure 1d is a graphical illustration of hardware, firmware, and software components for an initiator, including an initiator configuration step, in accordance with exemplary embodiments. Figure 1d is illustrated to include several components that can be common within an initiator 102. Initiator 102 may consist of multiple electrical components in order to both (i) communicate with a network 107 and (ii) transmit and receive messages with a responder 101x according to a device provisioning protocol 191. Initiator 102 can undergo an initiator configuration step 127 in order to configure the components in an initiator 102 to function as the enhanced initiator or initiator proxy as contemplated herein, such that an initiator bootstrap private key bi 104b can remain recorded with a network 107 as depicted in Figure 1a above.

In exemplary embodiments and as depicted in Figure 1d, initiator 102 can include an initiator identity 102i, a processor 102c (depicted as "CPU 102c"), random access memory (RAM) 102d, an operating system (OS) 102e, storage memory 102f (depicted as "nonvolatile memory 102f"), a WiFi radio 102w, a system bus 102j, and a wireless wide area network (WAN) radio 102h. Figure 1d depicts initiator 102 as both a default initiator 102 and a configured initiator 102'. The default initiator 102 can comprise the state of initiator 102 before a configuration step 127 and initiator 102' can comprise the state of initiator 102' after the configuration step 127. As depicted, both a default initiator 102 and a configured initiator 102' can contain the same hardware components such as CPU 102c, RAM 102d, etc., where a difference between initiator 102 and 102' can be in both (i) data recorded in storage memory 102f and (ii) the configuration of WiFi radio 102w.

Initiator identity 102i could comprise a preferably unique alpha-numeric or hexadecimal identifier for initiator 102, such as an Ethernet MAC address, an International Mobile Equipment Identity (IMEI), an owner interface identifier in an IPv6 network, a serial number, or other sequence of digits to uniquely identify each of the many different possible units for initiator 102. A system 100 depicted in Figure 1a could operate with a plurality of different initiators 102. Initiator identity

Petitioner Samsung Ex-1004, 1058

102i can preferably be recorded in a non-volatile memory or written directly to hardware in initiator 102 by (i) an initiator manufacturer upon device manufacturing, or (ii) a pre-configuration step for initiator 102, The CPU 102c can comprise a general purpose processor appropriate for typically low power consumption requirements for a initiator 102, and may also function as a microcontroller. CPU 102c can comprise a processor for initiator 102 such as an ARM® based processor or an Intel® based processor such as belonging to the Atom or MIPS family of processors, and other possibilities exist as well. CPU 102c can utilize bus 102j to fetch instructions from RAM 102d and operate on the instruction. CPU 102c can include components such as registers, accumulators, and logic elements to add, subtract, multiply, and divide numerical values and record the results in RAM 102d or storage memory 102f, and also write the values to an external interface such as WiFi radio 102w.

RAM 102d may comprise a random access memory for initiator 102. RAM 102d can be a volatile memory providing rapid read/write memory access to CPU 102c. RAM 102d could be located on a separate integrated circuit in initiator 102 or located within CPU 102c. The RAM 102d can include data recorded in initiator 102' for the operation when collecting and communicating DPP messages 122 through 124 with responder 101x depicted in Figure 1b and Figure 1c above. The system bus 102j may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures including a data bus. System bus 102j connects components within initiator 102 as illustrated in Figure 1d, such as transferring electrical signals between the components illustrated. Initiator 102 can include multiple different versions of bus 102j to connect different components, including a first memory bus 102j between CPU 102c and RAM 102d, and a second system bus 102j between CPU 102c and WiFi radio 102w, which could be an I2C bus, an SPI bus, PCIe bus, or similar data busses.

The operating system (OS) 102e can include Internet protocol stacks such as a User Datagram Protocol (UDP) stack, Transmission Control Protocol (TCP) stack, a domain name system (DNS) stack, a TLS stack, a datagram transport layer security (DTLS) stack, etc. The operating system 101e may include timers and schedulers for managing the access of software to hardware resources within initiator 102, including CPU 102c and WiFi radio 102w. The operating system shown of 102e can be appropriate for a low-power device with more limited memory and CPU resources (compared to a server such as DPP server 103, discovery server 110, etc.). Example operating systems 102e for a initiator 102 includes Linux, Android® from Google®, IOS from Apple®, Windows® 10 IoT Core, or Open AT® from Sierra Wireless®. Additional example operating systems 102e for an initiator 102 include eCos, uC/OS, LiteOs, Contiki, OpenWRT, Raspbian, and other possibilities exist as well without departing from the scope of the present invention. Although depicted as a separate element within initiator 102 in Figure 1d, OS 102e may reside in RAM 102d and/or storage memory 102f during operation of initiator 102.

Petitioner Samsung Ex-1004, 1059

Storage memory 102f (or "nonvolatile memory 102f") within initiator 102 can comprise a nonvolatile memory for storage of data when initiator 102 is powered off. Memory 102f may be a NAND flash memory or a NOR flash memory, and other possibilities exist as well without departing from the scope of the present invention. Memory 102f can record firmware for initiator 102. Memory 102f can record long-term and non-volatile storage of data or files for initiator 102. In an exemplary embodiment, OS 102e is recorded in memory 102f when initiator 102 is powered off, and portions of memory 102f are moved by CPU 102c into RAM 102d when initiator 102 powers on. Memory 102f (i) can be integrated with CPU 102c into a single integrated circuit (potentially as a "system on a chip"), or (ii) operate as a separate integrated circuit or a removable card or device, such as a removable SD card. Memory 101f may also be referred to as "device storage" and can include exemplary file systems of FAT16, FAT 32, NTFS, ext3, ext4, UDF, or similar file systems.

As depicted in Figure 1d, nonvolatile memory 101f for a default initiator 102 may also contain a secret key SK0.initiator 102s, a certificate cert0.initiator 102m, and a certificate cert.CA.root 102r. A secret key SK0.initiator 102s in a nonvolatile memory 102f could comprise the private key for a PKI key pair, where the corresponding public key could be recorded in a certificate cert0.initiator 102m. A secret key SK0.initiator 102s and the corresponding public key in a certificate 102m could comprise values to use with asymmetric public key infrastructure (PKI) cryptography, and could utilize exemplary algorithms based on either Rivest Shamir Adleman (RSA) or elliptic curve cryptography (ECC). For use of ECC algorithms, parameters within certificate 102m (and other certificates herein) can specify elliptic curve names (e.g. NIST P-256, sect283k1, sect283r1, sect409k1, sect409r1, etc.). The same ECC parameters can be used with bootstrap PKI keys in the form of parameters 199a. Further, elliptic curves that do not depend on parameters specified by NIST could be utilized as well, such as Curve22519 or FourQ. For use of RSA algorithms, parameters within certificate 102m can specify a modulus and other associated values for using an RSA PKI key pair. For either asymmetric algorithms RSA or ECC for a PKI key pair herein, parameters in a certificate such as cert0.initiator 102m can specify key lengths, a duration for key validity, uses of the PKI key pair such as for key derivation or signatures, encoding rules, message digest algorithms, etc. Parameters in certificate 102m (and other certificates herein) may also include identifying information associated with the PKI key pair such as a sequence number, a serial number, a certificate revocation list or URL, a domain name of the server associated with the PKI key pair.

Certificate cert.CA.root 102r can comprise a root certificate for initiator 102 to authenticate a server certificate received from DPP server 103, authentication server 111, or discovery server 110, where these servers are depicted and described in connection with Figure 1a, Figure 2a, and Figure 3, and other Figures below. Certificate cert.CA.root 102r can also be used to authenticate other servers or nodes or access points in a network 107 as well. After authenticating a server certificate

Petitioner Samsung Ex-1004, 1060

received with cert.CA.root 102r, initiator 102 can conduct additional authentication steps contemplated by secure protocol standards such as Transport Layer Security (TLS), Datagram Transport Layer Security (DTLS), IPSec, etc. Root certificate cert.CA root 102r can comprise the list or a subset of the list of included root certificates from the Mozilla Foundation with Mozilla projects, where the aggregate list of community approved root certificates and associated parameters is in the widely distributed file certdata.txt from the Mozilla web site.

Initiator 102 and 102' can include a WiFi radio 102w to communicate wirelessly with networks such as AP 105, access network 112, and device 101 depicted and described in Figure 1a above, including via WiFi data link 192. In exemplary embodiments, access network 112 can comprise all available wireless WAN and LAN networks in the range of initiator 102, such as the exemplary networks listed in a networks available list 304c depicted in Figure 3a below. WiFi access point 105 can comprise a member of the set of all access networks 112 for initiator 102. WiFi radio 102w could connect with an antenna in order to transmit and receive radio frequency signals.

As depicted in Figure 1d, a WiFi radio 102w in a configured initiator 102' can record or operate with an initiator configuration 130. The values for initiator configuration 130 could also specify a version of the 802.11 standards to utilize, such as, but not limited to, 802.11n, 802.11ac, 802.11ah, 802.11ax, or related and subsequent versions of these standards. The initiator configuration 130 can be compatible with the configuration of a target WiFi device 101 with responder 101x. The initiator configuration 130 could be obtained during an authentication step for initiator 102, such as depicted below in Figure 2a, although initiator configuration 130 could be obtained at other times as well, such as possibly directly from reading a tag 101c on device 101 or from a DPP server 103.

The initiator configuration 130 can include a device identity ID-token.device 206 (or alternatively ID.device 101i), a MAC address for responder 101x comprising MAC.R 101d, a device operating class 130a which can be selected from the plurality of potential global WiFi operating classes in table E-4 of IEEE standard specification 802.11ac, which is incorporated by reference herein. Recording a device identity 206 and MAC.R 101d with initiator configuration 130 can be used by an initiator 102' to select which set of initiator configuration 130 is associated with which devices 101. In other words, the present invention contemplates that an initiator 102' communicates with multiple responders 101x either concurrently, or over time. MAC.R 101d can be required for initiator 102' to send message DDP Authentication Request 122 as a unicast frame to responder 101x, such that a device 101 could receive message 122 by listening in a unicast mode for messages that include the device's MAC.R 101d. The device operating class 130 could specify if device 101 operates at 2.4 GHz, 5 GHz, etc.

The initiator configuration 130 can also include Device DPP Channel Authentication List 130b, which can comprise a list of channels for initiator 102' to send DPP authentication request 122

Petitioner Samsung Ex-1004, 1061

messages in order to establish initial connectivity with device 101 and responder 101x. In exemplary embodiments, initiator configuration 130 for initiator 102' provides data for an initiator 102' to send data such as message 122 to responder 101x, where responder 101x may operate with different possible values for operating classes, listening channels, or MAC addresses MAC.R 101d used by a responder 101x in device 101. In other words, without initiator configuration 130, an initiator 102' may not normally be able to (i) select the proper WiFi operating channel, and (ii) send unicast data to responder 101x. For embodiments where a Device DPP Channel Authentication List 130b is unknown or not available, then initiator 102' can periodically send DPP 122 messages over time on all available channels for a radio 102w.

WiFi radio 102w can include standard radio components such as RF filters, RF amplifiers, a clock, and phased loop logic (PLL). WiFi radio 102w can support protocols and specifications according to the IEEE 802.11 family of standards. For example, if WiFi radio 102w operates according to 802.11n standards, WiFi radio 102w could operate at a radio frequency of 2.4 or 5 Ghz. In other embodiments where WiFi radio 102w supports 802.11ac standards, then WiFi radio 102w could also support radio frequencies of 0.054 through 0.79 Ghz in order to operate in "white space" spectrum. Other possibilities exist as well for WiFi radio to support wireless LAN standards or 802.11 standards and radio frequencies, without departing from the scope of the present invention. WiFi radio 102w can access a nonvolatile memory 102f in order to record the depicted configuration values of user configuration 131 where the nonvolatile memory could be within WiFi radio 102w, or WiFi radio 102w could access memory 102f via bus 102j in order to record and read the values. User configuration 131 can include credentials for an initiator 102 with initiator user 102u, such as a list of SSIDs, identities, and passwords for operation of the initiator 102 with other networks besides network AP 105. For example, user configuration 131 may include a user operating class 131a, which could correspond to the WiFi global operating class, bands, and channel numbers for initiator 102 to communicate with other WiFi nodes besides device 101, such as a different access point for user 102u than access point 105.

As depicted in Figure 1d, a WiFi configured initiator 102' can include configuration data resulting from a configuration step 127, in order for initiator 102' to conduct a device provisioning protocol 191 with a device 101. A nonvolatile memory 102f for a WiFi configured initiator 102' can include data recorded for a default initiator 102, with the addition of recording a DPP application 102g and configuration parameters 151, and optionally shared secret key 198 for a PKEX key exchange, if bootstrap public keys between configured initiator 102' and device 101 are shared "in band". The contents and source of DPP application 102g and configuration parameters 151 for initiator 102 will be further described in connection with Figure 2a below, where the steps and message flows in Figure 2a below also conclude with initiator 102 completing a configuration step 127 in order to transition to a configured initiator 102'.

Petitioner Samsung Ex-1004, 1062

Initiator 102 may also optionally include user interface 102k which may include one or more devices for receiving inputs and/or one or more devices for conveying outputs. User interfaces are known in the art and may be simple for many devices such as a few LED lights or and LCD display, and thus user interfaces are not described in detail here. User interface 102k could comprise a touch screen if initiator 102 has sophisticated interaction with user 102u. Initiator 102 can optionally omit a user interface 102k, if no user input or display is required for conducting a device provisioning protocol 191 between initiator 102' and device 101 with responder 101x. Although not depicted in Figure 1d, initiator 102 can include other components to support operation, such as a clock, power source or connection, antennas, etc. Other possibilities exist as well without departing from the scope of the present invention.

In exemplary embodiments, initiator 102 can also include a wide area network radio 102h. WAN radio 102h could support 4G LTE, 5G, and subsequent or related standards for use of devices with licensed radio spectrum operating potentially as public land mobile networks (PLMN). WAN radio 102h could also support wide area narrow-band technologies such as NB-IoT or LPWAN, and other examples exist as well. WAN radio 101h can provide connectivity through an access network 112 in order for initiator 102 to communicate with network 107, DPP server 103, and discovery server 110 as depicted and described in connection with Figure 1a. In other words, in the present invention, WAN radio 102h can provide a radio connection to an access network 112, which provides connectivity to an IP network 113 in order for initiator 102 to communicate with remote servers in order to conduct a device provisioning protocol as contemplated herein. WAN radio 102h can include a MAC address MAC.N 102q, although the use of a MAC address for WAN radio 102h is not required. Further, WAN radio 102h can be omitted or not utilized by an initiator 102', in embodiments where initiator 102' can communicate with network 107 via network AP 105. For these embodiments, WAN radio 102h or WiFi radio 102w can also record or utilize a set of network credentials 109 in order to obtain access to network 107 while conducting a device provisioning protocol. Other possibilities exist as well for a configured initiator 102' to have access to network 107 while conducting a device provisioning protocol 191 without departing from the scope of the present invention.

Although not depicted in Figure 1d, the various servers shown above in Figure 1a such as DPP server 103, authentication server 111, and discovery server 110 and other servers as well can include equivalent internal electrical components depicted for an initiator 102 in order to operate as servers. The servers in Figure 1a could include a processor similar to CPU 102c, with primary differences for the processor in a server being increased speed, increased memory cache, an increased number and size of registers, with the use of a 64 bits for datapath widths, integer sizes, and memory address widths, etc., In contrast, an exemplary 32 bit datapaths could be used for CPU 102c in initiator 102 (although CPU 102c could also use 64 bit wide datapath widths if initiator 102

Petitioner Samsung Ex-1004, 1063

comprises a mobile phone such as a smart phone). For embodiments where initiator 102 comprises a home WiFi router, then a CPU 102c in initiator 102 could comprise an exemplary 32 bit processor, although other possibilities exist as well. Similarly, RAM in a server could be a RAM similar to RAM 102d in initiator 102, except the RAM in a server could have more memory cells such as supporting exemplary values greater than an exemplary 8 gigabytes, while RAM 102d in initiator 102 could support fewer memory cells such as less than an exemplary 4 gigabtyes. Non-volatile memory for storage in a server in Figure 1a could comprise disks, "solid state drives" (SSDs) or "storage area networks" (SAN) for a server. Instead of a radio 102w, in exemplary embodiments, a server in Figure 1a could use a physical, wired LAN interface such as a high-speed Ethernet or fiber optic connection.

Figure 2a

Figure 2a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by an initiator, in accordance with exemplary embodiments. System 200 can include an initiator 102, device 101, a discovery server 110 and authentication server 111. Initiator 102 can communicate with discovery server 110 and authentication server 111 via IP network 113, where access to IP network 113 can be provided via access network 112 for initiator 102 as depicted and described in Figure 1a and Figure 1d. Initiator 102 can comprise a smart phone such as, but not limited to, a phone based on an Android operating system from Google ® or IOS from Apple ®, and other possibilities exist as well. In addition, a mobile computing device with both a wireless WAN and wireless LAN connectivity could be used for an initiator 102 in system 200, such as a tablet, laptop computer, e-reader, etc. The present invention also contemplates that a fixed station initiator 102 could be utilized instead of a mobile unit in order to conduct the device provisioning protocol 191 and configuration step 106 with device 101. In exemplary embodiments, a fixed station initiator 102 could comprise a WiFi access point operated by a device owner, initiator user 102u, or network 107 in order to configure a plurality of device 101 in nearby physical proximity in order for the plurality of devices 101 to receive a plurality of sets of network access credentials 109.

Initiator 102 can include a battery, radio, and touch screen, as well as a camera 102z. Initiator 102 can connect with access network 112 in order to communicate with discovery server 110 and authentication server 111, as depicted in Figure 1a above. The access network 112 used in a system 200 can be a different access network than access point 105 used by device 101 in Figure 1a, although access network 112 for initiator 102 could also be through access point 105. For example, if device 101 is an access point to be configured, then initiator 102 could use WAN radio 102h in order to connect with network 107 via access network 112. If (i) an access point 105 is already

Petitioner Samsung Ex-1004, 1064

operating with network access credentials 109, and (ii) a device 101 is to be configured to connect with access point 105, then initiator 102 could also use access point 105 using a set of compatible network access credentials 109. In exemplary embodiments, access network 112 can comprise a wireless WAN such as 4G LTE, 5G, etc, which can comprise a primary access network for initiator 102 and backup network for device 101 (where the primary network for device 101 can comprise a wireless LAN through network access point 105).

Device 101 in system 200 can comprise a device 101 depicted in Figure 1a and device 101 can also include a tag 101c. Tag 101c can comprise a recorded image on device 101 or packaging of device 101 and could be a bar code, QR code, or a serial number for device 101. Tag 101c can provide a unique identifier for device 101. In exemplary embodiments, if device 101 is a medical device, then tag 101c can include a Unique Device Identification (UDI). Although a device 101 could include a plurality of different tags 101c or markings for different purposes (such as shipping, inventory, a distributor code, etc.), for the purposes of conducting (i) a device provisioning protocol 191 and (ii) a read step 203 below by an initiator 102, the tag 101c as contemplated herein for device 101 could include distinct markings for an initiator user 102u to identify as the proper code to read for conducting a configuration step 127 of initiator 102. The distinct markings could comprise a pictogram or icon for a mobile phone or a wireless network next to the tag 101c, an outline around the tag 101c in a bright paint or distinguished color, and other possibilities exist as well for a device 101 to distinguish a tag 101c for a initiator user 102u from other markings, codes, serial numbers, etc. that may exist on a device 101.

In another embodiment for Figure 2a, tag 101c could comprise a "near field communications" NFC tag such as a tag compatible with the NFC Forum standards for type 1 through type 5 tags (and subsequent or related standards). The NFC technology could also be NFC-A, NFC-B, or NFC-V, or subsequent standards. For these exemplary embodiments where tag 101c comprises an NFC tag, then initiator 102 could use an NFC reader instead of camera 102z, and the NFC reader could operate a NFC radio. Initiator 102 could utilize an NFC chip and antenna, where the NFC chip operates in read mode with tag 101c for device 101 as an NFC tag. In exemplary embodiments, discovery server 110 can include or be associated with a discover server database 110a. Database 110a could record information about a plurality of devices 101, such as ID.device 101i, and ID-token.device 206, and a dataset config-provisioning.ID.device 212, which will be described in further detail below for Figure 2a.

At step 201 in Figure 2a, device 101 can be in a powered off state. For a step 201 device 101 could be shipped from a manufacturer to a distributor and then to a device user or an initiator user 102u. A device 101 could also be taken to the physical proximity (e.g. WiFi RF range) of network access point 105 in a step 201. In some embodiments, device 101 could be powered on at step 201, although power for device 101 is not required for embodiments with the steps and message flows

Petitioner Samsung Ex-1004, 1065

outlined in Figure 2a. At step 202, initiator 102 could be powered on and placed in proximity of device 101. Initiator 102 could be operated by an initiator user 102u at step 202. At step 203, initiator user 102u could use initiator 102 to conduct a "read" step 203, where initiator 102 can (i) take a picture of a tag 101c for device 101, or (ii) read a bar code, QR code, printed serial number, or similar code or markings on device 101. As mentioned above, tag 101c could comprise a bar code, QR code, or other encoding of data for device 101, including a serial number. For the alternative embodiment where (i) initiator 102 uses an NFC reader and (i) tag 101c comprises an NFC tag, the initiator 102 could conduct an NFC read of tag 101c for step 203 and step 204.

At step 204, initiator 102 can receive data from tag 101c, such as a response, where the response can consist of a uniform resource locator (URL) for discovery server 110, URL-DS 205, an ID-token.device 206, and a tag value 101w. The response at step 204 could be recorded in the tag 101c. URL-DS 205 could include a domain name for discovery server 110 and parameters such as the use of HTTP or HTTPS and a destination TCP or UDP port number for sending packets to discovery server 110. ID-token.device 206 can be an identifier for device 101 and/or responder 101x and could be a token for ID.device 101i, or a value or unique pseudo-random number associated with device 101 and/or ID.device 101i. ID-token.device 206 could be a value that obfuscates ID.device 101i but can preferably uniquely map to ID.device 101i in a system 100 or 200 with a plurality of devices 101. Through the use of ID-token.device 206 that is readable by initiator 102, then a normal operating value for ID.device 101i with a network 107 can remain confidential at step 204. Or, in some exemplary embodiments an ID.device 101i can be the same as ID-token.device 206.

Tag value 101w in response 204 can include data associated with tag 101c and device 101, such as versions of software supported, product validity or expiration dates, a responder bootstrap public key Br 101a, cryptographic parameters 199 supported (e.g. RSA or ECC), a curve name utilized for an ECC cryptographic algorithm, a name or URL for Certificate Authority 107 used by manufactured device 101, a certificate for device cert0.device 101m (in Figure 1d), or a subset of the values in config-provisioning.ID.device 212. In an exemplary embodiment, tag value 101w could comprise some values of config-provisioning.ID.device 212 upon manufacturing of device 101, which could be subsequently updated at a later date in discovery server DB 110a before installation or operation of device 101. For exemplary embodiments where tag 101c comprises a bar code or QR code or similar markings, then tag value 101w can be limited to fewer bytes such as listing URL-DS 205 and ID-token.device 206. For exemplary embodiments where tag 101c comprises an NFC tag capable of storing an exemplary few thousand bytes, then tag value 101w could include more values and data listed in the first sentence of this paragraph.

In other exemplary embodiments where a lower level of security can be required in order to have simplified functionality, tag value 101w can comprise MAC.R 101d, responder bootstrap

Petitioner Samsung Ex-1004, 1066

public key Br 101a, a parameter set 199a, configuration 130, and/or a version number 212a. For these simplified embodiments, initiator 102 can (i) read tag value 101w, load a DPP application 102g compatible with version number 212a and parameter set 199a, (iii) optionally send Br 101a to a DSS server 103, and (iii) load configuration 130 for WiFi radio 102w, and thereby skip the remaining steps and message flows depicted in Figure 2a. In other exemplary embodiments where a higher level of security is desired and a system 100 or 200 could comprise millions of devices, or high-value devices, and thousands or more initiators 102, then the full set of steps and message flows depicted in Figure 2a may be preferred.

At step 207, initiator 102 could process the data received in response 204. For embodiments where tag 101c is an image such as a bar code or QR code, at step 207 initiator 102 could process the image and extract the data or values for ID-token.device 206, URL-DS 205, and tag value 101w. Continuing at step 207, initiator 102 could connect with an access network 112 in order to call the URL in URL-DS 205, which could be an HTTPS/TLS session in exemplary embodiments. In exemplary embodiments, URL-DS 205 is combined with ID-token.device 206, such that the HTTPS session is a unique request to discovery server 110 for each different device 101. In exemplary embodiments, the value ID-token.device 206 can have enough information entropy or pseudo randomness that it cannot be easily guessed, such as (i) appearing to be an exemplary 32 byte random number, and (ii) not related to a value ID-token.device 206 for another device 101 in any externally observable manner such as in sequence, or sharing similar common factors, etc. By combining a URL-DS 205 with a unique and a sufficiently randomized ID-token.device 206, only an initiator 102 physically in the presence of device 101 (or by reading tag 101c) could feasibly query discovery server 110 with a fully and properly formed query/URL to fetch the subsequent data from discovery server 110. At step 207, initiator 102 could perform additional checks and steps in preparation for communication with device 101, such as verifying data in tag value 101w is compatible with software or settings in initiator 102, including DPP application 102g (if DPP application 102g is already installed in initiator 102).

In message flow 208, initiator 102 can establish a secure HTTPS session with discovery server 110 using the URL-DS 205, and the secure session could comprise a transport layer security (TLS) session such as TLS version 1.2, TLS version 1.3, or similar and subsequent standards. As part of the setup of message flow 208, initiator 102 can receive a certificate cert.DS 110c from discovery server 110. Cert.DS 110c could comprise an X.509 certificate and include a signed public key for discovery server 110. In exemplary embodiments, initiator 102 verifies the certificate cert.DS 110c and parent certificates through root certificate cert.CA.root 102r. The verification of signatures using public keys in certificates can use the signature verification steps as specified in the US National Institute of Standards and Technology (NIST) FIPS 186-4 titled "Digital Signature Standard" and subsequent or related versions. Note that a secure session 208 is not required for some

Petitioner Samsung Ex-1004, 1067

exemplary embodiments, and confidentiality and/or data integrity could be acquired via other means besides HTTPS and/or TLS or DTLS, including embodiments were data flow from discovery server 110 to initiator 102 be signed. Other possibilities exist as well for a message flow 208 to be adequately secured.

In exemplary embodiments initiator 102 can also authenticate with discovery server 110, such as an initiator user 102u associated with initiator 102 entering user credentials in a web site for discovery server 110. Or initiator 102 could send discovery server 110 a certificate for initiator 102 cert.initiator 102m, which the discovery server 110 could verify using a signature verification step. As contemplated herein, a message flow 208 and all subsequent message flows depicted and described can comprise a series of parts or sub-messages, where the collection of sub-messages or parts can comprise the message. In other words, a message depicted with multiple elements for sets of data within the message does not require that all data be transferred together or in a single flow of TCP or UDP data. Transferring the message could comprise sending exemplary data within the message in parts, where upon conclusion of the message, (i) exemplary data depicted and described could be transferred in parts or portions and (ii) the collection of sending different parts or portions for the message could comprise sending the message.

In message 209 and after the secure HTTPS/TLS session setup in message flow 208, initiator 102 can send discovery server 110 ID-token.device 206 and data from tag value 101w. Or, the value for ID-token.device 206 could be included in the URL called by initiator 102 when accessing discovery server 110, and in this manner a value ID-token.device 206 does not need to be sent again in a separate message 209. Initiator 102 could alternatively send discovery server 110 a subset of the data for ID-token.device 206 and/or tag value 101w.

At step 210, discovery server 110 can use data in ID-token.device 206 to query discovery server database 110a for the values or data sets ID.device 101i and config-provisioning.ID.device 212. Discovery server database 110a may comprise a database with tables to track configuration information for a plurality of devices 101. A device manufacturer or device owner could send initial data for a device 101 to discovery server 110 and discovery server database 110a. In exemplary embodiments, config-provisioning.ID.device 212 contains data used by initiator 102 for the configuration of device 101, including conducting a device provisioning protocol 191. Exemplary data depicted for config-provisioning.ID.device 212 in Figure 2a includes DPP application 102g, configuration parameters 151, version number 212a, ID.Authentication Server 111a, URL-authentication Server 111b, cert.AS 111c, auth. parameters 111d. Additional or other data for initiator 102 and device 101 could be included in discovery server database 110a and values for config-provisioning.ID.device 212. The data for config-provisioning.ID.device 212 can comprise a set of values or a table of data for initiator 102 and/or a DPP application 102g to use in order to

Petitioner Samsung Ex-1004, 1068

conduct a configuration step 106 with device 101, where a specific device 101 can be identified by either ID-token.device 206 or ID.device 101i.

DPP application 102g in a config-provisioning.ID.device 212 can specify the name or URL of a software application for initiator 102 to utilize (e.g. a field with less than a few hundred bytes), instead of comprising the full software package for DPP application 102g in exemplary embodiments (where the full DPP application 102g could comprise several megabytes or more). For these exemplary embodiments, initiator 102 could utilize the name or URL to fetch the software for DPP application 102g. DPP application 102g could be a public or private application for the operating system 102e of initiator 102. For embodiments where initiator 102 utilizes an Android-based operating system, DPP application 102g could be downloaded from the Google Play store or a private web site associated with network 107. For embodiments where initiator 102 utilizes an Apple ® / IOS –based operating system, DPP application 102g could be distributed through a developer enterprise program or downloaded through iTunes ®. Other possibilities exist as well without departing from the scope of the present invention for an initiator 102 to download a DPP application 102g, where the name of DPP application 102g is received from a discovery server 110, after initiator 102 sends a message 209 to a discovery server 110.

As depicted in Figure 2a, values for config-provisioning.ID.device 212 sent to initiator 102 may also include configuration parameters 151 and version number 212a. Configuration parameters 151 may include values and data for initiator 102 to utilize with configuration application 102g and communicate with device 101 using a WiFi radio 102w. Exemplary uses of parameters 151 with initiator 102 conducting a configuration step 106 can comprise (i) cryptographic algorithms to utilize, (ii) ECC curve names and key lengths and associated data, (iii) selection and parameters for a WiFi radio 102w in initiator 102 to communicate with device 101, including the selection of one of multiple possible radios in initiator 102, (iv) addresses, names, and port numbers to utilize with IP network 113 including IP version number such as IP4 or IP6, (v) timers and retry counts for communication with device 101 or DPP server 113 or authentication server 111, (vi) parameters for network 107 including a certificate authority, RSA PKI keys, selected cryptographic algorithms, etc, and (vii) parameters for communicating with device 101, etc.

Version number 212a for config-provisioning.ID.device 212 may specify a version number of DPP application 102g to utilize, or a list or dataset of version numbers of standards supported, such as IP version of IP network 113, WiFi radio 101h type such as 802.11ac or 802.11ax, TLS or DLTS version supported by device 101, etc. In exemplary embodiments, not all of the values depicted for config-provisioning.ID.device 212 are sent from a discovery server 110 to an initiator 102, and only a subset of the depicted values are sent instead. In an exemplary embodiment, similar but different to that depicted in Figure 2a, the data within config-provisioning.ID.device 212 can include all the data for a message 220 below, and in these embodiments an initiator 102 could use

Petitioner Samsung Ex-1004, 1069

config-provisioning.ID.device 212 to conduct a configuration step 127 for initiator 102 and then proceed to step 301 in Figure 3a, thereby omitting the subsequent steps depicted in Figure 2a.

As depicted in Figure 2a, config-provisioning.ID.device 212 sent to initiator 102 may also include ID.authentication Server 111a, URL-authentication Server 111b, cert.AS 111c, auth. parameters 111d. ID.Authentication Server 111a could comprise an identity or name for authentication server 111, such as a domain name or multiple domain names for multiple servers. URL-Authentication Server 111b could comprise a URL for configuration application 102g to call in order to authenticate with authentication server 111. In some exemplary embodiments, URL-authentication server 111b includes either ID.Device 101i or ID-token.Device 206, so that when initiator 102 using DPP application 102g calls URL-authentication Server 111b, the URL includes a unique string that identifies device 101. Or, in another exemplary embodiment, initiator 102 can (a) call URL-authentication server 111b without an identity for device 101, and subsequently (b) send either ID.Device 101i or ID-token.Device 206. Cert.AS 111c could comprise a certificate for authentication server 111 where cert.AS 111c could include (i) a public key for authentication server 111 and (ii) a signature of a certificate authority. Authentication parameters 111d can include parameters for DPP application 102g to utilize with authentication server 111, such as, but not limited to, hash or message digest algorithms, digital signature algorithms, padding schemes, key lengths, nonce or challenge lengths, timers, and retry counts, encoding rules, TLS or DTLS versions supported, cipher modes supported (e.g. AES-CBC, AES-CTR), etc.

Message 211 from discovery server 110 to initiator 102 can communicate the above fields and values described for config-provisioning.ID.device 212 and ID.device 101i, as depicted in Figure 2a. In an exemplary embodiment, discovery server 110 can send initiator 102 the full DPP application 102g with the response 211 (e.g. several megabytes or more) instead of a name or URL for DPP application 102g (e.g. less than a few hundred bytes), if the current, updated DPP application 102g is not already downloaded by initiator 102. In another exemplary embodiment, initiator 102 can download DPP application 102g in a step 213 as depicted in Figure 2a, if initiator 102 has not already installed the software for DPP application 102g after receiving message 211. For example, response or message 211 could include a version number 212a that is higher or different than initially utilized by initiator 102 and/or an initial DPP application 102g, and consequently the higher version number 212a could indicate to initiator 102 that a different DPP application 102g would be required in order to communicate with device 101. In exemplary embodiments, DPP application 102g downloaded in a step 213 can include a hash signature that can be verified by initiator 102 before using the software for a DPP 191 with device 101.

Step 213 may comprise initiator 102 loading and launching the DPP application 102g and applying a subset of configuration parameters 151 or tag-value 101w. If any of the steps depicted in Figure 2a fail, error codes or messages could be displayed through initiator 102 to initiator user

Petitioner Samsung Ex-1004, 1070

102u. After successful launch of DPP application 102g, initiator 102 can connect with authentication server 111. The subsequent steps shown performed by an initiator 102 after a step 213 in Figure 2a could be performed by a DPP application 102g using data within config-provisioning.ID.device 212. In exemplary embodiments, DPP application 102g could be included with the operating system 102e for initiator 102, and in this case a separate download of DPP application 102g may not be required.

Secure session setup 214 could comprise a secure HTTPS session with authentication server 111 using the URL-authentication server 111b, and the secure session could comprise a transport layer security (TLS) session such as TLS version 1.2, TLS version 1.3, or similar and subsequent standards. Other networking standards besides HTTPS and TLS could be utilized as well for secure session setup 214, such as an IPSec tunnel. As part of the setup of secure session setup 214, initiator 102 can receive a certificate cert.AS 111c from authentication server 111, if initiator 102 had not already received cert.AS 111c from discovery server 110. Initiator 102 could verify cert.AS 111c after receiving the certificate, such as verifying a signature from a certificate authority and parent certificates of cert.AS 111c as well all the way to recorded cert.CA.root 102r, which is depicted above in Figure 1d. In other words, a secure session setup 214 can mutually authenticate both authentication server 111 and initiator 102, such as using certificates from both sides with TLS. An additional layer of authentication can be performed by an initiator user 102u, although authentication for all of authentication server 111, initiator 102, and initiator user 102u is not required for some embodiments.

In message 215, initiator 102 transmits authentication information to authentication server 111, and authentication server 111 verifies the authentication information in a step 217. The authentication information transmitted in a message 215 could comprise information to authenticate (i) initiator user 102u, (ii) initiator 102, or (iii) both initiator user 102u and initiator 102. The authentication information in message 215 may comprise an initiator user 102u entering identification and credential information in a touch screen of initiator 102, and the information is passed to authentication server 111 (possibly in an encrypted or hashed form). In exemplary embodiments, initiator user 102u provides biometric data such as a fingerprint, an image of the user's face, or similar data, where initiator 102 transmits confirmation or signatures for the data through secure session setup 214 to authentication server 111, and the authentication server 111 verifies the biometric data or signatures of the biometric data is associated to initiator user 102u.

Message 215 may also request the verification of initiator 102 with authentication server 111. In exemplary embodiments, initiator 102 records a certificate Cert0.initiator 102m for initiator 102. The certificate for initiator 102 includes a public key for the initiator 102 and a signature by a certificate authority. In an authentication step 217, authentication server can use the information in message 215 to verify the identity of an initiator user 102u and/or initiator 102. In embodiments

Petitioner Samsung Ex-1004, 1071

where initiator 102 uses a device certificate for initiator 102 (which could be a Cert0.initiator 102m), authentication server 111 could verify that initiator 102 also records the private key SK0.initiator 102s, by (i) sending a challenge/random nonce and (ii) verifying a signature for the challenge/nonce from initiator 102 using the private key 102s for cert0.initiator 102m. Authentication server 111 can verify the signature using the public key for initiator 102 from Cert0.initiator102m. Message 215 and authentication 217 could be conducted using authentication parameters 111d, which could be previously included in Config-provisioning.ID.device 212.

In step 216a of Figure 2a, initiator 102 may also authenticate authentication server 111, using cert.AS 111c. In step 216b, initiator user 102u can enter device owner information into initiator 102, such as an owner name, a DNS name associated with owner, or a code or other value to identify a device owner. Device owner information could be automatically recorded for DPP application 102g or initiator 102 before a configuration step 106 from Figure 1a. For example, initiator 102 with DPP application 102g may operate on a mobile phone, and the owner of device 101 can be associated with or specified by the initiator user 102u. For example, if a home owner with a mobile phone wishes to configure a plurality of devices 101 for his or her home, an owner identity 152 can be selected based on the user of a mobile phone which could be the initiator user 102u. Other possibilities exist as well without departing from the scope of the invention. The value identifying a device owner could be ID.owner 152, although the use of ID.owner 152 could be optionally omitted.

In some embodiments, the value of ID.owner 152 could be recorded in a tag 101c, such as if device 101 comprises leased equipment that is owned by a different entity than an initiator user 102u. Note that a tag 101c could also comprise a collection of different tags, such that a first tag could specify a URL-DS 205 (which could be a label or sticker applied by a manufacturer), and a second tag could specify ID.owner 152 (which could be a label or sticker or mark applied to device 101 by an owner). A step 216b can function to associate device 101 with device owner for a network 107. In exemplary embodiments, millions of devices 101 could be distributed out to millions of device owners. Network 107 may prefer to know which devices 101 are associated with which device owners by using a value of ID.owner 152. In an exemplary embodiment, the value ID.owner 152 could be included in a set of Config-provisioning.ID.device 212 for cases where device owner can register the ID.device 101i with discovery server 110 before initiator 102 sends message 209 to discovery server 110.

In message 218, initiator 102 can send a first set of identities to authentication server 111 through the secure session setup in 214, and also send authentication parameters 111d. Note that in some embodiments the set of authentication parameters 111d can be sent from authentication server 111 to an initiator 102 in a response to message 218. The list of identities in a message 218 can

Petitioner Samsung Ex-1004, 1072

include an ID.device 101i (or ID-token.device 206), ID.initiator 102bi, and ID.owner 152, as depicted in Figure 2a.

In step 219a, authentication server 111 can process the first set of identities received in message 218. A step 219a could comprise authentication server 111 verifying that initiator 102 with initiator user 102u is authorized to configure or provision device 101. In other words, the prior step 217 could be an initial step to verify or authenticate identities, and then the subsequent step 219a could be verifying that the authenticated identities have privileges or authorization to conduct a configuration step 106 or a DPP 191 with device 101. For a step 219a, authentication server 111 may communicate with a device database 104 in order to confirm that initiator 102 with an initiator user 102u is authorized to conduct configuration step 106 and a DPP 191 with device 101 identified by ID.device 101i or ID-token.device 206. Authentication server 111 can utilize any or all of the identities received in message 218 in order to identify and communicate with the correct device database 104 and determine authorization for initiator 102 and/or initiator user 102u.

Although not depicted in Figure 2a, authentication server 111 could utilize a mutually authenticated secure channel such as TLS v1.3 with a device database 104 in a step 219a. In some exemplary embodiments, a step 219a could be optionally omitted, for example if a device owner is also an initiator user 102u, such as a home owner using initiator 102 and installing a device 101 in their home. In these embodiments, an authentication step 217 may be sufficient to allow/authorize initiator 102 to conduct the DPP 191 and configuration step 106. Other possibilities exist as well for an authentication server 111 use a step 219a to authorize that an authenticated initiator 102 and/or initiator user 102u can conduct a DPP 191 and configuration step 106 without departing from the scope of the present invention.

After the authorization step 219a for initiator 102, authentication server 111 can query device database 104 for (i) initiator configuration 130 for device 101 using ID.device 101i or ID-token.device 206, and (ii) a DPP server URL 103b for initiator 102 to utilize in order to conduct a device provisioning protocol 191 and configuration step 106 with device 101, (iii) MAC address for responder 101x of MAC.R 101d, and (iv) optionally, responder public bootstrap keys Bi 104a and Br 101a or a PKEX key 198 for conducting a public key exchange protocol. The optional sending of bootstrap public keys and MAC address for responder 101x can depend on the selected operation of initiator 102 for conducting a DPP 191. For embodiments depicted in Figure 3a below, the bootstrap public keys Bi 104a and Br 101a can remain in DPP server 103, and in these embodiments then the query to device database 104 after step 219a can query for (i) through (iii) in the first sentence of this paragraph. In other words, for the embodiments depicted in Figure 3a and 5b, the public bootstrap keys Bi 104a and Br 101a, may optionally not be queried and sent back to authentication server 111 and initiator 102, but remain in a network 107 for enhanced security purposes.

Petitioner Samsung Ex-1004, 1073

For embodiments depicted in Figure 5a, the public bootstrap keys Bi 104a and Br 101a, may optionally be queried and sent back to authentication server 111 and initiator 102, in order for initiator 102 and DPP server 103 to conduct the steps and message flows depicted and described in connection with Figure 5a. Other possibilities exist as well for a query and communication of data between an authentication server 111 and a device database 104 after an authorization step 219a without departing from the scope of the present invention. Note that in exemplary embodiments, initiator bootstrap private key bi 104b can continue to reside within device database 104 and/or network 107 and may optionally not be sent back to initiator 102 in order to enhance security of the operation of a DPP 191 by combined operation of an initiator 102 and DPP server 103 as depicted in Figure 3a, Figure 5a, and Figure 6a.

A device database 104 in Figure 2c can comprise a database for a plurality of devices 101, and device database 104 was depicted in Figure 1a above and also Figure 2b below. Device database 104 can record data for the plurality of devices 101 in a system 100, including data for device identities, responder MAC addresses 101d, device public bootstrap keys Br 101a, initiator bootstrap PKI keys, PKEX keys 198, and initiator configuration 130, and other data as well in order to support the operation of a DPP 191 using an initiator 102 with DPP server 103. A device database 104 could also record additional data pertaining to device 101 with ID.device 101i, such as (i) a DPP server 103 to use with device 101 and/or initiator 102, where the location or identity of DPP server 103 could be in the form of a DPP server URL 103b (depicted as URL-DPPS 103b), and also (ii) a certificate for DPP server 103 comprising cert.DPPS 103c. In a step 219b. device database 104 can use the identities received from authentication server 111 in order to query for data such as (i) through (iv) described in two paragraphs above.

For a step 219b, if device database 104 does not record or have the device 101 identity ID.device 101i or ID-token.device 206, or initiator configuration 130, and/or URL-DPPS 103b, then for step 219b device database 104 could point authentication server 111 to another server or location for the exemplary queried data (i) through (iv) described in three paragraphs above and depicted in Figure 2a. The other server or location for queried data (i) through (iv) could be recorded possibly by a device manufacturer which could also record a second device database 104. Note that a device manufacturer could record some of the data for a device database 104, since the data such as MAC address, public bootstrap keys, etc. could be recorded upon manufacturing, although other possibilities exist as well.

Recording URL-DPPS 130b in a database 104 can provide greater flexibility for conducting a DPP 191 and configuration step 106 with a device 101 since an initiator user 102u or device owner may prefer that a DPP 191 or configuration step 106 be performed using a particular DPP server 103. However, the DPP server 103 may not be known by a device manufacturer at the time of manufacturing of device 101. In other words, (a) a device manufacturer cannot write a value for a

Petitioner Samsung Ex-1004, 1074

URL-DPPS 103b to memory of device 101 (or on a tag 101c) if (b) the device manufacturer does not know the value before the device 101 is shipped away from the device manufacturer facilities. Further, an initiator 102 may not know which of a plurality of possible DPP servers 103 to utilize when conducting a DPP 191 with a device 101 without receiving the data for a URL-DPPS 130b via IP network 113 such as from a network 107 or authentication server 111.

At step 219c, authentication server 111 could receive and process data from device database 104. The received data could comprise (i) initiator configuration 130 for ID.device 101i or ID-token.device 206, and (ii) a DPP server URL 103b for initiator 102 to utilize in order to conduct a device provisioning protocol 191 and configuration step 106 with device 101 (possibly with a certificate cert.DPPS 103c), (iii) MAC address for responder 101x of MAC.R 101d, and (iv) optionally, public bootstrap keys Bi 104a and Br 101a or a PKEX key 198 for conducting a public key exchange protocol. Although not depicted, an identity for device 101 or initiator 102 could also be returned in the data received for step 219c, since authentication server 111 may process data for a plurality of different initiators 102 and devices 101. In another exemplary embodiment, authentication server 111 could record a device database 104 for a plurality of devices 101 in a system 100 and system 200 and in this embodiment then step 219c could represent querying a locally stored device database 104 for authentication server 111 to obtain the depicted data. Further, in exemplary embodiments where initiator 102 can operate as an initiator 102* depicted in Figure 1b (and not as an initiator proxy 102' depicted in Figure 3a, Figure 5a, and Figure 5b below), then the present invention also contemplates that initiator bootstrap private key bi 104b could also be received by authentication server 111 and subsequently sent in a message 220 to initiator 102.

As depicted in Figure 2a, authentication server 111 can send initiator 102 a message 220 through secure connection 214, where message 220 can include a device 101 identity such as ID-token.device 206 (since initiator 102 may communicate with a plurality of devices 101), (i) initiator configuration 130 for ID.device 101i or ID-token.device 206, and (ii) a DPP server URL of URL-DPPS 103b for initiator 102 to utilize in order to conduct a device provisioning protocol 191 and configuration step 106 with device 101 (possibly with a certificate cert.DPPS 103c), (iii) MAC address for responder 101x of MAC.R 101d, and (iv) optionally, public responder bootstrap keys Bi 104a and Br 101a or a PKEX key 198 for conducting a public key exchange protocol. For embodiments where public bootstrap keys Bi 104a and Br 101a are sent in message 220, then also associated data for the keys could be sent such as a selected set of parameters 199a, such that an initiator 102 could determine which cryptographic parameters are associated with the keys (such as an ECC curve name, key lengths, hash algorithms, etc.) An authentication server 111 could receive a selected set of cryptographic parameters 199a for bootstrap keys from device database 104.

Initiator 102 could receive message 220 and read and record the data received. ID-token.device 206 can be useful in a message 220 for initiator 102, because initiator 102 could

Petitioner Samsung Ex-1004, 1075

communicate with a plurality of devices 101 over time and thus keep track of which message 220 is for which device 101. Note that ID.device 101i could be sent instead of or in addition to ID-token.device 206.

At step 221, initiator 102 could process data received in message 220. Initiator 102 could determine if data for initiator configuration 130 is supported by initiator 102, such as if a WiFi radio 102w supports a specified operating class 130a. For example, responder 101x may operate on WiFi at 60 Ghz and WiFi radio 102w may not support 60 Ghz, and in this case initiator 102 could in step 221 abort attempts to conduct a device provisioning protocol 191 with responder 101x. At step 221, initiator 102 could also determine if other data received in message 220 is supported by initiator 102, such as if key lengths or associated parameter sets 199a for any keys received in message 220 are supported by initiator 102 and potentially DPP application 102g. For example, if a bootstrap public key Br 101a is received in message 220 with a selected set of cryptographic parameters 199a that are not supported by DPP application 102g (such as a curve name not supported by initiator 102 with DPP application 102g), then initiator 102 could terminate further attempts at conducting a DPP 191 and also notify initiator user 101a.

Note that as contemplated herein, the transfer of a PKI key in a message can also be with the transfer of a corresponding selected set of cryptographic parameters 199a. If a message is depicted in Figure 2a (and other figures below), as transferring a PKI key, such as, but not limited to, Br 101a, then a selected set of cryptographic parameters 199a can be sent with the key as well. If parameters 199a are not received by a node when receiving a PKI key, or the parameters are otherwise unknown (such as the parameters not being previously selected and sent/received), then initiator 102 can operate with the assumption the PKI keys are for set "A" in a set of parameters 199 below in Figure 2b, which would also comprise the basic set of parameters for device interoperability as specified by cryptographic suite 1 in section 3.2.3 of DPP specification version 1.0.

At step 222, initiator 102 could determine an initiator mode 306a (from Figure 3a below) for conducting a device provisioning protocol 191 and configuration step 106, using data received in a message 220. If the key comprising Br 101a, with possibly parameters 199a is received in message 220 along with a command for initiator 102 to conduct responder only authentication 141, then initiator 102 could operate as an initiator 102* depicted in Figure 1b. If the keys comprising Br 101a, Bi 104a, bi 104b are received in a message 220 along with a command for initiator 102 to conduct a mutual authentication 142, then initiator 102 could operate as an initiator 102* depicted in Figure 1c. If no bootstrap PKI keys are received by initiator 102 in a message 220 (but other data is valid such as initiator configuration 130), then initiator 102 can begin operating as an initiator proxy 102' as depicted in Figure 3a. If only bootstrap PKI keys Br 101a and Bi 104a are received in a message 220, then initiator 102 could proceed with a configuration mode to support subsequent DPP 191 and configuration step 106 as depicted in Figure 5a below. Other possibilities exist as well for

Petitioner Samsung Ex-1004, 1076

an initiator 102 to select the steps for conducting a device provisioning protocol 191 using data or commands in a message 220 without departing from the scope of the present invention. In another exemplary embodiment, a DPP server 103 sends initiator 102' the initiator mode 306a (in Figure 3a) for initiator 102' to use for a DPP 191 with DPP server 103. Note that as contemplated herein, steps or actions that are described as initiator 102 or initiator 102' performing the action could also be performed by (i) a DPP application 102g or (ii) the computing device operating initiator 102, such as a mobile phone, smart phone, tablet, or even (iii) a WiFi access point 105 as depicted in Figure 8 below.

Initiator 102 could then conduct configuration step 127, which was depicted and described in connection with Figure 1d above. In summary, initiator 102 could backup or record any previously active user configuration 131 operating for a WiFi radio 102w in initiator 102, such as (a) recording user operating class 131a in a nonvolatile memory 102f in order to (b) restore the configuration 131 at a later time. In exemplary embodiments, a step 127 comprises initiator 102 reading the initiator configuration 130 and activating them with WiFi radio 102w in order to transmit DPP message 122 to responder 101x, optionally using MAC.R 101d in order to send unicast messages to responder 101x.

In an alternative exemplary embodiment to that depicted in Figure 2a, discovery server 110 could be optionally omitted, and data within tag 101c or tag value 101w could specify authentication server 111 as well as data for Config-provisioning.ID.device 212. However, the use of a discovery server 110 can be preferred for some exemplary embodiments since a discovery server 110 could provide increased flexibility over the lifetime of device 101, which could be a decade or longer, since parameters Config-provisioning.ID.device 212 and related data could change over time. Including all data for config-provisioning.ID.device 212 in tag value 101w or tag 101c may reduce flexibility for changing or updating the data over time, such as specifying (i) an updated DPP application 102g (where app versions for smartphones may be updated relatively frequently such as more than once a year), or other (ii) updated data for network 107 over time. Further, the use of a discovery server 110 can support potentially millions of devices communicating potentially with many different authentication servers 111 and DPP servers 103.

For another alternative exemplary embodiment to that depicted in Figure 2a, although message 220 with data for initiator 102 is depicted as flowing from authentication server 111 to initiator 102, other possibilities exist as well for initiator 102 to receive the data within message 220 without departing from the scope of the present invention. In some embodiments, data within message 220 could be obtained by initiator 102 from a DPP server 103 directly, and for these embodiments, a mutually authenticated secure session similar to secure session 214 could be set up between initiator 102 and DPP server 103. Further, the functionality for different servers in Figure 2a could be combined, such that an authentication server 111 also operates as a discovery server 110.

Petitioner Samsung Ex-1004, 1077

In another embodiment, a DPP server 103 could operate as all of a discovery server 110 and authentication sever 111, and other possibilities exist as well without departing from the scope of the present invention.

Figure 2b

Figure 2b is an illustration of an exemplary device database, with tables for a device database recording exemplary data, in accordance with exemplary embodiments. A device database 104 depicted and described above in connection with Figure 1a and Figure 2a can record data for (i) an unconfigured initiator 102 or (ii) a configured initiator 102' and a DPP server 103 to conduct a device provisioning protocol 191 with a device 101 operating a responder 101x. A database 104 could record a plurality of tables for a plurality of devices 101 and/or responders 101x, such as the depicted values for a key table 104k, a cryptographic parameters table 104p, and a device table 104d. Other possibilities exist as well for the organization, tables, and recorded data within a device database 104 without departing from the scope of the present invention. Data within device database 104 could be encrypted using a symmetric key. As depicted in Figure 1a, a network 107 could operate and record a device database 104, although another entity besides network 107 could also operate and record the device database 104, such as a device manufacturer or a device owner.

Device database 104 can record sets of ID-token.device 206, ID.device 101i, MAC.R 101d for the MAC address of the responder 101d, global operating class 130a, channel list 130b, cryptographic parameters 199, and a secret shared key 198 for a public key exchange protocol (PKEX). These values within a device database 104 could be recorded in a device 101 and/or responder 101x upon manufacturing of a device 101. The exemplary data in device database 104 could be obtained from the manufacturer. For embodiments where some data is not previously recorded in a device database 104 before an initiator 102' conducts a step 301 as depicted below in Figure 3a, then a network 107 operating device database 104 could query the device 101 manufacturer or a device 101 distributor in order to receive and record values for a device database 104, and other possibilities exist as well for the source of device 101 data within a device database 104.

In device database 104, ID-token.device 206 can comprise a unique, obfuscated identity for device 101, such as the identity read from a tag 101c on device 101. ID.device 101i can comprise an identity for device 101i that is not obfuscated, and could be a serial number for device 101. Although Figure 2a above depicts a tag 101c as recording ID-token.device 206, the tag 101c could alternatively record ID.device 101i and send the non-obfuscated identity to initiator 102 in Figure 2a. The two identity values shown for a device 101 in a device database 104 allows a DPP server 103 to map between ID-token.device 206 and ID.device 101i, and also convert one value into another.

Petitioner Samsung Ex-1004, 1078

MAC.R 101d can comprise the MAC address used by a responder for conducting a device provisioning protocol 191. In other words, MAC.R 101d can be the MAC address used by responder 101x in order to listen for DPP authentication request 122 messages from an initiator 102'. In other words, MAC.R 101d can be used by an initiator 102' in order to send a unicast message. In some exemplary embodiments, MAC.R 101d can be omitted from a device database 104 or otherwise not available to a network 107, and in this case an initiator 102' can send a DPP authentication request message 122 as a broadcast message on a WiFi radio link 192 to responder 101x.

Global operating class 130a and channel list 130b can specify the settings a device 101 WiFi radio 101h will use with responder 101x, where responder 101x periodically listens or monitor for DPP authentication request messages 122 over time as using different RF frequencies or channel numbers within ISM band RF radio spectrum. As depicted in Figure 2b, an initiator configuration 130 can include channel list 130b and cryptographic parameters 199. An initiator configuration 130 could also include the global operating class 130a and other information could be included as well in an initiator configuration, such as timer values for retry of sending DPP 191 messages, timer values for timeouts for sending and receiving messages, as well as counters for a maximum number of retries on sending messages.

Although not depicted in Figure 2b, an initiator configuration 130 in a device database 104 could also specify an initiator mode 306a (in Figure 3a and Figure 3b below) for initiator 102' to operate with responder 101x, where different modes could comprise (i) DPP server 103 records and operates on all boostrap PKI keys for initiator 102', or (ii) initiator 102' records and operates on bootstrap public keys and derived ephemeral keys, but not initiator bootstrap private key bi 104b (e.g. as depicted in Figure 5a), and other possibilities exist as well. Sets of cryptographic parameters 199 in device database can represent different sets of cryptographic parameters supported by responder 101x in order to conduct a device provisioning protocol 191. For example, some devices 101 may use a first set of cryptographic parameters with PKI keys such as bootstrap and ephemeral keys, and other devices may use a second set of cryptographic parameters, which could be specified by cryptographic parameters 199. Note that devices 101 can optionally support more than one set of cryptographic parameters, as depicted.

Device database 104 can record a key table 104k, which could comprise exemplary keys as depicted. A key table 104k can include a secret shared PKEX key 198, responder bootstrap public key Br 101a, initiator bootstrap public key Bi 104a and initiator bootstrap private key bi 104b. PKEX key 198 could be used with a public key exchange protocol as specified in IETF draft document "Public Key Exchange", which is hereby incorporated by reference in its entirety. In some exemplary embodiments, a PKEX key 198 can be omitted, such as for the example data depicted with an "NA" in Figure 2b. The values depicted of X1, X2, etc. for a PKEX key 198 could

Petitioner Samsung Ex-1004, 1079

comprise the secret key for each device 101 identified by ID-token.device 206 or MAC.R 101d. For the exemplary database 104 depicted in Figure 2a and other tables herein, a numeric value following a constant name can depict a subsequent or different instance or numeric value for the constant. In other words, "X1" and "X2" for a PKEX key 198 can represent different values of "X" for the PKEX. Similarly, values of "Br-1" and "Br-2" depict different values for a responder bootstrap public key Br 101a, but exemplary values of "Br1" and then a second "Br1" can depict that the same value or number may be used for two different responders 101x.

Responder bootstrap public key Br 101a in a device database 104 can represent the bootstrap public key for responder 101x. Br 101a was depicted and described above in connection with Figure 1a and Figure 1b above. In exemplary embodiments, a device database 104 can record a plurality of different responder bootstrap public keys Br 101a, where each key can be associated with different responders 101x. Although not depicted in Figure 2b, for some embodiments different responders 101x could also share the same numeric value for a responder bootstrap public key Br 101a. Different responder bootstrap public keys Br 101a can be associated with a different selected set of parameters 199a. The manufacturer or entity configuring responder 101x in device 101 may not know which of a plurality of different possible sets of cryptographic parameters 199 may be utilized by an initiator 102 for conducting a DPP 191. For example, DPP specification version 1.0 identifies several possible sets of cryptographic parameters 199 for bootstrap keys that could be utilized in a DPP 191, such as the different values for cryptographic parameters shown in Appendix A of DPP specification version 1.0. As depicted in Figure 2b, a responder bootstrap public key Br 101a of "Br-3 (A)" can represent a third instance or number for Br 101a which uses the selected set "A" of cryptographic parameters 199 (e.g. curve p256 with SHA-256), while responder bootstrap public key Br 101a of "Br-3 (B)" can represent a third instance of Br 101a which uses the selected set "B" of cryptographic parameters 199 (e.g. curve p384 with SHA-384).

Initiator bootstrap public key Bi 104a and corresponding initiator bootstrap private key bi 104b in a device database 104 can represent the bootstrap public key and the bootstrap private key for initiator 102'. Bi 104a and bi 104b were depicted and described above in connection with Figure 1a and Figure 1b above. In exemplary embodiments, a device provisioning protocol 191 using an initiator 102' and a DPP server 103 can utilize a plurality of different initiator bootstrap PKI keys, and different values for the initiator bootstrap PKI keys Bi 104a and bi 104b can be associated with different responders 101x. As discussed above in connection with Figure 1a and 1b, a responder 101x could optionally record an initiator bootstrap public key Bi 104a, and subsequently the value recorded for the responder 101x could also be stored in a device database 104. The initiator bootstrap public key Bi 104a and corresponding initiator bootstrap private key bi 104b could be stored by a device database 104 concurrently with the responder bootstrap public key Br 101a, where the responder bootstrap key could be received from a device manufacturer or device distributor. Or,

Petitioner Samsung Ex-1004, 1080

in exemplary embodiments a device manufacturer or a device distributor could operate a device database 104 and subsequently record the exemplary keys and values depicted for a responder 101x in a device database 104 in Figure 2a. As depicted in Figure 2b, some responders 101x could share a common initiator bootstrap public key Bi 104a, which could be associated with a common initiator bootstrap private key bi 104b.

The sharing of initiator bootstrap public keys Bi 104a among multiple responders 101x can simplify the manufacturing and distribution of devices by using a common initiator bootstrap public key Bi 104a for some embodiments. For the present invention, the potential tradeoff of greater simplicity for reduced security (by reusing an initiator bootstrap public key Bi 104a in multiple devices 101) can be avoided, since the initiator bootstrap private key bi 104b may optionally not be sent to the initiator 102'. Yet, with the present invention, the full set of device provisioning protocol messages 191 can be sent and received with a responder 101x using mutual authentication 142. Embodiments for the present invention depict and describe the initiator bootstrap private key bi 104b as residing within a DPP server 103 when an initiator 102' conducts a DPP 191 (as depicted and described in connection with Figure 3a, Figure 5a, and Figure 5b below). Consequently, (i) the initiator bootstrap private key bi 104b does not need to be recorded by an initiator 102* (as in Figure 1c), and (ii) an initiator bootstrap public key Bi 104a can be commonly shared between multiple responders 101x.

Thus, the present invention resolves the potential tradeoff of security versus simplification by supporting manufacturing and distribution of devices 101 with responders 101x with commonly shared initiator bootstrap public key Bi 104a (depicted in Figure 2b as "shared keys"). Additional details will be provided in Figure 3a, Figure 5a, and Figure 5b below. The initiator bootstrap private key bi 104b can be recorded in a device database 104 and subsequently sent to a DPP server 103, but optionally not sent to initiator 102', and initiator 102' can still conduct a device provisioning protocol 191 with mutual authentication 142 as depicted in Figure 1c. In exemplary embodiments, only an initiator bootstrap public key Bi 104a is recorded inside an initiator 102, while a device database 104 can record both the initiator bootstrap public key Bi 104a and the initiator bootstrap private key bi 104b.

Cryptographic parameters 199 can specify sets of cryptographic parameters that are supported by responder 101x when using bootstrap PKI keys. Cryptographic parameters 199 can be recorded in a device database 104 as a parameters table 104p. Cryptographic parameters 199 can record a collection of cryptographic algorithms or specifications such as a set identifier 199a, a key length 199b, an ECC curve name 199c, a hash algorithm 199e, a key length for a symmetric ciphering algorithm 199f, and a nonce length 199g. As contemplated herein, when a selected set of cryptographic parameters such as using the words or description "parameters 199a" or "cryptographic parameters 199a" can specify a row of parameters or values in a set of cryptographic

Petitioner Samsung Ex-1004, 1081

parameters 199, such that the collection of values in the row can be used with a device provisioning protocol 191. Set identifier 199a can be an identity for a row or set of values for cryptographic parameters 199. For example, set "A" can comprise cryptographic suite 1 as specified in section 3.2.3 of DPP specification version 1.0. Key length 199b can be the length of keys in bits for PKI keys used with a device provisioning protocol with a responder 101x. ECC Curve name 199b can be a name for an ECC curve used with PKI keys for a device provisioning protocol with a responder 101x.

Hash algorithm 199e in cryptographic parameters 199 can be the name of a secure hash algorithm, such as the exemplary SHA-256 algorithm depicted, which may also be referred to as "SHA-2". Hash algorithm 199e can be used in a key derivation function and also to generate a signature based on a received nonce. A key length for a symmetric ciphering algorithm 199f can specify the length of a symmetric key in bits to use with a symmetric ciphering algorithm. The depicted algorithm in Figure 2b for algorithm 199f is "AES-SIV", although other symmetric ciphering algorithms could be specified in a set of cryptographic parameters as well. In other words, an additional, different column in cryptographic parameters 199, such as an exemplary "symmetric algorithm" 199z (not depicted) could specify the symmetric ciphering algorithm to utilize with a responder for a device provisioning protocol. Nonce length 199g can specify the length in bits for nonces or random numbers generated by both initiator 102' and responder 101x, where the nonces can be used for generating signatures and encryption keys.

As depicted in Figure 2b, a device database 104 can also record a device table 104d. Device table 104d can record a plurality of different PKI keys for the same responder 101x. Device table 104d can include ID-device 206, initiator bootstrap keys sequence number 197, secret shared PKEX key 198, responder bootstrap public key Br 101a, initiator bootstrap public key Bi 104a and initiator bootstrap private key bi 104b. Devices 101 may be designed to be utilized for a decade or longer, and during that time it may be desirable for a device 101 with responder 101x to undergo more than one instance of a device provisioning protocol. As one of many possible examples, a device 101 could have performed a "factory reset" or equivalent clearing of nonvolatile memory, such that device 101 is restored to an unconfigured state. Or, device 101 may be sold or transferred to a different owner, where the new owner wishes to conduct a device provisioning protocol 191 with the same device 101 a second time. These are just a few of the many possible reasons why a device 101 could undergo more than one instance of a device provisioning protocol 191 during the lifetime of device 101, and other examples are possible as well. Through the use of a device table 104d, different values for an initiator bootstrap PKI keys could be utilized with the same device 101 with responder 101x over time, where the different values for PKI keys used with responder 101x could be identified by initiator bootstrap keys sequence number 197. Note that a PKEX key 198 could also

Petitioner Samsung Ex-1004, 1082

change for a device 101 over time, and initiator bootstrap keys sequence number 197 could also identify a new or different PKEX key 198.

In a device table 104d, a responder public bootstrap key Br 101a can remain relatively static over time, such as the depicted same numeric value for key Br 101a of "Br-1" used for each sequence of initiator bootstrap keys. In other words, although a device 101 may undergo a factory reset or similar second instance of a device provisioning protocol, in exemplary embodiments the same responder key Br 101a can be reused. If the responder bootstrap public key Br 101a is changed for some reason (such as a device user reprograms device 101), then a second, new responder bootstrap public key Br 101a could be recorded in device table 104d with a designation "Br-2" for the same device with a new value for sequence number 197.

In device table 104d, different values for initiator bootstrap public key Bi 104a and initiator bootstrap private key bi 104b could be utilized over time, represented by increasing numbers for sequence number 197 for the same device 101 identified by ID-token.device 206. A device table 104d could also identify devices by ID.device 101i instead. For a first sequence number 197 with a value of "1", initiator bootstrap public key Bi 104a could record a value represented as "Bi-1a". For a second sequence number 197 with a value of "2", initiator bootstrap public key Bi 104a could record a value represented as "Bi-1b", etc. In other words, the depicted number plus letter represents a different key for the same device. A different device could have initiator bootstrap public keys of "Bi-2a" for a first sequence number "1" and then a key "Bi-2b" for the second sequence number 197. Similar designations and tracking or mapping of keys could be utilized for an initiator bootstrap private key bi 104b as well.

In exemplary embodiments, the initiator bootstrap public key Bi 104a within a responder 101x can be updated by a DPP server 103, as depicted in Figure 6a below. One benefit for updating key Bi 104a is that although (i) multiple manufactured devices 101 with responders 101x could initially share a common initiator bootstrap public key Bi 104a (e.g. Bi-1a) in order to conduct a mutual authentication 142 (depicted with key table 104k in Figure 2b as "shared keys"), (ii) a DPP server 103 could subsequently update a device 101 with a different, new initiator bootstrap public key Bi 104a (e.g. Bi-1b). The tracking of (a) which initiator bootstrap public key Bi 104a (identified by a sequence number 197) is used with which responder 101x can be accomplished by (b) using a device table 104d. Alternatively, a device table could be omitted, and a device database 104 could record both (a) initial keys 104a and 104b for a manufactured responder 101x (for initial use or a "factory reset") and then (b) the subsequent, current keys 104a and 104b, and in this case a sequence number 197 could be optionally omitted.

In another embodiment keys 104a and 104b could be omitted from a device database 104, and only the responder bootstrap public key Br 101a is recorded. For these embodiments, key Br 101a may preferably not be sent to an initiator 102' using the embodiment depicted and described in

Petitioner Samsung Ex-1004, 1083

connection with Figure 3 below. Responder 101x and device 101 could be designed for use where all of (i) tag 101c does not record the responder bootstrap public key Br 101a, (ii) the entity recording key Br 101a is not designed to send or transfer the key Br 101a outside of a database 104 and a DPP server 103, and (iii) access to data, calculations, or algorithm output where key Br 101a is used as input would only be provided to an authenticated initiator 102'. Subsequently, responder 101x can know that any initiator 102' that has access to the responder bootstrap public key Br 101a (indirectly via DPP server 103) can then be mutually authenticated. In this manner, each of PXEX key 198, and initiator bootstrap PKI keys could also be omitted and still responder 101x can trust data from initiator 102', such as a configuration object 109. In other words, in the present invention a system 100 can (i) operate without a mutual authentication 142 from DPP Specification version 1.0 shown in Figure 1c, and (ii) without an initiator 102' receiving the responder bootstrap public key Br 101a, and (iii) with DPP server 103 operating on the key and securely storing the key such that no other entities unnecessarily receive the responder bootstrap public key Br 101a, and in this manner a responder 101x can strongly mutually authenticate the initiator 102' (to a sufficient level to receive credentials 109), since only an initiator 102' that is authenticated with DPP server 103 could successful send and receive DPP message 191 with responder 101x.

Figure 3a

Figure 3a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device database, a DPP server, an initiator proxy, and a responder, in accordance with exemplary embodiments. Before starting steps and message flows depicted in Figure 3a, the initiator proxy 102' and device database depicted for system 300 in Figure 3a may previously complete exemplary message flows and steps depicted in Figure 2a above. System 300 can include a device database 104, DPP server 103, initiator proxy 102', and responder 101x operating in device 101. Initiator proxy 102' can communicate with DPP server 103 via IP network 113 using an access network 112, as depicted in Figure 2a. As contemplated herein, an initiator proxy 102' depicted in Figure 3a may also be referred to as initiator 102', and an initiator 102' and initiator proxy 102' may be equivalent. In other words, an initiator 102' may operate as an initiator with some functions and operations by proxy with DPP server 103.

Initiator 102' can be different than initiator 102* in Figure 1b and Figure 1c, since (a) some PKI and cryptographic computations (plus recording of PKI keys) that would normally be performed by an initiator 102* as contemplated by DPP specification version 1.0 can (b) be performed by DPP server 103 and thus initiator 102' is different than an initiator 102*. For system 300, DPP server 103 and device database 104 may establish a secure session setup 302a, which could comprise establishing a secure communications link between the two servers using protocols such as TLS,

Petitioner Samsung Ex-1004, 1084

IPSec, a virtual private network (VPN), a secure shell (SSH), or similar networking, transport, or application layer technologies in order to establish secure communications between DPP server 103 and database 104. Secure session setup 302a can be (i) equivalent or similar to secure session setup 214 depicted in Figure 2a above, and (ii) utilize certificates cert.DPPS 103c for DPP server 103 and a certificate for database 104 in order to provide mutual authentication and mutual key derivation for a symmetric encryption key.

Initiator 102' in system 300 can operate a running DPP application 102g and WiFi radio 102w operating with initiator configuration 130 from Figure 2a and Figure 1d, in order to communicate with a WiFi radio 101h operating in device 101 with responder 101x, where the communications can use a WiFI data link 192. DPP application 102g may previously be downloaded in a step 213 as depicted in Figure 2a and subsequently launched or activated for operation in a system 300. In exemplary embodiments, DPP application 102g may be (i) included or distributed along with an operating system 102e running on initiator 102' or (ii) acquired at a different time than a step 213, and thus from a separate download of DPP application 102g. In exemplary embodiments, WiFi radio 102w operating in initiator 102' can send either broadcast or unicast messages or datagrams to device 101 with responder 101x when using initiator configuration 130. In exemplary embodiments, responder 101x listens for incoming messages using the same initiator configuration 130.

DPP server 103 and initiator 102' can establish a secure session 302b, where secure session 302b could comprise a session using TLS, DTLS, IPSec, a VPN, and other possibilities exist as well. Initiator 102' can previously record the URL to reach DPP server 103 based on the value URL-DPPS 103b received in a message 220 in Figure 2a. Initiator 102' can also use the certificate for DPP server 103 comprising cert.DPPS 103c also previously received in a message 220. As depicted in Figure 3a, secure session 302b could also comprise DPP server 103 receiving and using an initiator certificate cert0.initiator 102m in order to authenticate initiator 102'. Alternatively, DPP server 103 could authenticate an initiator user 103a via a user ID and password entered by user 102u into a user interface 102k for initiator 102'. In some exemplary embodiments, authentication of an initiator 102' or an initiator user 102u could be omitted, such as if an initiator certificate cert0.initiator 102m is not present in initiator 102.

At step 302c in Figure 3, device 101 may power on from an unpowered or "deep sleep" state. Power could be provided from a battery or converted "wall power". Device 101 could (i) load the equivalent or responder version of information contained in initiator configuration 130 using a bootloader program and (ii) power on WiFi radio 101h. Responder 101 using WiFi radio 101h in device 101 could periodically cycle through a channels list 130b in order to listen for incoming DPP authentication request messages 122. In exemplary embodiments, before initiator 102' has conducted a step 127 to load initiator configuration 130, a device 101 can periodically power up and

Petitioner Samsung Ex-1004, 1085

attempt to listen for incoming messages 122. In other words, device 101 can use equivalent information for initiator configuration 130 as the default state or condition, even without initiator 102' being present or configured, since device 101 may not know the time or place where initiator 102' with initiator configuration 130 may be present. Responder 101x could also read the values for responder bootstrap public key Br 101a and initiator bootstrap public key Bi 104a, if present, in a step 302c.

As depicted in Figure 3a, initiator 102' using DPP application 102g can conduct a step 303a to collect an identity list 304, where identity list 304 comprises a list of identities for elements of systems 100, 200, and 300 around or nearby to device 101 during installation and/or operation. Identity list 304 can include a networks available list 304c. In order to conduct a step 303a, initiator 102' may need to temporarily disconnect from an access network 112, such as a mobile phone network, in order to perform a full frequency scan of all available mobile network operators and network access technologies around device 101 in order to collect a networks available list 304a. For embodiments where initiator 102' temporarily disconnects from an access network 112 or other cases where secure session 302b may be dropped, initiator 102' and DPP server 103 can pause secure session 302b and then restore the session when connectivity for initiator 102' through access network 112 is restored.

The networks available list 304c can be useful for DPP server 103 to select the best or a preferred access point 105 from the networks available list 304c for device 101. As depicted in Figure 3a, a networks available list 304c can also include the access technology available from a plurality of access networks 112 and access points 105 around device 101, such as 2G, 3G, 4G LTE, 5G, WiFi, NB-IoT, LoRaWAN, LPWAN, etc. Identification of all available wireless networks around device 101 can support DPP server 103 or network 107 in a later step 308b below to receive or determine a working set of credentials 109 for device 101 to utilize in order to obtain network connectivity via the access network 112. In exemplary embodiments, the networks available list 304c of surrounding wireless networks also includes an associated RF signal strength for each, such as RSSI as well as a frequency band, frequency list in Mhz, or RF channel for each surrounding wireless network. In exemplary embodiments, initiator 102' can observe WiFi broadcast packets from network Wifi Access Point 105 depicted in Figure 1a using SSID.Network-AP 109a and include the value in the networks available list 304c.

A step 303a may also comprise initiator 102' collecting a list of identities for a monitored unit associated with a device 101, and include the identities in the identity list 304. In exemplary embodiments, a step 303a in Figure 3a may comprise initiator 102' using a camera 102z to scan for a bar code or QR code on a surrounding monitored unit in order to collect a monitored unit identity 304b. In exemplary embodiments, device 101 can include sensors and transducers for monitoring and controlling external devices or equipment. A monitored unit for a device 101 could comprise

Petitioner Samsung Ex-1004, 1086

exemplary values of industrial equipment, health care equipment, security systems, etc. A step 303a could also comprise initiator 102' obtaining physical location 304a for device 101, which could comprise a room number, zone in a building or floor, shelf number, rack number, etc. Physical location 304a could also comprise a set of geographical coordinates for a value of physical location 304a or (ii) estimated geographical coordinates or similar physical location of device 101. In exemplary embodiments, initiator 102' can comprise a mobile phone or smart phone and can have a GPS or similar receiver for outdoor location, or use WiFi or Bluetooth for approximate indoor locations, and record the location in a physical location 304a.

Although step 303a to collect an identity list 304 for device 101 is depicted as after secure session setup 302a (confirming mutual authentication between initiator 102' and DPP server 103 in session 302a), step 303a to collect an identity list 304 could be conducted at other times, such as before Figure 3a or after a initiator user 102u takes initiator 102 to the approximate physical location for installation or configuration of device 101, including possibly during a step 202 above in Figure 2a. Or, in other exemplary embodiments the use of an identity list 304 can optionally be omitted, if initiator 102' and DPP server 103 can record the set of credentials 109 for a selected network access point 105, where the selected network access point 105 can be recorded without an identity list 304 (such as if the correct or preferred network access point 105 is known to DPP server 103 before a read step 202 in Figure 2a). As depicted in Figure 3a, initiator 102' can then send DPP server 103 a message 305 through secure session 302a, where message 305 can include ID-token.device 206 (depicted and described for message 204 in Figure 2a above) and the identity list 304. DPP server 103 can receive message 305 and record the data received in a server database 103a. The values of Physical Location 304a and Monitored Unit ID 304b can be recorded for support of subsequent operation of device 101 after a configuration step 106, and DPP server 103 could also send the values to another server supporting device 101 during operation such as a reporting server (not shown). DPP Server 103 can then send ID-token.device 206 through secure session 302a to device database 104 in a message 306.

A step 302d for DPP server 103 can also comprise DPP server 103 querying a server database 103a for an initiator bootstrap keys sequence number 197 for device 101 using ID-token.device 206 (or ID.device 101i). In exemplary embodiments, a device 101 with responder 101x could use multiple different initiator bootstrap public keys Bi 104a over time, such as updating the initiator bootstrap public key Bi 104a using the steps depicted and described in connection with Figure 6a below after a DPP 191 and configuration step 106 has been previously completed. The multiple possible reasons and benefits for updating and initiator bootstrap public keys Bi 104a are discussed with a step 608 in Figure 6a below. A server database 103a could record a value initiator bootstrap keys sequence number 197 in order to track which initiator bootstrap public key Bi 104a is currently recorded in responder 101x for device 101. For other embodiments, a responder 101x in

Petitioner Samsung Ex-1004, 1087

device 101 could also record a plurality of different initiator bootstrap public keys Bi 104a concurrently, such as a first initiator bootstrap public key Bi 104a for a first network 107 and a second initiator bootstrap public key Bi 104a for a second network 107. Responder 101x could use the multiple different, concurrently available initiator bootstrap public keys Bi 104a to concurrently listen for a DPP authentication request message 122 from different possible networks 107. In a step 302d, A DPP server 103 could use an initiator bootstrap keys sequence number 197 to select the current, preferred initiator bootstrap public key Bi 104a for conducting a DPP 191 by an initiator 102'. DPP server 103 can send a database 104 a message 307 with an identity for device 101 of ID-token.device 206 (or identity ID.device 101i), and the selected initiator bootstrap keys sequence number 197.

At step 306, DPP server 103 can select the mode of operation for initiator 102' to conduct a DPP 191 with device 101. The selected mode of operation for initiator 102' can comprise a mode 306a or an initiator mode 306a, which could be recorded in a server database 103a as depicted and described in connection with Figure 3b below. The selected mode 306a can specify the distribution of PKI keys between DPP server 103 and initiator 102'. The selected mode 306a can determine the subsequent series of steps and message flows between DPP server 103 and initiator 102'. A first mode 306a could (a) specify that DPP server 103 records all public and private bootstrap and ephemerals keys for initiator 102', plus the public keys Br 101a and Pr 101e for responder 101x such that (b) resulting cryptographic operations and ECDH key exchanges and symmetric ciphering reside within DPP server 103 and results are transmitted to initiator 102'. This first mode 306a could represent the message flows and steps in Figure 3a. A second mode 306a could specify that (i.a) initiator bootstrap PKI keys 104a and 104b reside within DPP server 103 and (i.b) responder bootstrap and ephemeral public keys for responder 101x reside within initiator 102', such that (ii) resulting cryptographic operations and key exchanges are shared between DPP server 103 and initiator 102'. This second mode 306a could represent the message flows and steps depicted in Figure 5a.

A third mode 306a selected in a step 306 for DPP server 103 could specify the same recording of bootstrap and ephemeral keys as the second mode, with the addition that DPP server 103 receives and uses the initiator ephemeral private key pi 102b, also with resulting cryptographic operations and key exchanges are shared between DPP server 103 and initiator 102' (but with a different allocation than the second mode). This third mode 306a could represent the message flows and steps depicted in Figure 5b. In exemplary embodiments, all of the first, second, and third modes operate with initiator bootstrap private key bi 104b residing within DPP server 103 and not shared with initiator 102'. By not sharing initiator bootstrap private key bi 104b with initiator 102' (which could be a device with much less security than DPP server 103), the initiator bootstrap private key bi 104b can remain secured within network 107. These embodiments where initiator bootstrap private

Petitioner Samsung Ex-1004, 1088

key bi 104b remains in network 107 (e.g. in DPP server 103 and device database 104) support securely sharing the same initiator bootstrap public key Bi 104a potentially across a plurality of device 101 with responders 101x. Sharing the same initiator bootstrap public key Bi 104a across multiple devices supports recording the key Bi 104a in a device 101 during manufacturing. This simplifies and enables the use of a device provisioning protocol 191 across multiple devices 101 with mutual authentication, where the initiator 102' does not need to receive or be provisioned with the initiator bootstrap private key bi 104b.

In other words, without the technology described in the present invention, a device provisioning protocol with mutual authentication can essentially shift the problem of (a) securely mutually authenticating a device and providing credentials 109 to a device 101 to a new, but related problem of (b) securely providing an initiator bootstrap private key bi 104b to an initiator 102* . The first, second, and third modes 306a can solve this shifted/new problem since the initiator bootstrap private key bi 104b can permanently reside in a network 107 and does not need to be sent to the initiator 102' or an initiator 102*. However, with the technology in the present invention, the initiator 102' can still transmit and receive the full set of a device provisioning protocol 191 messages with a responder 101x as specified in DPP specification version 1.0.

A fourth mode 306a could specify that initiator 102' records all of the initiator bootstrap and ephemeral PKI keys, as well as the responder 101x public keys, such that resulting cryptographic operations and key exchanges reside in initiator 102'. This fourth mode 306a could represent the message flows and steps depicted in Figure 1b or Figure 1c, and in this case initiator 102' could comprise an initiator 102*. Other possibilities exist as well without departing from the scope of the present invention. Additional details for the different modes 306a selected in a step 306, with the resulting different allocation or recording of PKI keys are described below in Figure 3b for a server database 103a.

At step 308a, DPP server 103 can process data received in message 304 and the received identity list 304. DPP server 103 can use the values in a networks available list 304c in order to select an access network 112 for device 101. A step 308a could comprise DPP server 103 selecting a primary access network 308aa and a backup access network 308ab for a device 101, as depicted and described in connection with Figure 3b below. The selected access networks 308aa and 308ab for device 101 in a step 308a could be based on criteria such as bandwidth cost, expected energy or power requirements (i.e. 3G/4G may use more power than LPWAN), signal strength measurements recorded with a networks available list 304c, and commercial terms such as agreements between a (i) device owner for device 101 and (ii) networks in a networks available list 304c. DPP server 103 could also query other servers not depicted in Figure 3a to select an access network 112 from the networks available list 304c. In other words, a preferred access network 112 in the form of a primary access network 308aa for device 101 may need to be selected by DPP server 103 in a step

Petitioner Samsung Ex-1004, 1089

308a before the credentials 109 for the access network (or network access point 105) can be obtained. Although Figure 3a depicts DPP server 103 as conducting a step 308a in order to select a primary access network 308aa and backup access network 308ab, in exemplary embodiments a different server than DPP server 103 in a network 107 could (i) securely receive the networks available list 304c and (ii) select the primary access network 308aa and backup access network 308ab.

After the selection of a primary access network 112 (308aa) and a backup access network 112 (308ab) in a step 308a, DPP server 103 could perform a step 308b depicted in order to obtain a set of network credentials 109 for both the selected primary access network 308aa and the selected backup access network 308ab. For embodiments where (a) a network 107 operating DPP server 103 comprises a selected access network 308ab or 308ab, then (b) the credentials 109 for device 101 can be obtained from another server in a network 107 from Figure 1a and recorded in DPP server 103. For example, if a network access point 105 for network 107 is selected as an access network 112 for device 101, then network 107 would normally already record the set of credentials 109 for access point 105. For embodiments where a selected access network 112 (e.g. selected primary access network 308aa) for device 101 other than network 107 operates the access point 105, then for a step 308b DPP server 103 could query the access network 112 for the equivalent of credentials 109. Additional details regarding embodiments where a different network than network 107 provide credentials for device 101 are depicted and described in connection with Figure 3b and Figure 6b below.

Note that in exemplary embodiments, a DPP server 103 can preferably conduct a step 308a to select an access network 112 and a step 308b to obtain credentials 109 for the selected access network 112 before continuing with a device provisioning protocol 191. The reason is that if either (i) a network cannot be selected potential from a networks available list 304c or (ii) credentials 109 cannot be obtained for the selected network, then there may be no reason to continue with a device provisioning protocol 191 since DPP server 103 may not have credentials 109 to send to device 101. In Figure 3a, although a step 308a and step 308b are depicted as taking place after DPP server 103 sends message 307 with a query for PKI keys to device database 104, in some embodiments a step 308a and step 308b could take place before DPP server 103 sends message 307 to device database 104.

Further, in some exemplary embodiments where initiator 102' uses network access point 105 with SSID.network-AP 109a with a preshared secret key (PSK) or similar authentication (as an access network 112 for initiator 102'), then a message 305 could comprise initiator 102' sending DPP server 103 the set of network credentials 109 used by initiator 102'. DPP server 103 could then (i) record the credentials 109 in a server database 103a further described below in Figure 3b, (ii) use the value SSID.network-AP 109a as a selected primary access network 308aa, and (ii) use the

Petitioner Samsung Ex-1004, 1090

received credentials 109 from initiator 102' in a message 305 as the credentials 109 for device 101. Other possibilities regarding the source of credentials 109 exist without departing from the scope of the present invention.

Device database 104 can receive message 307 with ID-token.device 206 (or similarly ID.device 101i) and initiator bootstrap keys sequence number 197. Device database in step 309 can conduct a query of device database 104 for data associated with ID-token.device 206 and initiator bootstrap keys sequence number 197. In exemplary embodiments, a query in a step 309 can select an initiator boostrap private key bi 104b and an initiator bootstrap public key Bi 104a using the initiator bootstrap keys sequence number 197 from a device table 104d for device 101. In this manner, for embodiments where initiator bootstrap PKI keys for device 101 are updated over time (e.g. such as depicted and described in connection with Figure 6a), then the most recently updated initiator bootstrap PKI keys can be selected by a database query in a step 309. The query in step 309 can return values for a message 310 from device database 104 to DPP server 103. The values in a message 310 can comprise data for ID-token.device 206 and the values can include responder bootstrap public key Br 101a, initiator bootstrap public key Bi 104a, initiator bootstrap private key bi 104b, a device identity ID.device 101i, a selected set of cryptographic parameters 199a for the PKI keys, and optionally PKEX key 198. Although not depicted in Figure 3a, message 310 could also include ID-token.device 206 so that DPP server 103 can track which of a possible plurality of initiators 102' and devices 101 that message 310 is associated with. In some embodiments, a step 302d and the use of a sequence number 197 can be omitted, and a query in step 309 and message 310 can return the most recently updated or recorded values for responder bootstrap public key Br 101a, initiator bootstrap public key Bi 104a, and initiator bootstrap private key bi 104b.

For embodiments where (A) a DPP session 191 between initiator 102' and device 101 fails due to a possible mismatch between (i) initiator bootstrap PKI keys selected in a query for step 309 and (ii) the actual initiator bootstrap public key Bi 104a recorded and used by a responder 101x in device 101 (such as the case where device 101 has undergone a "factory reset" to restore previously recorded initiator bootstrap public key Bi 104a), then (B) DPP server 103 could send a second message 307 with a request to use the oldest recorded initiator bootstrap PKI keys from a device table 104d (such as possibly the lowest or first recorded sequence number 197 for device 101 in device table 104d). For these embodiments where (A) a DPP session 191 between initiator 102' and device 101 fails due to a possible mismatch between (i) initiator bootstrap PKI keys selected in a query for step 309 and (ii) the actual initiator bootstrap public key Bi 104a recorded and used by a responder 101x in device 101, then (B) a query in step 309 can be for the oldest recorded initiator bootstrap PKI keys from a device table 104d. In other exemplary embodiments, DPP server 103 could send message 307 with a specific initiator bootstrap keys sequence number 197, and a query in a step 309 could be for a specific set of responder bootstrap public key Br 101a, initiator bootstrap

Petitioner Samsung Ex-1004, 1091

public key Bi 104a, initiator bootstrap private key bi 104b designated by the initiator bootstrap keys sequence number 197 in a device table 104d. Further, a key table 104k in a device database 104 can record the most recently updated values and keys for a device 101, while a device table 104d can record the historical values and PKI keys for a device 101 used over time, where the historical different values or PKI keys can be tracked, queried, or selected using an initiator bootstrap keys sequence number 197.

DPP server 103 can receive the message 310 and record the data received in a server database 103a. As depicted in Figure 3a, DPP server 103 can then send message 311 to initiator 102', where message 311 can include (i) the selected mode 306a of operation for initiator 102' and DPP server 103 for conducting a DPP 191 with responder 101x, (ii) a selected set of cryptographic parameters 199a for use with PKI keys and a device provisioning protocol, and (iii) optionally a PKEX key 198 for conducting a public key exchange with responder 101x. Message 311 can also optionally include device identity ID-token.device 206 in order to associate the data in message 311 with a particular responder 101x, since initiator 102' could potentially communicate with a plurality of different devices 101 over time.

At step 303b initiator 102' can receive and process the message 311. Initiator 102' and DPP server 103 can begin operating in the selected mode 306a for conducting a device provisioning protocol 191. Initiator 102' and DPP server 103 can apply and begin using the selected set of cryptographic parameters 199a. For a step 308b, if a PKEX key 198 is present in a message 311, then initiator 102' can conduct a public key exchange protocol with responder 101x in order to obtain responder bootstrap public key Br 101a via WiFi radio 102w and "in band". The public key exchange protocol is specified in the DPP specification version 1.0 section 5.6. If PKEX is optionally used between responder 101x and initiator 102', then initiator 102' could receive the responder bootstrap public key Br 101a and send the key Br 101a to DPP server 103 (and in this case DPP server 103 or database 104 may not have previously recorded responder bootstrap public key Br 101a). The use of a PKEX protocol and key 198 can also be optionally omitted in some embodiments, if responder bootstrap public key Br 101a can be recorded by a network 107 prior to a step 303b.

As depicted in Figure 3a, the collection of steps and messages before a step 313 can be grouped together into a step 301. Step 301 with the collective group of sub-steps and messages depicted for a step 301 can also be performed in other embodiments depicted in Figure 5a and Figure 5b below. Subsequent messages and steps depicted in Figure 3a starting after step 301 can represent a first mode 306a for DPP server 103 and initiator 102', where the allocation of recording PKI keys and symmetric keys from key exchanges for the first mode 306a are depicted below in Figure 3b.

A step 313 can comprise deriving an initiator ephemeral private key pi 102b and an initiator ephemeral public key Pi 102a, and in Figure 3a the DPP server 103 is depicted as conducting the

Petitioner Samsung Ex-1004, 1092

step 313 to derive the initiator ephemeral PKI keys. Step 313 can use a random number generator to produce a random number, which could be input into a key pair generation algorithm using the set of cryptographic parameters 199a. The resulting PKI key values from conducting a step 313 can comprise the initiator ephemeral private key pi 102b and the initiator ephemeral public key Pi 102a. The resulting PKI key values from conducting a step 313, if conducted by a DPP server 103, can be recorded in a server database 103a, as depicted in Figure 3b below. Other embodiments depicted below contemplate a step 313 to derive initiator ephemeral PKI keys being conducted by an initiator 102'. In summary, some steps depicted as conducted by a DPP server 103 in Figure 3a could be conducted instead by an initiator 102' for other embodiments such as those depicted in Figure 5a and Figure 5b, with a corresponding change in supporting message flows between DPP server 103 and initiator 102'.

Step 314 can comprise DPP server 103 conducting a first initiator key exchange step 314 using an Elliptic Curve Diffie Hellman (ECDH) key exchange 401a, as depicted and described in connection with Figure 4a below. Data input for a step 314 can comprise the initiator ephemeral private key pi 102b and the recorded responder bootstrap public key Br 101a. Data output from a step 314 can comprise a key k1 as described in Figure 4a below. Step 314 can comprise the key exchange and symmetric key derivation step in DPP specification version 1.0 section 6.2.2 in order to for an initiator to obtain key k1.

At step 315, DPP server 103 can generate a random number for an initiator nonce 317a using parameters 199a. As used herein and throughout the present invention, the words "random number" can also mean a pseudo random number, since random numbers generated by nodes may not be truly random. However, their "randomness" may be sufficient in order to securely conduct the cryptographic operations contemplated herein. DPP server 103 can also determine the initiator capabilities, which could comprise a configurator for the initiator as depicted in Figure 3a (where responder 101x would be the enrollee). In other embodiments, the initiator 102' could have initiator capabilities as an enrollee (where the responder 101x could be the configurator). A step 315 can comprise encrypting the initiator nonce 317a and initiator capabilities using the derived key k1 from step 314 in order to create a ciphertext 315a. An encryption step 315 step is further depicted and described in connection with Figure 4a below.

At step 316, DPP server 103 can calculate secure hash values for initiator bootstrap public key Bi 104a and responder bootstrap public key Br 101a, using a secure hash algorithm 199e. The values are depicted in Figure 3a as "H(Bi)" and "H(Br)", respectively. DPP server 103 can then send initiator 102' message 317, which can include the hash value for the initiator bootstrap public key Bi 104a and the responder bootstrap public key Br 101a, the derived initiator ephemeral public key Pi 102a, and ciphertext 315a, where ciphertext 315a is encrypted with key k1 and includes an

Petitioner Samsung Ex-1004, 1093

initiator nonce 317a and the initiator capabilities. Initiator 102' can receive message 317 via the secure session 302b.

For a step 316a (not depicted in Figure 3a, but conducted by initiator 102' after receipt of message 317 and also depicted in Figure 5a below), initiator 102' can then send DPP Authentication Request message 122 to responder 101x, using WiFi radio 102w with initiator configuration 130. In exemplary embodiments, initiator 102' sends message 122 as a unicast message to responder 101x using MAC.R 101d as the destination/recpient, and initiator 102' received MAC.R 101d in a previous message such as message 220. Alternatively, initiator 102' sends message 122 as a broadcast message (and MAC.R 101d would not normally be required for broadcast). However, unicast for WiFi can be preferred, since unicast messages are acknowledged at the data link layer. In a step 316a, initiator 102' may need to repeat message 122 multiple times on several different channels in a channel list 130b from initiator configuration 130, since (i) responder 101x may be periodically sleeping in order to conserve battery power, and (ii) responder 101x may also be cycling through a channel list 130b in order to listen for message 122 on different channels (and the cycling by initiator 102' and responder 101x may not be synchronized).

In addition, DPP authentication request message 122 in Figure 3a can have some differences than message 122 in Figure 1b and Figure 1c. Message 122 in Figure 3a can originate from an initiator proxy 102' (where initiator proxy 102' may not record an initiator bootstrap private key bi 104b for responder 101x), while message 122 in Figure 1c can be from an initiator 102*, where initiator 102* can record and operate on the initiator bootstrap private key bi 104b. In other words, from the perspective of a responder 101x message 122 in Figure 1c and Figure 3a could be identical, but the initiator 102* and 102' can be different, depending on the recording and processing of PKI keys and shared secret keys. Further, although a message 122, message 123, etc. as depicted herein shows key values of "k1", "ke", etc., the depiction of the keys is to illustrate that ciphertext is encrypted with the corresponding key by the use of brackets. In other words, although the key values are shown with the messages, the actual key values are not transmitted inside or with the message in exemplary embodiments. The equivalent syntax is used in DPP specification version 1.0.

Responder 101x can receive message 122 and perform a series of steps to process the message and respond to the initiator 102'. Responder 101x in device 101 can compare (i) hash values received in message 122 with (ii) the calculated hash values for bootstrap public keys from a step 302c. If (a) the hash values match or are equal, then responder 101x can proceed to subsequent steps, and otherwise (b) continue to cycle through a channel list 130b in order to listen for incoming messages 122. At step 318, responder 101x can also generate two random numbers using a random number generator. The first random number can be used as a responder ephemeral private key pr 101f. The responder ephemeral private key pr 101f can be input into an ECC key pair generation

Petitioner Samsung Ex-1004, 1094

algorithm in order to output a responder ephemeral public key Pr 101e. The set of cryptographic parameters 199a used for the responder ephemeral PKI key pair can match the recorded parameters 199a used with the bootstrap public keys such as responder bootstrap public key Br 101a. The second random number generated by responder 101x can be used as a responder nonce 318b. Responder 101x can also determine responder capabilities 319b, which could comprise the capabilities for an enrollee.

Responder 101x can conduct a first responder key exchange step 319, where the details for a key exchange step 319 are depicted in Figure 4a below. In summary, the received initiator ephemeral public key Pi 102a (from message 122) and the responder bootstrap private key br 101b can be input into an ECDH key exchange in order to mutually derive a shared secret key M 410. The shared secret key M 410 can be input into a key derivation function 402a in order for responder 101x to derive mutually shared key k1 314a. A responder 101x can then conduct a decryption step 320 as depicted in Figure 4a below. Responder can input the received ciphertext 315a into a symmetric ciphering algorithm 405b and key k1 314a using parameter 199f in order to read the plaintext 403, which could comprise the initiator nonce 317a and the initiator capabilities. In exemplary embodiments, the initiator 102' can have capabilities of a configurator, and responder 101x can select responder capabilities or an enrollee in a step 320. Steps 319 and steps 320 can also comprise the first four paragraphs of section 6.2.3 of DPP specification version 1.0.

Responder 101x can then conduct a second responder key exchange step 314b, where the details for a key exchange step 314b are depicted and described in connection with Figure 4b below. In summary, the initiator ephemeral public key Pi 102a received in message 122 can be input into an ECDH key exchange algorithm 401a along with the responder ephemeral private key pr 101f derived in a step 318 above in order to output a secret shared key N 411. The value for shared secret key N 411 can be recorded and used in the derivation of a key ke 327a in a third key exchange step 321 below. For a step 314b, the value for the shared secret key N 411 can be input into a key derivation function 402b in order to derive a mutually shared symmetric key k2 314c.

Responder 101x can conduct a third responder key exchange step 321, where the details for a key exchange step 321 are depicted in Figure 4c below. In summary, both (i) the derived responder ephemeral private key pr 101f from a step 318 with the recorded responder bootstrap private key br 101b, and (ii) the recorded initiator bootstrap public key Bi 104a can be input into an ECDH key exchange algorithm 401c in order to output a secret shared key L 412. For embodiments where a DPP responder authentication 141 is conducted by responder 101x without the use of an initiator bootstrap public key Bi 104a, then calculation of a secret shared key L 412 can be optionally omitted in a step 321. Step 321 for responder 101x can then comprise a key derivation function 402c using the mutually derived secret shared keys M 410, N 411, and optionally L 412 in order to mutually

Petitioner Samsung Ex-1004, 1095

derive secret shared key ke 327. An example for a key exchange step 321 is described in point 2 of section 6.2.3 of DPP specification version 1.0 on page 48.

Responder 101x can then conduct an encryption step 322, where the details for an encryption step 322 are depicted and described in connection with Figure 4c below. Step 322 can also comprise responder 101x creating or calculating a value R-auth 404, where R-auth 404 can comprise a secure hash value of the initiator nonce 317a, the second random number R-nonce 318b from a step 318 above, and public PKI keys. The value for R-auth 404 in a step 322 can comprise the value in point 3 of section 6.2.3 of DPP specification version 1.0. An encryption step 322 can comprise responder 101x encrypting the value for R-auth 404 using the mutually derived key key 327 from the third key exchange step 321 above, where the output of encryption step 322 can comprise ciphertext 322a.

Responder 101x can then conduct an encryption step 315b, where the details for an encryption step 315b are depicted and described in connection with Figure 4b below. Ciphertext 322a from an encryption step 322 in the paragraph above can also be encrypted again for an encryption step 315b. The symmetric ciphering key for encryption step 315b can comprise symmetric key k2 314c. Encryption step 315b, encryption step 322, and decryption step 320 can use a common symmetric ciphering algorithm 405a, which could comprise the Advanced Encryption Standard with Synthetic Initialization Vectors (AES-SIV) (and deciphering algorithm) also with a common set of symmetric ciphering parameters 199f. Symmetric ciphering parameters 199f can also specify the use of a block chaining mode such as cipher block chaining (CBC), counter mode (CTR), or Galois/Counter mode (GCM) and other possibilities exist as well. In addition, symmetric ciphering parameters 199f could specify a mode for message authentication, which could comprise a CMAC mode as specified in NIST publication SP-800-38B. In exemplary embodiments, a symmetric ciphering algorithm 405a can comprise the AES-SIV algorithm as specified in IETF RFC 5297. The output from an encryption step 315b can be ciphertext 315c, as depicted in Figure 4b below. As depicted in Figure 3a, the collection of steps for responder 101x comprising steps 318, 319, etc. through step 315b can comprise a step 325.

Responder 101x can then send an DPP Authentication Response message 123, where message 123 can include hash values of the bootstrap public keys H(Br 101a), H(Bi 104a), the responder ephemeral public key Pr 101e, and the ciphertext 315c. The ciphertext 315c can have encrypted values for (i) R-nonce 318b, (ii) I-nonce 317a, (iii) R-capabilities 318c, and (iv) ciphertext 322a. The ciphertext 315c can be encrypted with mutually derived symmetric encryption key k2 314c as shown for an encryption step 315b in Figure 4b. Ciphertext 322a can be encrypted with the mutually derived symmetric key ke 327a and can include a plaintext value for R-auth 404. As described above, R-auth 404 can comprise a hash value over an initiator nonce 317a received and the responder nonce 318b.

Petitioner Samsung Ex-1004, 1096

As depicted in Figure 3a, initiator 102' can receive DPP Authentication Response message 123. In exemplary embodiments, initiator 102' can conduct a step 323 to process the received responder ephemeral public key Pr 101e, such as evaluating that the received key in message 123 uses the parameters 199b specified in cryptographic parameters 199a. In addition, a step 323 can comprise initiator 102' can conduct public key validation step to ensure that received responder ephemeral public key Pr 101e is properly formatted and does not contain weak or compromising cryptographic data. For example, a step 323 could comprise initiator 102' performing the steps for an ECC Full Public-Key Validation Routine in section 5.6.2.3.2 of FIPS publication SP 800-56A (revision 2). Alternatively, step 323 can comprise initiator 102' performing the steps ECC Partial Public-Key Validation Routine in section 5.6.2.3.3 of the same FIPS publication. Other example steps within a public key validation step 323 can comprise (i) verifying the public key is not at the "point of infinity", and (ii) verifying the coordinates of the point for the public key are in the range [0, p-1], where p is the prime defining the finite field. Other possibilities exist as well for evaluating and validating a received public key is cryptographically secure in a public key validation step 323, without departing from the scope of the present invention. As contemplated in the present invention, an initiator 102', a DPP server 103, or a responder 101x can conduct a public key validation step 323 each time a public key is received.

In exemplary embodiments, initiator 102' can send DPP server 103 a message 324 using secure session 302b. Message 324 can include the ID-token.device 206 (or alternatively ID.device 101b), and the ciphertext 315c. Hash values for public bootstrap keys can be useful over link 192 between responder 101x and initiator 102' in order to confirm identities, but the hash values can be substituted with an identity for device 101 in message 324 between initiator 102' and DPP server 103. The ciphertext 315c in message 324 can be the same as ciphertext 315c in message 123 in Figure 3a. In other embodiments, the hash values for the public bootstrap keys of initiator 102 and responder 101x can be included in a message 324, thereby increasing the message size by at least 64 bytes.

DPP server 103 can receive message 324 from initiator 102' and conduct a series of steps in order to process the message and reply to the initiator 102'. In exemplary embodiments, DPP server 103 can also perform a public key validation step 323 on the received responder ephemeral public key Pr 101e. Although a step 323 by DPP server 103 may appear redundant with a step 323 by initiator 102', DPP server 103 may trust its own computation for a step 323 with a higher level than relying solely on a step 323 performed by initiator 102'. Initiator 102' as operating in Figure 3a could also comprise a device with a lower level of security than DP server 103. After a step 323, DPP server 103 can record the validated, received responder ephemeral public key Pr 101e in a server database 103a, as depicted in Figure 3b below.

Petitioner Samsung Ex-1004, 1097

At step 319a, DPP server 103 can conduct a second initiator key exchange step 319a, where DPP server 103 can input the received, validated responder ephemeral public key Pr 101e and the derived initiator ephemeral private key pi 102b into an ECDH key exchange algorithm 401b in order to calculate a shared secret key N 411. The second initiator key exchange step 319a is also depicted and described in connection with Figure 4b below. Shared secret key N 411 can also be used by DPP server 103 in a step 327 below. DPP server 103 can input shared secret key N 411 into a key derivation function 402b in order to mutually derived the symmetric encryption key k2 314c.

At step 320b, DPP server 103 can use the derived key k2 314c from a step 319a in order to conduct a decryption step 320b on received ciphertext 315c. A decryption step 320b on received ciphertext 315c using key k2 314c is depicted an described in connection with Figure 4b below. DPP server 103 can read the plaintext 403b values from decryption step 320b of ciphertext 315c. The plaintext values 403b can include the initiator nonce 317a from a message 317 and message 122 above. The plaintext values 403b can also include a second ciphertext 322a (e.g. ciphertext 322a can be inside ciphertext 315c). DPP server 103 verifies that the initiator nonce 317a received in message 123 and decrypted in step 320b is equal to the initiator nonce 317a value sent in message 122 above. If the two initiator nonce 317a values are not equal, then DPP server 103 can abort the DPP procedure and subsequent steps.

At step 327, DPP server 103 can conduct a third initiator key exchange step 327, where DPP server 103 can determine the secret shared key L 412 by inputting (i) the sum or combination of the responder bootstrap public key Br 101a and the responder ephemeral public key Pr 101e, and (ii) the initiator bootstrap private key bi 104b into an ECDH key exchange algorithm 401c. A third server key exchange step 327 is depicted and described in connection with Figure 4c below. The shared secret key L 412 along with shared secret keys M 410 and N 411 can be used with a key derivation function 402c in order to output the mutually derived symmetric ciphering key ke 327a.

At step 328, DPP server 103 can conduct a decryption step 328 on the ciphertext 322a using the key ke 327a. A decryption step 328 is depicted and described in connection with Figure 4c below. DPP server 103 can use the decryption step 328 in order to read the plaintext value of R-auth 404. R-auth 404 can comprise a hash 199e value over the combination of initiator nonce 317a and the responder nonce 318b. At step 328, DPP server 103 can then also internally calculate the same value for R-auth 404, and if the internally calculated value and the received, decrypted value are equal, then the responder 101x can be consider authenticated with DPP server 103. The calculation of R-auth 404 by DPP server 103 can be according to section 6.2.3 of DPP specification version 1.0.

DPP server 103 can then conduct a step 322b, where step 322b can comprise calculating an initiator authentication value I-auth 404a. Initiator authentication value 404a can be similar to responder authentication value R-auth 404, except the values of initiator nonce 317a can be sequenced before a responder nonce 318b inside a hash 199e calculation over the two nonces. A

Petitioner Samsung Ex-1004, 1098

DPP server 103 can then encrypt the initiator authentication value 404a using an encryption step 322 using key ke 327a, as depicted in Figure 4c below. In other words, an encryption step 322 in Figure 4c depicts the encryption of R-auth 404, and encryption step 322b can comprise a DPP server 103 encrypting the initiator authentication value I-auth 404a using the key ke 327a derived in a step 327.

DPP server 103 can then send initiator 102' a message 329, where message 329 can include (i) an identity for device 101, such as ID-token.device 206, although ID.device 101i could be used instead, and (ii) the encrypted initiator authentication value I-auth 404a. Initiator 102' can receive message 329 over secure session 302b. In an exemplary embodiment, the value MAC.R 101d could also be within message 329, such that initiator 102' can use the MAC.R 101d to send data in message 329 to responder 101x using the MAC.R 101d. Initiator 102' can determine which of a plurality of responders 101x is directed towards based on the value of ID-token.device 206 or MAC.R 101d. Initiator 102' can previously record the hash 199e values for the initiator and responder bootstrap public keys. Initiator 102 can take the recorded hash 199e values for the initiator and responder bootstrap public keys and append to them the encrypted value for I-auth 404a received in message 329. Initiator 102' can send responder 101x a DPP authentication confirm message 123a, as depicted in Figure 3a. The data and steps to determine and calculate an DPP authentication confirm message 123a can be according to section 6.2.4 of DPP specification version 1.0.

Note that although the DPP authentication confirm message 123a can be equivalent to the DPP authentication confirm message from section 6.2.4 of DPP specification version 1.0, an important difference can be the operation of initiator 102' in Figure 3a versus an initiator as contemplated by DPP specification version 1.0. In the embodiment of the present invention as depicted in Figure 3a, initiator 102' can record neither the responder bootstrap public key Br 101a, nor the initiator bootstrap private key bi 104b. So, using the technology depicted in Figure 3a, an initiator 102' can (i) send and receive messages that are fully compatible with a DPP protocol 191, but (ii) DPP server 103 can record keys 101a and 104b. In other words, a responder 101x may not "know" that an initiator operates as either (i) an initiator 102* from Figure 1b or Figure 1c, or an initiator 102' from Figure 3a since the messages transmitted and received by the responder 101x can be fully compatible with a DPP 191.

Responder 101x can receive the DPP authentication confirm message 123a and process the message. Responder 101x can listen for either (i) a broadcast message with the hash 199e values for the bootstrap public keys 101a and 104a, or (ii) for a unicast message to MAC.R 101d. Responder 101x can first conduct a decryption step 328b, where decryption step 328b can use the same steps as a decryption step 328 as depicted in Figure 4c below, with a difference being decryption step 328b operates on the initiator authentication value I-auth 404a instead of the R-auth 404 value.

Petitioner Samsung Ex-1004, 1099

In step 331, responder 101x can compare the decrypted value for I-auth 404a with a calculated value for I-auth 404a and if the two values match then data from the initiator 102' can be considered by responder 101x to be authenticated. After an authentication step 331, in step 332, responder 101x can take on a roll of enrollee, such that responder 101x can prepare to send a configuration attribute 330. Configuration attribute 330 can be used to specify that responder 101x can take the roll of enrollee and is expecting a set of network credentials 109 from initiator 102'. A configuration attribute 330 can be formatted and contain data fields according to section 6.3.4 of DPP specification version 1.0. For the exemplary embodiment depicted in Figure 1a, device 101 with responder 101x would prefer a role of WiFi client (or "sta") for a configuration attribute 330, but other possibilities exist as well. For the exemplary embodiment depicted in Figure 7 below a responder 101x operating within an access point 105 would prefer a role of WiFi access point (or "ap") for a configuration attribute 330.

In step 332, responder 101x can also generate a third responder random number, for an enrollee nonce E-nonce 332a. E-nonce 332a and other nonce values in Figure 3a such as the initiator nonce I-nonce 317a and responder nonce R-nonce 318b can be processed using nonce parameters 199g. Responder 101x can then conduct an encryption step 322c, where encryption step 322c can comprise (i) an encryption step 322 using key ke 327a, and (ii) the plaintext to be encrypted can be the E-nonce 332a and the configuration attribute 330. The output of an encryption step 322c can be a ciphertext 322d. As depicted in Figure 3a, the collection of steps comprising 328b through 322c for responder 101x can comprise a step 339, where a responder 101x processes both message 123a and message 123b in a step 339. Responder 101x can then send or transmit to initiator 102' a DPP Configuration Request message 123b, where the configuration request 123b can include the ciphertext 322d, where the plaintext in ciphertext 322d can include E-nonce 332a and the configuration object 330. DPP Configuration Request message 123b can be processed by responder 101x according to section 6.3.3 of DPP specification version 1.0.

Initiator 102' can receive DPP Configuration Request message 123b via connection 192 and using DPP application 102g. As contemplated herein, a DPP application 102g can perform the steps for initiator 102' depicted in Figure 3a, such as transmitting/receiving DPP 191 messages using a first MAC.I 102n for connection 192 and transmitting/receiving messages with DPP server 103 using MAC.N 102q via access network 112. Although not depicted in Figure 3a, DPP application 102g could also perform the depicted steps 323 to validate the received responder ephemeral public key Pr 101e from message 123b. For embodiments where initiator 102' uses a WAN radio 102h to connect with access network 112 and DPP server 103, then DPP application 102g could also use a different identity than MAC.N 102q to identify the physical interface of WAN radio 102h, such as an IMEI, device identity ID.device 102i, or other possibilities exist as well. For other embodiments where initiator 102' uses network access point 105 for an access network 112 to connect with DPP

Petitioner Samsung Ex-1004, 1100

server 103, then MAC.I 102n could also potentially be the same as MAC.N 102q. Other possibilities exist as well without departing from the scope of the present invention for an identity used by initiator 102' to send messages to DPP server 103.

Initiator 102' can send DPP server 103 a message 334 via secure connection 302b, where message 334 can include the ciphertext 322d. Although not depicted in Figure 3a, message 334 can also include an identity for device 101, such as, but not limited to, ID-token.device 206 or ID.device 101b, since both DPP server 103 and initiator 102' can communicate with a plurality of different devices 101 over time, and DPP server 103 may prefer to know which ciphertext 322d is associated with which device 101 in order to process the ciphertext 322d. As depicted in Figure 3a, message 334 can include the encrypted values for E-nonce 332a and configuration attribute 330 in ciphertext 322d.

DPP server 103 can receive message 334 and process the message 334. DPP server 103 can conduct a decryption step 328c using decryption step 328 depicted in Figure 4c below. A difference between decryption step 328 and decryption step 328c is that decryption step 328c uses the ciphertext 322d as input into the symmetric decryption algorithm 405b. DPP server 103 can use the previously calculated key ke 327a in order to read the plaintext within ciphertext 322d. DPP server 103 can record the plaintext E-nonce 332a and configuration attribute 330.

In step 335, DPP server 103 can process the received configuration attribute 330. For the exemplary embodiment depicted in Figure 1a, configuration attribute 330 can specify that device 101 with responder 101x may have a role of WiFi client. In exemplary embodiments, DPP server 103 can previously select a primary access network 308aa in a step 308a above, using the received networks available list 304c from initiator 102'. Also, in a step 308b above, DPP server 103 could receive or record the set of network credentials 109 associated with the selected primary access network 308aa for device 101 using device identity ID-token.device 206 (or ID.device 101i). In step 335, DPP server can process the set of network credentials 109 for primary access network 308aa into a configuration object. The configuration object with network credentials 109 can comprise a connector or configuration object as specified in section 6.3.5 and 6.3.6 of DPP specification version 1.0. The configuration object with network credentials 109 can be signed using the recorded initiator ephemeral private key pi 102b, corresponding to public key Pi 102a sent in message 317 and 122 above. Or, in another embodiment the configuration object with network credentials 109 can be signed using the recorded initiator bootstrap private key bi 104b. The steps to create and verify a signature with ECC PKI keys can be according to FIPS publication 186-4, and other possibilities exist as well.

In a step 335, DPP server 103 can process a configuration object for network credentials 109, and the configuration object can include an SSID value of SSID.network-AP 109a, a pairwise master key value of PMK.network-AP 109b, and configuration values of config.network-AP 109c. The

Petitioner Samsung Ex-1004, 1101

value of PMK.network-AP 109b can also include a value for PMKID (e.g. PMK identifier), such that PMK.network-AP 109b is used by each of a client and an access point with the PMKID value. The configuration values of config.network-AP 109c in a configuration object could also specify required supporting data for operation of device 101 with an access point 105 after a configuration step 106, such as an operating class 130a, channel list 130b, an authentication mode for device 101 to operate, which could be PSK, PMK, SAE, EAP, also with any required supporting data and/or identities. The configuration values of config.network-AP 109c in a configuration object could also include a passphrase or pre-shared key values of PSK for legacy WPA2 networks. As contemplated herein, a set of network credentials 109 can include either (i) network credentials 109 for a selected primary access network 308aa, or (ii) network credentials 109 for both a selected primary access network 308aa and a selected backup access network 308ab. Or, in another exemplary embodiment, although a single configuration object of network credentials 109 is depicted in Figure 3a, DPP server 103 could process (i) a first set of network credentials 109 for selected primary access network 308aa into a first configuration object and then (ii) s second set of network access credentials 109 for selected backup access network 308ab into a second configuration object.

At step 322e, DPP server 103 can conduct an encryption step 322e using an encryption step 322 depicted and described in connection with Figure 4c below. Encryption step 322e can be performed using key ke 327a for device 101. For encryption step 322e, the plaintext values of (i) the E-nonce 332a and (ii) the configuration object comprising network credentials 109 can be input into a symmetric encryption algorithm 405a in order to generate a ciphertext 322f. DPP server 103 can then send initiator 102' a message 338, where message 338 includes the ciphertext 322f. Message 338 can also optionally include an identity for device 101 such as ID-token.device 206 or ID.device 101i. Message 338 can be sent via secure connection 302b, as depicted in Figure 3a.

Initiator 102' using DPP application 102g can receive message 338 via secure connection 302b. Initiator 102' can use an optional identity for device 101 in a message 338 in order to determine which of a potential plurality of devices 101 associated with ciphertext 332f in order to forward the data. Initiator 102' can then send responder 101x a DPP configuration response message 124 via connection 192, where message 124 can include the ciphertext 322f, and the ciphertext 322f can include the encrypted E-nonce 332a and (i) the encrypted, signed network credentials 109.

Responder 101x can receive message 124 from initiator 102' with ciphertext 322f. Responder 101x can conduct a decryption step 328d using decryption step 328 depicted in Figure 4c below. A difference between decryption step 328 and decryption step 328d is that decryption step 328d uses the ciphertext 322f as input into the symmetric decryption algorithm 405b. Responder 101x can use the previously calculated key ke 327a (from key exchange step 321) in order to read the plaintext within ciphertext 322f. Responder 101x can confirm that plaintext in ciphertext 322f

Petitioner Samsung Ex-1004, 1102

includes the previously send E-nonce 332a. Responder 101x can read (i) the plaintext values for credentials 109. Note that credentials 109 may be previously signed by DPP server 103 using a private key initiator ephemeral private key pi 102b for the initiator 102' (or initiator bootstrap private key bi 104b), where the private key corresponds to public key Pi 102a. The responder could also verify a signature for the credentials 109 using the ephemeral public key for the initiator Pi 102a recorded with the responder 101x, such as the initiator ephemeral public key Pi 102a or the initiator bootstrap public key Bi 104a.

After reading the network credentials 109 and optionally verifying a signature using a recorded public key 102a or 104a, device 101 can complete the configuration step 106 by loading the network credentials 109 and begin operating a WiFi radio 101h with the network credentials 109, as depicted in Figure 1a. In this manner, device 101 can obtain connectivity to network access point 105. In exemplary embodiments, the initiator 102' can conduct the initiator restore step 129 depicted and described in connection with Figure 1d above, such that initiator 102' can be returned to the previous state where WiFi radio 102w operates with a user configuration 131, thus minimizing impacts of the prior initiator configuration step 127 for an initiator user 102u.

Figure 3b

Figure 3b is an illustration of an exemplary server database, with tables for a server database recording exemplary data, in accordance with exemplary embodiments. A server database 103a depicted and described above in connection with Figure 1a and Figure 3a can record data for a DPP server 103 to work or operate with an initiator 102' in order for the initiator 102' to conduct a device provisioning protocol 191 with a device 101 operating a responder 101x. A server database 103a could record a plurality of tables for a plurality of devices 101 and a plurality of initiators 102', including the depicted device credentials table 340. Other possibilities exist as well for the organization, tables, and recorded data within a device database 103a without departing from the scope of the present invention. Data within server database 103a could be encrypted using a symmetric key. As depicted in Figure 1a, a DPP server 103 operating in a network 107 could operate and record a server database 103a.

Server database 103a can record a device identity of ID-token.device 206, a responder MAC address MAC. R 101d, an initiator MAC address MAC. I 102n (or MAC.I 102q), a mode of operation for the DPP server 103 and initiator 102' comprising initiator mode 306a, a selected set of Cryptographic Parameters 199a, an Initiator Ephemeral Public Key 102a, an Initiator Ephemeral Private Key 102b, a Responder Ephemeral Public Key 101e, Bootstrap keys 399, an optional PKEX Key 198, a Shared Secret k1 314a, a Shared Secret k2 314c, and a Shared Secret ke 327a. For exemplary embodiments, not all depicted values may be recorded in a server database 103a, as

Petitioner Samsung Ex-1004, 1103

shown by the depicted value of "-", which can mean that the value is not recorded in a server database 103a. An initiator 102' could record the value depicted as "-" for a server database 103a in Figure 3b for some initiator modes 306a, as depicted. For example, with the row in server database 103a depicted with a mode "Fig 5a", the initiator 102' could derive, record, and operate on the initiator ephemeral private key pr 102b and initiator ephemeral public key Pr 102a, and DPP server 103 may not need or record the keys. Similarly, with the row in server database 103a depicted in Figure 3b as a mode of "Fig 5b", the initiator 102' could derive shared secret keys k2 314c and ke 327e, while server database 103a could derive and record key k1 314a. Note that in alternative exemplary embodiments, a server database 103a could record the values M 410, N 411, and L 412 respectively instead of key k1 314a, k2 314c, and/or ke 327e, and either DPP server 103 or initiator 102' could use the key derivation function 402a, 402b, and/or 402c, respectively in order to convert the values into keys. In exemplary embodiments, the time that a device shared secret key or a derived ephemeral PKI key, or a bootstrap key is recorded in a server database 103a can be minimized to the minimum time necessary to conduct a DPP 191. For example, shared secret keys and ephemeral PKI keys recorded in a server database 103a can be deleted once a configuration step 106 has been completed for a device 101, such as at the time DPP server 103 receives a report 605aa from device 101 in Figure 6a below.

Initiator mode 306a can specify an operating mode for initiator 102 and DPP server 103, such that a sequence of steps and transmitting/receiving PKI and/or shared secret keys can be specified by the initiator mode 306a. An initiator mode 306a could also be referred to as a "server mode" 306a, since the mode specifies the operation of a DPP server 103. Or, initiator mode 306a could be referred to as a "mode 306a", since both steps for initiator 102' and DPP server 103 are affected by the mode. Figure 3a above also described a first, second, third, and fourth operating initiator modes 306a. A first mode is depicted as "Fig. 3a" in Figure 3b and can comprise the mode 306a depicted in Figure 3a. Note that all depicted ephemeral PKI keys and bootstrap PKI keys, as well as all depicted shared secret keys are recorded in a server database 103a for a device 101 (except responder private keys 101b and 101f) for the first mode. In other words, when conducting a DPP 191, initiator 102' and DPP server 103 would have all the depicted keys recorded in a server database 103a for the first mode. This first mode for an initiator mode 306a can be preferred for embodiments where (i) initiator 102' may have a lower level of trust with network 107, or (ii) initiator 102' could have a lower level of processing resources, or (iii) an owner for device 101 or a network 107 may prefer that responder bootstrap public key Br 101a is not shared externally from network 107, or (iv) regulatory requirements where device 101 could comprise a high value or sensitive piece of equipment. Other possibilities exist as well for the reasons different modes 306a may be used in a server database 103a to specify the steps for an initiator 102' and DPP server 103 to conduct a DPP 191 with device 101.

Petitioner Samsung Ex-1004, 1104

A second mode is depicted as "Fig. 5a" in Figure 3b, where the steps and message flows in Figure 5a for initiator 102' and DPP server 103 can comprise the second mode. Note that all relevant bootstrap PKI keys 399 are recorded in a server database 103a for the mode depicted in Figure 5a. The mode 306a can also have (i) the responder ephemeral public key Pr 101e and (ii) the shared secret key k1 314a recorded in a server database 103a. Note that initiator ephemeral PKI keys and secret shared keys can optionally not be recorded in a server database 103a for the second mode depicted in Figure 5a. This exemplary embodiment may be preferred for cases where (i) initiator 102' can handle more of the cryptographic operations (e.g. reducing DPP server 103 processing load), but (ii) a network 107 and initiator user 102u could prefer that the initiator bootstrap private key bi 104b is not transmitted to or recorded in initiator 102'. An exemplary reason the second mode of "Fig 5a" could be preferred may be embodiments where multiple different devices 101 record the same initiator bootstrap public key Bi 104a, and for these embodiments a network 107 may prefer to keep the initiator bootstrap private key bi 104b recorded in a server database 103a. This embodiment was also discussed above with the devices designated with "shared keys" in Figure 2b. As depicted in Figure 3b, devices 101 with the same recorded value or number for an initiator bootstrap public key Bi 104a could also record different responder bootstrap public keys Br 101a.

A third mode 306a is depicted as "Fig. 5b" in Figure 3b, where the steps and message flows in Figure 5b for initiator 102' and DPP server 103 can comprise the third mode. Note that all relevant bootstrap PKI keys 399 are recorded in a server database 103a for the third mode depicted in Figure 5b. The third mode 306a can be a mix between the first mode of "Fig. 3a" and the second mode of "Fig 5a". In other words, the third mode 306a of "Fig 5b" can have the same data recorded for operation of the mode in "Fig 5a", except the addition of recording the initiator ephemeral private key pi 102b and secret key k1 314a in server database 103a. Thus some additional computation resources are shifted from the initiator 102' to DPP server 103 from using a mode of "Fig. 5a" to the mode of "Fig. 5b". One reason to use the third mode of Fig. 5b could be a desire or requirement by DPP server 103 or initiator user 102u or a device 101 owner to have the cryptographic operations of a DPP 191 more evenly shared between an initiator 102' and a DPP server 103. In other words, a mode depicted in Fig. 3a could comprise a mode that is mostly on DPP server 103, a mode depicted in Figure 5a could comprise a mode that is mostly on initiator 102', and a mode depicted in Figure 5b could comprise a balanced mode where a mix of the keys and cryptographic operations are more evenly shared between DPP server 103 and initiator 102'.

A fourth mode 306a in Figure 3b can comprise initiator 102' operating as an initiator 102* depicted in Figure 1b and Figure 1c, and in this case initiator 102* can record the keys shown for the fourth mode 306a in Figure 3b. A DPP server 103 can still provide the bootstrap keys 399 shown to initiator 102* in this fourth mode of operation, or the initiator 102* could obtain the bootstrap keys 399 from other means, such as "out of band" messages 121 and 126 in Figure 1c. In addition, an

Petitioner Samsung Ex-1004, 1105

initiator 102* could obtain the bootstrap keys "in band" (e.g. via WiFi link 192) using PKEX key 198.

Bootstrap keys 399 in a server database 103a can record the bootstrap keys depicted in Figure 3b, which could comprise a responder bootstrap public key Br 101a, an initiator bootstrap public key Bi 104a, and an initiator bootstrap private key 104b. For embodiments where a device 101 can support multiple different sets of cryptographic parameters 199a, a server database 103a can record the selected set of cryptographic parameters 199a for communication between an initiator 102' and device 101 with responder 101x. The keys for "bootstrap keys 399" can be according to the selected set of cryptographic parameters 199a. In other words, a device database 104 in Figure 2b above can record all versions of the bootstrap PKI keys for all supported cryptographic parameters 199a for a device 101. However, since a set of cryptographic parameters 199a will need to be selected for a DPP 191 (e.g. one of the responder bootstrap public keys Br 101a selected), then the bootstrap keys 399 for that selected set of cryptographic parameters 199a can be recorded in a server database 103a.

A server database 103a can also record a preshared PKEX key 198, if PKEX is optionally supported by responder 101x. PKEX key 198 can be used by initiator 102' to securely transfer bootstrap public keys in band (i.e. via connection 192 in Figure 1a and Figure 3a). The use of a PKEX protocol to exchange the public keys is described in section 5.6 of DPP specification version 1.0. For embodiments where initiator 102' uses PKEX key 198 with responder 101x, the initiator bootstrap private key bi 104b preferably continues to reside within a network 107 and can be in a server database 103a. PKEX key 198 can be used with responder 101x MAC address MAC.R 101d. Embodiments can omit the use of PKEX key 198, as depicted with the value of "NA" for the key in Figure 3b. Derived, shared secret keys with responder 101x can also be recorded in a server database 103a. The derived shared secret keys could comprise key k1 314a, key k2 314c, and key ke 327e. The steps to derive and use these keys are depicted and described below in Figure 4a, Figure 4b, and Figure 4c, respectively. As discussed above, some keys may reside in server database 103a, and other keys may reside within an initiator 102' (depicted with a "-"), depending on the mode 306a.

The present invention contemplates other implementations of a device provisioning protocol besides the DPP specification version 1.0. These other implementations could have different mutually derived shared secret keys than those depicted in Figure 3b, and the different mutually derived shared secret keys could be recorded in a server database as well. For example, a different key derivation function 402 could be used to derive a secret shared key than that specified in DPP specification version 1.0. In addition, other or different PKI keys for a responder 101x and/or initiator 102' could be recorded in a server database 103a. For example, the DPP specification version 1.0 could be extended to explicitly include a DPP server 103 public and private key, in order

Petitioner Samsung Ex-1004, 1106

to extend support with a DPP server 103, and in these embodiments the DPP server 103 public and private keys could be included in a server database 103a. Other possibilities exist as well for (i) a server database 103a to record PKI keys and shared secret keys to support (ii) a device provisioning protocol without departing from the scope of the present invention.

A server database 103a could also record a device credentials table 340. Device credentials table 340 can record a device identity ID-token.device 206 (or ID.device 101i), a selected primary network 308aa, a selected backup network 308ab, a set of primary network credentials 109, and a set of backup network credentials 109. The selected primary network 308aa and selected backup network 308ab could be determined in a step 308a in Figure 3a by DPP server 103 or network 107. The set of primary network credentials 109 can comprise the set of credentials (plus configuration) for a device 101 to use with the selected primary network 308aa. The set of backup network credentials 109 can comprise the set of credentials (plus configuration) for a device 101 to use with the selected backup network 308ab. The credentials 109 in a device credentials table 340 could be obtained from during a step 308b as depicted and described above in connection with Figure 3b.

For example, with the first device depicted in a device credentials table 340, device 101 could record a selected primary network 308aa for network access point 105 in Figure 1a. The value recorded for 308aa could be SSID.network-AP 109a. The credentials in device table for the first device could comprise the credentials 109 depicted in Figure 1a. The depicted designation of "109a-1" means the first instance of a value of SSID.network-AP 109a. The device in the second row can use the same value for SSID.network-AP 109a. The device in the last depicted row can use a different value for a selected primary network 308aa of "109a-3" which could be the third distinct value in device credentials table 340. Values of "304c-1" for selected primary network 308aa can mean a first value from a networks identity list 304c from Figure 3a, values of "304c-2" can mean a second value from a networks available list 304c, etc.

For embodiments where a network other than network 107 is selected for device 101 to be configured with credentials, the credentials in a device credentials table 340 are depicted as "from ...", where "..." can be the depicted, selected network from a networks available list 304c. In exemplary embodiments, the credentials for networks external to network 107 can be encrypted by the external network using Elgamal encryption and either (i) the responder bootstrap public key Br 101a or (ii) the responder ephemeral public key Pr 101e. The DPP server 103 could send either (i) the responder bootstrap public key Br 101a or (ii) the responder ephemeral public key Pr 101e to the external network for the external network to encrypt the credentials using the public key Br 101a or Pr 101e and an Elgamal asymmetric encryption algorithm, which was also described above. Thus, the credentials are stored as ciphertext in a server database 103a and only the device 101 with responder 101x could feasibly decrypt the ciphertext in order to read the credentials for networks external to network 107. Additional details for receiving credentials from an external network, and

Petitioner Samsung Ex-1004, 1107

recording the encrypted credentials in a server database 103a are depicted and described below in connection with Figure 5a, depicted as credentials 109'.

Note the present invention contemplates that (a) a selected access network 112 for device 101 in a step 308a above in Figure 3a by a DPP server 103 or network 107 could (b) utilize wireless networking technology other than WiFi or 802.11 based technology. For example, surrounding wireless networks for device 101 could use exemplary potential use of 5G wireless technology or LPWAN networking technology depicted in a networks available list 304c in Figure 3a above for potential access networks 112 for device 101. Or, a WiFi access point could also be selected by a DPP server 103 or a network 107 for device 101 in step 308a above in Figure 3a using the networks available list 304c.

For the exemplary embodiment depicted in Figure 3a and also Figure 1a above, the exemplary network access point 105 can be selected in a step 308a using SSID.network-AP 109a from Figure 1a, although different networks in a networks available list 304c could be selected instead, based on exemplary criteria listed for a step 308a above in Figure 3a. In addition, a network access point 105 in Figure 1a can comprise a form of an access network 112 as contemplated herein, such that device 101 obtains access to IP network 113 or the globally routable Internet via an exemplary network access point 105. A selected access network 112 for device 101 in a step 308a does not need to belong to network 107 from Figure 1a, and could belong to a different network (so long as network 107 could obtain or point to credentials 109 for the access network 112 in order to conduct the configuration step 106 from Figure 1a). Further, a step 308a could comprise the selection of both a primary access network 112 for device 101 and a second, backup access network 112 for device 101. The use of a primary and backup network for device 101 can provide increased reliability for the operation of device 101, such that if the primary access network 112 is not available then device 101 could use the selected backup network in order for device 101 to maintain connectivity to an IP network 113

Figure 4a

Figure 4a is a flow chart illustrating exemplary steps for conducting a key exchange using PKI keys in order to derive a shared secret key, and for using the derived shared secret key to encrypt and decrypt data, in accordance with exemplary embodiments. Exemplary steps for an initiator 102' and a responder 101x to mutually derive keys can comprise (i) a key exchange 314 step for DPP server 103 or initiator 102' (depending on the mode 306a) to derive a symmetric encryption key k1 314a and (ii) a key exchange 319 step for a responder 101x to derive the same symmetric encryption key k1 314a. Exemplary steps for a responder 101x to decrypt data and initiator 102' for DPP server 103 (depending on a mode 306a) to encrypt data can comprise (a) an

Petitioner Samsung Ex-1004, 1108

encryption step 315 for initiator 102' or DPP server 103 (depending on a mode 306a) to utilize the symmetric encryption key k1 314a to convert plaintext to ciphertext, and (b) a decryption 320 step for responder 101x to utilize the symmetric encryption key k1 314a to convert the ciphertext received from initiator 102' into plaintext. The use of the steps for a key exchange 314, a key exchange 319, encryption 315, and decryption 320 were also depicted and described in connection with Figure 3a above. Additional detail regarding the use of these steps will be provided herein with Figure 4a. Further, although the steps are depicted specifically for the use of particular keys and plaintext/ciphertext combinations in Figure 4a, the steps illustrated can be used with different keys and plaintext/ciphertext combinations. For example, Figure 4b below depicts the same functionality used by the nodes as in Figure 4a, but with (i) different PKI keys input to mutually derive a different symmetric encryption key, (ii) a different key derivation function and (iii) different plaintext encrypted and different ciphertext decrypted.

The processes and operations, described below with respect to all of the logic flow diagrams and flow charts may include the manipulation of signals by a processor and the maintenance of these signals within data structures resident in one or more memory storage devices. For the purposes of this discussion, a process can be generally conceived to be a sequence of computer-executed steps leading to a desired result.

These steps usually require physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, compared, or otherwise manipulated. It is convention for those skilled in the art to refer to representations of these signals as bits, bytes, words, information, elements, symbols, characters, numbers, points, data, entries, objects, images, files, or the like. It should be kept in mind, however, that these and similar terms are associated with appropriate physical quantities for computer operations, and that these terms are merely conventional labels applied to physical quantities that exist within and during operation of the computer.

It should also be understood that manipulations within the computer are often referred to in terms such as listing, creating, adding, calculating, comparing, moving, receiving, determining, configuring, identifying, populating, loading, performing, executing, storing etc. that are often associated with manual operations performed by a human operator. The operations described herein can be machine operations performed in conjunction with various input provided by a human operator or user that interacts with the device, wherein one function of the device can be a computer.

In addition, it should be understood that the programs, processes, methods, etc. described herein are not related or limited to any particular computer or apparatus. Rather, various types of general purpose machines may be used with the following process in accordance with the teachings described herein.

Petitioner Samsung Ex-1004, 1109

The present invention may comprise a computer program or hardware or a combination thereof which embodies the functions described herein and illustrated in the appended flow charts. However, it should be apparent that there could be many different ways of implementing the invention in computer programming or hardware design, and the invention should not be construed as limited to any one set of computer program instructions.

Further, a skilled programmer would be able to write such a computer program or identify the appropriate hardware circuits to implement the disclosed invention without difficulty based on the flow charts and associated description in the application text, for example. Therefore, disclosure of a particular set of program code instructions or detailed hardware devices is not considered necessary for an adequate understanding of how to make and use the invention. The inventive functionality of the claimed computer implemented processes will be explained in more detail in the following description in conjunction with the remaining Figures illustrating other process flows.

Further, certain steps in the processes or process flow described in all of the logic flow diagrams below must naturally precede others for the present invention to function as described. However, the present invention is not limited to the order of the steps described if such order or sequence does not alter the functionality of the present invention. That is, it is recognized that some steps may be performed before, after, or in parallel other steps without departing from the scope and spirit of the present invention.

The processes, operations, and steps performed by the hardware and software described in this document usually include the manipulation of signals by a CPU or remote server and the maintenance of these signals within data structures resident in one or more of the local or remote memory storage devices. Such data structures impose a physical organization upon the collection of data stored within a memory storage device and represent specific electrical or magnetic elements. These symbolic representations are the means used by those skilled in the art of computer programming and computer construction to most effectively convey teachings and discoveries to others skilled in the art.

A key exchange 314 step for initiator 102' or DPP server 103 (depending on a mode 306a) to derive a symmetric encryption key 314a can utilize a selected set of cryptographic parameters 199a as depicted and described in connection with Figure 2b above. As depicted in Figure 4a, a key exchange algorithm 401a can receive input both of a responder bootstrap public key Br 101a and an initiator ephemeral private key pi 102b. The key exchange algorithm 401a could comprise a Diffie Hellman key exchange (DH), an Elliptic Curve Diffie Hellman key exchange (ECDH), and other possibilities exist as well without departing from the scope of the present invention. A key exchange algorithm 401a can support either PKI keys based on elliptic curves or RSA algorithms, although support of elliptic curves may be preferred in some exemplary embodiments due to their shorter key lengths and lower processing requirements.

Petitioner Samsung Ex-1004, 1110

A summary of ECDH as a key exchange algorithm 401a is included in the Wikipedia article titled "Elliptic Curve Diffie-Hellman" from March 9, 2018, which is herein incorporated by reference. An exemplary embodiment of key exchange algorithm 401a could comprise a "One-Pass Diffie-Hellman, C(1, 1, ECC CDH)" algorithm as described in section 6.2.2.2 on page 99 of the National Institute of Standards and Technology (NIST) document "NIST SP 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" from April, 2018 which is hereby incorporated by reference its entirety. Other key exchange algorithms in NIST SP 800-56A could be utilized as well for a key exchange algorithm 401a in Figure 4a, Figure 4b, and Figure 4c without departing from the scope of the present invention. Example calculations for an ECDH key exchange for a key exchange algorithm 401a are shown on pages 88 and 89 of DPP specification version 1.0, where a selected set of parameters 199a can use the ECC curve P-256 with PKI key lengths of 256 bits.

Other algorithms to derive a shared symmetric encryption key using public keys and private keys may also be utilized in a key exchange algorithm 401a (and 401b in Figure 4b and 401c in Figure 4c), such as, but not limited to, the American National Standards Institute (ANSI) standard X-9.63. Cryptographic parameters 199a can also include information for using a key exchange algorithm 401a and a key derivation function 402a in order to derive a commonly shared symmetric encryption key k1 314a. As contemplated herein, the terms "selected set of cryptographic parameters 199a" and "cryptographic parameters 199a", and "parameters 199a" can be equivalent. Parameters 199a input into a key exchange algorithm 401a can include a time-to-live for a key k1 314a that is derived, a supported point formats extension, where the supported point formats extension could comprise uncompressed, compressed prime, or "compressed char2" formats, as specified in ANSI X-9.62. In other words, an ECC keys input into a key exchange algorithm 401a may have several different formats and a set of parameters 199a can be useful to specify the format.

As depicted in Figure 4a, the output of a key exchange algorithm 401a, such as an ECDH key exchange, can comprise a secret shared key M 410. The secret shared key M 410 can be input into both (i) a first key derivation function 402a in order to calculate the key k1 314a and (ii) a second key derivation function 402c in order to calculate a symmetric encryption key ke 327a as depicted in Figure 4c below. Key derivation function 402a could use a hash algorithm 199e from parameters 199a, where an input into the hash algorithm 199e can comprise the value M 410. As contemplated herein, a secret shared key can also be referred to as a value, since the secret shared key can be a numeric value or a number in the form of a sequence of digits. In DPP specification version 1.0, a key derivation function 402a to determine key k1 314a is depicted on the fourth line of page 48. Other possibilities exist as well for a key derivation function 402a without departing from the scope of the present invention. For example, with other embodiments besides those within DPP specification version 1.0, the output of a key derivation function 402a can be both a symmetric

Petitioner Samsung Ex-1004, 1111

encryption key k1 314a and a message authentication code (MAC) key, where a MAC key can function to verify message integrity of ciphertexts generated with the symmetric encryption key 314a.

As depicted in Figure 4a, an encryption 315 step could be utilized with the derived symmetric encryption key k1 314a in order to convert plaintext 403b into ciphertext 315a. An encryption 315 step can utilize a symmetric encryption algorithm 405a, where symmetric encryption algorithm 405a could comprise an algorithm according to the Advanced Encryption Standard (AES), Blowfish, and Triple Data Encryption Standard (3DES), and other possibilities exist as well. As depicted in Figure 4a, a symmetric encryption algorithm 405a can comprise the AES-SIV algorithm as specified in IETF RFC 5297. Parameters 199f could specify settings for a symmetric ciphering algorithm 405a as well, such as key length and block cipher modes such as, but not limited to, ECB, CBC, OFB, and CFB. Parameters 199f may also specify an authenticated encryption mode such as CCM (NIST SP800-38C), GCM (NIST SP800-38D), and other examples exist as well.

Symmetric encryption algorithm 405a can accept input of (i) plaintext 403b, (ii) symmetric encryption key k1 314a, (iii) parameters 199f, and (iv) optionally an initialization vector (not shown in Fig. 4a), and output ciphertext 315a. Although not depicted in Figure 4a, and encryption 405a step can output a message authentication code (MAC), where the MAC can be calculated with a MAC key from a key derivation function 402a. In exemplary embodiments the use of a MAC and MAC key, as well as an initialization vector can be optionally omitted. The exemplary listed values for plaintext and ciphertext in step 315 were described in connection with depicted step 315 in Figure 3a.

Key exchange 319 step for responder 101x depicted in Figure 4a can correspond to key exchange 314 step by an initiator 102' or DPP server 103 (depending on a mode 306a). Key exchange 319 step can comprise inputting or using the initiator ephemeral Public key Pi 102a (from message 122) and the responder bootstrap private key br 101b into key exchange algorithm 401a, which can comprise the same or equivalent key exchange algorithm 401a depicted and described in connection with key exchange step 314 described above. Other elements or algorithms within a key exchange step 319 can be equivalent to a key exchange step 314 above, including the use of shared parameters 199a. In this manner, initiator 102' or DPP server 103 (depending on a mode 306a) and responder 101x can securely derive the same symmetric encryption key k1 314a as depicted in Figure 4a.

A decryption 320 step allows responder 101x to convert the ciphertext 315a received in a message 122 from Figure 3a into plaintext 403a. Decryption 320 step can utilize a symmetric decryption algorithm 405b, which could comprise the same algorithm used in symmetric encryption algorithm 405a except the algorithm being used for decryption instead of encryption. For example, if symmetric encryption algorithm 405a in Figure 4a comprises an AES-SIV encryption, then

Petitioner Samsung Ex-1004, 1112

symmetric decryption algorithm 405b can comprise an AES-SIV decryption. Note that the same values are input into symmetric decryption algorithm 405b as symmetric encryption algorithm 405a, such as symmetric encryption key k1 314a and parameters 199f in order to convert ciphertext 315a back into plaintext 403a. Responder 101x can the read and process plaintext 403a after a decryption 320 step.

For embodiments where a MAC code is received with ciphertext 315a, responder 101x can also verify the MAC code using a MAC key from a key derivation function 402a in a step 319 in order to verify the integrity of ciphertext 315a. Note that AES-SIV and similar symmetric ciphering algorithms can have a message integrity check built into the algorithm, and thus a separate MAC code and MAC key can be optionally omitted in some exemplary embodiments. In exemplary embodiments, the successful decryption of a ciphertext into a plaintext using decryption algorithms 405b supports mutual authentication between the two nodes, since successful decryption can only take place when the two nodes record the depicted private keys used for the depicted key exchanges 314 and 319.

Figure 4b

Figure 4b is a flow chart illustrating exemplary steps for conducting a key exchange using PKI keys in order to derive a shared secret key, and for using the derived shared secret key to encrypt and decrypt data, in accordance with exemplary embodiments. Exemplary steps for (i) an initiator 102' or a DPP server 103 (depending on the mode 306a) and (ii) a responder 102x to mutually derive keys can comprise (a) a key exchange 319a step for DPP server 103 or initiator 102' (depending on the mode 306a) to derive a symmetric encryption key k2 314c and (b) a key exchange 314b step for a responder 101x to derive the same symmetric encryption key k2 314c. Exemplary steps for a responder 101x to encrypt data and initiator 102' to decrypt data can comprise (i) an encryption 315b step for responder 101x to utilize the symmetric encryption key k2 314c to convert plaintext to ciphertext, and (ii) a decryption 320b step for either initiator 102' or DPP server 103 (depending on the mode 306a) to utilize the symmetric encryption key k2 314c to convert the ciphertext received from responder 101x into plaintext. The use of the steps for a key exchange 314a, a key exchange 319a, encryption 315b, and decryption 320b were also depicted and described in connection with Figure 3a above.

Note that steps in Figure 4b and the steps in Figure 4a can share some algorithms and values, and the descriptions for the algorithms and values in Figure 4a can be applicable for Figure 4b. For example, a same key exchange algorithm 401a depicted and described in Figure 4a can also be used in Figure 4b. The key exchange algorithm 401a can comprise an ECDH key exchange. The set of parameters 199a depicted and described in Figure 4a can also be used in Figure 4b. Further, a

Petitioner Samsung Ex-1004, 1113

symmetric encryption algorithm 405a and a symmetric decryption algorithm 405b can be the same in Figure 4a and Figure 4b. A set of parameters 199f for a symmetric encryption algorithm 405a and a symmetric decryption algorithm 405b can be the same in Figure 4a and Figure 4b

A responder 101x can conduct a key exchange step 314b. In a key exchange step 314b, the initiator ephemeral public key Pi 102a received in message 122 from Figure 3a can be input into an ECDH key exchange algorithm 401a along with the responder ephemeral private key pr 101f derived in a step 318 in Figure 3a in order to output a secret shared key N 411. The value for shared secret key N 411 can be recorded and also used in the derivation of a key ke 327a in a key exchange step 321 below in Figure 4c. For a step 314b, the value for the shared secret key N 411 can be input into a key derivation function 402b in order to derive a mutually shared symmetric key k2 314c. In exemplary embodiments which support DPP specification version 1.0, a key derivation function 402b can comprise the key derivation function for key k2 in point 2 of section 6.2.3 of DPP specification version 1.0.

Either an initiator 102' or a DPP server 103 can conduct a key exchange step 319a, depending on a mode 306a from a server database 103a. At step 319a, a received, validated responder ephemeral public key Pr 101e and a derived initiator ephemeral private key pi 102b can be input into an ECDH key exchange algorithm 401a in order to calculate a shared secret key N 411. Shared secret key N 411 can also be used by initiator 102' or DPP server 103 (depending on a mode 306a) in a step 327 below. DPP server 103 can input shared secret key N 411 into a key derivation function 402b in order to mutually derived the symmetric encryption key k2 314c. In exemplary embodiments which support DPP specification version 1.0, a key derivation function 402b can comprise the key derivation function for key k2 in section 6.2.4 of DPP specification version 1.0.

Responder 101x can conduct an encryption step 315b, where the use for an encryption step 315b are depicted and described in connection with Figure 3a above. Ciphertext 322a can comprise a ciphertext from an encryption step 322 below in Figure 4c. In other words, a responder 101x can conduct the encryption step 322 below and also key exchange step 321 below in Figure 4c before conducting the encryption step 315b. As depicted in Figure 4b, the ciphertext 322a can also be encrypted again for an encryption step 315b. The symmetric ciphering key for encryption step 315b can comprise symmetric key k2 314c from a key exchange step 314b in the paragraph above. The output from an encryption step 315b can be ciphertext 315c, as depicted in Figure 4b.

A decryption 320b step can be performed by either an initiator 102' or a DPP server 103, depending on a mode 306a recorded in a server database 103a as depicted in Figure 3b. A decryption 320b step converts the ciphertext 315c received in a message 123 from Figure 3a into plaintext 403b. Decryption 320b step can utilize a symmetric decryption algorithm 405b, which could comprise the same algorithm used in symmetric encryption algorithm 405a except the algorithm being used for decryption instead of encryption. Note that the same values are input into

Petitioner Samsung Ex-1004, 1114

symmetric decryption algorithm 405b as symmetric encryption algorithm 405a, such as symmetric encryption key k2 314c and parameters 199f in order to convert ciphertext 315c back into plaintext 403b. Initiator 102' or a DPP server 103 (depending on a mode 306a) can the read and process plaintext 403b after a decryption 320b step. In exemplary embodiments, the successful decryption of a ciphertext into a plaintext using decryption algorithms 405b supports mutual authentication between the two nodes, since successful decryption can only take place when the two nodes record the depicted private keys used for the depicted key exchange.

Figure 4c

Figure 4c is a flow chart illustrating exemplary steps for conducting a key exchange using PKI keys in order to derive a shared secret key, and for using the derived shared secret key to encrypt and decrypt data, in accordance with exemplary embodiments. Exemplary steps for (i) an initiator 102' or a DPP server 103 (depending on the mode 306a) and (ii) a responder 102x to mutually derive keys can comprise (a) a key exchange 327 step for DPP server 103 or initiator 102' (depending on the mode 306a) to derive a symmetric encryption key ke 327a and (b) a key exchange 321 step for a responder 101x to derive the same symmetric encryption key ke 327a. Note that in some exemplary embodiments, such a mode 306a depicted and described in connection with Figure 5a and Figure 5b below, (i) a key exchange algorithm 401c can be processed by a DPP server 103 a key derivation function 402c can be processed by an initiator 102'. In other words, for some embodiments the key exchange 327 step can be shared or distributed between a DPP server 103 and an initiator 102'.

Exemplary steps in Figure 4c for a responder 101x to encrypt data and initiator 102' to decrypt data can comprise (i) an encryption 322 step for responder 101x to utilize the symmetric encryption key ke 327a to convert plaintext to ciphertext, and (ii) a decryption 328 step for either initiator 102' or DPP server 103 (depending on the mode 306a) to utilize the symmetric encryption key ke 327a to convert the ciphertext received from responder 101x into plaintext. The use of the steps for a key exchange 321, a key exchange 327, encryption 322, and decryption 328 were also depicted and described in connection with Figure 3a above.

Note that steps in Figure 4c and the steps in Figure 4a can share some algorithms and values, and the descriptions for the algorithms and values in Figure 4a can be applicable for Figure 4c. For example, the key exchange algorithm 401c can comprise an ECDH key exchange. The set of parameters 199a depicted and described in Figure 4a can also be used in Figure 4c. Further, a symmetric encryption algorithm 405a and a symmetric decryption algorithm 405b can be the same in

Petitioner Samsung Ex-1004, 1115

Figure 4a and Figure 4c. A set of parameters 199f for a symmetric encryption algorithm 405a and a symmetric decryption algorithm 405b can be the same in Figure 4a and Figure 4c.

A responder 101x can conduct a key exchange step 321. In a key exchange step 321, (i) the initiator bootstrap public key Bi 104a received in message 126 from Figure 1c can be input into an ECDH key exchange algorithm 401c along with (ii) both the responder ephemeral private key pr 101f and the responder bootstrap private key br 101b. An ECDH key exchange algorithm 401c can use (pr 101f + br 101b) modulo q, as specified in DPP specification version 1.0 in point 2 of section 6.2.3, so algorithm 401c for responder 101x can be slightly different than algorithm 401a for responder 101x above in Figure 4a and Figure 4b. The output of key exchange algorithm 401c can be a shared secret key L 412.

The responder 101x can then conduct a key derivation function 402c in order to calculate a shared secret key ke 327a. Note that for a key derivation function 402c, the input includes all of key M 410 from a step 314, key N from a step 314b, and the derived key L 412 from step 321. As contemplated herein, a key derivation function 402a, 402b, and 402c can also use other data mutually shared between (i) responder 101x and (ii) initiator 102' or DPP server 103 (depending on a mode 306a). In exemplary embodiments which support DPP specification version 1.0, a key derivation function 402c can comprise the key derivation function for key ke 327a in point 2 of section 6.2.3 of DPP specification version 1.0.

Either an initiator 102' or a DPP server 103 can conduct a key exchange step 327, depending on a mode 306a from a server database 103a and transmitted to DPP server 103 in a message 311 as depicted in Figure 3a. At step 327, (i) a combination of a recorded responder bootstrap public key Br 101a and received responder ephemeral public key Pr 101e, and (ii) the recorded initiator bootstrap private key bi 104b can be input into an ECDH key exchange algorithm 401c in order to calculate a shared secret key L 412. Shared secret key L 412 can be input into a key derivation function 402c, along with shared secret key M 410 from a step 319 and shared secret key N 411 from a step 319a in order to mutually derive the symmetric encryption key ke 327a. In exemplary embodiments which support DPP specification version 1.0, a key derivation function 402c can comprise the key derivation function for key ke 327e in section 6.2.4 of DPP specification version 1.0.

Responder 101x can conduct an encryption step 322, where the use for an encryption step 322 are depicted and described in connection with Figure 3a above and also Figure 4b above. Plaintext 403c in a step 322 can comprise the responder authentication value 404, where value R-auth 404 can comprise a secure hash value with input including the initiator nonce 317a, the second random number R-nonce from a step 318 above in Figure 3a, and public PKI keys. The value for R-auth 404 in a step 322 (also described with step 322 in Figure 3a) can comprise the value in point 3 of section 6.2.3 of DPP specification version 1.0. The symmetric ciphering key for encryption step

Petitioner Samsung Ex-1004, 1116

322 can comprise symmetric key ke 327a from a key exchange step 321. The output from an encryption step 322 can be ciphertext 322a, as depicted in Figure 4c. Note that ciphertext 322a in a step 322 is also encrypted again in a step 315b above in Figure 4b. In other words, a responder 101x can conduct the encryption step 322 herein for Figure 4c (along with the key exchange 321 herein in Figure 4b) before conducting the encryption step 315b in Figure 4b above.

A decryption step 328 can be performed by either an initiator 102' or a DPP server 103, depending on a mode 306a recorded in a server database 103a as depicted in Figure 3b. A decryption 328 step converts the ciphertext 322a received in a message 123 (after decrypting a first layer of encryption of ciphertext 315b in a step 320b) from Figure 3a into plaintext 403c. Decryption 328 step can utilize a symmetric decryption algorithm 405b, which could comprise the same algorithm used in symmetric encryption algorithm 405a except the algorithm being used for decryption instead of encryption. Note that the same values are input into symmetric decryption algorithm 405b as symmetric encryption algorithm 405a in Figure 4c, such as symmetric encryption key ke 327a and parameters 199f in order to convert ciphertext 322a back into plaintext 403c. Initiator 102' or a DPP server 103 (depending on a mode 306a) can the read and process plaintext 403c after a decryption 328 step. In exemplary embodiments, the successful decryption of a ciphertext into a plaintext using decryption algorithms 405b supports mutual authentication between the two nodes, since successful decryption of ciphertext 322a can only take place when the two nodes record the depicted private keys used for the depicted key exchange.

As depicted and described in connection with Figure 3a, a responder 101x can also conduct both an encryption step 322 and a decryption step 328. The steps for responder 101x to conduct a decryption step 328 can comprise step 328b and step 328d as depicted and described in Figure 3a. When responder 101x conducts decryption step 328 using symmetric encryption key ke 327a, the ciphertext and plaintext will comprise different values than those depicted in Figure 4c. For example, in Figure 3a a responder 101x can use a decryption step 328b (e.g. step 328 but with different ciphertext than that depicted in Figure 4c) in order to decrypt the encrypted initiator authentication value 404a into a plaintext authentication value 404a. Also, in Figure 3a a responder 101x can use a decryption step 328d (e.g. step 328 but with different ciphertext 322f than that depicted in Figure 4c) in order to decrypt the encrypted network credentials 109.

Note that in some exemplary embodiments, the "plaintext" network credentials 109' (for the purposes of a step 328d in Figure 5a) can comprise network credentials 109' that can also be encrypted with an asymmetric ciphering algorithm such as Elgamal using a step 336 in Figure 5a below. In other words, network credentials 109' can comprise credentials that are "double encrypted" with both (i) a symmetric key 327a (which can be decrypted by responder 101x using a decryption step 328d, where decryption step 328d uses the depicted decryption 328 in Figure 4c), and (ii) an asymmetric ciphering algorithm such as Elgamal from a step 336, (which could be

Petitioner Samsung Ex-1004, 1117

decrypted by responder 101x using a step 337 from Figure 5a below and either (i) responder ephemeral private key pr 101f or (ii) responder bootstrap private key br 101b).

As depicted and described in connection with Figure 3a, a DPP server 103 or initiator 102' (in Figure 5a) can conduct both an encryption step 322 and a decryption step 328. The steps for a DPP server 103 or initiator 102' (depending on a mode 306a) to conduct an encryption step 322 can comprise step 322b and step 322e as depicted and described in Figure 3a. When DPP server 103 or initiator 102' conducts encryption step 322b or 322e using symmetric encryption key ke 327a, the ciphertext and plaintext will comprise different values than those depicted in Figure 4c. For example, in Figure 3a a DPP server 103 can use an encryption step 322b (e.g. step 322 but with different plaintext than that depicted in Figure 4c) in order to encrypt the initiator authentication value 404a. In Figure 5a, an initiator 102' can use an encryption step 322b (e.g. step 322 but with different plaintext than that depicted in Figure 4c) in order to encrypt the initiator authentication value 404a. In Figure 3a a DPP server 103 can use an encryption step 322e (e.g. step 322 but with different plaintext than that depicted in Figure 4c) in order to encrypt the network credentials 109. In Figure 5a, an initiator 102' can use an encryption step 322e (e.g. step 322 but with different plaintext than that depicted in Figure 4c) in order to encrypt the network credentials 109'.

Figure 5a

Figure 5a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device database, a DPP server, an initiator proxy, and a responder, in accordance with exemplary embodiments. Before starting steps and message flows depicted in Figure 5a, the initiator, DPP server, and device database depicted for system 500 in Figure 5a may previously complete exemplary message flows and steps depicted for a step 301 in Figure 3a above. System 500 can include a device database 104, DPP server 103, initiator proxy 102', and responder 101x operating in device 101. Initiator proxy 102' can communicate with DPP server 103 via IP network 113 using an access network 112, also as depicted in Figure 3a above. As contemplated herein, an "initiator proxy 102' " depicted in Figure 5a may also be referred to as initiator 102', and an initiator 102' and initiator proxy 102' may be equivalent. A difference between system 500 in Figure 5a and system 300 in Figure 3a can be that (i) responder bootstrap public key Br 101a, (ii) initiator bootstrap public key Bi 104a, (iii) initiator ephemeral public key Pi 102a, and (iv) initiator ephemeral private key pi 102b are recorded and operated within an initiator 102'. In exemplary embodiments depicted in Figure 5a the initiator bootstrap private key bi 104b can remain recorded in a DPP server 103 and not exposed to initiator 102'. The exemplary embodiment depicted in Figure 5a can comprise a second mode 306a of operation for initiator 102' and DPP server 103, as depicted in Figure 3b above.

Petitioner Samsung Ex-1004, 1118

At step 502, DPP server 103 can select a mode 306a for both initiator 102' and DPP server 103. Note that the mode 306a selected is not normally observable to a responder 101x, since the full and complete set of DPP 191 messages between responder 101x and initiator 102' can be transmitted and received. In other words, (a) the different distribution of PKI and shared secret keys in a server database 103a and initiator 102' for a first mode 306a in Figure 3a and a second mode 306a in Figure 5a (b) would not normally be observable to a responder 101x. The recording of PKI and shared secret keys in a server database is depicted in a server database 103a in Figure 3b, where the value "-" can mean that the key is recorded and/or operated by initiator 102'. Step 502 can also comprise DPP server 103 selecting an initiator bootstrap public key Bi 104a and a responder bootstrap public key Br 101a for device 101 using ID-token device 206.

In exemplary embodiments, a responder 101x for a device 101 could record a plurality of responder bootstrap public keys Br 101a, as depicted in Figure 2a above. Reasons could include the use of different keys Br 101a for different purposes, such as a first key Br 101a for a responder authentication 141 and a second key Br 101a for a mutual authentication 142. Or, a third key Br 101a could be recorded or mapped in a tag 101c, while a fourth key Br 101a in a responder 101x could only be recorded in a network 107. Responder 101x can designate the use of the fourth key Br 101a with a higher level of trust, with the following exemplary results (A) and (B). For (A), device 101 could provide additional privileges for a configuration step 106 for an initiator 102' or initiator 102* that uses the second key Br 101a (e.g. the first key Br 101a could be easier for potentially any initiator 102' to obtain if the first key Br 101a is recorded in a QR code for device 101). As one example of additional privileges, a set of credentials 109 delivered to device 101 using the fourth key Br 101a could allow the installation of a new certificate authority root certificate for device 101. As one example of normal or other privileges, a set of credentials 109 delivered to device 101 using the third key Br 101a could deny the installation of a new certificate authority root certificate for device 101 as well.

In summary for a step 502, (i) a device 101 with responder 101x can use a plurality of responder bootstrap public keys Br 101a (including for reasons other than simply use of different sets of cryptographic parameters 199a for the different keys), and (ii) a DPP server 103 can select the responder bootstrap public key Br 101a in a step 502. Also, although a step 502 is depicted in Figure 5a, a step 502 could be used in other Figures and embodiments herein, include embodiments depicted in Figure 3a and Figure 5b. In addition, for a step 502, DPP server 103 can use an initiator bootstrap keys sequence number 197 in order to select the appropriate initiator bootstrap public key Bi 104a. For example, if a pair of initiator bootstrap keys had be deprecated (such as depicted in Figure 6a below), then an updated pair of initiator bootstrap keys should be selected using a device table 104d and sequence number 197 (also selected with device ID 206).

Petitioner Samsung Ex-1004, 1119

DPP server 103 can then send initiator 102' a message 503, where message 503 can include an identity 206 for device 101 (since initiator 102' may communicate with a plurality of responders 101x over time), the selected responder bootstrap public key Br 101a from a step 502, the selected initiator bootstrap public key Bi 104a, and optionally a value for the MAC address of the responder 101x comprising MAC.R 101d. Message 503 can be sent via the secure session 302b setup during a step 301 depicted in Figure 3a above.

Initiator 102' can receive message 503 and process the data received in a step 504. A step 504 can include validating the received public keys from a message 503 using public key validation steps from FIPS publication SP 800-56A (revision 2) and subsequent and related versions. Initiator 102' can use a step 504 in order to verify the public PKI keys received in a message 503 are consistent or comply with a selected set of cryptographic parameters 199a. At step 505 the initiator 102' can determine the initiator capabilities for conducting a DPP 191 with device 101 and responder 101x. In the exemplary embodiment depicted in Figure 5a, initiator 102' can select and use the capabilities of the role as a configurator. In other embodiments, an initiator 102' could select and use the capabilities of the role an enrollee for a step 504. In addition, an initiator 102' can generate a random number for an initiator nonce I-nonce 317a in a step 504.

Initiator 102' can then conduct a step 313 to derive the initiator ephemeral public key Pi 102a and the initiator ephemeral private key pi 102b. A step 313 was previously depicted and described above in Figure 3a. Although DPP server 103 conducted the step 313 in Figure 3a, an initiator 102' can also conduct the step 313 in system 500 in order to derive the initiator ephemeral PKI key pairs. Further, subsequent steps depicted in Figure 5a that were previously depicted as conducted by a DPP server 103 can be conducted by an initiator 102' using the embodiment depicted in Figure 5a. For example, depicted subsequent steps such as steps 314, 315, 319a, 320b, 322b, 328, etc. can be conducted, performed, or processed by initiator 102' using a DPP application 102g in an exemplary system 500, which could also represent a different mode 306a than system 300 in Figure 3a.

Initiator 102' in step 314 can use (i) the derived initiator ephemeral private key pi 102b from step 313, and (ii) the received responder bootstrap public key Br 101a from a message 503 to conduct the first initiator key exchange step 314, which is also depicted in Figure 4a above. The output or results from a step 314 can comprise a mutually derived shared secret key k1 with responder 101x. An intermediate step within the key exchange step 314 can include calculating the value M 410 (which could also be a secret shared key). Initiator 102' in step 315 can then conduct the encryption step 315 depicted in Figure 4a, and also described in Figure 3a. The plaintext input into a symmetric encryption algorithm 405a can comprise the initiator nonce 317a and initiator capabilities from a step 504 above. The shared secret key k1 314a can be used with symmetric

Petitioner Samsung Ex-1004, 1120

ciphering algorithm 405a in a step 315 from Figure 4a. The ciphertext output from an encryption step 315 can comprise ciphertext 315a.

At step 316, initiator 102' can calculate secure hash values for initiator bootstrap public key Bi 104a and responder bootstrap public key Br 101a. The selected hash 199e could be received by initiator 102' in a set of selected parameters 199a from a message 311 in a step 301. In exemplary embodiments, initiator 102' records the full set of cryptographic parameters 199, and a message to select a set of cryptographic parameters 199a could then specify a hash 199e to use with the public keys. For example, a selected set of cryptographic parameters 199a from message 311 in a step 301 could be the value "A", which would then be used by initiator 102' to determine that hash 199e in a set of cryptographic parameters 199 would be "SHA-256", as depicted in Figure 2b. Or, a message 311 in a step 301 could specify the full set of selected cryptographic parameters 199, such as listing out the individual values of key length 199b, curve name 199c, hash 199e, etc. The hash values calculated by initiator 102' in a step 316 in Figure 5a are depicted in Figure 5a as "H(Bi)" and "H(Br)", respectively.

Initiator 102' can then conduct a step 316a to send responder 101x a message 122, where message 122 can include the hash values for (i) the initiator bootstrap public key Bi 104a and (ii) the responder bootstrap public key Br 101a, the derived initiator ephemeral public key Pi 102a, and ciphertext 315a, where ciphertext 315a is encrypted with key k1 and includes an initiator nonce 317a and the initiator capabilities. Sending DPP authentication request message 122 was also depicted and described in connection with Figure 1b, Figure 1c, and Figure 3a above.

In Figure 5a, responder 101x can receive message 122 and conduct the series of steps 325. The series of steps 325 for responder 101x were depicted and described in connection with Figure 3a above, and steps 325 can comprise the steps 318. 319. 320, 314b, 321, 322, and 315b. Responder 101x can process the message 122 using steps 325 and generate a DPP authentication response 123. The steps 325 for a responder 101x to process the received message 122 and generate the response message 123 can comprise section 6.2.2 of DPP specification version 1.0. Other possibilities exist as well without departing from the scope of the present invention. In general, responder 101x can use the recorded bootstrap private keys and the derived ephemeral private keys in order to conduct ECDH key exchanges. Responder 101x can also (i) derive a responder nonce 318b, (ii) decrypt ciphertext in message 122 and (iii) encrypt ciphertext in message 123 using derived secret shared keys from the ECDH key exchanges. Responder 101x can send initiator 102' the DPP authentication response 123 after conducting the steps 325.

Initiator 102' can receive the DPP authentication response 123 and process the message. Using broadcast messages over connection 192, initiator 102' can identify the message from responder 101x based on the hash values for the bootstrap public keys in response 123. For embodiments where message 122 and response 123 are sent as unicast messages, initiator 102' can

Petitioner Samsung Ex-1004, 1121

identify the message from responder 101x based on the MAC.R 101d in the response 123. Initiator 102' can then extract and record responder ephemeral public key Pr 101e (depicted for a response 123 in Figure 3a). Initiator 102' can also a key verification step on the received key Pr 101e according to FIPS publication SP 800-56A (revision 2).

Initiator 102' can then send DPP server 103 a message 506 using secure connection 302b via access network 112, where message 506 can include (i) an identity for device 101 such as ID-token.device 206 (or ID.device 101i), and (ii) the responder ephemeral public key Pr 101e. In an exemplary embodiment a message 506 can include the MAC address MAC.R 101d to also identify device 101 and the associated key Pr 101e. DPP server 103 can receive the message 506 and record the responder ephemeral public key Pr 101e along with the device identity in a server database 103a, as depicted in Figure 3b above.

DPP server 103 can then conduct an ECDH key exchange step 401c as depicted in Figure 4c in order to determine a secret shared key L 412. As depicted in Figure 5a and also Figure 4c, a step 401c can comprise DPP server 103 inputting (i) the recorded initiator bootstrap private key bi 104b and (ii) a combination of the received responder ephemeral public key Pr 101e and the recorded responder bootstrap public key Br 101a into key exchange algorithm 401c in order to (iii) calculate the secret shared key L 412. Secret shared key L 412 can also comprise a secret shared key L according to section 6.2.4 of DPP specification version 1.0.

As depicted in Figure 5a, an initiator 102' can then conduct a second key exchange step 319a, where the second key exchange step 319a was depicted and described above in Figure 4b and also Figure 3a. Although a DPP server 103 was depicted as conducting a key exchange step 319a in Figure 3a, for the embodiments depicted in Figure 5a, an initiator 102' can conduct the key exchange step 319a. At step 319a, initiator 102' can input the received, validated responder ephemeral public key Pr 101e and the derived initiator ephemeral private key pi 102b into an ECDH key exchange algorithm 401b in order to calculate a shared secret key N 411. Shared secret key N 411 can also be used by initiator 102' in a key derivation function 402c step below in Figure 5a. For a key exchange step 319a in Figure 5a, initiator 102' can input shared secret key N 411 into a key derivation function 402b in order to mutually derived the symmetric encryption key k2 314c. Note that secret shared key N 411 can also be used by initiator 102' for later use in a key derivation function step 402c, as depicted in Figure 5a.

As depicted in Figure 5a, initiator 102' can then conduct a decryption step 320b on received ciphertext 315c using the derived symmetric encryption key k2 314c from a step 319a above by initiator 102' in Figure 5a. The use and operation of a decryption step 320b was also depicted and described above in connection with Figure 4b and Figure 3a. Initiator 102' can read the plaintext 403b values from decryption step 320b. The plaintext values 403b can include the initiator nonce 317a from a message 122 above. The plaintext values 403b can also include a second ciphertext

Petitioner Samsung Ex-1004, 1122

322a (e.g. ciphertext 322a can be inside ciphertext 315c). Initiator 102' verifies that the initiator nonce 317a received in message 123 and decrypted in step 320b is equal to the initiator nonce 317a value sent in message 122 above. If the two initiator nonce 317a values are not equal, then initiator 102' can abort the DPP procedure and subsequent steps.

Initiator 102' can then receive a message 510 from DPP server 103, where message 510 can include an identity for device 101 such as ID-token.device 206 and the value L 412, where the value or secret shared key L 412 was calculated by DPP server 103 in a step 401c as depicted in Figure 5a. The value L 412 can be calculated by DPP server 103 using the initiator bootstrap private key bi 104b, and thus initiator 102' does not need to record or receive the initiator bootstrap private key bi 104b for embodiments depicted in Figure 5a.

Initiator 102' can then conduct a key derivation function step 402c using (i) the received value L 412 from message 510, (ii) the calculated shared secret key M 410 from a step 314, and (iii) the calculated shared secret key N 411 from a step 319a. A key derivation function step 402c is depicted and described in connection with Figure 4c above. The output from a key derivation function step 402c conducted by initiator 102' can be a shared secret key ke 327a, as depicted in Figure 4c.

At step 328, initiator 102' can conduct a decryption step 328 on the ciphertext 322a using the key ke 327a from the previous step 402c. A decryption step 328 is depicted and described in connection with Figure 4c above. Initiator 102' can use the decryption step 328 in order to read the plaintext value of R-auth 404 from the responder 101x in message 123. R-auth can comprise the value R-auth 404 from Figure 3a and Figure 4c and also comprise a hash 199e value that includes the combination of at least the initiator nonce 317a and the responder nonce 318b. At step 328, initiator 102' can then also internally calculate the same value for R-auth 404, and if the internally calculated value for R-auth 404 and the received, decrypted value for R-auth 404 are equal, then the responder 101x can be considered authenticated. The calculation of R-auth 404 by initiator 102' in a step 328 can be according to section 6.2.3 of DPP specification version 1.0.

Initiator 102' can then conduct a step 322b, where step 322b can comprise calculating an initiator authentication value I-auth 404a. Initiator authentication value 404a can be similar to responder authentication value R-auth 404, except the values of initiator nonce 317a can be sequenced before a responder nonce 318b inside a hash 199e calculation over the two nonces. Initiator 102' can then encrypt the initiator authentication value 404a using an encryption step 322 using key ke 327a, as depicted in Figure 4c above. In other words, an encryption step 322 in Figure 4c depicts the encryption of R-auth 404, and encryption step 322b can comprise initiator 102' encrypting the initiator authentication value I-auth 404a using the key ke 327a derived in a step 402c.

Petitioner Samsung Ex-1004, 1123

Initiator 102' can previously record the hash 199e values for the initiator and responder bootstrap public keys. Initiator 102 can take the recorded hash 199e values for the initiator and responder bootstrap public keys and append to them the encrypted value for I-auth 404a calculated by initiator 102 in step 322b. Initiator 102' can send responder 101x a DPP authentication confirm message 123a, as depicted in Figure 5a. The data and steps to determine and calculate a DPP authentication confirm message 123a can be according to section 6.2.4 of DPP specification version 1.0. As depicted in Figure 5a, the collection of steps and messages after a message 506 through transmitting a message 123a can collectively comprise a step 512.

Note that although the DPP authentication confirm message 123a can be equivalent to the DPP authentication confirm message from section 6.2.4 of DPP specification version 1.0, an important difference can be the operation of initiator 102' in Figure 5a versus an initiator as contemplated by DPP specification version 1.0. In the present invention as depicted in Figure 5a, initiator 102' does not need to record the initiator bootstrap private key bi 104b. So, using the technology depicted in Figure 5a, an initiator 102' can (i) send and receive messages that are fully compatible with a DPP protocol 191, but (ii) DPP server 103 can record key bi 104b. In other words, a responder 101x may not "know" that an initiator operates as either (i) an initiator 102* from Figure 1b or Figure 1c, or an initiator 102' from Figure 5a since the messages transmitted and received by the responder 101x can be fully compatible with a DPP 191.

The technology depicted in Figure 5a can enhance security since an initiator bootstrap private key bi 104b does not need to be shared with an initiator 102'. The initiator 102' can also not feasibly be able to determine the initiator bootstrap private key bi 104b using value L 412. By only recording the initiator bootstrap private key bi 104b in DPP server 103 (or also a device database 104), then a common, shared initiator bootstrap public key Bi 104a can be more securely shared across multiple devices 101, as contemplated by key table 104k in Figure 2b (with the depiction of "shared keys"). In other words, a manufacturer of device 101 could record a common initiator bootstrap public key Bi 104a across a plurality of devices 101, since a network 107 could share the initiator bootstrap public key Bi 104a with a device manufacturer of devices 101 (or a device distributor) before the devices are deployed near network access points 105. In other embodiments besides those depicted in Figure 5a, the initiator bootstrap private key bi 104b can be shared with initiator 102'.

In Figure 5a, responder 101x can receive message 123a and conduct the series of steps 339. The series of steps 339 for responder 101x were depicted and described in connection with Figure 3a above, and steps 339 can comprise the steps 328b, 331, 332, and 322c. Responder 101x can process the message 123a using steps 339 and generate a DPP configuration request 123b. In summary, responder 101x can decrypt the ciphertext in message 123a, where the ciphertext can include the initiator authentication value 404a. Responder 101x can also internally calculate a value for the

Petitioner Samsung Ex-1004, 1124

initiator authentication value 404a and the received value 404a, and if the two values are equal then responder 101x can consider the initiator 102' to be authorized. The responder 101x can take the roll of an enrollee and process a configuration attribute 330. The responder 101x can generate a random number comprising an enrollee nonce E-nonce 332a. Responder 101x can then conduct an encryption step 322c, where encryption step 322c can comprise (i) an encryption step 322 using key ke 327a, and (ii) the plaintext to be encrypted can be the E-nonce 332a and the configuration attribute 330. The output of an encryption step 322c can be a ciphertext 322d. Responder 101x can then send initiator 102' a DPP Configuration Request message 123b, where the configuration request 123b can include the ciphertext 322d, where the plaintext in ciphertext 322d can include E-nonce 332a and the configuration object 330. DPP Configuration Request message 123b can be processed by responder 101x according to section 6.3.3 of DPP specification version 1.0.

Initiator 102' can receive DPP Configuration Request message 123b via connection 192 and using DPP application 102g. As contemplated herein, a DPP application 102g can perform the steps for initiator 102' depicted in Figure 5a, such as transmitting/receiving DPP 191 messages using a first MAC.I 102n for connection 192 and transmitting/receiving messages with DPP server 103 using MAC.N 102q via access network 112. A DPP application 102g in initiator 102' can also perform the depicted steps for an initiator 102' in Figure 5a such as, but not limited to, 504, 505, 313, 314, etc. Initiator 102' can receive message 123b and process the message. Initiator 102' can conduct a decryption step 328c using decryption step 328 depicted in Figure 4c above. A difference between decryption step 328 and decryption step 328c is that decryption step 328c uses the ciphertext 332d as input into the symmetric decryption algorithm 405b. Initiator 102' can use the previously calculated key ke 327a from a step 402c in Figure 5a in order to read the plaintext within ciphertext 332d. Initiator 102' can record the plaintext E-nonce 332a and configuration attribute 330.

Initiator 102 can send DPP server 103 a message 513 via secure connection 302b from a step 301, where message 513 can include and identity for device 101 such as ID-token.device 206 and the plaintext value for the configuration attribute 330. Message 513 can be sent via secure session 302b, so the "plaintext" would be encrypted when transmitted over secure session 302b. In step 335, DPP server 103 can process the received configuration attribute 330. For the exemplary embodiment depicted in Figure 1a and Figure 5a, configuration attribute 330 can specify that device 101 with responder 101x may have a role of WiFi client. In exemplary embodiments, DPP server 103 can previously select a primary access network 308aa in a step 308a within a step 301 depicted in Figure 3a above, using the received networks available list 304c from initiator 102'. Also, in a step 308b in a step 301 in Figure 3a above, DPP server 103 could receive or record the set of network credentials 109 associated with the selected primary access network 308aa for device 101 using device identity ID-token.device 206 (or ID.device 101i).

Petitioner Samsung Ex-1004, 1125

In step 335, DPP server 103 can process the set of network credentials 109 for primary access network 308aa into a configuration object. The configuration object with network credentials 109 can comprise a connector or configuration object as specified in section 6.3.5 and 6.3.6 of DPP specification version 1.0. The configuration object with network credentials 109 can be optionally signed by DPP server 103 using the recorded initiator bootstrap private key bi 104b. In a different embodiment, a DPP server 103 can omit a signature for network credentials 109 in a step 335, and an initiator 102' can sign the network credentials 109 using the derived initiator ephemeral private key pi 102b. The steps to create and verify a signature with ECC PKI keys can be according to FIPS publication 186-4, and other possibilities exist as well. In Figure 5a, a DPP server 103 can also take the other steps described for a step 335 in Figure 3a above.

In a step 336, DPP server 103 can optionally encrypt network credentials 109 using an asymmetric encryption algorithm and a public key from the responder comprising one of (i) the responder bootstrap public key Br 101a or (ii) the derived responder ephemeral public key Pr 101e (received in message 506), or (iii) the public key for a device recorded by a certificate for device 101 in cert.device 101m. In some preferred exemplary embodiments, a step 336 can comprise encrypting network credentials 109 into credentials 109' using (i) the public key in cert.device 101m, and the elliptic curve Elgamal asymmetric encryption algorithm. The encryption algorithm could be from using Elgamal asymmetric encryption as summarized in the Wikipedia article for ElGamal encryption from July 10, 2018, which is herein incorporated by reference. A set of network credentials 109 that may be optionally encrypted using Elgamal asymmetric encryption and a responder public key is depicted as credentials 109' in Figure 5a. The optional Elgamal asymmetric encryption in a step 336 can use a set of parameters 199a in order to specify values such as encoding rules, byte or bit order, padding, etc. An optional encryption step 336 could also comprise (i) deriving a symmetric key and encrypting the symmetric key with Elgamal encryption and a responder public key and then (ii) encrypting the set of credentials 109 using a symmetric ciphering algorithm 405a and the derived symmetric key, in order to determine an encrypted set of credentials 109'.

In exemplary embodiments, an encrypted set of credentials 109' can be sent to initiator 102' instead of plainext credentials 109, since an initiator 102' can read plaintext credentials 109, but not feasibly read encrypted credentials 109'. Note that encrypted credentials 109' can include a plaintext public key "h" or "g^x" along with the credentials 109' in order for responder 101x to use a private key such as responder bootstrap private key br 101b or responder ephemeral private key pr 101f to decrypt the encrypted credentials 109'. A reason that a step 336 can be performed in Figure 5a is that (i) initiator 102' could otherwise read and operate on a plaintext network credentials 109 (since initiator 102' uses encryption key ke 327a to encrypt plaintext 403c), and (ii) a DPP server 103 or a network 107 may prefer for initiator 102' to not read plaintext network credentials 109

Petitioner Samsung Ex-1004, 1126

(since initiator 102' may not be secured or under the full control of a network 107). In other embodiments, an encryption step 336 can be optionally omitted and plaintext credentials 109 can be sent to an initiator 102' via the secure session 302b via message 514.

In another embodiment, the asymmetric encryption step 336 can be performed by an external network, such that the external network generates the asymmetrically encrypted credentials 109', and the asymmetrically encrypted credentials 109' could be stored in a device credentials table 340 from Figure 3b. For example, credentials 109' could comprise the exemplary depicted credentials in table 340 of "from ...", where "..." was an external network in a networks available list 304c. In other words, a DPP server 103 may not be able to read the plaintext credentials 109 within an asymmetrically encrypted credentials 109', since an external network could encrypt the credentials using a step 336 before providing the credentials to network 107 and DPP server 103.

DPP server 103 can then send initiator 102' a message 514, where the message 514 can include an identity for device 101 such as ID-token.device 206 and the network credentials 109', where the network credentials 109' can be in the form of a configuration object and optionally encrypted using an Elgamal asymmetric encryption algorithm. Although a single value for network credentials 109' in message 514 are depicted, network credentials 109' could provide credentials for device 101 to connect with both a selected primary network 308aa and a selected backup network 308ab (with different credentials for each). Or, two sets of configuration objects with two sets of network credentials 109' could be sent in a message 514. Other possibilities exist as well without departing from the scope of the present invention.

Initiator 102' can receive message 514 and process the received credentials. At step 322e, initiator 102 can conduct an encryption step 322e using an encryption step 322 depicted and described in connection with Figure 4c above. Encryption step 322e can be performed using key ke 327a for device 101. For encryption step 322e, the plaintext value for the E-nonce 332a and the configuration object comprising network credentials 109' (where network credentials 109' may be optionally encrypted) can be input into a symmetric encryption algorithm 405a in order to generate a ciphertext 322f. Note that the use of asymmetrically encrypted credentials 109' can be optionally omitted, and an initiator 102' in a step 322e can input plaintext credentials 109 into a symmetric ciphering algorithm 405a.

Responder 101x can receive message 124 from initiator 102' with ciphertext 322f. Responder 101x can conduct a decryption step 328d using decryption step 328 depicted in Figure 4c below. A difference between decryption step 328 and decryption step 328d is that decryption step 328d uses the ciphertext 322f as input into the symmetric decryption algorithm 405b. Responder 101x can use the previously calculated key ke 327a in order to read the plaintext within ciphertext 322f. Note that the plaintext within ciphertext 322f can include an encrypted network credentials

Petitioner Samsung Ex-1004, 1127

109'. Responder 101x can confirm that plaintext in ciphertext 322f includes the previously sent E-nonce 332a.

For embodiments that use an optional encryption step 336 on network credentials 109 by DPP server 103 in step 336 (or an external network) in order to create encrypted network credentials 109', a responder 101x can use an asymmetric decryption step 337 in order to convert the encrypted network credentials 109' into plaintext network credentials 109. Responder 101x could use an Elgamal asymmetric decryption algorithm and either a recorded (i) responder bootstrap private key bi 101b or (ii) responder ephemeral private key pr 101f. Note that device 101 could decrypt credentials 109' using a private key for the public key in a certificate cert.device 101m, and in this case responder 101x could pass the received asymmetrically encrypted credentials 109' to device 101. Note that when using elliptic curve Elgamal asymmetric decryption, the responder 101x or device 101 uses the private key that corresponds to the public key used with asymmetric encryption of the credentials 109 in a step 336. Responder 101x could also use a selected set of cryptographic parameters 199a in order to conduct the Elgamal asymmetric decryption.

Note that encrypted network credentials 109' could include the public key "h" or "g^x" in order to conduct the Elgamal asymmetric decryption. For embodiments (a) without an optional encryption step 336 on network credentials 109 to create encrypted credentials 109', then (b) responder 101x can read the plaintext network credentials 109 in a step 337. Note that credentials 109 may be previously signed by DPP server 103 or initiator 102' using a private key for the initiator 102'. The responder could also verify a signature for the credentials 109 using a public key for the initiator, such as the initiator ephemeral public key Pi 102a or the initiator bootstrap public key Bi 104a.

After reading the plaintext network credentials 109 and optionally verifying a signature using a recorded public key 102a or 104a, device 101 can complete the configuration step 106 by loading the network credentials 109 and begin operating a WiFi radio 101h with the network credentials 109, as depicted in Figure 1a. In this manner, device 101 can obtain connectivity to network access point 105. In exemplary embodiments, the initiator 102' can conduct the initiator restore step 129 depicted and described in connection with Figure 1d above, such that initiator 102' can be returned to the previous state where WiFi radio 102w operates with a user configuration 131, thus minimizing impacts of the prior initiator configuration step 127 for an initiator user 102u. As depicted in Figure 5a, the collection of steps beginning with step 339 for responder 101x through step 106 for responder 101x can comprise a step 513.

Figure 5b

Petitioner Samsung Ex-1004, 1128

Figure 5b is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device database, a DPP server, an initiator proxy, and a responder, in accordance with exemplary embodiments. Before starting steps and message flows depicted in Figure 5b, the initiator, DPP server, and device database depicted for system 501 in Figure 5b may previously complete exemplary message flows and steps depicted for a step 301 in Figure 3a above. System 501 can include a device database 104, DPP server 103, initiator proxy 102', and responder 101x operating in device 101. Initiator proxy 102' can communicate with DPP server 103 via IP network 113 using an access network 112, also as depicted in Figure 3a above. As contemplated herein, an "initiator proxy 102' " depicted in Figure 5b may also be referred to as initiator 102', and an initiator 102' and initiator proxy 102' may be equivalent herein. A difference between system 501 in Figure 5b and system 300 in Figure 3a can be that the exemplary secret shared keys comprising key k2 314c and ke 327a are recorded and operated within an initiator 102'. In other words, a DPP server 103 does not need access to these keys for the embodiment depicted in Figure 5b. In exemplary embodiments depicted in Figure 5b, both (i) the initiator bootstrap private key bi 104b and (ii) the responder bootstrap public key Br 101a can remain recorded in a DPP server 103 and not exposed to initiator 102'. The exemplary embodiment depicted in Figure 5ba can comprise a third mode 306a of operation for initiator 102' and DPP server 103, as depicted in Figure 3b above.

After a step 301 from Figure 3a, DPP server 103 can conduct a step 502 in order to select a responder bootstrap public key Br 101a for device 101 using an identity ID-token.device 206. The value ID-token.device 206 could be received within step 301 in message 305 as depicted in Figure 3a. The responder bootstrap public key Br 101a could be recorded in a device database 104 or received from a device manufacturer for a device 101. Other possibilities exist as well for a network 107 to have access to a responder boostrap public key 101a without departing from the scope of the invention. In some embodiments in order for a DPP server 103 to record a responder bootstrap public key Br 101a, an initiator 102' could conduct a PKEX key exchange in band via WiFi link 192 using PKEX key 198, where initiator 102' receives the key 198 in a message 311 in a step 301 in Figure 3a. Initiator 102' could (i) conduct the PKEX using key 198 with responder 101x during a step 303b, and then send the received responder bootstrap public key Br 101a to DPP server 103. Other embodiments depicted in Figure 3a and Figure 5a also could optionally use a PKEX with key 198 in order to receive the key Br 101a "in band" via WiFi link 192, such as if device database 104 does not record key Br 101a before a step 301 in Figure 3a. As mentioned for a step 502 in Figure 5a, a responder 101x could record a plurality of responder bootstrap public keys Br 101a, and the key 101a can be selected in a step 502. A MAC address for MAC.R 101d for device 101 using ID-token.device 206 can also be selected from a server database 103a in a step 502.

Petitioner Samsung Ex-1004, 1129

Initiator 102' can then conduct a step 313 to derive the initiator ephemeral public key Pi 102a and the initiator ephemeral private key pi 102b. A step 313 was previously depicted and described above in Figure 3a. Although DPP server 103 conducted the step 313 in Figure 3a, an initiator 102' can also conduct the step 313 in order to derive the ephemeral PKI key pairs. For example, initiator 102' also derives the initiator ephemeral PKI keys in a step 313 in Figure 5a. Initiator 102' can then send DPP server 103 the derived initiator ephemeral private key pi 102b in a message 521. Message 521 could be securely send to DPP server 103 via secure session 302b. Message 521 can also include device identity ID-token.device 206 (or ID.device 101i) in order for DPP server 103 to track the device 101 and initiator 102' associated with the key pi 102b. DPP server 103 can record the data received in message 521 in a server database 103a, as depicted in Figure 3b.

DPP server 103 for a step 314z can comprise DPP server 103 conducting a first initiator key exchange step 314 using an Elliptic Curve Diffie Hellman (ECDH) key exchange 401a, as depicted and described in connection with Figure 4a below. Data input for a step 314z can comprise the initiator ephemeral private key pi 102b and the recorded responder bootstrap public key Br 101a. Data output from a step 314z can comprise the secret shared key M 410 as described in Figure 4a above. In other words, the difference between a step 314z and a step 314 is the step 314z can terminate before the key derivation function 401a and the output of a step 314z can comprise the secret shared key M 410. In an alternative embodiment for system 501 in Figure 5b, DPP server 103 can also calculate key k1 using the full set of steps for a key exchange 314 (but key k1 may not be needed by DPP server 103 for the embodiment depicted in Figure 5b).

At step 316, DPP server 103 can calculate secure hash values for initiator bootstrap public key Bi 104a and responder bootstrap public key Br 101a, using a secure hash algorithm 199e. The values are depicted in Figure 3a as "H(Bi)" and "H(Br)", respectively. DPP server 103 can then send initiator 102' a message 522, where message 522 can include an identity for device 101 ID-token.device 206 (or ID.device 101i), the hash values for the initiator bootstrap public key Bi 104a and the responder bootstrap public key Br 101a, the shared secret key M 410 from a step 401a in key exchange step 314z, and the MAC address for the responder 101x comprising MAC.R 101d. Message 522 can be sent through secure session 302b and received and recorded by initiator 102'.

Initiator 102' can receive message 522 and conduct a series of steps in order to create and process a DPP authentication request message 123. Initiator 102' can input the received value M 410 from a message 522 into a key derivation function 402a in order to derive a secret shared key k1 314a, depicted as a step 402a for initiator 102' in Figure 5b. At step 504 the initiator 102' can determine the initiator capabilities for conducting a DPP 191 with device 101 and responder 101x. In the exemplary embodiment depicted in Figure 5b, initiator 102' can select and use the capabilities of the role as a configurator. In other embodiments, an initiator 102' could select and use the

Petitioner Samsung Ex-1004, 1130

capabilities of the role an enrollee for a step 504. In addition, an initiator 102' can generate a random number for an initiator nonce I-nonce 317a in a step 504. A step 504 can also take place before a step 402a in some exemplary embodiments.

Initiator 102' in step 315 can then conduct the encryption step 315 depicted in Figure 4a, and also described in Figure 3a and Figure 5a. The plaintext input into a symmetric encryption algorithm 405a can comprise the initiator nonce 317a and initiator capabilities from a step 504 above. The shared secret key k1 314a can be used with symmetric ciphering algorithm 405a in a step 315. The ciphertext output from an encryption step 315 can comprise ciphertext 315a.

Initiator 102' can then conduct a step 316a to send responder 101x a message 122, where message 122 can include the hash values for (i) the initiator bootstrap public key Bi 104a and (ii) the responder bootstrap public key Br 101a, the derived initiator ephemeral public key Pi 102a (from a step 313 above), and ciphertext 315a, where ciphertext 315a is encrypted with key k1 and includes an initiator nonce 317a and the initiator capabilities. Sending DPP authentication request message 122 was also depicted and described in connection with Figure 1b, Figure 1c, Figure 3a, and Figure 5a above.

In Figure 5b, responder 101x can receive message 122 and conduct the series of steps 325. The series of steps 325 for responder 101x were depicted and described in connection with Figure 3a above, and steps 325 can comprise the steps 318. 319. 320, 321, 314b, 322, and 315b. Responder 101x can process the message 122 using steps 325 and generate a DPP authentication response 123. The steps 325 for a responder 101x to process the received message 122 and generate the response message 123 can comprise section 6.2.2 of DPP specification version 1.0. Other possibilities exist as well without departing from the scope of the present invention. In general, responder 101x can use the recorded bootstrap private keys and the derived ephemeral private keys in order to conduct ECDH key exchanges. Responder 101x can also (i) derive a responder nonce 318b, (ii) decrypt ciphertext in message 122 and (iii) encrypt ciphertext in message 123 using derived secret shared keys from the ECDH key exchanges. Responder 101x can send initiator 102' the DPP authentication response 123 after conducting the steps 325.

Initiator 102' can receive the DPP authentication response 123 and process the message. Using broadcast messages over connection 192, initiator 102' can identify the message from responder 101x based on the hash values for the bootstrap public keys in response 123. For embodiments where message 122 and response 123 are sent as unicast messages, initiator 102' can identify the message from responder 101x based on the MAC.R 101d in the response 123. Initiator 102' can then extract and record responder ephemeral public key Pr 101e (depicted for a response 123 in Figure 3a). Initiator 102' can also a key verification step on the received key Pr 101e according to FIPS publication SP 800-56A (revision 2).

Petitioner Samsung Ex-1004, 1131

Initiator 102' can then send DPP server 103 a message 506 using secure connection 302b via access network 112, where message 506 can include (i) an identity for device 101 such as ID-token.device 206 (or ID.device 101i), and (ii) the responder ephemeral public key Pr 101e. In an exemplary embodiment a message 506 can include the MAC address MAC.R 101d to also identify device 101 and the associated key Pr 101e. DPP server 103 can receive the message 506 and record the responder ephemeral public key Pr 101e along with the device identity in a server database 103a, as depicted in Figure 3b above.

As depicted in Figure 5b, initiator 102', DPP server 103, and responder 101x can then conduct a step 512. A step 512 can comprise the series of steps and message depicted and described in connection with Figure 5a above. In summary, initiator 102' can conduct a key exchange step 319a to derive the key k2 314c and then decrypt the ciphertext from message 123 into plaintext using a step 320b. Key exchange step 319a can also calculate a shared secret key N 411 as depicted in Figure 4b. The plaintext from a step 320b can comprise a second ciphertext 322a encrypted with a key ke 327a. DPP server 103 can (i) conduct a key exchange step 401c in order to derive shared secret key L 412. DPP server 103 can send initiator 102' the shared secret key L 412. Initiator 102' can conduct a key derivation function step 402c from Figure 4c using the values M 410 (from message 522), N 411 (from a step 319a), and L 412 (from DPP server 103 in a message 510 in a step 512) in order to derive key ke 327a, as depicted in Figure 4c. Initiator 102' can decrypt the second ciphertext 322a in message 123 into plaintext with a decryption step 328 using the derived key ke 327a and read the value R-auth 404. Initiator 102' can also use a step 328 to compare the received, decrypted value for R-auth 404 with an internally calculated value for R-auth, and responder 101x can be authenticated if the two values for R-auth 404 are equal. Initiator 102' can calculate a value for I-auth 404a and encrypt the value with key ke 327a in a step 322b. Initiator 102' can then send responder 101x a DPP authentication confirm message 123a in a step 512, as depicted in Figure 5a and Figure 5b.

As depicted in Figure 5b, initiator 102', DPP server 103, and responder 101x can then conduct a step 513. A step 513 can comprise the series of steps and message depicted and described in connection with Figure 5a above. In summary, responder 101x can use a step 339 to receive and process the DPP authentication confirm message 123a. Responder 101 can decrypt the I-auth value 404a in a step 331. Responder 101x can calculate the value for I-auth 404, and if the received and calculated values for I-auth 404a are equal, then initiator 102' can be authenticated with responder 101x. Responder 101x can send a DPP configuration request 123b message, which can include a ciphertext 322d an E-nonce 332a and a configuration attribute 330. Initiator 102' can decrypt the ciphertext 322d and send DPP server 103 the configuration attribute 330. DPP server 103 can send initiator 102' a set of network credentials 109. The set of credentials 109 may optionally be encrypted with an asymmetric encryption algorithm using a public key for the responder, and for

Petitioner Samsung Ex-1004, 1132

these embodiments the set of credentials 109 can be designated an encrypted set of credentials 109'. The public key for the responder with an optional asymmetric encryption for the credentials 109' could be (i) the responder bootstrap public key Br 101a, or (ii) the responder ephemeral public key, or (iii) a private key for the device public key in cert.device 101m (e.g. step 337 in Figure 5a). Credentials 109 or credentials 109' can also be signed by either (i) the DPP server 103 using the recorded initiator bootstrap private key bi 104b, or (ii) using the derived initiator ephemeral private key pi 102b.

Continuing with a step 513 in Figure 5b, the credentials 109 or credentials 109' could be signed by both the DPP server 103 and the initiator 102 in a step 513. The initiator 102' can receive the signed network credentials 109' or 109 and encrypt the credentials and the E-nonce 332a using the derived key ke 327a. The initiator 102' can send responder 101x a DPP configuration response 124 message with the encrypted and signed credentials for device 101. Responder 101 can verify the signature and decrypt the credentials 109 or asymmetrically encrypted credentials 109'. Device 101 can apply the credentials 109 and connect with another node using the credentials 109, and the other node could be network access point 105. The other node could also be a WiFi peer-to-peer (P2P) connection. In another embodiment, device 101 could operate as an access point 105 for other WiFi clients, and in this embodiment the credentials 109 received in a message 124 could be for device 101 to operate as a WiFi access point using the credentials 109. Other possibilities exist as well for a responder 101x to receive a set of credentials 109 from an initiator 102' without departing from the scope of the present invention.

System 501 as depicted in Figure 5b can provide several benefits for conducting a device provisioning protocol, compared to the conventional technology depicted in Figure 1b and Figure 1c. First, an initiator bootstrap private key bi 104b can remain recorded in a network 107 and does not need to be shared with an initiator 102'. The operating environment and security for an initiator 102' or initiator 102* in Figure 1b and Figure 1c may be outside the control of a network 107 or a device 101 with responder 101x. If an initiator bootstrap private key bi 104b is sent to initiator 102' or initiator 102*, then the initiator bootstrap private key bi 104b cannot be securely reused (since initiator 102' or initiator 102* could then share the key bi 104b or attempt to reuse it with other devices 101 recording the same initiator bootstrap public key Bi 104a). Further, recording the same, shared initiator bootstrap public key Bi 104a across a plurality of devices 101 (such as during device manufacturing or before distribution) can provide benefits for scalability and ease of deployment, especially if the initiators 102' or initiators 102* are not known to a network 107 before distribution of devices 101.

In other words, recording a same initiator bootstrap public key Bi 104a across a plurality of devices may not be securely feasible if the initiator bootstrap private key bi 104b needs to be used by multiple, insecure initiators 102' or initiators 102*. In addition, the responder bootstrap public key

Petitioner Samsung Ex-1004, 1133

Br 101a can remain securely recorded in a network 107 and not exposed to an initiator 102' for a system 501 in Figure 5b (where Br 101a can be recorded in a device database 104 instead of being received by an initiator 102' from a message 503 in Figure 5a). During the lifetime of a device 101 that could be a decade or longer, a device owner or device user may prefer the device 101 to conduct a device provisioning protocol multiple times, such as after a "factory reset" for device 101. Keeping or storing the responder bootstrap public key Br 101a and initiator bootstrap private key bi 102b in a network 107 can support the steps for a device provisioning protocol to be securely reused for the same device 101.

Yet another benefit of a system 300, system 500, and system 501 is that an overall trust level by responder 101x for conducting a device provisioning protocol can be enhanced or increased, since both an initiator 102' and a DPP server 103 must work in conjunction in order to successfully compete a DPP 191. In other words, for a system 300, system 500, and system 501, security risks for an insecure initiator 102' can be compensated by likely a higher overall level of security for a DPP server 103. These are just a few of the exemplary benefits for systems 300, 500, and 501 compared to conventional technology for conducting a device provisioning protocol, and other benefits will be available as well for those of ordinary skill in the art.

Figure 6a

Figure 6a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device database, a DPP server, a WiFi access point, and a device, in accordance with exemplary embodiments. Before starting steps and message flows depicted in Figure 6a, a device 101 with responder 101x could have completed the exemplary configuration step 106 as depicted in Figures 1a, 3a, 5a, and 5b above in order for device 101 to receive and load a set of network access credentials 109. System 600 can include a device database 104, a DPP server 103, a WiFi access point 105, and a device 101. WiFi access point 105 can broadcast an SSID with a value of SSID.network-AP 109a, which also could have been observed by an initiator 102' in a step 303a to collect a networks available list 304c. A DPP server 103 (or another server within a network 107) could (i) conduct a step 308a to select WiFi access point 105 from the networks available list 304c and then (ii) obtain the credentials 109 using a step 308b, and then send the credentials 109 through initiator 102' in a DPP message 124.

Configuration data for device 101 to utilize with WiFi access point 105 could be included in data and values for network credentials 109 for device 101. For embodiments where a selected access network 308aa for device 101 is not a WiFi access point 105 (such as possibly a 5G wireless WAN connection), then network credentials 109 could comprise a profile for an embedded universal integrated circuit card (eUICC) as specified in GSMA standards SGP.22 and SGP.02. For

Petitioner Samsung Ex-1004, 1134

embodiments where credentials 109 comprise an eUICC profile, then the credentials 109 may preferably be asymmetrically encrypted into credentials 109' using a step 336 from Figure 5a by an external network recording the eUICC profile, and then asymmetrically decrypted by responder 101x or device 101 using a step 337 from Figure 5a. DPP server 103 would record the eUICC profile as credentials 109' (or "from ..." in Figure 3b). For embodiments where asymmetrically encrypted credentials 109' comprising an eUICC profile are received by responder 101x in a DPP Configuration Response 124 message, then device 101 can also operate an eUICC as specified in as specified in GSMA standards SGP.22 and SGP.02.

Values or data in credentials 109 could comprise frequencies or channels to utilize, a service set identifier (SSID) or network name, user identities and passwords, and/or public certificates for device 101, public certificates for a certificate authority, etc. As contemplated herein, a set of credentials 109 can include supporting data to use the credentials 109 and/or connect with an access network 112 or network 107, such as configuration data for a WiFi radio 101h in device 101. The supporting data could comprise Config.network-AP 109c as depicted and described in connection with Figure 1a. Data collected for a set of credentials 109 was also described in a step 335 in Figure 3a.

Device 101 and access point 105 can conduct a WiFi session setup 602 using the WiFi radio 101i with credentials 109. WiFi connection setup 602 can utilize standard WiFi security protocols such as, but not limited to, WPA2, WPA3, or subsequent and related standards. In exemplary embodiments, WiFi connection setup 602 utilizes PMK and a PMKID, where PMK and PMKID can be uniquely associated with device 101. Or WiFi connection setup 602 could use a preshared secret key (PSK) in credentials 109. At step 603, device 101 can select the DPP server 103 and a URL for DPP server 103. A URL for DPP server 103 could be included with credentials 109, or otherwise fetched by device 101 from an IP network 113 after session setup 602. A step 603 can also comprise device 101 confirming expected connectivity to an IP network 113 is provided via access point 105, such as performing domain name server (DNS) requests and downloading data via HTTP or HTTPS/TLS from an IP network 113.

Device 101 and DPP server 103 can then conduct a secure session setup 604. Secure session setup 604 could comprise a session using TLS, IPSec, VPN, SSH, or similar secure networking technologies to establish secure sessions. In exemplary embodiments, DPP server 103 uses the responder bootstrap public key Br 101a as a public key for device 101 in secure session 603 and device 101 uses the initiator bootstrap public key Bi 104a as a public key for DPP server 103 in secure session 603. In other words, (i) DPP server 103 uses the initiator bootstrap private key bi 104b as the private key for DPP server 103 in secure session 603, and (ii) device 101 uses the responder bootstrap private key br 101b as the private key for device 101 in secure session 603. In exemplary embodiments, the public keys responder bootstrap public key Br 101a and the initiator

Petitioner Samsung Ex-1004, 1135

bootstrap public key Bi 104a are not transmitted between the nodes in a secure session setup 603, but rather the public keys are selected for use since they are already recorded by the two nodes. In this manner, security of secure session 603 can be enhanced, because the public keys are not transmitted or received in a plaintext form, but rather are already previously recorded by the two nodes. In an exemplary embodiment, secure session setup 603 could designate or identify the initiator bootstrap public key Bi 104a (and corresponding private key) for device 101, which could be selected by DPP server 103 sending an initiator bootstrap keys sequence number 197 in a secure session setup 603, for embodiments where device 101 can record a plurality of initiator bootstrap PKI keys, as depicted in device table 104d in Figure 2b. Note that a secure session setup 604 could also use different public keys than those depicted in Figure 6a, such as a certificate for DPP server 103 cert.DPPS 103c and a certificate for device 101 cert.device 101m.

At step 605a, device 101 can download and install a configuration package for device 101 using WiFi session 602 in Figure 6a. The configuration package can be downloaded from network 107, and also potentially from DPP server 103 using secure session 604. Configuration package for device 101 can comprise an updated operating configuration for device 101. Configuration package for device 101 could be downloaded from a network 107. Configuration package for device 101 could include updated software or firmware for device 101, different credentials for device 101 to use when connecting with servers within a network 107, such as potentially a reporting server or a monitoring server within a network 107, timers for device 101 to use with servers in a network 107, and other data as well for the ongoing operation of device 101 with a network 107. After internally recording or loading the files for a configuration package in a step 605a, device 101 can perform a reboot, so that device 101 restarts with the new files from the configuration package. Upon a reboot in a step 605a, connections 602 and 603 may temporarily terminate with the reboot, but device 101 can re-establish connections 602 and 603 after reboot. A step 605a can then also comprise device 101 creating a report 605aa, where report 605aa includes a status code with success or errors for each file in a configuration package. In other words, report 605aa can record the success or errors of applying each of the files in a configuration package, which may be useful for network 107 in system 100 to know the configured state of device 101 after a step 605a. Note that step 605a can take place before a secure session setup 604.

At step 605b device 101 can perform an RF frequency scan or sweep all available mobile network operators and network access technologies around device 101 in order to collect a second networks available list 304a. The second networks available list 304c can be useful for DPP server 103 or network 107 to confirm that a selected primary access network 308aa for device 101 is in fact a preferred network. An exemplary first networks available list 304c was depicted in Figure 3a above (for initiator 102'). Although initiator 102' also collected a first networks available list 304c, device 101 can operate with a different antenna and set of radios 101h with different capabilities

Petitioner Samsung Ex-1004, 1136

than initiator 102'. For example, RSSI or RF signal strength measurements by initiator 102' in a step 303a may be different than for a device 101. As another example, device 101 may operate at a different location (such as an exemplary tens of meters away) from the location of initiator 102' in a step 303a, and thus a preferred access network 112 for device 101 may be different using the second networks available list 304a from device 101 in step 605b than a first networks available list 304a from initiator 102'. In other words, a network selection step 308a may provide an initial estimate for a preferred access network 112 (identified by selected primary network 308aa in Figure 3b), but the actual/updated preferred access network 112 for device 101 using the criteria from a step 308a could be selected in a step 605b using the second networks available list 304c from device 101. A step 605b for device 101 can include the actions for an initiator 102' in a step 303a to collect a first networks available list 304a for initiator 102'.

Device 101 can send DPP server 103 a message 606, where message 606 can include a device identity ID-token.device 206, the second networks available list 304c collected by device 101 in a step 605b, an initiator bootstrap sequence number 197 for the most recent initiator bootstrap public key Bi 104a used by device 101, and a report 605aa, which could comprise the report generated in a step 605a. As depicted in Figure 6a, message 606 can be sent via (i) access point 105 and then (ii) access network 112 for access point 105. Although a DPP server 103 is depicted in Figure 6a as communicating with device 101 and receiving message 606, another server in a network 107 or associated with network 107 could receive the message 606 (using a secure session similar to secure session 604 depicted in Figure 6a). Note that message 606 could also include the most recent initiator bootstrap public key Bi 104a recorded by device 101.

A DPP server 103 could then conduct a step 308a using the second networks available list 304c from device 101, in order to select an updated, selected primary access network 308aa. A step 308a was depicted and described above in connection with Figure 3a. For embodiments where a step 308a by DPP server 103 selects a different, second primary access network 308aa for device 101 in Figure 6a (e.g. different than a first primary access network 308aa in Figure 3a), a DPP server 103 could then conduct a step 308b in order to receive an updated, second set of credentials 109 for the second primary access network 308aa. Note that the credentials 109 received and recorded in a step 308b by DPP server 103 can comprise an asymmetrically encrypted credentials 109', where a server functioning as the source of asymmetrically encrypted credentials 109' could perform a step 336 depicted and described in connection with Figure 5a. In exemplary embodiments, only device 101 could feasibly read the plaintext credentials 109 for an asymmetrically encrypted credentials 109', where device 101 would use a recorded private key (e.g. key 101b, 101f, or the private key for cert.device 101m) in order to perform an elliptic curve Elgamal asymmetric decryption on the credentials 109'.

Petitioner Samsung Ex-1004, 1137

A DPP server 103 could conduct a step 607, where DPP server 103 can process the received report 605aa in order to confirm a configuration package for device 101 was successfully applied in a step 605a. DPP server 103 could also forward report 605aa to other servers in a network 107 in a step 607. If errors or problems are detected in report 605aa in a step 607, then DPP server 103 can send a command to device 101 to back out the configuration package (e.g. uninstall) or restore a previous state for device 101.

At step 608, DPP server 103 can determine if recorded initiator bootstrap public key Bi 104a for device 101 should be updated. There could be many reasons both device 101 and network 107 may prefer to update initiator bootstrap public key Bi 104a for a responder 101x operating in device 101. A device 101 may have a lifetime of a decade or longer, and increased key security, use of different cryptographic algorithms, or different cryptographic parameters 199 may be preferred over time. Another reason could be that device 101 or network 107, after (a) a first configuration step 106 via a first session of a device provisioning protocol, may prefer (b) to support a second configuration step 106 via a second session of a device provisioning protocol, where the second session could use a different initiator bootstrap PKI key pair. Another reason could be that a first device 101 used the same key (e.g. same numeric value) for an initiator bootstrap public key Bi 104a as second device 101 (or possibly multiple other devices 101), which is also depicted for "shared keys" in a key table 104k in a device database 104 in Figure 2b. Security could be increased for the first device 101 by changing the recorded initiator bootstrap public key Bi 104a after the key has been used. Other reasons for a device 101, DPP server 103, or network 107 may exist for a DPP server 103 to determine key Bi 104a should be updated for device 101, without departing from the scope of the present invention. Figure 6 depicts an embodiment where DPP server 103 selects to update the initiator bootstrap public key Bi 104a for device 101 in a step 608.

At step 609, DPP server 103 can derive a new, updated initiator bootstrap public key Bi 104a and a corresponding new, updated initiator bootstrap private key bi 104b. The DPP server 103 can use a selected set of cryptographic parameters 199a to derive the keys. The DPP server 103 can use a random number generator and a key pair generation algorithm in order to derive the new, update initiator bootstrap PKI keys. The DPP server 103 can use the step 313 depicted and described in Figure 3a to derived the initiator bootstrap PKI keys (e.g. using same or similar steps as deriving initiator ephemeral PKI keys). Although Figure 6a depicts DPP server 103 as conducting a step 609 to derive new initiator bootstrap PKI keys, another server in a network 107 could conduct a step 609, including device database 104. At step 609, DPP server 103 can also update initiator bootstrap keys sequence number 197 recorded in a device database 104 depicted in Figure 2b to reflect or account for the use of new initiator bootstrap PKI keys. The updated number 197 is depicted in Figure 6a an initiator bootstrap keys sequence number 197a. In exemplary embodiments, the initiator bootstrap keys sequence number 197 can be used at a later time by DPP server 103 when conducting a step

Petitioner Samsung Ex-1004, 1138

302d for device 101 (e.g. querying device database 104 for initiator bootstrap PKI keys in Figure 3a), such as if DPP server 103 supports a device provisioning protocol for device 101 again at a later time.

A DPP server 103 can then send device database 104 a message 609a, where message 609a can include a device identity ID-token.device 206 (or ID.device 101i), the new, updated initiator bootstrap public key Bi 104a and the corresponding new, updated initiator bootstrap private key bi 104b, where the PKI keys were generated in a step 609. Message 609a can also include the updated initiator bootstrap keys sequence number 197a. Note that DPP server 103 and device database 104 can continue to use secure session 302a from a Figure 3a in order to secure message 609a. In some embodiments, initiator bootstrap private key bi 104b can be recorded only within DPP server 103 in a server database 103a, and for these embodiments initiator bootstrap private key bi 104b can be omitted from a message 609a.

At step 610, device database 104 can record the data from a message 609a in a device table 104d for device 101 using ID-token.device 206 (or ID.device 101i). The data can be recorded by adding a row in device table 104d for device 101. The new, updated initiator bootstrap public key Bi 104a and the corresponding new, updated initiator bootstrap private key bi 104b could be recorded in the added row, along with the received value for an initiator bootstrap keys sequence number 197a. In this manner, device database 104 can record and identify the initiator bootstrap PKI keys for use with device 101 at a later time, for reasons such as, but not limited to, the reasons discussed in connection with step 608 above. Database 104 can then send DPP server 103 a message 611, where message 611 can confirm that step 610 was successfully completed and the new initiator bootstrap PKI keys were recorded in a device table 104d.

After receiving message 611, DPP server 103 can send device 101 a message 612 via secure session 604. Message 612 can include the new, updated initiator bootstrap public key Bi 104a, which was previously derived by DPP server 103 in a step 609 and recorded by device database 104 in a step 610. Message 612 can also include (i) a selected set of parameters 199a associated with the new Bi 104a (e.g. key length, ECC curve name, etc.), and (ii) the updated initiator bootstrap keys sequence number 197a. Device 101 can receive message 612 with the new, updated initiator bootstrap public key Bi 104a and updated initiator bootstrap keys sequence number 197a.

At step 613, device 101 could record the data received in message 612 with a responder 101x operating in device 101. In exemplary embodiments, in step 612, device 101 could also use the received parameters 199a to derive an updated responder bootstrap private key br 101b and responder bootstrap public key Br 101a. Device 101 could use the equivalent steps as DPP server 103 in step 313 depicted and described in Figure 3a in order for responder 101x to derive a new, updated bootstrap private key br 101b and responder bootstrap public key Br 101a. A step 613 can also comprise device 101 recording the derived PKI keys and received key Bi 104a in nonvolatile

Petitioner Samsung Ex-1004, 1139

memory of device 101. Device 101 can also update a responder 101x operating in device 101 to use the new PKI keys for any subsequent, later instance of a device provisioning protocol either with an initiator 102' or an initiator 102*.

In message 614, device 101 can send DPP server 103 and "OK" message noting the successful receipt and recording of the new, updated initiator bootstrap public key Bi 104a. Message 614 can also include the new, updated responder bootstrap public key Br 101a. At a step 615, DPP server 103 can receive message 614 and record the data in a server database 103a for a device 101 using device identity ID-token.device 206. In exemplary embodiments, the new, updated responder bootstrap public key Br 101a can be used by DPP server 103 when supporting an initiator 102' to conduct a device provisioning protocol with device 101 and responder 101x at a later time.

DPP server 103 can then send device database 104 a message 615, where message 615 can include a device identity ID-token.device 206 (or ID.device 101i) and a confirmation that device 101 received and recorded the new, updated initiator bootstrap public key Bi 104a. In this manner, device database 104 can confirm that the new key Bi 104a with corresponding private key bi 104b is "ready for use", and can be selected in a subsequent query 309 (in Figure 3a) from a DPP server 103 for device 101 with identity ID-token.device 206. As depicted in Figure 6a, message 615 can also include a new, updated responder bootstrap public key Br 101a. Device database 104 can also record the received new key Br 101a in a key table 104k for device 101 with identity ID-token.device 206 as depicted in Figure 2b. In this manner, device database 104 can select the new, updated responder bootstrap public key Br 101a (along with initiator PKI keys) in a subsequent query 309 from a DPP server 103 for device 101, in order to conduct a later device provisioning protocol 191.

Initiator 102' can conduct a step 617, where step 617 can comprise the DPP application 102g or initiator 102' sending a query 618 to DPP server 103 in order to confirm successful completing of configuration step 106. DPP server 103 could determine if a configuration step 106 is complete based on the successful establishment of secure session 604 and receipt of a report 605aa. DPP server 103 could also determine that a configuration step 106 is complete using a report 605aa and any of: (i) device 101 has wireless connectivity properly established for a selected primary network 308aa and/or selected backup network 308ab (e.g. credentials 109 were received by device 101 and installed), (ii) device 101 has successfully collected a networks available list 304c, (iii) a configuration package was downloaded and installed in a step 605, (iv) device 101 has received an updated initiator bootstrap public key Bi 104a, and/or (v) device 101 has successfully updated responder bootstrap PKI keys.

As depicted in Figure 6a, DPP server 103 can reply to query 618 with a message 619 of "device OK", if configuration step 106 has been confirmed by DPP server 103 as completed, such as through any of (i) through (v) in the paragraph above. Individual components within (i) through (v)

Petitioner Samsung Ex-1004, 1140

in the paragraph above could be recorded or confirmed in a report 605aa. Not all of (i) through (v) are required for some embodiments, although connectivity for device 101 to an IP network 113 can be required for DPP server 103 to consider a configuration step 106 completed. In exemplary embodiments, upon successful receipt of a message 619 with a status of "device OK" for device 101 or an equivalent message 619, then initiator 102' can display to initiator user 102u that a configuration step 106 for device 101 has been completed and device 101 can be left in operation, potentially without additional steps or manual changes to be performed by initiator user 102u. For embodiments where message 619 provides a status of "device not OK" or equivalent, then initiator 102' could display to initiator user 102u an error status and initiator user 102u could work with device 101 and network 107 in order to diagnose and rectify an error code or error condition in order to receive a subsequent message 619 of "device OK".

Figure 6b

Figure 6b is a graphical illustration of an exemplary system, where a responder records a plurality of initiator bootstrap public keys and the responder selects and uses an initiator bootstrap public key, in accordance with exemplary embodiments. The present invention contemplates that the systems and methods for (i) using an initiator 102' with a DPP server 103 and a responder 101x for a device provisioning protocol, where (ii) a responder 101x can record a plurality of initiator bootstrap public keys Bi 104a. Benefits can include the plurality of initiator bootstrap public keys Bi 104a can be recorded in a device 101 with responder 101x before device 101 is taken to the physical location of a network access point 105. The plurality of initiator bootstrap public keys can be written to device 101 during manufacturing or distribution, thereby increasing the flexibility of (a) recording at least one compatible or supported initiator bootstrap public key Bi 104a in a device 101 for initiator 102', before (b) an initiator 102' is selected or known for conducting the device provisioning protocol steps 191 with a responder 101x. In exemplary embodiments, different values for recorded initiator bootstrap public keys Bi 104a can represent different possible networks 107 for configuration 106 of device 101 with responder 101x. As contemplated herein, (a) pre-recording a plurality of initiator bootstrap public keys Bi 104a in a plurality of devices 101 can still keep the various systems depicted herein secure, because (b) the corresponding initiator bootstrap private keys bi 104b can remain recorded individually and separately within each of a plurality of networks 107 which can conduct the device provisioning protocol 191 with the plurality of devices 101 recording the same values for initiator bootstrap public keys Bi 104a.

System 601 in Figure 6b can include an initiator 102' and a device 101, where device 101 can operate a responder 101x. Initiator 102' in Figure 6b can operate as an initiator 102' in Figure 1a, Figure 3a, Figure 5a, etc. above. Responder 101x can record (i) responder bootstrap PKI keys Br

Petitioner Samsung Ex-1004, 1141

101a and br 101b, as well as (ii) at least one of a set of selected cryptographic parameters 199a. Although Figure 6b depicts only a single set of PKI keys Br 101a and br 101b, and a set of selected cryptographic parameters 199a, a responder 101x could also record a plurality of different sets of PKI keys Br 101a and br 101b, with different parameters 199a (e.g. a first set of responder bootstrap keys for a first ECC curve P-256, a second set of bootstrap keys for a second ECC curve P-384, etc.). Responder 101x could determine which of the different sets of responder bootstrap PKI keys to utilize for conducting a device provisioning protocol 191 based on the received hash value for a responder bootstrap public key Br 101a received in a message 122. Although not depicted in Figure 6b, a responder 101x could pre-compute the hash values for each of a possible plurality of bootstrap public keys Br 101a using a hash 199e using cryptographic parameters 199a for each different responder bootstrap public key pair. Responder 101x could (i) listen for receiving a hash 199e value for any of the recorded bootstrap public keys Br 101a in an incoming message 122. When responder 101x observes a matching hash 199e value for a recorded key Br 101a, then select the private key br 101b for use in creating a response message 123. In other embodiments, a single recorded set of bootstrap PKI keys Br 101a and br 101b can be sufficient in order for device 101 to securely conduct a device provisioning protocol 191.

In some embodiments of the present invention, a responder 101x could record a plurality of different initiator bootstrap public keys Bi 104a in an initiator bootstrap public keys table 622, as depicted in Figure 6b. Each of the different initiator bootstrap public keys 104a could also be recorded with both (i) a hash 199e value for each individual key Bi 104a, and (ii) a selected set of cryptographic parameters 199a. Each of the different initiator bootstrap public keys 104a could be associated with a different network 107, as depicted by the exemplary values for possible networks 107 (e.g. "Google", "Amazon", etc.). The corresponding bootstrap private key bi 104b for each of the different depicted initiator bootstrap public keys Bi 104a could be recorded in each of the different networks 107. The ninth listed exemplary network 107 in a table 622 in responder 101x can comprise the exemplary network 107 depicted in Figure 1a, which can also operate the DPP server 103 in Figure 3a or Figure 5a, etc.

Although a single initiator bootstrap public key Bi 104a is depicted in Figure 6b for each different exemplary network 107, each different exemplary network 107 could also record multiple different initiator bootstrap public keys Bi 104a, such as (i) a first initiator bootstrap public key Bi 104a for a network 107 with a first set of selected cryptographic parameters 199a (e.g. with exemplary curve P-256), and (ii) a second initiator bootstrap public key Bi 104a with a second set of selected cryptographic parameters 199a (e.g. with exemplary curve P-384). Other possibilities exist as well for recording a plurality of keys Bi 104a for the same network 107 in a table 622 without departing from the scope of the present invention.

Petitioner Samsung Ex-1004, 1142

Further, the exemplary recorded initiator bootstrap public key Bi 104a for multiple networks 107 in a table 622 can either comprise (i) a "shared key" depicted and described in a key table 104k in Figure 2b where the value or number for the key Bi 104a can be recorded across a plurality of different devices 101, or (ii) a "unique key", where the value or number for the key Bi 104a can be unique for a device 101 (e.g. not recorded in other devices 101). The use of unique initiator bootstrap public keys Bi 104a recorded for responders 101x is also depicted in a key table 104k in Figure 2b. For embodiments where each device 101 records a "unique key" for the initiator bootstrap public key Bi 104a (such as during device manufacturing), then a device database 104 could record both (i) the unique initiator bootstrap public key Bi 104a for device 101 and (ii) the corresponding unique initiator bootstrap private key bi 104b (such as receiving the keys from the device manufacturer). Other possibilities exist as well, where a set of initiator bootstrap public keys 104a recorded in a device 101 for a network 107 can comprise either (i) shared keys or (ii) unique keys, without departing from the scope of the present invention.

As depicted in Figure 6b the responder 101x in device 101 can receive the message 122 from an initiator 102'. Message 122 was depicted and described in connection with Figure 3a above, and can comprise a DPP authentication request message. Responder 101x can process the message 122 and extract the hash 199e value of an initiator bootstrap public key Bi 104a. Responder 101x can (a) query 620 a table 622 with the hash 199e value H(Bi) in order to (b) select and read the value for the initiator bootstrap public key Bi 104a. The results of a query 620 can comprise a result 621 with key Bi 104a and parameters 199a. The result 621 (comprising at least one key Bi 104a as depicted in Figure 6b) can be used with a set of steps 325 depicted and described in connection with Figure 3a and Figure 5a. Steps 325 can generate a DPP authentication response 123 for initiator 102' using the selected key Bi 104a.

Although not depicted in Figure 6b, for embodiments where responder 101x records a plurality of different responder bootstrap PKI keys 101a and 101b, a query 620 can also be performed by responder 101x in order to select one responder bootstrap public key Br 101a and the corresponding responder bootstrap private key br 101b. Responder 101x could record the plurality of different responder bootstrap PKI keys in a table 623, along with a hash value 199e for each responder bootstrap public key Br 101a (not shown). Upon receipt of a message 122, responder 101x could conduct a query 620 using the received hash 199e H(Br) for a key Br 101a in a table 623 with a plurality of different responder bootstrap public keys Br 101a in order to select at least one key Br 101a and the corresponding private key br 101b. In exemplary embodiments, if the hash 199e value received in a message 122 does not match an internally calculated or recorded hash 199e value for a recorded responder bootstrap public key Br 101a, then responder 101x and device 101 remains silent on WiFi link 192, where WiFi link 192 is depicted above in Figure 3a, Figure 5a, etc.

Petitioner Samsung Ex-1004, 1143

Figure 7

Figure 7 is a graphical illustration of an exemplary system, where an access point conducts a configuration step for a device using a hosted device provisioning protocol, in accordance with exemplary embodiments. The present invention contemplates that the systems and methods for using an initiator 102' with a DPP server 103 can support a device provisioning protocol 191 in order to configure a network access point 105'. System 700 in Figure 7 can include an unconfigured network access point 105', a mobile phone 701, and a network 107. Unconfigured network access point 105' can include a responder 101x, where responder 101x can have the same or equivalent functionality as a responder 101x for a device 101 depicted and described above in Figure 1a, Figure 1b, Figure 1c, Figure 3a, Figure 5a, and Figure 5b. Responder 101x in network access point 105' can include a responder bootstrap public key Br 101a, a responder bootstrap private key br 101b, and an initiator bootstrap public key Bi 104a. PKI keys could be recorded in access point 105' before access point 105' is installed at an operating location, such as during manufacturing or distribution of unconfigured access point 105'. Responder 101x operating in unconfigured access point 105' can have responder capabilities of an enrollee.

Mobile phone 701 can comprise a smart phone or a mobile handset, such as a mobile phone manufactured by Apple® or running an Android based operating system. Mobile phone 701 could also comprise another mobile and networked computing device, such as a laptop or a tablet. Mobile phone 701 could include an initiator 102', where initiator 102' could have the same or equivalent functionality as initiator 102' depicted and described in connection with Figure 1a, Figure 1b, Figure 1c, Figure 1d, Figure 2a, Figure 3a, Figure 5a, Figure 5b, and Figure 6a. In exemplary embodiments, initiator 102' can operate as an application for mobile phone 701 downloaded from a repository of mobile phone applications, such as an "app store" or "play store", or possibly a private application. In other words, embodiments herein contemplate that a mobile phone 701 could comprise the physical entity for an initiator 102' and the software or firmware instructions for operation of initiator 102' could comprise a DPP application 102g. Other physical and/or logical arrangements for the operation of an initiator 102' are possible as well, without departing from the scope of the present invention. Mobile phone 701 can have both a WiFi radio 102w and a WAN radio 102h. Mobile phone 701 can communicate with unconfigured access point 105' via the WiFi radio 102w and a network 107 via the WAN radio 102h and an access network 112. Initiator 102' for mobile phone 701 can have capabilities of a configurator.

Network 107 can operate a DPP server 103, where a DPP server 103 in Figure 7 can have the functionality of a DPP server 103 depicted and described in connection with Figures 1a, Figure 3a, Figure 3b, Figure 5a, Figure 5b, and Figure 6a. Although not depicted in Figure 7, a network 107 can also operate other supporting servers and/or running processes depicted in Figure 1a, Figure 2a,

Petitioner Samsung Ex-1004, 1144

etc. Network 107 could be connected to mobile phone 107 via the access network 112 and an IP network 113 depicted in Figure 1a.

As depicted in Figure 7, mobile phone 107 using initiator 102' could conduct the series of messages for a device provisioning protocol 191 with responder 101x operating in unconfigured access point 105'. The series of messages for a device provisioning protocol 191 could comprise the messages 122, 123, 123a, 123b, and 124 as depicted and described in connection with Figure 3a and Figure 5a. In order for initiator 102' to send and receive the device provisioning protocol messages 191 depicted in Figure 7, initiator 102 transmit and receive a series of messages with DPP server 103 in network 107, where the series of messages could comprise a hosted device provisioning protocol 702.

For a first mode 306a, representing embodiments depicted in Figure 3a (where initiator bootstrap and ephemeral secret keys remain in DPP server 103), the series of messages for a hosted device provisioning protocol 702 could comprise the messages between initiator 102' and DPP server 103 that are depicted in Figure 3a (e.g. messages 302b, 305, 311, 317, etc.). For a second mode 306a, representing embodiments depicted in Figure 5a where the initiator bootstrap private key bi 104b can remain in DPP server 103, the series of messages for a hosted device provisioning protocol 702 could comprise the messages between initiator 102' and DPP server 103 that are depicted in Figure 5a (e.g. messages 302b, 305, 311, 503, 506, 510, etc.). For a third mode 306a, representing embodiments depicted in Figure 5b where the initiator bootstrap private key bi 104b and responder bootstrap public key Br 101a can remain in DPP server 103, the series of messages for a hosted device provisioning protocol 702 could comprise the messages between initiator 102' and DPP server 103 that are depicted in Figure 5b (e.g. messages 302b, 305, 311, 521, 522, etc.).

Upon completion of a device provisioning protocol 191 and a hosted device provisioning protocol 702, unconfigured network access point 105' can conduct a configuration step 106 to load a set of network credentials 109 and begin operating with the credentials 109, as depicted in Figure 7. The configuration step 106 can convert the unconfigured network access point 105' into an operating network access point 105, such that WiFi clients (or "sta") can connect with network access point 105 using a compatible set of network credentials 109. As depicted in Figure 7, a mobile phone 701 could also operate as an initiator 102' with a roll of a configurator for the access point 105'. Mobile phone 701 could also conduct a device provisioning protocol 191 using a hosted device provisioning protocol 702 with DPP server 103 in order to transfer a set of compatible network credentials 109 to device 101 in Figure 7, such that device 101 could connect with network access point 105'. The compatible network credentials 109 for device 101 can specify that device 101 operates in a WiFi client mode with network access point 105.

Figure 8

Petitioner Samsung Ex-1004, 1145

Figure 8 is a graphical illustration of an exemplary system, where an access point operates as an initiator for a device with a responder, in order transfer a set of network credentials to the device, in accordance with exemplary embodiments. The present invention contemplates embodiments where a network access point 105 can operate as an initiator 102' for a device 101 operating as a responder 101x, in order for the network access point 105 to transfer network credentials 109 to the device 101. System 800 in Figure 8 can include an unconfigured device 101', a network access point 105, and a network 107.

Network access point 105 in a system 800 can operate with a set of network access credentials 109 and also operate as an initiator 102'. Network access point 105 can connect with a network 107 via an access network 112. Network 107 can operate a DPP server 103, where the function and operation of a DPP server 103 is depicted and described above in connection with Figure 1a, Figure 3a, Figure 5a, etc. Although not depicted in Figure 8, network 107 could also operate a device database 104 and DPP server 103 could include a server database 103a. Network access point 105 could comprise a WiFi router and operate a WiFi radio. Network access point 105 can have the internal electrical components equivalent to an initiator 102' depicted and described above in connection with Figure 1d, including a CPU, RAM, nonvolatile memory, and a system bus. For the embodiment depicted in Figure 8, the initiator 102' in network access point 105 could comprise software, firmware, or a running process within network access point 105. Initiator 102' in network access point 105 could have the same or equivalent functionality as initiator 102' depicted and described in connection with Figure 1a, Figure 1b, Figure 1c, Figure 1d, Figure 2a, Figure 3a, Figure 5a, Figure 5b, and Figure 6a. Network access point 105 could operate with capabilities as a configurator 801, as depicted in Figure 8.

An unconfigured device 101' could be a device with a WiFi radio 101h and operate a responder 101x. The electrical components inside device 101' could be similar to the electrical components depicted for an initiator 102' in Figure 1d, such as including a CPU, RAM, nonvolatile memory, an operating system, and a system bus. Responder 101x in unconfigured device 101 could have the same or equivalent functionality as a responder 101x depicted and described above in Figure 1a, Figure 1b, Figure 1c, Figure 3a, Figure 5a, and Figure 5b. Unconfigured device 101' can have capabilities as an enrollee 802.

As depicted in Figure 8, network access point 105 initiator 102' could conduct the series of messages for a device provisioning protocol 191 with responder 101x operating in unconfigured device 101'. The series of messages for a device provisioning protocol 191 could comprise the messages 122, 123, 123a, 123b, and 124 as depicted and described in connection with Figure 3a and Figure 5a. In order for initiator 102' to send and receive the device provisioning protocol messages 191 depicted in Figure 8, initiator 102' (or AP 105 operating the initiator 102') transmits and

Petitioner Samsung Ex-1004, 1146

receives a series of messages with DPP server 103 in network 107, where the series of messages could comprise a hosted device provisioning protocol 702. The series of messages for a hosted device provisioning protocol 702 were depicted and described in connection with Figure 7 above. Note that the steps and messages for conducting a hosted device provisioning protocol 702 can depend on a mode 306a, where the mode 306a can be selected by a DPP server 103 in a step 306. The present invention also contemplates that a mode 306a for a hosted device provisioning protocol 702 can be selected by an initiator 102'. For these embodiments, where mode 306a is selected by an initiator 102', a selected mode 306a could be transmitted from initiator 102' to DPP server 103 in a message 305 depicted and described in connection with Figure 3a above.

The combined (i) device provision protocol 191 between network access point 105 and device 101 and (ii) hosted device provisioning protocol 702 between network access point 105 and DPP server 103 can securely transfer a set of compatible network access credentials 109 to device 101'. Upon completion of a device provisioning protocol 191 and a hosted device provisioning protocol 702, unconfigured device 101' can conduct a configuration step 106 to (i) load a compatible set of network credentials 109 for network access point 105 and (ii) begin operating with the credentials 109, as depicted in Figure 8. The configuration step 106 can convert the unconfigured device 101' into an operating device 101, such that device 101 can conduct a WiFi connection setup step 803 using the received set of compatible network credentials 109.

Conclusion

Various exemplary embodiments have been described above. Those skilled in the art will understand, however, that changes and modifications may be made to those examples without departing from the scope of the claims.

Petitioner Samsung Ex-1004, 1147

CLAIMS

1. A method for supporting a device provisioning protocol (DPP), the method performed by a first computing device, the method comprising:

a) establishing, by the first computing device, a secure session with a server, wherein the secure session comprises at least authentication of the first computing device;

b) receiving, by the first computing device and from a second computing device, a tag value for a responder;

c) sending, by the first computing device and to the server, at least a portion of the received tag value;

d) sending, by the first computing device and to the second computing device, an initiator nonce, wherein the initiator nonce is encrypted using at least a first shared secret derived by the first computing device;

e) receiving, by the first computing device and from the second computing device, a responder ephemeral public key for the second computing device;

f) sending, by the first computing device, the responder ephemeral public key to the server;

g) receiving, by the first computing device and from the server, a second shared secret, wherein the second shared secret is derived using at least a responder bootstrap public key and the responder ephemeral public key;

h) deriving, by the first computing device, an encryption key using at least the first shared secret and the second shared secret;

i) encrypting, by the first computing device, an authorization value using at least the derived encryption key;

Petitioner Samsung Ex-1004, 1148

j) sending, by the first computing device and to the second computing device, the encrypted authorization value; and

k) sending, by the first computing device and to the second computing device, a ciphertext of a network credential, wherein the ciphertext is encrypted using at least the derived encryption key.

2. The method of claim 1, wherein the first computing device comprises an initiator and the second computing device comprises the responder, and wherein the responder ephemeral public key comprises a responder protocol public key.

3. The method of claim 2, wherein the first computing device comprises a mobile phone operating a software application for the initiator.

4. The method of any one of claims 1 to 3, wherein the responder bootstrap public key is received from the server, and wherein the first shared secret is derived using the responder bootstrap public key and an initiator ephemeral public key.

5. The method of any one of claims 1 to 4, wherein the tag value includes the responder bootstrap public key, wherein the portion of the received tag value comprises the responder bootstrap public key, and wherein the first shared secret is derived using the responder bootstrap public key and an initiator ephemeral public key.

6. The method of any one of claims 1 to 5, wherein the initiator nonce is sent in a DPP authentication request, wherein the responder ephemeral public key is received in a DPP

Petitioner Samsung Ex-1004, 1149

authentication response, wherein the authorization value is sent in a DPP authentication confirm message, and wherein the ciphertext is sent in a DPP configuration response message.

7. The method of any one of claims 1 to 6, wherein the first shared secret is derived using (i) an initiator ephemeral private key, (ii) the responder bootstrap public key, and (iii) a first elliptic curve Diffie Hellman key exchange, and wherein the second shared secret is derived by the server using an initiator bootstrap private key and a second elliptic curve Diffie Hellman key exchange.

8. A method for supporting a device provisioning protocol (DPP), the method performed by a network, the method comprising:

a) storing, by the network, an initiator bootstrap private key and a responder bootstrap public key;

b) establishing, by the network, a secure session with a computing device, wherein the computing device authenticates with the network, and wherein the network receives a tag value for a responder via the secure session;

c) sending, by the network and to the computing device, (i) an initiator bootstrap public key for the initiator bootstrap private key, and (ii) an initiator configuration;

d) receiving, by the network and from the computing device, a responder ephemeral public key;

e) conducting, by the network, a first elliptic curve Diffie-Hellman (ECDH) key exchange using the responder bootstrap public key, the responder ephemeral public key, and the initiator bootstrap private key in order to derive a shared secret key;

Petitioner Samsung Ex-1004, 1150

f) sending, by the network and to the computing device, the derived shared secret key via the secure session; and

g) receiving, by the network and from the computing device, a plaintext configuration attribute, wherein a ciphertext configuration attribute is decrypted into the plaintext configuration attribute using at least the derived shared secret key.

9. The method of claim 8, wherein the network includes a database and a DPP server, and wherein the database stores the initiator bootstrap private key and the DPP server conducts the first ECDH key exchange.

10. The method of claim 8 or 9, wherein the computing device reads the tag value for the responder, and wherein the computing device establishes a DPP session with the responder over a WiFi connection.

11. The method of any one of claims 8 to 10, wherein the derived shared secret key comprises a value of "L" according to the device provisioning protocol.

12. The method of any one of claims 8 to 11, wherein the computing device derives an initiator ephemeral private key at least for conducting a second ECDH key exchange, wherein the second ECDH key exchange derives a value of "M" according to the device provisioning protocol, and wherein the computing device derives the value of "M" before the computing device sends a DPP authentication request message.

Petitioner Samsung Ex-1004, 1151

13. The method of claim 12, further comprising decrypting, by the computing device, the ciphertext configuration attribute using at least (i) the shared secret key received from the network, and (ii) the value of "M".

14. A mobile device for supporting a device provisioning protocol (DPP), the mobile device comprising:

a camera for reading a tag value associated with a WiFi device;

a WAN radio for establishing a secure session with a server, for sending the tag value to the server, for receiving from the server (i) an initiator configuration (ii) a first hash value of a responder bootstrap public key and (iii) a first ciphertext;

a WiFi radio for operating with a user configuration before the mobile device reads the tag value, for operating with the initiator configuration after the mobile device reads the tag value, for transmitting a DPP authentication request message with the first ciphertext and the first hash value, and for receiving a DPP authentication response message with a second ciphertext and a responder ephemeral public key;

a nonvolatile memory for storing (i) the user configuration while the WiFi radio operates with the initiator configuration, and (ii) a DPP application for conducting the DPP over the WiFi radio;

a processor for validating the responder ephemeral public key, for encrypting the responder ephemeral public key and the second ciphertext for the secure session, and for receiving a third ciphertext from the server, wherein the third ciphertext includes data for a DPP authentication confirm message; and

Petitioner Samsung Ex-1004, 1152

a data bus for electrically connecting the camera, WAN radio, WiFi radio, nonvolatile memory, and processor, and for transferring a fourth ciphertext with a set of credentials for the WiFi device from the WAN radio to the WiFi radio.

15. The mobile device of claim 14, wherein the mobile device (i) does not store and (ii) does not receive the responder bootstrap public key for the WiFi device.

16. The mobile device of claim 14 or 15, wherein the DPP application applies the user configuration to the WAN radio after the mobile device sends the set of credentials to the WiFi device, and wherein the mobile device comprises at least one of (i) a mobile phone, (ii) a tablet computer, and (iii) a laptop computer.

17. The mobile device of any one of claims 14 to 16, wherein the WiFi device comprises a responder for the DPP, and wherein the DPP authentication request message is transmitted by the mobile device as a broadcast message over a WiFi link.

18. The mobile device of any one of claims 14 to 17, wherein the server records an initiator bootstrap private key corresponding to an initiator bootstrap public key, wherein the mobile device receives a second hash value of the initiator bootstrap public key via the secure session, and wherein the DPP authentication request message includes the second hash value.

19. The mobile device of any one of claims 14 to 18, wherein the third ciphertext is encrypted with a symmetric ciphering key, and wherein the symmetric ciphering key is derived

Petitioner Samsung Ex-1004, 1153

using an elliptic curve Diffie Hellman (ECDH) key exchange and at least (i) an initiator bootstrap private key, (ii) the responder bootstrap public key, and (iii) the responder ephemeral public key.

20. The mobile device of claim 19, wherein the WiFi radio transmits a DPP authentication confirm message that includes the third ciphertext, wherein the WiFi radio receives a DPP configuration request message after the WiFi radio transmits the DPP authentication confirm message, wherein the WiFi radio transmits a DPP configuration response message with the fourth ciphertext to the WiFi device, and wherein the fourth ciphertext is encrypted with the symmetric ciphering key.

21. A method for supporting a device provisioning protocol, the method performed by a server, the method comprising:

a) recording, by the server, an initiator bootstrap private key;

b) receiving, by the server and from an initiator, a responder bootstrap public key;

c) deriving, by the server, (i) an initiator ephemeral private key, (ii) a corresponding initiator ephemeral public key, and (iii) a pseudo random number comprising an initiator nonce;

d) conducting, by the server, (x) a first elliptic curve Diffie-Hellman (ECDH) key exchange using the responder bootstrap public key and the derived initiator ephemeral private key in order to (y) derive a first symmetric ciphering key;

e) encrypting, by the server, the initiator nonce into a first ciphertext using the first symmetric ciphering key;

f) sending, from the server to the initiator, the first ciphertext and the initiator ephemeral public key;

Petitioner Samsung Ex-1004, 1154

g) receiving, by the server and from the initiator, a second ciphertext and a responder ephemeral public key;

h) conducting, by the server, (x) a second ECDH key exchange using the responder ephemeral public key and the derived initiator private key in order to (y) derive a second symmetric ciphering key;

i) decrypting, by the server, the second ciphertext into a first plaintext using the second symmetric ciphering key, wherein the first plaintext from the second ciphertext includes a third ciphertext;

j) conducting, by the server, (x) a third ECDH key exchange using the received responder bootstrap public key, the received responder ephemeral public key, and the recorded initiator bootstrap private key in order to (y) derive a third symmetric ciphering key;

k) decrypting, by the server, the third ciphertext into a second plaintext using the third symmetric ciphering key, wherein the second plaintext includes a received responder authentication value;

l) determining, by the server, the received responder authentication value equals a calculated responder authentication value using at least, in part, the initiator nonce;

m) encrypting, by the server, a set of network credentials using the third symmetric ciphering key; and

n) sending, from the server to the initiator, the encrypted set of network credentials.

22. The method of claim 21, further comprising sending, from the server to the initiator, a fourth ciphertext before sending the set of network credentials, wherein the fourth ciphertext includes an initiator authentication value, and wherein the fourth ciphertext is encrypted using the third symmetric ciphering key.

Petitioner Samsung Ex-1004, 1155

23. The method of claim 21 or 22, further comprising the initiator operating as an initiator proxy, wherein the initiator proxy does not receive and does not operate with the initiator bootstrap private key.

24. The method of any one of claims 21 to 23, further comprising receiving, by a discovery server, a tag value from the initiator, wherein the discovery server uses the tag value to send the initiator a domain name for the server.

25. The method of any one of claims 21 to 24, wherein the server records a set of cryptographic parameters for the initiator bootstrap private key and the initiator ephemeral private key, wherein the server uses the set of cryptographic parameters for conducting the first, second, and third ECDH key exchanges.

26. The method of claim 25, wherein the server uses a key derivation function and the set of cryptographic parameters with the first, second, and third ECDH key exchanges in order to respectively derive the first, second, and third symmetric keys.

27. The method of any one of claims 21 to 26, further comprising establishing, by the server and the initiator, a secure session, wherein the first, second, and third ciphertexts are transmitted through the secure session.

Petitioner Samsung Ex-1004, 1156

28. A method for supporting a device provisioning protocol (DPP), the method performed by a mobile phone, the method comprising:

a) operating, by the mobile phone, a DPP application, wherein the DPP application comprises an initiator for the device provisioning protocol;

b) establishing, by the mobile phone, a secure session with a server, wherein the secure session comprises at least, in part, authentication of the DPP application;

c) receiving, by the mobile phone, a tag value for a responder;

d) sending, by the mobile phone and to the server, the received tag value;

e) receiving, by the mobile phone and from the server, an initiator configuration, wherein the mobile phone uses the received initiator configuration with a WiFi radio in the mobile phone.

f) receiving, by the mobile phone and from the server, an initiator ephemeral public key and a first ciphertext, wherein the first ciphertext includes an initiator nonce;

g) sending, by the mobile phone and to the device, the initiator ephemeral public key and the first ciphertext, wherein the mobile phone uses the received initiator configuration to send the initiator bootstrap public key and the first ciphertext to the device;

h) receiving, by the mobile phone and from the device, a responder ephemeral public key and a second ciphertext, wherein the second ciphertext includes a responder authentication value;

i) sending, by the mobile phone and to the server, the responder ephemeral public key and the second ciphertext; and

j) receiving, by the mobile phone and from the server, a third ciphertext, wherein the third ciphertext includes a set of network credentials for the device.

29. The method of claim 28, further comprising using a camera for the mobile phone in order to receive a QR code for the device, wherein the received QR code includes the tag value and

Petitioner Samsung Ex-1004, 1157

a set of cryptographic parameters, and wherein the tag value does not include the responder bootstrap public key.

30. The method of claim 28 or 29, wherein the mobile phone does not receive a responder bootstrap public key from the device, and wherein the server records the responder bootstrap public key before the mobile phone establishes the secure session with the server.

31. The method of any one of claims 28 to 30, further comprising the mobile phone conducting a radio-frequency scan in order to obtain a networks available list, wherein the mobile phone (i) sends the network available list to the server before (ii) receiving the set of network credentials in the third ciphertext.

32. The method of claim 31, further comprising the server using the networks available list in order to select a primary access network for the device, wherein the server uses the selected primary access network to select the set of network credentials from a server database.

33. The method of any one of claims 28 to 32, further comprising the mobile phone using (i) a channel list for the initiator configuration and (ii) the WiFi radio for the mobile phone to cycle through the channel list and send the initiator ephemeral public key and the first ciphertext to a responder on a plurality of channels in the channel list.

34. The method of any one of claims 28 to 33, wherein (a) sending, by the mobile phone and to the device, the initiator ephemeral public key and the first ciphertext comprises (b) a Device Provisioning Protocol (DPP) authentication request message.

Petitioner Samsung Ex-1004, 1158

35. A system for supporting a device provisioning protocol (DPP), the system comprising:

an authentication server for authenticating an initiator and for sending the authenticated initiator an initiator configuration;

a mobile phone for operating a DPP application, wherein the DPP application comprises at least, in part, the initiator, for receiving a responder bootstrap public key from a responder, wherein a secure hash value for the responder bootstrap public key is associated with a device identity, for using the initiator configuration to communicate over a WiFi link, for receiving a responder ephemeral public key using the WiFi link, for receiving an encrypted set of network credentials from a network, and for sending the encrypted set of network credentials to the responder over the WiFi link;

a device database for recording the device identity, an initiator bootstrap private key, and a set of cryptographic parameters for the bootstrap private key;

a DPP server for receiving the device identity from the initiator, for querying the device database using the device identity in order to receive the initiator bootstrap private key, for receiving the responder bootstrap public key from the mobile phone, for conducting an elliptic curve Diffie-Hellman (ECDH) key exchange using (i) the received responder bootstrap public key (ii) the received responder ephemeral public key. and (iii) the received initiator bootstrap private key, wherein the output of the ECDH key exchange comprises a symmetric ciphering key, for selecting an initiator mode for the initiator, and for sending the initiator mode to the mobile phone;

a processor in the DPP server for encrypting the set of network credentials using the symmetric ciphering key and the set of cryptographic parameters; and

Petitioner Samsung Ex-1004, 1159

a LAN interface in the DPP server for sending the encrypted set of network credentials to the mobile phone over a secure connection established between the DPP server and the mobile phone.

36. The system of claim 35, wherein the mobile phone operates as an initiator proxy, and wherein the mobile phone does not receive and does not operate with the initiator bootstrap private key.

37. The system of claim 35 or 36, further comprising a responder for recording an initiator bootstrap public key corresponding to the initiator bootstrap private key, and for using the recorded initiator bootstrap public key and to derive the symmetric ciphering key,

38. The system of claim 37, further comprising a network access point for using the set of network access credentials, and for providing connectivity with an IP network to the responder using the set of network access credentials

39. The system of any one of claims 35 to 38, wherein the mobile phone receives the responder ephemeral public key in a DPP authentication response message, wherein the DPP authentication response message includes a ciphertext of a responder authentication value, and wherein the ciphertext is encrypted with the symmetric ciphering key.

40. The system of any one of claims 35 to 39, wherein the DPP server uses the initiator bootstrap private key with a plurality of different responder bootstrap public keys, wherein the

Petitioner Samsung Ex-1004, 1160

plurality of different responder bootstrap public keys are associated with a plurality of different responders.

41. A method for supporting a device provisioning protocol (DPP), the method performed by a computing device, the method comprising:

a) recording, by the computing device, (i) a plurality of responder bootstrap private keys, and (ii) a plurality of initiator bootstrap public keys, wherein at least the plurality of initiator bootstrap public keys is recorded during one of a manufacturing and a distribution of the computing device;

b) operating, by the computing device, a WiFi radio and a responder, wherein the WiFi radio communicates with an initiator proxy;

c) receiving a DPP authentication request message with (i) a first secure hash value of a responder bootstrap public key and (ii) a second secure hash value of an initiator bootstrap public key;

d) selecting (i) a responder bootstrap private key from the plurality of responder bootstrap private keys using the first secure hash value and (ii) the initiator bootstrap public key from the plurality of initiator bootstrap public keys using the second secure hash value, wherein a DPP server records a corresponding initiator bootstrap private key for the initiator bootstrap public key;

e) generating a DPP authentication response using at least the responder bootstrap private key and the initiator bootstrap public key; and

f) receiving a DPP configuration response with network credentials for an access point.

Petitioner Samsung Ex-1004, 1161

42. The method of claim 41, wherein the plurality of initiator bootstrap public keys is recorded before the computing device is located with the access point.

43. The method of claim 41 or 42, wherein the computing device is associated with a tag, and wherein the initiator proxy reads the tag in order to receive the DPP authentication request message from the DPP server.

44. The method of any one of claims 41 to 43, further comprising:

recording, by the computing device a corresponding plurality of responder bootstrap public keys;

selecting a responder bootstrap public key from the plurality of responder bootstrap public keys using the first secure hash value;

generating the DPP authentication response using at least the responder bootstrap public key; and

an authentication server for authenticating an initiator and for sending the authenticated initiator an initiator configuration.

45. A method for supporting a device provisioning protocol (DPP), the method performed by a network, the method comprising:

a) recording a responder bootstrap public key, a set of cryptographic parameters, and an identity token for a device in a device database;

b) authenticating a mobile handset, wherein the network receives the identity token for a device from the authenticated mobile handset;

Petitioner Samsung Ex-1004, 1162

c) selecting the responder bootstrap public key and the set of cryptographic parameters using the identity token;

d) deriving, by the network and using the set of cryptographic parameters, (i) an initiator ephemeral private key, (ii) a corresponding initiator ephemeral public key, and (iii) a pseudo random number comprising an initiator nonce;

e) conducting, by the network, (x) a first elliptic curve Diffie-Hellman (ECDH) key exchange using the responder bootstrap public key and the derived initiator ephemeral private key in order to (y) derive a first symmetric ciphering key k1;

f) encrypting, by the network, at least the initiator nonce into a first ciphertext using the first symmetric ciphering key k1;

g) sending, from the network and to the mobile handset, the first ciphertext and the initiator ephemeral public key;

h) receiving, by the network and from the mobile handset, a second ciphertext and a responder ephemeral public key;

i) conducting, by the network, (x) a second ECDH key exchange using the responder ephemeral public key and the derived initiator private key in order to (y) derive a second symmetric ciphering key k2;

j) decrypting, by the network, the second ciphertext into a first plaintext using the second symmetric ciphering key k2, wherein the first plaintext from the second ciphertext includes a third ciphertext;

k) conducting, by the network, (x) a third ECDH key exchange using the recorded responder bootstrap public key and the received responder ephemeral public key in order to (y) derive a third symmetric ciphering key ke;

Petitioner Samsung Ex-1004, 1163

l) decrypting, by the network, the third ciphertext into a second plaintext using the third symmetric ciphering key ke, wherein the second plaintext includes a received responder authentication value;

m) determining, by the network, the received responder authentication value equals a calculated responder authentication value using at least, in part, the initiator nonce;

n) encrypting, by the network, a set of network credentials for the device using the third symmetric ciphering key ke; and

o) sending, from the network and to the mobile handset, the encrypted set of network credentials.

46. The method of claim 45, wherein the network receives the responder bootstrap public key from at least one of (i) a device manufacturer and (ii) a device owner.

47. The method of claim 45 or 46, wherein the identity token comprises a secure hash value of the responder bootstrap public key.

48. The method of claim 45 or 46, wherein the identity token comprises a device identity for the device.

49. The method of any one of claims 45 to 48, wherein the network and the mobile handset establish a secure session after the mobile handset authenticates with the network, and wherein the first, second, and third ciphertexts are transmitted through the secure session.

Petitioner Samsung Ex-1004, 1164

50. The method of any one of claims 45 to 49, wherein the network comprises at least an authentication server, the device database, and a DPP server.

51. The method of any one of claims 45 to 50, wherein the device operates a responder for the device provisioning protocol, and wherein the device (i) mutually derives the third symmetric ciphering key ke and (ii) decrypts the encrypted set of network credentials using the third symmetric ciphering key ke.

52. The method of any one of claims 45 to 51, wherein the device does not record an initiator bootstrap public key for the network.

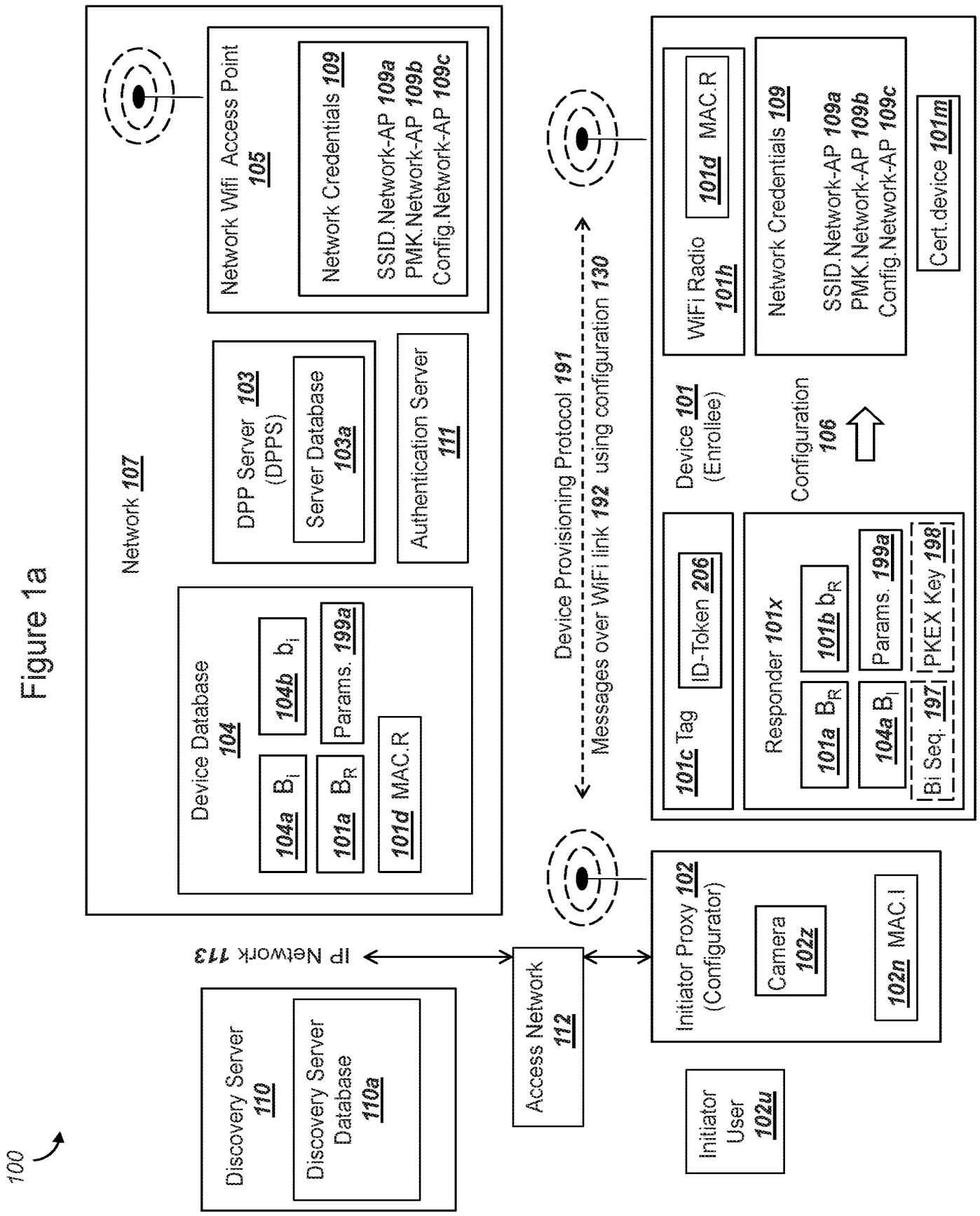
53. The method of any one of claims 45 to 52, wherein the network receives the set of network credentials for the device from the authenticated mobile handset.

54. The method of any one of claims 45 to 52, wherein the network receives the set of network credentials from a device owner.

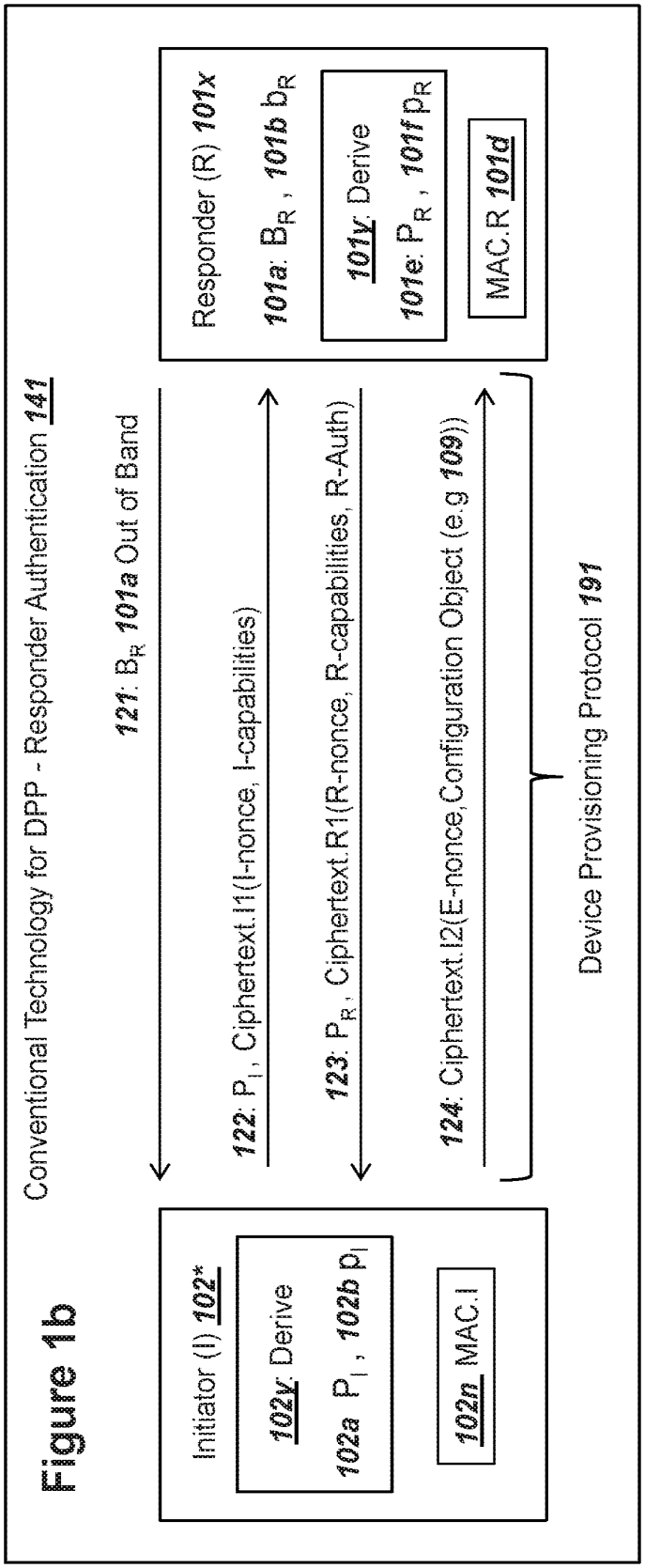
55. The method of any one of claims 45 to 54, wherein the initiator ephemeral public key comprises an initiator protocol public key, and wherein the responder ephemeral public key comprises a responder protocol public key.

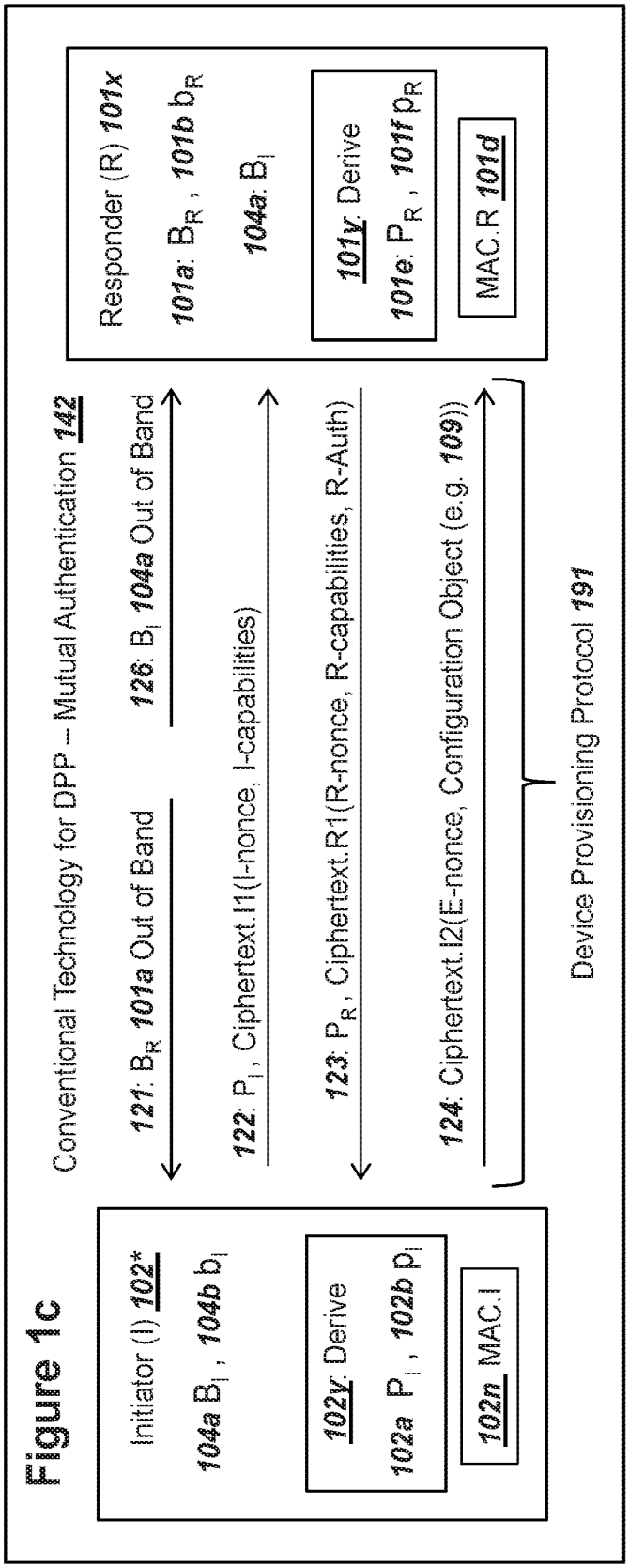
Petitioner Samsung Ex-1004, 1165

Figure 1a



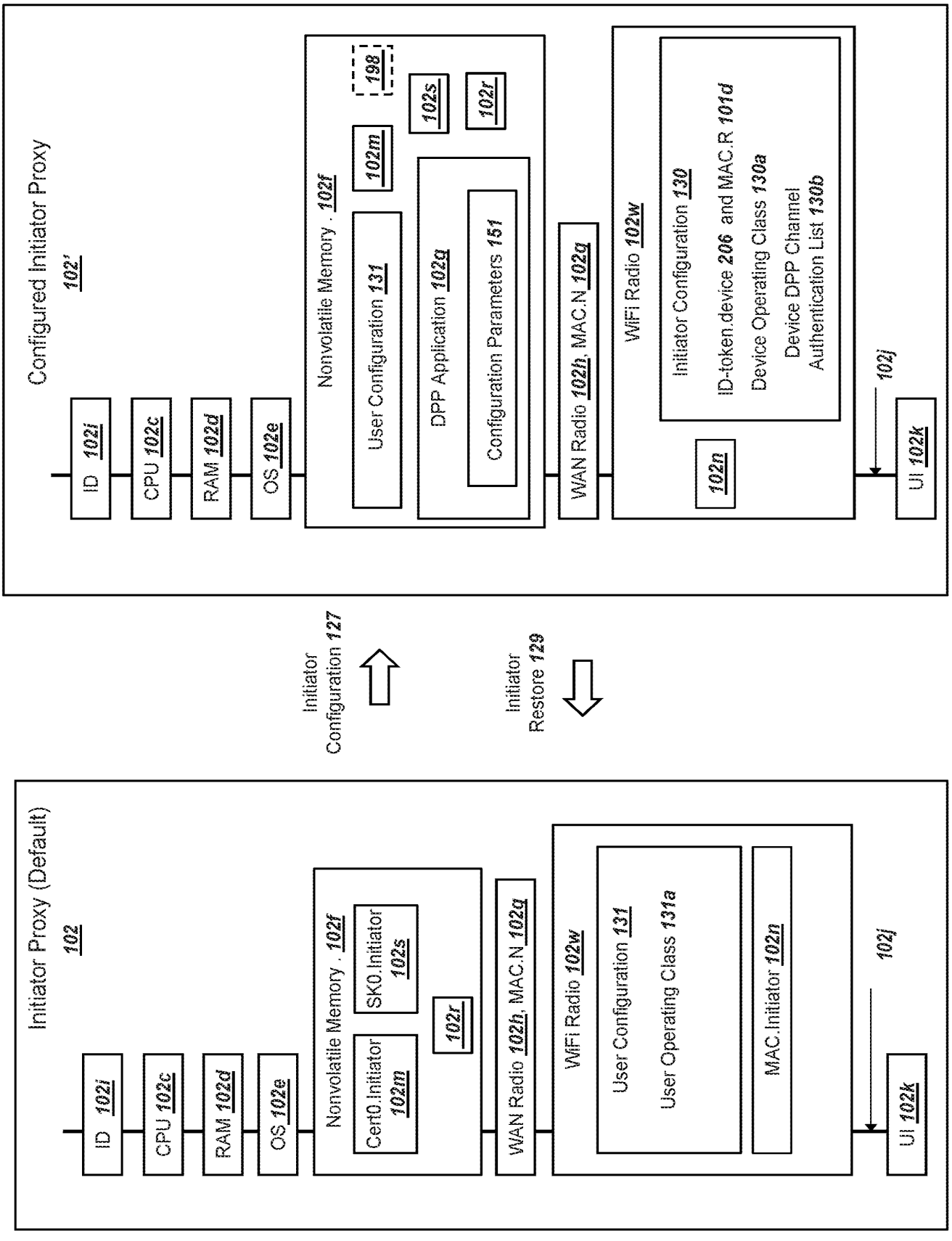
Petitioner Samsung Ex-1004, 1166





Petitioner Samsung Ex-1004, 1167

Figure 1d


Configured Initiator Proxy
102'
ID 102i
CPU 102c
RAM 102d
OS 102e
Nonvolatile Memory . 102f
User Configuration 131
102m
198
DPP Application 102g
102s
Configuration Parameters 151
102r
WAN Radio 102h, MAC.N 102g
WiFi Radio 102w
Initiator Configuration 130
ID-token.device 206 and MAC.R 101d
Device Operating Class 130a
Device DPP Channel
Authentication List 130b
102n
102j
UI 102k
Initiator
Configuration 127
Initiator
Restore 129
Initiator Proxy (Default)
102
ID 102i
CPU 102c
RAM 102d
OS 102e
Nonvolatile Memory . 102f
Cert0.Initiator
102m
SK0.Initiator
102s
102r
WAN Radio 102h, MAC.N 102g
WiFi Radio 102w
User Configuration 131
User Operating Class 131a
MAC.Initiator 102n
102j
UI 102k

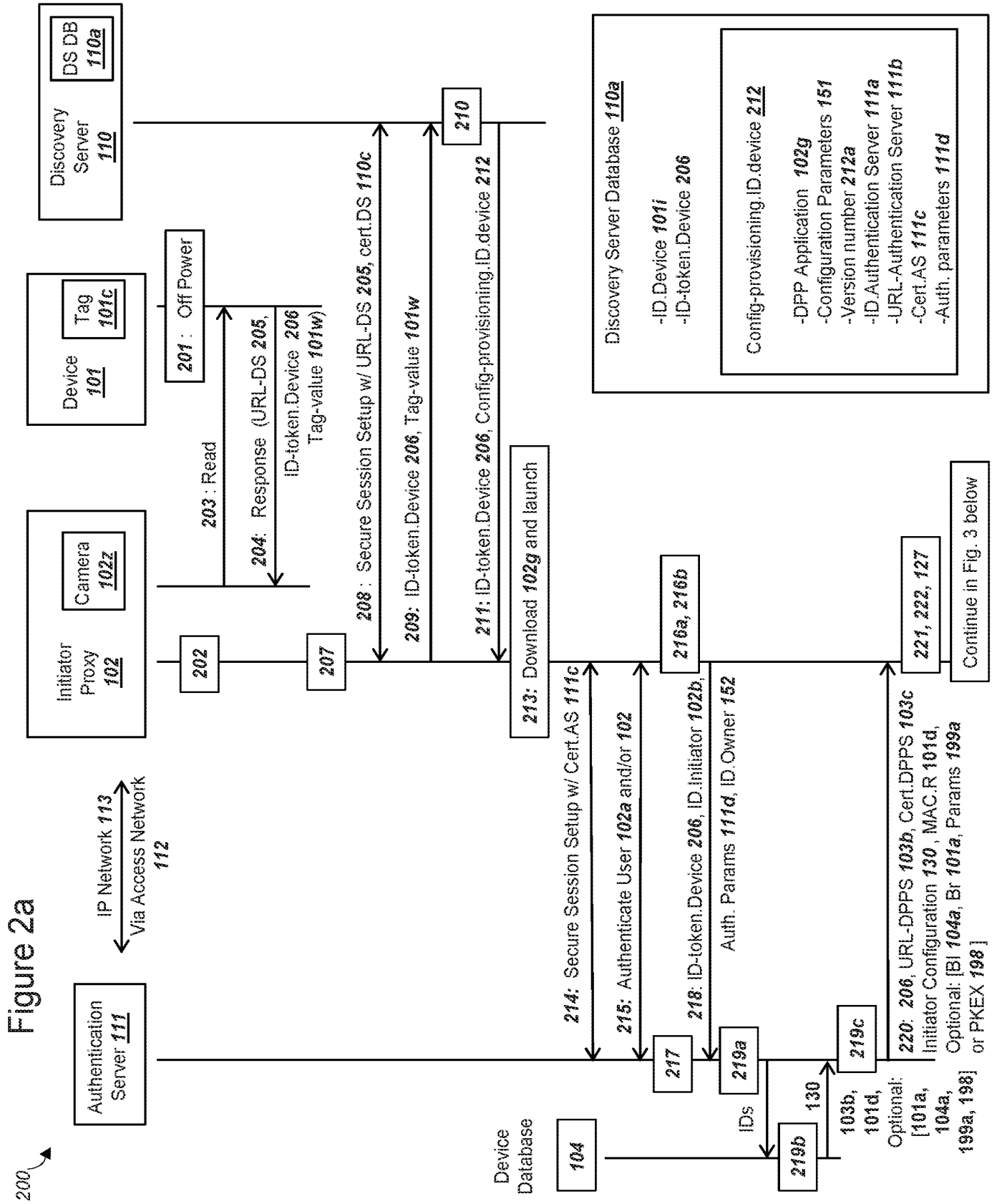


Figure 2a

Petitioner Samsung Ex-1004, 1169

Figure 2b

Device Database **104**

ID-token. Device **206**	Device Identity **101i**	MAC. Responder **101d**	Global Operating Class **130a**	Channel List **130b**	Cryptog. Params **199**
IRLRNZXW	20001	386077720A	81	12, 13	A
QFIXHKNC	20002	386077209	83	2, 7, 9	A
KOWXGVTS	20003	386077400	84	5, 6	A, B
VEMJSLMV	20004	3860773cc	117	40	A, C
WKYPUABJ	20005	386077a09	125	149, 153	A, D

Key Table **104k**

PKEX Key **198**	Responder Bootstrap Public Key **101a**	Initiator Bootstrap Public Key **104a**	Initiator Bootstrap Private Key **104b**
X1	Br-1	Bi-1 (Shared Bi-1 Keys)	bi-3(A), bi-3(B)
X2	Br-2	Bi-1 Keys) bi-1	
NA	Br-3(A), Br-3(B)	Bi-3(A), Bi-3(B)	bi-3(A), bi-3(B)
X4	Br-4(A), Br-4(C)	Bi-4(A), Bi-4(C)	bi-4(A), bi-4(C)
X5	Br-5(A), Br-5(D)	Bi-5(A), Bi-5(D)	bi-5(A), bi-5(D)

(Unique Keys)

Device Database **104**

Device Table **104d**

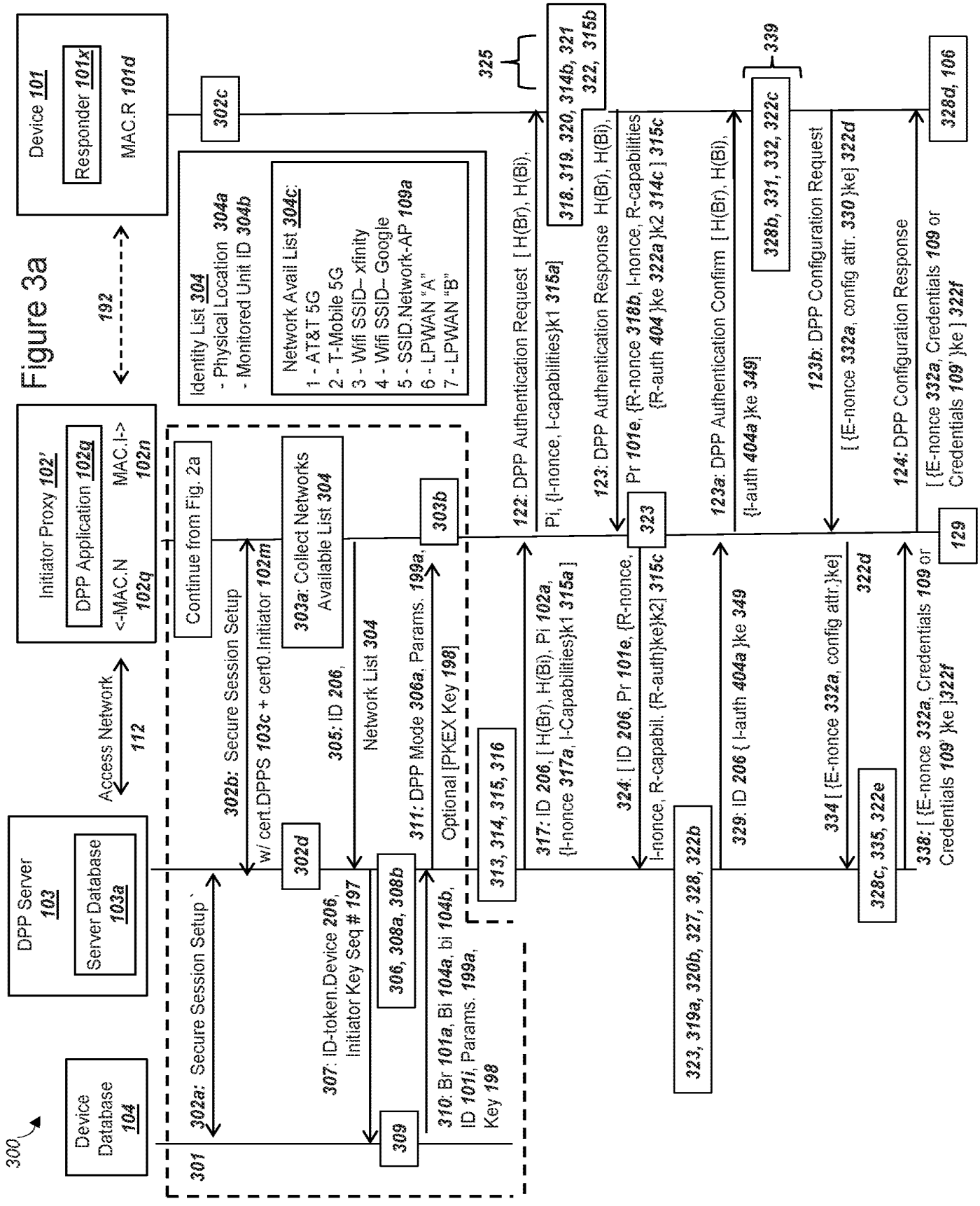
ID-token. Device **206**	Initiator Bootstrap Keys Seq. # **197**	PKEX Key **198**	Responder Bootstrap Public Key **101a**	Initiator Bootstrap Public Key **104a**	Initiator Bootstrap Private Key **104b**
IRLRNZXW	1	X1a	Br-1	Bi-1a	bi-1a
IRLRNZXW	2	X1b	Br-1	Bi-1b	bi-1b
IRLRNZXW	3	X1c	Br-1	Bi-1c	bi-1c
IRLRNZXW	4	X1c	Br-1	Bi-1d	bi-1d

Cryptographic Parameters **199**

Parameters Table **104p**

Set **199a**	Key Length **199b**	Curve **199c**	Hash **199e**	AES-SIV Key Length **199f**	Nonce Length **199g**
A	256	p256	SHA-256	256	128
B	384	p384	SHA-384	384	192
C	512	secp521r1	SHA-512	512	256
D	384	Curve25519	SHA-3	384	192

Petitioner Samsung Ex-1004, 1170



Petitioner Samsung Ex-1004, 1171

Figure 3b

Server Database ***103a***

ID-token. Device ***206***	MAC. Responder ***101d***	MAC. Initiator ***102q***	Initiator Mode ***306a***	Cryptog. Params ***199a***	Initiator Ephem. Public Key ***102a***	Initiator Ephem. Private Key ***102b***	Resp. Ephem. Public Key ***101e***	Initiator Bootstrap Keys Seq. # ***197***	Bootstrap keys ***399***	PKEX Key ***198***	Shared Secret k1 ***314a***	Shared Secret k2 ***314c***	Shared Secret ke ***327a***
IRLRNZXW	38607720A	6cf049080	Fig. 3	A	Pi-1	pi-1	Pr-1	4	Br-1, Bi-1, bi-1	X1	k1-1	k2-1	ke-1
QFIXHKNC	386077209	1c6f654d2	Fig 5a	A	-	-	Pr-2	1	Br-2, Bi-1, bi-1	NA	-	-	-
KOWXGVTS	386077400	7071bce29	Fig 5b	B	-	pi-3	Pr-3	2	{Br-3, Bi-3, bi-3} B	NA	-	-	-
VPMJSLMV	3860773cc	7071bce29	Fig 3	A	Pi-4	pi-4	Pr-4	1	{Br-4, Bi-4, bi-4} A	X4	k1-4	k2-4	ke-4
WKYPUABJ	386077a09	7071bce29	102*	D	-	-	-	1	{Br-5, Bi-5, bi-5} D	X5	-	-	-

●●●

Device Credentials Table ***340***

ID-token.Device ***206***	Selected Network (Primary) ***308aa***	Selected Network (Backup) ***308ab***	Primary Network Credentials ***109***	Backup Network Credentials ***109***
IRLRNZXW	109a - 1	304a - 3	109 - 1	From xFinity
QFIXHKNC	109a - 1	304a - 6	109 - 1	From LPWAN A
KOWXGVTS	304c -2	109a -1	From T-Mobile	109 - 1
VPMJSLMV	304c - 4	109a -2	From Google	109 - 2
WKYPUABJ	109a - 3	304a - 4	109 - 3	From Google

●●●

Petitioner Samsung Ex-1004, 1172

Figure 4a

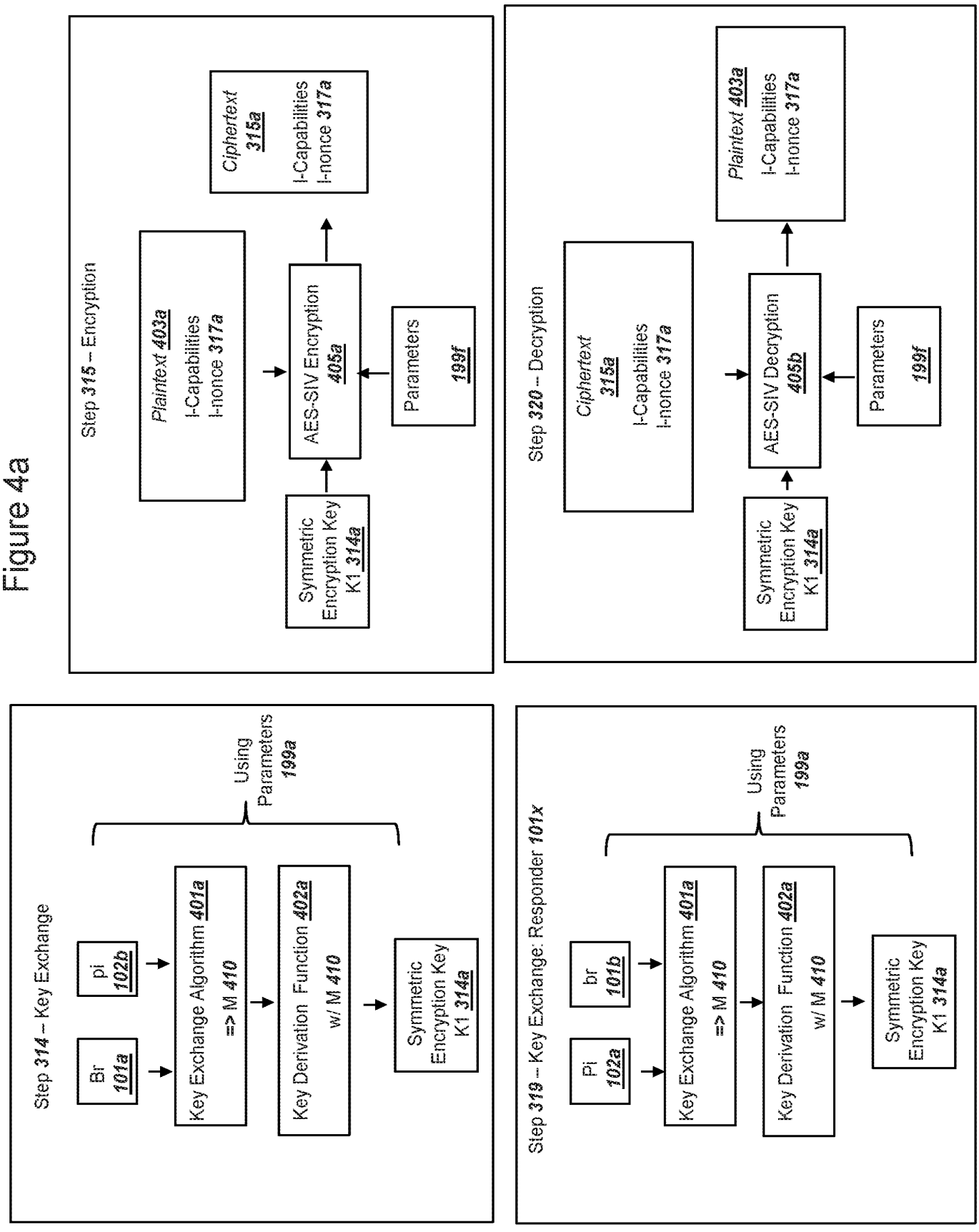

Step 315 – Encryption
Plaintext 403a
I-Capabilities
I-nonce 317a
AES-SIV Encryption 405a
Symmetric Encryption Key K1 314a
Parameters 199f
Ciphertext 315a
I-Capabilities
I-nonce 317a
Step 320 – Decryption
Ciphertext 315a
I-Capabilities
I-nonce 317a
AES-SIV Decryption 405b
Symmetric Encryption Key K1 314a
Parameters 199f
Plaintext 403a
I-Capabilities
I-nonce 317a
Step 314 – Key Exchange
Br 101a
pi 102b
Key Exchange Algorithm 401a
=> M 410
Key Derivation Function 402a
w/ M 410
Symmetric Encryption Key K1 314a
Using Parameters 199a
Step 319 – Key Exchange: Responder 101x
Pi 102a
br 101b
Key Exchange Algorithm 401a
=> M 410
Key Derivation Function 402a
w/ M 410
Symmetric Encryption Key K1 314a
Using Parameters 199a

Figure 4b

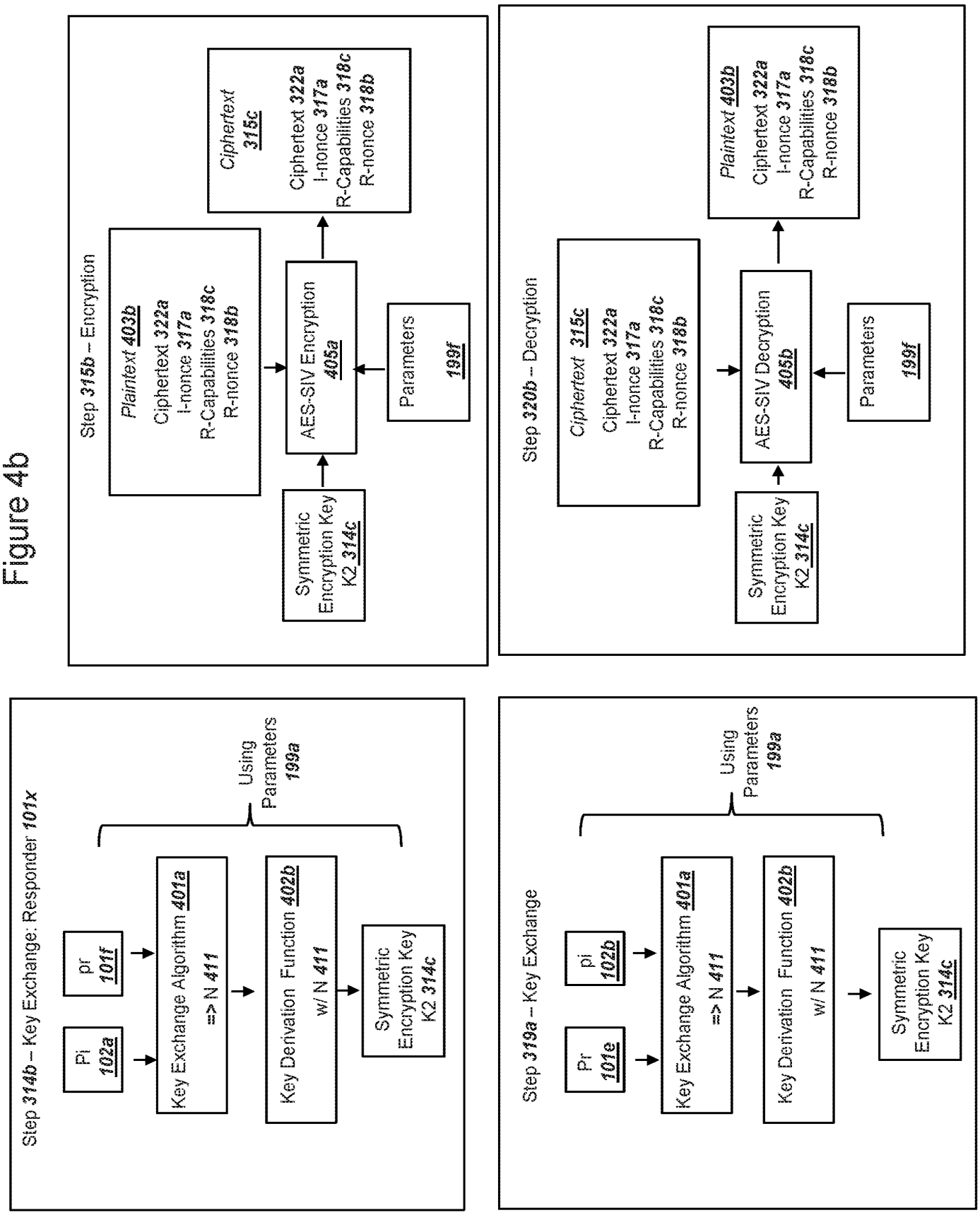
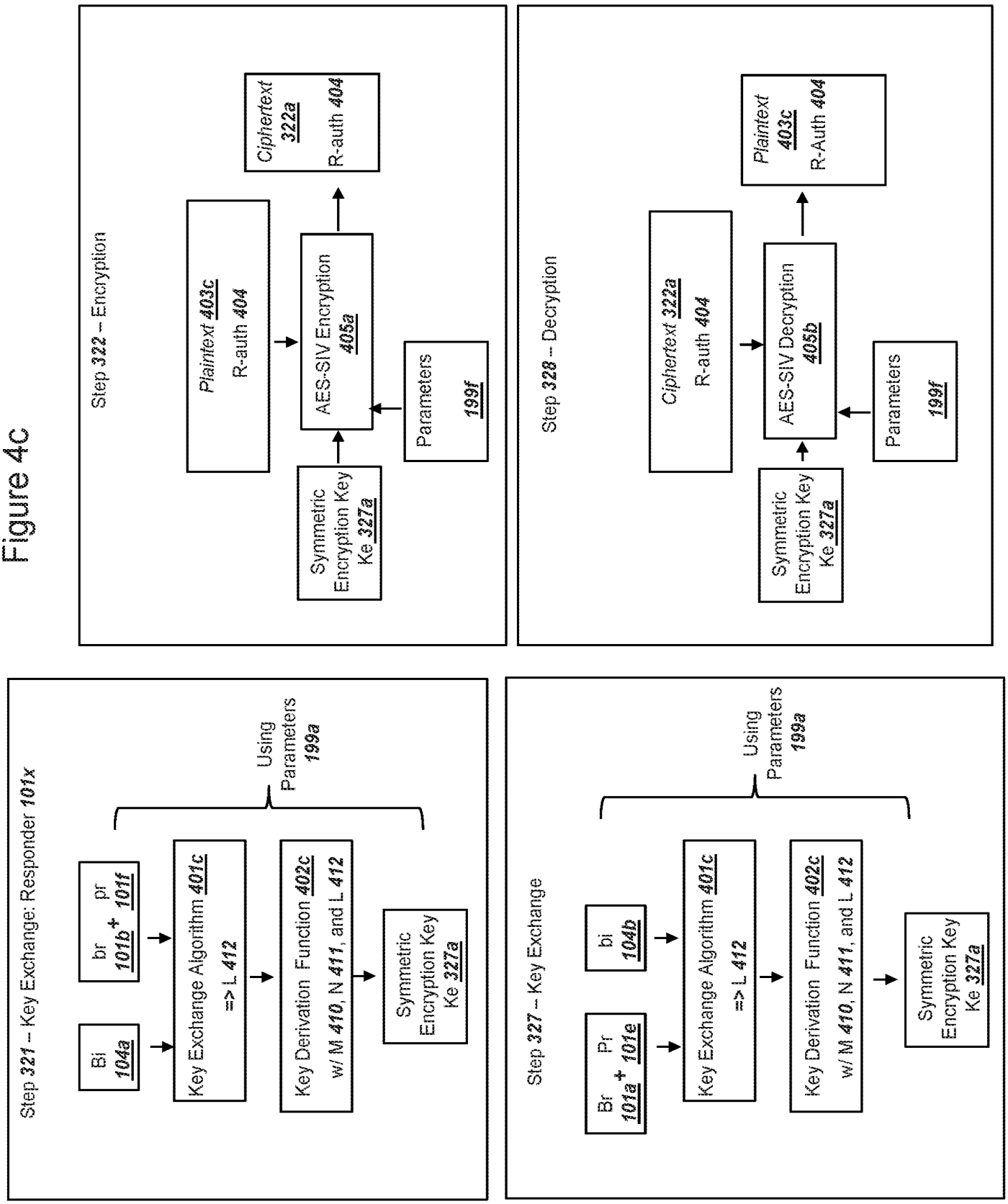
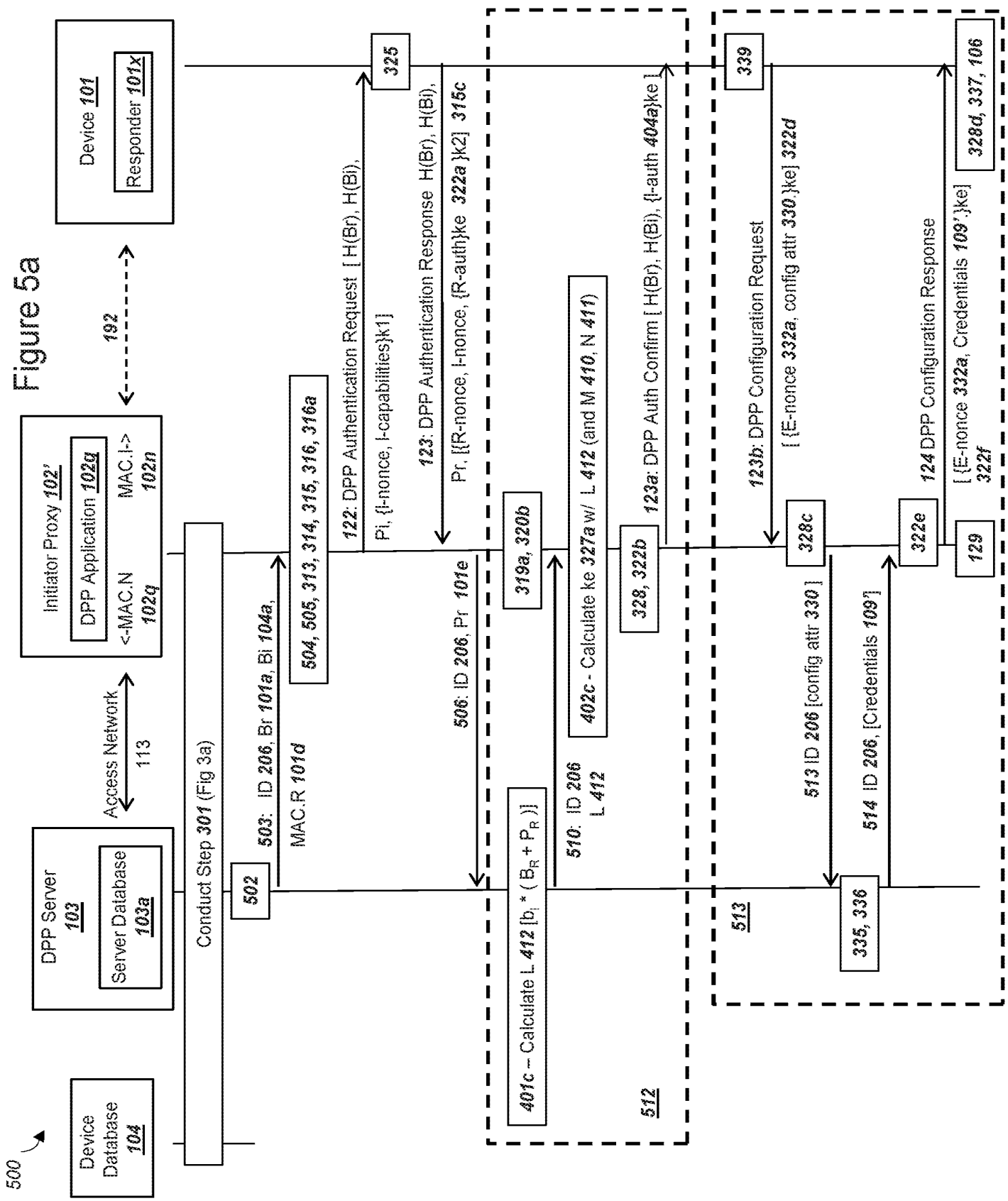

Step 315b – Encryption
Plaintext 403b
Ciphertext 322a
I-nonce 317a
R-Capabilities 318c
R-nonce 318b
AES-SIV Encryption 405a
Parameters 199f
Symmetric Encryption Key K2 314c
Ciphertext 315c
Ciphertext 322a
I-nonce 317a
R-Capabilities 318c
R-nonce 318b
Step 320b – Decryption
Ciphertext 315c
Ciphertext 322a
I-nonce 317a
R-Capabilities 318c
R-nonce 318b
AES-SIV Decryption 405b
Parameters 199f
Symmetric Encryption Key K2 314c
Plaintext 403b
Ciphertext 322a
I-nonce 317a
R-Capabilities 318c
R-nonce 318b
Step 314b – Key Exchange: Responder 101x
Pi 102a
pr 101f
Key Exchange Algorithm 401a
=> N 411
Key Derivation Function 402b
w/ N 411
Symmetric Encryption Key K2 314c
Using Parameters 199a
Step 319a – Key Exchange
Pr 101e
pi 102b
Key Exchange Algorithm 401a
=> N 411
Key Derivation Function 402b
w/ N 411
Symmetric Encryption Key K2 314c
Using Parameters 199a

Figure 4c



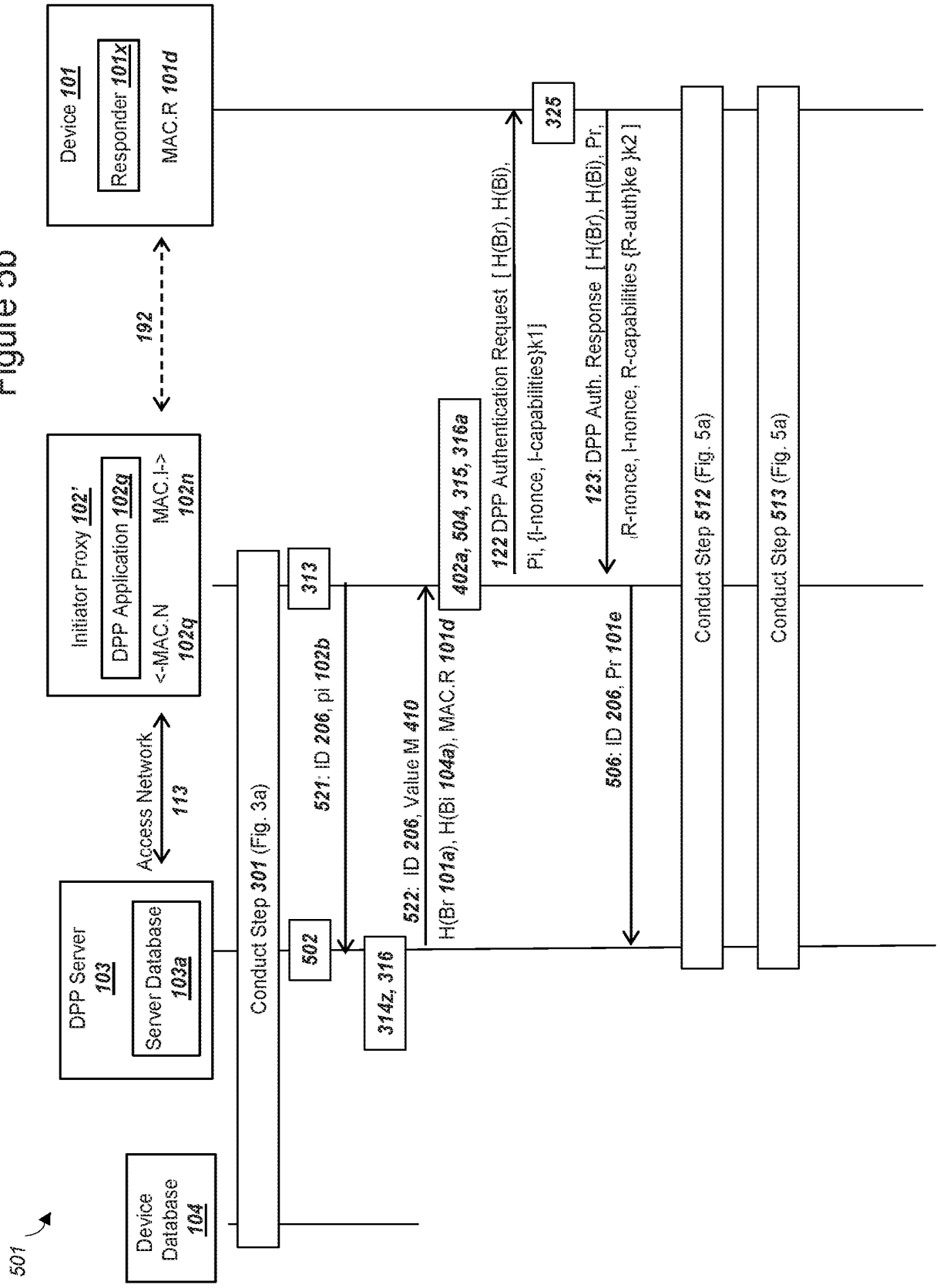
Petitioner Samsung Ex-1004, 1175

Figure 5a



Petitioner Samsung Ex-1004, 1176

Figure 5b



Petitioner Samsung Ex-1004, 1177

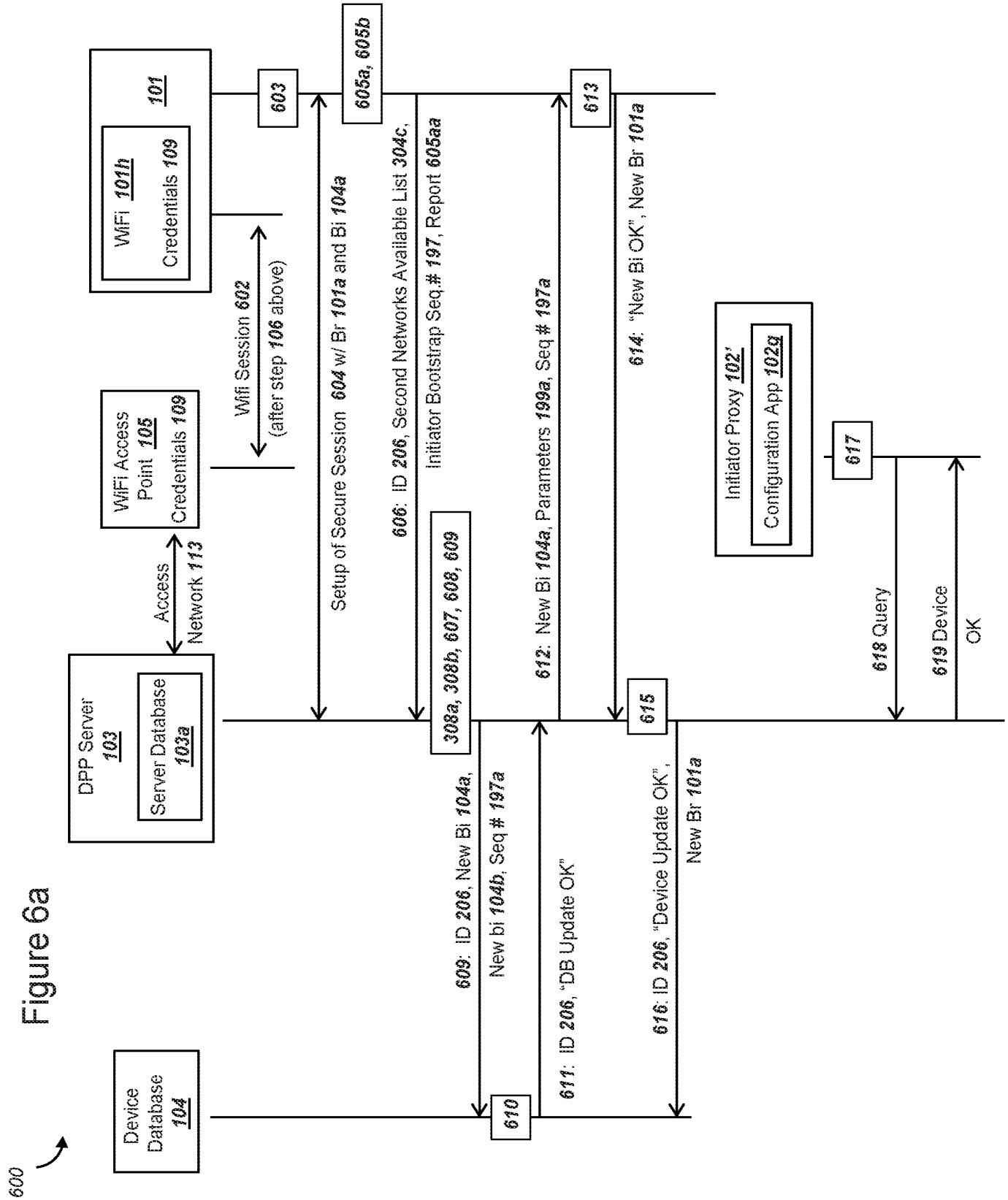


Figure 6a

Petitioner Samsung Ex-1004, 1178

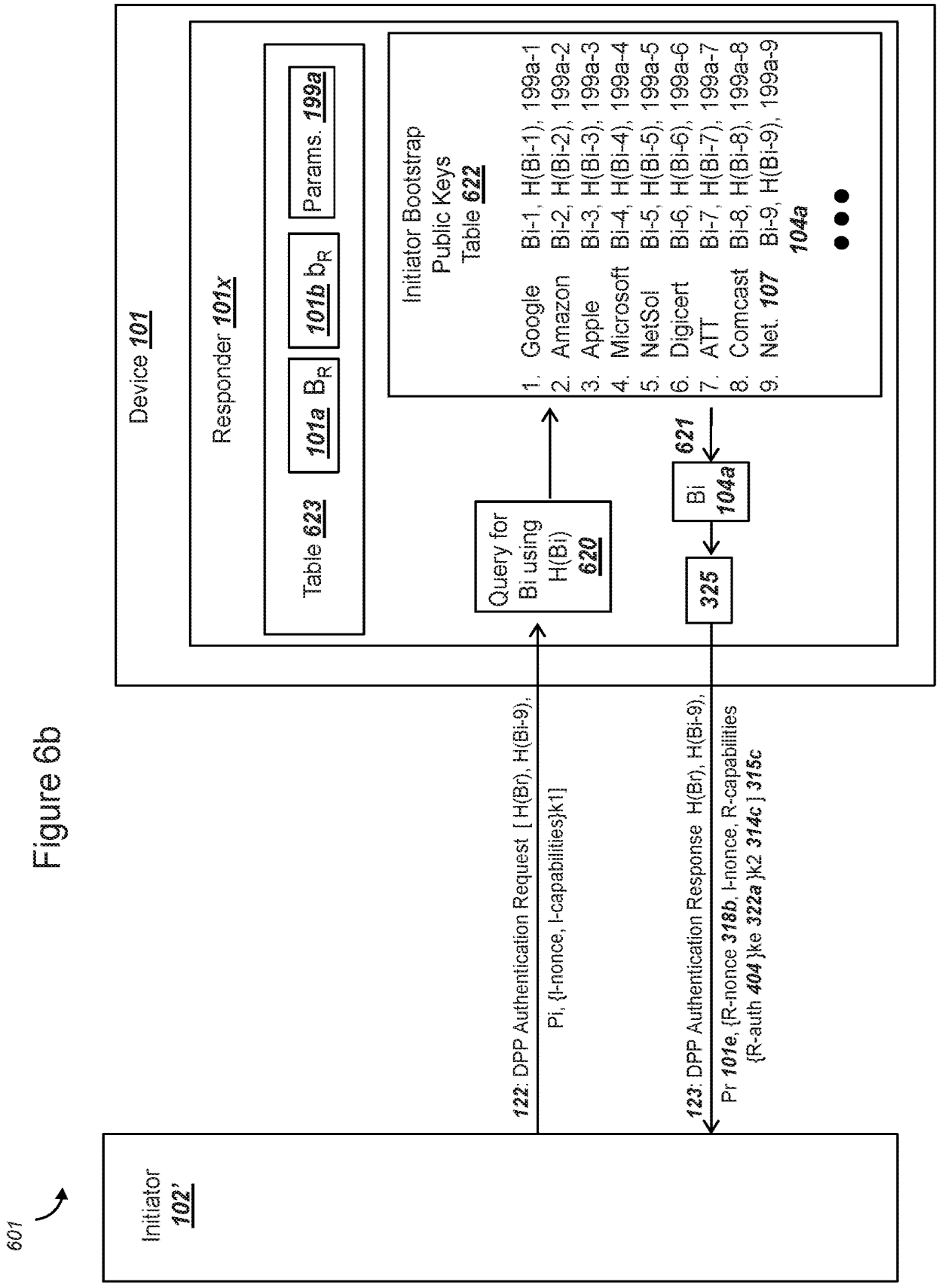
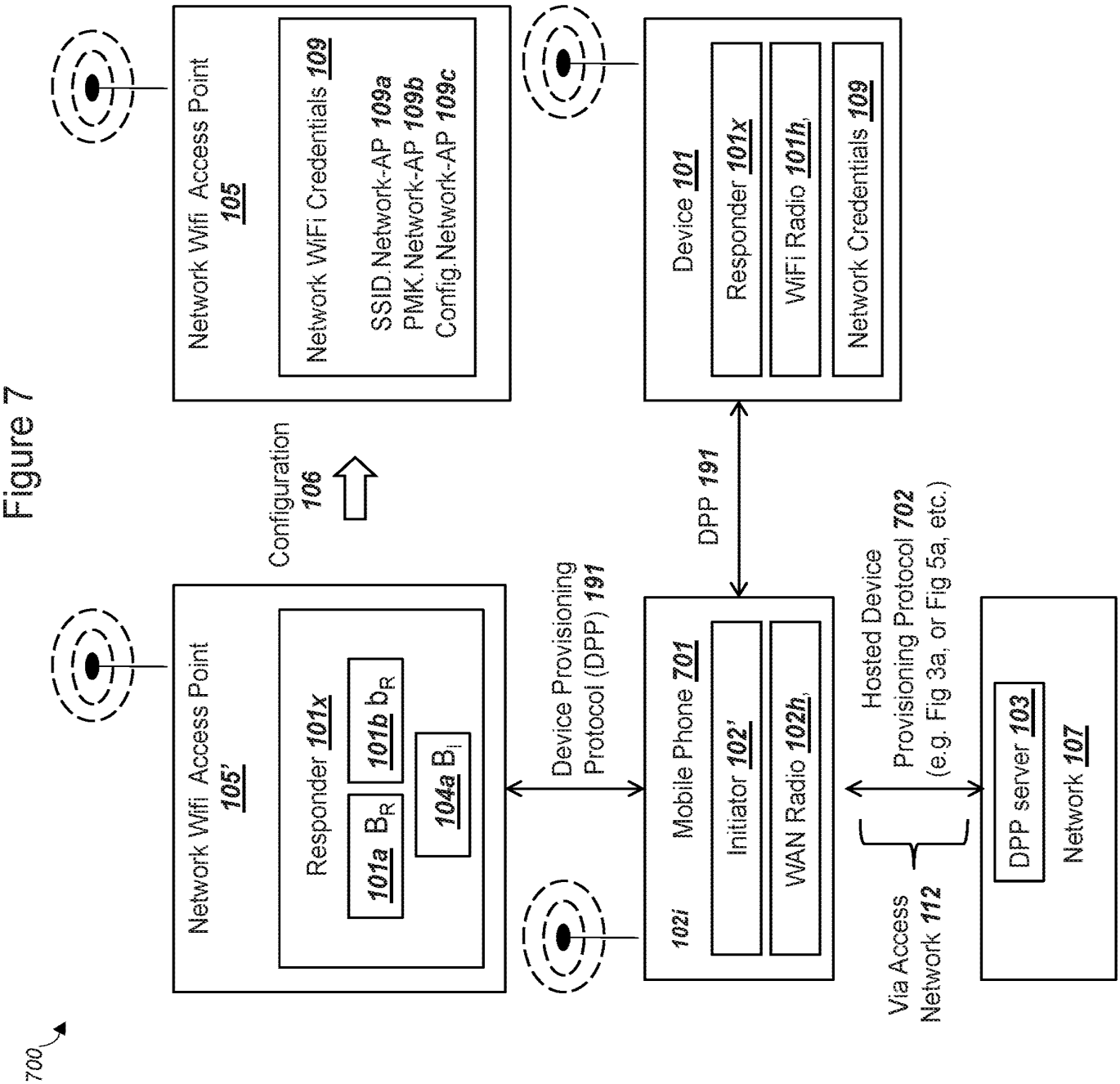


Figure 6b

Petitioner Samsung Ex-1004, 1179

Figure 7



Petitioner Samsung Ex-1004, 1180

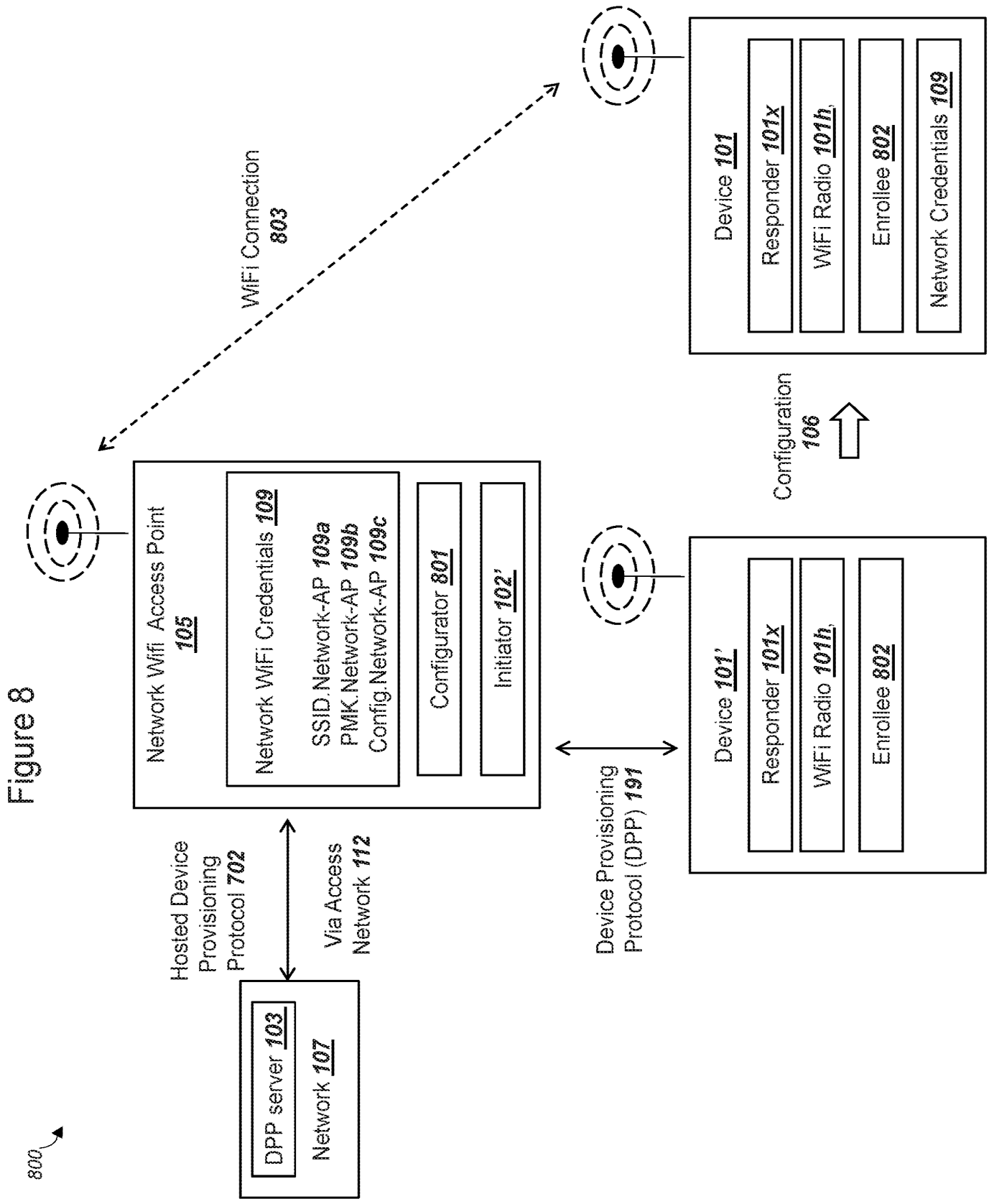


Figure 8

Petitioner Samsung Ex-1004, 1181

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2019/027968

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L9/08 H04L9/32 H04L29/06 H04W12/06 H04W12/08
H04L29/08 H04W12/00 G06F21/62 H04L9/14 H04W12/04
H04W84/12 H04W88/02

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)
H04W H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ANONYMOUS: "Wi-Fi Alliance Device Provisioning Protocol (DPP) - Draft Released for Public Review and Comments v0.0.23", WIFI ALLIANCE, , vol. v0.0.23 26 July 2016 (2016-07-26), pages 1-89, XP009501333, Retrieved from the Internet: URL:https://www.wi-fi.org/downloads-registered-guest/Wi-Fi_DPP_Tech_Spec_v0_0_23.pdf pages 24 and 45-50 ----- -/--	1-55

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
8 July 2019	16/07/2019

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Jardak, Christine

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2019/027968

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Wikipedia: "Hash table - Wikipedia", , 31 January 2018 (2018-01-31), pages 1-17, XP055602768, Retrieved from the Internet: URL:https://en.wikipedia.org/w/index.php?title=Hash_table&oldid=823310127 [retrieved on 2019-07-05] page 1	41-44
A	US 2018/109418 A1 (CAMMAROTA ROSARIO [US] ET AL) 19 April 2018 (2018-04-19) figures 4,7	1-55
A	US 2018/109381 A1 (CAMMAROTA ROSARIO [US] ET AL) 19 April 2018 (2018-04-19) paragraph [0064] - paragraph [0071]; figures 5,6	1-55

Petitioner Samsung Ex-1004, 1183

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2019/027968

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2018109418 A1	19-04-2018	US 2018109418 A1	19-04-2018
		WO 2018075198 A1	26-04-2018
US 2018109381 A1	19-04-2018	CA 3036803 A1	26-04-2018
		CN 109863770 A	07-06-2019
		US 2018109381 A1	19-04-2018
		WO 2018075135 A1	26-04-2018

Form PCT/ISA/210 (patent family annex) (April 2005)

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
02 January 2020 (02.01.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/006162 A1

(51) International Patent Classification:
H04L 9/08 (2006.01) *G06F 21/30* (2013.01)
H04L 9/14 (2006.01)

(21) International Application Number:
PCT/US2019/039380

(22) International Filing Date:
27 June 2019 (27.06.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/691,255 28 June 2018 (28.06.2018) US

(71) Applicant: **IOT AND M2M TECHNOLOGIES, LLC** [US/US]; 807 Davis Street, Unit 2207, Evanston, IL 60201-7104 (US).

(72) Inventor: **NIX, John, A.**; c/o Iot and M2m Technologies, LLC, 807 Davis Street, Unit 2207, Evanston, IL 60201-7104 (US).

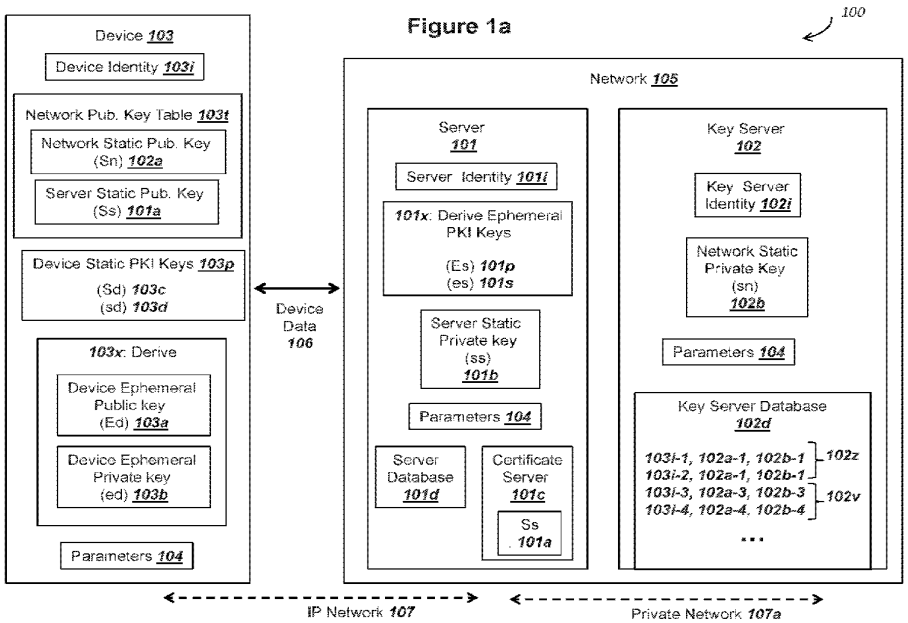
(74) Agent: **THOMSON, Kirsten, L.**; McDonnell Boehnen Hulbert & Berghoff LLP, 300 South Wacker Drive, Chicago, IL 60606 (US).

(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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, 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,

(54) **Title:** ECDHE KEY EXCHANGE FOR MUTUAL AUTHENTICATION USING A KEY SERVER

Figure 1a



(57) **Abstract:** A server can record a device static public key (Sd) and a server static private key (ss). The server can receive a message with (i) a device ephemeral public key (Ed) and (ii) a ciphertext encrypted with key Ki. The server can (i) conduct an EC point addition operation on Sd and Ed and (ii) send the resulting point/secret XO to a key server. The key server can (i) perform a first elliptic curve Diffie-Hellman (ECDH) key exchange using XO and a network static private key to derive a point/secret XI, and (ii) send XI to the server. The server can conduct a second ECDH key exchange using the server static private key and point XO to derive point X2. The server can conduct an EC point addition on XI and X2 to derive X3. The server can derive Kl using X3 and decrypt the ciphertext.

[Continued on next page]

WO 2020/006162 A1

Petitioner Samsung Ex-1004, 1185

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))*

ECDHE KEY EXCHANGE FOR MUTUAL AUTHENTICATION USING A KEY SERVER

CROSS-REFERENCE TO RELATED APPLICATION

This international PCT application claims the benefit of the filing date of U.S. Provisional Patent Application Serial No. 62/691,255, filed June 28, 2018, which is hereby incorporated by reference in its entirety.

BACKGROUND

Technical Field

The present systems and methods relate to conducting an ephemeral elliptic curve Diffie Hellman key exchange (ECDHE) with authentication and multiple parties, and more particularly to communications between a computing device, a server, and a key server over a network in order for the computing device and the server to mutually derive a symmetric ciphering key with mutual authentication.

Description of Related Art

The use of elliptic curve cryptography (ECC) for computing devices has expanded over the past decade and is also expected to continue to grow. Many applications use or propose using ephemeral elliptic curve Diffie Hellman (ECDHE) key exchanges in order to derive a symmetric ciphering key. Prominent examples today include embedded universal integrated circuit cards (eUICCs) also known as embedded SIMs, Transport Layer Security (TLS) version 1.3 from the Internet Engineering Task Force (IETF), and the Device Provisioning Protocol (DPP) from the WiFi Alliance ™. Other examples are expected in the future as well, such as the use of ECDHE in order to protect the Subscription Permanent Identifier (SUPI) for 5G mobile networks, where the SUPI is equivalent to an International Mobile Subscriber Identity (IMSI). ECDHE can be considered a subset of elliptic curve Diffie-Hellman key exchanges (ECDH), where ECDHE key exchanges use at least one ephemeral or short-term elliptic curve PKI key pair. Applications use ECDHE key exchanges in order for two nodes to mutually derive a symmetric ciphering key and a message authentication code (MAC) key using a key derivation function. The symmetric ciphering key can subsequently be used with a symmetric ciphering algorithm such as the Advanced

Petitioner Samsung Ex-1004, 1187

Encryption Standard (AES) and the MAC key can be used to verify message integrity. In this manner, secure communication can be established between two nodes.

ECDHE key exchanges depend on a first node deriving a first ephemeral private and public key pair and a second node deriving or using a second private and public key, where the public key infrastructure (PKI) keys use a common elliptic curve. The elliptic curve can be specified in parameters that define a named curve such as secp256r1 (p256), secp256k1, secp385r1, etc., and many other possibilities exist as well for named curves, include curve 25519. ECDHE key exchanges have multiple benefits over older generation technology such as Diffie Hellman key exchanges. With ECDHE, elliptic curve cryptography can be utilized with shorter keys and faster processing times compared to previous technology, for the equivalent level of security or bit length of keys. For example, a 256 bit ECC PKI key pair can be used to obtain a comparable level of security as that obtained from using a 3072 bit RSA based PKI key pair. Calculation or processing time for conducting an ECDHE key exchange can also be faster than a traditional Diffie Hellman key exchange for the same level of security, as defined by the resulting key length of a derived shared secret from the key exchange.

Although the use of ECDHE key exchanges is growing rapidly, improvements can be made for ECDHE key exchanges in order to further enhance security and also leverage existing keys that may be recorded by the nodes participating in an ECDHE key exchange. As one example, an ECDHE key exchange as contemplated for (a) the exemplary applications and standards from two paragraphs above do not normally (b) provide authentication of either node. Separate steps than an ECDHE key exchange have to be conducted in order to authenticate endpoints, such as using an elliptic curve digital signature algorithm (ECDSA) with static or long-term ECC PKI keys recorded by the nodes. ECDSA algorithms also have challenges, where the reuse of a value k for two different signatures can reveal the private key. As another example and related to the authentication issue above, an ECDHE is susceptible to "man in the middle" attacks, where an intermediate node or different node than the intended node can perform the ECDHE key exchange instead of the intended node. Thus, although ECDHE can securely establish a symmetric ciphering key for confidentiality of data communications, the confidentiality could be established with a party or node that is not the intended recipient of the confidential communications. Consequently, a need exists in the art for the intended two nodes for confidential communications to use an ECDHE key exchange in a manner where the two nodes can be mutually authenticated.

A primary goal of ECDHE key exchanges is also to obtain forward secrecy, where an ECDHE key exchange can periodically be re-conducted in order to rotate or re-establish a new symmetric ciphering key. In this manner, if a private key is compromised then only the subset of historical data encrypted using the compromised private key is subject to decryption by third

Petitioner Samsung Ex-1004, 1188

parties, and other communications using a different private key can remain secured. An authenticated ECDH key exchange can be conducted using at least one static PKI key pair (e.g. not an ephemeral key exchange with ephemeral PKI keys), but without the benefits of forward secrecy. A need exists in the art where two parties can conduct an authenticated ECDHE key exchange (e.g. by using ephemeral PKI keys) in order to obtain the benefits of forward secrecy.

The use of ECDH key exchanges (e.g. with at least one static PKI key pair) is also subject to greater security risks over time, where repeated use of one static PKI key pair is subject to cryptographic analysis and "leakage" of equivalent bits of security over time. Further, the use of ECDH key exchanges with one static PKI key pair and one ephemeral PKI key pair is more subject to risks of attacks from specifically chosen ephemeral PKI keys, such as ephemeral public keys that are either (i) not on the curve or (ii) specifically selected to expose information about the static private key. Thus, (a) repeated use of ECDHE key exchanges over time with different ephemeral PKI keys, compared to (b) using an ECDH key exchange with one static PKI key pair will result in greater security regarding confidentiality of communications. A need exists in the art where the greater security of ECDHE key exchanges can be obtained while also using static ECC PKI keys recorded by at least one of the nodes deriving a symmetric ciphering key using the ECDHE key exchange.

Many applications or new standards such as TLS version 1.3, DPP version 1.0 and 5G network standards from the 3rd Generation Partnership Project (3GPP) implement ECDHE key exchanges in order to quickly establish confidentiality early in the communications between two nodes. As noted above, a traditional ECDHE key exchange establishes confidentiality without authentication, and authentication must be obtained through other means, such as ECDSA or DSA, message digest, etc. However, the nodes participating in communications with the above standards typically have access to other, secure and previously recorded PKI keys besides the ephemeral PKI keys derived in order to conduct the ECDHE key exchange. A need exists in the art for a node to use the previously recorded PKI keys for (a) a new ECDHE key exchange in order to establish an authenticated key exchange without (b) the risks of ECDH key exchanges for static PKI keys as discussed above.

Solutions have been proposed in the art for an authenticated Diffie-Hellman or elliptic curve Diffie-Hellman key exchange using ephemeral keys and static keys. Blake-Wilson et al in the paper "Key Agreement Protocols and their Security Analysis", which is herein incorporated by reference, propose the use of both long-term static keys and short-term ephemeral keys with a DH key exchange in order to conduct the key exchange in an authenticated manner in order to address some needs in the art mentioned above. Likewise, the Internet Engineering Task Force (IETF) proposes the use of elliptic curve ephemeral and static PKI keys in the "Request for Comments"

Petitioner Samsung Ex-1004, 1189

(RFC) 5753 document "Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)", which is also hereby incorporated by reference.

However, the methods described for Blake-Wilson, RFC 5753, and related systems depend on (a) the recipient/responder of an ephemeral ECC public key from a sender/initiator to (b) also to record or operate with the static private ECC key corresponding to the static public key recorded by the sender. This can reduce scalability of a system with (i) a plurality of sendors/initiators and (ii) a plurality of recipients/responders receiving ephemeral ECC public keys for ECDHE key exchanges, since each recipient/responder also needs to record and operate on the static ECC private key corresponding to the static ECC public key recorded by the sender/initiator. The overall security of a system can be decreased for a system of potentially millions of devices and several servers, where the servers need to record server static private ECC keys corresponding to server static public ECC keys recorded by devices. A need exists in the art for (a) a recipient/responder to support authenticated ephemeral ECDH key exchanges without (b) the recipient/responder also recording the static ECC private key corresponding to the static ECC public key recorded by the sender/initiator.

Other proposed solutions contemplate (i) the use of both ECDHE key exchanges and (ii) pre-shared PKI key pairs between two nodes for conducting an ECDHE key exchange with mutual authentication. An example would be the draft IETF RFC document "ECDH-based Authentication using Pre-Shared Asymmetric Keypairs for (Datagram) Transport Layer Security ((D)TLS) Protocol version 1.2", which is hereby incorporated by reference. Benefits of this and similar proposed systems is that mutual authentication and forward secrecy can be obtained. However, the system as presented requires that the server also record and operate with a static private key corresponding to the static public key recorded by the client (which could be a device).

Recording a static private key in a server communicating with devices or clients through the public Internet has security drawbacks, especially for single server public keys recorded and used with devices designed to be operational for a decade or longer. A need exists in the art to support mutual authenticated ECDHE key exchanges where the server communicating with the client does not record or operate with a static private key corresponding to the server static public key recorded by the client. A need exists in the art where a plurality of different server static public keys could be recorded by a plurality of different devices, such that compromise of a single server static private key only potentially compromises a single device.

The draft IETF RFC discussed in the paragraph above also solved problems for resource constrained devices, where the use of digital signature algorithms can be bypassed because ECDHE key exchanges that also use static public keys can provide mutual authentication (thereby bypassing the need for recording an operating digital signature algorithms). However, a significant drawback

Petitioner Samsung Ex-1004, 1190

for devices deployed for a decade or longer is (a) the risk that server static private key is compromised, or (b) the server recording or having access to the server static private key is no longer authenticated. For the system in the draft IETF RFC and related systems, there is no method or contemplation for a "key revocation" both (i) after deployment of devices/clients with a server static public key and (ii) from a "parent" authority over the server (but without direct control of the server).

As one example for needs in the art, a device owner could own but not operate a device, and an example could be a leasing or rental company, where a user under contract with the device owner may operate the device. The device owner may not operate or control a server the device/client communicates with, and the server could be specified by the user. If the device owner wishes to terminate service for the device, the device owner may not be able to directly stop communications between the deployed device/client and the server. A need exists in the art for a device to record an additional static public key (in addition to the server static public key) that are required for use with an ECDHE key exchange, in order for a "parent authority" to participate in the key exchange and control the device, or effectively perform revocation of the ability to communicate with a server. A need exists in the art for the server communicating with the device to not record or operate with (a) the corresponding additional static private key for the (b) additional static public key recorded in the device.

Although the use of a key server with DH key exchanges has been deployed for different networking environments, they have not been extended to conducting an ECDHE key exchange. The company Cloudflare ® has promoted the use of a key server for TLS version 1.2 operations and DH key exchanges, where the private key for a network could be operated by a network to either (i) conduct an asymmetric decryption of a master secret sent by a client or (ii) perform a digital signature operation for a Diffie Hellman key exchange. As of mid-2018, the product is called "Keyless SSL". This, and other examples of conducting networking, transport, or application layer security have not suggested how a private key for a network could be recorded and used by a key server for conducting a mutually authenticated ECDHE key exchange. Additional steps using a different cryptographic framework (e.g. mutually authenticated ECDHE key exchanges) than conventional technology (e.g. "one-way" or server authenticated key exchanges) would need to be conducted in a coordinated manner between the server and a key server. There is no teaching or suggestion for these additional steps with conventional technology using key servers.

A need exists in the art for using a key server recording and operating with private keys in a manner that supports a separate server conducting an ECDHE key exchange with mutual authentication. A need exists in the art for operating a server in an insecure location, such that

Petitioner Samsung Ex-1004, 1191

static private keys associated with the server (or the network operating the server) can be recorded in a different, secured location than the server. In addition, with the continued development of quantum computers, a need exists in the art for a device and a server to securely derive a symmetric ciphering key in a manner that is resistant to quantum computers feasibly solving elliptic curve Diffie Hellman problems and/or discrete logarithm problems.

Many other examples exist as well for needs in the art to conduct an ECDHE key exchange in a secure manner where the two nodes can be mutually authenticated, and the above are examples are just a few and intended to be illustrative instead of limiting.

Petitioner Samsung Ex-1004, 1192

SUMMARY

Methods and systems are provided for a server to conduct a mutually authenticated ephemeral elliptic curve Diffie-Hellman key exchange (ECDHE) with a device and using a key server. The device and the server can record and operate a set of compatible values and algorithms for a key pair generation algorithm, an ECDH key exchange algorithm, a key derivation function, a symmetric ciphering algorithm, and a random number generator, and a set of cryptographic parameters.

The device can comprise a computing device with a network interface to communicate with the server via an IP network. The device can comprise a transducer device for operating a transducer and communicating the transducer data with the server via secured communications. The device can comprise a device for "the Internet of Things", a mobile phone, a tracking device, a security system, a remote server on user premises, a module, a WiFi client, a WiFi access point, or similar devices. The server can comprise a computing device with a network interface to communicate with (i) the device via the IP network and (ii) the key server via a private network. The device can record a domain name service (DNS) name or uniform resource locator (URL) for the server. The key server can record a network static private key. The server can record and operate a server database. The device can be one of a plurality of different devices communicating with the server.

Before distribution to an end user of the computing device, a device manufacturer or a device distributor or a device owner could record a set of data in nonvolatile memory for the device. In addition to regular operating data and programs for the device, such as an operating system and a transducer driver, the data recorded in device before distribution could include (i) a network static public key, (ii) a server static public key, (iii) a device static public key, (iv) a set of cryptographic parameters associated with the static public keys, and (v) a device identity for the computing device. For a first exemplary embodiment, the network static public key can be unique for the device and not shared with other devices. For a second exemplary embodiment, the network static public key can be shared across a set of devices and thus the network static public key would not be uniquely recorded in an individual device, but the network static public key could be recorded in a set of devices. For a third exemplary embodiment, the server static public key can be unique for the device and not shared with other devices. For a fourth exemplary embodiment, the server static public key can be shared across a set of devices and thus the server static public key would not be uniquely recorded in an individual device, but the sever static public key could be recorded in a set of devices.

Petitioner Samsung Ex-1004, 1193

After power up and/or connecting with the IP network, the device can use the random number generator, the cryptographic parameters, and the key pair generation algorithm to derive a device ephemeral private key and a device ephemeral public key. The device can conduct a first ECDH key exchange using the device static private key, the device ephemeral private key, the server static public key, and the network static public key. The output from the first ECDH key exchange can comprise a shared secret or point X3. The device can derive a symmetric ciphering key using shared secret X3 and a key derivation function. The device can generate a first random number (different than the random number which could comprise the device ephemeral private key). The device can process a first ciphertext that contains or includes at least the first random number, where the ciphertext is encrypted with the derived symmetric ciphering key. The device can send the device ephemeral public key, the first ciphertext, identification information, and the associated set of cryptographic parameters to the server in a first message using the recorded DNS name or a URL for the server.

The device can also optionally send a device identity or a secure hash value for the network static public key to the server, in order for the server to identify the device or set of devices. Identification information for the device in the first message can comprise the device identity or secure hash value, although other values for the device identification information are possible as well. In some exemplary embodiments, the identity of the device and also the secure hash value can be omitted from the first message and the server identifies the group of devices by a particular IP address and port number and/or URL and/or DNS name used by the server and receiving data from devices. In other words, without identifying data for the device in the first message, (X) a subset of devices sending data to the server where the server uses a particular name, URL, or IP address and/or port number can be identified by (Y) the server receiving data from the devices using the IP address and port number and/or URL and/or DNS name.

The server can receive the first message and process the first message. The server can use the received set of cryptographic parameters to conduct an ECC point validation step to verify that the received ECC public key comprises a point on a named curve specified by the set of cryptographic parameters. The server can also verify that the received set of cryptographic parameters is valid or acceptable, such as using an acceptable named curve and also supporting a sufficient level of security such as an exemplary 256 bit long keys, although other values for key lengths are possible as well. The server can use identification information for the device from the first message to select PKI keys for the device from a server database. The selected PKI keys can comprise a device static public key and a server static private key. The server can also select a key server for use with communications and authentication with the device. The server can also select a device identity from the server database using the identification information for the device, for

Petitioner Samsung Ex-1004, 1194

embodiments where the identification information in the first message comprises other information than a device identity for the device.

In order to select a key server, the server could use several different methods based on the identifying information of the device in the first message. The identifying information from the first message for the device used to select a key server could comprise any of the following in the first message: (i) an optional identity of the device, (ii) an optional secure hash value over a unique public key recorded by the device and the server database, (iii) the use of a particular set of cryptographic parameters, where the set of cryptographic parameters are associated with a particular key server, or (iv) the server can operate such that a particular URL or IP address and port number as a destination address is mapped to a particular key server.

The server can operate or be associated with a server database in order to record data for the server communicating with a plurality of different devices, such that different keys for different devices could be tracked by the server. In exemplary embodiments the first message is received with a random number generated by the device (in the first ciphertext) and also a source IP address and port number, and the server records the source IP address and port number for the first message in the server database. The server can conduct a first elliptic curve (EC) point addition operation using the received device ephemeral public key and the selected device static public key to derive a point X0 using the set of cryptographic parameters.

The server can record the name, address, or URL for a key server and communicate with the key server through a private network. The server can establish a secure session with the key server. The server can send the key server the point X0 in a second message. The second message can optionally include the set of cryptographic parameters, although the device, server, and key server could agree on the use of a particular set of cryptographic parameters before the device sends the first message.

The key server can receive the second message from the server over the secure connection. The second message can include point X0, the set of cryptographic parameters, and the device identity. The key server can select or read the network static private key using the second message received from the server (including possibly identifying information of the device in order to select a specific network static private key for the device). The network static private key can comprise a secret key corresponding to the network static public key recorded by the device. The key server can conduct a second ECDH key exchange using (i) the selected network static private key for the device and the received point X0 and (ii) the set of cryptographic parameters in order to derive a second shared secret, which can comprise a point X1. The key server can send a response to the second message in the form of a third message to the server, where the third message includes the derived point X1 and a device identity.

Petitioner Samsung Ex-1004, 1195

The server can receive the third message with the point X1 and the device identity. The server can conduct a third ECDH key exchange using the set of cryptographic parameters, the derived point X0 and the selected server static private key in order to derive a point X2. The server can conduct a key derivation step in order to mutually derive the symmetric ciphering key which was also previously derived by the device. The key derivation step used by the server can comprise (i) an EC point addition operation on the derived point X2 and the received point X1 from the key server, which results in the point X3 also derived by the device from the first ECDH key exchange, and (ii) input of the point X3 into the same key derivation function used by the device. The output of the key derivation function (KDF) in the key derivation step can comprise the mutually derived symmetric ciphering key. In exemplary embodiments the derived symmetric ciphering key can comprise a first portion for encrypting and decrypting data sent from the server to the device and a second, different portion for encrypting and decrypting data from sent from the device and to the server.

The server can then use a decryption step with the symmetric ciphering key in order to read plaintext from the first ciphertext received in the first message from the device. The plaintext read can include the first random number generated by the device. The successful decryption of the first ciphertext can indicate that the device is authenticated, since the server could only feasibly decrypt the ciphertext from the first message if the device conducted the first ECDH key exchange with the device static private key which corresponds to the device static public key used by the server and key server in the second and third ECDH key exchanges (in the form of point X0). The plaintext from the first message can comprise a first random number generated by the device and also optionally include device data.

The server can encrypt (i) a second random number generated by the server, (ii) the first random number received by the device, and (iii) a response to the device data received in the first message. The encryption step used by the server can include encryption using the mutually derived symmetric ciphering key. The output of the encryption step can comprise a second ciphertext. The server can send the device a fourth message, where the fourth message includes the device ephemeral public key and the second ciphertext.

The server could conduct additional steps such as (i) deriving server ephemeral private and public keys, (ii) deriving a second symmetric ciphering key using at least the device ephemeral public key and server ephemeral private key (possibly also with the server static private key and device static public key), and (iii) encrypting a third ciphertext with the second symmetric ciphering key. The third ciphertext could be internal or external to the second ciphertext. In exemplary embodiments, the fourth message includes at least the second ciphertext and may optionally also include the third ciphertext.

Petitioner Samsung Ex-1004, 1196

The device can receive the fourth message from the server and take steps to process the message. The device can decrypt the second ciphertext using the symmetric ciphering key derived by the device using the first ECDH key exchange. The device can read the plaintext from the second ciphertext. The device can confirm that the plaintext from the second ciphertext received in the fourth message includes the first random number. Successful decryption of the second ciphertext and reading the first random number from the second ciphertext, which could be sent by the device in the first ciphertext, can confirm that the server and/or network is authenticated.

In other words, the server and/or network could only feasibly successfully encrypt the second ciphertext if the server and/or network record and operate with both the server static private key and the network static private key which correspond to the server static public key and the network static public key recorded by the device and used in the first ECDH key exchange. The plaintext from the second ciphertext received by the device can also include the response from the server. The device can process the response and perform steps or instructions based on the plaintext response from the server.

The systems and methods described above can also be used with various implementations for the computing device and the server. A 5th generation or 6th generation wireless WAN network, such as based on standards from 3GPP, could utilize the steps above in order to conduct an ECDHE key exchange with mutual authentication and a key server. For this embodiment, the computing device could comprise a wireless device or wireless terminal, including a mobile phone or smart phone. The server could comprise a "g Node B" for "next generation node b", or gNb, which provides equivalent functionality of a base transceiver station and manages the radio-frequency communications with the wireless device. The key server could comprise a secured server (i) operating within the authentication function of a wireless network or (ii) associated with the authentication function of a wireless network for a mobile network operator. For the embodiment in this paragraph, the cryptographic parameters could comprise the values for curve 25519, although other ECC curves could be utilized as well.

The systems and methods described above can also be used with a device provisioning protocol. The computing device as described above can comprise an initiator according to the Device Provisioning Protocol specification version 1.0 from the WiFi Alliance ®. The server can comprise a responder according to the same specification. Subsequent versions of the specification can utilize the methods and systems described herein as well. The device can receive and record the network static public key in the form of a responder bootstrap public key. A key server could record the network static private key in the form of a responder bootstrap private key.

The responder/server can receive the first message with (a) the device ephemeral public key from the initiator/device along with (b) a secure hash value of the responder bootstrap public

Petitioner Samsung Ex-1004, 1197

key, and (c) an initial ciphertext. The responder/server can use the secure hash value of the responder bootstrap public key to select the key server for the device. The responder/server can forward the device ephemeral public key (as a point X0 which also combined the initiator bootstrap public key) to the selected key server. The key server can (i) conduct the second ECDH key exchange with the point X0 and the responder bootstrap private key and (ii) send the shared secret point X3 to the responder/server. The responder/server can use the second shared secret point X3 to decrypt the initial ciphertext received with the first message.

These as well as other aspects and advantages will become apparent to those of ordinary skill in the art by reading the following detailed description, with reference where appropriate to the accompanying drawings.

Petitioner Samsung Ex-1004, 1198

BRIEF DESCRIPTION OF THE DRAWINGS

Various exemplary embodiments are described herein with reference to the following drawings, wherein like numerals denote like entities.

Figure 1a is a graphical illustration of an exemplary system, where device communicates data with a network in order to conduct a key exchange, in accordance with exemplary embodiments;

Figure 1b is a graphical illustration of hardware, firmware, and software components for a server, in accordance with exemplary embodiments;

Figure 1c is an illustration of exemplary network static public keys and server static public keys recorded by a plurality of devices, in accordance with exemplary embodiments;

Figure 2a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device, a server, and a key server, in accordance with exemplary embodiments;

Figure 2b is a simplified message flow diagram illustrating an exemplary messages transmitted from a device to a server, in accordance with exemplary embodiments;

Figure 2c is a flow chart illustrating exemplary steps for conducting a key exchange using PKI keys in order to derive a shared secret key, and for using the derived shared secret key to encrypt and decrypt data, in accordance with exemplary embodiments;

Figure 2d is a flow chart illustrating exemplary steps for conducting an elliptic curve (EC) point addition operation to combine public keys, for conducting a key exchange using PKI keys in order to derive shared secrets, and for conducting a key derivation function using the derived shared secrets, in accordance with exemplary embodiments;

Figure 2e is a flow chart illustrating exemplary steps for using a derived shared secret key to encrypt and decrypt data, in accordance with exemplary embodiments;

Figure 2f is an illustration of an exemplary server database, in accordance with exemplary embodiments;

Figure 3 is an illustration of an exemplary set of cryptographic parameters, in accordance with exemplary embodiments;

Figure 4 is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device, a server, a first key server, and a second key server, in accordance with exemplary embodiments;

Figure 5a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a mobile device, a g node b, and a key server, in accordance with exemplary embodiments;

Petitioner Samsung Ex-1004, 1199

Figure 5b is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a client, a server, and a key server, in accordance with exemplary embodiments;

Figure 5c is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by an initiator, a responder, and a key server, in accordance with exemplary embodiments; and

Figure 5d is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a radio-frequency identification (RFID) tag, a RFID reader, and a key server, in accordance with exemplary embodiments.

Petitioner Samsung Ex-1004, 1200

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS OF THE INVENTION

Figure 1a

Figure 1a is a graphical illustration of an exemplary system, where device communicates data with a network in order to conduct a key exchange, in accordance with exemplary embodiments. The system 100 can include a device 103 and a network 105, where the nodes can communicate data 106 over an Internet Protocol (IP) network 107. Network 105 can comprise a plurality of servers supporting communication such as data 106 with a plurality of devices 103. In exemplary embodiments, network 105 can include a server 101 and a key server 102. The exemplary servers shown for network 105 in system 100 can be either different physical computers such as rack-mounted servers, or different logical or virtual servers or instances operating in a "cloud" configuration. Or, server 101 and key server 102 could represent different logical "server-side" processes within a network 105, including different programs running on a server that listen and communicate using different IP port numbers within one physical server.

In exemplary embodiments, server 101 and key server 102 can operate using the physical electrical components depicted and described for a server 101 in Figure 1b below. Other possibilities exist as well for the physical embodiment of server 101 and key server 102 without departing from the scope of the present disclosure, including exemplary embodiments depicted and described in connection with Figure 5a through Figure 5d below. In exemplary embodiments, server 101 can be described as a "first server" and key server 102 can be described as a "second server". Further, the combination of a first server 101 and a second server 102 can comprise a network 105. The combination of a first server 101 and a second server 102 can also comprise a "set of servers".

Although server 101 and key server 102 are depicted in Figure 1a as belonging to the same network 105, server 101 and key server 102 could be associated with different networks and communicate in a secure manner. Secure sessions between server 101 and key server 102 could be established over IP network 107 using methods including a physical wired connection via a local area network (LAN), transport layer security (TLS), a virtual private network (VPN), and IP Security (IPSEC), a secure shell (SSH) tunnel, and other possibilities exist as well. As depicted in Figure 1a, server 101 and key server 102 could communicate over a private network 107a, where the private network 107a could be established using the methods for a secure session in the previous sentence.

Device 103 can be a computing device for sending and receiving data. Device 103 can take several different embodiments, such as a general purpose personal computer, a mobile phone based

Petitioner Samsung Ex-1004, 1201

on the Android ® from Google ® or the IOS operating system from Apple ®, a tablet, a device with a sensor or actuator for the "Internet of Things", a module for "machine to machine" communications, a device that connects to a wireless or wired Local Area Network (LAN), an initiator according to the Device Provisioning Protocol specification (DPP) from the WiFi alliance, a router, and/or a server, and other possibilities exist as well without departing from the scope of the present disclosure. Exemplary electrical components within a device 103 can be similar to the electrical components for a server 101 depicted and described in Figure 1b below, where device 103 can use electrical components with smaller capacities and lower overall power consumption, compared to the capacity and power consumption for the same electrical components in a server 101.

Device 103 can include a device identity 103i, which could comprise a string or number to uniquely identify device 103 with network 105 and/or server 101 and server 102. Device identity 103i could comprise a medium access control (MAC) address for a physical interface such as Ethernet or WiFi, a Subscription Permanent Identifier (SUPI) with 5G networks, an international mobile subscriber identity (IMSI) or international mobile equipment identity (IMEI) with 2G/3G/4G networks, and other possibilities exist as well without departing from the scope of the present disclosure. In exemplary embodiments, device identity 103i can be written to hardware in device 103 and operate as a unique, long-term identity for device 103.

Device 103 can record at least one elliptic curve cryptography (ECC) static public key for network 105 comprising network static public key Sn 102a. Device 103 can record at least one elliptic curve cryptography (ECC) static public key for server 101 comprising server static public key Ss 101a. Network static public key Sn 102a and server static public key Ss 101a could be recorded in nonvolatile or volatile memory within device 103. For embodiments where keys 102a and/or 101a are recorded in nonvolatile memory, keys 102a and/or 101a could be recorded by a device manufacturer or device distributor. Or, key 102a could be recorded by a device owner before device 103 connects with network 105, and key 101a could be recorded by a device distributor or device manufacturer. Other possibilities exist as well for the source and secure recording of keys 102a and 101a without departing from the scope of the present disclosure.

As contemplated in the present disclosure, the use of a capital letter as the first character for a PKI key can represent a public key, the use of a lower case letter as the first character for a PKI key can represent a private key. The use of a character "S" or "s" for the first character in a key can represent a static key and the use of a character "E" or "e" for the first character can represent an ephemeral key. In some embodiments, an ephemeral key can be referred to as a protocol key or a temporary key. As contemplated in the present disclosure, the second letter for a PKI key can represent the entity the key is associated with or belongs to (e.g. "d" for device 103

Petitioner Samsung Ex-1004, 1202

and "s" for server 101 and "n" for network 102). Thus, using the notation described in the previous three sentences, the corresponding private key for network static public key Sn 102a can comprise network static private key ss 102b. Likewise, a key "es" would represent an ephemeral private key for a server, while key "Es" would represent an ephemeral public key for a server. Similarly, "Ed" would represent an ephemeral public key for a device 103 and "sd" would represent a static private key for a device 103, etc.

For embodiments where public keys Sn 102a and Ss 101a are recorded in volatile memory, device 103 could obtain keys 102a and 101a from a different server than server 101 for network 105 before sending data 106, such as device 103 obtaining keys 102a and/or via a secure session from a different server before sending data 106. A device 103 can record a plurality of different network static public keys Sn 102a and server static public keys Ss 101a in a network public key table 103t. Different keys 102a in a table 103t could be associated with different networks 105 that device 103 communicates with over time. Or (i) a first set of keys 102a and 101a could be used with a first network, and (ii) a second, different set of keys 102a and 101a in a table 103t could be used as a backup or failover second set of keys 102a and 101a for the first network, and (iii) a third set of keys 102a and 101a could be used with a second network. Exemplary data for a network public key table 103t for device 103 is depicted and described in connection with Figure 1c below. The different keys 102a and/or 101a can be associated with network names and/or Uniform Resource Locators (URLs) or domain names, such that device 103 can select the network static public key 102a and/or 101a based on a URL or domain name where device 103 will send data 106.

Network static public key Sn 102a and server static public key Ss 101a can be obtained by device 103 before conducting an elliptic curve Diffie-Hellman (ECDH) key exchange or an ephemeral elliptic curve Diffie-Hellman (ECHDE) key exchange. Network static public key 102a and/or server static public key Ss 101a could be obtained by device 103 in several different ways. Network static public key 102a and/or server static public key 101a could be written into memory by a manufacturer, distributor, or owner of device 103 before device 103 connects with server 101 or a network 107. Keys 102a and 101a could be received by device 103 over an IP network 107 via a secured session, such as a TLS, IPSec, or VPN connection before sending data 106 to server 101. In exemplary embodiments, network static public key 102a and server static public key 101a are recorded in device 103 in a secured and authenticated manner, such that device 103 can trust network static public key 102a.

As one exemplary embodiment, network static public key 102a and/or server static public key 101a could be public keys within different certificates, where the public keys 102a and/or 101a are signed by a certificate authority. Although not depicted in Figure 1a, device 103 could also

Petitioner Samsung Ex-1004, 1203

record a certificate authority root certificate or public key, and device 103 could (a) verify the signature of a certificate authority in a certificate for the public key 102a and/or 101a using (b) the recoded root certificate or public key for the certificate authority (and any intermediary parent certificates). Network static public key 102a and server static public key 101a could be processed or formatted according to a set of cryptographic parameters 104, and network static public key 102a and server static public key 101a could also be compatible with parameters 104.

Although public keys 102a and 101a are described as "static" herein, the key could change over time such as with the expiration of a validity date when recorded in a certificate. Public keys 102a and 101a could remain static over the period of time for device 103 to conduct at least two separate ECDHE key exchanges, where one ECDHE key exchange uses an ephemeral or derived ECC PKI keys for device 103. Public keys 102a and 101a could comprise a long-term public key for use by device 103 or a plurality of devices 103 when communicating with network 107. Although the use of a certificate for public keys 102a and 101a is described in the paragraph above for public keys 102a and 101a, the use of a certificate is not required. In an embodiment depicted in Figure 5c below, (i) public key 102a could comprise a responder bootstrap public key and (ii) device 103 could comprise an initiator according to the DPP standard, which is also depicted and described in connection with Figure 5c below.

Cryptographic parameters 104 can specify values or settings for (i) conducting an ECDH or ECDHE key exchange, (ii) mutually deriving a symmetric ciphering key through a key derivation function and/or a secure hash algorithm, and (iii) using a symmetric ciphering algorithm. As contemplated herein, cryptographic parameters 104 may also be referred to as parameters 104. Each of device 103, server 101, and key server 102 can record at least one compatible subset of parameters within a set of cryptographic parameters 104. Parameters 104 can specify values for an elliptic curve cryptography (ECC) curve name, key length, key formatting (e.g. compressed or uncompressed), encoding rules, etc.

As contemplated herein, the parameters 104 and cryptographic algorithms used with ECC PKI keys and a key exchange in the present disclosure can be compatible and substantially conform with ECC algorithms and keys as specified in (i) the IETF Request for Comments (RFC) 6090 titled "Fundamental Elliptic Curve Cryptography Algorithms", and (ii) IETF RFC 5915 titled "Elliptic Curve Private Key Structure", and also subsequent and related versions of these standards. Other possibilities exist as well for cryptographic parameters 104 without departing from the scope of the present disclosure.

For use of ECC algorithms, parameters 104 can specify elliptic curve names such as, but not limited to NIST P-256, sect283k1, sect283r1, sect409k1, sect409r1, and other possibilities exist as well. Further, elliptic curves that do not depend on curves currently specified by the National

Petitioner Samsung Ex-1004, 1204

Institute of Standards and Technology (NIST) could be utilized as well, such as, but not limited to, Curve22519, curve448, or FourQ. Parameters 104 can specify domain parameters for nodes in system 100 to calculate values or numbers in a compatible manner, such as common base point G for use with ECC PKI key pairs and a defining equation for an elliptic curve. An exemplary set of cryptographic parameters 104 is depicted and described in connection with Figure 3 below.

Device 103 can include an ECC key pair generation algorithm 103x and server 101 can include a compatible ECC key pair generation algorithm 101x. A key pair generation algorithm 103x or 101x can use (i) a random number generator in order to derive the ephemeral PKI private key and (ii) a selected set of cryptographic parameters 104 in order to calculate the ephemeral PKI public key. In exemplary embodiments, a random number for the ephemeral PKI private key multiplies the base point G from the parameters 104 in order to obtain the corresponding ephemeral PKI public key. An ECDH key exchange or key derivation step can conduct an elliptic curve point multiplication with a scalar value (e.g. a private key) and a point. Other possibilities exist as well for the algorithms 103x and 101x to derive an ephemeral ECC PKI key pair without departing from the scope of the present disclosure. A key pair generation algorithm 103x for device 103 can output an ephemeral ECC PKI pair comprising device ephemeral public key Ed 103a and device ephemeral private key ed 103b. A key pair generation algorithm 101x for server 101 can output an ephemeral ECC PKI pair comprising server ephemeral public key Es 101a and server ephemeral private key es 101b.

Device 103 can also record a device static PKI key pair 103p in nonvolatile memory or within a secure processing environment within device 103. The key pair 103p can be either (i) generated by device 103 during device manufacturing or device distribution, or (ii) generated externally from device 103 and written to device 103 in a secure manner during device manufacturing or device distribution. The PKI key pair 103p can comprise a device static private key sd 103d and a device static public key Sd 103c. The keys sd 103d and Sd 103c could be formatted and compatible with a selected subset of cryptographic parameters 104a from set of cryptographic parameters 104. In exemplary embodiments, public key Sd 103c can be recorded in an X.509 certificate from a certificate authority.

In some embodiments, device 103 could renew static key Sd 103c and sd 103d, or possibly record a plurality of the values upon manufacturing or distribution. Device 103 could switch from using a first pair of static device PKI keys to a second pair of static device PKI keys over time. In addition, a first pair of device static PKI keys could be associated with a first set of cryptographic parameters 104a and a second pair of static PKI keys (e.g. key 103c and key 103d) could be associated with a second set of cryptographic parameters 104a. Thus, device 103 could record a

Petitioner Samsung Ex-1004, 1205

plurality of different device static PKI key pairs 103p associated with different parameters 104a, and select the PKI key pair 103p based on the parameters 104a for use with data 106.

As depicted in Figure 1a, server 101 can include a server identity 101i, a key pair generation algorithm 101x, a set of cryptographic parameters 104, a server database 101d, a server certificate 101c, and at least one server static private key ss 101b. Server identity 101i can comprise a name or number to uniquely identify server 101 in network 105 and/or IP network 107. In exemplary embodiments, server identity 101i can comprise a domain name service (DNS) name, which could comprise a string of characters and/or numbers. Server identity 101i could also comprise a realm portion of a network access identifier. Server identity 101i could be associated with an IP address, such that the exemplary data 106 from device 103 could be routed to server 101 via the IP network 107. Server identity 101i could also comprise a MAC address, and a server identity 101i could comprise multiple different values such as all of (i) a MAC address, (ii) a DNS name, (iii) and virtual instance identity if server 101 operates as a virtual server.

In summary, server identity 101i can allow (a) a plurality of different devices 103 to (b) select and route data 106 to server 101 from a potential plurality of different servers and nodes. Other possibilities exist as well for the format, structure, or value for a server identity 101i without departing from the scope of the present disclosure. Server identity 101i could also comprise a server name indication (SNI) value.

The set of cryptographic parameters 104 for server 101 can be equivalent to or a superset of the cryptographic parameters 104 used by device 103. The description above for a set of parameters 104 used by a device 103 is also applicable to a set of parameters 104 used by a server 101. Cryptographic parameters 104 and a subset of cryptographic parameters 104a are depicted and described in connection with Figure 3 below.

Server database 101d for server 101 can comprise a database or memory recording data for server 101 to communicate with both a plurality of devices 103 and also at least one server static private key ss 102b. An exemplary server database 101d is depicted and described in connection with Figure 2f below. Server database 101d can record values for PKI keys, selected parameters 104a for use with PKI keys, derived shared secrets, derived symmetric ciphering keys, random numbers used in secure sessions, and related values in order to support the communications with both device 103 and server 102, as well as a plurality of different devices 103. Server certificate 101c can comprise a certificate formatted according to the X.509 family of standards and include a static server 101 public key Ss 101a. Server certificate 101c can include a signature from a certificate authority for server public key Ss 101a.

As depicted in Figure 1a, server 101 can also record and operate with a private key corresponding to public key Ss 101a, where the server static private key ss 101b can is depicted in

Petitioner Samsung Ex-1004, 1206

Figure 1a. Note that a server 101 could operate with a plurality of different pairs of PKI keys comprising public key Ss 101a and private key ss 101b in order to support communications with a plurality of different devices 103. The use of different values for keys Ss 101a and ss 101b with different devices 103 is depicted and described in connection with Figure 2f below, such as a first pair 101a-1, 101b-1 with a first device identity 103i-1 and a second pair 101a-3, 101b-3 with a second device identity 103i-3.

As depicted in Figure 1a, key server 102 can include a key server identity 102i, a set of cryptographic parameters 104, a network static private key sn 102b, and a key server database 102d. Key Server identity 102i can comprise a name or number to uniquely identify key server 102 in network 105 and/or IP network 107. Key Server identity 102i can be similar to server identity 101i, except using a different value, name, or number in order to uniquely identify key server 102 within network 105. The set of cryptographic parameters 104 for server 102 can be equivalent to or a superset of the cryptographic parameters 104 used by device 103 and parameters 104 was also described above for device 103.

In exemplary embodiments, the parameters 104 used by both key server 102 and server 101 can be fully compatible, such as using the same ECC named curve, key lengths, encoding rules, etc. Server database 102d for key server 102 can comprise a database or memory recording data for key server 102 to (i) communicate with a plurality of servers 101 and (ii) support server 101 communicating with a plurality of devices 103. Key server database 102d can be similar to server database 101d depicted in Figure 2f, except that key server database 102d can record values and data calculated by key server 102. Key server database 102d can record values for PKI keys, derived shared secrets, and related values in order to support the communications between (i) network 105 and/or server 101 and (ii) device 103.

As depicted in Figure 1a, key server database 102d can record sets of data for different devices 103, where each set can comprise a row in a table with a device identity 103i, the network static public key value Sn 102a, and the network static private key sn 102b. In exemplary embodiments, a key server database 102d can also include secure hash values for the network static public key Sn 102a, and key server 102 could select the network static private key sn 102b using the secure hash value for the network static public key Sn 102a. Or, key server 102 could select the network static private key sn 102b using the network static public key Sn 102a.

As depicted for a key server database 102d in Figure 1a, some devices 103 could share the same keys 102a and 102b, which could comprise shared keys 102z for the devices 103 as depicted and described in connection with Figure 1c below. Other devices 103 could record unique keys 102v, where devices 103 record a value for the network static public key Sn 102a that is uniquely recorded in each device. A key server database 102d could record and track the associated network

Petitioner Samsung Ex-1004, 1207

private and public keys for each device. In other exemplary embodiments, a key server 102 could omit recording device identities 103i in a database 102d, and key server 102 could associate and use a network static private key sn 102b with a particular server 101 (e.g. all data from a server 101 could use or be associated with the private key sn 102b).

Other possibilities exist as well for the mapping of network static private keys to either servers 101 or devices 103 without departing from the scope of the present disclosure. Also, although a single value for sn 102b is depicted as associated with a device 103 using a device identity 103i in a key server database 102d, a key server 102 could also record and use multiple different values of network static private key sn 102b, such as (i) different values for sn 102b for different parameters 104 (e.g. different named curves), or (ii) separate values for sn 102b for digital signatures and ECDH key exchanges. In other words, a device 103 could also record the corresponding different multiple values for Sn 102a, and select and use the public keys depending on requirements such as parameters 104 used or if the network public key will be used for verifying digital signatures or conducting ECDH key exchanges. The use of multiple different values for Sn 102a by device 103 for a network 105 is also depicted and described in connection with Figure 1c below.

Key server 102 can record at least one network static private key sn 102b, which can be the private key corresponding to the network static public key Sn 102a recorded by a set of devices 103 and described above for device 103. In exemplary embodiments and as depicted in Figure 1a and also Figure 2a below, key server 102 may not communicate with device 103 directly, but rather communicates with server 101 through a private network 107a. Although not depicted in Figure 1a, a network 105 could operate a firewall in order to prevent packets or data from the public Internet (other than server 101) from reaching key server 102. In this manner by isolating key server 102 from IP network 107, security for the key server 102 and the network static private key sn 102b can be enhanced, since only authenticated and authorized nodes within network 105 and connected to private network 107a could communicate with key server 102.

IP network 107 could be either a Local Area Network (LAN) or a Wide Area Network (WAN), or potentially a combination of both. IP network 107 could include data links supporting either IEEE 802.11 (WiFi) standards. Device 103 also utilize a variety of WAN wireless technologies to communicate data 106 with server 101, including Low Power Wide Area (LPWA) technology, 3rd Generation Partnership Project (3GPP) technology such as, but not limited to, 3G, 4G Long-Term Evolution (LTE), or 4G LTE Advanced, NarrowBand – Internet of Things (NB-IoT), LTE Cat M, proposed 5G networks, and other examples exist as well. Server 101 can connect to the IP network 107 via a wired connection such as, but not limited to, an Ethernet, a fiber optic, or a Universal Serial Bus (USB) connection (not shown).

Petitioner Samsung Ex-1004, 1208

IP network 107 could also be a public or private network supporting Internet Engineering Task Force (IETF) standards such as, but not limited to, such as, RFC 786 (User Datagram Protocol), RFC 793 (Transmission Control Protocol), and related protocols including IPv6 or IPv4. A public IP network 107 could utilize globally routable IP addresses. Private IP network 107a could utilize private IP addresses which could also be referred to as an Intranet. Other possibilities for IP Network 107 and Private Network 107a exist as well without departing from the scope of the disclosure.

Figure 1b

Figure 1b is a graphical illustration of hardware, firmware, and software components for a server, in accordance with exemplary embodiments. Figure 1b is illustrated to include several components that can be common within a server 101. Server 101 may consist of multiple electrical components in order to communicate with a plurality of devices 101 and at least one key server 102. In exemplary embodiments and as depicted in Figure 1b, server 101 can include a server identity 101i, a processor 101e (depicted as "CPU 101e"), random access memory (RAM) 101f, an operating system (OS) 101g, storage memory 101h (depicted as "nonvolatile memory 101h"), a Wide Area Network (WAN) interface 101j, a LAN interface 101k, a system bus 101n, and a user interface (UI) 101m.

Server identity 101i could comprise a preferably unique alpha-numeric or hexadecimal identifier for server 101, such as an Ethernet MAC address, a domain name service (DNS) name, a Uniform Resource Locator (URL), an owner interface identifier in an IPv6 network, a serial number, an IP address, or other sequence of digits to uniquely identify each of the many different possible nodes for a server 101 connected to an IP network 105. Server identity 101i can preferably be recorded in a non-volatile memory and recorded by a network 105 upon configuration of a server 101. Server identity 101i may also be a number or string to identify an instance of server 101 running in a cloud or virtual networking environment.

In exemplary embodiments, server 101 can operate with multiple different server identities 101i, such as a first server identity 101i comprising a DNS name and a second server identity 101i comprising an IP address and a port number. A different server 101 could be associated with a different IP address and port number for a network 105. In exemplary embodiments, a first server 101 with a first server identity 101i can operates with a first Internet Protocol address and port (IP:port) number with a first server static private key ss 101b, and a second server 101 with a second server identity 101i can operate with a second IP:port number and a second server static private key ss 101b.

Petitioner Samsung Ex-1004, 1209

The CPU 101e can comprise a general purpose processor appropriate for higher processing power requirements for a server 101, and may operate with multiple different processor cores. CPU 101e can comprise a processor for server 101 such as an ARM® based process or an Intel® based processor such as belonging to the XEON ® family of processors, and other possibilities exist as well. CPU 101e can utilize bus 101n to fetch instructions from RAM 101f and operate on the instruction. CPU 101e can include components such as registers, accumulators, and logic elements to add, subtract, multiply, and divide numerical values and record the results in RAM 101f or storage memory 101h, and also write the values to an external interface such as WAN interface 101j and/or LAN interface 101k. In exemplary embodiments, CPU 101e can perform the mathematical calculations for a key pair generation step 101x and also an ECDH key exchange algorithm 220 depicted in Figure 2a, Figure 2d, etc., below.

CPU 101e can also contain a secure processing environment (SPE) 101u in order to conduct elliptic curve cryptography (ECC) operations and algorithms, such as EC point addition steps 213 and 220 as depicted in Figure 2d below, as well as deriving ephemeral ECC PKI keys such as with key generation step 101x depicted and described in connection with Figure 1a above. SPE 101u can comprise a dedicated area of silicon or transistors within CPU 101e in order to isolate the ECC operations from other programs or software operated by CPU 101e, including many processes or programs running operating system 101g. SPE 101u could contain RAM memory equivalent to RAM 101f and nonvolatile memory equivalent to storage memory 101h, as well as a separately functioning processor on a smaller scale than CPU 101e, such as possibly a dedicated processor core within CPU 101e. SPE 101u can comprise a "secure enclave" or a "secure environment", based on the manufacturer of CPU 101e. In some exemplary embodiments, an SPE 101u can be omitted and the CPU 101e can conduct ECC operations or calculations without an SPE 101u.

RAM 101f may comprise a random access memory for server 101. RAM 101f can be a volatile memory providing rapid read/write memory access to CPU 101e. RAM 101f could be located on a separate integrated circuit in server 101 or located within CPU 101e. The RAM 101f can include data recorded in server 101 for the operation when communicating with a plurality of devices 103 or a key server 102. The system bus 101n may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures including a data bus. System bus 101n connects components within server 101 as illustrated in Figure 1b, such as transferring electrical signals between the components illustrated. Server 101 can include multiple different versions of bus 101n to connect different components, including a first system bus 101n between CPU 101e and RAM 101f (which could be a memory bus), and a second system bus 101n between CPU 101e and WAN interface 101j or LAN interface 101k, which could be an I2C bus, an SPI bus, a PCI bus, or similar data busses.

Petitioner Samsung Ex-1004, 1210

In exemplary embodiments, RAM 101f operating with server 101 can record values and algorithmic steps or computer instructions for conducting an ECDH key exchange, including a key pair generation step 101x. Additional shared secret values or points on an elliptic curve can be recorded in RAM 101f as well, such as the values or points depicted and described in connection with Figure 2d below, which could comprise a point X0 213a, a secret X1 216a, secret X2 218a, and a secret X3 207a. The depicted values and algorithms can be recorded in RAM 101f so that CPU 101e can conduct ECC operations and calculations quickly using the values. The depicted values could also be recorded in other locations for longer-term or nonvolatile storage, such as within a server database 101d. Additional or other values besides the ones depicted in Figure 1b can also be recorded in RAM 101f in order to support server 101 conducting the communications, steps, and message flows depicted in Figure 2a and Figure 4 below, and also other Figures herein.

The operating system (OS) 101g can include Internet protocol stacks such as a User Datagram Protocol (UDP) stack, Transmission Control Protocol (TCP) stack, a domain name system (DNS) stack, a TLS stack, a DPP stack, etc. The operating system 101g may include timers and schedulers for managing the access of software to hardware resources within server 101, where the hardware resources managed by OS 101g can include CPU 101e, RAM 101f, nonvolatile memory 101h, and system bus 101n, and well as connections to the IP network 107 via a WAN interface 101j. The operating system shown of 101g can be appropriate for a higher power computing device with more memory and CPU resources (compared to a device 103). Example operating systems 101g for a server 101 includes Linux or Windows® Server, and other possibilities exist as well. Although depicted as a separate element within server 101 in Figure 1b, OS 101g may reside in RAM 101f and/or nonvolatile memory 101h during operation of server 101.

As depicted in Figure 1b, OS 101g in Figure 1b can contain algorithms, programs, or computer executable instructions (by processor 101e or SPE 101u) for an ECDH key exchange algorithm 207 (depicted and described in Figure 2d below), a key derivation function (KDF) 208 (depicted and described in Figure 2d below), and also EC point addition operations 213 and 220. ECC point addition operation 213 and 220 are depicted and described in connection with Figure 2d below. OS 101g could also record and operate a decryption step 221 and an encryption step 223, as depicted and described below in Figure 2e. The steps or algorithms could be included either (i) within the kernel of OS 101g, or (ii) as a separate program or process loaded by OS 101g and operated by OS 101g. OS 101g can also read and write data to a secure processing environment SPE 101u, if CPU 101e contains SPE 101u.

Nonvolatile memory 101h or "storage memory" 101h (which can also be referred to herein as "memory 101h") within server 101 can comprise a non-volatile memory for long-term storage of data, including times when server 101 may be powered off. Memory 101h may be a NAND

Petitioner Samsung Ex-1004, 1211

flash memory or a NOR flash memory and record firmware for server 101, such as a bootloader program and OS 101g. Memory 101h can record long-term and non-volatile storage of data or files for server 101. In an exemplary embodiment, OS 101g is recorded in memory 101h when server 101 is powered off, and portions of memory 101h are moved by CPU 101e into RAM 101f when server 101 powers on.

Memory 101h (i) can be integrated with CPU 101e into a single integrated circuit (potentially as a "system on a chip"), or (ii) operate as a separate integrated circuit or a removable card or "disk", such as a solid state drive (SSD). Storage memory 101h can also comprise a plurality of spinning hard disk drives in a redundant array of independent disks (RAID) configuration. Memory 101h may also be referred to as "server storage" and can include exemplary file systems of FAT16, FAT 32, NTFS, ext3, ext4, UDF, or similar file systems. As contemplated herein, the terms "memory 101h", "storage memory 101h", and "nonvolatile memory 101h" can be considered equivalent.

As depicted in Figure 1b, non-volatile memory 101h can record a server database 101d, at least one device static public key Sd 103c, and cryptographic parameters 104. Exemplary data within a server database 101d is depicted and described in connection with Figure 2f below. Although depicted in Figure 1b as recorded within memory 101h, a server database 101d could also operate as a separate server than server 101 in a network 105, and server 101 could query the server database 101d using a private network 107a. The device static public key Sd 101c could be received by server 101 from a device manufacturer or a device owner, or directly from device 103 through IP network 107 (preferably in a secured manner). In addition, as depicted in Figure 1b, memory 101h can record the parameters 104 which were depicted and described in connection with Figure 1a above and also Figure 3 below.

Server 101 can include a WAN interface 101j to communicate with IP network 107 and a plurality of devices 103, as depicted in Figure 1a above (where Figure 1a depicts a single device 103). WAN interface 101j can comprise either a wired connection such as Ethernet or a wireless connection. For wireless configurations of server 101, then WAN interface 101j can comprise a radio, which could connect with an antenna in order to transmit and receive radio frequency signals. For a wireless configuration of server 101, WAN interface 101j within server 101 can provide connectivity to an IP network 107 through 3GPP standards such as 3G, 4G, 4G LTE, and 5G networks, or subsequent and similar standards.

In some exemplary embodiments, server 101 can comprise a "g node b" or gNb in a 5G network (or equivalent functionality in 6G or subsequent networks), and WAN interface 101j can comprise a 5G radio access network (RAN) interface. WAN interface 101j can also comprise a wired connection such as digital subscriber line (DSL), coaxial cable connection, or fiber optic

Petitioner Samsung Ex-1004, 1212

connection, and other possibilities exist as well without departing from the scope of the present disclosure.

Server 101 may also operate a LAN interface 101k, where LAN interface 101k can be used to connect and communicate with other servers in a network 105, such as key server 102 through private network 107a. LAN interface 101k can comprise a physical interface connected to system bus 101n for server 101. In exemplary embodiments, LAN interface 101k can comprise an Ethernet or fiber optic wired connection. In other words, (i) LAN interface 101k can connect server 101 to private network 107a (which could comprise an IP network with private IP addresses that are not globally routable), and (ii) WAN interface 101j can comprise an interface for communicating with a plurality of devices 103 through insecure networks such as the globally routable public Internet. The use of a separate WAN interface 101j and LAN interface 101k can increase the security of operation for server 101. However, the use of separate physical interfaces for LAN interface 101k and WAN interface 101j can be omitted, and a single physical interface such as Ethernet or fiber-optic could be used by server 101 to communicate with both devices 103 and key server 102.

Server 101 may also optionally include user interface 101m which may include one or more sub-servers for receiving inputs and/or one or more sub-servers for conveying outputs. User interfaces are known in the art and may be simple for many servers 101 such as a few LED lights or and LCD display, and thus user interfaces are not described in detail here. User interface 101m could comprise a touch screen or screen display with keyboard and mouse, if server 101 has sophisticated interaction with a user, such as a network administrator. Server 101 can optionally omit a user interface 101m, if no user input or display is required for establishing communications within a network 105 and/or IP network 107.

Although not depicted in Figure 1b, server 101 can include other components to support operation, such as a clock, power source or connection, antennas, etc. Other possibilities exist as well for hardware and electrical components operating in a server 101 without departing from the scope of the present disclosure. Using the electrical components depicted in Figure 1b, a server 101 could send and receive the data 106 in Figure 1a in an encrypted and secure manner after conducting the authenticated ECDHE key exchange as contemplated herein, in order to derive a symmetric ciphering key to encrypt and decrypt messages within data 106 with a plurality of devices 103.

Although not depicted in Figure 1b, devices 103 such as the device 103 depicted in Figure 1a above can include (a) equivalent internal electrical components depicted for a server 101 in order to (b) operate as devices 103. A device 103 in Figure 1a could include a processor similar to CPU 101e, with primary differences for the processor in a device being reduced speed, a smaller memory cache, a smaller number and size of registers, with an exemplary use of 32 or 16 bits for datapath

Petitioner Samsung Ex-1004, 1213

widths, integer sizes, and memory address widths, etc., for a device 103. In contrast, an exemplary 64 bit datapaths could be used for CPU 101e in server 101 (although device 103 could also use 64 bit wide datapath widths if device 103 comprises a mobile phone such as a smart phone). For embodiments where device 103 comprises a transducer device for sending and receiving transducer data with a network 105, then a CPU in device 103 could comprise an exemplary 32 bit processor, although other possibilities exist as well.

Similarly, RAM in a device 103 could be a RAM similar to RAM 101f in server 101, except the RAM in a device 103 could have fewer memory cells such as supporting exemplary values less than or equal to an exemplary 4 gigabytes, while RAM 103f in server 101 could support more memory cells such as greater than or equal to an exemplary 8 gigabytes. In exemplary embodiments, the electrical and physical components of a key server 102 can be equivalent to the electrical components for a server 101 in Figure 1b, with different data recorded in RAM 101f for a key server 102, as well as different data recorded in memory 101h for a key server 102. For example, a key server 102 could record the network static private key sn 102b in memory 101h, which could comprise secure disk storage using disk or file encryption.

Figure 1c

Figure 1c is an illustration of exemplary network static public keys and server static public keys recorded by a plurality of devices, in accordance with exemplary embodiments. Figure 1c depicts PKI keys recorded for an exemplary three different devices 103, although a system 100 and other systems herein could operate with potentially millions or more devices 103. The data depicted for each device in Figure 1c can comprise exemplary data for a network public key table 103t for a device 103, which is also depicted and described in connection with Figure 1a above.

The exemplary values recorded for network static public keys depicts different embodiments where both (i) a device 103 can record a network static public key Sn 102a and/or a server static public key Ss 101a that is shared with other devices 103, and (ii) the network static public key Sn 102a and/or a server static public key Ss 101a recorded by device 103 could be unique for device 103 (e.g. not shared with other devices 103 in a system 100 above or a system 200 below, as well as other systems herein). Although not depicted for a network public key table 103t, the network public key table 103t could also store or record (i) secure hash values for each public key stored, and (ii) the parameters 104a associated with the public keys.

A network public key table 103t for device 103 can record values of a key identity, a network name for network 105, an identity for server 101 comprising ID.server 101i, and also a value for the network static public key Sn 102a and also server static public key Ss 101a. As

Petitioner Samsung Ex-1004, 1214

depicted in Figure 1c, a device 103 can record multiple different values for use with multiple different networks 105 and/or servers 101. As depicted in Figure 1c, a device 103 can also record multiple different values for use with a single network 105, such as a primary set of PKI keys (e.g. row 1), and a backup set of PKI keys (e.g. row 2).

The first two entries for network static public keys Sn 102a for a first device 103 (1) and a second device 103 (2) in Figure 1c (e.g. row 1 for each device 103) depicts the same alphanumeric values for basE91 binary to text encoding for an exemplary network static public keys Sn 102a-1 in a first device 103 (1) and Sn 102a-2 for a second device 103 (2), where the key value is depicted for a network 105 of "Network A". In other words, a value Sn 102a-1 and Sn 102a-2 can be equal. Likewise, the second two entries for network static public keys Sn 102a for a first device 103 (1) and a second device 103 (2) in Figure 1c (e.g. row 2 for each device 103) depicts the same alphanumeric values for basE91 binary to text encoding for an exemplary network static public key Sn 102a in a first device 103 (1) and a second device 103 (2).

Note that although a single value is depicted for PKI keys in a network public key table 103t, the values or numbers for keys recorded could comprise a point on an ECC curve with both an X coordinate and a Y coordinate. For illustration purposes in Figure 1c, only the X coordinate are displayed and the Y coordinate could be calculated from the X coordinate using the equation for an ECC curve in a set of cryptographic parameters 104a for the PKI keys.

The depiction of these exemplary keys Sn 102a illustrates the use of shared network public keys 102z for a plurality of different devices 103. Although only three devices are depicted with shared network public keys 102z, many more devices could also record the same shared network public keys for Sn 102a. Each of the shared network public keys 102z is associated with the same network 105, identified with an exemplary same network name "A", although row 1 and row 2 for device 103-1 and device 103-2 could be for different networks as well (e.g. "network A" for row 1 and a "network X" for row 2). In this manner, a plurality of different devices 103 can record and use the same value for a network static public key Sn 102a. As described above, the value in a table 103t including network static public key Sn 102a could be written in device before the device sends the first message 299 in Figure 2a below. The data could be recorded by a device manufacturer, a device distributor, or a device owner, and other possibilities exist as well for the source of Sn 102a without departing from the scope of the present disclosure.

The same values for shared network public keys 102z across different devices 103 could be recorded in device 103 during manufacturing or before distribution to end users of device 103. In this manner, devices 103 could be received by end users in a "partially configured" yet secure state, such that a device 103 could use the recorded keys Sn 102a with a server 101 and/or network

Petitioner Samsung Ex-1004, 1215

105, where a server 101 does not operate or record the corresponding network static private key sn 102b.

As depicted and described in connection with Figures 2a, 4, etc. below, a key server 102 could record and operate with the corresponding network static private key sn 102b and thus the key sn 102b can remain secured and not distributed out or sent to a server 101. In this manner, encrypted communications for data 106 in Figure 1a can be transferred between device 103 and server 101 without server 101 recording the key sn 102b. This increases the security of a system 100 and other systems herein, because server 101 may be exposed to an IP network 107 while key server 102 recording the sn 102b can be connected to a private network 107a. Recording key sn 102b in a separate key server 102 also increases the operational and commercial flexibility for running a network 105, such that server 101 could be controlled and operated by a first entity and key server 102 could be controlled or operated by a second entity, and other potential benefits exist as well.

The first two entries for server static public keys Ss 101a for a first device 103 (1) and a second device 103 (2) in Figure 1c (e.g. row 1 for each device 103) depicts the same alphanumeric values for basE91 binary to text encoding for an exemplary server static public keys Ss 101a-1 in a first device 103 (1) and Ss 101a-2 for a second device 103 (2), where the key value is depicted for a network 105 of "Network A". In other words, a value Ss 101a-1 and Ss 101a-2 can be equal. However, the third two entries for server static public keys Ss 101a-1 for a first device 103 (1) and a second device 103 (2) in Figure 1c (e.g. row 3 for each device 103) depicts different alphanumeric values for basE91 binary to text encoding for an exemplary server static public key Ss 101a in a first device 103 (1) and a second device 103 (2). Thus, a first server static public key Ss 101a for device 103-1 and device 103-2 can comprise the same number or key (e.g. "shared server public keys" 101z) and a different set of server static public keys Ss 101a for device 103-1 and device 103-2 can comprise different values (e.g. unique server public keys 101v).

The depiction of the first row for Ss 101a for device 103-1 and device 103-2 illustrates the use of shared server public keys 101z for a plurality of different devices 103. Although only two devices are depicted with shared server public keys 101z, many more devices could also record the same shared keys for Ss 101a. Each of the shared keys 101z is associated with the same server 101 (or group of servers sharing key ss 101b) within a network 105, identified with an exemplary same network name "A". In this manner, a plurality of different devices 103 can record and use the same value for a server static public key Ss 101a. As described above, the value in a table 103t for server static public key Ss 101a could be written in device before the device sends the first message 299 in Figure 2a below. The data could be recorded by a device manufacturer, a device distributor, or

Petitioner Samsung Ex-1004, 1216

a device owner, and other possibilities exist as well for the source of Ss 101a without departing from the scope of the present disclosure.

The same values for shared server public keys 101z across different devices 103 could be recorded in device 103 during manufacturing or before distribution to end users of device 103. In this manner, devices 103 could be received by end users in a "partially configured" yet secure state, such that a device 103 could use the recorded keys Ss 101a with a server 101 and/or network 105, where a at least one server 101 operates or records the corresponding server static private key ss 101b. Note that in some exemplary embodiments, such as those depicted and described in connection with Figure 4 below, then server static private key ss 101b can also be recorded by a second key server 102, such that server 101 does not operate or record server static private key ss 101b which corresponds to a server static public key Ss 101a recorded by at least one device 103.

By using a set of shared keys 102z or shared keys 101z or across a plurality of devices 103, a key server 102, server 101, or a network 105 can control access of the devices 103 as a group. For example, a network 105 could deny transactions requiring access to the private key corresponding to the public key for the first depicted value of Sn 102a in a first device 103 (1). That action by network 105 would also deny a second device 103 (2) transactions requiring access to the private key corresponding to the public key for the first depicted value of Sn 102a in the second device 103 (2). In this manner, network 105 could control access to a plurality of different devices 103 by controlling access to a single value of sn 102b, where (i) the plurality of different devices 103 record the corresponding Sn 102a as shared keys 102z. Or, key server 102 could deny access to transactions requiring network static private key sn 102b to server 101 or a collection of servers 101 (where Ss 101a is associated with Sn 102a in a set of devices 103).

Other benefits for using shared keys 102z or shared keys 101z can be available as well, such as simplifying manufacturing or distribution, since the same key value for Sn 102a or Ss 101a could be recorded with multiple different devices 103. In other words, a device manufacturer or device distributor would not need to keep track of which values for Sn 102a or Ss 101a belongs with which device 103 for embodiments where shared keys 102z or 101z are utilized. However, the use of shared keys 102z or 101z for multiple different devices 103 is not required for some exemplary embodiments.

In exemplary embodiments, network static public keys Sn 102a can also comprise a unique key for each device 103 in a system 100 and other systems herein. Thus, some exemplary embodiments also support the use of a network static public key Sn 102a that is not shared across multiple different devices 103. For these exemplary embodiments, and as depicted in Figure 1c, a device 103 can record a unique key 102v (depicted as "Unique Network Static Public Key" 102v in Figure 1c). For example, the depicted value for the third key Sn 102a for device 103 (1), (2),

Petitioner Samsung Ex-1004, 1217

and (3) in Figure 1c is shown as unique for each device. A key server 102 could also record the corresponding network static private key sn 102b that is unique for each device in a key server database 102d as depicted for unique keys 102v in Figure 1a. In this manner, a network 105 can control access to server 101 and/or network 105 on a per-device basis using the unique key 102v.

In exemplary embodiments where devices 103 record unique keys 102v for network static public keys Sn 102a, key server 102 could deny access to device 103 (3) (while continuing to allow service for device 103 (1) and 103 (2)), by denying access or cryptographic operations with the secret key sn 102b in a key server 102 corresponding to the public key Sn 102a recorded by device 103 (3). Other benefits for recording network static public keys Sn 102a as unique keys 102v for devices 103 exist as well without departing from the scope of the present disclosure, such as increased security.

If (a) a single network static secret key sn 102b is compromised, then (b) communication with the device recording the corresponding public key Sn 102a could be compromised, while (c) devices recording other, different values for network static public key Sn 102a would not be compromised. Cryptographic analysis or leakage of information pertaining to network static secret keys sn 102b, such as via side channel attacks, timing attacks, malformed ephemeral public keys, etc., risk the compromise of a single key for a single device (when using unique keys 102v), which has less impact than compromise of a network static secret key sn 102b supporting shared keys 102z across a plurality of devices 103.

In exemplary embodiments, server static public keys Ss 101a can also comprise a unique key for each device 103 in a system 100 and other systems herein. Thus, some exemplary embodiments also support the use of a server static public key Ss 101a that is not shared across multiple different devices 103. For these exemplary embodiments, and as depicted in Figure 1c, a device 103 can record a unique key 101v (depicted as "Unique Server Static Public Key" 101v in Figure 1c). For example, the depicted value for the third key Ss 101a for device 103 (1), (2), and (3) in Figure 1c is shown as unique for each device. A server 101 or key server 102x (e.g. Figure 4) could also record the corresponding server static private key ss 101b that is unique for each device in a server database 101d or key server database 102d'. In this manner, a network 105 can control access to server 101 and/or network 105 on a per-device basis using the unique key 101v. For example, server 101 could deny access to device 103 (3) (while continuing to allow service for device 103 (1) and 103 (2)), by denying access or cryptographic operations requiring the secret key ss 101b in a server 101 corresponding to the public key Ss 101a recorded by device 103 (3).

Other benefits for recording server static public keys Ss 101a as unique keys 101v for devices 103 exist as well without departing from the scope of the present disclosure, such as increased security. If (a) a single server static secret key ss 101b is compromised, then (b)

Petitioner Samsung Ex-1004, 1218

communication with the device recording the corresponding public key Ss 101a could be compromised, while (c) devices recording other, different values for server static public key Ss 101a would not be compromised. Cryptographic analysis or leakage of information pertaining to server static secret keys ss 101b, such as via side channel attacks, timing attacks, malformed ephemeral public keys, etc., risk the compromise of a single key for a single device (when using unique keys 101v), which has less impact than compromise of a server static secret key ss 101b supporting share keys 101z across a plurality of devices 103. The use of unique keys 101v and/or unique keys 102v supports the revocation or deprecation of a server PKI key pair (101a/101b) and/or network PKI pair (102a/102b), respectively, and thus can increase the security and flexibility of a system 100, system 200, and other systems herein.

Figure 2a

Figure 2a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device, a server, and a key server, in accordance with exemplary embodiments. System 200 can include a device 103, server 101, and a key server 102. Device 103 was depicted and described in connection with Figure 1a and Figure 1c above. Server 101 and key server 102 were depicted and described in connection with Figure 1a above, and server 101 was depicted and described in connection with Figure 1b above. Server 101 can record and operate a server database 101d, and key server 102 can record and operate a database 102d.

Individual steps and components used in system 200 in Figure 2a are also additionally depicted and described in subsequent Figures 2b, 2c, and 2d, etc. Before starting the steps and message flows depicted in Figure 2a, device 103 can securely receive and record a network public key Sn 102a and a server static public key Ss 101a, which was also depicted and described in connection with Figure 1a and Figure 1c. The corresponding private key for Sn 102a can be securely recorded in key server 102 within network 105 as sn 102b. The corresponding private key for Ss 101a can be securely recorded in (i) server 101 within network 105 as ss 101b for the embodiment depicted in Figure 2a, or (ii) a second key server 102x for the embodiment as depicted in Figure 4 below.

For system 200, server 101 and key server 102 may establish a secure session 201a, which could comprise establishing a secure communications link between the two servers using protocols such as TLS, IPSec, a virtual private network (VPN), a secure shell (SSH), or similar networking, transport, or application layer technologies in order to establish secure communications between key server 102 and server 101. Secure session 201a can utilize certificates for the two servers in order to provide mutual authentication and mutual key derivation for a symmetric encryption key

Petitioner Samsung Ex-1004, 1219

in secure session 201a. Secure session 201a can also be conducted over private network 107a, although the secure session 201a could also be established or conducted through an IP network 107 such as the globally routable Public Internet. Other possibilities exist as well for establishing a secure session 201a between server 101 and key server 102 without departing from the scope of the present disclosure.

Although not depicted in Figure 2a, firewalls between server 101 and key server 102 could also be utilized in order to establish or conduct secure session 201a. At step 201b, server 101 can begin listening for incoming messages from a device 103 using a physical network interface that provides connectivity to the IP network 107 and server 101 can use a specific port number such as TCP port 443 to listen for incoming data 106 from a device 103.

At step 201c, device 103 can be "pre-configured" with operating data such as an operating system similar to OS 101g in Figure 1b (but for device 103), a set of cryptographic algorithms 104 or a subset of cryptographic algorithms 104a, a DNS name for network 105 or server 101, and at least one for each of a device static private key sd 103d, a network static public key Sn 102a, and a server static public key Ss 101a. At a step 201c, a plurality of different keys for at least one network 105 could be recorded in a network public key table 103t as depicted in Figure 1b. Other configuration data for device 103 could be recorded in a step 201c as well, such as timers, drivers for transducers, user information, etc. A step 201c could be performed by a device manufacturer, a device distributor, or a device owner, and other possibilities exist as well without departing from the scope of the present disclosure. A step 201c could be performed in parts, such that a first part is conducted by a device manufacturer, a second part conducted by a device owner, etc.

At step 202, device 103 can be powered on and begin operating, in order to establish connectivity with an IP network 107. At step 202, device 103 can read an address for server 101 from memory or a network public key table 103t, and the address can comprise a DNS name or an IP address for server 101, and can also comprise a server identity 101i. The DNS name or IP address for server 101 could be recorded or received along with the key Ss 102a, or device 103 could conduct a DNS query to obtain the address. As step 202, device 103 can also collect data from transducers to send to a network 105, and the data from transducers could comprise data 202b. Other data besides transducer data could be included in data 202b, such as a device identity ID.device 103i, registration information for device 103 with network 105, and other data as well. Device 103 can also read a device static secret key sd 103d recorded in nonvolatile memory in a step 202.

At step 202, device 103 can also read the set of cryptographic parameters 104 and select a subset of the cryptographic parameters 104a in order to establish communications with server 101. An exemplary subset of cryptographic parameters 104a in a step 202 can comprise a member of

Petitioner Samsung Ex-1004, 1220

the set the cryptographic parameters 104 depicted and described in connection with Figure 3 below (e.g. one line of values in cryptographic parameters 104 in Figure 2d below). In step 202, device 103 can select a subset of cryptographic parameters 104a that is compatible with all of device static secret key sd 103d and network static public key 102a and server static public key 101a. The subset of cryptographic parameters 104a that are compatible with network public keys could also be recorded in nonvolatile memory in device 103 along with network public keys 102a and 101a at the time the network public keys were recorded or received by device 103, which could include during a step 201c above.

A step 203 can also comprise device 103 also using a random number generator in order to output a random number 203a for use in subsequent communications with server 101. Although the term "random number" is described herein, a random number could comprise a pseudo random number processed by device 103 using information entropy available to device 103. In exemplary embodiments, a device 103 can use data from transducers in order to obtain information entropy and generate random numbers in a step 203. The random number 203a processed in a step 203 could contain the number of bits specified by a selected subset of cryptographic parameters 104 from a step 202. Random number 203a generated or derived by a device 103 in a step 203 could also comprise a "number used once" (nonce).

Device 103 can then conduct a key pair generation step 103x as depicted and described in connection with Figure 1a above using the selected subset of cryptographic parameters 104a. The parameters 104 could specify a named curve and parameters to derive a device ephemeral private key ed 103b and a device ephemeral public key Ed 103a. The device ephemeral private key ed 103b can comprise a random number generated using a random number generator, including a second random number generated in a step 203 above. The device ephemeral public key Ed 103a could be derived using (i) ECC point multiplication from a base point G for a named curve within cryptographic parameters 104a and (ii) the device ephemeral private key ed 103b. Other possibilities exist as well for the steps a device 103 can use in a key pair generation step 103x without departing from the scope of the present disclosure.

Device 103 can then conduct a step 204 to calculate a value for use with an ECDH key exchange algorithm 207, where the calculation of the value for a step 204 is depicted and described in connection with a step 204 in Figure 2c below. In summary, device 103 can calculate the sum of device static private key sd 103d and the derived device ephemeral private key ed 103b, and then calculate the modulus of the sum using the value n, where the value n can be defined or determined based upon the selected set of cryptographic parameters 104a. As contemplated herein a "secret key" for use with a PKI key pair can also be referred to as a "private key". As one

Petitioner Samsung Ex-1004, 1221

exemplary embodiment, when using the named elliptic curve secp128r1, the value of n can comprise the decimal number:

340282366762482138443322565580356624661

Device 103 can then conduct a step 205 to perform an ECC point addition operation on the server static public key Ss 101a and the network static public key Sn 102a in order to obtain a point on the elliptic curve defined by the selected subset of cryptographic parameters 104a. The calculation for a step 205 is depicted and described in connection with a step 205 in Figure 2c below. Device 103 can then conduct an ECDH key exchange step 206 using the value from a step 204 and a point from step 205 in order to calculate a symmetric ciphering key K1 206a. The calculation of a symmetric ciphering key K1 206a using a key exchange step 206 is depicted and described in connection with Figure 2c below. Note that a MAC key 206b can also be calculated using a key exchange step 206 as well by device 103. Device 103 can then conduct an encryption step 209 using (i) the symmetric ciphering key 206a and (ii) data to encrypt comprising the data 202b and the random number 203a. The use of an encryption step 209 by device 103 is depicted and described in connection with Figure 2c below. The output of an encryption step 209 can comprise ciphertext 209b.

Device 103 can then use (i) the recorded address for server 101 (possibly from a table 103t) and (ii) connectivity to IP network 107 from step 202 to send a message 299 to server 101. Message 299 and other messages contemplated herein can be sent as either TCP or UDP messages, and other possibilities exist as well for the formatting and transfer of messages without departing from the scope of the present disclosure. In exemplary embodiments, device 103 both uses an IP address and port number to send message 299 to server 101 and then also the same IP address and port number to listen for responses or messages from server 101. In this manner, device 103 can send a message 299 and receive a response message 225 below through an IP network 107, where intermediate nodes on the IP network 107 may conduct network address translation (NAT) routing.

In exemplary embodiments, message 299 from device 103 can include ciphertext 209b and the device ephemeral public key Ed 103a. Although not depicted in Figure 2a, message 299 can also include the selected subset of cryptographic parameters 104a associated with key Ed 103a (and also related keys such as keys 101a, 102a, 103c, and 103d). Message 299 may also include identity information for device 103 or PKI keys recorded by device 103, which could comprise any of (i) a device identity of ID.device 103i as plaintext, (ii) an obfuscated device identity comprising ID.device-obfuscated 103o, or (iii) a secure hash value over device static public key Sd 101d, which could be H(Sd) 250. Message 299 could be with different information for network 105 and/or server 101 to identify device 103 or process message 299, and different embodiments for message 299 are depicted and described in connection with Figure 2b below. The selection of identifying

Petitioner Samsung Ex-1004, 1222

information for device 103 to use with message 299 could be written to device during a step 201c above.

In a message 299, device identity of ID.device 103i can be omitted from plaintext in a message 299, and rather device identity of ID.device 103i could be included within the ciphertext 209b. For embodiments where message 299 optionally excluded device identity ID.device 103i in plaintext, then an identity for device 103i can optionally be transmitted in later messages. Omitting ID.device 103i as plaintext from message 299 can increase security for message 299 since an identity for device 103 would not be sent as plaintext in a message 299. Or, the obfuscated identity for device 103i comprising ID.device-obfuscated 103o could be sent as plaintext in message 299 instead of ID.device 103i. The use of an obfuscated identity of ID-device-obfuscated 103o by a server 101 to map to an ID.device 103i is depicted and described in connection with Figure 2f below. An obfuscated identity for device 103i comprising ID.device-obfuscated 103o could be an apparently random string or number that is uniquely associated with device 103, and the obfuscated identity 103o for device 103 could periodically rotate or change over time.

Although not depicted in Figure 2a, message 299 could also optionally include an identity for key server 102 comprising ID.key-server 102i, such that server 101 can determine which key server 102i should be associated with message 299. Note that an identity for key server 102 of ID.key-server 102i can be omitted from a message 299, and server 101 can select a key server 102 from other means in a step 212 below.

As depicted in Figure 2a, message 299 could also optionally include a secure hash value 250 such as, but not limited to, SHA-256 of the device static public key Sd 103c. Device 103 can send the hash value 250 of key 103c to server 101, in order for server 101 to identify both (i) which device static public key Sd 103c server 101 should utilize for subsequent operations (such as steps 216 and 218 below) and (ii) which of a plurality of possible key servers 102 could be used to process data within message 299, which is further described for a server 101 below.

For embodiments where a secure hash value 250 of key 103c is included in a message 299 (or other identifying information for device 103 such as ID.device 103i or ID.device-obfuscated 103o), then the message 299 could optionally exclude the selected subset of cryptographic parameters 104a. Server 101 receiving the message 299 with the hash value 250 (or device identity information) could determine the set of parameters 104a to use for key Ed 103a based on the hash value 250 (or device identity information) and a server database 101d. Server 101 can use device identity information in a message 299 to select the set of cryptographic parameters 104a from a server database 103 in order to obtain the set of cryptographic parameters 104a to process message 299.

Petitioner Samsung Ex-1004, 1223

As one exemplary embodiment, and as depicted in Figure 2f below, a server database 101d could maintain mapping of hash values 250 and parameters 104a, and server 101 could conduct a query of database 101d using the received hash value 250 in order to select the parameters 104a for further processing and cryptographic operations with key Ed 103a. Or, in an exemplary embodiment cryptographic parameters 104a as transmitted via an IP network 107 or private network 107a could include the secure hash 250 of key 103c, where the secure hash 250 of key 103c can specify which subset of a set of cryptographic parameters 104 to utilize for subsequent ECC cryptographic operations. Although not depicted in Figure 2a, but depicted for a message 299a in Figure 2b below, a message 299 could also include the selected set of cryptographic parameters 104a.

Server 101 can receive message 299 and begin conducting steps in order to process the message. At step 210a, server 101 can determine the subset of cryptographic parameters 104a to use with received device ephemeral public key Ed 103a. Parameters 104a could be determined in a step 210a in several different ways. If parameters 104a are included in message 299, then parameters 104a could be read from message 299. Or, server 101 could use identifying information for device 103 from message 299 in order to select parameters 104a, such as any of (i) ID.device 103i, (ii) ID.device-obfuscated 103o, or (iii) a secure hash value for a public key unique to device 103 such as H(Sd) 250.

Note that for embodiments where device 103 uses a unique key 101v or a unique key 102v, then the hash value for the unique key could be sent in message 299, and server 101 could query a server database 101d in order to look up the cryptographic parameters 104a (and PKI public keys for device 103) associated with the hash value of the unique key 101v or unique key 102v. In another exemplary embodiment, a server 101 with a server DNS name or listening to a server IP:port number could use a selected subset of cryptographic parameters 104a, and the receipt of a key Ed 103a in a message 299 could specify the use of parameters 104a with the received key Ed 103a.

At step 210b, server 101 can conduct a public key validation step on received device ephemeral public key Ed 103a in order to ensure the key is valid and on the selected curve in parameters 104a. Step 210b by server 101 can comprise conducting the steps for an ECC Full Public-Key Validation Routine in section 5.6.2.3.2 of FIPS publication SP 800-56A (revision 2) for the received device ephemeral public key Ed 103a. Alternatively, step 210b can comprise server 101 performing the steps ECC Partial Public-Key Validation Routine in section 5.6.2.3.3 of the same FIPS publication.

Other example steps within a public key validation step 210b can comprise (i) verifying the public key is not at the "point of infinity", and (ii) verifying the coordinates of the point for the

Petitioner Samsung Ex-1004, 1224

public key are in the range [0, p-1], where p is the prime defining the finite field. Other possibilities exist as well for evaluating and validating a received public key is cryptographically secure in a public key validation step 210b, without departing from the scope of the present disclosure.

In exemplary embodiments, a public key validation step 210b can include verifying that a point or public key received in a message such as message 299 is not reused, and in this manner a step 210b can be used to reject messages 299 that could comprise a "replay attack". Server 101 could record in a server database 101d a list of received device ephemeral public keys Ed 103a for an extended period of time, and if any key Ed 103a is reused then message 299 could be rejected. The period of time could be suitable for the application used by device 103 and server 101, including potentially a period of years. As contemplated in the present disclosure a device 103, server 101, and key server 102 can conduct a public key validation step 210b each time a public key or a point on an elliptic curve is received.

At step 211 and after a key validation step 210b, server 101 can query a server database 101d using identifying information for device 101 from message 299 in order to select a device static public key Sd 103c. As mentioned above, message 299 can include identifying information for device 103, such as, but not limited to (i) device identity ID.device 103i, (ii) obfuscated device identity ID.device-obfuscated 103o, or (iii) a secure hash value over the device static public key comprising H(Sd) 250. Server database 101d can record the depicted information for device 103, as well as other information required for conducting the steps contemplated herein, before server 101 receives message 299.

The data for device 103 in a server database 101d could be received from any of a device manufacturer, a device distributor, or a device owner, and other possibilities exist as well. In exemplary embodiments, a step 201b could include a device registration process, where the data for a device 103 in a server database 101d could be recorded in a secure and authenticated manner. Exemplary values and data for a server database 101d recording data for a plurality of devices 103 are depicted and described in connection with Figure 2f below. At step 211, server 101 can record in server database 101d the values device ephemeral public key Ed 103a, and other data received in a message 299, including information to identity device 103.

At step 211, server 101 can also record the originating source IP address and port number 299p (depicted in Figure 2f below) for message 299, in order to subsequently transmit a message 225 below back to the same IP address and port number 299p. In this manner, message 225 below can be routed by intermediate nodes on IP network 107 back to the source IP address and port number used by device 103 to transmit message 299. In other words, (i) the destination IP address and port number of a subsequent message 225 from server 101 to device 103 can comprise the source IP address and port number 299p (depicted in Figure 2f below) received in message 299,

Petitioner Samsung Ex-1004, 1225

and (ii) the source IP address and port number 299p (depicted in Figure 2d below) from message 299 can be recorded in a server database 101d. In this manner, communications with device 103 can be tracked or identified by server 101 during the brief period of time of the message flows in Figure 2a using the source IP address and port number from message 299. A step 211 can also comprise server 101 generating a random number 211a using parameters 104a (e.g. using random length 104g from Figure 3 below) for use in subsequent messages with device 103.

At step 212, server 101 can use identifying information in message 299 for device 103 to select a server static private key ss 101b from a server database 101d, which can correspond to the server static public key Ss 101a recorded and used by device 103. For embodiments where server 101 does not record or operate with server static private key ss 101b, such as embodiments depicted and described in connection with Figure 4 below, then a step 212 by server 101 could comprise server 101 selecting (a) a key server 102x that records or operates with server static private key ss 101b using (b) identifying information for device 103 in message 299. At step 212, server 101 can also select key server 102 for subsequent communications and processing of the received device ephemeral public key Ed 103a.

Note that a system 100 could comprise both a plurality of devices 103 and a plurality of key servers 102. In exemplary embodiments server 101 could select in step 212 the proper key server 102 for conducting subsequent steps in Figure 2a. In other words, without data or values from a message 299, server 101 may know which of a possible plurality of key server 102 may record the network static private key sn 102b for use with or associated with device ephemeral public key Ed 103a. Server 101 could use one of several possible methods for selecting key server 102 in a step 212, including a combination of the following embodiments.

A first embodiment for selecting key server 102 in a step 212 could comprise server 101 selecting the same key server 102 for all keys Ed 103a from all devices 103. For example, for this first method, server 101 could listen or operate on (i) a specific IP address and port number or (ii) with a specific DNS name or URL in step 201b, where the use if (i) or (ii) could be specified or associated with network static public key Sn 102a. As mentioned above for a step 203, device 103 can select the address of server 101 using the server address or name of server 101 recorded with Sn 102a (possibly from a table 103t in Figure 1c). Server 101 could determine that all messages 299 received using (i) or (ii) are associated with a specific key server 102, where the key server 102 records the network static private key sn 102b corresponding to the network static public key Sn 102a recorded by the device 103.

A second embodiment of a step 212 for selecting key server 102 of received message 299 could comprise using an identity of key server 102 in a message 299 from device 103. As described above for a message 299, the message 299 can optionally include an identity for key server 102

Petitioner Samsung Ex-1004, 1226

comprising ID.key-server 102i. For these embodiments, server 101 can select the key server 102 using the ID.key-server 102i in message 299. A third embodiment for a step 212 of selecting key server 102 for received device ephemeral public key Ed 103a could comprise using an identity of device 103 in a message 299 comprising (i) ID.device 103i, (ii) ID.device-obfuscated 101o, or (iii) a secure hash value H(Sd) 250.

A secure hash value over a unique key 102v or unique key 101v could be used as well to determine a key server 102, where a server database 101d could include tables that map the unique secure hash value to a key server 102. As described above for a message 299, the message 299 can optionally include unique identifying information for device 103, and server 101 using database 101d could include a table to map any of (i), (ii), and (iii) in this paragraph to key server 102. For this third embodiment of a step 212, server 101 could conduct a query of server database 101d to select the key server 102 for device 103 using ID.device 103i.

A fourth embodiment for a step 212 to select a key server 102 for received message 299 could comprise using the subset of cryptographic parameters 104a from a step 210a. Server 101 could record that a first subset of cryptographic parameters 104a are associated with a first key server 102, and a second subset of cryptographic parameters 104a are associated with a second key server 102, etc. Other possibilities exist as well for server 101 to conduct a step 212 to select a key server 102 using data in a message 299 without departing from the scope of the present disclosure. For embodiments depicted and described in connection with Figure 4 below, where a second key server 102x also records server static private key ss 101b, then server 101 could use a step 212 as well for selecting the second key server 102x in order to obtain point X2 218a (as depicted and described in Figure 4a).

After selecting key server 102 in a step 212, server 101 can then conduct an EC point addition step 213. ECC point addition 213 can use (i) the recorded device static public key Sd 103c selected by server 101 in a step 211 and (ii) the received device ephemeral public key Ed 103a from a message 299. The output from an ECC point addition step 213 can comprise a point X0 213a. Exemplary details for conducting an ECC point addition step 213 are depicted and described for a server 101 in Figure 2d below.

Server 101 can then send key server 102 a message 214 through the secure session 201a. Message 214 can include an identity for server 101 comprising ID.server 101i, the point X0 213a from a step 213 above, a device identity 103i, and the subset of cryptographic parameters 104a. Note that although the use of a step 213 and point X0 213a are depicted in Figure 2a, a message 214 could include separately both (i) the received device ephemeral public key Ed 103a and (ii) the device static public key Sd 103c, and in this embodiment point X0 213a could be omitted from a message 214. The use of a point X0 213a for a message 214 may be preferred for some

Petitioner Samsung Ex-1004, 1227

embodiments, since it would provide additional security by not exposing key server 102 to the separate values for keys 103a and 103c.

For embodiments where device 103 records network static public key Sn 102a as shared keys 102z, then a device identity 103i in message 214 could be omitted, and key server 102 could use the same network static private key sn 102b for multiple communications representing multiple devices 103 with server 101. In other words, key server 102 could use the server 101 identity 101i in a message 214 for some embodiments to determine the proper server static private key sn 102b for conducting subsequent steps, where server identity 101i is associated with shared keys 102z as depicted in Figure 1c.

For embodiments depicted in Figure 1c where a device 103 records and operates with a unique key 102v, then a message 214 can preferably include identification information for device 103 such as device identity ID.device 103i in order for key server 102 to select the appropriate network static private key sn 102b from database 102d for subsequent calculations. With use of either (i) unique keys 102v or (ii) shared keys 102z, messages 299 and 214 could include a secure hash value over the network static public key Sn 102a. Key server 102 could select the key sn 102b based on the secure hash value of the corresponding public key (e.g. H(Sn 102a)). Or, messages 299 and 214 could include the network static public key Sn 102a used by device 103. Server identity ID.server 103i also can be useful for communications between key server 102 and server 101 for a system 100 and system 200 and other systems herein, since either (i) server 101 may communicate with a plurality of different key servers 102, and/or (ii) key server 102 may communicate with a plurality of different servers 101.

Key server 102 can receive the message 214 via the secure session 201a and conduct a series of steps to process the message and respond. A first step conducted by key server 102 can comprise a key validation step 210b, where the key validation step 210b conducted by key server 102 can be equivalent or compatible with the key validation step 210b conducted by a server 101 as described above. Key server 102 could use the set of cryptographic parameters 104a in message 214 in order to conduct a key validation step 210b.

For a key validation step 210b, a node can reply with a failure or reject message if the key validation step 210a fails, such as if a received ECC public key fails to fall on the named elliptic curve. As mentioned above for a step 210b for server 101, a key validation step 201b for key server 102 could be conducted to verify that point X0 213a is not reused, in order to deny replay attacks. Key server 102 could record all values X0 213a received in a key server database 102d for an extended period of time, which could comprise years, and reuse of a value X0 213a could be rejected. Or, in other exemplary embodiments, the verification that value or point X0 213a is not reused could be omitted by key server 102 in a step 210b, and key server 102 could rely on server

Petitioner Samsung Ex-1004, 1228

101 to deny replay attacks. Or, in other exemplary embodiments, a server 101 could omit the verification that key Ed 103a is not reused, and server 101 could rely on key server 102 verifying that point X0 213a is not reused.

Key server 102 can then conduct a step 215 in order to select a network static private key sn 102b to use with the received point X0 213a from message 214. For a step 215, key server 102 could use (i) device identity 103i received in message 214 to query a key server database 102d to select key sn 102b associated with device identity 103i, or (ii) other identification information for device 103 from a message 299. The other identification information can include a secure hash value over a public key recorded by device 103 or a public key recorded by device 103.

As depicted in Figure 1a, a key server database 102d could record a plurality of different network static private keys sn 102b for use with different devices 103. Or, a step 215 could use other data from a message 214 to select key sn 102b, such as parameters 104a or server identity 101i. In an exemplary embodiment, multiple different devices could record network static public key Sn 102a as shared keys 102a, where the shared keys 102z are associated with a particular set of cryptographic parameters 104a (such as a particular named ECC curve 104c). Key server 102 could select a key sn 102b to use with the shared keys 102z based on the parameters 104a from message 214. Other possibilities exist as well for key server 102 to conduct a step 215 to select a network static private key sn 102b using data from a message 214 without departing from the scope of the present disclosure.

Key server 102 can then conduct an ECDH key exchange step 216 using (i) the selected network static private key sn 102b and (ii) the received point X0 213a from message 214. Or, for some embodiments the values of device ephemeral public key Ed 103a and device static public key Sd 103c could be received separately (and not as one point X0 213) in message 214. Exemplary details for an ECDH key exchange step 216 are depicted and described in connection with Figure 2d below. The output of an ECDH key exchange step 216 can comprise point X1 216a.

For embodiments where message 214 includes either (i) an identity for device 103 such as ID.device 103i, or (ii) identifying information for sn 102b for key server 102 to utilize (such as a secure hash over Sn 102a, where Sn 102a comprises a unique key 101v), then key server 102 could use the identifying information in message 214 to select the network static private key sn 102b from a key server database 102d for ECHD key exchange step 216, where a key server database 102d is depicted in Figure 1a above.

Key server 102 can then send server 101 a message 217, where the message 217 includes point X1 216a from step 216 above, as well as an identity for key server 102 comprising ID.key-server 102i and cryptographic parameters 104a associated with point X1 216a. Message 217 can be transmitted through secure session 201a. If device identity 103i or other identifying information

Petitioner Samsung Ex-1004, 1229

such as a secure hash value over a unique key 102v was included in message 214, then message 217 could also include device identity 103i or the other identifying information for a device 103. Or, both message 214 and message 217 can include a transaction identity or session identity, such that server 101 can associate the received value X1 216a with a received device ephemeral public key Ed 103a and associated device 103.

Server 101 can receive message 217 with point X1 216a and conduct a series of steps in order to derive a mutually shared and authenticated key exchange with device 103. As contemplated herein, the authentication performed between server 101 and device 103 can comprise a "two-way" authentication or mutual authentication between the nodes. Authentication of server 101 or network 105 with device 103 can be provided by a key exchange, since network 105 from system 100 with both server 101 and key server 102 conducts an ECDH key exchange using at least, in part, the network static private key sn 102b, the server static private key ss 101b, and device static public key Sd 103c. Authentication of device 103 with server 101 or network 105 can be provided by the equivalent key exchange conducted by device 103, since device 103 conducts an ECDH key exchange using at least, in part, the device static private key sd 103d.

The "two-way" authentication or mutual authentication from the ECDH key exchange is also not completed until both sides have successfully used a symmetric ciphering key derived from the ECDH key exchange by each side. In other words, a device 103 that successfully mutually derives a symmetric ciphering key with a server 101 can authenticate that server 101 has access to the network static private key sn 102b and the server static private key ss 101b. A server 101 that successfully mutually derives a symmetric ciphering key with device 103 can authenticate that device 103 records or operates with device static private key sd 103d.

One benefit of the system depicted in Figure 2a is that the network static private key sn 102b does not need to be recorded by or operated with server 101. Another benefit is that mutual authentication is derived with forward secrecy, since the symmetric ciphering key is derived with an ephemeral key. A third benefit is that mutual authentication is provided without requiring the use of additional, significant cryptographic steps for a complete PKI-based ECDSA scheme, where a certificate may need to be authenticated to a root certificate or certificate authority public key by a device, and the multiple layers of authentication can require significant processing time, such as an exemplary 10 seconds for a device with a constrained processing resources such as 16 bits data widths running at 50 MHz, while the ECDH key exchanges contemplated herein could require less than 1 second for the same processor.

After receiving message 217, server 101 can conduct a point validation step 210c for received value or point X1 216a. Note that point validation step 210c is related to a key validation step 210b and can use several of the same sub-steps depicted and described for a key validation

Petitioner Samsung Ex-1004, 1230

step 210b for server 101 above. A point validation step 210c is different than a key validation step 210b since (i) the value X1 216a is preferably not used as a public key to be shared with other parties outside network 105 from Figure 1a, but rather (ii) represents a point on the ECC curve from parameters 104a that will subsequently undergo a point addition operation in order to mutually derive a shared secret with device 103.

Further, point X1 216a can be received through a secure session 201a with a trusted party comprising key server 102, and thus the point X1 216a can have a higher level of confidence or trust as being correct and properly formatted than a device ephemeral public key Ed 103a received potentially via the Public Internet. A point validation step 210c for server 101 can comprise verifying that received point X1 211a is on the ECC curve as specified in parameters 104a and that the point is not the "point at infinity". Other possibilities exist as well for conducting a point validation step 210c on the received point X1 216a without departing from the scope of the present disclosure.

After conducting a point validation step 210c, server 101 can then conduct an ECDH key exchange step 218, where a key exchange step 218 is depicted and described in connection with Figure 2d below. In summary, server 101 can input (i) the point X0 213a from a step 213 in Figure 2d and (ii) the server static private key ss 101b into an ECDH key exchange algorithm 207 (in Figure 2d) in order to calculate a point X2 218a. Server 101 can then conduct a key derivation step 219 as depicted and described in connection with Figure 2d below. In summary, server 101 can conduct an EC point addition step 220 (in Figure 2d) using both (i) point X1 216a from message 217 from key server 102 and (ii) point X2 218a from step 218 in order to mutually derive a shared secret X3 207a. Shared secret X3 207a can be input into a key derivation function in order to output a symmetric ciphering key K1 206a and also optionally a MAC key.

Server 101 can then conduct a decryption step 221 (i) using the key K1 206a output from key derivation step 219 in order to (ii) convert ciphertext 209b from message 299 into plaintext. Exemplary details for a decryption step 221 for server 101 are depicted and described in connection with Figure 2e below. Server 101 can then record, operate with, and store data read from the decrypted plaintext in message 299. The decrypted plaintext data could include the random number 203a generated by device 103 in a step 203, and device data 202b, which could include transducer data or other data for device 103. In exemplary embodiments, device data 202b can include device identity 103i and device identity 103i is not transmitted as plaintext in message 299 (but other identifying information could be sent in message 299 such as a hash value for a unique key 101v or 102v, or an obfuscated device identity 103o, etc.). In exemplary embodiments, server 101 stores the plaintext random number 203a in a server database 101d, along with the device identity 101i, as depicted for server database 101d in Figure 2f below.

Petitioner Samsung Ex-1004, 1231

Server 101 can then conduct a step 222 to create a response 222a. Response 222a could be processed or determined by server 101 applying programmatic steps to plaintext device data 202b from a step 221 above. Response 222a could be an "OK" message that the data from device 103 in message 299 has been properly processed. Response 222a could be an instruction or command for device 103 to take particular actions, such as operating an actuator, taking a sensor measurement, requesting user input for device 103, etc.

Server 101 can then conduct an encryption step 223 in order to create a ciphertext2 223b for transmission of data back to device 103. Exemplary details for an encryption step 223 are depicted and described in connection with Figure 2e below, and an encryption step 223 can use a symmetric ciphering algorithm. The symmetric ciphering algorithm and encryption key for a step 223 can comprise the same or equivalent symmetric ciphering algorithm and encryption key used for a decryption step 221 above. Note that the symmetric ciphering key K1 206a can comprise two portions, where a first portion is used by server 101 to decrypt ciphertext1 209b and a second portion is used by server 101 to encrypt ciphertext2 223b.

The plaintext within ciphertext2 223b can comprise at least, in part, the random number 203a and random number 211a. Other data could be included in plaintext for ciphertext2 223b such as, but not limited to the response 222a from a step 222 above as well as parameters 104a, without departing from the scope of the present disclosure. As depicted in Figure 2a, the collection of steps beginning with step 201a through step 222 can comprise a step 224. A step 224 is depicted in other figures below as a condensed or consolidated summary of the individual steps.

Server 101 can then send device 103 a message 225, where the destination IP address and port number of message 225 can comprise the source IP address and port number 299p received with message 299 and recorded in server database 101d. Message 225 can include the device ephemeral public key Ed 103a and the ciphertext2 223b, as depicted in Figure 2a. Message 225 can include identifying information for device 103, such as ID.device 103i or ID.device-obfuscated 103o, or a hash value of a unique key 101v or 102v received in a message 299. The value "K1 206a" depicted in Figure 2a is shown to illustrated that the derived symmetric ciphering key 206a from a key derivation step 219 is used to encrypt ciphertext 223b (indicated by the brackets shown in Figure 2a for message 225), and the value K1 206a is not transmitted in message 225.

Note that key Ed 103a can be omitted from message 225 since device 103 records the data. However, key Ed 103a can be useful to confirm that server 101 is sending message 225 to the correct device 103, and other benefits exist as well. Although not depicted in Figure 2a, message 225 can include a server ephemeral public key Es 101p from Figure 1a and a third ciphertext, where the third ciphertext is encrypted using the server ephemeral private key es 101s. Ciphertext2 223c can be internal or external to the third ciphertext.

Petitioner Samsung Ex-1004, 1232

As contemplated in the present disclosure, a message such as message 225 and also other messages such as message 299, message 214, etc. can be transmitted or sent in parts, where the data for the message can be transmitted and received in separate datagrams or portions over time. For these embodiments, the message can comprise the collection of separate datagrams or portions transmitted or sent separately. For example, with separate datagrams or portions for a message 299 in Figure 2a, a first datagram or portion for message 299 could comprise device ephemeral public key Ed 103a, which could be sent (i) after a key pair generation step 103x, and (ii) before device 103 conducts steps 204 through 209 in a step 231. A second datagram or portion for message 299 could comprise ciphertext 209b, which could be sent after device 103 sends key Ed 103a.

In this manner, by sending message 299 as a first portion and a second portion, the overall speed of (i) conducting a step 231 for device 103 and (ii) steps 210a through 213 for server 101 could be increased. For example, by receiving the first portion of message 299 comprising key Ed 103a, server 101 could then (a) begin conducting steps below of 210a, 210b, etc., while (b) waiting for the second portion of message 299 comprising ciphertext 209b to be sent separately and after the first portion. By increasing the overall speed for conducting a step 224 for device 103 (and server 101), then electrical power consumption or battery usage for device 103 can be reduced. Other possibilities and benefits from sending a message in the present disclosure as a first portion and a second portion, without departing from the scope of the present disclosure.

Device 103 can then receive message 225 and conduct a series of steps in order to process the message. Device 103 can perform a decryption step 221a in order to decrypt ciphertext2 223b from message 225 using the derived symmetric ciphering key K1 206a from the key exchange step 206 for device 103. The symmetric ciphering key K1 206a derived by device 103 could include two portions, where a first portion is for encryption and a second, different portion is for decryption. A decryption step 221a for device 103 can be equivalent to a decryption step 221 for server 101, except (i) the ciphertext2 223b is input into decryption step 221a for device 103, and (ii) ciphertext 209b is input into decryption step 221 for server 103. A decryption step 221 is also depicted and described in connection with Figure 2e below.

Device 103 can then read the plaintext within ciphertext2 223b, as well as verifying message integrity of ciphertext2 223b using a MAC key derived in a step 206. Note that for all ciphertext values transmitted in Figure 2a, the ciphertext can include a MAC code and an initialization vector. Or, the MAC code and the initialization vector could be omitted from being transmitted with the ciphertext from some symmetric ciphering algorithms such as AES-SIV (e.g. IETF RFC 5297). Device 103 in a decryption step 221a can read the plaintext values of random number 203a, random number 211a, and response 222a from server 101. The plaintext data could include other data from server 101 or network 105 as well.

Petitioner Samsung Ex-1004, 1233

At step 226, device 103 can determine if decryption step 221a was successful, such as the ability to read plaintext data including the random number 203a send by device 103 in a message 299. If decryption step 221a fails, such as not being able to read plaintext data, then device 103 could determine that server 101 and/or network 105 did not properly record and operate with the private keys corresponding to the selected public keys from a network public key table 103t. For example, decryption step 221a could fail if either (a) server 101 did not record the private key ss 101b corresponding to the public key Ss 101a recorded by device 101, or (b) server 101 did not have access to a key server 102 for calculating values with private key sn 102b corresponding to the public key Sn 102a recorded by device 103.

As one example for failure of a decryption step 221a, the entity controlling key server 102 may deny server 101 access to computations with key sn 102b for reasons that could be equivalent to a key revocation for a traditional PKI key using certificates. Or, the entity controlling key server 102 may consider server 101 or key ss 101b to be compromised, and thus device 103 should stop using the corresponding public key. Other reasons could exist as well for the failure of a decryption step 221a, but a network 105 could intentionally cause decryption step 221a to fail by denying calculations with private keys sn 102b and/or ss 101b.

Upon determining that decryption step 221a failed in a step 226, device 103 could then select another server and corresponding keys from a network public key table 103t. For example, and as depicted in Figure 1c, a device 103-1 could use the first row as a primary communications path and method for communicating with network 105. If decryption step 221a fails, then at a step 226 then device 103 could select a second row or "backup" to the primary, such as selecting the keys and data for row 2 in Figure 1c for device 103-a. Device 103 could then return to the top of a step 224 and conduct a step 231 again using the data from the second row of a network public key table 103t. In this manner, a network 105 could deprecate a row of keys in a network public key table 103t, while device 103 could continue to communicate with new or different keys by using a step 231 and 228 with the new or different keys from table 103t (where network 105 caused a decryption step 221a to intentionally fail). Other reasons could exist for device 103 to use a different row of keys in a network public key table 103t as well, such as an instruction in a response 222a.

At step 227, device 103 can process the plaintext data from message 225, which was decrypted by step 221a. Device 103 can determine if random number 203a received in message 225 equals random number 203a previously sent in a message 299 above. Device 103 can record random number 211a from server 101 and use the random number 211a for subsequent messages to server 101 after receiving message 225. Device 103 in a step 227 could also process response 222a from server 101, where response 222a could include an instruction for device 103, such as an

Petitioner Samsung Ex-1004, 1234

actuator setting, a command to change state of device 103, a setting or value or parameter for device 103 to begin using, and other possibilities exist as well for the data in a response 222a. The data for a response 222a could depend on the application used by device 103, and exemplary different applications are depicted and described in connection with Figures 5a through 5d below. As depicted in Figure 2a, the collection of steps comprising message 225 and steps 221a, 226, and 227 could collectively comprise a step 228 for device 103 and server 101.

Although not depicted in Figure 2a, device 103 could then send server 101 a subsequent message such as an encrypted signal of "OK" or other data using symmetric ciphering key K1 206a. Device 103 could send server 101 the "OK" message (or any message) with at least the random number 211a from server 101 in a ciphertext. Receipt by the server 101 of the random number 211a can authenticate the device 103 with the server 101. The ciphertext of a message from device 103 to server 101 after a message 225 could be encrypted with either the symmetric ciphering key K1 206a or a subsequently derived, different second symmetric ciphering key where the second symmetric ciphering key could be derived using the data from message 225.

Message 225 could include a server ephemeral public key Es 101p, then for a step 228, device 103 could use the key Es 101p with other keys such as keys Ss 101a, sd 103c, ed 103b, etc. in order to derive the second symmetric ciphering key. For some embodiments, message 225 could include a third ciphertext encrypted with the second symmetric ciphering key. In exemplary embodiments, device 103 sends server 101 a message with ciphertext after message 225, where (i) the ciphertext includes the random number 211a and (ii) the symmetric ciphering key for the ciphertext is derived using mutual authentication.

Figure 2b

Figure 2b is a simplified message flow diagram illustrating an exemplary messages transmitted from a device to a server, in accordance with exemplary embodiments. System 200 in Figure 2b can be equivalent to system 200 in Figure 2a. System 200 in Figure 2b can include a device 103 and a server 101. As depicted and described in connection with Figure 2a, a message 299 transmitted from device 103 to server 101 can comprise several different embodiments, including the use of different possible values for identification information for a device 103 in a message 299. Figure 2b depicts different embodiments for a message 299 from a device. Although the message 299 may comprise different possible embodiments, a server 101 and a key server 102 can conduct the same or equivalent steps as depicted and described in connection with Figure 2a in order to conduct a step 224 and a step 228.

Petitioner Samsung Ex-1004, 1235

A first exemplary embodiment for a message 299 could comprise a message 299a for a step 224a. For message 299a, the message can include the depicted values of device ephemeral public key Ed 103a, ciphertext 209b, a digital signature 103s, and a selected set of cryptographic parameters 104a. Note that identifying information for device 103 is omitted from plaintext a message 299a, but device identity comprise ID.device 103i is included in ciphertext 209b within message 299a. In this manner, device identity 103i can remain secured and not transmitted over IP network 107 in an insecure manner.

For embodiments with message 299a and without identity information for device 103, then server static public key Ss 101a recorded by device 103 could comprise a shared key 101z as depicted and described in connection with Figure 1c. Or, in some embodiments a device 103 could omit recording Ss 101a (as discussed in Figure 2c and Figure 2d below), and authentication of a network 105 could be performed for device 103 with network static public key Sn 102a.

Server 101 could receive message 299a and conduct subsequent operations in a common manner for all devices 103 transmitting the same set of cryptographic parameters 104a. For a message 299a, server 101 could conduct a step 212a, where a step 212a could select the same key server 102 for sending message 214 in Figure 2a for each message 299a received (e.g. key Ed 103a is forwarded by server 101 to key server 102 in a message 214). Note that in a step 224a that a modified step 213 is utilized, comprising a step 213x. For a step 213x, a device static public key Sd 103c cannot normally be selected by server 101 since no identifying information for device 103 is received as plaintext in a message 299a.

Consequently, a step 213x can comprise server 101 omitting a key Sd 103c from a point addition operation 213x and in this embodiment point X0 213a can comprise the point Ed 103a. All subsequent operations by server 101 and key server 102 in a step 224a with point X0 213a could use point Ed 103a as the point X0 213a. Key server 102 can still receive point X0 213a and reply with point X1 216a, where key server 102 uses the same value or number for key sn 102b for all points X0 213a received. Server 101 can derive key K1 206a using a key derivation step 219. Server 101 can then decrypt ciphertext 209a using key K1 206a with a decryption step 221. Server 101 can subsequently read the device identity ID.device 103i from the plaintext in ciphertext 209a.

Server 101 could conduct a key exchange step 218 using the same key ss 101b for all messages 299a received with a given set of cryptographic parameters 104a. Note that for embodiments depicted in Figure 2b with a step 224a, that decryption of ciphertext 209a does not provide authentication for device 103, since device 103 has not demonstrated use and cryptographic operations with device static private key sd 103d. For a system 224a, other steps could be conducted in order to establish authentication for device 103 with server 101 or network 105. As

Petitioner Samsung Ex-1004, 1236

one example, device 103 could include a digital signature 103s with message 299a, and digital signature 103s could be over at least the key Ed 103a.

After reading device identity 103i from ciphertext 209b in a message 299a, server 101 could select a device static public key Sd 103c from a database 101d and conduct a signature verification step 233 for the digital signature 103s in a message 299a using the selected device static public key Sd 103c. Upon successful verification of a digital signature 103s from a message 299a in a signature verification step 233, then device 103 could be considered authenticated with server 101 and/or network 105. In this manner of including a digital signature 103s with message 299a, a device 103' can be authenticated with a server 101 and/or a network 105 using a single message 299a. Note that the other steps and messages from Figure 2a would be utilized as well, such as a step 210b to confirm that key Ed 103a is not retransmitted, and thus preventing replay attacks.

A second exemplary embodiment for a message 299 could comprise a message 299b for a step 224b. For message 299b, the message can include the depicted values of device ephemeral public key Ed 103a, a secure hash value over server static public key Ss 101a comprising H(Ss) 252, and ciphertext 209b. For a step 224b, a device 103 could record server static public key Ss 101a in a network public key table 103t as a unique key 101v. Recording record server static public key Ss 101a in a network public key table 103t as a unique key 101v is depicted and described in connection with Figure 1c above. Although H(Ss) 252 is depicted for a message 299b in Figure 2b, a hash value for a different unique public key recorded by device 103 in table 103t could be utilized as well or instead, such as a secure hash value over network static public key Sn 102a, which could be equivalent to H(Sd) 250 as depicted for a message 299 in Figure 2a. For embodiments using H(Sn), then the network static public key Sn 102a recorded by device 103 could also comprise a unique key 102v as depicted and described in connection with Figure 1c.

Server 101 could receive message 299b and conduct steps to process the message. Server 101 could conduct a step 210a to select a set of cryptographic parameters 104a for use with PKI keys such as Ed 103a from message 299b. Server 101 could conduct a step 210b to validate public key Ed 103a and also confirm that key Ed 103a is not reused. A step 212b in Figure 2b can be equivalent to a step 212 in Figure 2a, but with modifications for server 101 to identify device 103. A step 212b could comprise server 101 reading H(Ss) 252 and querying server database 101d as depicted in Figure 2f below in order to obtain PKI keys for device 103 in order to conduct subsequent steps.

Since Ss 101a can comprise a unique key 101v, the hash value of H(Ss) 252 would also reasonably be a unique number or value for querying database 101d to obtain keys Sd 101c, Ss 101a, and also a key server 102 that would record network static private key sn 102b corresponding to the network static public key Sn 102a recorded by device 103. As contemplated herein, a hash

Petitioner Samsung Ex-1004, 1237

value over a public key, such as H(Ss) 252 or H(Sd) 250 can reasonably be considered a unique number to identify a device 103, where unique public keys 101v and/or 102v are recorded in a device 103, as depicted and described in connection with Figure 1c. After selecting keys Sd 101c, Ss 101a, and also a key server 102 that would record network static private key sn 102b in a step 212b, server 101 could then conduct the subsequent series of steps for server 101 as depicted and described in connection with Figure 2a with key server 102 and also device 103.

A third exemplary embodiment for a message 299 could comprise a message 299c for a step 224c. For message 299c, the message can include the depicted values of device ephemeral public key Ed 103a, device static public key Sd 103c, and ciphertext 209b. Before processing message 299c, a server 101 could separately record the same value for device static public key Sd 103c, in a server database 101d as depicted and described in connection with Figure 2f below. As depicted in Figure 2b, the value for device static public key Sd 103c, potentially along with a device identity 103i, could be received and recorded in a step 201b, although other possibilities exist as well for a server 101 to receive key Sd 103c before receiving a message 299c. In other words, server 101 could obtain key Sd 103c separately and in a secure manner before receiving the same value for a key Sd 103c from a message 299c.

Note that the separate receipt and recording of Sd 103c, such as in a step 201b, can be important for security of a system 100, system 200, and other systems herein, since authentication of device 103 using Sd 103c can rely on the security and trust of the method used to receive and record key Sd 103c. In other words, without the separate secure recording of key Sd 103c by server 101 before receiving a message 299c, an imposter device 103 or a hacker could simply derive and send both key Ed 103a and key Sd 103c in a message 299c.

A step 212c could comprise server 101 reading Sd 103c and querying server database 101d as depicted in Figure 2f below in order to confirm that key Sd 103c received in message 299c matches a previously securely received and recorded same value or number for a key Sd 103c in a database 101d. Server 101 could then also query server database 101d in a step 212c for other PKI keys for device 103 in order to conduct subsequent steps. Server 101 could use key Sd 103c to obtain (i) key ss 101b and (ii) the identity for a key server 102 which could record network static private key sn 102b corresponding to the network static public key Sn 102a used by device 103 in a key derivation step 206 in step 230.

Server 101, key server 102, and device 103 could then conduct the series of steps and messages depicted in Figure 2a in order to conduct a mutual authentication and securely and confidentially transmit data 106 between the nodes. Other possibilities exist as well for a message 299 to include identifying information for device 103 in order for server 101 to conduct a step 212 without departing from the scope of the present disclosure.

Petitioner Samsung Ex-1004, 1238

Figure 2c

Figure 2c is a flow chart illustrating exemplary steps for conducting a key exchange using PKI keys in order to derive a shared secret key, and for using the derived shared secret key to encrypt and decrypt data, in accordance with exemplary embodiments. Exemplary steps for a device 103 to mutually derive a shared secret X3 207a and symmetric key 206b with server 101 can comprise a key exchange step 206. Exemplary steps in Figure 2c for a device 103 to encrypt plaintext data using the mutually derived symmetric key 206b can comprise an encryption step 209. The use of the steps for a key exchange 206 and encryption 209 were also depicted and described in connection with Figure 2a above.

Note that steps in Figure 2c and the steps in Figure 2d below can share some algorithms and values, and the descriptions for the algorithms and values in Figure 2c can be applicable for Figure 2d. For example, the key exchange algorithm 207 in Figure 2c can comprise an ECDH key exchange equivalent to key exchange algorithm 207 in Figure 2d (but with different numbers input for the algorithm in the two different Figures). The set of parameters 104a depicted and described in Figure 2c can also be used in Figure 2d.

The processes and operations, described below with respect to all of the logic flow diagrams and flow charts may include the manipulation of signals by a processor and the maintenance of these signals within data structures resident in one or more memory storage devices. For the purposes of this discussion, a process can be generally conceived to be a sequence of computer-executed steps leading to a desired result.

These steps usually require physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, compared, or otherwise manipulated. It is convention for those skilled in the art to refer to representations of these signals as bits, bytes, words, information, elements, symbols, characters, numbers, points, data, entries, objects, images, files, or the like. It should be kept in mind, however, that these and similar terms are associated with appropriate physical quantities for computer operations, and that these terms are merely conventional labels applied to physical quantities that exist within and during operation of the computer.

It should also be understood that manipulations within the computer are often referred to in terms such as listing, creating, adding, calculating, comparing, moving, receiving, determining, configuring, identifying, populating, loading, performing, executing, storing etc. that are often associated with manual operations performed by a human operator. The operations described

Petitioner Samsung Ex-1004, 1239

herein can be machine operations performed in conjunction with various input provided by a human operator or user that interacts with the device, wherein one function of the device can be a computer.

In addition, it should be understood that the programs, processes, methods, etc. described herein are not related or limited to any particular computer or apparatus. Rather, various types of general purpose machines may be used with the following process in accordance with the teachings described herein.

The present disclosure may comprise a computer program or hardware or a combination thereof which embodies the functions described herein and illustrated in the appended flow charts. However, it should be apparent that there could be many different ways of implementing the disclosure in computer programming or hardware design, and the disclosure should not be construed as limited to any one set of computer program instructions.

Further, a skilled programmer would be able to write such a computer program or identify the appropriate hardware circuits to implement the disclosure without difficulty based on the flow charts and associated description in the application text, for example. Therefore, disclosure of a particular set of program code instructions or detailed hardware devices is not considered necessary for an adequate understanding of how to make and use the disclosure. The inventive functionality of the claimed computer implemented processes will be explained in more detail in the following description in conjunction with the remaining Figures illustrating other process flows.

Further, certain steps in the processes or process flow described in all of the logic flow diagrams below must naturally precede others for the present disclosure to function as described. However, the present disclosure is not limited to the order of the steps described if such order or sequence does not alter the functionality of the present disclosure. That is, it is recognized that some steps may be performed before, after, or in parallel other steps without departing from the scope and spirit of the present disclosure.

The processes, operations, and steps performed by the hardware and software described in this document usually include the manipulation of signals by a CPU or remote server and the maintenance of these signals within data structures resident in one or more of the local or remote memory storage devices. Such data structures impose a physical organization upon the collection of data stored within a memory storage device and represent specific electrical or magnetic elements. These symbolic representations are the means used by those skilled in the art of computer programming and computer construction to most effectively convey teachings and discoveries to others skilled in the art.

A device 103 can conduct a key exchange step 206. At step 206, device 103 can use a step 204 to calculate the sum of device static private key sd 103d and the derived device ephemeral private key ed 103b, and then calculate the modulus of the sum using the value n, where the value

Petitioner Samsung Ex-1004, 1240

n can be defined or determined based upon the selected set of cryptographic parameters 104a. Exemplary data for a step 204 will be shown below. Device 103 can then conduct a step 205 to perform an EC point addition operation on the server static public key Ss 101a and the network static public key Sn 102a in order to obtain a point on the elliptic curve defined by the selected subset of cryptographic parameters 104a. Exemplary data for a step 205 will also be shown below.

The combination of output from step 204 and step 205 can be input can be input into an ECDH key exchange algorithm 207 using parameters 104a in order to calculate the shared secret X3 207a. Note that in some exemplary embodiments, the use of a server static PKI key pair comprising Ss 101a and ss 101b could be omitted, and for these embodiments then ECC point addition for step 205 can be omitted and the public key input into an ECDH key exchange algorithm 207 can comprise the single value for network static public key Sn 102a. In other words, a key X3 207a can be calculated without requiring a server static public key Ss 101a for these embodiments (where private key ss 101b would also be omitted from key exchange calculations by network 105), although the use of a server static public key Ss 101a can be preferred for a step 205 in other embodiments.

A summary of ECDH as a key exchange algorithm 207 is included in the Wikipedia article titled "Elliptic Curve Diffie-Hellman" from March 9, 2018, which is herein incorporated by reference. An exemplary embodiment of key exchange algorithm 207 could comprise a "One-Pass Diffie-Hellman, C(1, 1, ECC CDH)" algorithm as described in section 6.2.2.2 on page 81 of the National Institute of Standards and Technology (NIST) document "NIST SP 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" from March, 2007 which is hereby incorporated by reference its entirety.

Other key exchange algorithms in NIST SP 800-56A could be utilized as well for a key exchange algorithm 207 in Figure 2a and Figure 2d without departing from the scope of the present disclosure. Example calculations for an ECDH key exchange for a key exchange algorithm 207 are shown below. In exemplary embodiments, the key exchange algorithm 207 used by server 101 and key server 102 can comprise the equivalent key exchange algorithm 207 used by device 103 in a key exchange step 206 above in Figure 2c.

Other algorithms to derive a secret keys using public keys and private keys may also be utilized in a key exchange algorithm 207, such as, but not limited to, the American National Standards Institute (ANSI) standard X-9.63. Cryptographic parameters 104a can also include information, values, or settings for conducting (i) a key exchange algorithm 207 in step 206 (and steps 216 and 218 in Figure 2d below) and (ii) a key derivation function 208 in order to derive a commonly shared symmetric encryption key K1 206a. As contemplated herein, the terms "selected set of cryptographic parameters 104a" and "cryptographic parameters 104a", and "parameters

Petitioner Samsung Ex-1004, 1241

104a" can be equivalent, and can also comprise a subset of exemplary cryptographic parameters depicted and described in connection with Figure 1a and Figure 2g below.

Parameters 104a input into a key exchange algorithm 207 can include a time-to-live for a key K1 206a that is derived, a supported point formats extension, where the supported point formats extension could comprise uncompressed, compressed prime, or "compressed char2" formats, as specified in ANSI X-9.62 and related IETF standards. In other words, (i) an ECC keys input into a key exchange algorithm 207 and (ii) secret keys output from key exchange algorithm 207 may have several different formats and a set of parameters 104a can be useful to specify the format.

Exemplary data and numbers can be provided to demonstrate the calculations for (i) EC point addition step 205, (ii) step 204 to combine private keys, and (iii) key exchange step 206. Parameters 104a can comprise the elliptic of "secp128r1" with key lengths of 128 bit long keys. Although the exemplary numbers for keys, points, and a named curve are provided, other values for keys, points, and named ECC curves could be utilized as well. Other example named curves and parameters 104a could comprise curve from IETF RFC 5480.

The device ephemeral private key ed 103b can comprise the exemplary following number, and can be recorded in device 103 after a key pair generation step 103x from Figure 1a and Figure 2a above:

330225223315954763304200471137380016969

The device static private key sd 103d can comprise the exemplary following number, and can be recorded in device 103 before key pair generation step 103x, such as during device manufacturing or device distribution:

209122135174501513062984245101620420255

Note that the private keys ed 103b and sd 103d above correspond to the public keys Ed 103a and Sd 103c recorded by a network 105, as depicted in Figure 1a and also with exemplary numeric values below in Figure 2d below for the example numbers given above. The network static public key Sn 102a can comprise the exemplary values with X and Y numbers (or "coordinates") of:

X: 941715349842376856782565856182414170 39

Y: 203945269464835729838690547089813292056

Note that the public key Sn 102a corresponds to the private key sn 102b from Figure 2d below. The server static public key Ss 101a can comprise the following exemplary values with X and Y numbers (or "coordinates") of:

Petitioner Samsung Ex-1004, 1242

X: 319423829544285733939020505180109110187

Y: 242179187598040154943588326777101424083

An ECC point addition 205 for a step 206 with the above two keys Ss 101a and Sn 102a will result in the following exemplary point for input into an ECDH key exchange algorithm 207 in step 206:

X: 157250524327743828409297614402748325 89

Y: 217317805140710190286653933543727803288

The above combination of both Ss 101a and Sn 102a for a key exchange step 206 via an EC point addition 205 is depicted in Figure 2c with the "+" symbol between the public keys.

Device 103 can conduct step 204 to calculate the sum of device static private key sd 103d and the derived device ephemeral private key ed 103b, and then calculate the modulus of the sum using the value n, where the value n can be defined or determined based upon the selected set of cryptographic parameters 104a. For the exemplary values for keys above, when using the named elliptic curve secp128r1, the value of n can comprise the decimal number:

340282366762482138443322565580356624661

Consequently, the modulus of (i) the sum of private keys sd 103d and ed 103b and (ii) the value n above will equal the following number for a step 204:

199064991727974137923862150658643812563

The output of the above ECC point addition 205 for public keys Ss 101a and Sn 102a can be input into ECDH key exchange algorithm 207 using parameters 104a. An ECDH key exchange algorithm 207 in key exchange step 206 can input (i) the point calculated above from the EC point addition 205 on the public keys 101a and 102a and (ii) the value calculated from a step 204 (e.g. (sd 103d + ed 103b) mod n). The output of ECDH key exchange algorithm 207 in key exchange step 206 can be the secret value or point X3 207a. Note that the secret X3 207a as derived by device 103 in a key exchange step 206 equals or is the same numeric value as the secret X3 207a derived by server 101 in a key derivation step 219 below. An exemplary number or value for secret X3 207a calculated by device 103 using a key exchange step 206 using the above exemplary numeric values for ed 101b, sd 103d, Sn 102a, and Ss 101a would be:

X: 192457465648897421085529769283600671459

Y: 126144984806909677418281309675999642 69

Petitioner Samsung Ex-1004, 1243

Although the use of two public keys (Sn 102a and Ss 101a) and two private keys (ed 101b and sd 103d) are depicted for device 103, an ECDH key exchange step 206 could include more keys than those depicted. Additional public keys, such as a server ephemeral public key Es 101p could be input into the EC point addition step 205 by device 103, and a server 101 could input the corresponding server ephemeral private key es 101s into a step 218 below in Figure 2d (via modular addition with key ss 101b). The present disclosure contemplates that additional or different public and private keys could be used with a key exchange step 206 for device 103 and key exchange steps 216 and 218 by a network 105.

For a key exchange step 206, derived shared secret key X3 207a can be input into a key derivation function 208 where the key derivation function 208 can be equivalent to the key derivation function 208 depicted and described in connection with Figure 2d below for a key derivation step 208. Note that for key derivation steps in the present disclosure, the X coordinate of a derived shared secret can be taken or used as input into the key derivation function 208. The output of a key derivation function 208 can comprise both (i) a symmetric ciphering key K1 206a and (ii) a MAC key 206b. MAC key 206b can be used with a symmetric ciphering algorithm in order to generate a MAC code, such that the other party using the same key K1 206a and MAC key 206b can process the ciphertext and calculate the same MAC code in order to verify message integrity. The use of key K1 206a and MAC key 206b are described in connection with encryption step 209 and decryption step 221.

Key derivation function 208 can use a secure hash algorithm such as, but not limited to, SHA-256, SHA-384, SHA-3, etc. and additional values such as a text string with secret X3 207a. The specification of a secure hash algorithm and the text string for use with a key derivation function 208 could be commonly shared between server 101 and device 103 by commonly shared parameters 104a. The output of a secure hash algorithm within a key derivation function 208 could have a subset of bits selected or possibly a secure hash expanded in order to obtain the number of bits required for a symmetric key with a symmetric ciphering algorithm, such as key K1 206a.

A key derivation function (KDF) 208 could comprise a KDF compatible with or specified by ANSI standards for "X9.63 Key Derivation Function". Other possibilities exist for a key derivation function 207 to convert a secret X3 207a into a symmetric ciphering key K1 206a and a MAC key 206b without departing from the scope of the present disclosure. As contemplated in the present disclosure, although an ECC public key such as secret X3 207a can comprise a coordinate with an X value and a Y value, in exemplary embodiments a single number comprising the X value can be selected and input into a key derivation function 208. Symmetric ciphering key K1 206a can

Petitioner Samsung Ex-1004, 1244

comprise two portions, where a first portion is used for encryption and a second portion is used for decryption.

Device 103 can conduct an encryption step 209, where the use for an encryption step 209 is depicted and described in connection with Figure 2a above. Plaintext 209a in a step 209 can comprise the random number 203a from device 103 and data 202b, where data 202b for device 103 was depicted and described in Figure 2a in connection with a step 202. Other or different exemplary data could be included as plaintext 209a in an encryption step 209. The symmetric ciphering key for encryption step 209 can comprise symmetric key K1 206a from a key derivation step 208 in step 206 above and a MAC key 206b can be input into a symmetric ciphering algorithm 230 as well. Encryption step 209 in Figure 2c and decryption step 221 in Figure 2e can use a common symmetric ciphering algorithm 230, which could comprise the Advanced Encryption Standard with Synthetic Initialization Vectors (AES-SIV) (and deciphering algorithm) also with a common set of symmetric ciphering parameters 104f from a set of cryptographic parameters 104. Note that MAC key 206b can also be input into symmetric ciphering algorithm 230 along with an initialization vector 209i.

Other or different symmetric ciphering algorithms 230 could be utilized as well, such as, but not limited to such as AES, Triple Data Encryption Standard (3DES), Blowfish, or related algorithms. Symmetric ciphering parameters 104f can also specify the use of a block chaining mode such as cipher block chaining (CBC), counter mode (CTR), or Galois/Counter mode (GCM) and other possibilities exist as well. In addition, symmetric ciphering parameters 104f could specify a mode for message authentication, which could comprise a CMAC mode as specified in NIST publication SP-800-38B. In some exemplary embodiments, a symmetric ciphering algorithm 230 can comprise the AES-SIV algorithm as specified in IETF RFC 5297. The output from an encryption step 209 using a symmetric ciphering algorithm 230 and the depicted values input can be ciphertext 209b, as depicted in Figure 2c. The output from an encryption step 209 using a symmetric ciphering algorithm 230 and the depicted values input can also include MAC code 209m, where MAC code 209m can be used by the receiving party with the MAC key 206b to verify message integrity. The initialization vector 209i can be sent along with the ciphertext 209b in order for both sides to commonly initiate block chaining.

In addition, although a key exchange step 206 for device 103 in Figure 2c depicts (i) the point addition of the public keys Ss 101a and Sn 102a, and (ii) the addition of private keys ed 103b and sd 103d, other sequences of EC point addition and EC point multiplication are possible as well without departing from the scope of the present disclosure in order to derive the same value for X3 207a. In summary, X3 207a can be calculated by device 103 through any of the following combinations in addition to those described above, and other possibilities may exist as well:

Petitioner Samsung Ex-1004, 1245

1. X3 207a = [(Ss 101a * {(ed 103b + sd 103d) mod n}] + [(Ss 101a * {(ed 103b + sd 103d) mod n}]
2. X3 207a = [(Ss 101a + Sn 102a) * ed 103b] + [(Ss 101a + Sn 102a) * sd 103d]
3. X3 207a = [(Ss 101a + Sn 102a) * ed 103b] + [(Ss 101a * sd 103d) + (Sn 102a * sd 103d)]
4. X3 207a = [(Ss 101a * ed 103b) + (Sn 102a * ed 103b)] + [(Ss 101a + Sn 102a) * sd 103d]
5. X3 207a = [(Ss 101a * ed 103b) + (Sn 102a * ed 103b)] + [(Ss 101a * sd 103d) + (Sn 102a * sd 103d)]

In summary, the present disclosure contemplates that and ECDH key exchange by device 103 in order to derive a point X3 207a can include at least one point addition using the server static public key Ss 101a and the network static public key Sn 102a. For example, combinations above multiply the keys Ss 101a and Sn 102a individually by a scalar value via EC point multiplication before EC point addition operations. However, the calculation of point X3 207a uses at least one EC point addition of (i) a first value calculated with the server static public key Ss 101a and (ii) a second value calculated with the network static public key Sn 102a.

Figure 2d

Figure 2d is a flow chart illustrating exemplary steps for conducting an elliptic curve (EC) point addition operation to combine public keys, for conducting a key exchange using PKI keys in order to derive shared secrets, and for conducting a key derivation function using the derived shared secrets, in accordance with exemplary embodiments. Server 101 can conduct an EC point addition operation 213 to derive secret key X0 213a. Key server 102 can conduct a key exchange step 216 in order to derive a secret key X1 216a. Server 101 can conduct a key exchange step 218 in order to derive a secret key X2 218a. Server 101 can receive the secret key X1 216a in a message 214 from key server 102 in Figure 2a above through a secure connection 201a. Server 101 can then conduct a key derivation function 219 using secret keys X1 216a and X2 218a in order to derive a symmetric ciphering key K1 206a.

Using the methods and ECC PKI keys described in the present disclosure, a device 103 can also derive the same symmetric ciphering key K1 216a as depicted and described below for a key exchange step 206 in Figure 2c above. In other words, for exemplary embodiments (i) the corresponding key exchange step 206 (in Figure 2c above) for a device 103 by network 105 can be (ii) shared or distributed between a server 101 and key server 102 in order to secure or isolate network static private key sn 102b.

Petitioner Samsung Ex-1004, 1246

Server 101 can conduct an EC point addition step 213 in order to combine (i) device static public key Sd 103c recorded in a server database 101d with (ii) device ephemeral public key Ed 103a received in a message 299. Exemplary calculations for an ECC point addition 213 can comprise the calculations shown for point addition in the Wikipedia article for "Elliptic Curve Point Multiplication" dated May 15, 2018, which is herein incorporated by reference in its entirety. The output of an EC point addition step 213 can comprise the point or value X0 213a. An EC point addition step 213 can be performed with a named elliptic curve from a selected set of parameters 104a, where server 101 selects parameters 104a in a step 210a.

A key exchange step 216 for key server 102 to derive a secret key X1 216a can utilize a selected set of cryptographic parameters 104a as depicted and described in connection with Figure 1a and Figure 2a above. Parameters 104a could be received by key server 102 along with the point X0 213a in a message 214 from server 101. As depicted in Figure 2d, a key exchange algorithm 207 in step 216 for key server 102 can receive input both of point X0 213a and network static private key sn 102b. The key exchange algorithm 207 could comprise a Diffie Hellman key exchange (DH), an Elliptic Curve Diffie Hellman key exchange (ECDH), and other possibilities exist as well without departing from the scope of the present disclosure.

Although the use of a point X0 213a is depicted in Figure 2d for a step 216, the value of X1 216a could also be calculated as X1 216a = [(Sd 103c * sn 102b) + (Ed 103a * sn 102b)]. A key exchange algorithm 207 can support either PKI keys based on elliptic curves or RSA algorithms, although support of elliptic curves may be preferred in some exemplary embodiments due to their shorter key lengths and lower computational processing requirements. In some exemplary embodiments, the value X1 216a can comprise two points of (i) (Sd 103c * sn 102b) and (ii) (Ed 103a * sn 102b) and key server 102 can send the two separate points for the value X1 216a to a server 101 in a message 217 in Figure 2a.

As depicted in Figure 2d, the output of a key exchange algorithm 207 in a step 216, such as an ECDH key exchange, can comprise a secret value X1 216a. In exemplary embodiments, secret value X1 216a can comprise a point on an elliptic curve, where the equation and values for the elliptic curve can be specified in parameters 104a. As contemplated herein, the secret value X1 216a (as well as X2 218a below) comprises both an X coordinate and a Y coordinate, in order to support subsequent ECC point addition operations. After conducting key exchange step 216, key server 102 can send the secret value X1 216a to server 101 in a message 217, as depicted in Figure 2a above.

Key exchange step 218 for a sever 101 depicted in Figure 2a can correspond to key exchange 218 in Figure 2d. Key exchange step 218 can comprise inputting or using the point X0 213a from a step 213 and the server static private key ss 101b into an ECDH key exchange

Petitioner Samsung Ex-1004, 1247

algorithm 207, which can comprise the same or equivalent key exchange algorithm 207 depicted and described in connection with key exchange step 216 described above. Other elements or algorithms within a key exchange step 218 can be equivalent to a key exchange step 216 above, including the use of shared parameters 104a. The output of a key exchange algorithm 207 in a step 218 can comprise a secret key or value X2 218a. Although the use of a point X0 213a is depicted in Figure 2d for a step 218, the value of X2 218a could also be calculated as X2 218a = [(Sd 103c * ss 101b) + (Ed 103a * ss 101b)]. In other words, the value of X2 218a could comprise two separate points, where the points can be (i) (Sd 103c * ss 101b) and (ii) (Ed 103a * ss 101b).

In exemplary embodiments, secret value X2 218a can comprise a point on an elliptic curve, where the equation and values for the elliptic curve can be specified in parameters 104a. Exemplary numeric values for using a key exchange algorithm 207 using a set of parameters 104a are depicted and described below. In exemplary embodiments, a server 101 can record the value X2 218a derived from a step 218 and also the value X1 216a received in a message 217 in a server database 101d. The time the values are stored in a server database 101d can be minimized in order to increase security, and, for example, the recording of the values can be deleted after server 101 both (i) receives a message from device 103 using key K1 206a and (ii) server 101 sends message 225 in Figure 2a above. Note that in some exemplary embodiments, the use of a server static PKI key pair comprising Ss 101a and ss 101b could be omitted, and for these embodiments then a step 218 can be omitted and the calculation and use of a value X2 218a can be omitted from Figure 2d, including omitting X2 218a from a key derivation step 219 below.

A key derivation step 219 for server 101 can (i) combine the output of key exchange steps 216 and 218 in order to calculate or derived the shared secret X3 207a and then (ii) perform a key derivation function step 208 on the derived or calculated shared secret X3 207a in order to determine or calculate shared symmetric ciphering key K1 206a. Note that shared secret symmetric ciphering key K1 206a can be also mutually derived by device 103, where device 103 uses the key exchange step 206 depicted and described in connection with Figure 2c above.

In exemplary embodiments, a server 101 can conduct the key derivation step 219 using (i) the value X1 216a received from key server 102 (where receipt of X1 216a by server 101 can be in a message 217 as shown in Figure 2a above), and (ii) the value or key X2 218a output from a key exchange step 218 for server 101 in the paragraph above. As contemplated herein, the values of X0 213a, X1 216a, X2 218a, and X3 207a may be described as "shared secrets" or "shared secret keys" or simply points. Although the values may not be used as a key directly with a symmetric ciphering algorithm in exemplary embodiments, these values and the output of an ECDH key exchange algorithm 207 can comprise a secret or a key or point.

Petitioner Samsung Ex-1004, 1248

Key derivation step 219 for server 101 can comprise two primary steps. A first step in key derivation 219 can comprise an EC point addition 220 on the value X1 216a and the value X2 218a. The result of the ECC point addition will be equal to the value X3 207a. Note that device 103 can also derive the same value for value X3 207a (in step 206 above in Figure 2c) without EC point addition 220 using a step 219. In other words, although (a) the related key exchange step 206 for device 103 may include a point addition for public keys, (b) the key exchange step 206 for device 103 will not use EC point addition for points derived from two separate private keys in two separate servers (e.g. X1 216a uses private key sn 102b and X2 218a uses private key ss 101b).

Exemplary calculations for an EC point addition 220 can comprise the calculations shown for point addition in the Wikipedia article for "Elliptic Curve Point Multiplication" dated May 15, 2018, which is herein incorporated by reference in its entirety. ECC point addition 220 can be equivalent to ECC point addition 213, but with different values. As depicted in Figure 2d, (a) the calculation of X3 207a by server 101 using an ECC point addition 220 over X1 216a and X2 218a will equal (b) the value for X3 207a calculated by device 103 using a key exchange algorithm 207 in a step 206 from Figure 2c above. A second step in key derivation step 219 as depicted in Figure 2d can comprise a key derivation function step 208 using (a) input from ECC point addition step 220 (e.g. value X3 207a output from step 220), where (b) the output of key derivation function step 208 can comprise key K1 206a and also an associated MAC key 206b. In exemplary embodiments, the X coordinate from shared secret X3 207a can be used with key derivation function 208.

By server 101 conducting a key derivation step 219 as depicted in Figure 2b (where key server 102 conducts the calculations for step 216 using the network static private key sn 102b), (i) sever 101 can calculate symmetric ciphering key K1 206a without recording or operating on the network static private key sn 102b. In this manner, the security of a system 100 or system 200 can be significantly enhanced, since the network static private key sn 102b does not need to be recorded or operated by server 101, which can communicate with a plurality of devices 103 over an IP network 107.

In other words, by server 101 (i) using the EC point addition over key X1 216a instead of (ii) conducting a key exchange 207 directly with sn 102b, then server 101 does not need to record or operate with the network static private key sn 102b, thereby increasing security and flexibility of a system 100 and system 200 and other systems herein. Also, since (i) key X1 216a can be the equivalent of an ECC public key as a point on an elliptic curve, and (ii) it is not computationally feasible to determine network static private key sn 102b from key X1 216a, then key X1 216a does not reveal meaningful information about network static private key sn 102b.

Many benefits can be achieved by server 101 conducting a key derivation step 219 using key X1 216a instead of recording and operating with network static private key sn 102b. As one

Petitioner Samsung Ex-1004, 1249

example, the corresponding network static public key Sn 102a could potentially be both (i) recorded in millions of distributed devices connecting to server 101 through many different physical locations and networks, and (ii) used for a decade or longer. Keeping network static private key sn 102b secure for this embodiment could be economically essential, since a compromise of network static private key sn 102b may (i) render the devices 103 insecure (or unable to authenticate network 105 using an ECDHE key exchange), and (ii) require the secure distribution or re-installation of a new, different network static public key Sn 102a in the devices, which may not be economically feasible due to the prior distribution of devices.

Exemplary data and numbers can be provided to demonstrate the calculations for (i) ECC point addition step 213, (ii) key exchange step 216, (iii) key exchange step 218, and (iv) key derivation step 219 using an ECC point addition 220. Parameters 104a can comprise the elliptic of "secp128r1" with key lengths of 128 bit long keys. Although the exemplary numbers for keys, points, and a named curve are provided, other values for keys, points, and named ECC curves could be utilized as well.

The network static private key sn 102b can comprise the exemplary following number, and can be recorded in key server 102:

221902394438765368991155818063875293908

The server static private key ss 101b can comprise the exemplary following number, and can be recorded by server 101:

246768250079261690512638148137618184294

Note that the corresponding public keys Ss 101a and Sn 102a are recorded by device 103 and used in step 206 above in Figure 2c. The device ephemeral public key Ed 103a can comprise the following exemplary values with X and Y numbers (or "coordinates") of:

X: 23935689655101766341272667257968262709 4

Y: 20957074553997392973951207096190580225 0

Note that the above device ephemeral public key Ed 103a corresponds to the device ephemeral private key ed 103b above from Figure 2c. The device static public key Sd 103c can comprise the following exemplary values with X and Y numbers (or "coordinates") of:

X: 20347342661252050681290227003882720119 6

Y: 64318327833120582836973711848343026891

Petitioner Samsung Ex-1004, 1250

ECC point addition step 213 by server 101 can combine the device static public key Sd 103c and the device ephemeral public key Ed 103a in order to output the following value for point X0 213a:

X: 591219228124585796004467517906621607 96

Y: 30493450923577826897895586717020091 7057

Key exchange step 216 for an ECDH algorithm key exchange 207 by key server 102 can input the point X0 213a in the paragraph above and the network static private key sn 102b (both with numbers above) in order to calculate a secret X1 216a. An exemplary number or value for secret X1 216a from the values above using parameters 104a can be:

X: 116816232651214939512035210922980929925

Y: 266578617588050771886642464870165918 12

Key exchange step 218 for an ECDH algorithm key exchange 207 by server 101 can input the point X0 213a from a step 213 above and the server static private key ss 101b (both with numbers above) in order to calculate a secret X2 218a. An exemplary number or value for key X2 218a from the values above using parameters 104a can be:

X: 283470377637256529257925581999478035172

Y: 117395441238388206677723127104680679540

An ECC point addition 219 for the above two derived points (or "keys") X1 216a (from point X0 213a and sn 102b) and X2 218a (from point X0 213a and ss 101b) will result in the following point that also equals X3 207a.

X: 192457465648897421085529769283600671459

Y: 126144984806909677418281309675999642 69

Note that the same numeric value for key X3 207a can also be derived by device 103 from a key exchange step 206 above using ECDH key exchange algorithm 207. For exemplary embodiments, although private key sn 102b and private key ss 101b can be recorded and operated by physically separated devices, device 103 can record and operate on the corresponding public keys Sn 102a and public key SS 101a (at the same physical location as device 103).

After an EC point addition 220, for a key derivation step 219 in Figure 2d, server 101 can input the shared secret key X3 207a, where key X3 207a was output from the EC point addition 220, into a key derivation function 208. The key derivation function 208 can comprise the same

Petitioner Samsung Ex-1004, 1251

key derivation function 208 used by a device 103 in a step 206 above. The output of a key derivation function 208 can comprise both (i) a symmetric ciphering key K1 206a and (ii) a MAC key 206b. MAC key 206b can be used with a symmetric ciphering algorithm in order to generate a MAC code 209m, such that the other party using the same key K1 206a and MAC key 206b can process the ciphertext and calculate the same MAC code 209m in order to verify message integrity.

In summary, X3 207a can be calculated by network 105 through any of the following combinations in addition to those described above, and other possibilities may exist as well:

1. X3 207a = [(Sd 103c * {(ss 101b + sn 102b) mod n}] + [(Ed 103a * {(ss 101b + sn 102b) mod n}]
2. X3 207a = [(Sd 103c + Ed 103a) * ss 101b] + [(Ed 103a + Sd 103c) * sn 102b]
3. X3 207a = [(Sd 103c + Ed 103a) * ss 101b] + [(Sd 103c * sn 102b) + (Ed 103a * sn 102b)]
4. X3 207a = [(Sd 103c * ss 101b) + (Ed 103a * ss 101b)] + [(Sd 103c + Ed 103a) * sn 102b]
5. X3 207a = [(Sd 103c * ss 101b) + (Ed 103a * ss 101b)] + [(Sd 103c * sn 102b) + (Ed 103a * sn 102b)]

In summary, the present disclosure contemplates that an ECDH key exchange by network 105 can include at least one EC point addition using the device ephemeral public key Ed 103a and the device static public key Sd 103c, in order to mutually derive point X3 207a with a device 103. For example, combinations above multiply the keys Ed 103a and Sd 103c individually by a scalar value via EC point multiplication before EC point addition operations. However, the calculation of point X3 207a uses at least one EC point addition of (i) a first value calculated with the device static public key Sd 103c and (ii) a second value calculated with the device ephemeral public key Ed 103a.

Figure 2e

Figure 2e is a flow chart illustrating exemplary steps for using a derived shared secret key to encrypt and decrypt data, in accordance with exemplary embodiments. A decryption step 221 can be performed by server 101, and an encryption step 223 can be performed by server 101. The use of a decryption step by server 101 and an encryption step 223 by server 101 was depicted and described in connection with Figure 2a above, and additional exemplary details are provided herein in Figure 2e. A decryption 221 step converts the ciphertext 209b received in a message 299 from Figure 2a into plaintext 209a. Decryption step 221 can utilize a symmetric ciphering algorithm 230 for decryption, which could comprise the same algorithm used in symmetric ciphering

Petitioner Samsung Ex-1004, 1252

algorithm 230 in a step 209 from Figure 2c, except the algorithm being used for decryption in step 221 instead of encryption.

Note that the same values are input into symmetric ciphering algorithm 230 for decryption in step 221 as symmetric encryption algorithm 230 above for encryption in step 209, such as symmetric ciphering key K1 206a and parameters 104f in order to convert ciphertext 209b back into plaintext 209a. Additional data input into symmetric ciphering algorithm 230 in step 221 or step 229 can comprise an initialization vector 209i and MAC code 206m. Initialization vector 209i and MAC code 206m which could be sent along with ciphertext 209b in message 209. MAC key 206b could be input into symmetric ciphering algorithm 230 as well, in order to calculate the MAC code 209m. The calculated MAC code 209m could be compared to the received MAC code 209m, where equal values to the calculated MAC code 209m and the received MAC code 209m can indicate that ciphertext 209b was received with message integrity (e.g. no bit errors or modifications to ciphertext 209b).

Server 103 can read and process plaintext 209a after a decryption step 221. The plaintext 209a as written by device 103 can comprise random number 203a generated by device 103 in a step 203 and device data 202b. Device data 202b could comprise data or information from device 103 for server 101 and/or network 105 for use with an application managing the communications between device 103 and server 101. Exemplary data for device data 202b was described above for a step 202 by device 103 in Figure 2a. In exemplary embodiments, the successful decryption of a ciphertext into a plaintext using decryption algorithm 230 in decryption step 221 can comprise an authentication of the device 103, since successful decryption by server 101 can only take place when the device 103 has access to device static private key sd 103d.

A successful decryption of ciphertext 209b can also include the use of MAC key 206b to calculate MAC code 209m, as described in the paragraph above. In other words, only the nodes could mutually derive key K1 206a and MAC key 206b in Figure 2c and Figure 2d by (i) device 103 recording private key sd 103d and (ii) server 101and/or network 105 securely receiving and recording the corresponding device static public key Sd 103c. Thus, data that is successfully encrypted by the device 103 and decrypted by the server 101 using key K1 206a would confirm the device 103 is authenticated with server 101.

Server 101 can conduct an encryption step 223, where the use for an encryption step 223 is depicted and described in connection with Figure 2a above. Plaintext 223a in a step 223 can comprise the (i) random number 203a from device 103 from message 299 decrypted from ciphertext 209a by decryption step 221 above, (ii) random number 211a from server 101 using step 211, and (iii) response 222a from a step 222 by server 101. The plaintext response 222a could comprise data for device 103 from server 101 in response to device data 202a received in message

Petitioner Samsung Ex-1004, 1253

299. Other or different exemplary data could be included as plaintext 223a in an encryption step 223. The keys for encryption step 223 can comprise symmetric key K1 206a and MAC key 206b from a key derivation step 208 in step 219 by server 101 in Figure 2d. The keys can be input into a symmetric ciphering algorithm 230 with parameters 104f.

Encryption step 209 in Figure 2c and decryption step 221 in Figure 2e and encryption step 223 in Figure 2e can use a common symmetric ciphering algorithm 230. Note that MAC key 206b can also be input into symmetric ciphering algorithm 230 along with an initialization vector 209i. The output from a symmetric ciphering algorithm 230 in an encryption step 223 can comprise ciphertext 223b and MAC code 209m. MAC code 209m and initialization vector 209i can be sent along with ciphertext 223b in a message 225 in Figure 2a. Note that an initialization vector 209i for encryption step 223 can comprise a different number than initialization vector 209i for encryption step 209 and decryption step 221.

As depicted and described in connection with Figure 2a, device 103 can also conduct a decryption step 221a, which can comprise the exemplary decryption step 221 for server 101 but with different plaintext and ciphertext data. A decryption step 221a by device 103 can use ciphertext 223b received in a message 225. Thus, when device 103 conducts decryption step 221a using symmetric encryption key K1 206a and/or MAC key 206b, the ciphertext and plaintext will comprise different values than those depicted for a decryption step 221 depicted in Figure 2d.

In exemplary embodiments, the successful decryption of a ciphertext 223b into a plaintext 223a by device 103 using decryption algorithm 230 in decryption step 221a can comprise an authentication of the server 101, since successful decryption by device 103 can only take place when the server 101 and/or network 105 has access to both (i) server static private key ss 101b and (ii) network static private key sn 102b. A successful decryption of ciphertext 223b by device 103 can also include the use of MAC key 206b to calculate MAC code 209m, as described for a decryption step 221 by server 101.

In other words, only the nodes could mutually derive key K1 206a and MAC key 206b in Figure 2c and Figure 2d by (i) server 101 and/or network 105 recording and using private keys ss 101b and sn 102b and (ii) device 103 recording the corresponding server static public key Ss 101a and network static public key Sn 102a. Thus, data that is successfully encrypted by the server 101 in step 223 and decrypted by the device 103 in a decryption step 221a using key K1 206a would confirm the server 101 is authenticated with device 103. Note that the use of two private keys comprising ss 101b and sn 102b provides device 103 two layers of authentication, where a decryption step 221a is only successful for device 103 by network 105 in Figure 1a recording and using private keys ss 101b and sn 102b.

Petitioner Samsung Ex-1004, 1254

For some exemplary embodiments, the use of a server static PKI keys of Ss 101a and ss 101b could be omitted as discussed above in Figure 2c and Figure 2d. For these embodiments, symmetric ciphering key 206a can still be mutually calculated by device 103 and server 101 using the other PKI keys for device 103 and network 105 described herein. As described in Figure 2c, device 103 could mutually calculate symmetric ciphering key K1 206a without using a server static public key Ss 101a. As described in Figure 2d, server 101 could mutually calculate symmetric ciphering key K1 206a without using server static private key ss 101b. For these embodiments, a device 103 can still authenticate network 105 by successful decryption of ciphertext 223b in a step 221a, since ciphertext 223b could only be processed by network 105 recording network static private key sn 102b.

In other words, instead of (a) device 103 obtaining two layers of authentication for network 105 from a successful decryption step 221a (from network 105 recording private keys ss 101b and sn 102b for the public keys Ss 101a and Sn 102a recorded by device 103), (b) device 103 could obtain a single layer of authentication for network 105 from a successful decryption step 221a (from network 105 recording private key sn 102b for the public key Sn 102a recorded by device 103).

Figure 2f

Figure 2f is an illustration of an exemplary server database, in accordance with exemplary embodiments. A server database 101d depicted and described above in connection with system 100 and system 200 can record data for server 101 to work with a plurality of devices 103 and at least one key server 102. A server database 101d could record at least one set of values, keys, and/or numbers for each device 103 in a plurality of devices 103. Other possibilities exist as well for the organization, tables, and recorded data within a server database 101d as depicted in Figure 2f without departing from the scope of the present disclosure. Data within server database 101d could be encrypted using a symmetric key.

Although system 100 and system 200 depict a server database 101d as operating or recorded within a server 101, a server database 101d could comprise a separate server within a network 105 and communicating with server 101 via a secure session 201a or a private network 107a. Further, a server database 101d, when operating or recorded in a separate server than server 101, then server database 101d could contain electrical components equivalent to a server 101 depicted and described in connection with Figure 1b.

Server database 101d can record values for devices 103 in multiple tables or a single table. Figure 2f depicts two exemplary tables comprising a PKI key table 101da and a device table 101db.

Petitioner Samsung Ex-1004, 1255

Exemplary data for a PKI key table 101da could comprise an obfuscated device identity 103o, a device identity 103i, a secure hash value for the device static public key Sd 103c comprising H(Sd) 250, a secure hash value for the network static public key Sn 102a used by a device 103, which could comprise H(Sn) 251, a device static public key Sd 103c, a device ephemeral public key Ed 103a, a server static public key Ss 101a, a server static private key ss 101b, a network static public key Sn 102a, an identity of key server 102 for device 103 (or key Ed 103a) which could comprise ID.key-server 102i. Note that the identity of key server 102 could identify which key server 102 records the corresponding network static private key 102b for the depicted network static public key 102a. For embodiments where server 101 derives a server ephemeral PKI key pair using a step 101x, then server 101 could also record in a PKI key table 101da in a server database 101d the values for a server static public key Es 101p and server static private key es 101s.

Exemplary data for a device tables 101db in a server database 101d can include a device identity 103i, random number 203a (depicted as "Random 1"), a selected set of cryptographic parameters 104a, a source IP address and port number 299p received in message 299, a random number 211a (depicted as "Random 2"), ECC point values or secrets for X0 213a, X1 216a, X2 218a, X3 207a, and derived symmetric ciphering key K1 216a. In exemplary embodiments, the values depicted in (a) the first row of both PKI key table 101da and device table 101db in a server database 101d could comprise (b) data recorded by a server 101 for a device 103 while conducting the series of steps for a step 224 and step 228 depicted and described in connection with Figure 2a above with a first device 103. The values depicted in the second row of both PKI key table 101da and device table 101db in a server database 101d could comprise data recorded by a server 101 while conducting the series of steps for a step 224 and step 228 depicted and described in connection with Figure 2a above with a second device 103, etc.

In exemplary embodiments for a server database 101d, a first device 103 could send server 101 a first value for device ephemeral public key Ed 103a, and the first value is depicted in Figure 2f as "103a-1" in table 101da. Since server 101 could communicate with a plurality of devices 103, the second row in the depicted table 101da could comprise data for the equivalent steps conducted with a second device 103, such as recording a second value for device ephemeral public key Ed 103a for the second device. The second value for device ephemeral public key Ed 103a with the second device 103 is depicted in Figure 2d as "103a-2".

Equivalent notations for other keys or values are applicable as well, such as table 101db in server database 101d recording a first secret X1 216a depicted as "216a-1" for a first device 103, and then recording a second secret X1 216a depicted as "216a-2" for a second device. Thus, as depicted a server database 101d can record and operate with a plurality of different values for a key, where each are utilized by a different device. Server 101 could keep track of TCP IP addresses

Petitioner Samsung Ex-1004, 1256

and port numbers associated with different devices 103 for conducting the steps in Figure 2a by the source IP:port number 299p, such as recording IP:port number 299p as a source IP:port number from a message 299 from a device 103, and then using the same IP:port number 299p as a destination IP:port number for a message 225 in Figure 2a

In some exemplary embodiments, a message 299 can include a secure hash value H(Sd) 250, as described for a message 299 in Figure 2a above. The receipt of a secure hash value H(Sd) 250 could be mapped to or associated with a key server 102 via a key server identity ID.key-server 102i, where the mapping of H(Sd) 250 to ID.key-server 102i could be recorded in a server database 101d. For these embodiments, server 101 could conduct a query in a step 212 of server database 101d using the received H(Sd) 250 in a message 299 in order to select a key server 102 with ID.key-server 102i in order to send the message 214 to key server 102. In this manner, server 101 can communicate with a plurality of different key servers 102, and the destination of a message 214 (or key server 102) can be selected by the value H(Sd) 250 received in a message 299. Other values could be used for a query step 212 from Figure 2a as well in order to determine a key server 102 to use for sending message 214.

In other words, for a plurality of different devices 103 communicating with a server 101, a first subset of devices 103 could record and use a first network static public key Sn 102a, and a second subset of devices 103 could record and use a second network static public key Sn 102a. By receiving a value or identifier of device 103 in message 299 (which could comprise ID.device 103i, or ID.device-obfuscated 103o or a hash value for a unique key 101v or 102v as depicted in Figure 1c), server 101 could use the identifying information or identifying value for device 103 depicted for a server database 101d to select or identify the correct key server 102 from a step 212 in order to (i) send a message 214 and (ii) receive the correct secret X1 216a for the device 103, where the device 103 uses a particular network static public key Sn 102a.

In an exemplary embodiment, server 101 could receive the plaintext Sn 102a in a message 299 and (i) match the network static public key Sn 102a in a server database 101d and (ii) use the value Sn 102a to select key server 102 with key server identity ID.key-server 102i. In another exemplary embodiment, an identity for key server 102 (such as ID.key-server 102i) could be selected or determined by server 101 using the selected set of cryptographic parameters 104a received in message 299a from Figure 2b and recorded in a database 101d. For these embodiments, a first selected set of cryptographic parameters 104a could be associated with a first key server 102 (and first ID.key-server 102i) and a second set of cryptographic parameters 104a could be associated with a second key server 102 (and second ID.key-server 102i). Other possibilities exist as well for a server database 101d to record data in order to select a key server 102 for sending message 210 with point X0 213a based on data received in message 299, without departing from

Petitioner Samsung Ex-1004, 1257

the scope of the present disclosure. As one example, an identity for key server 102 of ID.key-server 102i could be included in message 299 and the value for ID.key-server 102i could be recorded in a server database 101d by server 101.

In a server database 101d, although separate values are depicted for some data, such as values "102i-1" and "102i-2" for identities of key servers 102, some of the exemplary values can comprise identical strings or numbers. For example, data for two different devices 103 in a server database 101d could record the same name or value of "102i-2" for a single key server 102 to be associated with the two different devices 103. Likewise, two different devices 103 could share the same server static public key Ss 101a, and thus Ss 101a for two different devices 103 can be the same value of an exemplary "101a-1" for two different devices 103, which could also comprise shared server public keys 101z as depicted in Figure 1c above.

Similarly, two different devices 103 could use different server static public keys Ss 101a, and thus Ss 101a for two different devices could comprise a first value of "101a-3" for the third device or row in a PKI key table 101da and second, different value of "101a-4" for the fourth device or row. The use of different values for server static public keys Ss 101a for different devices 103 could comprise unique server public keys 101v as depicted in Figure 1c above. Note that server database 101d as depicted in Figure 2f also shows the use of shared network public keys 102z for Sn 102a (e.g. rows 1-3 in table 101da) and unique network public keys 102v for Sn 102a (e.g. rows 4-5 in table 101da). Further, a key server 102 could use more than one network static private key sn 102b, as shown by the use of two different network static public keys Sn 102a for two different devices in row 3 and row 4 of table 101da.

A server database 101d could also record additional data and values than those depicted in Figure 2d for some exemplary embodiments. For example, server database 101d could record timestamps for when messages are transmitted or received, such that stale or data older than a specified range could be purged. Server database 101d could also record plaintext data received from device 103 in a message 299, which could include device data 202b. Device data 202b could comprise transducer data or data input by a user of device 103.

Some data within a server database 101d could be recorded and operated on separately by server 101, such as server 101 not recording secrets such as X1 216a or X2 218a, etc. in a database 101d, but rather server 101 could record the values in volatile memory 101f of server 101. In exemplary embodiments, server database 101d could also operate in a distributed or "cloud" configurations such that multiple different servers 101 could query and record data in server database 101d, where data for server database 101d is recorded in multiple, physically separated servers.

Petitioner Samsung Ex-1004, 1258

As depicted for a server database 101d, some data could be optionally omitted. For devices 103 that do not use an obfuscated device identity ID.device-obfuscated 103o, then the fields in a server database could be left blank or empty. Likewise, if a hash value is not received in a message 299 then the corresponding hash values in a server database 101d could also be left blank or empty. For embodiments discussed in Figure 2c and Figure 2d where the use of server static PKI keys could be omitted from a step 224, then values or data for keys Ss 101a and ss 101b could be omitted, empty, or left blank within a server database 101d. As discussed above with a decryption step 221a in Figure 2e, device 103 could authenticate network 105 using network static public key Sn 102a for embodiments that omit server static PKI keys.

Figure 3

Figure 3 is an illustration of an exemplary set of cryptographic parameters, in accordance with exemplary embodiments. Cryptographic parameters 104 can specify sets of cryptographic parameters that are supported by server 101, device 103, and key server 102 in order to process message 299 and process a response message 225 from Figure 2a. Cryptographic parameters 104 can be recorded in nonvolatile memory in each of server 101, device 103, and key server 102. As depicted in Figure 1a, each of device 103, server 101, and key server 102 can record and operate with a set of cryptographic parameters 104. Cryptographic parameters 104 can record a collection of cryptographic algorithms or specifications such as a set identifier 104a, a key length 104b, an ECC curve name 104c, a hash algorithm 104d, symmetric ciphering key length 104e, settings for a symmetric ciphering algorithm 104f, and a random number length 104g.

As contemplated herein, when a selected set of cryptographic parameters such as using the words or description "parameters 104a" or "cryptographic parameters 104a" can specify a row of parameters or values in a set of cryptographic parameters 104, such that the collection of values in the row can be used with key pair generation functions 101x, ECDH key exchange 207, EC point addition operations 213 and 220, key derivation function 208, and other cryptographic operations and steps as contemplated herein. Set identifier 104a can be an identity for a row or set of values for cryptographic parameters 104. For example, set "A" can comprise cryptographic suite 1 as specified in section 3.2.3 of DPP specification version 1.0. Key length 104b can be the length of keys in bits for PKI keys used in system 100 and system 200. ECC Curve name 104c can be a name for an ECC curve used with PKI keys and key exchange algorithms in system 100, system 200, and other systems herein.

Hash algorithm 104d in cryptographic parameters 104 can be the name of a secure hash algorithm, such as the exemplary SHA-256 algorithm depicted, which may also be referred to as

Petitioner Samsung Ex-1004, 1259

"SHA-2". Hash algorithm 104d can also be used in a key derivation function (e.g. KDF 208 above in Figure 2c and Figure 2d) and also with digital signature step 233 from Figure 2b. Settings for a symmetric ciphering algorithm 104f can specify the identity or name of a symmetric ciphering algorithm 225 such as "AES", "AES-SIV", 3DES, Blowfish, etc. Random length 104g can specify the length in bits for random numbers or "nonces" generated by both device 103 and server 101, where the nonces can be used to prevent replay attacks and require messages transmitted and received to be unique. Other possibilities exist as well for data within cryptographic parameters 104, such as the specification of point compression, encoding rules such as distinguished encoding rules (DER), ASN or CSN syntax notation, padding rules, byte or bit orders such as big endian, little endian, etc.

Figure 4

Figure 4 is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device, a server, a first key server, and a second key server, in accordance with exemplary embodiments. System 400 can include a device 103, server 101, a first key server 102x a second key server 102. Device 103 was depicted and described in connection with Figure 1a, Figure 1c, and Figure 2a above. Server 101 was depicted and described in connection with Figure 1a, Figure 1b, Figure 2a, and other Figures above. The second key server 102 in Figure 4 can comprise a key server 102 as depicted in Figure 1a and Figure 2a above. The first key server 102x in Figure 4 can operate in a similar manner as key server 102, except with the difference being (i) the first key server 102x records and operates with a server static private key ss 101b in a key server database 102d' and (ii) the second key server 102 records and operates with a network static private key sn 102b in a key server database 102d. Server 101 can record and operate a server database 101d'. Individual steps and components used in system 400 in Figure 4 are also additionally depicted and described in above Figures 2b, 2c, and 2d, etc.

Before starting the steps and message flows depicted in Figure 2a, device 103 can securely receive and record a network public key Sn 102a and a server static public key Ss 101a, which was also depicted and described in connection with Figure 1a and Figure 1c. The corresponding private key for Ss 101a can be securely recorded in a first key server 102. The corresponding private key for public key Sn 102a can be securely recorded in the second key server 102 within network 105 as sn 102b. Steps and messages for system 400 in Figure 4 that are equivalent to steps and messages for system 200 in Figure 2a will be described in summary form herein, where differences between system 400 in Figure 4 and system 200 in Figure 2a will be highlighted and described with additional details. In exemplary embodiments, a server database 101d' can comprise a server

Petitioner Samsung Ex-1004, 1260

database 101d as depicted and described in connection with Figure 2f, but with the differences of (i) server database 101d' does not record server static private key ss 101b and (ii) server database 102d' records data for the first key server 102x such as, but not limited to, a first key server 102x identity in addition to private key ss 101b.

A server 101 can establish a secure connection 201d with the first key server 102x in addition to a secure connection 201a with the second key server 102 from Figure 2a above. Device 103 can conduct a step 201c as depicted and described in connection with Figure 2a to record data for conducting a step 224, such as, but not limited to, device static private key sd 101c, public keys Sn 102a and Ss 101a, etc. Device 103 can conduct a step 231 in order to process data for a message 299, where the individual steps in a step 231 for device 103 were depicted and described in connection with Figure 2a above. Device 103 can send server 101 a message 299. Before receiving message 299, server 101 could conduct a step 201b, where step 201b is also depicted and described above in connection with Figure 2a.

Server 101 can receive message 299 and conduct a series of steps to process the message. Server 101 could perform steps 210a, 210b, 211, and 212 as depicted and described above in connection with Figure 2a. A step 212 in Figure 4 can comprise server 101 conducting steps to select the second key server 102 which records and operates with network static private key sn 102b. Server 101 can then conduct a step 401 in Figure 4 in order to select the first key server 102x that records and operates with server static private key ss 101b. Server 101 can use the equivalent steps in (a) a step 401 to select the first key server 102x as (b) a step 212 to select the second key server 102. In summary, server 101 can use information or data received in message 299 in conjunction with a server database 101d' in order to select the first key server 102x in a step 401. Server 101 could use one of several possible methods for selecting a first key server 102x in a step 401, including a combination of the following embodiments.

A first embodiment for selecting a first key server 102x in a step 401 could comprise server 101 selecting the same first key server 102x for all keys Ed 103a received from all devices 103. For example, for this first method, server 101 could listen or operate on (i) a specific IP address and port number or (ii) with a specific DNS name or URL in step 201b, where the use of (i) or (ii) could be specified or associated with a server static public key Ss 101a. As mentioned above for a step 203 in step 231 in Figure 2a, device 103 can select the address of server 101 using the server address of server 101 recorded with Ss 102a (possibly from a table 103t in Figure 1c). Server 101 could determine that all messages 299 received using (i) or (ii) are associated with a specific first key server 102x.

A second embodiment of a step 401 for selecting key server 102x of received device ephemeral public key Ed 103a could comprise using an identity of key server 102 in a message 299

Petitioner Samsung Ex-1004, 1261

from device 103. The message 299 can optionally include an identity for key server 102x comprising ID.key-server2 102xi. For these embodiments, server 101 can select the first key server 102x using the ID.key-server2 102xi in message 299. A third embodiment for a step 401 of selecting key server 102x for received device ephemeral public key Ed 103a could comprise using an identity of device 103 in a message 299 comprising (i) ID.device 103i, (ii) ID.device-obfuscated 101o, or a secure hash value H(Sd) 250. A secure hash value over a unique key 102v or unique key 101v could be used as well to determine a first key server 102x, where a server database 101d' could include tables that map the unique secure hash value to a first key server 102x. As described above for a message 299, the message 299 can optionally include unique identifying information for device 103, and server 101 using database 101d' in Figure 2f above could include a table to map any of (i), (ii), and (iii) in this paragraph to a first key server 102x. For this third embodiment of a step 401, server 101 could conduct a query of server database 101d' to select the first key server 102x for device 103 using ID.device 103i.

A fourth embodiment for a step 401 to select a first key server 102x for received device ephemeral public key Ed 103a could comprise using the subset of cryptographic parameters 104a from a step 210a. Server 101 could record that a first subset of cryptographic parameters 104a are associated with a key server 102x, and a second subset of cryptographic parameters 104a are associated with a different key server 102x, etc. Other possibilities exist as well for server 101 to conduct a step 401 to select a first key server 102x using data in a message 299 without departing from the scope of the present disclosure. After conducting a step 401 to select a first key server 102x, server 101 can conduct an ECDH point addition operation 213 over device public keys Sd 103c and Ed 103a to calculate point X0 213a, as depicted and described in connection with Figure 2d above.

Server 101 can then send the first key server 102x a message 402 through the secure session 201a. Message 402 can include an identity for server 101 comprising ID.server 101i, the point X0 213a from a step 213 above, a device identity 103i, and the subset of cryptographic parameters 104a. Note that although the use of a step 213 and point X0 213a are depicted in Figure 4, a message 402 could include separately both the received device ephemeral public key Ed 103a and the device static public key Sd 103c, and in this embodiment point X0 213a could be omitted from a message 402. The use of a point X0 213a for a message 402 may be preferred for some embodiments, since it would provide additional security by not exposing the first key server 102x to the separate values for keys 103a and 103c.

For embodiments where device 103 records server static public key Ss 101a as shared keys 101z, then a device identity 103i in message 402 could be omitted, and the first key server 102x could use the same server static private key ss 101b for multiple communications representing

Petitioner Samsung Ex-1004, 1262

multiple devices 103 with server 101. In other words, the first key server 102x could use the server 101 identity 101i in a message 402 for some embodiments to determine the proper server static private key ss 101b for conducting subsequent steps, where server identity 101i is associated with shared keys 101z as depicted in Figure 1c. For embodiments depicted in Figure 1c where a device 103 records and operates with a unique key 101v, then a message 402 can preferably include device identity ID.device 103i in order for the first key server 102x to select the appropriate server static private key ss 101b for subsequent calculations. Server identity ID.server 103i also can be useful for communications between the first key server 102x and server 101 for a system 400, since either (i) server 101 may communicate with a plurality of different first key servers 102x, and/or (ii) the first key server 102x may communicate with a plurality of different servers 101.

Key server 102x can receive the message 402 via the secure session 201d and conduct a series of steps to process the message and respond. A first step conducted by first key server 102x can comprise a key validation step 210b, where the key validation step 210b conducted by first key server 102x can be equivalent or compatible with the key validation step 210b conducted by a server 101 as described in Figure 2a above. Key server 102x could use the set of cryptographic parameters 104a in message 402 in order to conduct a key validation step 210b. As mentioned above for a step 210b for server 101, a key validation step 210b for first key server 102x could be conducted to verify that point X0 213a is not reused, in order to deny replay attacks. First key server 102x could record all values X0 213a received in a key server database 102d' for an extended period of time, which could comprise years, and reuse of a value X0 213a could be rejected. Or, in other exemplary embodiments, the verification that value or point X0 213a is not reused could be omitted by key server 102x in a step 210b, and key server 102 could rely on server 101 to deny replay attacks.

The first key server 102x can then conduct a step 403 in order to select a server static private key ss 101b to use with the received point X0 213a from message 402. For a step 403, key server 102x could use device identity 103i received in message 402 to query a key server database 102d' to select key ss 101b associated with device identity 103i. A key server database 102d' could record a plurality of different server static private keys ss 101b for use with different devices 103. Or, a step 403 could use other data from a message 402 to select key ss 101b, such as parameters 104a or server identity 101i. In an exemplary embodiment, multiple different devices could record server static public key Ss 101a as shared keys 101z, where the shared keys 101z are associated with a particular set of cryptographic parameters 104a (such as a particular named ECC curve 104c). The first key server 102x could select a key ss 101b to use with the shared keys 101z based on the parameters 104a from message 402. Other possibilities exist as well for a first key server

Petitioner Samsung Ex-1004, 1263

102x to conduct a step 403 to select a server static private key ss 101b using data from a message 402 without departing from the scope of the present disclosure.

The first key server 102 can then conduct an ECDH key exchange step 218 using (i) the recorded server static private key ss 101b and (ii) the received point X0 213a from message 402. Exemplary details for an ECDH key exchange step 218 are depicted and described in connection with Figure 2d above. The output of an ECDH key exchange step 218 can comprise point X2 218a. For embodiments where message 402 includes either (i) an identity for device 103 such as ID.device 103i, or (ii) identifying information for ss 101b for a first key server 102x to utilize (such as a secure hash over Ss 101a), then key server 102x could use the identifying information in message 402 to select the server static private key ss 101b from a key server database 102d" for ECHD key exchange step 219.

The first key server 102x can then send server 101 a message 404, where the message 404 includes point X2 218a from step 218 above, as well as an identity for key server 102x comprising ID.key-server 102xi and cryptographic parameters 104a associated with point X2 218a. Message 404 can be transmitted through secure session 201d. If device identity 103i or other identifying information such as a secure hash value over a unique key 101v was included in message 402, then message 404 could also include device identity 103i or the other identifying information for a device 103. Or, both message 402 and message 404 can include a transaction identity or session identity, such that server 101 can associate the received value X2 218a with a received device ephemeral public key Ed 103a and associated device 103.

Server 101 can receive message 404 with point X2 218a and conduct a series of steps in order to derive a mutually shared and authenticated key exchange with device 103. Server 101 can send the second key server 102 a message 214, where a message 214 is depicted and described in connection with Figure 2a above. The second key server 102 can receive message 214 and conduct the steps 210b, 215, and 216 as depicted and described in connection with Figure 2a above. The second key server 102 can send server 101 a message 217, where message 217 includes point X1 216a. Point X1 216a could be calculated using (i) the network static private key sn 102b recorded in a key server database 102d, and (ii) the point X0 213a from message 214.

Server 101 can receive message 217 with point X1 216a and conduct a series of depicted steps 210c though 223, where the series of depicted steps 210c through 223 were depicted and described in connection with Figure 2a above. Note that a difference in steps 210c through 223 from Figure 4 and Figure 2a is that step 218 is omitted from Figure 4 and performed by the first key server 102x, while Figure 2a has server 101 conducting step 218. In other words, server 101 conducts step 218 with server static private key ss 101b for system 200 in Figure 2a, while the first

Petitioner Samsung Ex-1004, 1264

key server 102x conducts step 218 with the server static private key ss 101b for system 400 in Figure 4.

Using the system 400 in Figure 4, additional security can be achieved since server 101 does not record or operate with server static private key ss 101b. The benefits which result from key ss 101b being isolated from IP network 107, where devices 103 may connect with server 101 through insecure networks and server 101 may need to connect with the insecure networks connected to IP network 107. In other words, using the system 400 in Figure 4, server static private key ss 101b does not need to be recorded in a server 101 that connects with insecure networks. An equivalent benefit for the security of network static private key 102b is achieved from the use of key server 102 as well.

After step 223, server 101 can send device 101 a message 225, where the message 225 was depicted and described in connection with Figure 2a above. Device 103 can receive the message 225 and conduct a decryption step 221a. As described for a decryption step 221a in Figure 2e, server 101 and network 105 can be authenticated with device 103 by device 103 successfully conducting a decryption step 221a. Decryption step 221a can use key K1 206a from a key exchange step 206 for device 103 in a step 231. Device 103 can then conduct a step 226 to process the plaintext data from a decryption step 221a, where decryption step 221a decrypts the ciphertext in a message 225.

Figure 5a

Figure 5a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a mobile device, a g node b, and a key server, in accordance with exemplary embodiments. System 501 can include a mobile device 103', a "next generation node b" 101', and a key server 102. Mobile device 103' can comprise a smart phone, a device for the "Internet of Things" (IoT), a tablet with a modem, or possibly a fixed station device that connects with a 5G or 6G network. Mobile device 103' can operate similar to a device 103, with the additional functionality of connecting to a wireless network, where the wireless network can support 3GPP standards and can also comprise a wide area network such as a public land mobile network. A "next generation node b" 101' (depicted as gNb 101') can contain the equivalent electrical components as those depicted for a server 101 in Figure 1b, except gNb 101' can also operate as a base transceiver station to send and receive data wirelessly with mobile device 103'. The key server 102 could operate as part of an Authentication Server Function (AUSF) or equivalent functionality.

Petitioner Samsung Ex-1004, 1265

Note that the distributed nature of the ECDH key exchanges as depicted with system 200 in Figure 2a and system 400 in Figure 4 have benefits for the wireless WAN architecture in Figure 5a. One benefit is that sn 102b for a mobile device 103' does not need to be recorded or operated by a gNb 101'. A second potential benefit using the embodiment of system 400 in Figure 4, a gNb 101' does not need to record or operate with the server static private key ss 101b. These two benefits increase security, since a gNb 101' may be operated in insecure locations.

In exemplary embodiments, a mobile device 103', a gNb 101', and a key server 102 can conduct a step 224' or a step 405'. A system 500a can use the same or equivalent components, steps, and messages as a system 200, with some differences between the systems. Note that before the steps 224' depicted in Figure 5a, a mobile device 103' and a gNb 101' could conduct steps to establish communications between the nodes, such as recording parameters for RF communications by the mobile device 103' in a SIM card or eUICC. A mobile device 103' could also conduct steps to authenticate the network 105 operating a gNb 101', such as obtaining keys for accessing network 105 through a different gNb 101'. Step 224' can comprise primarily the step 224 as depicted and described in Figure 2a. Step 405' can comprise primarily the step 405 as depicted and described in Figure 4

For a step 224' or step 405', a mobile device 103' can send message 299 through RF link 501. Message 299 can include the device ephemeral public key Ed 103a and also an obfuscated identity ID.device-obfuscated for device 103', where the obfuscated identity can also comprise a temporary identity for device 103. A gNb 101' can use the obfuscated identity to track the device 103 from a potential plurality of devices 103 communicating over a wireless network.

The gNb 101' can forward the device identity and the point X0 213a from a step 224 to the key server 102. The key server 102 can look up a unique key 102v for device 103 for the network static private key sn 102b corresponding to the network static public key 102a recorded by the device 103. The key server 102 can calculate value X1 216a as depicted in Figure 2d, and send the gNb 101' the value X1 216a over a secure session. For a step 224', the gNb 101' can conduct an ECDH key exchange step 218 and calculate value X2 218a, using the received device ephemeral public key Ed 103a and the server static private key ss 101b. For a step 405', the gNb 101' can send the device ephemeral public key Ed 101a and identifying information for device 103 to a key server 102x, and key server 102x could conduct the ECHD key exchange step 218 and send the resulting value X2 218a to gNb 101'

The gNb 101' can calculate the value X3 207a via ECC point addition over X1 216a and X2 218a. The gNb 101' can calculate a symmetric ciphering key K1 206a using the value X3 207s and a KDF 208. KDF 208 could derive multiple different keys in addition to the symmetric ciphering key K1 206a. In exemplary embodiments for a system 500a, the device identity

Petitioner Samsung Ex-1004, 1266

ID.device 103i, which could comprise an IMSI value for 4G LTE networks, or a SUPI value for 5G and 6G networks, can be included in ciphertext 209b in message 299. The gNb 101' can use the derived key K1 206a to decrypt the ciphertext 209a and read the plaintext ID.device 103i. Using a system 500a, a mobile device can securely send a permanent or long-term subscription identity to a network in a single message comprising message 299. The security of a message 299 in a system 500a can be enhanced compared to conventional technology, because the static private keys for network 105 do not need to be recorded on gNb 101'.

In exemplary embodiments for a system 500a, either the static server public key Ss 101a or the static network public key Sn 102a recorded in mobile device 103' can comprise a unique key for mobile device 103. In other words, key Ss 101a can comprise a unique key 101v as depicted and described in connection with Figure 1c, and key Sn 102a can comprise a unique key 102v as depicted and described in connection with Figure 1c. Or, for other embodiments, keys Ss 101a and Sn 102a recorded by device 103 could comprise shared keys 101z and 102z. The mobile device 103' can then conduct a step 228 to receive a ciphertext 223b from gNb 101'. The successful decryption of ciphertext 223b by mobile device 103' can confirm that gNb 101' is authenticated as operating for network 105, where network 105 can record the private keys corresponding to the public keys recorded in mobile device 103'.

Figure 5b

Figure 5b is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a client, a server, and a key server, in accordance with exemplary embodiments. System 500b can include a client 103', a server comprising server 101, and a key server 102. In exemplary embodiments, client 103' can comprise a client using security steps as described in by transport layer security (TLS) sessions version 1.3 and also subsequent and related versions of IETF RFC standards. Client 103' can also comprise a client using security steps as described in datagram transport layer security (DTLS) RFC 6347 and subsequent versions that incorporate ECDH key exchanges. Although depicted in Figure 3b as a client 103', the client 103' could also comprise a device 103, where the device 103 can conduct the steps of a client 103' at the networking, transport, and application layer of the traditional Open Systems Interconnection (OSI) model.

Client 103' can comprise a computing device such as a personal computer, a laptop computer, or another server that records a network static public key Sn 102a and a server static public key Ss 101a. As discussed in Figure 2c and Figure 2d, a server static public key Ss 101a could optionally be omitted. Note that TLS version 1.3 and DTLS version 1.3 contemplate that the

Petitioner Samsung Ex-1004, 1267

client and a server can use ephemeral ECDH key exchanges (one on the client and one on the server) in order to establish a mutually derived secret shared key for a symmetric ciphering algorithm. The difference between (i) a client 103' (which can comprise a device 103 supporting TLS or DTLS standards) and (ii) a client for TLS or DTLS standards can be that client 103' can record a network static public key Sn 102a and server static public key Ss 101a.

As depicted in Figure 1c, the network static public keys could comprise either (i) a shared keys 102z or shared keys 101z across a plurality of different devices 103 (or clients 103'), or (ii) a unique key 102v or unique key 101v, where the network static public key Sn 102a and server static public key Ss 101a is a unique number or string or point specifically for client 103'. The public keys could be received by client 103' in a secure manner before a client 103' conducts a step 224 with server 101. In exemplary embodiments, Ss 101a and Sn 101a could be received in the form of a certificates from a prior TLS or DTLS session (or other secure manners) before client 103' begins the TLS or DTLS session depicted in Figure 5b. Or, network static public key Sn 102a could be recorded with a set of certificate authority certificates stored with installation of an operating system for device 103 operating a client 103'.

The use of a network static public key Sn 102a and/or server static public key Ss 101a by client 103' in a step 224 to conduct an ECDHE key exchange with server 101 can have many benefits. The standard handshake as currently proposed for TLS version 1.3 as of June 2018 assumes that a client 103' and a server 101 have no prior relationship. However, for many instances of communication between a client 103' and a server 101, the client 103' may have previously communicated with another server on a network 105 other than server 101. For example, with web browsing a web browser client such as a client 103' will often revisit the same web sites over time, such as a first web site for social networking, a second web site for a search engine, a third web site for news, etc.

A TLS or a DTLS session could utilize the fact that the same sites are often re-visited in order to increase security, using the depicted steps of 224 or step 405 and 228 for a client 103', server 101, and key server 102. Steps 224 in Figure 5b can comprise the set of steps 224 depicted and described in connection with Figure 2a, and steps 405 in Figure 5b can comprise the set of steps 405 depicted and described in connection with Figure 4, and steps 228 in Figure 5b can also comprise the set of steps 228 depicted and described in connection with Figure 2a.

Before conducting step 224 in Figure 5b, a client 103' could receive keys Sn 102a and/or Ss 101a from another server in network 105, such as a different web server providing functionality equivalent to server 101. The client 103' could record the network static public key Sn 102a and server static public key Ss 101a in a table 103t from Figure 1c. Each public key could also be recorded with a URL or domain name, such that the client 103' would use the public keys when

Petitioner Samsung Ex-1004, 1268

establishing a subsequent TLS or DTLS session with server 101, where server 101 uses the recorded URL or domain name. Further, server 101 could be configured so that any key Ed 103a received from IP network 107 on an IP address and/or port number used by server 101 would be forwarded to key server 102, where key server 102 could record and operate with the sn 102b corresponding to the public key for Sn 102a recorded by client 103'.

For a step 224, a client 103' can derive a device ephemeral public key Ed 103a and send server 101 a message 299. The message 203 can include the key Ed 103a and a set of cryptographic parameters 104a. In exemplary embodiments where client 103' implements TLS or DTLS, when message 299 can optionally omit a device identity ID.device 103i. Server 101 could operate in a manner such that (i) Ed 103a is forwarded to key server 102. Key server 102 can conduct an ECDHE key exchange as depicted for a step 224 in Figure 2a using a step 216 in order to calculate the secret value X1 216a. Key server 102 can send server 101 the value X1 216a.

Server 101 can use the value X1 216a, along with the derivation of a second secret X2 218a in order to calculate a symmetric ciphering key K1 206a, using the key derivation step 219 with ECC point addition 220 over X1 216a and X2 218a. Thus, by using the embodiment depicted in Figure 5b, a transport layer security session can have security increased, where (a) the ECDHE key exchange contemplated by TLS v1.3 can also add (b) the additional key exchange step 216 by a key server 102 and the additional key exchange step 218 by server 101. Note that the mutual derivation of symmetric ciphering key K1 206a by client 103' and server 101 can comprise a mutual authentication of server 101 and device 103, since server 101 and device 103 can only derive the key K1 206a if (a) server 101 operates in a network 105 that also records and operates with key sn 102b and (b) device 103 records and operates with device static private key sd 103d.

Message 225 could comprise a "Server Hello" according to TLS v1.3 in the document "draft-ietf-tls-tls13-28". The ciphertext in the Server Hello can be ciphertext 223b as depicted in Figure 2a, where the ciphertext 223b is encrypted with the mutually derived symmetric ciphering key K1 206a. Note that a step 224 for Figure 5b increases security for a TLS session, since an active attacker could operate as a "man in the middle" between a real client or "true client" and the server 101, where the "man in the middle" could derive its own key Ed 103a and substitute that for the real key Ed 103a from the real client or "true client".

Without use of public keys Sn 102a and/or Ss 101a in a key exchange step 206 by device 103, a "man in the middle" (deriving and substituting a key Ed 103a) could (a) mutually derive a symmetric ciphering key similar to K1 206a with server 101 and then (b) receive and decrypt the ciphertext 223b. However, the use of Sn 102a and/or Ss 101a for deriving encryption key K1 206a can stop a "man in the middle" attack since a "man in the middle" cannot derive key K1 206a without also recording the sn 102b and/or ss 101b, which can remain secret and not available to the

Petitioner Samsung Ex-1004, 1269

"man in the middle". In this manner, the communications for a TLS session or DTLS session can remain secured using a step 224 or step 405 and step 228, while recording and using (i) sn 102b with network 105 and ss 101b with a server 101 or key server 102x and (ii) Ss 101a and Sn 102a with client 103'.

Note that the technology depicted in Figure 5b supports mutual authentication without requiring either client 103' or server 101 to send static public keys in the form of certificates. In other words, communication can be secured without transfer of static public keys between the nodes in data 106 over an insecure network such as the public Internet 107. Over approximately the next decade, quantum computers may feasibly break common certificates and algorithms used with TLS and DLTS as of 2018, and thus the technology depicted in Figure 5b can be used to secure a system 500b (and other systems herein) against attacks using quantum computers. Other possibilities exist for the use of a step 224 or step 405 and a step 228 between a client 103' and server 101 without departing from the scope of the present disclosure.

Figure 5c

Figure 5c is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by an initiator, a responder, and a key server, in accordance with exemplary embodiments. System 500c can include an initiator 103', a responder 101' and a key server 102. Initiator 103' can comprise a computing device 103, with the specific additional functionality of an initiator according to the DPP Specification Versions 1.0 from the WiFi Alliance. Responder 101' can comprise a device with (i) electrical components similar or equivalent to a server 101 depicted in Figure 1b above, and (ii) the specific additional functionality of a responder according to the DPP Specification Version 1.0 of the WiFi Alliance. For example, initiator 103' and responder 101' can communicate via a WiFi network on a wireless LAN 502 between the two devices, which could also comprise a specific implementation of an IP network 107. Responder 101' can operate in a networked configuration to communicate with key server 102 via a private network 107a or a secure session 201a as depicted in Figure 2a. In some embodiments, responder 101' can communicate with key server 102 via an IP network 107, where the use of secure session 201a can create a private network 107a between responder 101' and key server 102.

An initiator 103', responder 101' and a key server 102 can conduct a step 224, where a step 224 is depicted and described in connection with Figure 2a above. Or, an initiator 103', responder 101' and a key server 102 can conduct a step 405, where a step 405 is depicted and described in connection with Figure 4 above. An initiator 103', responder 101' and a key server 102 can then

Petitioner Samsung Ex-1004, 1270

conduct a step 228, where a step 228 is depicted and described in connection with Figure 2a above. Note that additional steps in addition to those depicted in Figure 5c can be conducted by an initiator 103' and a responder 101', such as responder 101' deriving PKI keys in a step 101x from Figure 1a and also conducting additional ECDH key exchanges in order to derive a symmetric ciphering key K1 206a, which can comprise key ke in the DPP standard. In other words, initiator 103' and responder 101' could perform additional ciphering that that depicted for a step 224 in Figure 2a, but for exemplary embodiments such as that depicted in Figure 5c the initiator 103' and responder 101' could conduct at least the steps depicted in order to mutually derive a symmetric ciphering key K1 206a and use the key to create a ciphertext 223b by responder 101' and decrypt the ciphertext 223b by initiator 101'.

As depicted in Figure 5c, several PKI keys within a DPP specification version 1.0 can have corresponding keys for a step 224 or step 405 and a step 228. As depicted in Figure 5c, the device ephemeral public key Ed 103a can comprise the initiator protocol public key Pi 503a. The device ephemeral private key ed 103b can comprise the initiator protocol private key pi 503b. The device static public key Sd 103c can comprise the initiator bootstrap public key Bi 504a. The device static private key sd 103d can comprise the initiator bootstrap private key bi 504b. The server static public key Ss 101a can comprise the responder protocol public key Pr 505a. The server static private key ss 101b can comprise the responder protocol private key pr 505b. The network static public key Sn 102a can comprise the responder bootstrap public key 506a. The network static private key sn 102b can comprise the responder bootstrap private key 506b. As described below, other steps from Figure 2a can be equivalent to those depicted in Figure 5c. One difference for the key described above and Figure 2a is that (i) server static public key Ss 101a (e.g. key Pr 505a) could be not recorded in device 103 before sending message 299, and (ii) device 103 receives key Ss 101a with message 225 in a step 228. Also, the key K1 206a as contemplated herein can comprise the key ke 511 from the DPP specification version 1.0

Before sending a message 299, initiator 103' can conduct a series of steps to process data and keys for the message 299. Initiator 103' can conduct a key derivation step 103x to generate keys Ed 103a and ed 103b using parameters 104a. Initiator 103' can conduct an ECDH key exchange step 507 using (i) public key Br 506a (e.g. Sn 102a) and private key ed 103b to derive a symmetric ciphering key. Note the symmetric ciphering key from a step 507 can be different than key K1 206a, which can equal ke 511. Initiator 103' can then conduct a step 508, which can comprise the steps for an initiator to generate a "DPP authentication request" message in DPP specification version 1.0. Initiator 103' can process a ciphertext 209b' using the symmetric ciphering key from a step 507. Initiator 103' can then send responder 101' a message 299, which could comprise a "DPP authentication request" message per the DPP specification version 1.0. The

Petitioner Samsung Ex-1004, 1271

message 299 can include a hash value over the initiator bootstrap public key Bi 504a, which is depicted as H(Sd) 250, the derived initiator protocol public key Ed 101a, and the ciphertext 209b'. Ciphertext 209b' can use the symmetric ciphering key from step 507.

Responder 101' can receive message 299 and conduct steps to process the message. Responder 101' can conduct a step 509, which can comprise the steps for a responder to process a "DPP authentication request" message from an initiator in the DPP specification version 1.0. Responder 101' can take the additional step 213 to conduct an ECC point addition operation over the initiator bootstrap public key Bi 504a (e.g. Sd 103c) and the received initiator protocol public key Pi 503a (e.g. Ed 101a). Responder 101' can send key server the resulting point X0 213a in a message 214, along with identifying information for device 103 such as the secure hash H(Sd) 250, which could comprise a secure hash over the initiator bootstrap public key Bi 504a.

Key server 102 can receive message 214 and take steps to process the message. Key server 102 can conduct a step 215 in order to select from database 102d the responder bootstrap private key br 506b (e.g. sn 102b) using the identifying information for device 103 from message 214 (e.g. H(Sd) 250). Key server 102 can conduct an ECDH key exchange step 216 using point X0 213a and the selected responder bootstrap private key br 506b (e.g. sn 102b) in order to output point X1 216a. Key server 102 can then send responder 101' a message 217 with the point X1 216a. Other identifying information for device 103 could be included in a message 217 in order for responder 101' to determine which device 103 message 217 is associated with, such as H(Sd) 250.

Responder 101' can then conduct an ECDH key exchange step 218 using point X0 213a and responder protocol private key pr 504b (e.g. ss 101b), where point X0 213a comprises an ECC point addition 213 for Bi 504a (e.g. Sd 103c) and Pi 503a (e.g. Ed 103a). The output of step 218a can be point X2 218a. Responder 101' can then conduct a key derivation step 219 as depicted in Figure 2d above in order to calculate a shared secret X3 207a. Shared secret X3 207a can comprise the output from an ECC point addition step 220 over shared secret X1 216a from message 217 and shared secret X2 218a from a step 218. A key derivation step 219 can use a KDF 208 to convert shared secret X3 207a into a symmetric ciphering key K1 206a. Note that for the exemplary embodiment depicted in Figure 5c, symmetric ciphering key K1 206a can comprise key ke 511 for an initiator and a responder according to DPP specification version 1.0.

Responder 101' can then conduct an encryption step 223 in order to generate a ciphertext 223b. Note that ciphertext 223b can be encrypted using the derived symmetric ciphering key K1 201a which can also comprise key ke 511. The plaintext for an encryption step 223 in Figure 5c can comprise the plaintext data for a responder to use for a responder authentication value "r-auth" according to DPP specification version 1.0. Although not depicted in Figure 4, a responder 101'

Petitioner Samsung Ex-1004, 1272

could take other steps to perform additional encryption of ciphertext 223b, such as using the symmetric ciphering key from a step 507.

Responder 101' can then send initiator 103' a message 225', where message 225' could comprise a "DPP authentication response" message according to DPP specification version 1.0. Message 225' can be similar to message 225 depicted in Figure 2a above, but with the addition of message 225' including the responder protocol public key Pr 505a, which is depicted in Figure 5c as key Ss 101a. Message 225' can also include ciphertext 223b, which is encrypted using the mutually derived key K1 206a which can comprise key ke 511 according to the DPP specification version 1.0. Initiator 103' can then conduct a step 510 in order to process the message 225' which can comprise a "DPP authentication response" message according to DPP specification version 1.0. For a step 510, initiator 101' can take the steps specified in DPP specification version 1.0 in order to process a "DPP authentication response" message. In this manner and as depicted in Figure 5c, an initiator 103' and a responder 101' can be mutually authenticated. Initiator 103' and responder 101' can then conduct additional steps contemplated in DPP specification version 1.0.

A benefit for the use of the steps and messages for an initiator 103', a responder 101', and a key server 102 in Figure 5c is that the responder bootstrap private key br 506b (e.g. sn 102b) can remain securely recorded in a network 105 and does not need to be recorded and operated by responder 101. In this manner, the responder bootstrap public key Br 302a can be freely shared with multiple different initiators 103', including recording the key Br 302a in a plurality of initiators 103' in the form of a shared key 102z as depicted in Figure 1c.

The use of a shared key 102z with multiple different initiators 103' (while keeping sn 102b or key br 506b securely recorded in a key server 102) simplifies the distribution of key Br 506a to multiple different initiators 103'. Authentication of responder 101' is secured, because the responder 101' can only derive key K1 206a (as key ke 511) when the responder 101' also records the initiator bootstrap public key Bi 504a (Sd 103c). In other words, the authentication of responder 101' by initiator 103' depends on the security of the method responder 101' used to obtain the initiator bootstrap public key Bi 504a, which could be via an "out of band" transfer of the initiator bootstrap public key Bi 504a before initiator 103' sends a message 299'.

For exemplary embodiments, the initiators 103' could have a key Br 506a recorded during manufacturing or distribution of the computing device operating initiator 103'. In other words, a device manufacturer upon device distribution for an initiator 103' may not know which responder 101' may communicate with initiator 103' during a subsequent DPP session. However, a manufacturer of device with initiator 103' could record a plurality of different keys Br 506a for different networks 105 (similar to different keys Sn 102a in for a table 103t Figure 1c), and in this manner initiator 103' can have a higher probability of successfully using a pre-recorded key Br

Petitioner Samsung Ex-1004, 1273

506a (or key Sn 102a) in order to conduct a DPP session without requiring a separate or different additional step of acquiring the key Br 506a "out of band". Thus, the use of the embodiment for an initiator 103' and a responder 101' can simplify the use and deployment of DPP sessions, while simultaneously increasing the securing of the session, since the responder bootstrap private key br 506b (in the form of sn 102b) can remain securely recorded within a network 105 on a key server 102.

Figure 5d

Figure 5d is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a radio-frequency identification (RFID) tag, a RFID reader, and a key server, in accordance with exemplary embodiments. System 500d can include an RFID tag 103', a RFID reader 101' and a key server 102. RFID tag 103' (or "tag 103') could comprise a tag supporting International Standards Organization (ISO) standards such as ISO 15693, and other possibilities exist as well. Tag 103' could comprise an NFC tag such as a tag compatible with the NFC Forum standards for type 1 through type 5 tags (and subsequent or related standards). The NFC technology could also be NFC-A, NFC-B, or NFC-V, or subsequent standards.

RFIC reader 101' could comprise a compatible reader of RFID tag 103' and could support ISO standards 11784 and 11785 and other possibilities exist as well. Key server 102 can comprise a key server 102 as depicted and described above in connection with Figure 1a, Figure 1c, Figure 2a, and other figures herein. RFID tag 103' can comprise a device 103 with the specific functionality to communicate using near field communications (NFC) such as supporting a NFC session setup 512. RFID reader 101' can comprise the functionality and component of a server 101 depicted and described in connection with Figure 1a, Figure 1b, Figure 2a, and other figures herein. For the exemplary embodiment depicted in Figure 5d, both RFID tag 103' and RFID reader 101' (or "reader 101") can include a radio for NFC communications.

In exemplary embodiments, tag 103' and reader 101' can conduct an NFC session setup 512. The air interface for NFC session setup 512 could comprise a session using ISO/IEC 18092 / ECMA-340, "Near Field Communication Interface and Protocol-1" and ISO/IEC 21481 / ECMA-352, "Near Field Communication Interface and Protocol-2 (NFCIP-2)". The air interface for NFC session setup 512 could also comprise subsequent or related versions to these standards. Other possibilities exist as well for the protocol used in an NFC session setup 512 without departing from the scope of the present disclosure. NFC session setup 512 can comprise the establishment of the physical and data-link layers of communications between tag 103' and reader 101'.

Petitioner Samsung Ex-1004, 1274

Data between tag 103' and reader 101' could be transferred using NFC Data Exchange Format (NDEF) and support a Multipurpose Internet Mail Extensions (MIME) typed object or file transferred between tag 103' and reader 101'. In other words, subsequent messages and files transferred using NFC radios could utilize NDEF and MIME objects, although other possibilities exist as well for the NFC standards utilized without departing from the scope of the present disclosure. NFC session setup 512 could also implement additional air-interface security, such as ECMA-409 2nd edition - NFC-SEC-02: NFC-SEC and related standards, where the air interface is encrypted using AES and Diffie-Hellman key exchanges. In exemplary embodiments, NFC session setup 303 utilizes standard ECMA-352 in order to select ECMA-340, ISO/IEC 14443 and ISO/IEC 15693 as a communication mode between tag 103' and reader 101'

As depicted in Figure 5, tag 103', reader 101' and key server 102 could conduct a step 224 or a step 405, where a step 224 is depicted in Figure 2a and a step 405 is depicted in Figure 4. For a step 224 or step 405, a reader 101' can send a probe or query request as part of NFC session setup 512. A tag 103' can send message 299 to reader 101' through NFC session 512. Message 299 can include the device ephemeral public key Ed 103a and also an obfuscated identity ID.device-obfuscated 103o for tag 103', where the obfuscated identity can also comprise a temporary identity for device 103. Or, message 299 for a step 224 or step 405 in Figure 4 could comprise a hash value over a public key recorded by tag 101' (such as H(Sd) 250 or H(Ss) 252) and the hash value can be used by reader 101' in order to identify tag 103'. Or, message 299 could include an identity for tag 103' comprising an ID.device 103i. Other possibilities exist as well for a tag 103' to send plaintext identification information or omit plaintext identification information in a message 299 without departing from the scope of the present disclosure.

The reader 101' can use identifying information from message 299 to query a database 101d for additional PKI key information for tag 103' in a database 101d, such as (i) reading a key ss 101b for a key Ss 101a recorded by the tag 103', as well as (ii) selecting a key server 102 recording a key sn 102b for a key Sn 102a recorded by the tag 103', and also (iii) selecting a key Sd 103c for tag 103'. Or, in some exemplary embodiments such with a message 299c described in Figure 2b, message 299c from tag 103' could include a device static public key Sd 103c as plaintext in a message 299.

Reader 101' can conduct an EC point addition operation 213 over key Sd 103c and key Ed 103a in order to calculate a point X0 213a. Reader 101' can send key server 102 the point X0 213a in a message 214 along with identifying information for tag 103' and/or reader 101'. Key server 102 can use a step 215 to select a key sn 102b for tag 103' using the identifying information from a message 214. Key server 102 can conduct an ECDH key exchange step 216 with point X0 213a

Petitioner Samsung Ex-1004, 1275

and key sn 102b in order to calculate a point X1 216a. Key server 102 can send reader 101' the point X1 216a in a message 217 for a step 224 or step 405.

For a step 405, the reader 101' can send the point X0 213a and identifying information for device 103 to a key server 102x, and key server 102x could conduct the ECHD key exchange step 218 and send the resulting value X2 218a, as depicted and described in connection with a step 405 in Figure 4. For a step 224, reader 101' can conduct an ECDH key exchange step 218 with point X0 213a and key ss 101b in order to calculate a point X2 218a. Reader 101' can conduct a key derivation step 219 in order to calculate a shared secret X3 207b. Step 219 can use an ECC point addition operation 220 with point X1 216a and X2 218a to calculate X3 207b. Reader 101' can input X3 207b into a KDF 208 in order to calculate or process a resulting symmetric ciphering key K1 206a.

The tag 103' can mutually derive the same symmetric ciphering key K1 206a using a key exchange step 206 as depicted in Figure 2c. The tag 103' could conduct a step 206 before sending message 299 and after setup of NFC session 512. The tag 103' could encrypt device data 202b in a ciphertext 209b in message 299 using the key K1 206a, as depicted in Figure 2c. The reader 101' can decrypt the ciphertext 209b in message 299 using the key K1 206a derived by reader 101 in a step 219. The reader 101' can use a decryption step 221 in order to read plaintext from ciphertext 209b. Successful reading of plaintext from a decryption step 221 (plus optionally a correct MAC code 209m received in message 299) can confirm to reader 101' that tag 103' is authenticated and the decrypted plaintext from message 299 can be trusted as from a device 103' that records a device static private key sd 103d. The reader 101' can conduct an encryption step 223 using key K1 206a to create a ciphertext 223b.

The reader 101' can conduct a step 228 with the tag 103' to send the ciphertext 223b to tag 103' in a message 225. The tag 103' can then conduct a step 228 to receive a ciphertext 223b from reader 101'. The successful decryption of ciphertext 223b using key K1 206a from step 206 by tag 103' can confirm that reader 101' is authenticated as operating for network 105, where network 105 can record the private keys corresponding to the public keys recorded in tag 103'.

Conclusion

Various exemplary embodiments have been described above. Those skilled in the art will understand, however, that changes and modifications may be made to those examples without departing from the scope of the claims.

Petitioner Samsung Ex-1004, 1276

CLAIMS

1. A method for supporting an authenticated elliptic curve Diffie Hellman (ECDH) key exchange, the method performed by a first server, the method comprising:

a) recording a set of cryptographic parameters, a server static private key, and a device static public key;

b) receiving a first message comprising a device ephemeral public key and a ciphertext;

c) sending, to a second server, a second message, wherein the second message includes at least one of (i) a first point and (ii) the device ephemeral public key, and wherein the first point comprises a first elliptic curve (EC) point addition operation of the device ephemeral public key and the device static public key;

e) receiving, from the second server, a second point, wherein the second point is derived using at least a network private key and the second message;

f) conducting the ECDH key exchange using the set of cryptographic parameters, the device ephemeral public key, the device static public key, and the server static private key in order to derive at least a third point;

g) conducting a second EC point addition operation with the second point and at least the third point in order to derive a fourth point;

h) deriving a symmetric ciphering key using the derived fourth point; and

i) decrypting the ciphertext using the derived symmetric ciphering key.

2. The method of claim 1, wherein the first point comprises a value X0, the second point comprises a value X1, the third point comprises a value X2, and the fourth point comprises a value X3.

Petitioner Samsung Ex-1004, 1277

3. The method of claim 1 or 2, further comprising: conducting, by the first server, the ECDH key exchange in order to derive the third point and a fifth point, wherein the third point comprises an first EC multiplication of the device ephemeral public key by the server static private key, wherein the fifth point comprises a second EC multiplication of the device static public key and the server static public key, and wherein the second EC point addition operation uses the third point and the fifth point in order to derive the fourth point.

4. The method of any one of claims 1 to 3, wherein the second message includes the device ephemeral public key and the device static public key.

5. The method of claim 4, further comprising: conducting, by the second server, a second ECDH key exchange using the device ephemeral public key, the device static public key, and the network static private key in order to derive the second point.

6. The method of any one of claims 1 to 3, wherein the second message includes the first point.

7. The method of claim 6, further comprising: conducting, by the second server, a second ECDH key exchange using the first point and the network static private key in order to derive the second point.

8. The method of any one of claims 1 to 7, further comprising: selecting, by the first server, the device static public key using an identity for a device, and wherein the first message includes the identity for the device.

9. The method of claim 8, wherein the identity comprises a hash value over a unique key for the device, wherein the first server stores the hash value in a database, and wherein the unique key comprises one of a server static public key and a network static public key stored in the device.

10. The method of any one of claims 1 to 8, further comprising selecting, by the second server, the network private key using the second message.

Petitioner Samsung Ex-1004, 1278

11. The method of any one of claims 1 to 10, further comprising: conducting, by the device, a third ECDH key exchange using the device ephemeral private key, the device static private key, the server static public key and the network static public key in order to mutually derive the fourth point.

12. A method for supporting an authenticated elliptic curve Diffie Hellman (ECDH) key exchange, the method performed by a set of servers, the method comprising:

a) recording, by a first server and a second server, a set of cryptographic parameters;

b) recording, by the first server, a server static private key and a device static public key;

c) recording, by the second server, a network static private key;

d) receiving, by the first server, a message comprising (i) a device ephemeral public key and (ii) a ciphertext;

e) conducting, by the first server, a first elliptic curve (EC) point addition operation on (i) the received device ephemeral public key and (ii) the recorded device static public key in order to derive a first point;

f) sending the first point from the first server to the second server;

g) conducting, by the second server, a first ECDH key exchange using (i) the set of cryptographic parameters, (ii) the first point, and (iii) the network static private key in order to derive a second point;

h) sending the derived second point from the second server to the first server;

i) conducting, by the first server, a second ECDH key exchange using (i) the set of cryptographic parameters, (ii) the first point, and (iii) the server static private key in order to derive a third point;

j) conducting, by the first server, a second EC point addition operation on the received second point and the derived third point in order to derive a fourth point;

Petitioner Samsung Ex-1004, 1279

k) deriving, by the first server, a symmetric ciphering key using the derived fourth point; and

l) decrypting, by the first server, the received ciphertext using the symmetric ciphering key.

13. The method of claim 12, further comprising: selecting, by the first server, the device static public key using the message, wherein the message includes identification information for a device.

14. The method of 13, wherein the identification information includes at least one of (i) a device identity, (ii) an obfuscated device identity, and (iii) a secure hash value over a public key, wherein the public key comprises a unique key for the device.

15. The method of any one of claims 12 to 14, further comprising:

receiving, in step d), the set of cryptographic parameters; and

sending, in step f), the set of cryptographic parameters.

16. The method of any one of claims 12 to 15, wherein the first point comprises a value X0, the second point comprises a value X1, the third point comprises a value X2, and the fourth point comprises a value X3.

17. The method of any one of claims 14 to 16, further comprising: conducting, by a device, a third ECDH key exchange using the device ephemeral private key, the device static private key, the server static public key and the network static public key in order to mutually derive the fourth point.

18. The method of any one of claims 12 to 17, wherein step i) occurs before step f).

19. A method for supporting an authenticated elliptic curve Diffie Hellman (ECDH) key exchange, the method performed by a set of servers, the method comprising:

a) recording, by a first server and a second server, a set of cryptographic parameters;

Petitioner Samsung Ex-1004, 1280

b) recording, by the first server, a server static private key and a device static public key;

c) recording, by the second server, a network static private key;

d) receiving, by the first server, a message comprising (i) a device ephemeral public key and (ii) a ciphertext;

e) sending, from the first server to the second server, the device ephemeral public key and the device static public key;

f) conducting, by the second server, a first ECDH key exchange using (i) the set of cryptographic parameters, (ii) the device ephemeral public key and the device static public key, and (iii) the network static private key in order to derive a first point;

g) conducting, by the first server, a second ECDH key exchange using (i) the set of cryptographic parameters, (ii) the device ephemeral public key and the device static public key, and (iii) the server static private key in order to derive a second point;

h) sending, by the second server to the first server, the first point;

i) conducting, by the first server, an elliptic curve (EC) point addition operation on the first point and the derived second point in order to derive a third point;

j) deriving, by the first server, a symmetric ciphering key using the derived third point; and

k) decrypting, by the first server, the received ciphertext using the symmetric ciphering key.

20. The method of claim 19, wherein the first point comprises a value X1, the second point comprises a value X2, and the third point comprises a value X3.

21. The method of claim 19 or 20, wherein step g) occurs before step h).

Petitioner Samsung Ex-1004, 1281

22. The method of any one of claims 19 to 21, wherein the first server uses the message to select the device static public key, and wherein the second server uses the device static public key to select the network static private key.

23. The method of any one of claims 19 to 22, further comprising: conducting, by the device, a third ECDH key exchange using the device ephemeral private key, the device static private key, the server static public key and the network static public key in order to mutually derive the third point.

24. A method for conducting a secure session between a device and a server, the method performed by the device, the method comprising:

a) recording, by the device, a device private key and a certificate authority public key;

b) receiving, by the device, a first certificate for a network public key with a set of cryptographic parameters and a second certificate for a server public key with the set of cryptographic parameters;

c) verifying, by the device, the first and second certificates with at least the certificate authority public key;

b) deriving, by the device, (i) a device ephemeral public key and a corresponding device ephemeral private key using at least the set of cryptographic parameters, and (ii) a random number;

d) conducting, by the device, an elliptic curve Diffie Hellman (ECDH) key exchange using at least the server public key, the network public key, the device ephemeral private key, and the device private key in order to derive a shared secret, wherein the ECDH key exchange uses at least one elliptic curve point addition operation with the server public key and the network public key;

h) deriving, by the device, a symmetric ciphering key using at least (i) a key derivation function and (ii) the shared secret;

Petitioner Samsung Ex-1004, 1282

i) encrypting, by the device, a ciphertext using the symmetric ciphering key, wherein the ciphertext includes the random number; and

j) sending, by the device to the server, the device ephemeral public key and the ciphertext.

25. The method of claim 24, wherein the device conducts the ECDH key exchange by conducting (i) an elliptic curve EC point addition operation with the server public key and the network public key to derive a point, (ii) an addition of the device private key and the device ephemeral private key to derive a value, and (iii) a EC point multiplication of the point and the value.

26. The method of claim 24, wherein the device conducts the ECDH key exchange by conducting at least an EC point multiplication of (i) the server public key and the device private key to derive a first point, (ii) the network public key and the device private key to derive a second point, (iii) the server public key and the device ephemeral private key to derive a third point, and (iv) the network public key and the device ephemeral private key to derive a fourth point.

27. The method of claim 26, wherein the device generates the shared secret by an EC point addition of the first, the second, the third, and the fourth points.

Petitioner Samsung Ex-1004, 1283

28. The method of any one of claims 24 to 25, further comprising: conducting, by the server, a second ECDH key exchange with (i) the received device ephemeral public key, (ii) a stored device public key, and (iii) a server private key in order to derive a first point; conducting, by a key server, a third ECDH key exchange with (i) the device ephemeral public key and the device public key, and (ii) a network private key in order to derive a second point; sending, by the key server and to the server, the second point; and conducting, by the server, an EC point addition operation with first point and the second point in order to derive the shared secret.

29. The method of claim 28, wherein the first point comprises a value X1, the second point comprises a value X2, and wherein the shared secret comprises a value X3.

30. The method of any one of claims 24 to 29, further comprising before step b), (i) storing a network public key table with a plurality of network public keys, plurality of server public keys, and associated domain names, and (ii) selecting the network public key and the server public key from the network public key table with a domain name for the server.

31. The method of any one of claims 24 to 30, wherein the network public key comprises a unique key for the device, and wherein the device sends the server a secure hash value for the network public key with the device ephemeral public key.

32. The method of any one of claims 24 to 30, wherein the network public key comprises a shared key, wherein the server selects a key server using a domain name for the server in the set of cryptographic parameters, wherein the key server stores a network private key for the network public key.

33. A method for a key server to support an authenticated key exchange, the method performed by a key server, the method comprising:

a) recording in a database (i) a plurality of network private keys with parameters and (ii) a plurality of secure hash values for corresponding network public keys;

Petitioner Samsung Ex-1004, 1284

b) establishing a secure session with a server, wherein the server communicates with a plurality of devices over a public Internet;

c) receiving, from the server, identification information for a device, a secure hash value, and at least one of (i) a first point and (ii) a device ephemeral public key and a device static public key, wherein the first point comprises an elliptic curve (EC) point addition over the device ephemeral public key and the device static public key;

d) selecting, from the database, a network private key and the parameters for the network private key using the identification information;

e) conducting a first elliptic curve Diffie Hellman (ECDH) key exchange in order to derive a second point, using the network private key, the parameters, and one of (i) the first point and (ii) the device ephemeral public key and the device static public key; and

f) sending, to the server, the derived second point, wherein the derived second point is used with a second ECDH key exchange and a server static private key in order to derive a shared secret and a symmetric ciphering key for the device.

34. The method of claim 33, wherein the identification information comprises a secure hash value over a network public key corresponding to the network private key.

35. The method of claim 34, wherein the network public key comprises a unique key for the device.

36. The method of claim 34, wherein the network public key comprises a shared key, and wherein a plurality of devices store the network static public key.

37. The method of any one of claims 33 to 36, wherein the first point comprises a value X0, the second point comprises a value X1, and the shared secret comprises a value X3.

Petitioner Samsung Ex-1004, 1285

38. The method of any one of claims 33 to 37, further comprising: conducting, by the server, a second EC point addition operation with the derived second point and a third point from the second ECDH key exchange in order to derive the shared secret.

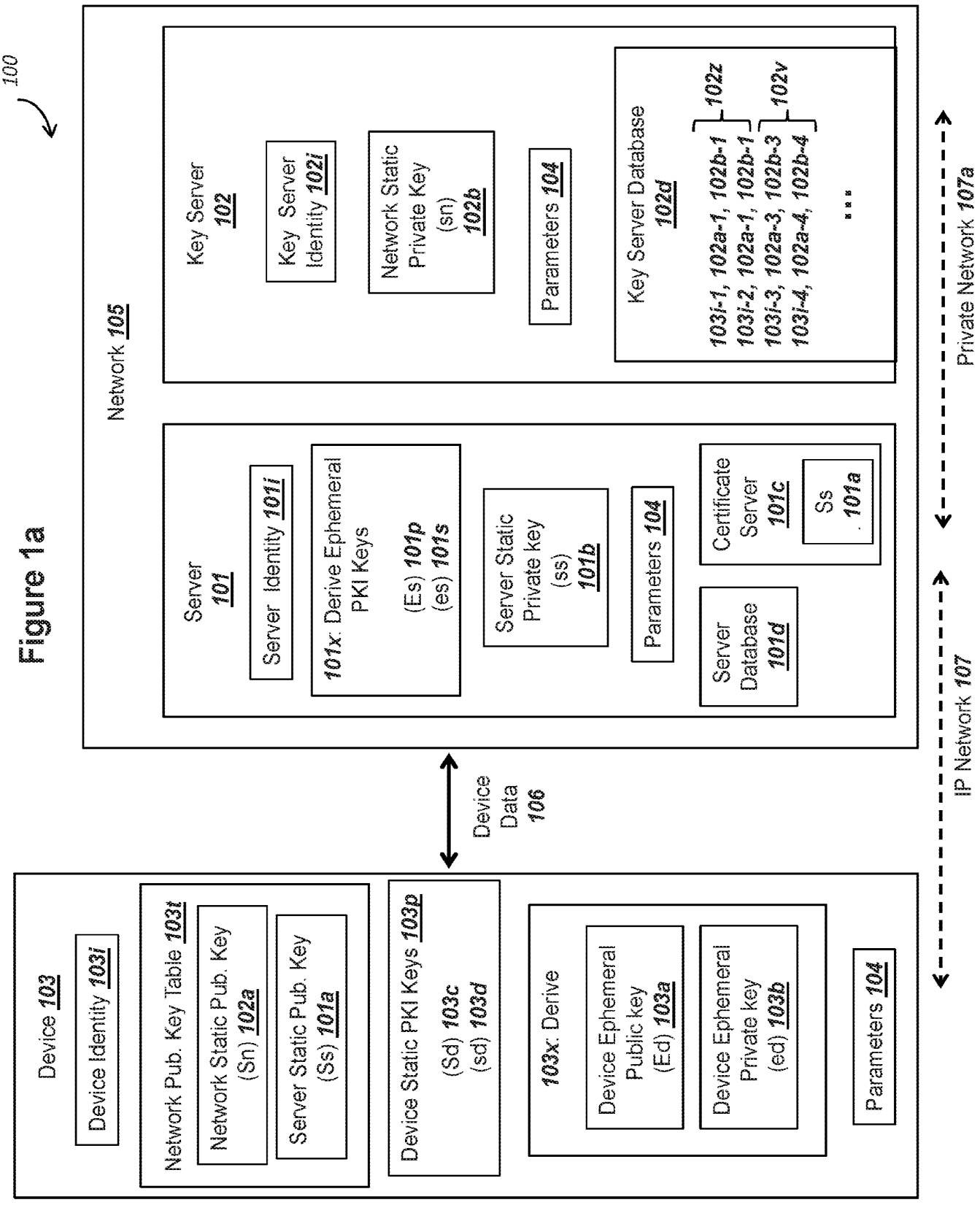
39. The method of any one of claims 33 to 38, further comprising: conducting, by the device, a third ECDH key exchange in order to derive the shared secret, and wherein the third ECDH key exchange uses at least (i) a stored server public key for the server private key, (ii) a stored network public key for the network private key, (iii) a device ephemeral private key for the device ephemeral public key, and (iv) a device static private key for the device static public key.

40. The method of any one of claims 33 to 39, further comprising: establishing, by the device and the server a secure session over the public Internet using at least the symmetric ciphering key.

41. The method of any one of claims 33 to 40, wherein the network private key comprises a unique key for the device, and wherein the device receives the network public key in a certificate.

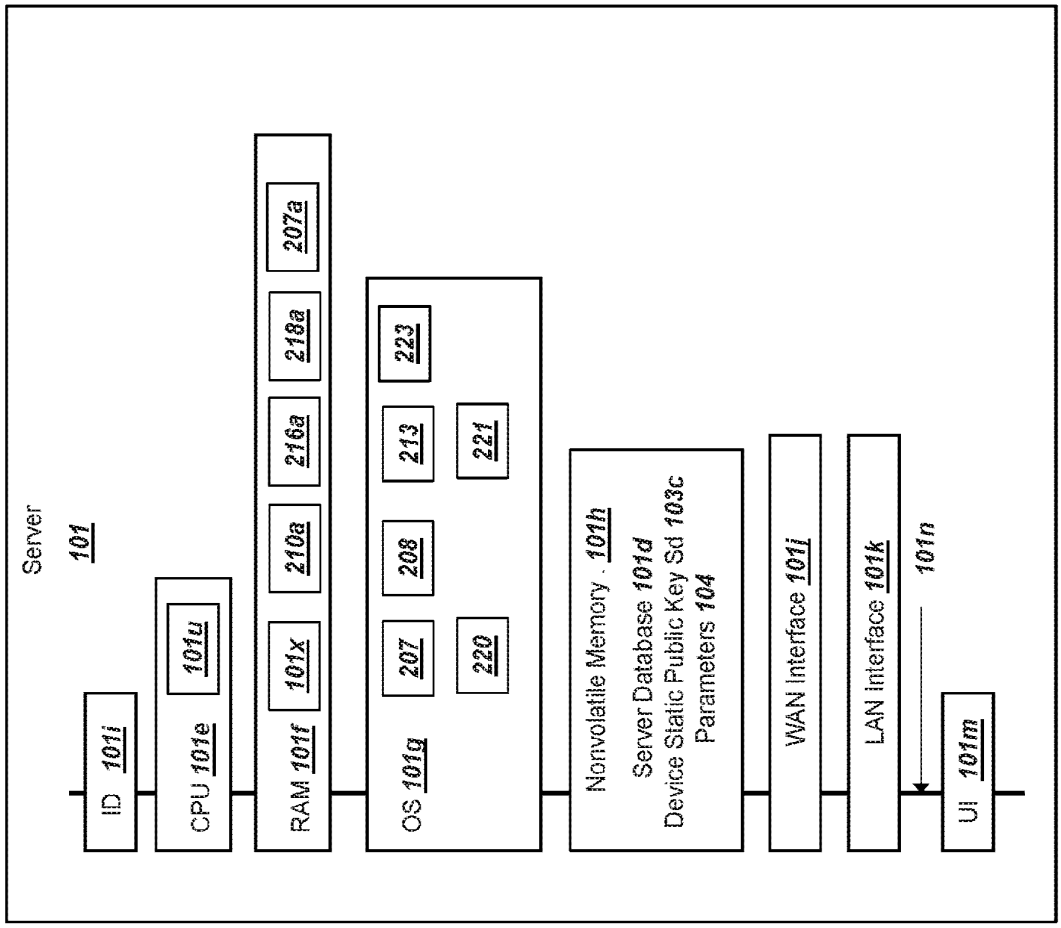
42. The method of any one of claims 33 to 41, further comprising: after step d) and before step e), verifying, by the key server, the received device ephemeral public key with the selected parameters.

Petitioner Samsung Ex-1004, 1286



Petitioner Samsung Ex-1004, 1287

Figure 1b



Petitioner Samsung Ex-1004, 1288

Figure 1c

Petitioner Samsung Ex-1004, 1289

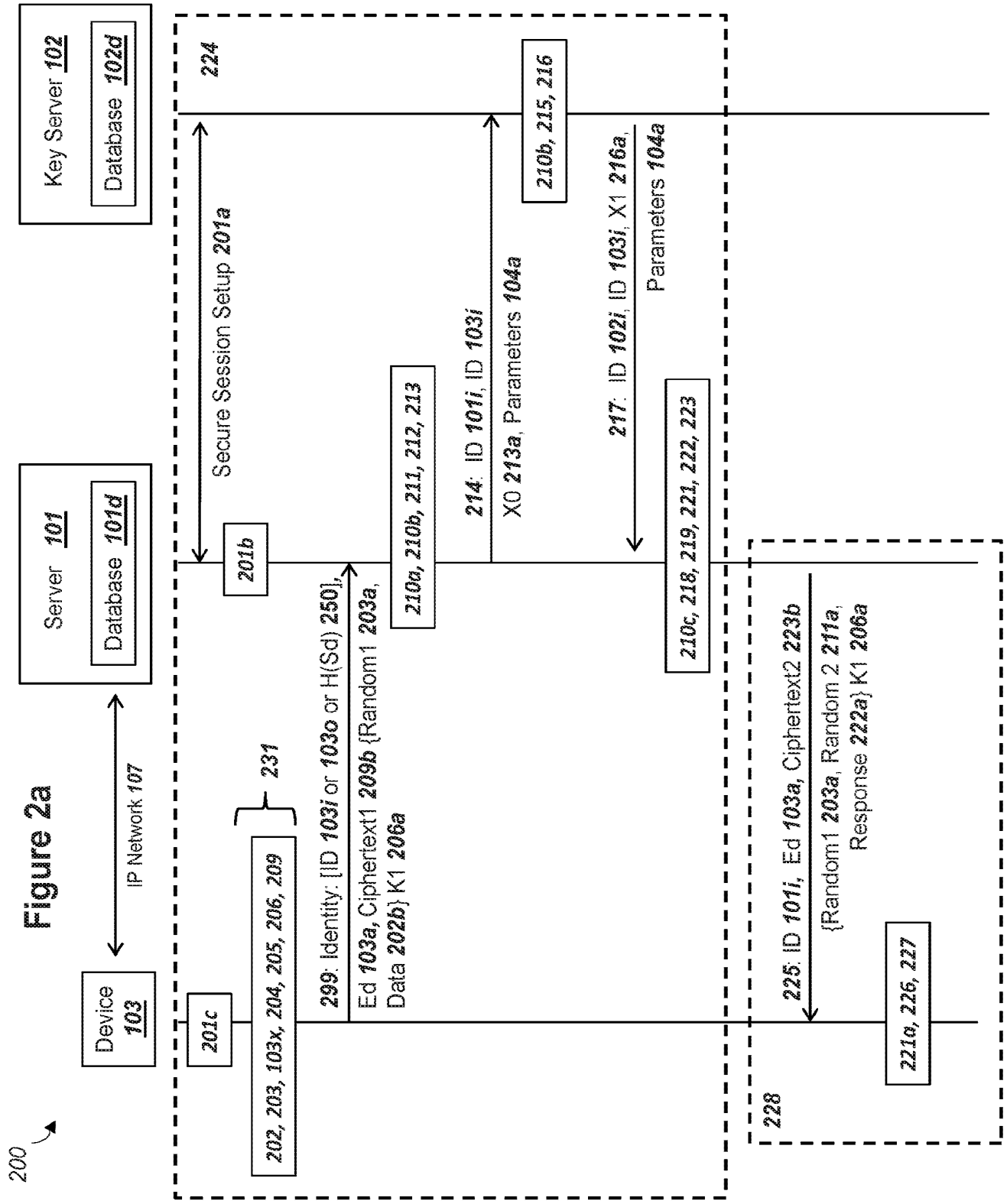
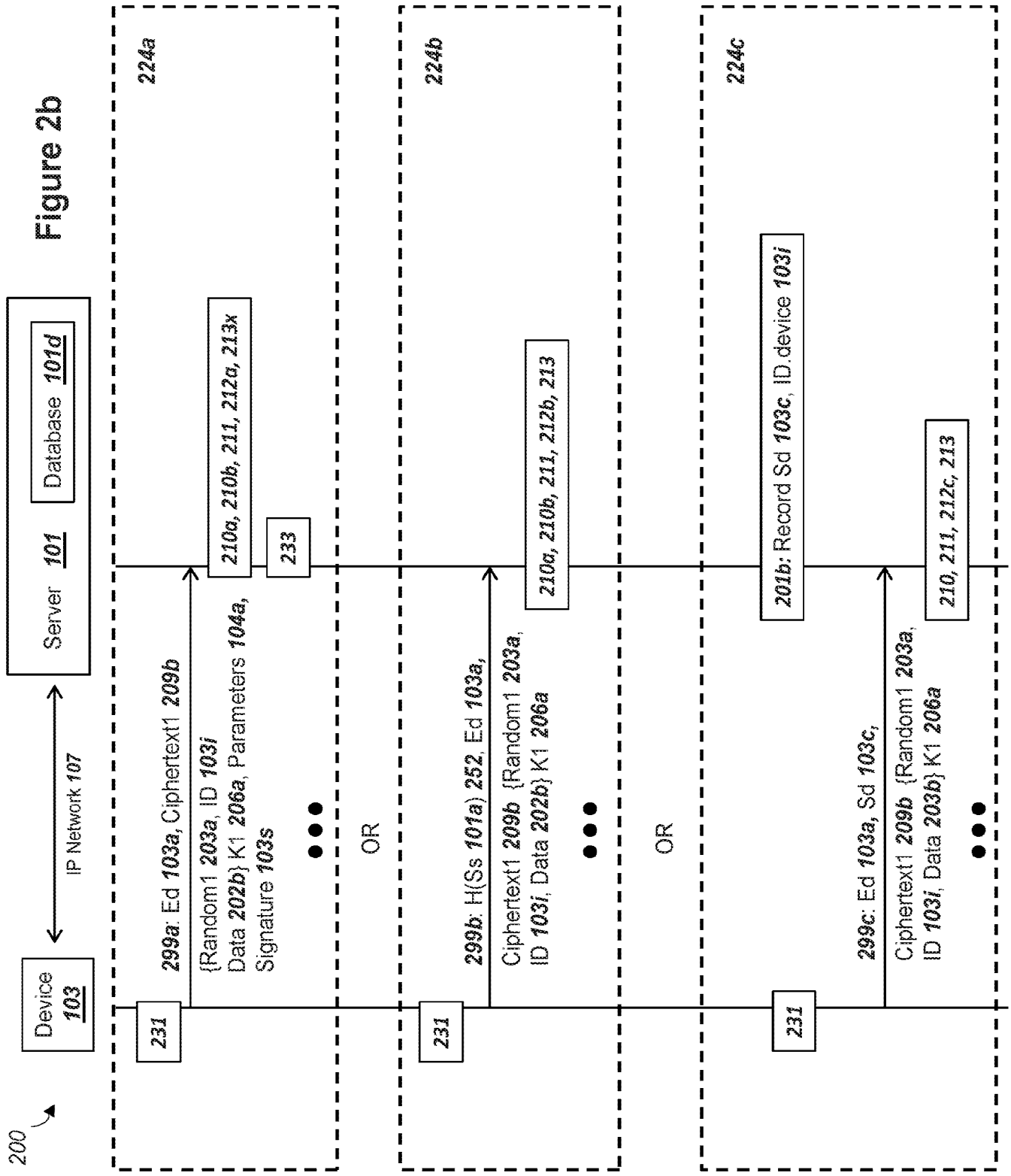


Figure 2a

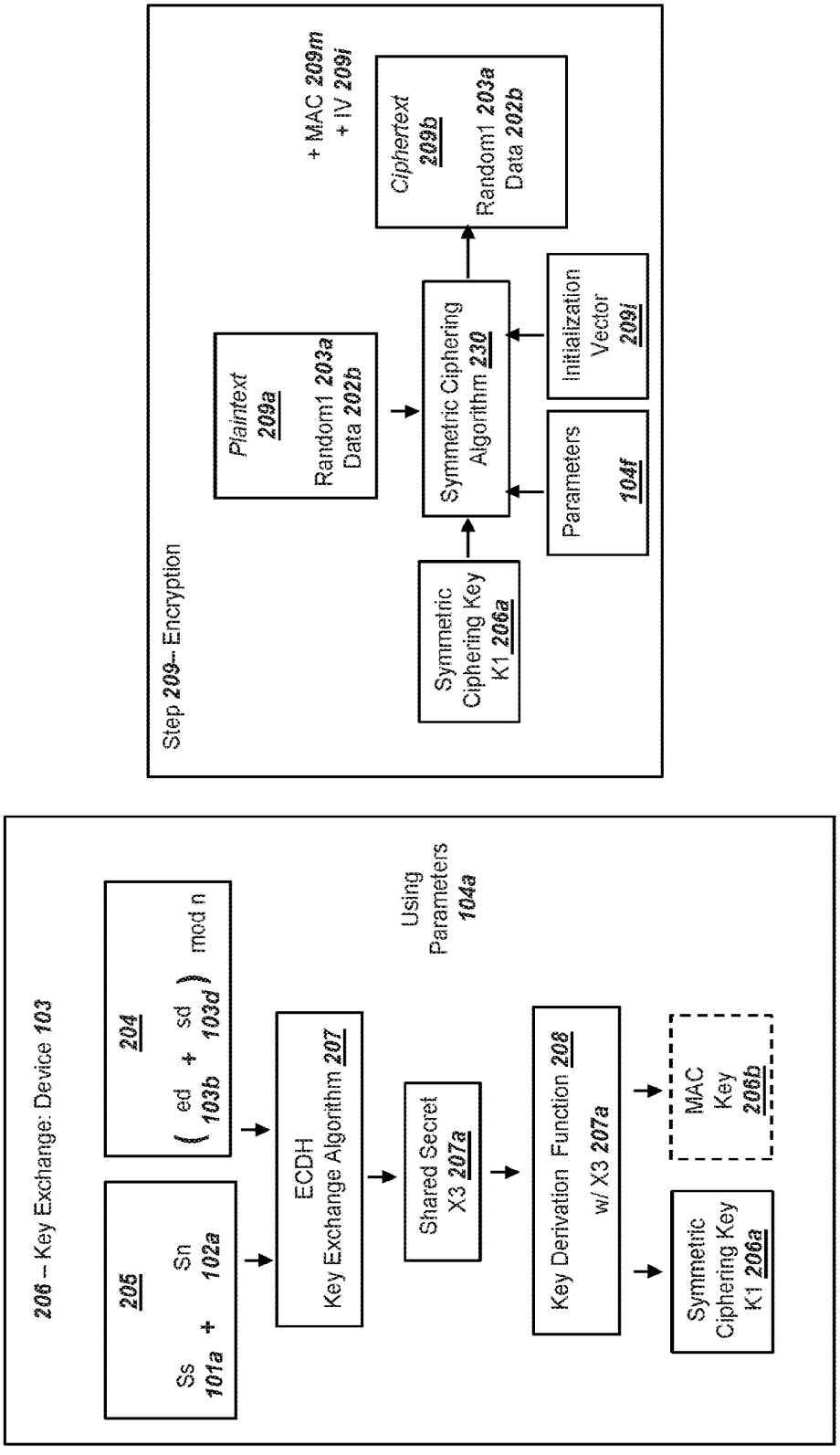
Petitioner Samsung Ex-1004, 1290

Figure 2b



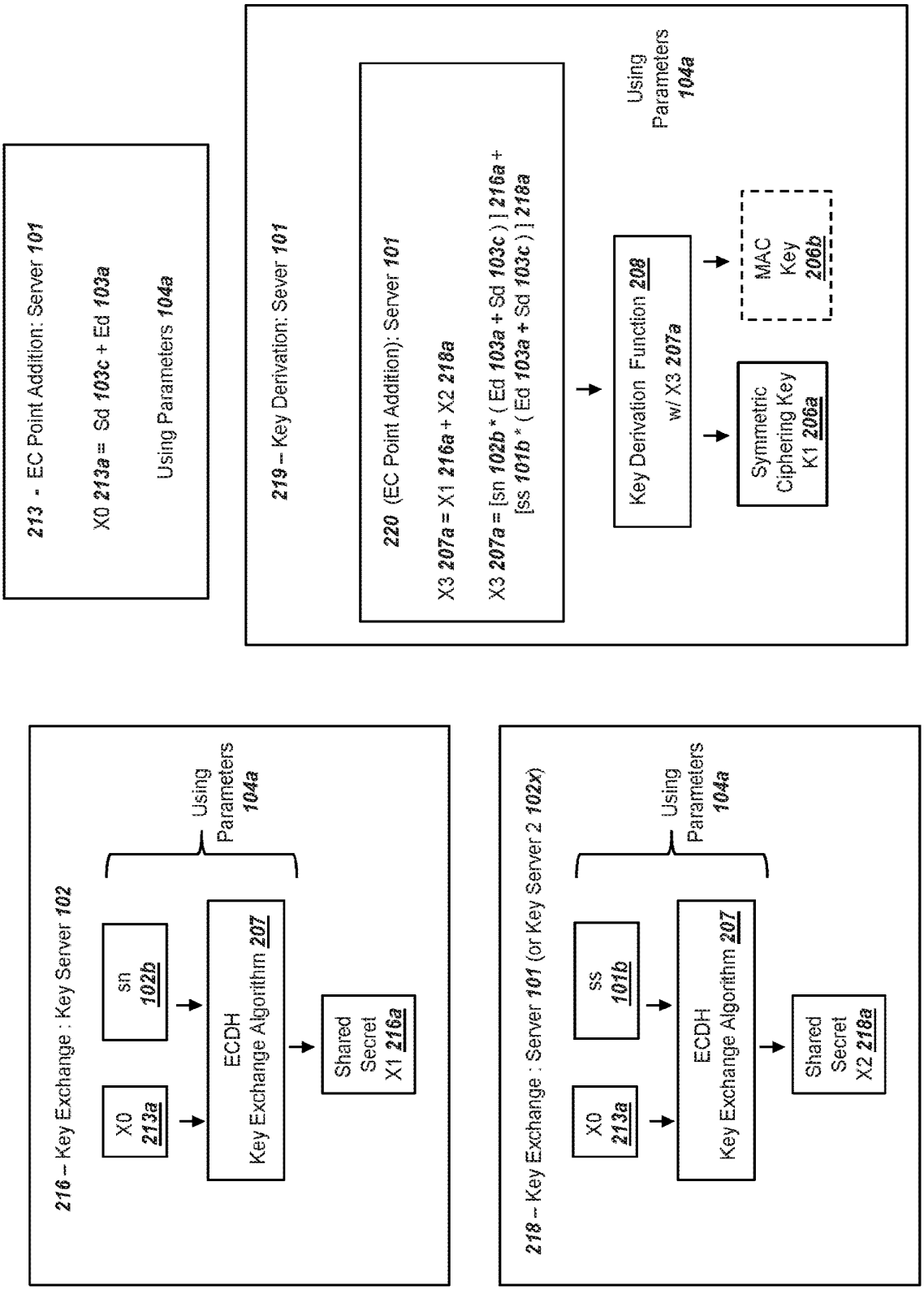
Petitioner Samsung Ex-1004, 1291

Figure 2c



Petitioner Samsung Ex-1004, 1292

Figure 2d



Petitioner Samsung Ex-1004, 1293

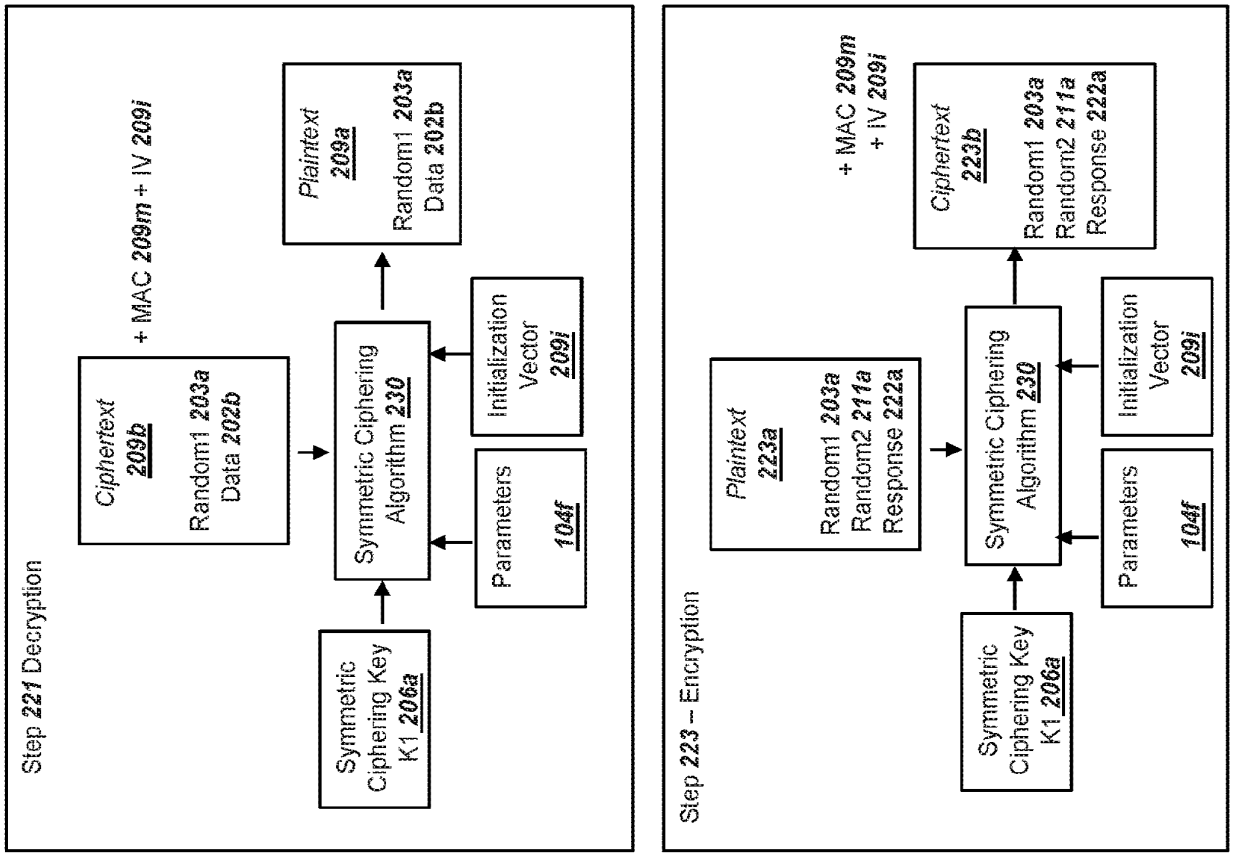


Figure 2e

Petitioner Samsung Ex-1004, 1294

Figure 2f

Server Database ***101d***

PKI Key Table ***101da***

Row	ID-token device ***103o***	ID Device ***103i***	H(Sd) ***250***	H(Sn) ***251***	Sd ***103c***	Ed ***103a***	Ss ***101a***	ss ***101b***	Sn ***102a***	ID key-server ***102i***
1	103o-1	103i-1			103c-1	103a-1	101a-1	101b-1	102a-1	102i-1
2		103i-2	250-2		103c-2	103a-2	101a-1	101b-1	102a-1	102i-1
3	103o-3	103i-3		251-1	103c-3	103a-3	101a-3	101b-3	102a-1	102i-1
4		103i-4	250-4	251-4	103c-4	103a-4	101a-4	101b-4	102a-4	102i-1
5	103o-5	103i-5	250-5	251-5	103c-5	103a-5	101a-5	101b-5	102a-5	102i-2
				●●●						

Device Table ***101db***

Row	ID-Device ***103i***	Random 1 ***203a***	Parameters ***104a***	Source IP:Port ***299p***	Radom 2 ***211a***	X0 ***213a***	X1 ***216a***	X2 ***218a***	X3 ***207a***	K1 ***206a***
1	103i-1	202a-1	104a-1	203a-1	211a-1	213a-1	216a-1	218a-1	207a-1	206a-1
2	103i-2	202a-2	104a-1	203a-2	211a-2	213a-2	216a-2	218a-2	207a-2	206a-2
3	103i-3	202a-3	104a-1	203a-3	211a-3	213a-3	216a-3	218a-3	207a-3	206a-3
4	103i-4	202a-3	104a-2	203a-4	211a-4	213a-4	216a-4	218a-4	207a-4	206a-4
5	103i-5	202a-3	104a-4	203a-5	211a-5	213a-5	216a-5	218a-5	207a-5	206a-5
				●●●						

Petitioner Samsung Ex-1004, 1295

Figure 3

Cryptographic Parameters **104**

Set **104a**	PKI Key Length **104b**	Curve **104c**	Hash **104d**	Symmetric Ciphering Key Length **104e**	Symmetric Ciphering Parameters **104f**	Random Length **104g**
A	256	p256	SHA-256	128	C1	128
B	384	p384	SHA-384	192	C2	192
C	512	secp521r1	SHA-512	256	C3	256
D	256	Curve25519	SHA-3	128	C4	128
E	448	Curve448	SHA-3	192	C5	192
			●●●			

Petitioner Samsung Ex-1004, 1296

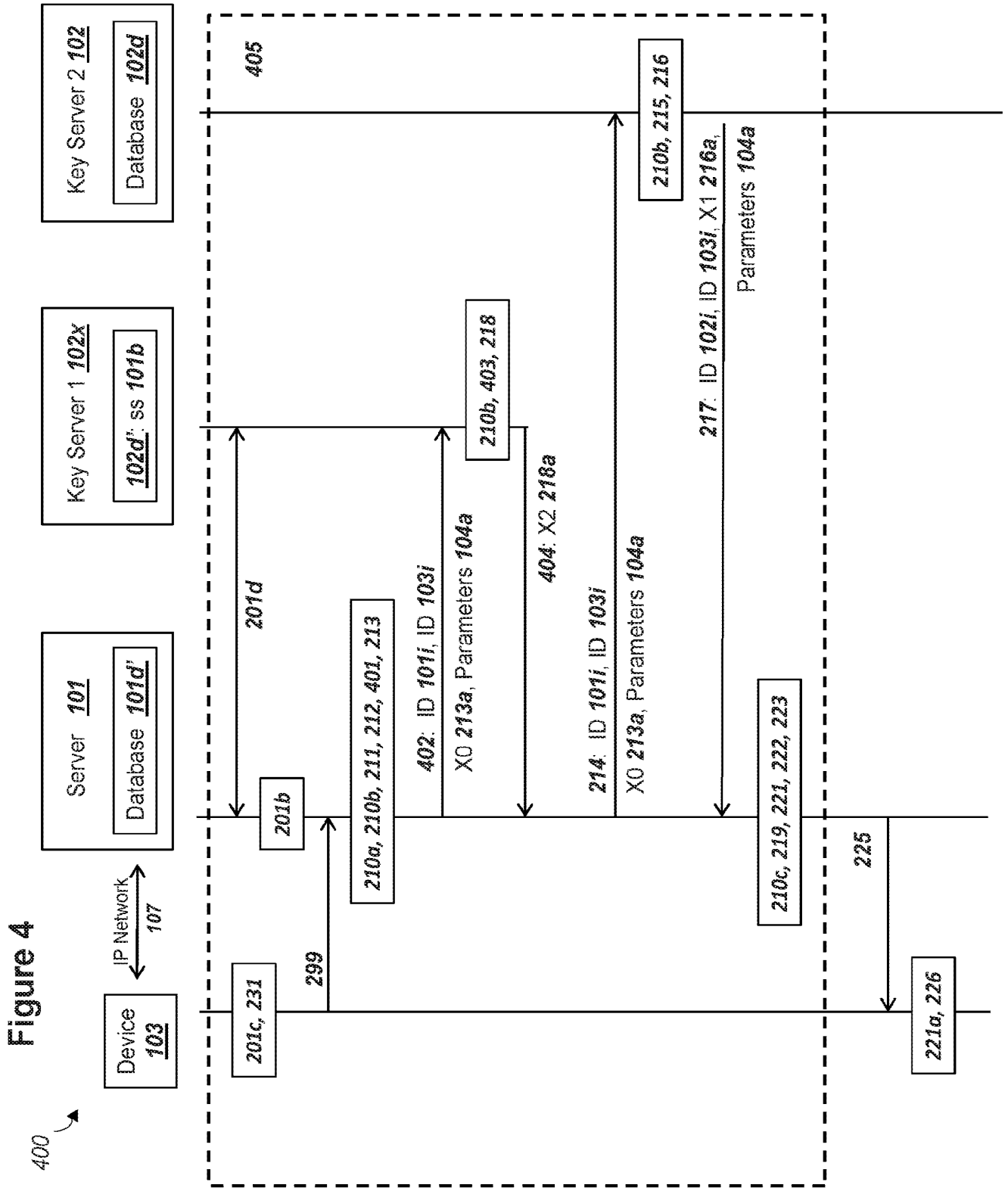


Figure 4

Petitioner Samsung Ex-1004, 1297

Figure 5a

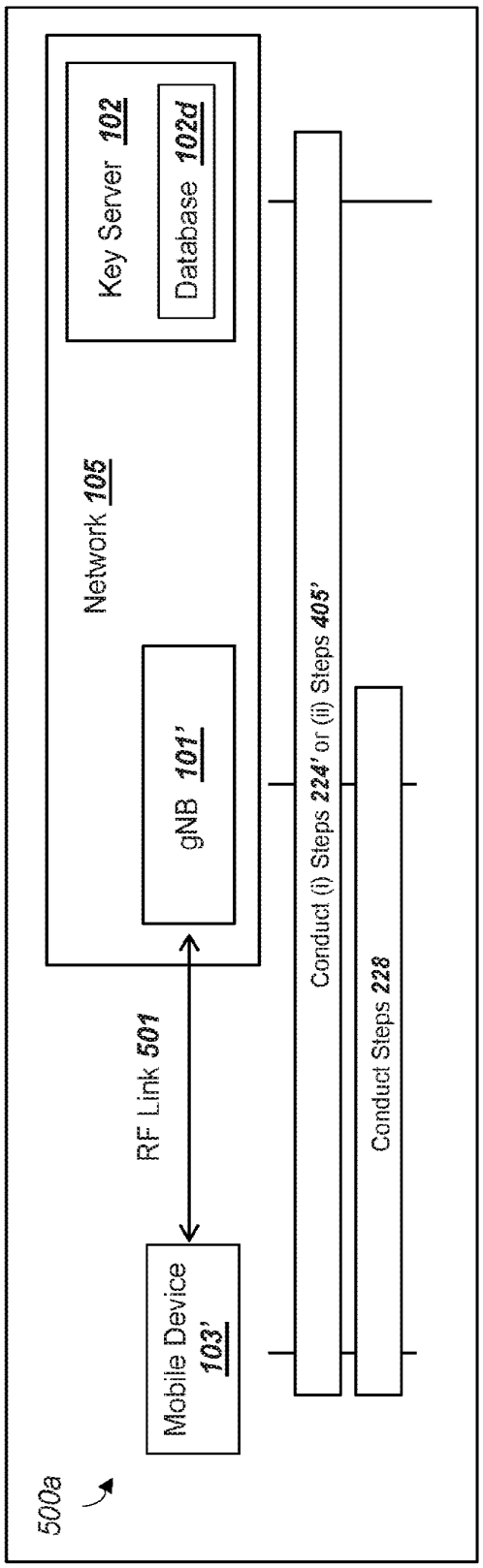
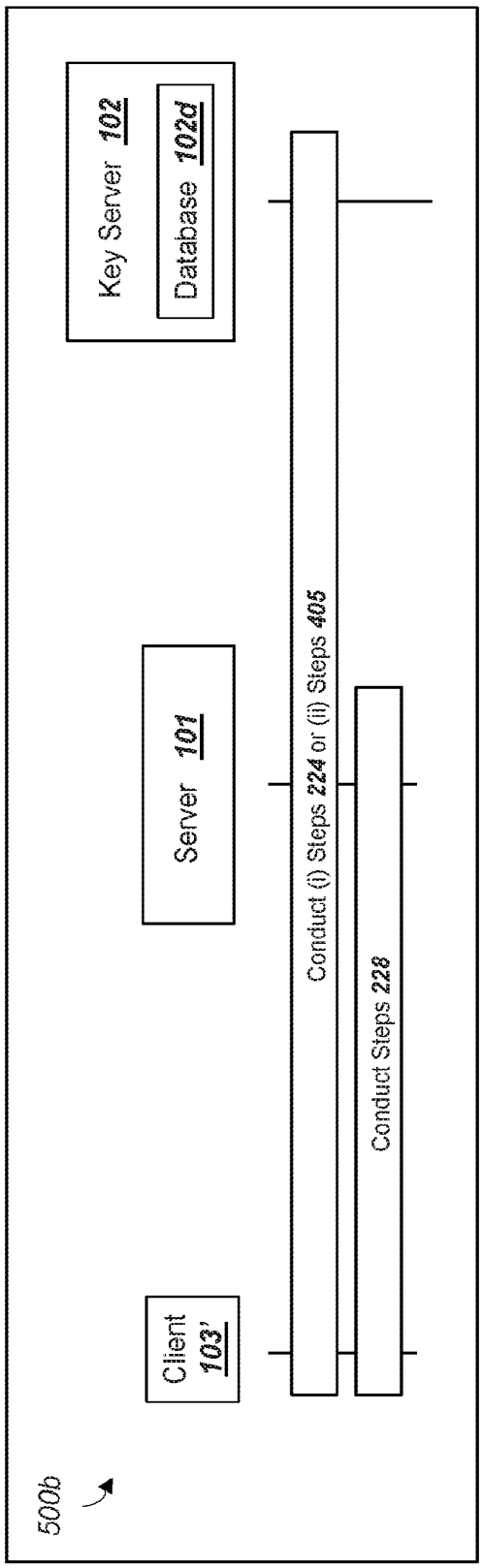


Figure 5b



Petitioner Samsung Ex-1004, 1298

Figure 5c

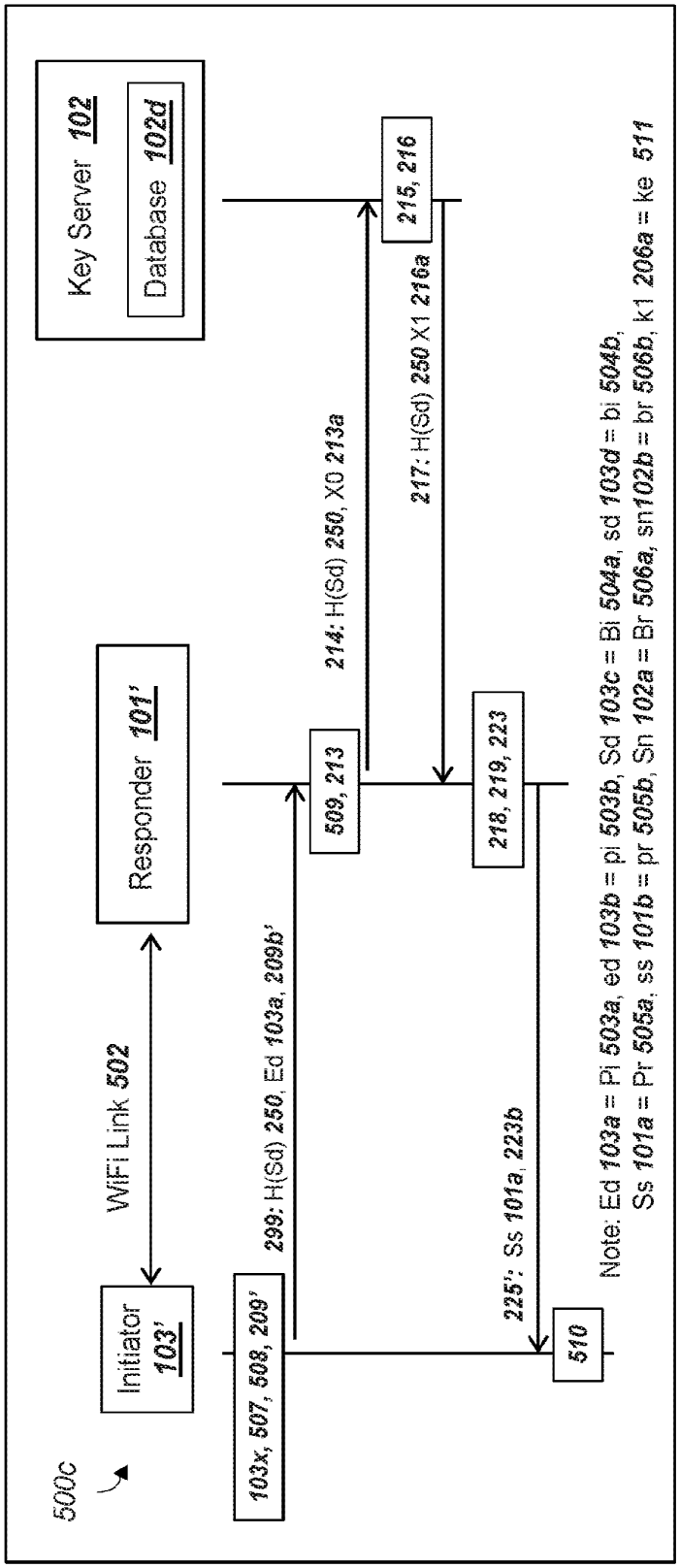
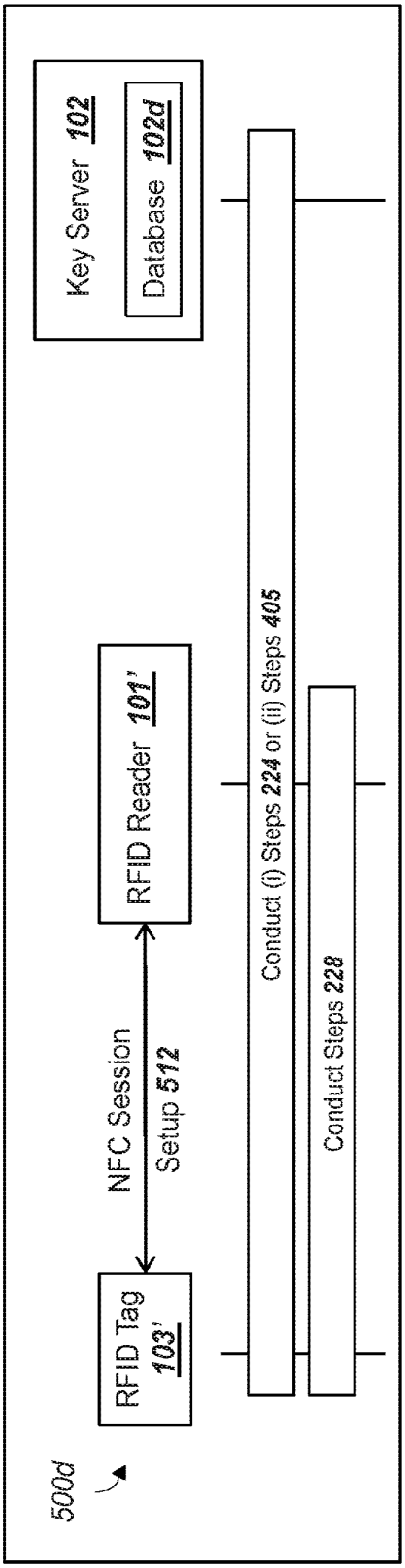


Figure 5d



Petitioner Samsung Ex-1004, 1299

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 2019/039380

A. CLASSIFICATION OF SUBJECT MATTER

H04L 9/08 *(2006.01)*
H04L 9/14 *(2006.01)*
G06F 21/30 *(2013.01)*

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)

H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

PatSearch (RUPTO internal), USPTO, PAJ, K-PION, Esp@cenet, Information Retrieval System of FIPS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2017/0111179 A1 (AKAMAI TECH INC) 20.04.2017	1-3, 6, 7, 12-15, 19-21, 24-29, 33-37
A	US 2016/0014114 A1 (CLOUDFLARE INC) 14.01.2016	1-3, 6, 7, 12-15, 19-21, 24-29, 33-37
A	US 2014/0010371 A1 (KHAZAN ROGER I) 09.01.2014	1-3, 6, 7, 12-15, 19-21, 24-29, 33-37

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

☐ See patent family annex.

* 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 document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
23 September 2019 (23.09.2019)	26 September 2019 (26.09.2019)

Name and mailing address of the ISA/RU:	Authorized officer
Federal Institute of Industrial Property, Berezhkovskaya nab., 30-1, Moscow, G-59, GSP-3, Russia, 125993 Facsimile No: (8-495) 531-63-18, (8-499) 243-33-37	A. Tokarev Telephone No. (495) 531-65-15

Form PCT/ISA/210 (second sheet) (January 2015)

Petitioner Samsung Ex-1004, 1300

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 2019/039380

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☒ Claims Nos.: 4, 5, 8-11, 16-18, 22, 23, 30-32, 38-42
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
☐ No protest accompanied the payment of additional search fees.

Petitioner Samsung Ex-1004, 1301

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
26 December 2019 (26.12.2019)

WIPO | PCT

(10) International Publication Number
WO 2019/246206 A1

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

(21) International Application Number:
PCT/US2019/037911

(22) International Filing Date:
19 June 2019 (19.06.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/687,411 20 June 2018 (20.06.2018) US

(71) Applicant: **IOT AND M2M TECHNOLOGIES, LLC** [US/US]; 807 Davis Street, Unit 2207, Evanston, IL 60201-7104 (US).

(72) Inventor: **NIX, John, A.**; c/o IoT and M2M Technologies, LLC, 807 Davis Street, Unit 2207, Evanston, IL 60201-7104 (US).

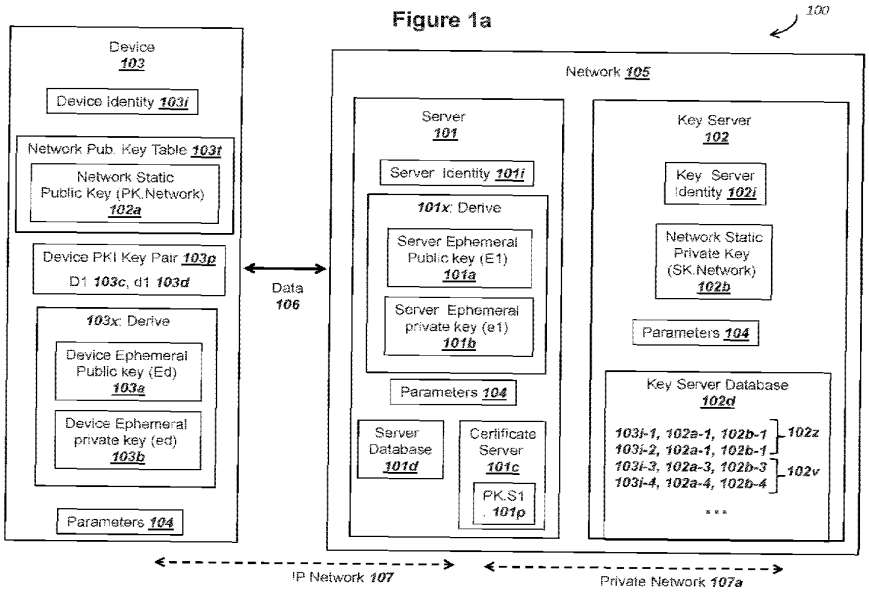
(74) Agent: **THOMSON, Kirsten, L.**; McDonnell Boehnen Hulbert & Berghoff LLP, 300 South Wacker Drive, Chicago, IL 60606 (US).

(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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, 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.

WO 2019/246206 A1

(54) **Title:** AN ECDHE KEY EXCHANGE FOR SERVER AUTHENTICATION AND A KEY SERVER

Figure 1a



(57) **Abstract:** A server can receive a device public key and forward the device public key to a key server. The key server can perform a first elliptic curve Diffie-Hellman (ECDH) key exchange using the device public key and a network private key to derive a secret XL The key server can send the secret X1 to the server. The server can derive an ECC PKI key pair and send to the device the server public key. Tire server can conduct a second ECDH key exchange using the derived server secret key and the device public key to derive a secret X2. The server can perforin an ECC point addition using the secret XI and secret X2 to derive a secret X3. The device can derive the secret X3 using (i) the server public key, a network public key, and the device private key and (ii) a third ECDH key exchange.

[Continued on next page]

Petitioner Samsung Ex-1004, 1302

(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))*

Petitioner Samsung Ex-1004, 1303

AN ECDHE KEY EXCHANGE FOR SERVER AUTHENTICATION AND A KEY SERVER

CROSS-REFERENCE TO RELATED APPLICATION

This international PCT application claims the benefit of the filing date of U.S. Provisional Patent Application Serial No. 62/687,411, filed June 20, 2018, which is hereby incorporated by reference in its entirety.

BACKGROUND

Technical Field

The present systems and methods relate to conducting an ephemeral elliptic curve Diffie Hellman key exchange (ECDHE) with authentication and multiple parties, and more particularly to communications between a computing device, a server, and a key server over a network in order for the computing device and the server to mutually derive a symmetric ciphering key with authentication of the server.

Description of Related Art

The use of elliptic curve cryptography (ECC) for computing devices has expanded over the past decade and is also expected to continue to grow. Many applications use or propose using ephemeral elliptic curve Diffie Hellman (ECDHE) key exchanges in order to derive a symmetric ciphering key. Prominent examples today include embedded universal integrated circuit cards (eUICCs) also known as embedded SIMs, Transport Layer Security (TLS) version 1.3 from the Internet Engineering Task Force (IETF), and the Device Provisioning Protocol (DPP) from the WiFi Alliance ™. Other examples are expected in the future as well, such as the use of ECDHE in order to protect the Subscription Permanent Identifier (SUPI) for 5G mobile networks, where the SUPI is equivalent to an International Mobile Subscriber Identity (IMSI). ECDHE can be considered a subset of elliptic curve Diffie-Hellman key exchanges (ECDH), where ECDHE key exchanges use at least one ephemeral on short-term elliptic curve PKI key pair. Applications use ECDHE key exchanges in order for two nodes to mutually derive a symmetric ciphering key and a message authentication code (MAC) key using a key derivation function. The symmetric ciphering key can subsequently be used with a symmetric ciphering algorithm such as the Advanced Encryption Standard (AES) and the MAC key can be used to verify message integrity. In this manner, secure communication can be established between two nodes.

Petitioner Samsung Ex-1004, 1304

ECDHE key exchanges depend on a first node deriving a first ephemeral private and public key pair and a second node deriving or using a second private and public key, where the public key infrastructure (PKI) keys use a common elliptic curve. The elliptic curve can be specified in parameters that define a named curve such as secp256r1 (p256), secp256k1, secp385r1, etc., and many other possibilities exist as well. ECDHE key exchanges have multiple benefits over older generation technology such as Diffie Hellman key exchanges. With ECDHE, elliptic curve cryptography can be utilized with shorter keys and faster processing times compared to previous technology, for the equivalent level of security or bit length of keys. For example, a 256 bit ECC PKI key pair can be used to obtain a comparable level of security as that obtained from using a 3072 bit RSA based PKI key pair. Calculation or processing time for conducting an ECDHE key exchange can also be faster than a traditional Diffie Hellman key exchange for the same level of security, as defined by the resulting key length of a derived shared secret from the key exchange.

Although the use of ECDHE key exchanges is growing rapidly, improvements can be made for ECDHE key exchanges in order to further enhance security and also leverage existing keys that may be recorded by the nodes participating in an ECDHE key exchange. As one example, an ECDHE key exchange as contemplated for (a) the exemplary applications and standards from two paragraphs above do not normally (b) provide authentication of either node. Separate steps than an ECDHE key exchange have to be conducted in order to authenticate endpoints, such as using an elliptic curve digital signature algorithm (ECDSA) with static or long-term ECC PKI keys recorded by the nodes. ECDSA algorithms also have challenges, where the reuse of a value k for two different signatures can reveal the private key. As another example and related to the authentication issue above, an ECDHE is susceptible to "man in the middle" attacks, where an intermediate node or different node than the intended node can perform the ECDHE key exchange instead of the intended node. Thus, although ECDHE can securely establish a symmetric ciphering key for confidentiality of data communications, the confidentiality could be established with a party or node that is not the intended recipient of the confidential communications. Consequently, a need exists in the art for the intended two nodes for confidential communications to use an ECDHE key exchange in a manner where at least one of the two nodes can be authenticated.

A primary goal of ECDHE key exchanges is also to obtain forward secrecy, where an ECDHE key exchange can periodically be re-conducted in order to rotate or re-establish a new symmetric ciphering key. In this manner, if a private key is compromised then only the subset of historical data encrypted using the compromised private key is subject to decryption, and other communications using a different private key can remain secured. An authenticated ECDH key exchange can be conducted using at least one static PKI key pair (e.g. not an ephemeral key exchange with ephemeral PKI keys), but without the benefits of forward secrecy. A need exists in the art where two parties can

Petitioner Samsung Ex-1004, 1305

conduct an authenticated ECDHE key exchange (e.g. by using ephemeral PKI keys) in order to obtain the benefits of forward secrecy.

The use of ECDH key exchanges (e.g. with at least one static PKI key pair) is also subject to greater security risks over time, where repeated use of one static PKI key pair is subject to cryptographic analysis and "leakage" of equivalent bits of security over time. Further, the use of ECDH key exchanges with one static PKI key pair and one ephemeral PKI key pair is more subject to risks of attacks from specifically chosen ephemeral PKI keys, such as ephemeral public keys that are either (i) not on the curve or (ii) specifically selected to expose information about the static private key. Thus, (a) repeated use of ECDHE key exchanges over time with different ephemeral PKI keys, compared to (b) using an ECDH key exchange with one static PKI key pair will result in greater security regarding confidentiality of communications. A need exists in the art where the greater security of ECDHE key exchanges can be obtained while also using static ECC PKI keys recorded by at least one of the nodes deriving a symmetric ciphering key using the ECDHE key exchange.

Many applications or new standards such as TLS version 1.3, DPP version 1.0 and 5G network standards from the 3rd Generation Partnership Project (3GPP) implement ECDHE key exchanges in order to quickly establish confidentiality early in the communications between two nodes. As noted above, a traditional ECDHE key exchange establishes confidentiality without authentication, and authentication must be obtained through other means, such as ECDSA or DSA with certificate verification, message digest, etc. However, the nodes participating in communications with the above standards typically have access to other, secure and previously recorded PKI keys besides the ephemeral PKI keys derived in order to conduct the ECDHE key exchange. A need exists in the art for a node to use the previously recorded PKI keys for (a) a new ECDHE key exchange in order to establish an authenticated key exchange without (b) the risks of ECDH key exchanges for static PKI keys as discussed above.

In addition, a device seeking to establish secured communication with a server may not be able to efficiently or securely verify a full certificate chain for a certificate from a server or a network, due to limitations such as lack of Internet connectivity for the device, lack of global date and time to properly check certificate revocation, incompatible parameters for verifying signatures from intermediate certificate authorities, etc. A need exists in the art for a device to use a previously recorded public key for a server or a network in order to include the public key in an authenticated ECDHE key exchange such that communications with a server can be secured without a separate requirement for full certificate verification through all intermediate root certificates to a root certificate stored in the device.

Solutions have been proposed in the art for an authenticated Diffie-Hellman or elliptic curve Diffie-Hellman key exchange using ephemeral keys and static keys. Blake-Wilson et al in the paper "Key Agreement Protocols and their Security Analysis", which is herein incorporated by reference,

Petitioner Samsung Ex-1004, 1306

propose the use of both long-term static keys and short-term ephemeral keys with a DH key exchange in order to conduct the key exchange in an authenticated manner in order to address some needs in the art mentioned above. Likewise, the Internet Engineering Task Force (IETF) proposes the use of elliptic curve ephemeral and static PKI keys in the "Request for Comments" (RFC) 5753 document "Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)", which is also hereby incorporated by reference.

However, the methods described for Blake-Wilson, RFC 5753, and related systems depend on (a) the recipient/responder of an ephemeral ECC public key from a sender/initiator to (b) also to record or operate with the static private ECC key corresponding to the static public key recorded by the sender. This can reduce scalability of a system with (i) a plurality of senders/initiators and (ii) a plurality of recipients/responders receiving an ephemeral ECC public keys for ECDHE key exchanges, since each recipient/responder also needs to record and operate on the static ECC private key corresponding to the static ECC public key recorded by the sender/initiator. The overall security of a system can be decreased for a system of potentially millions of devices and several servers, where the servers need to record server static private ECC keys corresponding to server static public ECC keys recorded by devices. A need exists in the art for (a) a recipient/responder to support authenticated ephemeral ECDH key exchanges without (b) the recipient/responder also recording the static ECC private key corresponding to the static ECC public key recorded by the sender/initiator.

Many other examples exist as well for needs in the art to conduct an ECDHE key exchange in a secure manner where at least one of the nodes can be authenticated, and the above are examples are just a few and intended to be illustrative instead of limiting.

Petitioner Samsung Ex-1004, 1307

SUMMARY

Methods and systems are provided for a server to conduct an ephemeral elliptic curve Diffie-Hellman key exchange (ECDHE) with a device and a key server. The device and the server can record and operate a set of compatible values and algorithms for a key pair generation algorithm, an ECDH key exchange algorithm, a key derivation function, a symmetric ciphering algorithm, and a random number generator, and a set of cryptographic parameters. The device can comprise a computing device with a network interface to communicate with the server via an IP network. The device can comprise a transducer device for operating a transducer and communicating the transducer data with the server via secured communications. The device can comprise a device for "the Internet of Things", a mobile phone, a tracking device, a security system, a module, or similar devices. The server can comprise a computing device with a network interface to communicate with the device via the IP network and the key server via a private network. The device can record a network static public key and a domain name service (DNS) name or uniform resource locator (URL) for the server. The key server can record the network static private key. The server can record and operate a server database. The device can be one of a plurality of different devices communicating with the server.

Before distribution to an end user of the computing device, a device manufacturer or a device distributor or a device owner could record a set of data in nonvolatile memory for the device. In addition to regular operating data and programs for the device, such as an operating system and a transducer driver, the data recorded in device before distribution could include (i) a network static public key, (ii) a set of cryptographic parameters associated with the network static public key, and (iii) a device identity for the computing device. For a first exemplary embodiment, the network static public key can be unique for the device and not shared with other devices. For a second exemplary embodiment, the network static public key can be shared across a set of devices and thus the network static public key would not be uniquely recorded in an individual device, but the network static public key could be uniquely recorded in a set of devices.

After power up and/or connecting with the IP network, the device can use the random number generator, the cryptographic parameters, and the key pair generation algorithm to derive a device ephemeral private key and a device ephemeral public key. The device can send the device ephemeral public key and the associated set of cryptographic parameters to the server in a first message using the DNS name or a URL for the server. The device can also optionally send a device identity or a secure hash value for the network static public key to the server, in order for the server to identify the device or set of devices. In some exemplary embodiments, the identity of the device and also the secure hash value can be omitted from the first message and the server identifies the group of devices by a particular IP address and port number and/or URL and/or DNS name used by the server and receiving

Petitioner Samsung Ex-1004, 1308

data from devices. In other words, without identifying data, (X) a subset of devices sending data (i) to the server where (ii) the server uses a particular name, URL, or IP address and/or port number can be identified by (Y) the server receiving data from the devices using the IP address and port number and/or URL and/or DNS name.

The server can receive the first message and process the first. The server can use the received set of cryptographic parameters to conduct an ECC point validation step to verify that the received ECC public key comprises a point on a named curve specified by the set of cryptographic parameters. The server can record the name or URL for a key server and communicate with the key server through a private network or a secured session over a public network such as the Internet. The server can establish a secure session with the key server. The server can (a) select the key server for the device using identifying information from the first message and then (b) forward the device ephemeral public key in a second message to the key server after the validation step. The second message can also include the set of cryptographic parameters. The identifying information from the first message for the device could comprise any of (i) an optional identity of the device in the first message, (ii) an optional secure hash value for the network static public key in the first message, (iii) the use of a particular set of cryptographic parameters in the first message, where the set of cryptographic parameters are associated with a particular key server, or (iv) the server can operate such that the use of a particular URL or IP address and port number is mapped to a particular key server.

The server can use a random number generator and a key pair generation algorithm and the set of cryptographic parameters to derive a random number for a server ephemeral private key and then use the server ephemeral private key to generate a server ephemeral public key. The server can conduct a first elliptic curve Diffie-Hellman key exchange (ECDH) using (i) the derived server ephemeral private key and received device ephemeral public key and (ii) the set of cryptographic parameters in order to derive a first shared secret. The server can also operate and record a key derivation function and a symmetric ciphering algorithm. The server can operate or be associated with a server database in order to record data for the server communicating with a plurality of different devices, such that different keys for different devices could be tracked by the server. In exemplary embodiments the first message is received with a random number generated by the device and also a source IP address and port number, and the server records the random number and the source IP address and port number for the first message in the server database.

The key server can receive the second message from the server over the secure connection. The second message can include the device ephemeral public key and the set of cryptographic parameters. For embodiments where the first message includes identifying information for the device (e.g. any of (i) through (iv) in the above paragraph), then the second message to the key server can also include the identifying information for the device. The key server can select or read the network static private key using the second message received from the server (including possibly identifying

Petitioner Samsung Ex-1004, 1309

information of the device such as, but not limited to, a secure hash value for the network static public key, to select a specific network static private key for the device). The key server can conduct a second ECDH key exchange using (i) the selected network static private key for the device and the received device ephemeral public key and (ii) the set of cryptographic parameters in order to derive a second shared secret. The key server can send a response to the second message in the form of a third message to the server, where the third message includes the derived second shared secret.

The server can receive the derived second shared secret from the key server in the third message. The third message can also include identifying information such that the server can track which of the devices the third message from the key server is associated with. The server can conduct an ECC point verification step to verify that the received point from the key server comprising the second shared secret is a point on the ECC curve specified by the set of cryptographic parameters received in the first message. The server can conduct an ECC point addition operation using (i) the derived first shared secret by the server and (ii) the received second shared secret from the key server. The resulting value from the ECC point addition operation can comprise a third shared secret. The server can input the third shared secret into a key derivation function in order to derive a symmetric ciphering key. The server can encrypt a random number generated by the server and a certificate for the server using the derived symmetric ciphering key and the symmetric ciphering algorithm. The server can generate a digital signature for a fourth message with the certificate and the random number using the private key corresponding to the public key in the certificate. The data encrypted by the server, including the digital signature, can comprise a first ciphertext. The server can send the device the fourth message, where the fourth message includes the server ephemeral public key and the first ciphertext.

The device can receive the fourth message from the server and take steps to process the message. The device can conduct a third ECDH key exchange with the received server ephemeral public key. The device, using the set of cryptographic parameters, can perform an elliptic curve point addition operation on (i) the received server ephemeral public key from the fourth message and (ii) the recorded network static public key. The device can input (x) the point derived from the ECC point addition and (y) the device ephemeral private key into an ECDH key exchange algorithm in order to mutually derive the third shared secret with the server. Device can input the third shared secret into a key derivation function in order to derive the same symmetric ciphering key derived and used by the server. The device can decrypt the first ciphertext using the derived symmetric ciphering key. The device can read the plaintext from the first ciphertext. The device can take additional steps to communicate with the server, such as (i) verifying a signature in a certificate within the first ciphertext, (ii) using the public key from the certificate to verify a server digital signature for the fourth message, (iii) recording and using a public key for the server from the certificate in the first ciphertext, and other possibilities exist as well. The device can then use the derived symmetric ciphering key to encrypt a

Petitioner Samsung Ex-1004, 1310

second ciphertext for the server and send the second ciphertext to the server in a fifth message. In exemplary embodiments, the derived symmetric ciphering key can comprise a first portion for encrypting and decrypting data sent from the server to the device and a second, different portion for encrypting and decrypting data sent from the device and to the server. The server can receive the fifth message and decrypt the second ciphertext using the same symmetric ciphering key derived by the server.

The systems and methods described above can also be used with particular implementations for a computing device and a server. A 5th generation or 6th generation wireless WAN network such as from 3GPP could utilize the steps above in order to conduct an ECDHE key exchange with a server authentication and a key server. For this embodiment, the computing device could comprise a wireless device or wireless terminal, including a mobile phone or smart phone. The server could comprise a "g Node B" for "next generation node b", which provides equivalent functionality of a base transceiver station and manages the radio-frequency communications with the wireless device. The key server could comprise a secured server operating within the authentication function of a wireless network or associated with the authentication function of a wireless network for a mobile network operator. For the embodiment in this paragraph, the cryptographic parameters could comprise the values for curve 25519, although other ECC curves could be utilized as well.

The systems and methods described above can also be used with a device provisioning protocol. The computing device as described above can comprise an initiator according to the Device Provisioning Protocol specification version 1.0 from the WiFi Alliance ®. The server can comprise a responder according to the same specification. Subsequent versions of the specification can utilize the methods and systems described herein as well. The device can receive and record the network static public key in the form of a responder bootstrap public key. A key server could record the network static private key in the form of a responder bootstrap private key. The responder/server can receive the first message with (a) the device ephemeral public key from the initiator/device along with (b) a secure hash value of the responder bootstrap public key, and (c) an initial ciphertext. The responder/server can use the secure hash value of the responder bootstrap public key to select the key server for the device. The responder/server can forward the device ephemeral public key to the selected key server. The key server can conduct the second ECDH key exchange with the device ephemeral public key and the responder bootstrap private key and send the second shared secret to the responder/server. The server can use the second shared secret to decrypt the initial ciphertext received with the first message.

These as well as other aspects and advantages will become apparent to those of ordinary skill in the art by reading the following detailed description, with reference where appropriate to the accompanying drawings.

Petitioner Samsung Ex-1004, 1311

BRIEF DESCRIPTION OF THE DRAWINGS

Various exemplary embodiments are described herein with reference to the following drawings, wherein like numerals denote like entities.

Figure 1a is a graphical illustration of an exemplary system, where device communicates data with a network in order to conduct a key exchange, in accordance with exemplary embodiments;

Figure 1b is a graphical illustration of hardware, firmware, and software components for a server, in accordance with exemplary embodiments;

Figure 1c is an illustration of exemplary network static public keys recorded by a plurality of devices, in accordance with exemplary embodiments;

Figure 2a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device, a server, and a key server, in accordance with exemplary embodiments;

Figure 2b is a flow chart illustrating exemplary steps for conducting a key exchange using PKI keys in order to derive shared secrets, and for conducting a key derivation function using the derived shared secrets, in accordance with exemplary embodiments;

Figure 2c is a flow chart illustrating exemplary steps for conducting a key exchange using PKI keys in order to derive a shared secret key, and for using the derived shared secret key to encrypt and decrypt data, in accordance with exemplary embodiments;

Figure 2d is an illustration of an exemplary server database and an exemplary set of cryptographic parameters, in accordance with exemplary embodiments;

Figure 2e is a flow chart illustrating exemplary steps for conducting a key exchange using PKI keys in order to derive a shared secret key using ECC point multiplication, in accordance with exemplary embodiments;

Figure 3a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a mobile device, a "g node b", and a key server, in accordance with exemplary embodiments;

Figure 3b is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a client, a server, and a key server, in accordance with exemplary embodiments; and,

Figure 3c is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by an initiator, a responder, and a key server, in accordance with exemplary embodiments.

Petitioner Samsung Ex-1004, 1312

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS OF THE INVENTION

Figure 1a

Figure 1a is a graphical illustration of an exemplary system, where device communicates data with a network in order to conduct a key exchange, in accordance with exemplary embodiments. The system 100 can include a device 103 and a network 105, where the nodes can communicate data 106 over an Internet Protocol (IP) network 107. Network 105 can comprise a plurality of servers supporting communication such as data 106 with a plurality of devices 103. In exemplary embodiments, network 105 can include a server 101 and a key server 102. The exemplary servers shown for network 105 in system 100 can be either different physical computers such as rack-mounted servers, or different logical or virtual servers or instances operating in a "cloud" configuration. Or, server 101 and key server 102 could represent different logical "server-side" processes within a network 105, including different programs running on a server that listen and communicate using different IP port numbers within one physical server. In exemplary embodiments, server 101 and key server 102 can operate using the physical electrical components depicted and described for a server 101 in Figure 1b below. Other possibilities exist as well for the physical embodiment of server 101 and key server 102 without departing from the scope of the present disclosure. In exemplary embodiments, server 101 can be described as a "first server" and key server 102 can be described as a "second server". Further, the combination of a first server 101 and a second server 102 can comprise a network 105. The combination of a first server 101 and a second server 102 can also comprise a "set of servers".

Although server 101 and key server 102 are depicted in Figure 1a as belonging to the same network 105, server 101 and key server 102 could be associated with different networks and communicate in a secure manner. Secure sessions between server 101 and key server 102 could be established over IP network 107 using methods including a physical wired connection via a local area network (LAN), transport layer security (TLS), a virtual private network (VPN), and IP Security (IPSEC), and other possibilities exist as well. As depicted in Figure 1a, server 101 and key server 102 could communicate over a private network 107a.

Device 103 can be a computing device for sending and receiving data. Device 103 can take several different embodiments, such as a general purpose personal computer, a mobile phone based on the Android ® from Google ® or the IOS operating system from Apple ®, a tablet, a networked device with a sensor or actuator for the "Internet of Things", a module for "machine to machine" communications, a device that connects to a wireless or wired Local Area Network (LAN), an initiator according to the Device Provisioning Protocol specification (DPP) from the WiFi alliance, a router, and/or a server, and other possibilities exist as well without departing from the scope of the present

Petitioner Samsung Ex-1004, 1313

disclosure. Exemplary electrical components within a device 103 can be similar to the electrical components for a server 101 depicted and described in Figure 1b below, where device 103 can use electrical components with smaller capacities and lower overall power consumption, compared to the capacity and power consumption for the same electrical components in a server 101.

Device 103 can include a device identity 103i, which could comprise a string or number to uniquely identify device 103 with network 105 and/or server 101 and server 102. Device identity 103i could comprise a medium access control (MAC) address for a physical interface such as Ethernet or WiFi, a Subscription Permanent Identifier (SUPI) with 5G networks, an international mobile subscriber identity (IMSI) or international mobile equipment identity (IMEI) with 2G/3G/4G networks, and other possibilities exist as well without departing from the scope of the present disclosure. In exemplary embodiments, device identity 103i can be written to hardware in device 103 and operate as a unique, long-term identity for device 103.

Device 103 can record at least one elliptic curve cryptography (ECC) static public key for network 105 comprising network static public key PK.network 102a. Network static public key 102a could be recorded in nonvolatile or volatile memory within device 103. For embodiments where key 102a is recorded in nonvolatile memory, key 102a could be recorded by a device manufacturer or device distributor. For embodiments where key 102a is recorded in volatile memory, device 103 could obtain key 102a from a different server than server 101 for network 105 before sending data 106, such as device 103 obtaining key 102a via a secure session from a different server before sending data 106. A device 103 can record a plurality of different network static public keys 102a in a network public key table 103t. Different keys 102a in a table 103t could be associated with different networks 105 or different servers 101 that device 103 communicates with over time. Exemplary data for a network public key table 103t for device 103 is depicted and described in connection with Figure 1c below. The different keys 102a can be associated with network names and/or Uniform Resource Locators (URLs) or domain names, such that device 103 can select the network static public key 102a based on a URL or domain name for a network 105 or a server 101 where device 103 will send data 106.

Network static public key PK.network 102a can be obtained by device 103 before conducting an elliptic curve Diffie-Hellman (ECDH) key exchange or an ephemeral elliptic curve Diffie-Hellman (ECHDE) key exchange. Network static public key 102a could be obtained by device 103 in several different ways. Network static public key 102a could be written into memory by a manufacturer, distributor, or owner of device 103 before device 103 connects with server 101 or a network 105. Network static public key 102a could be received by device 103 over an IP network 107 via a secured session, such as a TLS, DTLS, IPSec, or VPN connection before sending data 106 to server 101. In exemplary embodiments, network static public key 102a is recorded in device 103 in a secured and authenticated manner, such that device 103 can trust network static public key 102a.

Petitioner Samsung Ex-1004, 1314

As one exemplary embodiment, network static public key 102a could be a public key within a certificate, where the public key 102a is signed by a certificate authority. Although not depicted in Figure 1a, device 103 could also record a certificate authority root certificate, and device 103 could (a) verify the signature of a certificate authority in a certificate for the public key 102a using (b) the recoded root certificate for the certificate authority (and any intermediary parent certificates). Network static public key 102a could be processed or formatted according to a set of parameters 104, and network static public key 102a could also be compatible with parameters 104. Although public key 102a is described as "static", the key could change over time such as with the expiration of a validity date when recorded in a certificate. Public key 102a could remain static over the period of time for device 103 to conduct at least one ECDHE key exchange, where the ECDHE key exchange uses ephemeral or derived ECC PKI keys. Public key 102a could comprise a long-term public key for use by device 103 when communicating with network 105. Although the use of a certificate for public key 102a is described in this paragraph for public key 102a, the use of a certificate is not required. In an embodiment depicted in Figure 3c below, (i) public key 102a could comprise a responder bootstrap public key and (ii) device 103 could comprise an initiator according to the DPP standard, which is also depicted and described in connection with Figure 3c below.

Cryptographic parameters 104 can specify values or settings for (i) conducting an ECDH or ECDHE key exchange, (ii) mutually deriving a symmetric ciphering key, and (iii) using a symmetric ciphering algorithm. As contemplated herein, cryptographic parameters 104 may also be referred to as parameters 104. Each of device 103, server 101, and key server 102 can record at least one compatible subset of parameters within a set of cryptographic parameters 104. Parameters 104 can specify values for an elliptic curve cryptography (ECC) curve name, key length, key formatting (e.g. compressed or uncompressed), encoding rules, etc. As contemplated herein, the parameters 104 and cryptographic algorithms used with ECC PKI keys and a key exchange in the present disclosure can be compatible and substantially conform with ECC algorithms and keys as specified in (i) the IETF Request for Comments (RFC) 6090 titled "Fundamental Elliptic Curve Cryptography Algorithms", and (ii) IETF RFC 5915 titled "Elliptic Curve Private Key Structure", and also subsequent and related versions of these standards. For use of ECC algorithms, parameters 104 can specify elliptic curve names such as, but not limited to NIST P-256, sect283k1, sect283r1, sect409k1, sect409r1, and other possibilities exist as well. Further, elliptic curves that do not depend on curves currently specified by the National Institute of Standards and Technology (NIST) could be utilized as well, such as, but not limited to, Curve22519, curve448, or FourQ. Parameters 104 can specify domain parameters for nodes in system 100 to calculate values or numbers in a compatible manner, such as common base point G for use with ECC PKI key pairs and a defining equation for an elliptic curve. Other values within sets of cryptographic parameters 104 are possible as well, without departing from the scope of the present disclosure. An exemplary set of cryptographic parameters 104 is depicted and described in connection

Petitioner Samsung Ex-1004, 1315

with Figure 2e below, and PKI keys used by device 103, server 101, and key server 102 could be associated with a member of the set of cryptographic parameters, such as a single row in the parameters 104 depicted in Figure 2e below.

Device 103 can include an ECC key pair generation algorithm 103x and server 101 can include a compatible ECC key pair generation algorithm 101x. A key pair generation algorithm 103x or 101x can use (i) a random number generator in order to derive the ephemeral PKI private key and (ii) a selected set of cryptographic parameters 104 in order to calculate the ephemeral PKI public key. In exemplary embodiments, a random number for the ephemeral PKI private key multiplies the base point G from the parameters 104 in order to obtain the corresponding ephemeral PKI public key. Other possibilities exist as well for the algorithms 103x and 101x to derive an ephemeral ECC PKI key pair without departing from the scope of the present disclosure. A key pair generation algorithm 103x for device 103 can output an ephemeral ECC PKI pair comprising device ephemeral public key Ed 103a and device ephemeral private key ed 103b. A key pair generation algorithm 101x for server 101 can output an ephemeral ECC PKI pair comprising server ephemeral public key E1 101a and server ephemeral private key e1 101b. As contemplated in the present disclosure, the use of a capital letter as the first character for a PKI key can represent a public key, the use of a lower case letter as the first character for a PKI key can represent a private key. As contemplated in the present disclosure, the second letter for a PKI key can represent the entity the key is associated with or belongs to (e.g. "d" for device 103 and "1" for server 101).

Device 103 can also record a device static PKI key pair 103p in nonvolatile memory or within a secure processing environment within device 103. The key pair 103p can be either (i) generated by device 103 during device manufacturing or device distribution, or (ii) generated externally from device 103 and written to device 103 in a secure manner during device manufacturing or device distribution. The PKI key pair 103p can comprise a device static private key d1 103d and a device static public key D1 103c. The keys d1 103d and D1 103c could be formatted and compatible with the set of cryptographic parameters 104. In exemplary embodiments, public key D1 103c can be recorded in an X.509 certificate from a certificate authority.

As depicted in Figure 1a, server 101 can include a server identity 101i, a key pair generation algorithm 101x, a set of cryptographic parameters 104, a server database 101d, and a server certificate 101c. Server identity 101i can comprise a name or number to uniquely identify server 101 in network 105 and/or IP network 107. In exemplary embodiments, server identity 101i can comprise a domain name service (DNS) name, which could comprise a string of characters and/or numbers. Server identity 101i could be associated with an IP address, such that the exemplary data 106 from device 103 could be routed to server 101 via the IP network 107. Server identity 101i could also comprise a MAC address, and a server identity 101i could comprise multiple different values such as all of a MAC address, a DNS name, and virtual instance identity if server 101 operates as a virtual server. In

Petitioner Samsung Ex-1004, 1316

summary, server identity 101i can allow (a) a plurality of different devices 103 to (b) select and route data 106 to server 101 from a potential plurality of different servers and nodes. Server identity 101i could also comprise a server name indication (SNI) value. Other possibilities exist as well for the format, structure, or value for a server identity 101i without departing from the scope of the present disclosure.

A key pair generation algorithm 101x for server 101 was described above in connection with key pair generation algorithm 103x for device 103. Key pair generation algorithm 101x can derive ephemeral ECC PKI keys for server 101 to use with ECDHE key exchanges for a plurality of different devices 103. Note that although a single ECC PKI key pair of public key E1 101a and private key e1 101b is depicted for system 100, server 101 could derive and operate with a plurality of different keys E1 101a and e1 101b with different devices 103. The plurality of different keys E1 101a and e1 101b for communicating with different devices 103 could be recorded in a server database 101d as depicted and described in connection with Figure 2d below. The set of cryptographic parameters 104 for server 101 can be equivalent to or a superset of the cryptographic parameters 104 used by device 103. The description above for a set of parameters 104 used by a device 103 is also applicable to a set of parameters 104 used by a server 101.

Server database 101d for server 101 can comprise a database or memory recording data for server 101 to communicate with both a plurality of devices 103 and also at least one key server 102. An exemplary server database 101d is depicted and described in connection with Figure 2d below. Server database 101d can record values for PKI keys, derived shared secrets, derived symmetric ciphering keys, random numbers used in secure sessions, and related values in order to support the communications with both device 103 and key server 102. Server certificate 101c can comprise a certificate formatted according to the X.509 family of standards and include a static server 101 public key PK.S1 101p. Server certificate 101c can include a signature from a certificate authority for server public key PK.S1 101p. Although not depicted in Figure 1a, server 101 can also record and operate with a private key corresponding to public key PK.S1 101p.

As depicted in Figure 1a, key server 102 can include a key server identity 102i, a set of cryptographic parameters 104, a network static private key SK.network 102b, and a key server database 102d. Key Server identity 102i can comprise a name or number to uniquely identify key server 102 in network 105 and/or IP network 107. Key Server identity 102i can be similar to server identity 101i, except using a different value, name, or number in order to uniquely identify key server 102 within network 105. The set of cryptographic parameters 104 for server 102 can be equivalent to or a superset of the cryptographic parameters 104 used by device 103 and parameters 104 was also described above for device 103.

In exemplary embodiments, the parameters 104 used by both key server 102 and server 101 can be fully compatible, such as using the same ECC named curve, key lengths, encoding rules, etc.

Petitioner Samsung Ex-1004, 1317

Server database 102d for key server 102 can comprise a database or memory recording data for key server 102 to (i) communicate with a plurality of servers 101 and (ii) support server 101 communicating with a plurality of devices 103. Key server database 102d can be similar to server database 101d depicted in Figure 2d, except that key server database 102d can record values and data calculated by key server 102. Key server database 102d can record values for PKI keys, derived shared secrets, and related values in order to support the communications between (i) network 105 and/or server 101 and (ii) device 103. As depicted in Figure 1a, key server database 102d can record sets of data for different devices 103, where each set can comprise a row in a table with a device identity 103i, the network static public key value PK.network 102a, and the network static private key SK.network 102b. Although not depicted in Figure 1a, a key server database 102d could also record or store a secure hash value for the network public key 102a, where the algorithm for the secure hash value could be specified in a member of the set of cryptographic parameters 104. For some exemplary embodiments, (i) a device identity 103i could be omitted from a key server database 102d or (ii) the device identity 103i could comprise a secure hash value over either the network public key 102a or the device static public key 103c.

As depicted for a key server database 102d in Figure 1a, some devices 103 could share the same keys 102a and 102b, which could comprise shared keys 102z for the devices 103 as depicted and described in connection with Figure 1c below. Other devices 103 could record unique keys 102v, where devices 103 record a value for the network static public key PK.network 102a that is uniquely recorded in each device. A key server database 102d could record and track the associated network private and public keys for each device. In other exemplary embodiments, a key server 102 could omit recording device identities 103i in a database 102d, and key server 102 could associate and use a network static private key SK.network 102b with a particular server 101 (e.g. all data from a server 101 could use or be associated with the private key SK.network 102b).

Other possibilities exist as well for the mapping of network static private keys to either servers 101 or devices 103 without departing from the scope of the present disclosure. Also, although a single value for SK.network 102b is depicted as associated with a device 103, a key server 102 could also use multiple different values of SK.network 102b, such as (i) different values for SK.network 102b for different parameters 104 (e.g. different named curves), or (ii) separate values for SK.network 102b for digital signatures and ECDH key exchanges. In other words, a device 103 could also record the corresponding different multiple values for PK.network 102a, and select and use the public keys depending on requirements such as parameters 104 used or if the network public key will be used for verifying digital signatures or conducting ECDH key exchanges.

Key server 102 can record at least one network static private key SK.network 102b, which can be the private key corresponding to the network static public key PK.network 102a recorded by device 103 and described above for device 103. In exemplary embodiments and as depicted in Figure 1a and

Petitioner Samsung Ex-1004, 1318

also Figure 2a below, key server 102 may not communicate with device 103 directly, but rather communicates with server 101 through a private network 107a. Although not depicted in Figure 1a, a network 105 could operate a firewall in order to prevent packets or data from the public Internet (other than server 101) from reaching key server 102. In this manner by isolating key server 102 from IP network 107, security for the key server 102 and the network static private key SK.network 102b can be enhanced, since only authenticated and authorized nodes within network 105 and connected to private network 107a could communicate with server 102.

IP network 107 could be either a Local Area Network (LAN) or a Wide Area Network (WAN), or potentially a combination of both. IP network 107 could include data links supporting either IEEE 802.11 (WiFi) standards. Device 103 also utilize a variety of WAN wireless technologies to communicate data 106 with server 101, including Low Power Wide Area (LPWA) technology, 3rd Generation Partnership Project (3GPP) technology such as, but not limited to, 3G, 4G Long-Term Evolution (LTE), or 4G LTE Advanced, NarrowBand – Internet of Things (NB-IoT), LTE Cat M, proposed 5G networks, and other examples exist as well. Server 101 can connect to the IP network 107 via a wired connection such as, but not limited to, an Ethernet, a fiber optic, or a Universal Serial Bus (USB) connection (not shown). IP network 107 could also be a public or private network supporting Internet Engineering Task Force (IETF) standards such as, but not limited to, such as, RFC 786 (User Datagram Protocol), RFC 793 (Transmission Control Protocol), and related protocols including IPv6 or IPv4. A public IP network 107 could utilize globally routable IP addresses. Private IP network 107a could utilize private IP addresses which could also be referred to as an Intranet. Other possibilities for IP Network 107 and Private Network 107a exist as well without departing from the scope of the disclosure.

Figure 1b

Figure 1b is a graphical illustration of hardware, firmware, and software components for a server, in accordance with exemplary embodiments. Figure 1b is illustrated to include several components that can be common within a server 101. Server 101 may consist of multiple electrical components in order to communicate with a plurality of devices 101 and a key server 102. In exemplary embodiments and as depicted in Figure 1b, server 101 can include a server identity 101i, a processor 101e (depicted as "CPU 101e"), random access memory (RAM) 101f, an operating system (OS) 101g, storage memory 101h (depicted as "nonvolatile memory 101h"), a Wide Area Network (WAN) interface 101j, a LAN interface 101k, a system bus 101n, and a user interface (UI) 101m.

Server identity 101i could comprise a preferably unique alpha-numeric or hexadecimal identifier for server 101, such as an Ethernet MAC address, a domain name service (DNS) name, a Uniform Resource Locator (URL), an owner interface identifier in an IPv6 network, a serial number,

Petitioner Samsung Ex-1004, 1319

an IP address, or other sequence of digits to uniquely identify each of the many different possible nodes for a server 101 connected to an IP network 105. Server identity 101i could comprise a server name indicator (SNI). Server identity 101i can preferably be recorded in a non-volatile memory and recorded by a network 105 upon configuration of a server 101. Server identity 101i may also be a number or string to identify an instance of server 101 running in a cloud or virtual networking environment. In exemplary embodiments, server 101 can operate with multiple different server identities 101i, such as a first server identity 101i comprising a DNS name and a second server identity 101i comprising an IP address and a port number. A different server 101 could be associated with a different IP address and port number for a network 105.

The CPU 101e can comprise a general purpose processor appropriate for higher processing power requirements for a server 101, and may operate with multiple different processor cores. CPU 101e can comprise a processor for server 101 such as an ARM® based process or an Intel® based processor such as belonging to the XEON ® family of processors, and other possibilities exist as well. CPU 101e can utilize bus 101n to fetch instructions from RAM 101f and operate on the instruction. CPU 101e can include components such as registers, accumulators, and logic elements to add, subtract, multiply, and divide numerical values and store or record the results in RAM 101f or storage memory 101h, and also write the values to an external interface such as WAN interface 101j and/or LAN interface 101k. In exemplary embodiments, CPU 101e can perform the mathematical calculations for a key pair generation step 101x and also an ECDH key exchange algorithm 220 depicted in Figure 2a, Figure 2b, etc., below.

CPU 101e can also contain a secure processing environment (SPE) 101s in order to conduct elliptic curve cryptography (ECC) operations and algorithms, such as an ECC point addition step 214 depicted in Figure 2c below, as well as deriving ephemeral ECC PKI keys such as with key generation step 101x depicted and described in connection with Figure 1a above. SPE 101s can comprise a dedicated area of silicon or transistors within CPU 101e in order to isolate the ECC operations from other programs or software operated by CPU 101e, including many processes or programs running operating system 101g. SPE 101s could contain RAM memory equivalent to RAM 101f and nonvolatile memory equivalent to storage memory 101h, as well as a separately functioning processor on a smaller scale than CPU 101e, such as possibly a dedicated processor core within CPU 101e. SPE 101s can comprise a "secure enclave" or a "secure environment", based on the manufacturer of CPU 101e. In some exemplary embodiments, an SPE 101s can be omitted and the CPU 101e can conduct ECC operations or calculations without an SPE 101s.

RAM 101f may comprise a random access memory for server 101. RAM 101f can be a volatile memory providing rapid read/write memory access to CPU 101e. RAM 101f could be located on a separate integrated circuit in server 101 or located within CPU 101e. The RAM 101f can include data recorded in server 101 for the operation when communicating with a plurality of devices 103 or a key

Petitioner Samsung Ex-1004, 1320

server 102. The system bus 101n may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures including a data bus. System bus 101n connects components within server 101 as illustrated in Figure 1b, such as transferring electrical signals between the components illustrated. Server 101 can include multiple different versions of bus 101n to connect different components, including a first system bus 101n between CPU 101e and RAM 101f (which could be a memory bus), and a second system bus 101n between CPU 101e and WAN interface 101j or LAN interface 101k, which could be an I2C bus, an SPI bus, a PCI bus, or similar data busses.

In exemplar embodiments, RAM 101f operating with server 101 can record values and algorithmic steps or computer instructions for conducting an ECDH key exchange, including a key pair generation step 101x, a secret X1 211a (depicted in Figure 2b below) and also a secret X2 212a (depicted in Figure 2b below). The depicted values and algorithms can be recorded in RAM 101f so that CPU 101e can conduct ECC operations and calculations quickly using the values. The depicted values could also be recorded in other locations for longer-term or nonvolatile storage, such as within a server database 101d. Additional or other values besides the ones depicted in Figure 1b can also be recorded in RAM 101f in order to support server 101 conducting the communications, steps, and message flows depicted in Figure 2a below and other Figures herein.

The operating system (OS) 101g can include Internet protocol stacks such as a User Datagram Protocol (UDP) stack, Transmission Control Protocol (TCP) stack, a domain name system (DNS) stack, a TLS stack, a DPP stack, etc. The operating system 101g may include timers and schedulers for managing the access of software to hardware resources within server 101, where the hardware resources managed by OS 101g can include CPU 101e, RAM 101f, nonvolatile memory 101h, and system bus 101n, and well as connections to the IP network 107 via a WAN interface 101j. The operating system shown of 101g can be appropriate for a higher power computing device with more memory and CPU resources (compared to a device 103). Example operating systems 101g for a server 101 includes Linux or Windows® Server, and other possibilities exist as well. Although depicted as a separate element within server 101 in Figure 1b, OS 101g may reside in RAM 101f and/or nonvolatile memory 101h during operation of server 101.

As depicted in Figure 1b, OS 101g in Figure 1b can contain algorithms, programs, or computer executable instructions (by processor 101e or SPE 101s) for an ECDH key exchange algorithm 220 (depicted and described in Figure 2b and Figure 2e below), a key derivation function (KDF) 216 (depicted and described in Figure 2b and Figure 2e below), and also an ECC point addition operation 214 (depicted and described in Figure 2b and Figure 2e below). The algorithms could be included either (i) within the kernel of OS 101g, or (ii) as a separate program or process loaded by OS 101g and operated by OS 101g. OS 101g can also read and write data to a secure processing environment SPE 101s, if CPU 101e contains SPE 101s.

Petitioner Samsung Ex-1004, 1321

Nonvolatile memory 101h or "storage memory" 101h (which can also be referred to herein as "memory 101h") within server 101 can comprise a non-volatile memory for long-term storage of data, including times when server 101 may be powered off. Memory 101h may be a NAND flash memory or a NOR flash memory and record firmware for server 101, such as a bootloader program and OS 101g. Memory 101h can record long-term and non-volatile storage of data or files for server 101. In an exemplary embodiment, OS 101g is recorded in memory 101h when server 101 is powered off, and portions of memory 101h are moved by CPU 101e into RAM 101f when server 101 powers on. Memory 101h (i) can be integrated with CPU 101e into a single integrated circuit (potentially as a "system on a chip"), or (ii) operate as a separate integrated circuit or a removable card or "disk", such as a solid state drive (SSD). Storage memory 101h can also comprise a plurality of spinning hard disk drives in a redundant array of independent disks (RAID) configuration. Memory 101h may also be referred to as "server storage" and can include exemplary file systems of FAT16, FAT 32, NTFS, ext3, ext4, UDF, or similar file systems. As contemplated herein, the terms "memory 101h", "storage memory 101h", and "nonvolatile memory 101h" can be considered equivalent.

As depicted in Figure 1b, non-volatile memory 101h can record a server database 101d, a device static public key D1 103c, and cryptographic parameters 104. Exemplary data within a server database 101d is depicted and described in connection with Figure 2d below. Although depicted in Figure 1b as recorded within memory 101h, a server database 101d could also operate as a separate server than server 101 in a network 105, and server 101 could query the server database 101d using a private network 107a. The device static public key D1 101c could be received by server 101 from a device manufacturer or a device owner, or directly from device 103 through IP network 107. In addition, as depicted in Figure 1b, memory 101h can record the parameters 104 which were depicted and described in connection with Figure 1a above and also Figure 2d below.

Server 101 can include a WAN interface 101j to communicate with IP network 107 and a plurality of devices 103, as depicted in Figure 1a above (where Figure 1a depicts a single device 103). WAN interface 101j can comprise either a wired connection such as Ethernet or a wireless connection. For wireless configurations of server 101, then WAN interface 101j can comprise a radio, which could connect with an antenna in order to transmit and receive radio frequency signals. For a wireless configuration of server 101, WAN interface 101j within server 101 can provide connectivity to an IP network 107 through 3GPP standards such as 3G, 4G, 4G LTE, and 5G networks, or subsequent and similar standards. In some exemplary embodiments, server 101 can comprise a "g node b" or gNb in a 5G network (or equivalent functionality in 6G or subsequent networks), and WAN interface 101j can comprise a 5G radio access network (RAN) interface. WAN interface 101j can also comprise a wired connection such as digital subscriber line (DSL), coaxial cable connection, or fiber optic connection, and other possibilities exist as well without departing from the scope of the present disclosure.

Petitioner Samsung Ex-1004, 1322

Server 101 may also operate a LAN interface 101k, where LAN interface 101k can be used to connect and communicate with other servers in a network 107, such as key server 102 through private network 107a. LAN interface 101k can comprise a physical interface connected to system bus 101n for server 101. In exemplary embodiments, LAN interface 101k can comprise an Ethernet or fiber optic wired connection. In other words, (i) LAN interface 101k can connect server 101 to private network 107a (which could comprise an IP network with private IP addresses that are not globally routable), and (ii) WAN interface 101j can comprise an interface for communicating with a plurality of devices 103 through insecure networks such as the globally routable public Internet. The use of a separate WAN interface 101j and LAN interface 101k can increase the security of operation for server 101. However, the use of separate physical interfaces for LAN interface 101k and WAN interface 101j can be omitted, and a single physical interface such as Ethernet or fiber-optic could be used by server 101 to communicate with both devices 103 and key server 102.

Server 101 may also optionally include user interface 101m which may include one or more sub-servers for receiving inputs and/or one or more sub-servers for conveying outputs. User interfaces are known in the art and may be simple for many servers 101 such as a few LED lights or and LCD display, and thus user interfaces are not described in detail here. User interface 101m could comprise a touch screen or screen display with keyboard and mouse, if server 101 has sophisticated interaction with a user, such as a network administrator. Server 101 can optionally omit a user interface 101m, if no user input or display is required for establishing communications within a network 105 and/or IP network 107. Although not depicted in Figure 1b, server 101 can include other components to support operation, such as a clock, power source or connection, antennas, etc. Other possibilities exist as well for hardware and electrical components operating in a server 101 without departing from the scope of the present disclosure. Using the electrical components depicted in Figure 1b, a server 101 could send and receive the data 106 in Figure 1a in an encrypted and secure manner after conducting the authenticated ECDHE key exchange as contemplated herein, in order to derive a symmetric ciphering key to encrypt and decrypt messages within data 106 with a plurality of devices 103.

Although not depicted in Figure 1b, devices 103 such as the device 103 depicted in Figure 1a above can include (a) equivalent internal electrical components depicted for a server 101 in order to (b) operate as devices 103. A device 103 in Figure 1a could include a processor similar to CPU 101e, with primary differences for the processor in a device being reduced speed, a smaller memory cache, a smaller number and size of registers, with an exemplary use of 32 bits for datapath widths, integer sizes, and memory address widths, etc., for a device 103. In contrast, an exemplary 64 bit datapaths could be used for CPU 101e in server 101 (although device 103 could also use 64 bit wide datapath widths if device 103 comprises a mobile phone such as a smart phone). For embodiments where device 103 comprises a transducer device for sending and receiving transducer data with a network

Petitioner Samsung Ex-1004, 1323

105, then a CPU in device 103 could comprise an exemplary 32 bit processor, although other possibilities exist as well.

Similarly, RAM in a device 103 could be a RAM similar to RAM 101f in server 101, except the RAM in a device 103 could have fewer memory cells such as supporting exemplary values less than or equal to an exemplary 4 gigabytes, while RAM 103f in server 101 could support more memory cells such as greater than or equal to an exemplary 8 gigabtyes. In exemplary embodiments, the electrical and physical components of a key server can be equivalent to the electrical components for a server 101 in Figure 1b, with different data recorded in RAM 101f for a key server 102, as well as different data recorded in memory 101h for a key server 102. For example, a key server 102 could record the network static private key SK.network 102b in memory 101h, which could include secure disk storage using disk or file encryption.

Figure 1c

Figure 1c is an illustration of exemplary network static public keys recorded by a plurality of devices, in accordance with exemplary embodiments. Figure 1c depicts PKI keys recorded for an exemplary three different devices 103, although a system 100 and other systems herein could operate with potentially millions or more devices 103. The data depicted for each device in Figure 1c can comprise exemplary data for a network public key table 103t for a device 103, which is also depicted and described in connection with Figure 1a above. The exemplary values recorded for network static public keys depicts different embodiments where both (i) a device 103 can record a network static public key PK.network 102a that is shared with other devices 103, and (ii) the network static public key PK.network 102a recorded by device 103 could be unique for device 103 (e.g. not shared with other devices 103 in a system 100 above or a system 200 below, as well as other systems herein). A network public key table 103t for device 103 can record values of a key identity, a network name for network 105, an identity for server 101 comprising ID.server 101i, and also a value for the network static public key PK.network 102a. As depicted in Figure 1c, a device 103 can record multiple different values for use with multiple different networks 105 and/or servers 101.

The first two entries for network static public keys PK.network 102a for a first device 103 (1) and a second device 103 (2) in Figure 1c depicts the same alphanumeric values for basE91 binary to text encoding for an exemplary network static public keys PK.network 102a in a first device 103 (1) and a second device 103 (2), where the key value is depicted for a network 105 of "Network A". Likewise, the second two entries for network static public keys PK.network 102a for a first device 103 (1) and a second device 103 (2) in Figure 1c depicts the same alphanumeric values for basE91 binary to text encoding for an exemplary network static public key PK.network 102a in a first device 103 (1)

Petitioner Samsung Ex-1004, 1324

and a second device 103 (2). Note that although a single value is depicted for PKI keys in a network public key table 103t, the values or numbers for keys recorded could comprise a point on an ECC curve with both an X coordinate and a Y coordinate. For illustration purposes in Figure 1c, only the X coordinate is displayed and the Y coordinate could be calculated from the X coordinate using the equation for an ECC curve in a set of cryptographic parameters 104a for the PKI keys.

The depiction of these keys PK.network 102a illustrates the use of shared keys 102z for a plurality of different devices 103. Although only two devices are depicted with shared keys 102z, many more devices could also record the same shared keys for PK.network 102a. Each of the shared keys 102z is associated with a different network 105, identified with an exemplary different network name. In this manner, a plurality of different devices 103 can record and use the same value for a network static public key PK.network 102a. As described above, the value in a table 103t including network static public key PK.network 102a could be written in device before the device sends the first message 203 in Figure 2a below. The data could be recorded by a device manufacturer, a device distributor, or a device owner, and other possibilities exist as well for the source of PK.network 102a without departing from the scope of the present disclosure.

The same values for shared keys 102z across different devices 103 could be recorded in device 103 during manufacturing or before distribution to end users of device 103. In this manner, devices 103 could be received by end users in a "partially configured" yet secure state, such that a device 103 could use the recorded keys PK.network 102a with a server 101 and/or network 105, where a server 101 does not operate or record the corresponding network static private key SK.network 102b. As depicted and described in connection with Figures 2a, 2b, etc. below, a key server 102 could record and operate with the corresponding network static private key SK.network 102b and thus the key SK.network 102b can remain secured and not distributed out or sent to a server 101. In this manner, encrypted communications for data 106 in Figure 1a can be transferred between device 103 and server 101 without server 101 recording the key SK.network 102b. This increases the security of a system 100 and other systems herein, because server 101 may be exposed to an IP network 107 while key server 102 recording the SK.network 102b can be connected to a private network 107a.

By using a set of shared keys 102z across a plurality of device 103, a key server 102 or a network 105 can control access of the devices 103 as a group. For example, a network 105 could deny access to the private key corresponding to the public key for the first depicted value of PK.network 102a in a first device 103 (1). That action by network 105 would also deny a second device 103 (2) access to the private key corresponding to the public key for the first depicted value of PK.network 102a in the second device 103 (2). In this manner, network 105 could control access to a plurality of different devices 103 by controlling access to a single value of SK.network 102b, where (i) the plurality of different devices 103 record the corresponding PK.network 102a as shared keys 102z. Other benefits for using shared keys 102z can be available as well, such as simplifying manufacturing

Petitioner Samsung Ex-1004, 1325

or distribution, since the same key value for PK.network 102a could be recorded with multiple different devices 103. In other words, a device manufacturer or device distributor would not need to keep track of which value for PK.network 102a belongs with which device 103 for embodiments where shared keys 102z are utilized. However, the use of shared keys 102z for multiple different devices 103 is not required for some exemplary embodiments.

In exemplary embodiments, network static public keys PK.network 102a can also comprise a unique key for each device 103 in a system 100 and other systems herein. Thus, some exemplary embodiments also support the use of a network static public key PK.network 102a that is not shared across multiple different devices 103. For these exemplary embodiments, and as depicted in Figure 1c, a device 103 can record a unique key 102v (depicted as "Per Device or Unique Network Static Public Keys 102v" in Figure 1c). For example, the depicted value for the third key for device 103 (1), (2), and (3) in Figure 1c is shown as unique for each device. A key server 102 could also record the corresponding network static private key SK.network 102b that is unique for each device in a key server database 102d as depicted for unique keys 102v in Figure 1a. In this manner, a network 105 can control access to server 101 and/or network 105 on a per-device basis using the unique key 102v. For example, key server 102 could deny access to device 103 (3) (while continuing to allow service for device 103 (1) and 103 (2)), by denying access or cryptographic operations with the secret key SK.network 102b in a key server 102 corresponding to the public key PK.network 102a recorded by device 103 (3). Other benefits for recording network static public keys PK.network 102a as unique keys 102v for devices 103 exist as well without departing from the scope of the present disclosure.

Although not depicted in Figure 1c, each row or network static public key PK.network 102a could also be stored with a set of cryptographic parameters 104a, such as specifying an ECC named curve associated with the public key 102a. A network 105 or a server ID 101i could be associated with multiple different network static public keys PK.network 102a, where the different keys 102a for the same network 105 or server ID 101i are associated with different parameters 104a. Although depicted as alphanumeric values for the network static public key PK.network 102a, a network public key table 103t could store the public key 102a as separate certificates for the public keys. In addition, a network public key table 103t could store a secure hash value for the network static public key PK.network 102a, where the secure hash algorithm 104d for the secure hash value could be specified by parameters 104, as depicted and described in connection with Figure 2d below. In addition, a table 103t could include a key server identity 102i associated with the network static public key PK.network 102a.

Petitioner Samsung Ex-1004, 1326

Figure 2a

Figure 2a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device, a server, and a key server, in accordance with exemplary embodiments. System 200 can include a device 103, server 101, and a key server 102. Device 103 was depicted and described in connection with Figure 1a, and Figure 1c above. Server 101 and key server 102 were depicted and described in connection with Figure 1a above, and server 101 was depicted and described in connection with Figure 1b above. Server 101 can record and operate a server database 101d, and key server 102 can record and operate a database 102d. Individual steps and components used in system 200 in Figure 2a are also additionally depicted and described in subsequent Figures 2b, 2c, and 2d. Before starting the steps and message flows depicted in Figure 2a, device 103 can securely receive and record a network public key PK.network 102a, which was also depicted and described in connection with Figure 1a and Figure 1c. The corresponding private key for PK.network 102a can be securely recorded in key server 102 within network 105 as SK.network 102b.

For system 200, server 101 and key server 102 may establish a secure session 221, which could comprise establishing a secure communications link between the two servers using protocols such as TLS, IPSec, a virtual private network (VPN), a secure shell (SSH), or similar networking, transport, or application layer technologies in order to establish secure communications between key server 102 and server 101. Secure session 221 can utilize certificates for the two servers in order to provide mutual authentication and mutual key derivation for a symmetric encryption key in secure session 221. Secure session 221 can also be conducted over private network 107a, although the secure session 221 could also be established or conducted through an IP network 107 such as the globally routable Public Internet. Other possibilities exist as well for establishing a secure session 221 between server 101 and key server 102 without departing from the scope of the present disclosure. Although not depicted in Figure 2a, firewalls between server 101 and key server 102 could also be utilized in order to establish or conduct secure session 221. At step 201b, server 101 can begin listening for incoming messages from a device 103 using a physical network interface such as WAN interface 101j that provides connectivity to the IP network 107 and server 101 can use a specific port number such as, but not limited to, TCP port 443 to listen for incoming data 106 from a device 103.

At step 201a, device 103 can be powered on and begin operating, in order to establish connectivity with an IP network 107. At step 202, device 103 can read an address for server 101 from memory or a network public key table 103t, and the address can comprise a DNS name or an IP address for server 101. The DNS name or IP address for server 101 could be recorded or received along with the key PK.network 102a, or device 103 could conduct a DNS query to obtain the address. At step 202, device 103 can also read the set of cryptographic parameters 104 and select a subset of the cryptographic parameters 104a in order to establish communications with server 101. An exemplary

Petitioner Samsung Ex-1004, 1327

subset of cryptographic parameters 104a in a step 202 can comprise a member of the set the cryptographic parameters 104 depicted and described in connection with Figure 2d below (e.g. one line of values in cryptographic parameters 104 in Figure 2d below). In step 202, device 103 can select a subset of cryptographic parameters 104a that is compatible with PK.network 102a. The subset of cryptographic parameters 104a that are compatible with PK.network 102a could also be recorded in nonvolatile memory in device 103 along with network public key PK.network 102a at the time PK.network 102a was recorded or received by device 103.

A step 202 can also comprise device 103 also using a random number generator in order to output a random number 202a for use in subsequent communications with server 101. Although the term "random number" is described herein, a random number could comprise a pseudo random number processed by device 103 using information entropy available to device 103. The random number 202a processed in a step 202 could contain the number of bits specified by a selected subset of cryptographic parameters 104, such as a random length 104g. Random number 202a generated or derived by a device 103 in a step 202 could also comprise a "number used once" (nonce).

Device 103 can then conduct a key pair generation step 103x as depicted and described in connection with Figure 1a above using the selected subset of cryptographic parameters 104a. The parameters 104 could specify a named curve and parameters to derive a device ephemeral private key ed 103b and a device ephemeral public key Ed 103a. The device ephemeral private key ed 103b can comprise a random number generated using a random number generator. The device ephemeral public key Ed 103a could be derived using (i) ECC point multiplication from a base point G for a named curve within cryptographic parameters 104a and (ii) the device ephemeral private key ed 103b. Other possibilities exist as well for the steps a device 103 can use in a key pair generation step 103x without departing from the scope of the present disclosure.

Device 103 can then use (i) the recorded address for server 101 (possibly from a table 103t) and (ii) connectivity to IP network 107 from step 202 to send a message 203 to server 101. Message 203 and other messages contemplated herein can be sent as either TCP or UDP messages, and other possibilities exist as well for the formatting and transfer of messages without departing from the scope of the present disclosure. In exemplary embodiments, device 103 both uses an IP address and port number as a source IP address and port to send message 203 to server 101 and then also the same IP address and port number to listen for responses or messages from server 101. In this manner, device 103 can send a message 203 and receive a response message 206c below through an IP network 107, where intermediate nodes on the IP network 107 may conduct network address translation (NAT) routing.

Message 203 can include the random number random1 202a from a step 202, the device ephemeral public key Ed 103a, and the subset of cryptographic parameters 104a. Message 203 may also optionally include a device identity of ID.device 103i, but the device identity of ID.device 103i

Petitioner Samsung Ex-1004, 1328

can also be omitted from a message 203 in some exemplary embodiments. For embodiments where message 203 optionally excluded device identity ID.device 103i, then an identity for device 103i can optionally be transmitted in later messages. Omitting ID.device 103i from message 203 can increase security for message 203 since an identity for device 103 would not be sent as plaintext in a message 203. Although not depicted in Figure 2a, message 203 could also optionally include an identity for key server 102 comprising ID.key-server 102i, such that server 101 can determine which key server 102i should be associated with message 203. Note that an identity for key server 102 of ID.key-server 102i can be omitted from a message 203, and server 101 can select a key server 102 from other means in a step 205b below.

As depicted in Figure 2a, message 203 could also optionally include a secure hash value 250 (also depicted in Figure 2d below) such as, but not limited to, SHA-256 of the network static public key PK.network 102a. Device 103 can send the hash value 250 of key 102a to server 101 in a message 203, in order for server 101 to identify which of a plurality of possible key servers 102 could be used to process data within message 203, which is further described for a step 205b below. For embodiments where a secure hash value 250 of key 102a is included in a message 203, then the message 203 could optionally exclude the selected subset of cryptographic parameters 104a associated with keys PK.network 102a and Ed 103a. For other embodiments, a key identity for key 102a could be selected by device 103 from a table 103t and the key identity for key 102a could be sent in a message 203 instead of a hash value 205 for key 102a. A server 101 and key server 102 could store the key identity for key 102a and select the key 102a using the key identity for key 102a.

Server 101 receiving the message 203 with the hash value 250 could determine the set of parameters 104a to use for key Ed 103a based on the hash value 250. For example, and as depicted in Figure 2d below, a server database 101d could maintain mapping of hash values 250 and parameters 104a, and server 101 could conduct a query of database 101d using the received hash value 250 in order to select the parameters 104a for further processing and cryptographic operations with key Ed 103a. Or, in an exemplary embodiment cryptographic parameters 104a as transmitted via an IP network 107 or private network 107a could comprise the secure hash 250 of key 102a, where the secure hash 250 of key 102a can specify which subset of a set of cryptographic parameters 104 to utilize for ECC operations (in other words the subset of parameters 104 can comprise parameters 104a). For embodiments where device 103 uses a unique key 102v, then the secure hash value 250 can also comprise a device identity 103i (since the secure hash value 250 would be unique for device 103). Secure hash value 250 could also be omitted from message 203 in some exemplary embodiments.

Server 101 can receive message 203 and begin conducting steps in order to process the message. At step 204, server 101 can read the subset of cryptographic parameters 104a in the message 203 and being using the subset of cryptographic parameters. Or, for embodiments that include hash

Petitioner Samsung Ex-1004, 1329

value 250, then parameters 104a could be omitted from message 203 and server 101 could select the parameters 104a from a server database 101d using the hash value 205, such as with the exemplary server database depicted in Figure 2d below. At step 204, server 101 can comprise a public key validation step on received device ephemeral public key Ed 103a in order to ensure the key is valid and on the selected curve in parameters 104a. Step 204 by server 101 can comprise conducting the steps for an ECC Full Public-Key Validation Routine in section 5.6.2.3.2 of FIPS publication SP 800-56A (revision 2) for the received device ephemeral public key Ed 103a. Alternatively, step 204 can comprise server 101 performing the steps ECC Partial Public-Key Validation Routine in section 5.6.2.3.3 of the same FIPS publication. Other example steps within a public key validation step 204 can comprise (i) verifying the public key is not at the "point of infinity", and (ii) verifying the coordinates of the point for the public key are in the range [0, p-1], where p is the prime defining the finite field. Other possibilities exist as well for evaluating and validating a received public key is cryptographically secure in a public key validation step 204, without departing from the scope of the present disclosure. As contemplated in the present disclosure a device 103, server 101, and key server 102 can conduct a public key validation step 204 each time a public key or a point on an elliptic curve is received.

At step 205a and after a key validation step 204, server 101 can record the data received from the message 203 in a server database 101d. Exemplary values and data for a server database 101d are depicted and described in connection with Figure 2d below. At step 205a, server 101 can record in server database 101d the values of random number 202a, device ephemeral public key Ed 103a, and the subset of cryptographic parameters 104d. For embodiments where device identity ID.device 103i is also received in message 203, then server 101 can also record device identity ID.device 103i in server database 101d. A step 205a can also include (i) storing both Ed 101a and random1 202a in database 101d, and (ii) confirming that Ed 101a and random1 202a are not reused. Security of a system 200 and system 100 and other systems herein can be increased through prohibiting the reuse of ephemeral PKI key pairs and also random numbers. If numbers or keys are reused, then server 101 could respond with a request for device 103 to generate a new ephemeral PKI key pair and/or random number 202a before proceeding to further steps. For embodiments requiring higher security, then hash values for received keys Ed 101a could be stored in a database 101d (instead of the value of key Ed 101a), and a new Ed 101a received by server 101 could be determined as new or reused by calculating a hash value for the received key Ed 101a and comparing with stored values for Ed 101a.

At step 205a, server 103 can also record the originating source IP address and port number 203a (depicted in Figure 2d below) for message 203, in order to subsequently transmit a message 206c below back to the same IP address and port number. In this manner, message 206c below can be routed by intermediate nodes on IP network 107 back to the source IP address and port number used by device 103 to transmit message 203. In other words, (i) the destination IP address and port number

Petitioner Samsung Ex-1004, 1330

of a subsequent message 206c from server 101 to device 103 can comprise the source IP address and port number 203a (depicted in Figure 2d below) received in message 203, and (ii) the source IP address and port number 203a (depicted in Figure 2d below) from message 203 can be recorded in a server database 101d. In this manner, device 103 can be tracked or identified by server 101 during the brief period of time of the message flows in Figure 2a using the source IP address and port number from message 203 for embodiments where device identity ID.device 103i is not included in message 203. A step 205a can also comprise server 101 generating a second random number 205r using parameters 104a for use in subsequent messages with device 103. The first random number can comprise random number random1 202a derived by device 103.

At step 205b, server 101 can select key server 102 for subsequent communications and processing of the received device ephemeral public key Ed 103a. Note that a system 100 could comprise both a plurality of devices 103 and a plurality of key servers 102. In exemplary embodiments server 101 should select in step 205b the proper key server 102 for conducting subsequent steps in Figure 2a. In other words, without data or values from a message 203, server 101 may know which of a possible plurality of key server 102 may record the network static private key SK.network 102b for use with or associated with device ephemeral public key Ed 103a. Server 101 could use one of several possible methods for selecting key server 102 in a step 205b, including a combination of the following embodiments.

A first embodiment for selecting key server 102 in a step 205b could comprise server 101 selecting the same key server 102 for all keys Ed 103a from all devices 103. For example for this first method, server 101 could listen or operate on (i) a specific IP address and port number or (ii) with a specific DNS name or server name indicator (SNI) in step 201b, where the use of (i) or (ii) could be specified or associated with network static public key PK.network 102a. As mentioned above for a step 201a, device 103 can select the address of server 101 using the server address of server 101 recorded with PK.network 102a (possibly from a table 103t in Figure 1c). Server 101 could determine that all messages 203 received using (i) or (ii) are associated with a specific key server 102. For this first embodiment of a step 205b, a plurality of devices 103 could store shared keys 102v for PK.network 102a, as depicted and described in connection with Figure 1c.

A second embodiment of a step 205b for selecting key server 102 of received device ephemeral public key Ed 103a could comprise using an identity of key server 102 in a message 203 from device 103. As described above for a message 203, the message 203 can optionally include an identity for key server 102 comprising ID.key-server 102i. For these embodiments, server 101 can select the key server 102 using the ID.key-server 102i in message 203. A third embodiment for a step 205b of selecting key server 102 for received device ephemeral public key Ed 103a could comprise using an identity of device 103 in a message 203 comprising ID.device 103i. As described above for a message 203, the message 203 can optionally include an identity for device 103, and server 101 using database

Petitioner Samsung Ex-1004, 1331

101d could include a table to map ID.device 103i to key server 102. For this third embodiment of a step 205b, server 101 could conduct a query of server database 101d to select the key server 102 for device 103 using ID.device 103i.

A fourth embodiment for a step 205b to select a key server 102 for received device ephemeral public key Ed 103a could comprise using the subset of cryptographic parameters 104a in a message 203 from device 103. Server 101 could record that a first subset of cryptographic parameters 104a are associated with a first key server 102, and a second subset of cryptographic parameters 104a are associated with a second key server 102, etc. A fifth embodiment for a step 205b to select a key server 102 for received device ephemeral public key Ed 103a could comprise message 205 including a secure hash value 250 (in Figure 2d) of network static public key PK.network 102a, and server 101 with database 103d could include a table to map the secure hash value 250 of PK.network 102a to key server 102. Other possibilities exist as well for server 101 to conduct a step 205b to select a key server 102 using data in a message 203 without departing from the scope of the present disclosure. For embodiments of step 205b, the selection of key server 102 can comprise the selection of an identity for key server 102 of key server identity 102i, and subsequent data such as message 206a could be sent or routed through IP network 107a using the key server identity 102i.

After selecting key server 102 in a step 205b, server 101 can then send key server 102 a message 206a through the secure session 221. Message 206a can include an identity for server 101 comprising ID.server 101i, the received device ephemeral public key Ed 103a, and the subset of cryptographic parameters 104a. For embodiments where device identity ID.device 103i was included in a message 203, then ID.device 103i could be included in a message 206a as well. However, device identity ID.device 103i could be omitted from a message 203 and for these embodiments then message 206a can exclude device identity ID.device 103i as well. Server identity ID.server 103i can be useful for communications between key server 102 and server 101 for a system 100 and system 200, since either (i) server 101 may communicate with a plurality of different key servers 102, and/or (ii) key server 102 may communicate with a plurality of different servers 101.

Server 101 can then conduct a key pair generation step 101x as depicted and described in connection with Figure 1a above using the selected subset of cryptographic parameters 104a. The parameters 104a could specify a named curve and parameters to derive a server ephemeral private key e1 101b and a server ephemeral public key E1 101a. The server ephemeral private key e1 101b can comprise a random number generated using a random number generator. The server ephemeral public key E1 101a could be derived using (i) ECC point multiplication from a base point G for a named curve within cryptographic parameters 104a and (ii) the server ephemeral private key e1 101b. Although message 206a is depicted in Figure 2a as transmitted or sent by server 101 to key server 102 before server 101 derives ephemeral server PKI keys in a step 101x, a message 206a could be sent by server 101 after server 101 conducts the step 101x. Key pair generation step 101x can also confirm

Petitioner Samsung Ex-1004, 1332

that the server ephemeral PKI key pair for server 101 is not reused, such as storing hash values for public keys E1 101a in a database 101d and then comparing the hash value for a new key E1 101a from a step 101x with the stored hash values. If the derived new key E1 101a matches a stored hash value 101a from a database 101d, then the new key E1 101a could be discarded and a different key E1 101a derived.

Key server 102 can receive the message 206a via the secure session 221 and conduct a series of steps to process the message and respond. A first step conducted by key server 102 can comprise a key validation step 204, where the key validation step 204 conducted by key server 102 can be equivalent or compatible with the key validation step 204 conducted by a server 101 as described above. For a key validation step 204, a node can reply with a failure or reject message if the key validation step 204 fails, such as if a received ECC public key fails to fall on the named elliptic curve as specified by a subset of cryptographic parameters 104a.

At step 205c, key server 102 can use data from message 206a in order to select a network static private key SK.network 102b for subsequent steps such as a step 211. For embodiments where message 206a includes either (i) an identity for device 103 such as ID.device 103i, or (ii) identifying information for SK.network 102b for key server 102 to utilize (such as hash 250 of the public key PK.network 102a for SK.network 102b), then key server 102 could use the identifying information in message 206a to select the network static private key SK.network 102b from a key server database 102d, where an exemplary key server database 102d is depicted and described in connection with in Figure 1a above. For some exemplary embodiments, the key server database 102d can record a network static private key SK.network 102b for each set of cryptographic parameters 104a, and subsequently select the key 102b using the parameters 104a received in a message 206a. In other words, an identity for device 103 or hash 250 of PK.network 102a could be omitted, and a key server 102 could use a step 205c to select a network static private key SK.network 102b using a set of cryptographic parameters 104a.

Key server 102 can then conduct an ECDH key exchange step 211 using (i) the selected network static private key SK.network 102b, (ii) the received device ephemeral public key Ed 103a, and (iii) the set of cryptographic parameters 104a. Exemplary details for an ECDH key exchange step 211 are depicted and described in connection with Figure 2b below. The output of an ECDH key exchange step 211 can comprise point X1 211a.

Key server 102 can then send server 101 a message 206b, where the message 206b includes point X1 211a, as well as an identity for key server 102 comprising ID.key-server 102i and cryptographic parameters 104a associated with point X1 211a. Message 206b can be transmitted through secure session 221. If device identity 103i or other identifying information such as hash 250 was included in message 206a, then message 206b could also include device identity 103i or the other identifying information for a device 103. Or, both message 206a and message 206b can include a

Petitioner Samsung Ex-1004, 1333

transaction identity or session identity, such that server 101 can associate the received value X1 211a with a received device ephemeral public key Ed 103a.

Server 101 can receive message 206a with point X1 211a and conduct a series of steps in order to derive a mutually shared and authenticated key exchange with device 103. As contemplated herein, the authentication performed by server 101 can comprise a "one-way" authentication with device 103. Authentication of server 101 or network 105 can be provided by the depicted key exchange with steps 211 and 213, since network 105 from system 100 with both server 101 and key server 102 conducts an ECDH key exchange using at least, in part, the network static private key SK.network 102b. The "one-way" authentication from the ECDH key exchange is also not completed until both sides have successfully used a symmetric ciphering key derived from the ECDH key exchange. In other words, a device that successfully mutually derives a symmetric ciphering key with a server 101 can authenticate that server 101 has secure access to the network static private key SK.network 102b. One benefit of the system depicted in Figure 2a is that the network static private key SK.network 102b does not need to be recorded by or operated with server 101. Further authentication of both parties can be completed via other means including digital signatures in later steps, and the "one-way" authentication in this paragraph refers to the authentication that results from using the ECDH key exchange using at least network static private key SK.network 102b.

Note that the authenticated ECDH key exchange depicted in Figure 2a, with additional details in subsequent Figures, can solve problems in the art discussed in the Description of Related Art. Specifically, through the use of a PK.network 102a recorded by a device and SK.network 102b recorded by a network 105, combined with the use of ephemeral PKI keys for both device 103 and server 101, the depicted and described ECDH key exchange herein can simultaneously achieve both (i) authentication of a network 105 with device 103 and (ii) forward secrecy. As discussed in the Description of Related Art, a device 103 may not have full access to the Internet (such as other servers or networks besides those for a network 105), or other resource limitations such as not storing (x) intermediate certificate authority certificates for servers or (y) compatible parameters or algorithms for intermediate certificate authority certificates for servers, and consequently device 103 may not be able to readily verify a certificate for server 103 such as cert.server 101c without storing and using (x) and (y) above. The mutually authenticated ECDH key exchange with forward secrecy depicted in Figure 2a and subsequent Figures herein supports devices with those limitations. Other benefits are possible as well, such as faster and less resource-intensive authentication of a network 105 with device 103.

After receiving message 206a, server 101 can conduct a point validation step 204a for received value or point X1 211a. Note that point validation step 204a is related to a key validation step 204 and can use several of the same sub-steps depicted and described for a key validation step 204 for server 101 above. A point validation step 204a is different than a key validation step 204 since (i) the

Petitioner Samsung Ex-1004, 1334

value X1 211a is preferably not used as a public key to be shared with other parties, but rather (ii) represents a point on the ECC curve from parameters 104a that will subsequently undergo a point addition operation in order to mutually derive a shared secret with device 103. Further, point X1 211a can be received through a secure session 221 with a trusted party comprising key server 102, and thus the point X1 211a can have a higher level of confidence or trust as being correct and properly formatted than a device ephemeral public key Ed 103a received potentially via the Public Internet. A point validation step 204a for server 101 can comprise verifying that received point X1 211a is on the ECC curve as specified in parameters 104a and that the point is not the "point at infinity". Other possibilities exist as well for conducting a point validation step 204a on the received point X1 211a without departing from the scope of the present disclosure.

After conducting a point validation step 204a, server 101 can then conduct an ECDH key exchange step 212, where a key exchange step 212 is depicted and described in connection with Figure 2b below. In summary, server 101 can input (i) the server derived ephemeral private key e1 101b from a step 101x and (ii) the received device ephemeral public key Ed 103a from message 203 into an ECDH key exchange algorithm 220 (in Figure 2b) in order to calculate a point X2 212a. Server 101 can then conduct a key derivation step 213 as depicted and described in connection with Figure 2b below. In summary, server 101 can conduct an ECC point addition step 214 (in Figure 2b) using both (i) point X1 211a from message 206b and (ii) point X2 212a from step 212 in order to mutually derive a shared secret X3 213a. Shared secret X3 213a can be input into a key derivation function in order to output a symmetric ciphering key K1 216a and also optionally a MAC key.

Server 101 can then conduct a step 207a to create a digital signature 101s, using an elliptic curve digital signature algorithm (ECDSA) over the values of at least, in part, random number random1 202a and random number random2 205r. The ECDSA could use (i) the private key corresponding to the public key in certificate cert.server 101c as (ii) the private key for creating digital signature 101s in a step 207a. The ECDSA can be compatible with IETF RFC 6979, IETF RFC 4574, and also related FIPS standards or other standards for digital signatures using ECC PKI keys. Additional data to sign for signature 101s in a step 207a could comprise the cryptographic parameters 104a and the certificate cert.server 101c. In addition, other digital signature algorithms besides ECDSA could be used in a step 207a such as the use of RSA based digital signature algorithms, or even post-quantum cryptography algorithms. If other digital signature algorithms besides ECDSA are used in a step 207a, then the public key in certificate cert.server 101c and corresponding private key can support the other digital signature algorithms. In general, the digital signature algorithms used to create digital signature 101s can support cryptographic algorithms and PKI keys that are different than the set of cryptographic algorithms 104 in order to conduct a mutually authenticated ECDH key exchange with forward secrecy as contemplated herein.

Petitioner Samsung Ex-1004, 1335

Server 101 can then conduct an encryption step 217 (i) using the key K1 216a output from key derivation step 213 in order to (ii) create a ciphertext1 217b. Exemplary details for an encryption step 217 is depicted and described in connection with Figure 2c below, and an encryption step 217 can use a symmetric ciphering algorithm. The plaintext within ciphertext1 217b can comprise at least, in part, the random number random1 202a and random number random2 205r. Other data could be included in plaintext for ciphertext 217b such as the certificate cert.server 101c, digital signature 101s, as well as parameters 104a, without departing from the scope of the present disclosure. For some exemplary embodiments the use or inclusion of a certificate cert.server 101c and digital signature 101s for plaintext in ciphertext 217b could be omitted, since the mutually derived symmetric ciphering key K1 216a can be derived with authentication of server 101 and network 105 to device 103.

Server 101 can then send device 103 a message 206c, where the destination IP address and port number of message 206c can comprise the source IP address and port number 203a received with message 203 and recorded in server database 101d. Message 206c can include the server ephemeral public key E1 101a and the ciphertext1 217b, as depicted in Figure 2a. The value "K1 216a" depicted in Figure 2a is shown to illustrated that the derived symmetric ciphering key 216a from a key derivation step 213 is used to encrypt ciphertex1 217b (indicated by the brackets shown in Figure 2a for message 206c), and the value K1 216a is not normally transmitted as plaintext or ciphertext in message 206c. Ciphertext1 217b can include plaintext values of random number random1 202a, parameters 104a, certificate cert.server 101c, random number random2 205r, and signature 101s. Other data could be included as plaintext in ciphertext 217b such as extensions for a TLS or DTLS handshake, data supporting an application for device 103, and other possibilities exist as well. As depicted in Figure 2a, the series of steps and messages beginning with step 201a for device 103 though the receipt of message 206c by device 103 can comprise a step 222, where the combined step 222 can be used in additional embodiments depicted below.

As contemplated in the present disclosure, a message such as message 206c and also other messages such as message 203, message 206a, etc. can be transmitted or sent in parts, where the data for the message can be transmitted and received in separate datagrams or portions over time. For these embodiments, the message can comprise the collection of separate datagrams or portions transmitted or sent separately. For example, with separate datagrams or portions for a message 206c in Figure 2a, a first datagram or portion for message 206c could comprise server ephemeral public key E1 101a, which could be sent (i) after a key pair generation step 101x, and (ii) before receiving message 206a from key server 102. A second datagram or portion for message 206c could comprise ciphertext1 217b, which could be sent after server 101 receives message 206a from key server 102. In this manner, by sending message 206c as a first portion and a second portion, the overall speed of conducting a step 223 for device 103 could be increased. For example, by receiving the first portion of message 206c comprising key E1 101a, device 103 could then (a) begin conducting steps below of 204 and 218,

Petitioner Samsung Ex-1004, 1336

while (b) waiting for the second portion of message 206c comprising ciphertext1 217b to be sent separately and after the first portion. By increasing the overall speed for conducting a step 223 for device 103, then electrical power consumption or battery usage for device 103 can be reduced. Other possibilities and benefits from sending a message in the present disclosure as a first portion and a second portion, without departing from the scope of the present disclosure. Messages depicted and described herein may be sent and received as multiple portions over time, where the message can comprise the collection of the multiple portions.

Device 103 can then receive message 206c and conduct a series of steps in order to process the message. Device 103 can conduct a key validation step 204 in order to verify that server ephemeral public key E1 101a in message 206c is properly formatted and is a valid point on the named curve for parameters 104a. Validation step 204 for device 103 can be equivalent to the validation step 204 for server 101 described above. Device 103 can then conduct an ephemeral ECDH (ECDHE) key exchange step 218 in order to mutually derive symmetric ciphering key K1 216a. Details for an ECDHE key exchange step 218 is depicted and described in connection with Figure 2c below. In summary, device 103, using parameters 104a, can perform an elliptic curve point addition operation on (i) the server ephemeral public key E1 101a received in message 206c and (ii) the recorded network static public key PK.network 102a. Device 103 can input (i) the point derived from ECC point addition and (ii) the device ephemeral private key ed 103b into an ECDH key exchange algorithm in order to mutually derive shared secret key X3 215 with server 101. The mutual derivation of shared secret key X3 215 by server 101 is depicted and described in connection with key exchange step 213 for server 101 in Figure 2b below. Device 103 can input shared secret key X3 215 into a key derivation function in order to mutually derive symmetric ciphering key K1 216a. Note that a MAC key could also be derived in step 218.

Device 103 can then perform a decryption step 219 in order to decrypt ciphertext1 217b from message 206c using the derived symmetric ciphering key K1 216a from the key exchange step 218, where symmetric ciphering key K1 216a was derived as described in the paragraph above. A decryption step 219 is also depicted and described in connection with Figure 2c below. Device 103 can then read the plaintext within ciphertext1 217b, as well as verifying message integrity of ciphertext1 217b using a MAC key derived in a step 218. Device 103 in a decryption step 219 can read the plaintext values of random number random1 202a, random number random2 205r, and certificate cert.server 101c, as well as a digital signature 101s. Note that digital signature 101s can be over at least the random number random1 202a that device 103 sent in a message 203.

At step 208, device 103 can conduct a verification step for the plaintext certificate cert.server 101c in order to validate the certificate. Device 103 in a step 208 can verify a signature from a certificate authority for the server static public key PK.server 101p in the certificate (plus any intermediate certificate signatures) using a root certificate for the certificate authority. The root

Petitioner Samsung Ex-1004, 1337

certificate for the certificate authority could be recorded in a nonvolatile memory for device 103. Device 103 can verify both the certificate authority signature in cert.server 101c using an elliptic curve digital signature algorithm (ECDSA). The ECDSA could use a certificate authority public key for from a root certificate for verifying the certificate authority signature in a certificate cert.server 101c. The ECDSA can be compatible with IETF RFC 6979, IETF RFC 4574, and also FIPS 186-4 standards or related and subsequent standards for digital signatures using ECC PKI keys.

Note that a certificate cert.server 101c could also specify parameters different than the use of an ECC algorithm, such as using RSA based signatures. For these embodiments using RSA based keys for digital signatures, device 103 could use a digital signature algorithm (DSA) and server static public key PK.server 101p can comprise an RSA-based key. Note that in some exemplary embodiments, the use of a server certificate cert.server 101c could be omitted, since device 103 can authenticate server 101 using the authenticated ECDH key exchange step 218 (where successful decryption of ciphertext1 217b proves to device 103 that server 101 has access to SK.network 102b). Further, a server certificate cert.server 101c could be included in a message 206c and ciphertext1 217b, but device 103 could omit a separate certificate verification step 208 and still trust the server public key PK.S1 101p in a cert.server 101c. In other words, successful decryption of the cert.server 101c with the symmetric ciphering key K1 216a can signal or indicate that cert.server 101c can be trusted using the stored PK.network 102a, since the cert.server 101c could only be encrypted by a server 101 with access to SK.network 102b.

After a step 208 to verify certificate cert.server 101c, device 103 can conduct a signature verification step 209a to verify signature 101s. For a step 209, device 103 could use the server static public key PK.server 101p for server 101 from certificate cert.server 101c and an ECDSA signature algorithm in order to verify signature 101s. The signed data verified by a signature verification step 209a can comprise at least, in part, both random number random1 202a from device 103 and random number random2 205r from server 101, as well as other data within message 206c such as certificate cert.server 101c. If the signature verification step 209a fails, then device 103 can stop further processing of message 206c and return an error message.

Device 103 can conduct a signature creation step 207b in order to create digital signature 103s over data received in message 206c. The data signed by a signature creation step 207b for signature 103s can comprise at least, in part, random number random2 205r. A set of parameters 104a can specify values and settings to utilize with an ECDSA in a step 209a, such as a secure hash algorithm to utilize, the use of a deterministic ECC signature algorithm (avoiding the need to include a unique random number from device 103 with the signature 103s), padding rules, encoding rules, etc. Device 103 can use device private key d1 101d in order to create signature 103s.

Device 103 can then conduct an encryption step 217c, where encryption step 217c can use the exemplary encryption step 217 depicted and described below in Figure 2c with different plaintext data

Petitioner Samsung Ex-1004, 1338

than the depicted data for a step 217 in Figure 2c. The encryption key for a step 217c can comprise the symmetric ciphering key K1 216a derived by device 103 above in a step 218, and a MAC key 216b (from Figure 2c below) can also be utilized. In some exemplary embodiments, the encryption step 217c can use a different symmetric ciphering key K1 216a than key K1 216a used by server 101 to encrypt ciphertext1 217b. In other words, different symmetric ciphering keys could be used by (i) server 101 to encrypt ciphertext1 217b and (ii) device 103 to encrypt a ciphertext 217d. However, both server 101 and device 103 can mutually derive the different symmetric ciphering keys using at least the mutually derived shared secret X3 215. For some exemplary embodiments, the key K1 216a from a KDF 216 can comprise two portions, where (i) a first portion is used by server 101 to encrypt data and device 103 to decrypt data and (ii) a second portion is used by device 103 to encrypt data and server 101 to decrypt data.

The plaintext data for an encryption step 217c can comprise at least, in part, an identity for device 103 of ID.device 103i, and the random number random2 205r from server 101. Other data could be included in the plaintext for an encryption step 217c without departing from the scope of the present disclosure, such as, but not limited to, data from a transducer connected to device 103. In addition, the device 103 static public key D1 103c, or a certificate for device 103 with public key D1 103c could be included as plaintext data for an encryption step 217c. The output of an encryption step 217c can comprise ciphertext2 217d, as depicted in Figure 2a. As depicted and described in connection with Figure 2c below, the output of an encryption step 217c could also include an initialization vector and a MAC code, which could be included as metadata or plaintext along with ciphertext2 217d in a message 210a. The initialization vector can be used to chain blocks in order to scramble data across the multiple blocks and the MAC code can be used to confirm message integrity using a MAC key output from key exchange algorithm 218. For embodiments where server 101 could store or receive device static public key D1 103c before receiving a message 210a (such as receiving the key D1 103c from a server associated with device 103), then key D1 103c and/or a certificate for device 103 could be omitted from ciphertext2 217d and a message 210a.

After step 217c, device 103 can send server 101 a message 210a, where message 210a can include ciphertext2 217c. In exemplary embodiments, message 210a is transmitted by device 103 using the same source IP address and port number as message 203. In addition, message 210a is transmitted by device 103 using the same destination IP address and port number for server 101 as message 203. Although the signature 103s is depicted in Figure 2a as being internal to ciphertext2 217c, in some exemplary embodiments signature 103s can be external to ciphertext2 217c. Likewise, although a signature 101s is depicted as within a ciphertext 217b from server 101, in some embodiments a signature 101s could be external to ciphertext 217b in a message 206c. Server 101 can receive message 210a by listening to the same local IP address and port number used to receive message 203 above.

Petitioner Samsung Ex-1004, 1339

After server 101 receives message 210a, server 101 can conduct a series of steps in order to process the message. Server 101 can conduct a decryption step 219a, which can comprise a decryption step 219 depicted and described below in connection with Figure 2c, but with different ciphertext data. The ciphertext data for a decryption step 219a can comprise the ciphertext2 217c received by server 101 in message 210a. A decryption step 219a can also use an initialization vector and MAC code received along with ciphertext2 217c in message 210a. After conducting a decryption step 219a, server 101 can read the plaintext data within ciphertext2 217c. In exemplary embodiments, the plaintext data can include an identity for device 103 of ID.device 103i, the device static public key D1 103c, and also the random number random2 205r. Although not depicted in Figure 2a, ciphertext2 217a as received by server 101 can include input from a transducer or sensor operated by device 103, such as, but not limited to, keyboard input, temperature data from a thermocouple or thermistor, pressure data from a transducer, the state of an actuator, the state of an electronic switch, gate, or relay, etc. operated by device 103. Other possibilities exist as well for transducer data in ciphertext2 217a which is decrypted into plaintext by server 101 in a decryption step 219a without departing from the scope of the present disclosure.

At step 210b, server 101 can process the plaintext data output from a decryption step 219a. Server 101 can read and record the device identity ID.device 103i for use in subsequent messages. Server 101 can read the value for random number random2 205r to confirm the value or number equals the random number random2 205r sent above in message 206c. In exemplary embodiments, server 101 can record the plaintext data decrypted from ciphertext2 217c in a server database 101d along with a timestamp, after completing the signature verification step 209c. Server 101 can conduct a signature verification step 209b for signature 103s using the same signature verification algorithm and parameters as signature verification step 209a, except using the device static public key D1 103c. Parameters 104 can specify settings or values for conducting a signature verification step 209a. In exemplary embodiments, signature verification step 209b comprises an ECDSA signature verification for digital signature 103s using key D1 103c. Note that signature 103s is over data that includes at least random number random2 205r sent by server 101 in message 206c. Device static public key D1 103c could be recorded in nonvolatile memory or disk storage of server 101 as depicted in Figure 1b above.

Upon successful completion of a signature verification step 209b for digital signature 103s, server 101 and device 103 can conduct additional steps to securely transfer data 106 between the two nodes. Although not depicted in Figure 2a, server 101 could send device 103 commands, files, configuration data, or other data using ciphertext encrypted with derived symmetric ciphering keys. Server 101 and device 103 could also update key K1 216a or rotate key K1 216a using a key derivation function (such as key derivation function 216 depicted in Figure 2b and Figure 2c below). As depicted in Figure 2a, after a step 210b and a step 209b, server 101 can send key server 102 a message 210b,

Petitioner Samsung Ex-1004, 1340

where message 210b can include the device identity ID.device 103i and an "OK" message, where the "OK" signals to key server 102 that server 101 and device 103 have successfully derived and used symmetric ciphering key 216a using PKI keys and an ECDH point addition of shared secret X1 211a and X2 212a. As depicted in Figure 2a, the series of steps beginning with a step 204 for device 103 through the receipt of message 210b can collectively comprise a step 223.

Figure 2b

Figure 2b is a flow chart illustrating exemplary steps for conducting a key exchange using PKI keys in order to derive shared secrets, and for conducting a key derivation function using the derived shared secrets, in accordance with exemplary embodiments. Key server 102 can conduct a key exchange step 211 in order to derived a secret key X1 211a. Server 101 can conduct a key exchange step 212 in order to derive a secret key X2 212a. Server 101 can receive the secret key X1 211a in a message 206b from key server 102 in Figure 2a above through a secure connection 221. Server 101 can then conduct a key derivation function 213 using shared secrets X1 211a and X2 212a in order to derive a symmetric ciphering key K1 216a. Using the methods and ECC PKI keys described in the present disclosure, a device 103 can also derive the same symmetric ciphering key K1 216a as depicted and described below for a key exchange step 218 in Figure 2c. In other words, for exemplary embodiments (i) the corresponding key exchange step 218 (in Figure 2c below) for a device 103 by network 105 can be (ii) shared or distributed between a server 101 and key server 102 in order to secure or isolate network static private key SK.network 102b.

The processes and operations, described below with respect to all of the logic flow diagrams and flow charts may include the manipulation of signals by a processor and the maintenance of these signals within data structures resident in one or more memory storage devices. For the purposes of this discussion, a process can be generally conceived to be a sequence of computer-executed steps leading to a desired result.

These steps usually require physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, compared, or otherwise manipulated. It is convention for those skilled in the art to refer to representations of these signals as bits, bytes, words, information, elements, symbols, characters, numbers, points, data, entries, objects, images, files, or the like. It should be kept in mind, however, that these and similar terms are associated with appropriate physical quantities for computer operations, and that these terms are merely conventional labels applied to physical quantities that exist within and during operation of the computer.

It should also be understood that manipulations within the computer are often referred to in terms such as listing, creating, adding, calculating, comparing, moving, receiving, determining,

Petitioner Samsung Ex-1004, 1341

configuring, identifying, populating, loading, performing, executing, storing etc. that are often associated with manual operations performed by a human operator. The operations described herein can be machine operations performed in conjunction with various input provided by a human operator or user that interacts with the device, wherein one function of the device can be a computer.

In addition, it should be understood that the programs, processes, methods, etc. described herein are not related or limited to any particular computer or apparatus. Rather, various types of general purpose machines may be used with the following process in accordance with the teachings described herein.

The present invention may comprise a computer program or hardware or a combination thereof which embodies the functions described herein and illustrated in the appended flow charts. However, it should be apparent that there could be many different ways of implementing the invention in computer programming or hardware design, and the invention should not be construed as limited to any one set of computer program instructions.

Further, a skilled programmer would be able to write such a computer program or identify the appropriate hardware circuits to implement the disclosed invention without difficulty based on the flow charts and associated description in the application text, for example. Therefore, disclosure of a particular set of program code instructions or detailed hardware devices is not considered necessary for an adequate understanding of how to make and use the invention. The inventive functionality of the claimed computer implemented processes will be explained in more detail in the following description in conjunction with the remaining Figures illustrating other process flows.

Further, certain steps in the processes or process flow described in all of the logic flow diagrams below must naturally precede others for the present invention to function as described. However, the present invention is not limited to the order of the steps described if such order or sequence does not alter the functionality of the present invention. That is, it is recognized that some steps may be performed before, after, or in parallel other steps without departing from the scope and spirit of the present invention.

The processes, operations, and steps performed by the hardware and software described in this document usually include the manipulation of signals by a CPU or remote server and the maintenance of these signals within data structures resident in one or more of the local or remote memory storage devices. Such data structures impose a physical organization upon the collection of data stored within a memory storage device and represent specific electrical or magnetic elements. These symbolic representations are the means used by those skilled in the art of computer programming and computer construction to most effectively convey teachings and discoveries to others skilled in the art.

A key exchange step 211 for key server 102 to derive a secret key X1 211a can utilize a selected set of cryptographic parameters 104a as depicted and described in connection with Figure 1a and Figure 2a above. As depicted in Figure 2b, an key exchange algorithm 220 in step 211 for key

Petitioner Samsung Ex-1004, 1342

server 102 can receive input both of device ephemeral public key Ed 103a and network static private key SK.network 102b. The key exchange algorithm 220 could comprise a Diffie Hellman key exchange (DH), an Elliptic Curve Diffie Hellman key exchange (ECDH), and other possibilities exist as well without departing from the scope of the present invention. A key exchange algorithm 220 can support either PKI keys based on elliptic curves or RSA algorithms, although support of elliptic curves may be preferred in some exemplary embodiments due to their shorter key lengths and lower computational processing requirements.

A summary of ECDH as a key exchange algorithm 220 is included in the Wikipedia article titled "Elliptic Curve Diffie-Hellman" from March 9, 2018, which is herein incorporated by reference. An exemplary embodiment of key exchange algorithm 220 could comprise a "One-Pass Diffie-Hellman, C(1, 1, ECC CDH)" algorithm as described in section 6.2.2.2 on page 81 of the National Institute of Standards and Technology (NIST) document "NIST SP 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" from March, 2007 which is hereby incorporated by reference its entirety. Other key exchange algorithms in NIST SP 800-56A could be utilized as well for a key exchange algorithm 220 in Figure 2a and Figure 2b without departing from the scope of the present disclosure. Example calculations for an ECDH key exchange for a key exchange algorithm 220 are shown below in Figure 2c.

Other algorithms to derive a secret keys using public keys and private keys may also be utilized in a key exchange algorithm 220, such as, but not limited to, the American National Standards Institute (ANSI) standard X-9.63. Cryptographic parameters 104a can also include information, values, or settings for conducting (i) a key exchange algorithm 220 in step 211 and step 212 and (ii) a key derivation function 216 in order to derive a commonly shared symmetric encryption key K1 216a. As contemplated herein, the terms "selected set of cryptographic parameters 104a" and "cryptographic parameters 104a", and "parameters 104a" can be equivalent, and can also comprise a subset of exemplary cryptographic parameters depicted and described in connection with Figure 1a and Figure 2d below. Parameters 104a input into a key exchange algorithm 220 can include a time-to-live for a key K1 216a that is derived, a supported point formats extension, where the supported point formats extension could comprise uncompressed, compressed prime, or "compressed char2" formats, as specified in ANSI X-9.62. In other words, (i) an ECC keys input into a key exchange algorithm 220 and (ii) secret keys output from key exchange algorithm 220 may have several different formats and a set of parameters 104a can be useful to specify the format. As depicted in Figure 2b, the output of a key exchange algorithm 220 in a step 211, such as an ECDH key exchange, can comprise a secret value X1 211a. In exemplary embodiments, secret value X1 211a can comprise a point on an elliptic curve, where the equation and values for the elliptic curve can be specified in parameters 104a. As contemplated herein, the secret value X1 211a (as well as X2 212a below) comprises both an X coordinate and a Y coordinate, in order to support subsequent ECC point addition operations.

Petitioner Samsung Ex-1004, 1343

Key exchange step 212 for a sever 101 depicted in Figure 2a can correspond to key exchange 212 in Figure 2b. Key exchange step 212 can comprise inputting or using the device ephemeral public key Ed 103a (from message 203 in Figure 2a) and the server ephemeral private key e1 101b (from a key generation step 101x) into a key exchange algorithm 220, which can comprise the same or equivalent key exchange algorithm 220 depicted and described in connection with key exchange step 211 described above. Other elements or algorithms within a key exchange step 212 can be equivalent to a key exchange step 211 above, including the use of shared parameters 104a. The output of a key exchange algorithm 220 in a step 212 can comprise a secret key or value X2 212a. In exemplary embodiments, secret value X2 212a can comprise a point on an elliptic curve, where the equation and values for the elliptic curve can be specified in parameters 104a. Exemplary numeric values for using a key exchange algorithm 220 are depicted and described below, and key exchange algorithm 220 can utilize an ECC point multiplication of a public key by the scalar value of a private key. In exemplary embodiments, a server 101 can record the value X2 212a derived from a step 212 and also the value X1 211a received in a message 206b in a server database 101d. The time the values are stored in a server database 101d can be minimized in order to increase security, and, for example, the recording of the values can be deleted before server 101 sends the "OK" message 210b to key server 102 in Figure 2a.

A key derivation step 213 for server 101 can (i) combine the output of key exchange steps 211 and 212 in order to calculate or derived the shared secret X3 215 and then (ii) perform a key derivation function step 216 on the derived or calculated shared secret X3 215 in order to determine or calculate shared secret key K1 216a, which can comprise a symmetric ciphering key. Note that shared secret key K1 216a can be also mutually derived by device 103, where device 103 uses the key exchange step 218 depicted and described in connection with Figure 2c below. In exemplary embodiments, a server 101 can conduct the key derivation step 213 using (i) the value X1 211a received from key server 102 (where receipt of X1 211a by server 101 can be in a message 206b as shown in Figure 2a above), and (ii) the value or key X2 212a output from a key exchange step 212 for server 101 in the paragraph above. As contemplated herein, the values of X1 211a, X2 212a, and X3 215 may be described as either "shared secrets" or "shared secret keys". Although the values may not normally be used as a key directly with a symmetric ciphering algorithm, these values and the output of a key exchange algorithm 220 can comprise a secret or a key.

Key derivation step 213 for server 101 can comprise two primary steps. A first step in key derivation 213 can comprise an ECC point addition 214 on the value X1 211a and the value X2 212a. The result of the ECC point addition will be equal to the value X3 215. Note that device 103 can also derive the same value for value X3 215 (in step 218 below) without ECC point addition 214 using a step 218. In other words, although (a) the related key exchange step 218 for device 103 may include a point addition for public keys, (b) the key exchange step 218 for device 103 will not use ECC point

Petitioner Samsung Ex-1004, 1344

addition for points derived from two separate private keys in two separate servers (e.g. X1 211a uses private key SK.network 102b and X2 212a uses private key e1 101b).

Exemplary calculations for an ECC point addition 214 can comprise the calculations shown for point addition in the Wikipedia article for "Elliptic Curve Point Multiplication" dated May 15, 2018, which is herein incorporated by reference in its entirety. As depicted in Figure 2b, (a) the calculation of X3 215 by server 101 using an ECC point addition 214 over X1 211a and X2 212a will equal (b) the value for X3 215 calculated by device 103 using a key exchange algorithm 220 in a step 218 from Figure 2c below. A second step in key derivation step 213 as depicted in Figure 2b can comprise a key derivation function step 216 using (a) input from ECC point addition step 214 (e.g. value X3 215 output from step 214), where (b) the output of key derivation function step 216 can comprise key K1 216a and also an associated MAC key 216b. In exemplary embodiments, the X coordinate from shared secret X3 215 can be used with key derivation function 216.

By server 101 conducting a key derivation step 213 as depicted in Figure 2b (where key server 102 conducts the calculations for step 211 using the network static private key SK.network 102b), (i) sever 101 can calculate symmetric ciphering key K1 216a without recording or operating on the network static private key SK.network 102b. In this manner, the security of a system 100 or system 200 can be significantly enhanced, since the network static private key 102b does not need to be recorded or operated by server 101, which can communicate with a plurality of devices 103 over an IP network. In other words, by server 101 (i) using the ECC point addition over key X1 211a instead of (ii) conducting a key exchange 220 directly with SK.network 102b, then server 101 does not need to record or operate with the network static private key SK.network 102b, thereby increasing security. Also, since (i) key X1 211a can be the equivalent of an ECC public key as a point on an elliptic curve, and (ii) it is not computationally feasible to determine network static private key SK.network 102b from key X1 211a, then key X1 211a does not reveal meaningful information about network static private key SK.network 102b.

Many benefits can be achieved by server 101 conducting a key derivation step 213 using key X1 211a instead of recording and operating with network static private key SK.network 102b. As one example, the corresponding network static public key PK.network 102a could potentially be both (i) recorded in millions of distributed devices connecting to server 101 through many different physical locations and networks, and (ii) used for a decade or longer. Keeping network static private key SK.network 102b secure for this embodiment could be economically essential, since a compromise of network static private key SK.network 102b may (i) render the devices 103 insecure (or unable to authenticate network 105 using an ECDHE key exchange), and (ii) require the secure distribution or re-installation of a new, different network static public key SK.network 102a in the devices, which may not be economically feasible due to the prior distribution of devices.

Petitioner Samsung Ex-1004, 1345

Exemplary data and numbers can be provided to demonstrate the calculations for (i) key exchange step 211, (ii) key exchange step 212, and (iii) key derivation step 213 using an ECC point addition 214. The exemplary data can comprise decimal numbers for the example ECC PK1 keys and exchanged keys listed in "Test vectors for DPP Authentication using P-256 for mutual authentication" on pages 88 and 89 of the DPP specification version 1.0. Parameters 104a can comprise the elliptic curve of "secp256r1" with key lengths of 256 bit long keys.

The network static private key SK.network 102b can comprise the exemplary following number, and can be recorded in key server 102:

38358416135251014160802731750427376395128366423455574545250035236739593908128

The server ephemeral private key e1 101b can comprise the exemplary following number, and can be recorded by server 101:

111991471310604289774359152687306247761778388605764559848869154712980108827301

The device ephemeral public key Ed 103a can comprise the following exemplary values with X and Y numbers (or "coordinates") of:

X:

618316885049238173674842721030568484577216011069879115485152191196611409919 66

Y:

436821274116052626307636850969789027573720854595612820926922498255090826944

Key exchange step 211 for an ECDH algorithm key exchange 220 by key server 102 can input the device ephemeral public key Ed 103a and the network static private key SK.network 102b (both with numbers above) in order to calculate a secret X1 211a. An exemplary number or value for secret X1 211a from the values above using parameters 104a can be:

X:

114900471986805225153115909625996714820294170643513373033139066428057435731 19

Y:

279339665602382047312450979433990845238094818334347544097236049703660820218 55

Key exchange step 212 for an ECDH algorithm key exchange 220 by server 101 can input the device ephemeral public key Ed 103a and the server ephemeral private key e1 101b (both with

Petitioner Samsung Ex-1004, 1346

numbers above) in order to calculate a secret X2 212a. An exemplary number or value for key X2 212a from the values above using parameters 104a can be:

X:
78944719651774206698250588701582570633503182903415394243006529481189158194650
Y:
112277127029246845818349358288374891402018204245360629120510863824589445237

An ECC point addition 213 for the above two derived points (or "keys") X1 211a (from keys Ed 103a and SK.network 102b) and X2 212a (from keys Ed 103a and e1 101b) will result in the following point that also equals X3 215.

X:
113734500629065545557893524064610113740858966831672649615565042035695230713090
Y:
689614296913074291667967608810956893480888757713349706445933063883757419652 62

Note that the same numeric value for key X3 215 can also be derived by device 103 from a key exchange step 218 below using ECDH key exchange algorithm 220a. For exemplary embodiments, although private key SK.network 102b and ephemeral private key e1 101b are recorded and operated by physically separated devices, device 101 can record and operate on the corresponding public keys PK.network 102a and ephemeral public key E1 101a (at the same physical location as device 103).

After an ECC point addition 213, for a key derivation step 218 in Figure 2b, server 101 can input the shared secret key X3 215, where key X3 215 was output from the ECC point addition 214, into a key derivation function 216. The key derivation function 216 can comprise the same key derivation function 216 used by a device 103 in a step 218 below. The output of a key derivation function 216 can comprise both (i) a symmetric ciphering key K1 216a and (ii) a MAC key 216b. MAC key 216b can be used with a symmetric ciphering algorithm in order to generate a MAC code, such that the other party using the same key K1 216a and MAC key 216b can process the ciphertext and calculate the same MAC code in order to verify message integrity.

Key derivation function 216 can use a secure hash function such as, but not limited to, SHA-256, SHA-384, SHA-3, etc. and additional values such as a text string with secret X3 215. The specification of a secure hash algorithm and the text string for use with a key derivation function 216 could be commonly shared between server 101 and device 103 by commonly shared parameters 104a. In some exemplary embodiments, the text string for use with secret X3 215 can be from data, text, or

Petitioner Samsung Ex-1004, 1347

values transmitted in (i) message 203 (for KDF 216 by server 101 in step 213) and/or (ii) message 206c (for KDF 216 by device 103 in step 218). The output of a secure hash algorithm within a key derivation function 216 could have a subset of bits selected or possibly a secure hash expanded in order to obtain the number of bits required for a symmetric key with a symmetric ciphering algorithm, such as key K1 216a. A key derivation function (KDF) 216 could comprise a KDF compatible with or specified by ANSI standards for "X9.63 Key Derivation Function". Other possibilities exist for a key derivation function 216 to convert a secret X3 215 into a symmetric ciphering key K1 216a and a MAC key 216b without departing from the scope of the present disclosure. As contemplated in the present disclosure, although an ECC public key such as secret X3 215 can comprise a coordinate with an X value and a Y value, in exemplary embodiments a single number comprising the X value can be selected and input into a key derivation function 216. In addition, the key K1 216a can comprise two portions, where (i) a first portion can be a key for encrypting data by server 101 and decrypting the data by device 103 and (ii) a second portion can be a key for encrypting data by device 102 and decrypting the data by server 101.

Figure 2c

Figure 2c is a flow chart illustrating exemplary steps for conducting a key exchange using PKI keys in order to derive a shared secret key, and for using the derived shared secret key to encrypt and decrypt data, in accordance with exemplary embodiments. Exemplary steps for a device 103 to mutually derive a shared secret X3 215 and symmetric key 216a can comprise a key exchange step 218. Exemplary steps in Figure 2c for a server 101 to encrypt plaintext data using the mutually derived symmetric key 216a can comprise an encryption step 217. Exemplary steps in Figure 2c for a device 103 to decrypt ciphertext data using the mutually derived symmetric key 216a can comprise a decryption step 219. The use of the steps for a key exchange 218, encryption 217, and decryption 219 were also depicted and described in connection with Figure 2a above. Note that steps in Figure 2c and the steps in Figure 2b can share some algorithms and values, and the descriptions for the algorithms and values in Figure 2ab can be applicable for Figure 2c. For example, the key exchange algorithm 220a can comprise an ECDH key exchange equivalent to key exchange step 220. The set of parameters 104a depicted and described in Figure 2b can also be used in Figure 2c.

A device 103 can conduct a key exchange step 218. At step 218, (i) a combination of a recorded network static public key PK.network 102a and received server ephemeral public key E1 101a, and (ii) the derived device ephemeral private key ed 103b can be input into an ECDH key exchange algorithm 220a in order to calculate the shared secret X3 215. The recorded network static public key PK.network 102a and received server ephemeral public key E1 101a can be combined via elliptic curve point addition. Exemplary data and numbers can be provided to demonstrate the

Petitioner Samsung Ex-1004, 1348

calculations for (i) key exchange step 218. The exemplary data can comprise decimal numbers for the example ECC PKI keys and exchanged keys listed in "Test vectors for DPP Authentication using P-256 for mutual authentication" on pages 88 and 89 of the DPP specification version 1.0. Parameters 104a can comprise the elliptic of "secp256r1" with key lengths of 256 bit long keys.

The device ephemeral private key ed 103b can comprise the exemplary following number, and can be recorded in device 103 after a key pair generation step 103x from Figure 1a above:

98142297182445185535509586920610248294802812794507930861676847471456420049 23

The network static public key PK.network 102a can comprise the exemplary values with X and Y numbers (or "coordinates") of:

X:

44198070003813586561115061476516229802700291105541193294933359539128224522 87

Y:

37427159939572325965354914097696269740713866333885143374269952770772578794844

The server ephemeral public key E1 101a can comprise the following exemplary values with X and Y numbers (or "coordinates") of:

X:

42629956901026513598149966301519681371972968598637962756879877886841583606416

Y:

20486612594265388212565154850034967164732043090221075006612427172869133074917

An ECC point addition for the above two keys E1 101a and PK.network 102a will result in the following exemplary point. which comprises (a) both E1 101a and PK.network 102a for a key exchange step 218 then (b) input into an ECDH key exchange algorithm 220a:

X:

28114613657326475531346375416853531696489050589415231447375990921521198001 80

Y:

93903335977032690345879985966890561591048675256101157964834025539587687968435

The above combination of both E1 101a and PK.network 102a for a key exchange step 218 via an ECC point addition operation is depicted in Figure 2c with the "+" symbol between the public keys.

Petitioner Samsung Ex-1004, 1349

The output of the above ECC point addition for public keys E1 101a and PK.network 102a can be input into ECDH key exchange algorithm 220a using parameters 104a. All of the exemplary calculations for a key exchange step 218 can use the exemplary subset of cryptographic parameters 104a. An ECDH algorithm key exchange 220a in key exchange step 218 can input (i) the exemplary point immediately above from the ECC point addition operation on the public keys 101a and 102a and (ii) the device ephemeral private key ed 103b into the ECDH key exchange 220a, and output the point X3 215. Note that the secret X3 215 as derived by device 103 in a key exchange step 218 equals or is the same numeric value as the secret X3 215 derived by server 101 in a key derivation step 213 in Figure 2b. An exemplary number or value for secret X3 215 calculated by device 103 using a key exchange step 218 using the above exemplary numeric values for ed 103b, PK.network 102a, and E1 101a would be:

X:
113734500629065545557893524064610113740858966831672649615565042035695230713090

Y:
689614296913074291667967608810956893480888757713349706445933063883757419652 62

Although Figure 2c depicts an ECC point addition operation over public keys E1 101a and PK.network 102a, the same shared secret value X3 215 could be generated or derived by conducting (i) a first ECC point multiplication operation with the server ephemeral public key E1 101a and the device ephemeral private key ed 103b to derive a first point, and (ii) a second ECC point multiplication operation with the network ephemeral public key PK.network 102a and the device ephemeral private key ed 103b to derive a second point, and (iii) an ECC point addition operation with the first point and the second point to derive the shared secret value X3 215. In other words, the value X3 215 can be calculated as either:

(i) X3 215 = [E1 101a + PK.network 102a] * ed 103b, or

(ii) X3 215 = [E1 101a * ed 103b] + [PK.network 102a * ed 103b]

For a key derivation step 218, derived shared secret key X3 215 can be input into a key derivation function 216 where the key derivation function 216 can be equivalent to the key derivation function 216 depicted and described in connection with Figure 2b above for a key derivation step 213. Note that for key derivation steps in the present disclosure, the X coordinate of a derived shared secret can be taken or used as input into the key derivation function. The output of a key derivation function 216 can comprise both (i) a symmetric ciphering key K1 216a and (ii) a MAC key 216b. MAC key 216b can be used with a symmetric ciphering algorithm in order to generate a MAC code, such that the other party using the same key K1 216a and MAC key 216b can process the ciphertext and

Petitioner Samsung Ex-1004, 1350

calculate the same MAC code in order to verify message integrity. The use of key K1 216a and MAC key 216b are described in connection with encryption step 217 and decryption step 219.

Server 101 can conduct an encryption step 217, where the use for an encryption step 217 is depicted and described in connection with Figure 2a above. Plaintext 217a in a step 217 can comprise the first random number random1 202a from device 103, the second random number random2 205r, and the server certificate cert.server 101c. Other or different exemplary data could be included as plaintext 217a in an encryption step 217, including extensions for a TLS or DLTS handshake. The symmetric ciphering key for encryption step 217 can comprise symmetric key K1 216a from a key derivation step 213 and a MAC key 216b can be input into a symmetric ciphering algorithm 225 as well. Encryption step 217 and decryption step 219 can use a common symmetric ciphering algorithm 225, which could comprise the Advanced Encryption Standard with Synthetic Initialization Vectors (AES-SIV) (and deciphering algorithm) also with a common set of symmetric ciphering parameters 104f from a set of cryptographic parameters 104. Other or different symmetric ciphering algorithms 225 could be utilized as well, such as, but not limited to such as AES, Triple Data Encryption Standard (3DES), Blowfish, or related algorithms. A mutually derived symmetric ciphering key K1 216a can comprise two portions, where a first portion is used by server 101 for encryption and a second portion is used by device 103 for encryption. At least the first portion of key K1 216a can be used in an encryption step 217.

Symmetric ciphering parameters 104f can also specify the use of a block chaining mode such as cipher block chaining (CBC), counter mode (CTR), or Galois/Counter mode (GCM) and other possibilities exist as well. In addition, symmetric ciphering parameters 104f could specify a mode for message authentication, which could comprise a CMAC mode as specified in NIST publication SP-800-38B. In some exemplary embodiments, a symmetric ciphering algorithm 225 can comprise the AES-SIV algorithm as specified in IETF RFC 5297. The output from an encryption step 217 using a symmetric ciphering algorithm 225 and the depicted values input can be ciphertext 217b, as depicted in Figure 2c.

A decryption step 219 can be performed by device 103. A decryption 219 step converts the ciphertext 217b received in a message 206c from Figure 2a into plaintext 217a. Decryption step 219 can utilize a symmetric decryption algorithm 225, which could comprise the same algorithm used in symmetric encryption algorithm 225 except the algorithm being used for decryption instead of encryption. Note that the same values are input into symmetric decryption algorithm 225 as symmetric encryption algorithm 225 above, such as symmetric encryption key K1 216a (or the first portion of key K1 216a if a second portion of key K1 216a is used by device 103 for encryption) and parameters 104f in order to convert ciphertext 217b back into plaintext 217a. Additional data input into symmetric decryption algorithm 211b can comprise an initialization vector 217i and MAC code 216c which could be sent along with ciphertext 217b.

Petitioner Samsung Ex-1004, 1351

Device 103 can the read and process plaintext 217a after a decryption 219 step. The plaintext 217a as read by device 103 can comprise the first random number random1 202a from device 103, the second random number random2 205r, and the server certificate cert.server 101c. In exemplary embodiments, the successful decryption of a ciphertext into a plaintext using decryption algorithm 225 supports one-way authentication of the server 101 and/or network 105, since successful decryption by device 103 can only take place when the server 101 has access to network static private key SK.network 102b. In other words, only the nodes could mutually derive key K1 216a in Figure 2b and Figure 2c by (i) device 103 recording PK.network 102a and (ii) server 101 having access to SK.network 102b (via key server 102). Thus, data that is successfully encrypted by the server 101 and decrypted by the device 103 using key K1 216a would confirm the server 101 is authenticated.

As depicted and described in connection with Figure 2a, server 101 or device 103 can also conduct both an encryption step 217 and a decryption step 219. The steps for server 101 to conduct a decryption step 219 for can comprise step 219a as depicted and described in Figure 2a. When server 101 conducts decryption step 219a using symmetric encryption key K1 216a, the ciphertext and plaintext will comprise different values than those depicted in Figure 2c, where the ciphertext for a decryption step 219a can comprise ciphertext2 217d. Further, a device 103 can conduct an encryption step 217c in with key K1 216a in order to create ciphertext 217d, as depicted in Figure 2a.

Figure 2d

Figure 2d is an illustration of an exemplary server database and an exemplary set of cryptographic parameters, in accordance with exemplary embodiments. A server database 101d depicted and described above in connection with system 100 and system 200 can record data for server 101 to work with a plurality of devices 103 and at least one key server 102. A server database 101d could record in at least one set of values, keys, and/or numbers for a plurality of devices 103. Other possibilities exist as well for the organization, tables, and recorded data within a server database 101d without departing from the scope of the present disclosure. Data within server database 101d could be encrypted using a symmetric key. Although system 100 and system 200 depict a server database 101d as operating or recorded within a server 101, a server database 101d could comprise a separate server within a network 105 and communicating with server 101 via a secure session 221 or a private network 107a. Further, a server database 101d, when operating or recorded in a separate server than server 101, then server database 101d could contain electrical components equivalent to a server 101 depicted and described in connection with Figure 1b.

Server database 101d can record values or numbers for a first random number random1 202a, received device ephemeral public key Ed 103a, a selected set of cryptographic parameters 104a, a source IP address and port number 203a received for message 203, a secure hash value over

Petitioner Samsung Ex-1004, 1352

PK.network 102a comprising H(PK.network 102a) 250, and identity for key server 102 comprising ID.key-server 102i, an ECC point value X1 211a, a server ephemeral public key E1 101a, a server ephemeral private key e1 101b, an ECC point value X2 212a, an ECC point value X3 215, a derived symmetric ciphering key K1 216a, and a second random number random2 205r. In exemplary embodiments, the values depicted in the first row of server database 101d could comprise data recorded by a server 101 while conducting the series of steps for a step 222 and step 223 depicted and described in connection with Figure 2a above with a first device 103. The values depicted in the second row of server database 101d could comprise data recorded by a server 101 while conducting the series of steps for a step 222 and step 223 depicted and described in connection with Figure 2a above with a second device 103, etc.

In exemplary embodiments for a server database 101d, a first device 103 could send server 101 a first value for device ephemeral public key Ed 103a, and the first value is depicted in Figure 2d as "103a-1". Since server 101 could communicate with a plurality of devices 103, the second row in the depicted server database 101d could comprise data for the equivalent steps conducted with a second device 103, such as recording a second value for device ephemeral public key Ed 103a for the second device. The second value for device ephemeral public key Ed 103a with the second device 103 is depicted in Figure 2d as "103a-2". Equivalent notations for other keys or values are applicable as well, such as server database 101d recording a first secret X1 211a depicted as "211a-1" for a first device 103, and then recording a second secret X1 211a depicted as "211a-2". Thus, as depicted a server database 101d can record and operate with a plurality of different values for a key, where each are utilized by a different device. Although not depicted in Figure 2d, a server database could record device identity ID.device 103i as well. For embodiments where a device identity 103i is not available, then server 101 could keep track of different devices 103 for conducting the steps in Figure 2a by the source IP:port number 203a.

In some exemplary embodiments, a message 203 can include a secure hash value H(PK.network 102a) 250, as described for a message 203 in Figure 2a above. The receipt of a secure hash value H(PK.network 102a) 250 could be mapped to or associated with a key server 102 via a key server identity ID.key-server 102i, where the mapping of H 250 to ID.key-server 102i could be recorded in a server database 101d before device 103 sends a message 203. For these embodiments and after receipt of message 203, server 101 could conduct a query of server database 101d using the received H 250 in a message 203 in order to select a key server 102 with ID.key-server 102i in order to send the message 206a to key server 102. In this manner, server 101 can communicate with a plurality of different key servers 102, and the destination of a message 206a (or key server 102) can be selected by the value H 250 received in a message 203. In other words, for a plurality of different devices 103 communicating with a server 101, a first subset of devices 103 could record and use a first network static public key PK.network 102a, and a second subset of devices 103 could record and use

Petitioner Samsung Ex-1004, 1353

a second network static public key PK.network 102a. By receiving a value or identifier of the first or second key 102a in message 203 (such as H(PK.network 102a) 250), server 101 could use the data depicted for a server database 101d to select or identify the correct key server 102 in order to (i) send a message 206a and (ii) receive the correct secret X1 211a for the device 103 using a particular PK.network 102a.

Although the value H(PK.network 102a) 250 is depicted as recorded in a server database 101d in Figure 2d, a different value or identifier for the PK.network 102a could be recorded and utilized as well. In an exemplary embodiment, server 101 could receive the plaintext PK.network 102a in a message 203 and record the plaintext PK.network 102a in a server database 101d (instead of a hash value H 250). In another exemplary embodiment, an identity for key server 102 (such as ID.key-server 102i) could be selected or determined by server 101 using the selected set of cryptographic parameters 104a received in message 203 and recorded in a database 101d. For these embodiments, a first selected set of cryptographic parameters 104a could be associated with a first key server 102 (and first ID.key-server 102i) and a second set of cryptographic parameters 104a could be associated with a second key server 102 (and second ID.key-server 102i). Other possibilities exist as well for a server database 101d to record data in order to select a key server 102 for sending message 206a with device ephemeral public key Ed 103a based on data received in message 203, without departing from the scope of the present disclosure. As one example, the identity for key server 102 of ID.key-server 102i could be included in message 203 and the value for ID.key-server 102i could be recorded in a server database 101d by server 101.

In a server database 101d, although separate values are depicted for some data, such as values "102i-1" and "102i-2" for identities of key servers 102, some of the exemplary values can comprise identical strings or numbers. For example, data for two different devices 103 in a server database 101d could record the same name or value of "102i-2" for a single key server 102 to be associated with the two different devices 103. Likewise, two different devices 103 could share the same network static public key PK.network 102a, and thus H 250 can be the same value of an exemplary "250-2" for two different devices 103. A server database 101d could also record additional data and values than those depicted in Figure 2d for some exemplary embodiments. For example, server database 101d could record timestamps for when messages are transmitted or received, such that stale or data older than a specified range could be purged. Server database 101d could also record data received from device 103 in a message 210a, which could include data from a transducer operated by device 103.

Some data within a server database 101d could be recorded and operated on separately by server 101, such as server 101 not recording secrets such as X1 211a or X2 212a, etc. in a database 101d, but rather server 101 could record the values in volatile memory 101f of server 101. In exemplary embodiments, server database 101d could also operate in a distributed or "cloud"

Petitioner Samsung Ex-1004, 1354

configurations such that multiple different servers 101 could query and record data in server database 101d, where data for server database 101d is recorded in multiple, physically separated servers.

Cryptographic parameters 104 can specify sets of cryptographic parameters that are supported by server 101 in order to process message 203 and send response message 206c from Figure 2a. Cryptographic parameters 104 can be recorded in a server database 101d, or in other locations within a system 100 and system 200. As depicted in Figure 1a, each of device 103, server 101, and key server 102 can record and operate with a set of cryptographic parameters 104. Cryptographic parameters 104 can record a collection of cryptographic algorithms or specifications such as a set identifier 104a, a key length 104b, an ECC curve name 104c, a hash algorithm 104d, symmetric ciphering key length 104e, settings for a symmetric ciphering algorithm 104f, and a random number length 104g.

As contemplated herein, when a selected set of cryptographic parameters such as using the words or description "parameters 104a" or "cryptographic parameters 104a" can specify a row of parameters or values in a set of cryptographic parameters 104, such that the collection of values in the row can be used with key pair generation functions 101x and 103x, ECDH key exchange 220, and other cryptographic operations and steps as contemplated herein. Set identifier 104a can be an identity for a row or set of values for cryptographic parameters 104. For example, set "A" can comprise cryptographic suite 1 as specified in section 3.2.3 of DPP specification version 1.0. Key length 104b can be the length of keys in bits for PKI keys used in system 100 and system 200. ECC Curve name 104c can be a name for an ECC curve used with PKI keys and key exchange algorithms in system 100 and system 200.

Hash algorithm 104d in cryptographic parameters 104 can be the name of a secure hash algorithm, such as the exemplary SHA-256 algorithm depicted, which may also be referred to as "SHA-2". Hash algorithm 104d can also be used in a key derivation function (e.g. KDF 216 above in Figure 2b and Figure 2c) and also with digital signature steps 207a and 209a. Settings for a symmetric ciphering algorithm 104f can specify the identity or name of a symmetric ciphering algorithm 225 such as "AES", "AES-SIV", 3DES, Blowfish, etc. Random length 104g can specify the length in bits for random numbers or "nonces" generated by both device 103 and server 101, where the nonces can be used to prevent replay attacks and require messages transmitted and received to be unique. Other possibilities exist as well for data within cryptographic parameters 104, such as the specification of point compression, encoding rules such as distinguished encoding rules (DER), ASN or CSN syntax notation, padding rules, etc.

Figure 2e

Figure 2e is a flow chart illustrating exemplary steps for conducting a key exchange using PKI keys in order to derive a shared secret key using ECC point multiplication, in accordance with

Petitioner Samsung Ex-1004, 1355

exemplary embodiments. An ECDH key exchange step 218n can be conducted by a device 103, and use the steps for an ECDH key exchange step 218, with the additional steps of conducting an ECC point multiplication using numbers N1 298 and N2 299. A key derivation step 213n can be conducted by a server 101, and use the steps for a key derivation step 213, with the additional steps of conducting an ECC point multiplication using the same numbers N1 298 and N2 299. In other words, (i) ECDH key exchange step 218 can comprise the depicted ECDH key exchange step 218n where the numbers for N1 298 and N2 299 are equal to the value of "1", and (ii) key derivation step 213 can comprise the depicted key derivation step 213n where the numbers for N1 298 and N2 299 are also equal to the value of "1". In some exemplary embodiments, (i) an ECDH key exchange step 218 depicted and described in connection with Figure 2a for device 103 can comprise the ECDH key exchange step 218n with point multiplication, and key derivation step 213 for server 101 can comprise the key derivation step 213n with point multiplication. The set of parameters 104a from figures above, such as with Figure 2a, can be used with both ECDH key exchange step 218n and key derivation step 213n.

A device 103 can conduct a key exchange step 218n. At step 218n, a device 103 can conduct a first ECDH key exchange step 220 and a second ECDH key exchange step 220. For a step 218n, a first ECDH key exchange step 220 can be conducted by device 103 with (i) the server ephemeral public key E1 101a received in a message 206c from Figure 2a and (ii) the recorded device ephemeral private key ed 103b, and the resulting point multiplied by the number N1 298. Note that the ECC point resulting from the first ECDH key exchange 220 in the previous sentence will also equal the point X2 212a multiplied by the number N1 298, where the calculation of point X2 212a is depicted and described in connection with a key exchange step 212 in Figure 2b.

Continuing with step 218n, a device 103 can conduct the second ECDH key exchange step 220 with (i) the network static public key PK.network 102a recorded in device 103 and (ii) the recorded device ephemeral private key ed 103b, and the resulting point multipled by the number N2 299. Note that the ECC point resulting from the second ECDH key exchange 220 in the previous sentence will also be equal to the point X1 211a multiplied by the number N2 299, where the calculation of point X1 211a is depicted and described in connection with key exchange step 211 in Figure 2b.

Continuing with step 218n, a device 103 can conduct an ECC point addition operation on the two points resulting from (i) the first ECDH key exchange step 220 multiplied by N1 298 and (ii) the second ECDH key exchange step multiplied by N2 299. In other words, a device 103 can conduct an ECDH point addition operation with (i) the value X2 212a multiplied by N1 298 and (ii) the value X1 211a multiplied by the value N2 299, in order to derive a secret X3' 215a that is mutually shared with server 101.

Exemplary data and numbers can be provided to demonstrate the calculations for (i) key exchange step 218n and (ii) key derivation step 213n. The exemplary data can comprise decimal numbers for the example ECC PKI keys and exchanged keys described above in Figure 2b. The first

Petitioner Samsung Ex-1004, 1356

ECDH key exchange 220 for device 103 using (i) the exemplary numerical value for device ephemeral private key ed 103b in Figure 2c and (ii) the exemplary numerical value for server ephemeral public key E1 101a in Figure 2c, using parameters 104a, will result in the exemplary number or value for secret X1 211a, where parameters 104a can comprise the elliptic curve of "secp256r1" with key lengths of 256 bit long keys:

X:
114900471986805225153115909625996714820294170643513373033139066428057435731 19
Y:
279339665602382047312450979433990845238094818334347544097236049703660820218 55

For an exemplary value of "3" for N1 298, the resulting ECC point multiplication of X1 211a by N1 298 with the value of "3" will result in the following point "3 x X1":

X:
607427538132779561340867228013871340157492336492288842361876516538141762255 36
Y:
586113352884631322682758701748943371458887868634413506837084431769263282989 69

The second ECDH key exchange 220 for device 103 in a step 218n using (i) the exemplary numerical value for device ephemeral private key ed 103b in Figure 2c and (ii) the exemplary numerical value for network static public key PK.network 102a in Figure 2c, using parameters 104a, will result in the exemplary number or value for secret X2 212a:

X:
789447196517742066982505887015825706335031829034153942430065294811891581946 50
Y:
112277127029246845818349358288374891402018204245360629120510863823245894452 37

For an exemplary value of "7" for N2 299, the resulting ECC point multiplication of X2 212a by N2 299 with the value of "7" will result in the following point "7 x X2":

X:
978720966385822157273046423892267022085755948504731360759940073372408675565 63
Y:
309011137620506296286117894127593905256160030790408724299409977798545007282 55

Petitioner Samsung Ex-1004, 1357

An ECC point addition for the two points "3 x X1" and "7 x X2" will result in the following point, which can equal the shared secret X3' 215a for a key exchange step 218n:

X:
1074603086866211116849007956196958747011322587763881216882979583258134105077 48
Y:
1047970399126449198109988535123604349303368671413820171654965147986947554899 00

The above values for N1 298 and N2 299 are exemplary, and any numeric value less than the large prime number p for a named elliptic curve could be selected for both N1 298 and N2 299.

Continuing with step 218n, derived shared secret key X3' 215a can be input into a key derivation function 216 where the key derivation function 216 can be equivalent to the key derivation function 216 depicted and described in connection with Figure 2b above for a key derivation step 213. Note that for key derivation steps in the present disclosure, the X coordinate of a derived public key can be taken or used as input into the key derivation function. The output of a key derivation function 216 can comprise both (i) a symmetric ciphering key K1 216a and (ii) a MAC key 216b. The use of key K1 216a and MAC key 216b are described in connection with encryption step 217 and decryption step 219 in Figure 2c.

For a key derivation step 213n by server 101, server 101 can conduct the equivalent steps as key derivation step 213 in Figure 2b, with point multiplication operations depicted in Figure 2e. Server 101 can perform an ECC point addition and point multiplication step 214a using the values X1 211a and X2 212a, as well as the numbers N1 298 and N2 299. The value X1 211a could be received by server 101 from key server 102 in message 206a. Note that the value X1 211a is derived by key server 102 using an ECDH key exchange step 211 as depicted and described in connection with Figure 2b. A server 101 could calculate the value for X2 212a using an ECDH key exchange step 212 in Figure 2b. The value or point X1 211a can be multiplied by number N2 299. The value or point X2 212a can be multiplied by the number N1 298. An ECC point addition can be performed on the two ECC points obtained in each of the previous two sentences in order to calculate a value X3' 215a. The exemplary calculations for point multiplication on X1 211a (with N2 299) and X2 212a (with N2 298) by device 103 would also be calculated by server 101. In other words, the exemplary data and numbers depicted above for the calculations by device 103 could also be calculated by server 101 in order to mutually derive the same value for X3' 215a. The mutually derived value for X3' 215a can be input into key derivation function 216 in order to calculate a symmetric ciphering key K1 216a and a MAC key 216b, which can comprise the same numbers as calculated by device 103 in a step 218n.

Petitioner Samsung Ex-1004, 1358

The source of values for N1 298 and N2 299 for both device 103 and server 101 could be mutually obtained in several ways. N1 298 and N2 299 could be recorded and shared with a set of cryptographic parameters 104, such that selecting a subset of the cryptographic parameters 104a could determine the values or numbers to use for N1 298 and N2 299. In another exemplary embodiment, N1 298 and/or N2 299 could comprise pre-shared secret values or keys, such that device 103 receives the values in a secure manner before sending message 203, such as, but not limited to, recording the values at functionally the same time network static public key PK.network 102a is recorded in device 103. Server 101 could receive the values N1 298 and N2 299 in a secure manner, such as from key server 102 in a secure session 221. Other possibilities exist as well for a device 103 and a server 101 to obtain the numbers N1 298 and N2 299 without departing from the scope of the present disclosure. In exemplary embodiments, the number for N1 298 or N2 299 can be either equal, or the numbers could comprise different values.

A device 103 and a server 101 could also conduct a number derivation step 297 in order to obtain the numbers N1 298 and N2 299, which is also depicted in Figure 2e. For a number derivation step 297, a static public key can be input into a secure hash algorithm 291, such as SHA-256. The static public key can be any public key shared between a device 103 and server 101 (e.g. where one node records the public key and the other node records the corresponding private key). In exemplary embodiments depicted in Figure 2e, the public key for a number derivation step 297 can comprise the network static public key PK.network 102a, where a server 101 can derive or calculate the network static public key can be derived from the network static private key SK.network 102b using parameters 104. Other exemplary public keys shared between device 103 and server 101 can comprise any of public keys Ed 103a, E1 101a, D1 103c, etc. The node recording the corresponding private key can calculate the public key using the parameters.

The output of the secure hash algorithm 291 can be input into a select digits function 292. The select digits function 292 could take a subset of the hash value resulting from hash 291, such as leading digits for N1 298 and trailing digits for N2 299. Or, a number N1 298 could be derived from a select digits function 292 over a hash 291 of the X coordinate of a public key and the number N2 299 could be derived by a select digits function 292 over a hash 291 of the Y coordinate of the same public key. Other subsets or logic for the select digits function 292 using the hash value from hash algorithm 291 can be used as well, without departing from the scope of the present disclosure. The output of the select digits function 292 can comprise the value N1 298 and N2 299. Since both device 103 and server 101 and/or network 105 can securely share PK.network 102a, then the same calculations for a number derivation step 297 can be performed by the nodes in order to mutually obtain the numbers N1 298 and N2 299. The values for N1 298 and N2 299 can be used by (i) device 101 when conducting the key exchange step 218n and (ii) server 101 when conducting the key derivation step 213n.

Petitioner Samsung Ex-1004, 1359

Figure 3a

Figure 3a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a mobile device, a g node b, and a key server, in accordance with exemplary embodiments. System 301 can include a mobile device 103', a "next generation node b" 101', and a key server 102. Mobile device 103' can comprise a smart phone, a device for the "Internet of Things" (IoT), a tablet with a modem, or possibly a fixed station device that connects with a 5G or 6G network. Mobile device 103' can operate similar to a device 103, with the additional functionality of connecting to a wireless network, where the network supports 3GPP standards and can also comprise a wide area network such as a public land mobile network. A "next generation node b" 101' (depicted as gNb 101') can contain the equivalent electrical components as those depicted for a server 101 in Figure 1b, except gNb 101' can also operate as a base transceiver station to send and receive data wirelessly with mobile device 103'. The key server 102 could operate as part of an Authentication Server Function (AUSF) or equivalent functionality. Note that the distributed nature of the ECDH key exchanges as depicted in Figure 2a and Figure 2b and Figure 2c have benefits for the wireless WAN architecture in Figure 3a, SK.network 102b for a mobile device 103' does not need to be recorded or operated by a gNb 101'

In exemplary embodiments, a mobile device 103', a gNb 101', and a key server 102 can conduct a step 222', where a step 222' can comprise primarily the step 222 as depicted and described in Figure 2a. There can be some differences between a step 222 and a step 222'. Note that before the steps 222' depicted in Figure 3a, a mobile device 103' and a gNb 101' could conduct steps to establish communications between the nodes, such as recording parameters for RF communications by the mobile device 103' in a SIM card or eUICC. A mobile device 103' could also conduct steps to authenticate the network 105 operating a gNb 101'. For a step 222', a mobile device 103' can send message 203 with the device ephemeral public key Ed 103a and also an obfuscated identity for device 103', where the obfuscated identity can also comprise a temporary identity for device 103. A gNb 101' can use the obfuscated identity to track the device 103 from a potential plurality of devices 103 communicating over a wireless network.

The gNb 101' can forward the device identity and the received device ephemeral public key to the key server 102. The key server 102 can look up a unique key 102v for device 103 for the network static private key 102b corresponding to the network static public key 102a recorded by the device 103. The key server 102 can calculate value X1 211a as depicted in Figure 2b, and send the gNb 101' the value X1 211a over a secure session. The gNb 101' can conduct an ECDH key exchange step 212 and calculate value X2 212a, using the received device ephemeral public key Ed 103a and the derived server ephemeral private key e1 101b. The gNb 101' can calculate the value X3 215 via ECC point addition over X1 211a and X2 212a. The gNb 101' can calculate a symmetric ciphering

Petitioner Samsung Ex-1004, 1360

key K1 216a using the value X3 215 and a KDF 216. The gNb 101' can send the mobile device 103' the derived server ephemeral public key E1 101 in a message 206c from a step 222. Note that some data within ciphertext 217b can be omitted from a message 206c in a step 222', where step 222' is depicted in Figure 3a and comprises equivalent steps as a step 222 in Figure 2a.

The mobile device 103' can receive the message 206c from a step 222'. The mobile device 103', gNb 101', and key server 102 can conduct a step 223, where a step 223 was depicted and described in connection with Figure 2a above. The mobile device 103' can send gNb 101' a message 210a with ciphertext 217d, where ciphertext 217d can include a device identity ID.device 103i as plaintext encrypted in the ciphertext 217d. The ciphertext 217d can be encrypted with the derived symmetric ciphering key K1 216a and a symmetric ciphering algorithm 225, where key K1 216a was derived by mobile device 103' in a step 222'. The identity for the mobile device 103i can comprise a subscription permanent identifier (SUPI), and by transmitting the SUPI within a ciphertext 217d, the SUPI can remain confidential and not transmitted in the clear through a wireless network. Other possibilities for the use of a step 222' and a step 223 between a mobile device 103' and gNb 101' exist without departing from the scope of the present disclosure.

Figure 3b

Figure 3b is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a client, a server, and a key server, in accordance with exemplary embodiments. System 302 can include a client 103', a server comprising server 101, and a key server 102. In exemplary embodiments, client 103' can comprise a client using security steps as described in by transport layer security (TLS) sessions version 1.3 and also subsequent and related versions of IETF RFC standards. Client 103' can also comprise a client using security steps as described in datagram transport layer security (DTLS) RFC 6347 and subsequent versions that incorporate ECDH key exchanges. Although depicted in Figure 3b as a client 103', the client 103' could also comprise a device 103, where the device 103 can conduct the steps of a client 103' at the networking, transport, and application layer of the traditional Open Systems Interconnection (OSI) model.

Client 103' can comprise a computing device that records a network static public key PK.network 102a. Note that TLS version 1.3 and DTLS version 1.3 contemplate that the client and a server can use ephemeral ECDH key exchanges (one on the client and one on the server) in order to establish a mutually derived secret shared key for a symmetric ciphering algorithm. The difference between (i) a client 103' (which can comprise a device 103 supporting TLS or DTLS standards) and (ii) a client for TLS or DTLS standards can be that client 103' can record a network static public key PK.network 102a. As depicted in Figure 1c, the network static public key PK.network 102a could comprise either (i) a shared key 102z across a plurality of different devices 103 (or clients 103'), or

Petitioner Samsung Ex-1004, 1361

(ii) a unique key 102v, where the network static public key PK.network 102a is a unique number or string or point for client 103'. The key PK.network 102a could be received by client 103' in a secure manner before a client 103' conducts a step 222 with server 101. In exemplary embodiments, PK.network 102a could be received in the form of a certificate with PK.network 102a from a prior TLS or DTLS session before client 103' begins the TLS or DTLS session depicted in Figure 3b. Or, PK.network 102a could be recorded with a set of certificate authority certificates stored with installation of an operating system for device 103.

The use of a network static public key PK.network 102a by client 103' in a step 222 to conduct an ECDHE key exchange with server 101 can have many benefits. The standard handshake as currently proposed for TLS version 1.3 as of June 2018 assumes that a client 103' and a server 101 have no prior relationship. However, for many instances of communication between a client 103' and a server 101, the client 103' may have previously communicated with another server on a network 105 other than server 101. For example, with web browsing a web browser client such as a client 103' will often revisit the same web sites over time, such as a first web site for social networking, a second web site for a search engine, a third web site for news, etc. A TLS or a DTLS session could utilize the fact that the same sites are often re-visited in order to increase security, using the depicted steps of 222 and 223 for a client 103', server 101, and key server 102. Steps 222 in Figure 3b can comprise the set of steps 222 depicted and described in connection with Figure 2a, and steps 223 in Figure 3b can also comprise the set of steps 223 depicted and described in connection with Figure 2a.

Before conducting step 222 in Figure 3b, a client 103' could receive key PK.network 102a from another server in network 105, such as a different web server providing functionality equivalent to server 101. PK.network 102a could also be stored or recorded by a client 103' along with a set of certificate authority certificates (including root certificates) for an operating system of a device operating the client 103'. Or, PK.network 102a could be securely received in a previous TLS or DTLS session, such as receiving PK.network 102a in a certificate verified by client 103' before client 103' conducts a step 222 in Figure 3b. The certificate could be verified by client 103' using a certificate authority root certificate, including verification through any intermediate certificate authority certificates. The client 103' could record the network static public key PK.network 102a in a table 103t along with parameters 104a associated with PK.network 102a. In exemplary embodiments, a table 103t could include certificates such as X.509 v3 certificates for the network static public keys PK.network 102a, where the certificates include digital signatures from a certificate authority. The key PK.network 102a could also be recorded with a URL or domain name (e.g. a server name indication), such that the client 103' would use the key PK.network 102a when establishing a subsequent TLS or DTLS session with server 101, where server 101 uses the recorded URL or domain name. Further, server 101 could be configured so that any key Ed 103a received from IP network 107 on an IP address and/or port number used by server 101 would be forwarded to key server 102, where

Petitioner Samsung Ex-1004, 1362

key server 102 could record and operate with the SK.network 102b corresponding to the public key for PK.network 102a recorded by client 103'. Server 101 could also operate such that a URL is associated with a key server 102 and/or PK.network 102a, such that a call or request of the URL could be used to select the key server 102 and/or PK.network 102a.

For a step 222, a client 103' can (i) derive a device ephemeral public key Ed 103a and private key ed 103b using parameters 104a stored with PK.network 102a and (ii) send server 101 a message 203. The message 203 can include the key Ed 103a and the set of cryptographic parameters 104a associated with Ed 103a. In some exemplary embodiments client 103' implements TLS or DTLS, and message 203 can optionally omit a device identity ID.device 103i. Server 101 could operate in a manner such that (i) Ed 103a is forwarded to key server 102, and (ii) server 101 derives an ephemeral PKI key pair. Key server 102 can conduct an ECDHE key exchange as depicted for a step 222 in Figure 2a using a step 211 in order to calculate the secret value X1 211a. Key server 102 can send server 101 the value X1 211a. Server 101 can use the value X1 211a, along with the derivation of a second secret X2 212a in order to calculate a symmetric ciphering key K1 216a, using the key derivation step 213 with ECC point addition 214 over X1 211a and X2 212a. Thus, by using the embodiment depicted in Figure 3b, a transport layer security session can have security increased, where (a) the ECDHE key exchange contemplated by TLS v1.3 (which would be key exchange 212 in Figure 2b) can also add (b) the additional key exchange step 211a by a key server 102. Note that the mutual derivation of symmetric ciphering key K1 216a by client 103' and server 101 can comprise a one-way authentication of server 101, since server 101 can only derive the key K1 216a if server 101 operates in a network 105 that also records and operates with key SK.network 102b.

The server 101 can send the client 103' the derived server ephemeral public key E1 101a in a message 206c from a step 222. Key E1 101a could be derived by a step (ii) in the above paragraph. Message 206c could comprise a "Server Hello" according to TLS v1.3 in the document "draft-ietf-tls-tls13-28". The ciphertext in the Server Hello can be ciphertext 217b as depicted in Figure 2a, where the ciphertext 217a is encrypted with the mutually derived symmetric ciphering key K1 216a. Note that a step 222 for Figure 3b increases security for a TLS session, since an active attacker could operate as a "man in the middle" between a real client or "true client" and the server 101, where the "man in the middle" could derive its own key Ed 103a and substitute that for the real key Ed 103a from the real client or "true client". Without use of a PK.network 102a, a "man in the middle" (deriving and substituting a key Ed 103a) could (a) mutually derive a symmetric ciphering key similar to K1 216a with server 101 and then (b) receive and decrypt the ciphertext 217b. However, the use of PK.network 102a can stop a "man in the middle" attack since a "man in the middle" cannot derive key K1 216a without also recording the SK.network 102b, which can remain secret and not available to the "man in the middle".

Petitioner Samsung Ex-1004, 1363

The client 103' can receive the message 206c from a step 222 from a server 101. The client 103', server 101, and key server 102 can conduct a step 223, where a step 223 was depicted and described in connection with Figure 2a above. The client 103' can derive the same key K1 216c using a step 218 and the PK.network 102a. The client 103' can decrypt ciphertext 217b using key K1 216a. The client 103' can process the plaintext data, such as recording a certificate for server 101 (e.g. cert.server 101c from Figure 2a), and verifying a signature 101s from server 101. The client can also read a random number transmitted in the ciphertext 217b and create a digital signature over the random number. The client can encrypt a ciphertext 217d with data to respond to server 101. The ciphertext 217d can be encrypted with the derived symmetric ciphering key K1 216b and a symmetric ciphering algorithm 211a, where key K1 216a was derived by client 103' in a step 223. Other possibilities exist for the use of a step 222 and a step 223 between a client 103' and server 101 without departing from the scope of the present disclosure.

For the exemplary embodiment depicted in Figure 3b for support of TLS and DTLS secured data sessions, a message 203 can comprise a "client hello" message, a message 206c can comprise a "server hello" message, and message 210a can comprise a "finished" message from the client 103'. For exemplary embodiments, message 203 as a "client hello" message can omit a device identity 103i (such as a permanent identifier for client 103' or device 103, but the "client hello" message could include other identifying information for client 103' such as (i) an originating IP address and source port number for message 203, (ii) an obfuscated and/or temporary identity such as a random number for a session, and other possibilities exist as well without departing from the scope of the present disclosure.

In addition, embodiments depicted in Figure 3b solve a significant challenge for resource constrained devices to fully authenticate a certificate cert.server 101c. There could be many layers of intermediate certificates between cert.server 101c and a certificate authority root certificate stored in device 103. Checking for certificate validity for all intermediate certificates and for revocation or OSCP signatures and/or stapling could add many levels of signature verifications. ARM reported a 32 Cortex M4 processor with 32 bits and operating at 84 Mhz requires ~420 ms for a single ECDSA signature verification (secp521r1) ("Performance of State-of-the-Art Cryptography on ARM-based Microprocessors", July 21, 2015). There could be 8 or more signatures to be verified for a full certificate chain verification of cert.server 101c and related OSCP signatures. A device could conduct the single authenticated key exchange step 218 in less than 15% of the time and power required for the full, traditional certificate chain verification. Also, there are reduced chances for errors due to unsupported parameters for (x) a single authenticated ECDH key exchange step compared to (y) multiple certificate verifications steps with OSCP verification. Consequently, the communications for a TLS session or DTLS session can remain secured more efficiently using a step 222 and step 223, while recording and using (i) SK.network 102b with network 105 and (ii) PK.network 102a with client

Petitioner Samsung Ex-1004, 1364

103', compared to traditional TLS or DTLS implementations with multiple layers of certificate authorities through root certificates.

Figure 3c

Figure 3c is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by an initiator, a responder, and a key server, in accordance with exemplary embodiments. System 303 can include an initiator 103', a responder 101' and a key server 102. Initiator 103' can comprise a computing device 103, with the specific additional functionality of an initiator according to the DPP Specification Versions 1.0 from the WiFi Alliance. Responder 101' can comprise a device with (i) electrical components similar or equivalent to a server 101 depicted in Figure 1b above, and (ii) the specific additional functionality of a responder according to the DPP Specification Version 1.0 of the WiFi Alliance. For example, initiator 103' and responder 101' can communicate via a WiFi network on a LAN between the two devices, which could also comprise the IP network 107. Responder 101' can operate in a networked configuration to communicate with key server 102 via a private network 107a or a secure session 221 as depicted in Figure 2a. In some embodiments, responder 101' can communicate with key server 102 via an IP network 107, where the use of secure session 221 can create a private network 107a between responder 101' and key server 102.

An initiator 103', responder 101' and a key server 102 can conduct a step 222, where a step 222 is depicted and described in connection with Figure 2a above. An initiator 103', responder 101' and a key server 102 can then conduct a step 223, where a step 223 is depicted and described in connection with Figure 2a above. As depicted in Figure 3c, several PKI keys within a DPP specification version 1.0 can have corresponding keys for a step 222 and step 223. Note that additional steps in addition to those depicted in Figure 3c can be conducted by an initiator 103' and a responder 101', such as responder 101' deriving PKI keys in a step 101x from Figure 1a and also conducting additional ECDH key exchanges in order to derive a symmetric ciphering key ke in addition to symmetric ciphering key K1 216a. In other words, initiator 103' and responder 101' could perform additional ciphering than that depicted for a step 222 in Figure 2a, but for exemplary embodiments such as that depicted in Figure 3c the initiator 103' and responder 101' could conduct at least the steps depicted in order to mutually derive a symmetric ciphering key K1 216a and use the key to create a ciphertext 217b by responder 101' and decrypt the ciphertext 217b by initiator 103'.

As depicted in Figure 3c, the device ephemeral public key Ed 103a can comprise the initiator protocol public key Pi 303a. The device ephemeral private key ed 103b can comprise the initiator protocol private key pi 303b. The server ephemeral public key E1 101a can comprise the responder

Petitioner Samsung Ex-1004, 1365

protocol public key Pr 301a. The server ephemeral private key e1 101b can comprise the responder protocol private key pr 301b. The network static public key PK.network 102a can comprise the responder bootstrap public key 302a. The network static private key SK.network 102b can comprise the responder bootstrap private key 302b. As described below, other steps from Figure 2a can be equivalent to those depicted in Figure 3c.

For a message 203 sent from initiator 103' to responder 101', the message 203 with the key Pi 303a can also include a ciphertext. The message 203 in a step 222 can comprise a "DPP Authentication Request" message from the DPP v1.0 standard. Responder 101' can communicate with key server 102 and receive the value X1 211a. Responder 101' can also derive the server ephemeral public key E1 101 (comprising the responder protocol public key Pr 301a) and the server ephemeral private key e1 101b (comprising the responder protocol private key pr 301b). The Responder 101' can use KDF 216 to convert X1 211a into a symmetric encryption key (which can be different than key K1 216a from Figures above). Responder 101' can use the symmetric encryption key from X1 211a to decrypt the ciphertext with a message 203. Responder 101' can then conduct the key exchange step 212 and step 213, along with modified versions of KDF 216 in order to derive a key ke. Responder 101' can encrypt data with the key ke and send initiator 103' a message 206c with the encrypted data. The message 206c can comprise a "DPP Authentication Response" message from the DPP v1.0 standard. Initiator 103' can then send responder 101' a "DPP Configuration Request" message, which could comprise message 210a in a step 223 as depicted in Figure 2a.

A benefit for the use of a step 222 and step 223 for an initiator 103' and a responder 101' is that the responder bootstrap private key br 302b can remain securely recorded in a network 105 and does not need to be recorded and operated by responder 101. In this manner, the responder bootstrap public key Br 302a can be freely shared with multiple different initiators 103', including recording the key Br 302a in a plurality of initiators 103' in the form of a shared key 102z as depicted in Figure 1c. The use of a shared key 102z with multiple different initiators 103' (while keeping SK.network 102b or key br 302b securely recorded in a key server 102) simplifies the distribution of key Br 302a to multiple different initiators 103'.

For exemplary embodiments, the initiators 103' could have a key Br 302a recorded during manufacturing or distribution of the computing device operating initiator 103'. In other words, a device manufacturer upon device manufacturing with initiator 103' may not know which responder 101' may communicate with initiator 103' during a subsequent DPP session. However, a manufacturer of device with initiator 103' could record a plurality of different keys Br 302a for different networks 105 (similar to different keys PK.network 102a in for a table 103t Figure 1c), and in this manner initiator 103' can have a higher probability of successfully using a pre-recorded key Br 302a (or key PK.network 102a) in order to conduct a DPP session without requiring a separate or different additional step of acquiring the key Br 302a "out of band". Thus, the use of the embodiment

Petitioner Samsung Ex-1004, 1366

for an initiator 103' and a responder 101' can simplify the use and deployment of DPP sessions, while simultaneously increasing the securing of the session, since the responder bootstrap private key br 302b (in the form of SK.network 102b) can remain securely recorded within a network 105 on a key server 102.

5

Conclusion

Various exemplary embodiments have been described above. Those skilled in the art will understand, however, that changes and modifications may be made to those examples without departing from the scope of the claims.

Petitioner Samsung Ex-1004, 1367

CLAIMS

1. A method for securing communications, the method performed by a set of servers, the method comprising:

a) receiving, by a first server and from a device, a first message comprising a device ephemeral public key, wherein the device ephemeral public key is associated with a set of cryptographic parameters;

b) deriving, by the first server, a server ephemeral public key and a corresponding server ephemeral private key;

c) receiving, by a second server, the device ephemeral public key from the first server;

d) conducting, by the second server, a first elliptic curve Diffie Hellman (ECDH) key exchange using at least (i) the device ephemeral public key and (ii) a network static private key in order to derive a first shared secret;

e) sending, by the second server and to the first server, the first shared secret over a secure connection;

f) conducting, by the first server, a second ECDH key exchange using at least, (i) the received device ephemeral public key and (ii) the server ephemeral private key in order to derive a second shared secret;

g) conducting, by the first server, an ECC point addition operation using the first shared secret and the second shared secret in order to derive a third shared secret;

h) deriving, by the first server, a symmetric ciphering key using at least (i) a key derivation function and (ii) the third shared secret;

i) generating, by the first server, a ciphertext using at least the symmetric ciphering key; and

j) sending, by the first server and to the device, a second message with the server ephemeral public key and the ciphertext.

Petitioner Samsung Ex-1004, 1368

2. The method of claim 1, wherein the set of servers comprises the first server and the second server, and wherein the first server and the second server record the set of cryptographic parameters before the first server receives the first message.

3. The method of claim 2, further comprising recording, by the second server, the network static private key before the first server receives the first message.

4. The method of claim 1, further comprising recording, by the device before step a), (i) a network static public key corresponding to the network static private key and (ii) the set of cryptographic parameters.

5. The method of claim 4, further comprising conducting, by the device, a third ECDH key exchange using at least (i) a device ephemeral private key corresponding to the device ephemeral public key, (ii) the server ephemeral public key, and (iii) the network static public key in order to derive the third shared secret.

6. The method of claim 5, further comprising conducting, by the device, a second ECC point addition operation with the device ephemeral public key and the network static public key in order to derive the third shared secret, wherein the device derives the symmetric ciphering key using at least (i) the key derivation function and (ii) the third shared secret.

7. A method for conducting a secure session between a device and a server, the method performed by the device, the method comprising:

a) receiving, by the device, a certificate for a network public key with a set of cryptographic parameters, wherein the device verifies the certificate with at least a root certificate stored by the device.

b) deriving, by the device, a device ephemeral public key and a corresponding device ephemeral private key using at least the set of cryptographic parameters;

c) sending, by the device and to the server, the device ephemeral public key and the set of cryptographic parameters;

Petitioner Samsung Ex-1004, 1369

d) receiving, by the device and from the server, a server ephemeral public key and a ciphertext;

e) conducting, by the device, an elliptic curve Diffie Hellman (ECDH) key exchange using at least (i) the server ephemeral public key, (ii) the network public key, and (iii) the device ephemeral private key in order to derive a shared secret;

h) deriving, by the device, a symmetric ciphering key using at least (i) a key derivation function and (ii) the shared secret; and

i) decrypting, by the device, the ciphertext using the symmetric ciphering key.

8. The method of claim 7, wherein the device conducts the ECDH key exchange by conducting (i) an ECC point addition operation with the server ephemeral public key and the network public key to derive a point, and (ii) an ECC point multiplication operation with the point and the device ephemeral private key.

9. The method of claim 7, wherein the device conducts the ECDH key exchange by conducting (i) a first ECC point multiplication operation with the server ephemeral public key and the device ephemeral private key to derive a first point, (ii) a second ECC point multiplication operation with the network ephemeral public key and the device ephemeral private key to derive a second point, and (iii) an ECC point addition operation with the first point and the second point to derive the shared secret.

10. The method of any one of claims 7-9, further comprising: conducting, by the server , a second ECDH key exchange with (i) the received device ephemeral public key and (ii) a server ephemeral private key in order to derive a second shared secret; conducting, by a key server, a third ECDH key exchange with (i) the device ephemeral public key and (ii) a network static private key in order to derive a third shared secret; sending, by the key server and to the server, the third shared secret; and conducting, by the server, an ECC point addition operation with the second shared secret and the third shared secret in order to derive the shared secret.

Petitioner Samsung Ex-1004, 1370

11. The method of any one of claims 7-10, further comprising before step b), (i) storing a network public key table with a plurality of network public keys and associated domain names, and (ii) selecting the network public key from the network public key table with a domain name for the server.

12. The method of any one of claims 7-11, wherein the network public key comprises a unique key for the device, and wherein the device sends the server a secure hash value for the network public key with the device ephemeral public key.

13. The method of claim 7, wherein the network public key comprises a shared key, wherein the server selects a key server using a domain name for the server in the set of cryptographic parameters, wherein the key server stores a network private key for the network public key.

14. A method for a key server to support an authenticated key exchange, the method performed by the key server, the method comprising:

a) recording in a database (i) a plurality of network private keys with parameters and (ii) a plurality of secure hash values for corresponding network public keys;

b) establishing a secure session with a server, wherein the server communicates with a plurality of devices over a public Internet;

c) receiving, from the server, a device ephemeral public key and a secure hash value;

d) selecting, from the database, a network private key and the parameters for the network private key using the secure hash value;

e) conducting a first elliptic curve Diffie Hellman (ECDH) key exchange using the network private key, the device ephemeral public key, and the parameters in order to derive a point; and

f) sending, to the server, the derived point, wherein the derived point is used with a second ECDH key exchange and a server ephemeral private key in order to derive a shared secret and a symmetric ciphering key.

Petitioner Samsung Ex-1004, 1371

15. The method of claim 14, further comprising: conducting, by the server, an ECC point addition operation with the derived point and a value from the second ECDH key exchange in order to derive the shared secret.

16. The method of any one of claims 14-15, further comprising: conducting, by a device, a third ECDH key exchange in order to derive the shared secret, and wherein the third ECDH key exchange uses at least (i) a received server ephemeral public key for the server ephemeral private key, (ii) a stored network public key for the network private key, and (iii) the device ephemeral private key.

17. The method of claim 16, further comprising: establishing, by the device and the server a secure session over the public Internet using at least the symmetric ciphering key.

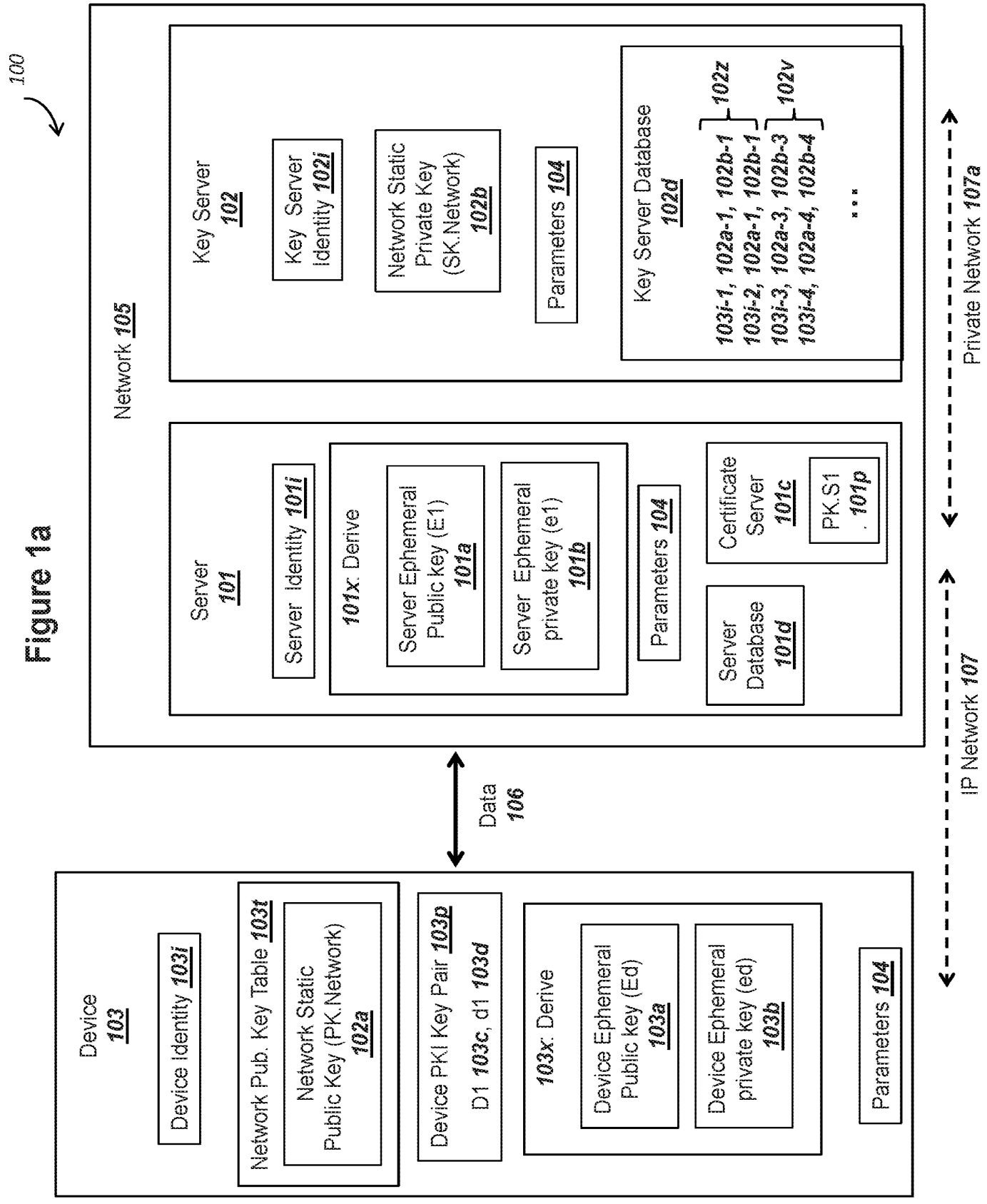
18. The method of claim 16, wherein the network private key comprises a unique key for the device, and wherein the device receives the network public key in a certificate.

19. The method of any one of claims 14-18, further comprising, after step d) and before step e), verifying, by the key server, the received device ephemeral public key with the selected set of cryptographic parameters.

20. The method of any one of claims 14-19, further comprising, after step f), receiving, from the server, a message after the server decrypts a ciphertext with the symmetric ciphering key.

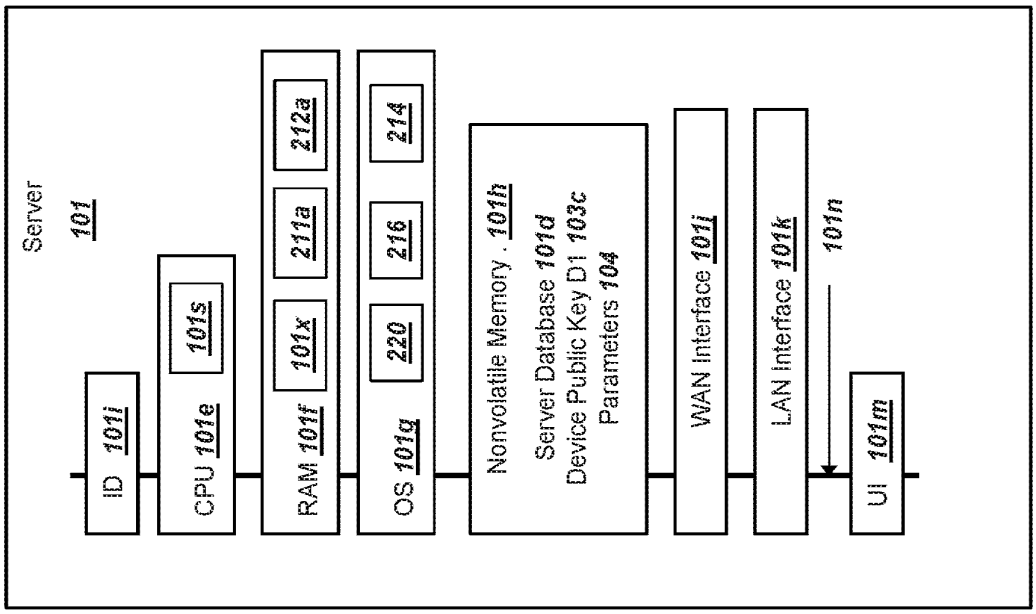
Petitioner Samsung Ex-1004, 1372

Figure 1a



Petitioner Samsung Ex-1004, 1373

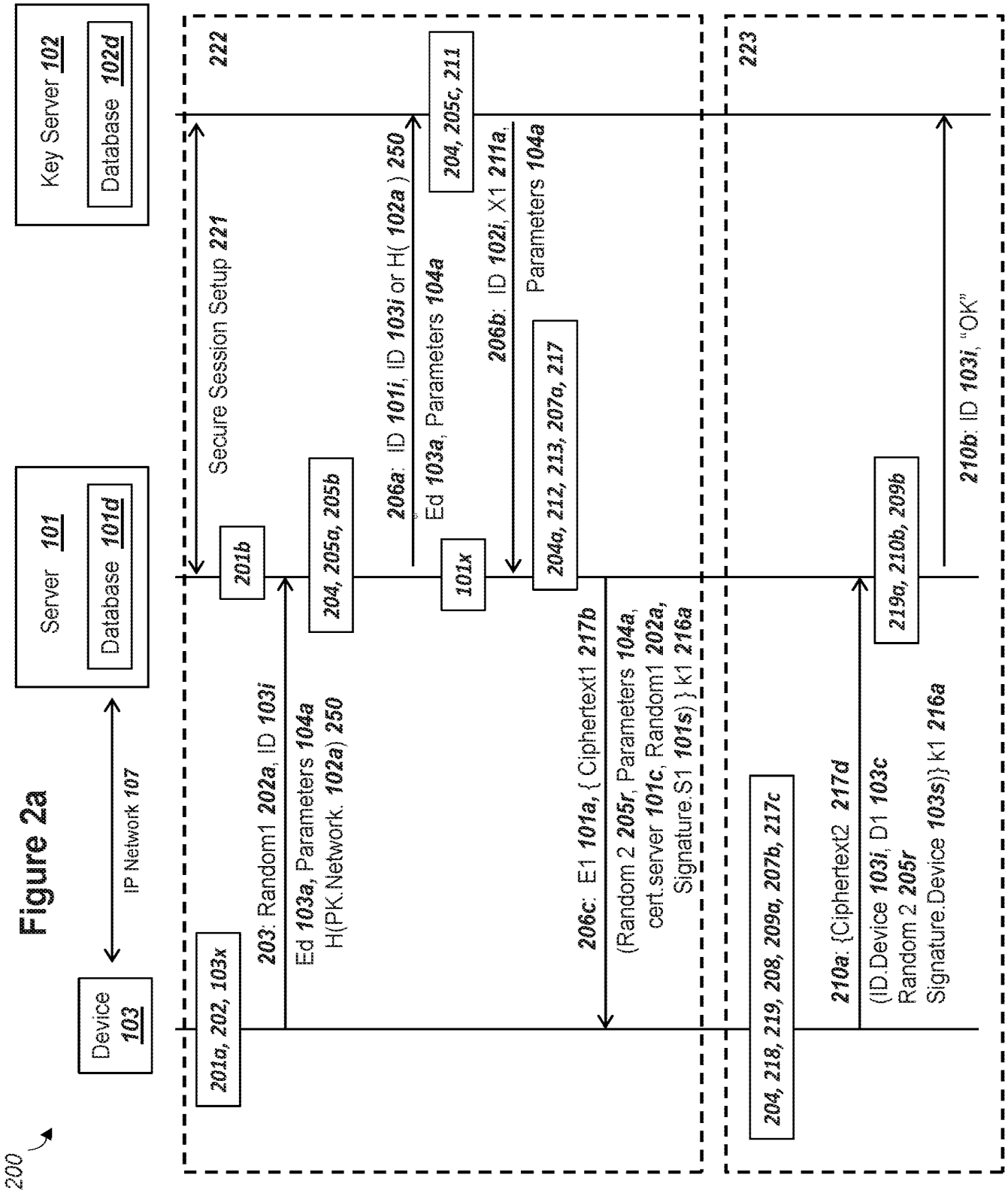
Figure 1b



Petitioner Samsung Ex-1004, 1374

Figure 1c

Device 103 (1)
Network Pub. Key Table 103t
Key ID
Network 105
ID.server 101i
PK.Network 102a
1
Network A
Server.Network-A.com
)M[a2D.*2Br6xZf[n}1AM(HF*c2GW)ouc2tQ_[qW
2
Network B
Server.Network-B.com
#w-?dRL7fz=UpUP^515:}gN1d9hTMcp.?15tag2M
3
Network C
Server.Network-C.com
&^GB?%c.N^hztm69];wIfT}NvWSDqs^i]sxHr&]9
Shared Network Public Key 102z
"Per-Device" or Unique Network Public Keys 102v
Device 103 (2)
Network Pub. Key Table 103t
Key ID
Network 105
ID.server 101i
PK.Network 102a
1
Network A
Server.Network-A.com
)M[a2D.*2Br6xZf[n}1AM(HF*c2GW)ouc2tQ_[qW
2
Network B
Server.Network-B.com
#w-?dRL7fz=UpUP^515:}gN1d9hTMcp.?15tag2M
3
Network C
Server.Network-C.com
+.K8Ka%pq0WXPNQtH4jI.LbMfKk?74?>d)rY~^#:
Shared Network Public Key 102z
"Per-Device" or Unique Network Public Keys 102v
Device 103 (3)
Network Pub. Key Table 103t
Key ID
Network 105
ID.server 101i
PK.Network 102a
1
Network A
Server.Network-A.com
)M[a2D.*2Br6xZf[n}1AM(HF*c2GW)ouc2tQ_[qW
2
Network B
Server.Network-B.com
#w-?dRL7fz=UpUP^515:}gN1d9hTMcp.?15tag2M
3
Network C
Server.Network-C.com
%LR!rFb}Lm9C.^Dp>+FaZzH=T[okv/p9g/4U*W.m
Shared Network Public Key 102z
"Per-Device" or Unique Network Public Keys 102v



Petitioner Samsung Ex-1004, 1376

Figure 2b

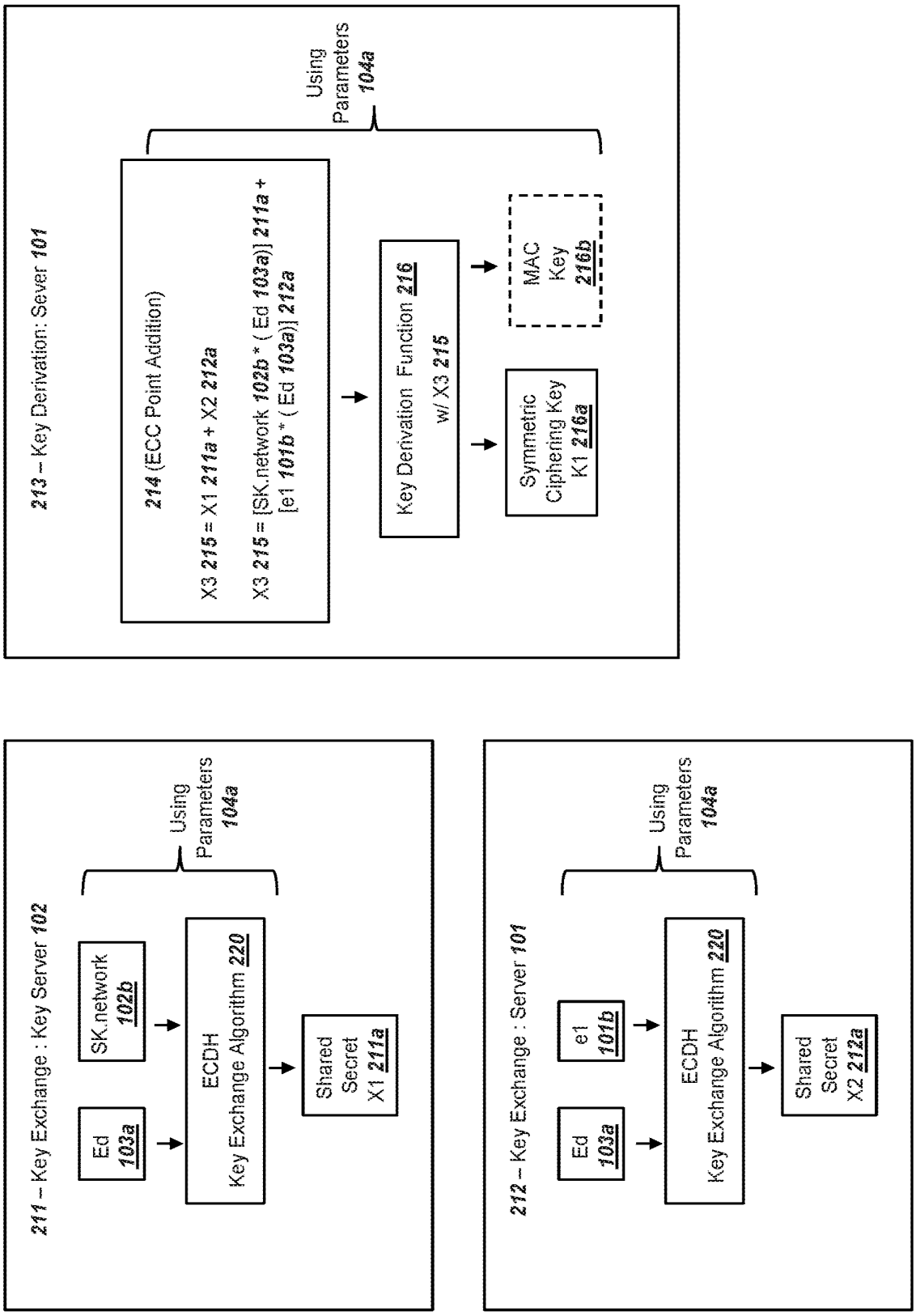
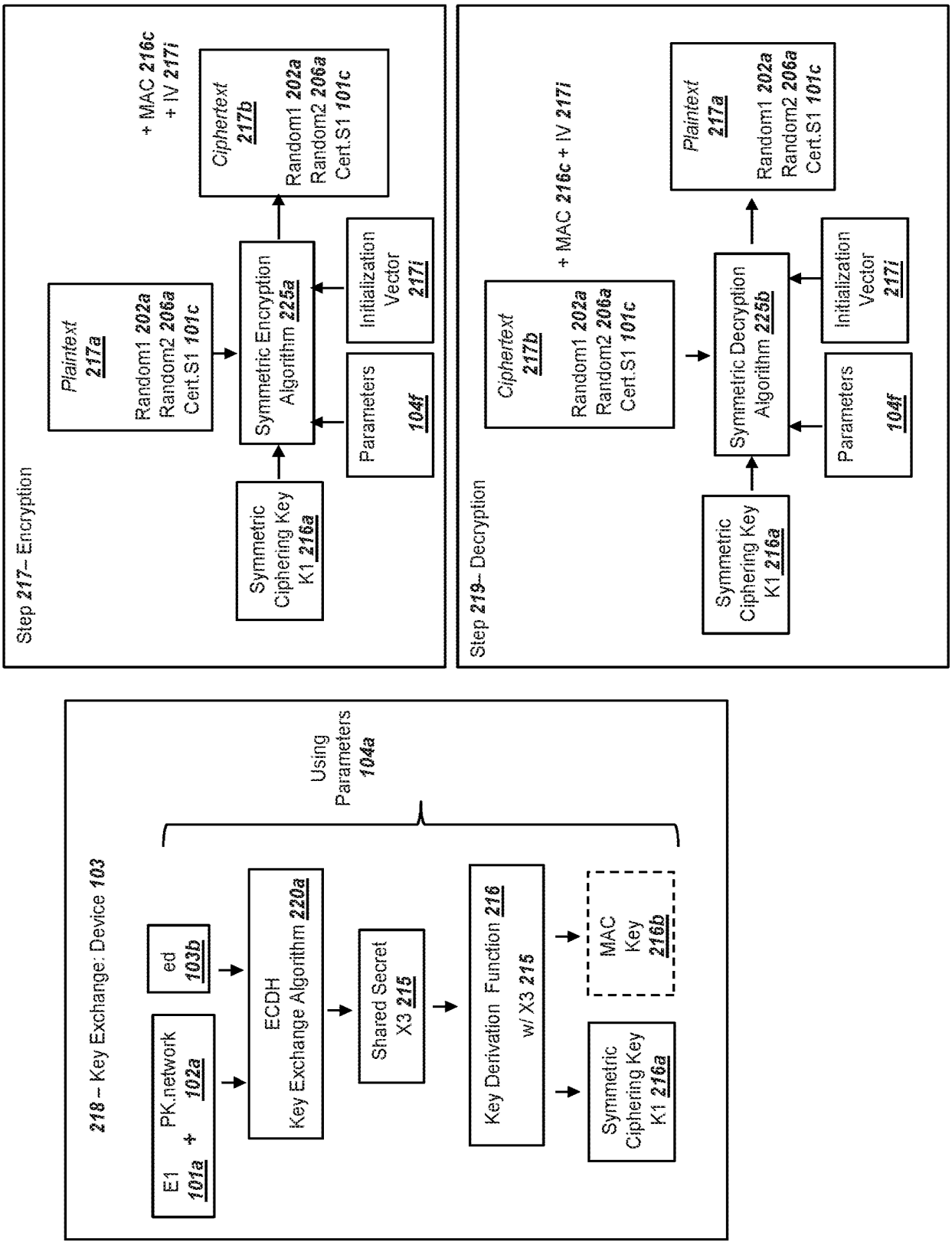

213 – Key Derivation: Sever 101
214 (ECC Point Addition)
X3 215 = X1 211a + X2 212a
X3 215 = [SK.network 102b * (Ed 103a)] 211a + [e1 101b * (Ed 103a)] 212a
Using Parameters 104a
Key Derivation Function 216 w/ X3 215
Symmetric Ciphering Key K1 216a
MAC Key 216b
211 – Key Exchange : Key Server 102
Ed 103a
SK.network 102b
ECDH Key Exchange Algorithm 220
Shared Secret X1 211a
Using Parameters 104a
212 – Key Exchange : Server 101
Ed 103a
e1 101b
ECDH Key Exchange Algorithm 220
Shared Secret X2 212a
Using Parameters 104a

Figure 2c



Petitioner Samsung Ex-1004, 1378

Figure 2d

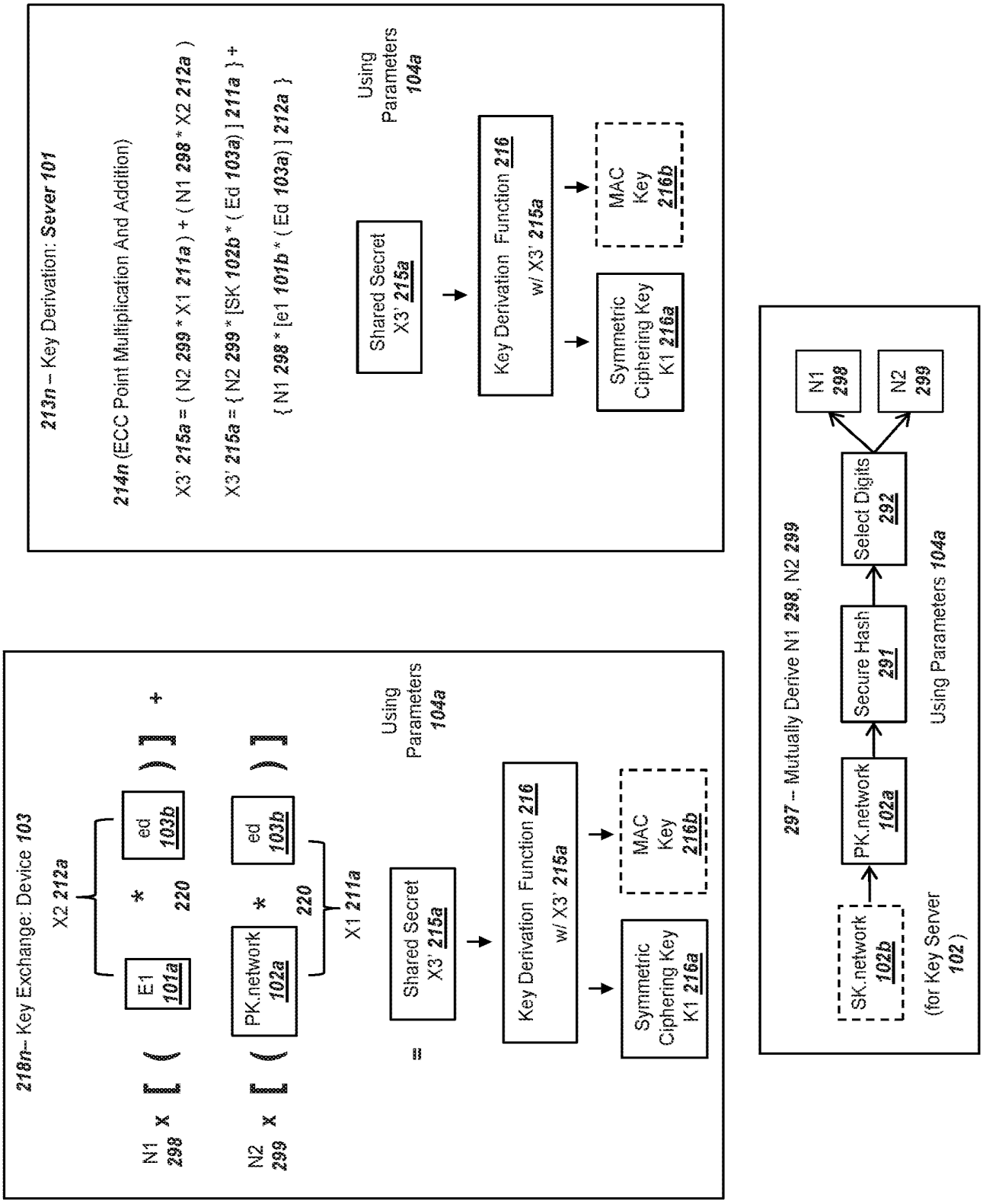
Server Database ***101d***

Random 1	Ed	Parameters	Source IP.Port	H(PK network ***102a***)	ID key-server	X1	E1	e1	X2	X3	K1	Random 2
202a	***103a***	***104a***	***203a***	***250***	***102i***	***211a***	***101a***	***101b***	***212a***	***215***	***216a***	***205r***
202a-1	103a-1	104a-1	203a-1	250-1	102i-1	211a-1	101a-1	101b-1	212a-1	215-1	216a-1	205r-1
202a-2	103a-2	104a-3	203a-2	250-2	102i-2	211a-2	101a-2	101b-2	212a-2	215-2	216a-2	205r-2
202a-3	103a-3	104a-1	203a-3	250-2	102i-2	211a-3	101a-3	101b-3	212a-3	215-3	216a-3	205r-3
					●●●							

Cryptographic Parameters ***104***

Set	PKI Key Length	Curve	Hash	Symmetric Ciphering Key Length	Symmetric Ciphering Parameters	Random Length
104a	***104b***	***104c***	***104d***	***104e***	***104f***	***104g***
A	256	p256	SHA-256	256	C1	128
B	384	p384	SHA-384	384	C2	192
C	512	secp521r1	SHA-512	512	C3	256
D	384	Curve25519	SHA-3	384	C4	192
			●●●			

Figure 2e



Petitioner Samsung Ex-1004, 1380

Figure 3a

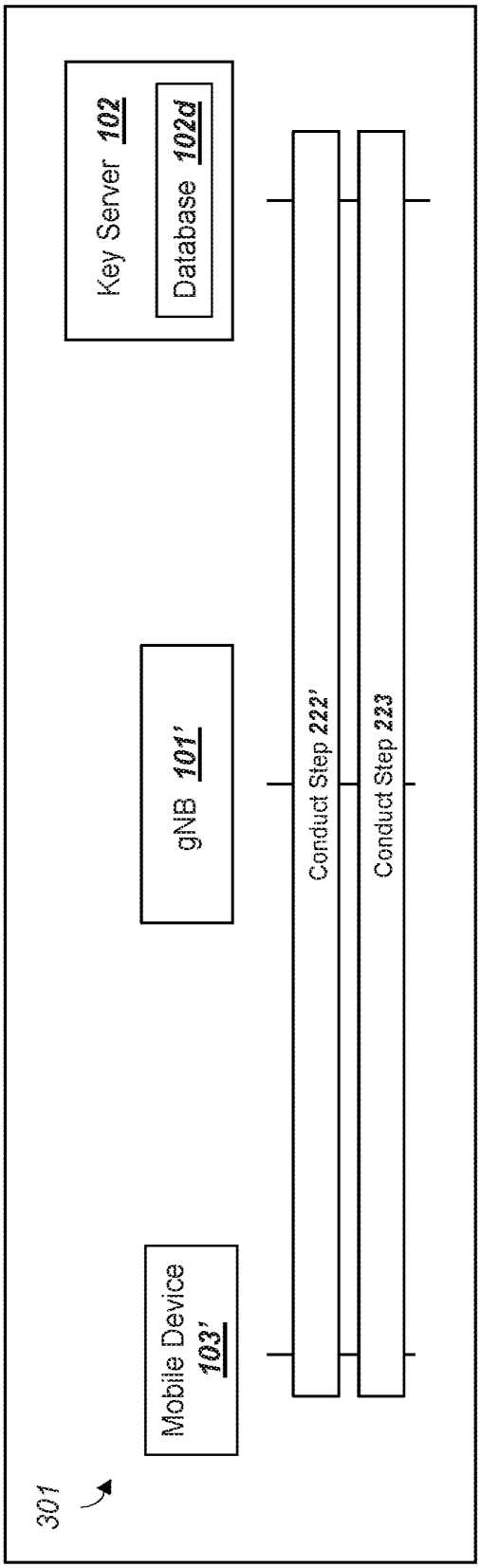


Figure 3b

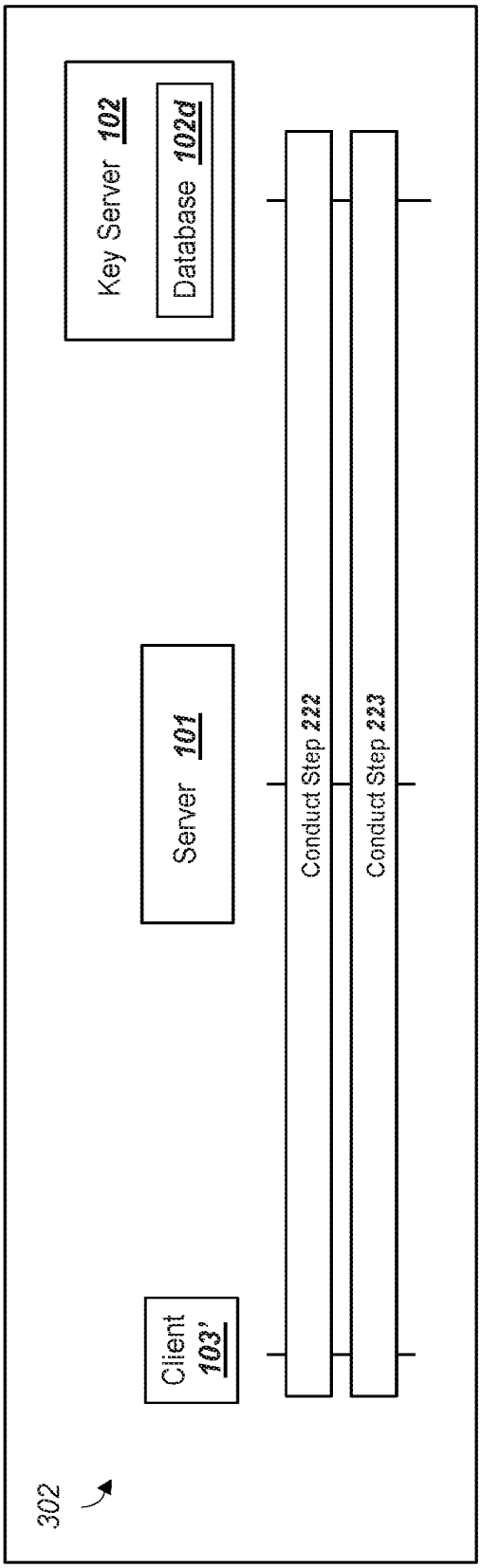
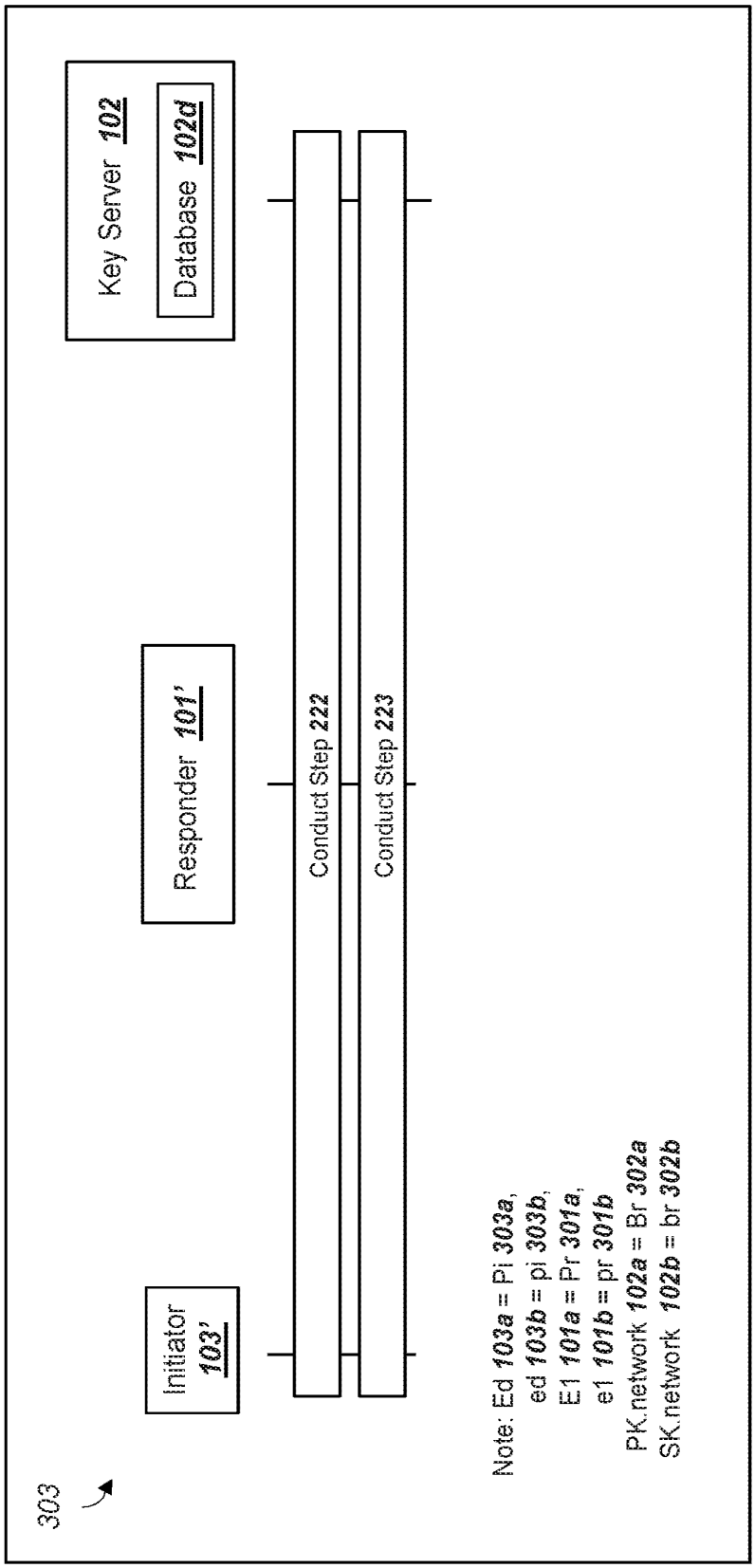


Figure 3c


303
Initiator 103'
Responder 101'
Key Server 102
Database 102d
Conduct Step 222
Conduct Step 223
Note: Ed 103a = Pi 303a,
ed 103b = pi 303b,
E1 101a = Pr 301a,
e1 101b = pr 301b
PK.network 102a = Br 302a
SK.network 102b = br 302b

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 2019/037911

A. CLASSIFICATION OF SUBJECT MATTER

H04L 9/00 *(2006.01)*

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)

H04L H04W

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

PatSearch (RUPTO internal), USPTO, PAJ, K-PION, Esp@cenet, Information Retrieval System of FIPS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2017/0237571 A1 (CLOUDFLARE, INC.) 17.08.2017, paragraphs [0043]-[0050], [0076], [0086], [0087], [0139]-[0144], [0370]-[0374]	14
A		1-6, 15-18
X	US 2015/0067338 A1 (AKAMAL YECHNOLOGIES, INC.) 05.03.2015, paragraphs [0045]-[0056]	7
A		8-10, 13
A	US 2018/0026784 A1 (MASTERCARD INTERNATIONAL INCORPORATED) 25.01.2018	1-10, 13-18

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

☐ See patent family annex.

* 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 document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16 September 2019 (16.09.2019)	26 September 2019 (26.09.2019)

Name and mailing address of the ISA/RU:	Authorized officer
Federal Institute of Industrial Property, Berezhkovskaya nab., 30-1, Moscow, G-59, GSP-3, Russia, 125993 Facsimile No: (8-495) 531-63-18, (8-499) 243-33-37	P. Skulakov Telephone No. (495) 531-65-15

Form PCT/ISA/210 (second sheet) (January 2015)

Petitioner Samsung Ex-1004, 1383

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 2019/037911

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☒ Claims Nos.: 11-12, 19-20
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
☐ No protest accompanied the payment of additional search fees.

Petitioner Samsung Ex-1004, 1384

Innovation, Sciences et Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and Economic Development Canada
Canadian Intellectual Property Office

CA 3110468 A1 2019/11/21

(21) 3 110 468

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(86) Date de dépôt PCT/PCT Filing Date: 2019/05/15
(87) Date publication PCT/PCT Publication Date: 2019/11/21
(85) Entrée phase nationale/National Entry: 2021/02/23
(86) N° demande PCT/PCT Application No.: US 2019/032371
(87) N° publication PCT/PCT Publication No.: 2019/222319
(30) Priorité/Priority: 2018/05/17 (US62/672,977)

(51) Cl.Int./Int.Cl. *G06F 21/30* (2013.01), *G06F 21/62* (2013.01)

(71) Demandeur/Applicant:
IOT AND M2M TECHNOLOGIES, LLC, US

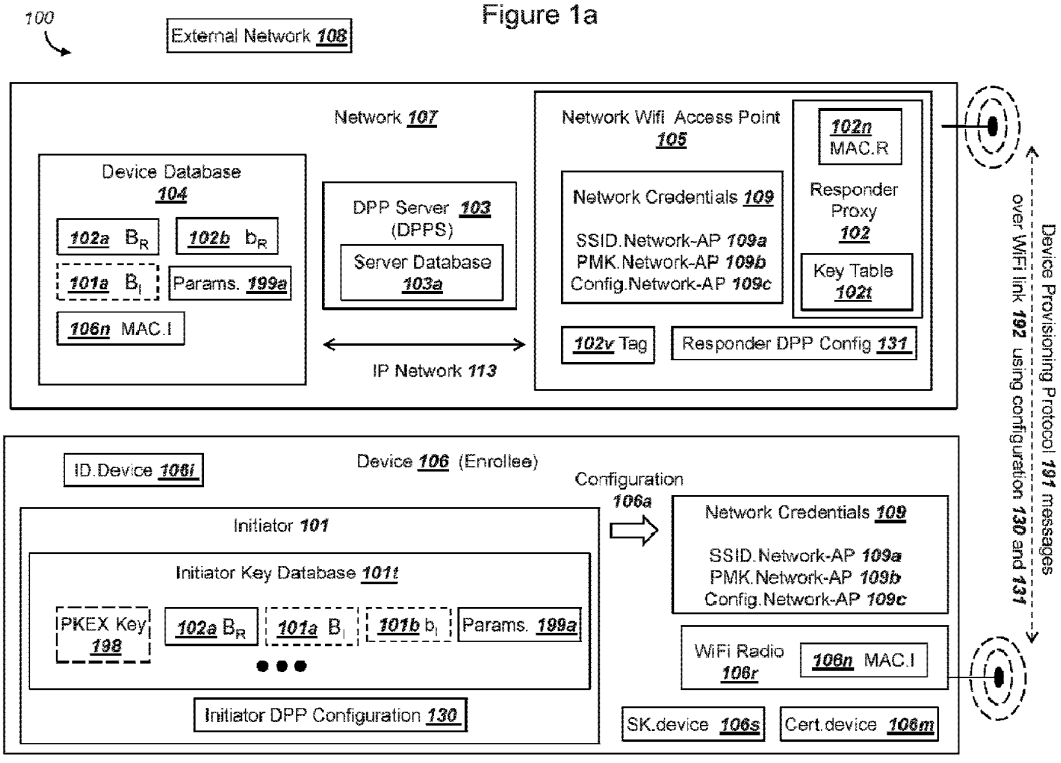
(72) Inventeur/Inventor:
NIX, JOHN A., US

(74) Agent: HICKS & ASSOCIATES

(54) Titre : PROTOCOLE HEBERGE DYNAMIQUE DE FOURNITURE DE SERVEURS ET UN ACCEPTEUR MIS EN RESEAU

(54) Title: A HOSTED DYNAMIC PROVISIONING PROTOCOL WITH SERVERS AND A NETWORKED RESPONDER

Figure 1a



(57) Abrégé/Abstract:
A network can operate a WiFi access point with credentials. An unconfigured device can support a Device Provisioning Protocol (DPP), and record bootstrap public keys and initiator private keys. The network can record bootstrap public and responder private

(21) 3 110 468

(13) A1

(57) Abrégé(suite)/Abstract(continued):
keys and operate a DPP server. A responder proxy can establish a secure and mutually authenticated connection with the network. The network can (i) derive responder ephemeral public and private keys, (ii) record the initiator bootstrap public key, and (iii) select a responder mode for the responder. The network can derive a shared secret with at least the (i) recorded the initiator bootstrap public key and (ii) derived responder ephemeral private key. The network can encrypt credentials using at least the derived shared secret and send the encrypted credentials through the responder proxy to the initiator, which can forward the encrypted credentials to the device, thereby supporting a device configuration.

Petitioner Samsung Ex-1004, 1386

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 November 2019 (21.11.2019)

WIPO | PCT

(10) International Publication Number
WO 2019/222319 A1

(51) International Patent Classification:
G06F 21/30 (2013.01) *G06F 21/62* (2013.01)

(21) International Application Number:
PCT/US2019/032371

(22) International Filing Date:
15 May 2019 (15.05.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/672,977 17 May 2018 (17.05.2018) US

(71) Applicant: **IoT AND M2M TECHNOLOGIES, LLC** [US/US]; 807 Davis Street, Unit 2207, Evanston, IL 60201-7104 (US).

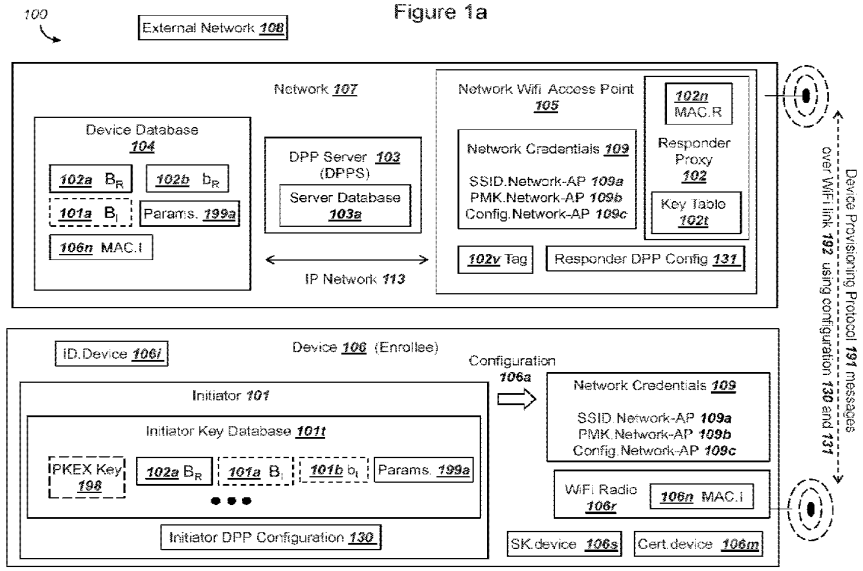
(72) Inventor: **NIX, John, A.**; c/o IoT And M2M Technologies, LLC, 807 Davis Street, Unit 2207, Evanston, IL 60201-7104 (US).

(74) Agent: **THOMSON, Kirsten, L.**; Mcdonnell Boehnen Hulbert & Berghoff LLP, 300 South Wacker Drive, Chicago, IL 60606 (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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, 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,

(54) Title: A HOSTED DYNAMIC PROVISIONING PROTOCOL WITH SERVERS AND A NETWORKED RESPONDER

Figure 1a



(57) **Abstract:** A network can operate a WiFi access point with credentials. An unconfigured device can support a Device Provisioning Protocol (DPP), and record bootstrap public keys and initiator private keys. The network can record bootstrap public and responder private keys and operate a DPP server. A responder proxy can establish a secure and mutually authenticated connection with the network. The network can (i) derive responder ephemeral public and private keys, (ii) record the initiator bootstrap public key, and (iii) select a responder mode for the responder. The network can derive a shared secret with at least the (i) recorded the initiator bootstrap public key and (ii) derived responder ephemeral private key. The network can encrypt credentials using at least the derived shared secret and send the encrypted credentials through the responder proxy to the initiator, which can forward the encrypted credentials to the device, thereby supporting a device configuration.

[Continued on next page]

WO 2019/222319 A1

Petitioner Samsung Ex-1004, 1387

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))*

Petitioner Samsung Ex-1004, 1388

A HOSTED DYNAMIC PROVISIONING PROTOCOL WITH SERVERS AND A NETWORKED RESPONDER

CROSS-REFERENCE TO RELATED APPLICATION

This international PCT application claims the benefit of the filing date of U.S. Provisional Patent Application Serial No. 62/672,977, filed May 17, 2018, which is hereby incorporated by reference in its entirety.

BACKGROUND

Technical Field

The present systems and methods relate to authentication and configuration of WiFi radios, and more particularly, to using a Device Provisioning Protocol with a networked responder and a set of servers that record PKI keys, in order securely transfer a set of credentials to a device.

Description of Related Art

The ability to connect transducers such as sensors and actuators with a wireless network is a growing field with many economical applications. As the costs for both electronic hardware and bandwidth continue to decrease, the use of networked transducers is expected to continue increasing over the coming decades. Connecting transducers to a network can be referred to as "machine-to-machine (M2M) communications" or "the Internet of Things (IoT)." Among many potential benefits, IoT technologies allow automated monitoring and/or control of assets, equipment, personnel, or a physical location where manual monitoring may not be economical. Many applications for the "Internet of Things" significantly reduce costs by using automated monitoring instead of manual techniques. Prominent examples of IoT applications today include monitoring and control for building heating/air-conditioning, automobiles, alarm systems, and tracking devices. Fast growing markets for IoT applications today include health applications such as the remote monitoring of a person's fitness activity, heartbeat, or glucose levels, monitoring of industrial equipment deployed remotely in the field, and also security systems.

Many IoT applications can leverage wireless networking technologies, in addition to wired technologies such as Ethernet. Wireless technologies such as wireless local area networks and wireless wide area networks have proliferated around the world over the past 30 years, and usage of these wireless networks is also expected to continue to grow. Wireless local area network (LAN) technologies include WiFi based on the series of 802.11 standards from the Institute of Electrical and Electronics Engineers (IEEE). The many options to connect a transducer device to a wireless network

Petitioner Samsung Ex-1004, 1389

create opportunities for new products and services, but also create several classes of problems that need to be solved. Many of these problems or needs in the art pertain to securely and efficiently configuring the transducer devices. Without configuration including a set of network access credentials, a device cannot normally securely connect with a wireless network. A need exists in the art to allow a user or a network to securely and easily upload to the device a set of network access credentials for a wireless network.

Manually configuring devices to (i) use different security keys, (ii) apply new network access credentials to connect with the owner's selected wireless network can be time consuming and difficult for users. Entering network access credentials is also prone to errors and especially difficult for devices with limited user interfaces, where a limited user interface is common for many devices supporting applications for the "Internet of Things". Loading credentials and network configuration into a device in order to obtain connectivity to a network can also be known as device provisioning or device configuration. In an attempt to address these needs to simplify or automate device provisioning, the WiFi Alliance ™ released a Device Provisioning Protocol (DPP) specification on April 9, 2018 as version 1.0 (DPPv1.0), which is herein incorporated by reference in its entirety. Although the DPPv1.0 defines a series of messages between an initiator and a responder in order to support a device configuration, several unmet needs remain in the art in order to securely and easily deploy a Device Provisioning Protocol at wide scale with potentially billions of devices over many years.

To highlight one example unmet need by a DPPv1.0, WiFi capable devices currently being designed and deployed may have an expected life of more than a decade. During that timeframe, the device may need to undergo a configuration or device provisioning step more than one time. Reasons could include either (i) the device undergoes a "factory reset", or (ii) the device needs to be provisioned a second time, such as if the device owner changes or the device is moved to a different network, and other reasons exist as well. But, as stated in DPPv1.0, "threats against this bootstrapping method are possible if public keys and/or codes are reused." A need exists in the art to support a Device Provisioning Protocol in a manner where bootstrapping public keys can be securely reused. A need exists in the art for a responder bootstrap public key to be securely shared among a plurality of devices, while the responder bootstrap private key can remain secured.

In order to securely provision a device, a network may prefer to conduct mutual authentication with an initiator operating in the device that records and operates with initiator PKI keys. The device could be a high value piece of equipment, such as an automobile, health monitoring equipment in a hospital, or industrial equipment, etc. that could be targets for potential hackers seeking to "provision" the devices to a network under their control or a network providing the hackers access to the device. In these cases, an owner or user of a device may prefer that a device provisioning protocol use mutual authentication. The DPPv1.0 supports mutual authentication in order to securely authenticate a device and an initiator before transferring network access credentials to the device, but creates a new and

Petitioner Samsung Ex-1004, 1390

shifted problem of both (i) securely providing a responder bootstrap public key to an initiator and (ii) keeping the responder bootstrap private key secure. In other words, recording a responder bootstrap public key for a responder that is located nearby the initiator can be difficult if there is no pre-existing relationship or communication between a device with an initiator and a responder before the start of a device provisioning protocol. In addition, an insecure responder may have no pre-existing or sufficiently secured relationship with a network, yet the network may need to securely transfer credentials for the network to the device through the insecure responder. A need exists in the art to securely and readily enable an initiator to have access to an responder bootstrap public key, while also enabling a responder to have access to the corresponding initiator bootstrap public key. A need exists in the art for a network to securely transfer credentials to a device though an insecure responder.

With the widespread deployment of WiFi access points, a device may have access to potentially a plurality of different access points associated with potentially different networks. Although DPPv1.0 supports configuration of a device, DPPv1.0 does not suggest how a device could be pre-configured to support each of the potential plurality of different networks in order to support a device provisioning protocol. Also, DPPv1.0 does not suggest how a pre-configured device could attempt to communicate with the potential plurality of different access points in order to obtain at least one set of network credentials from the potential plurality of different networks. In other words, a need exists in the art for an initiator to use a pre-configuration in order to automatically and securely receive a set of credentials for the device without having to separately conduct an "out of band" receipt of a responder bootstrap public key when the device is taken to the vicinity of a supported network. A need exists in the art for (i) the credentials to be encrypted in a manner such that the responder cannot feasibly read the credentials, where (ii) the device can receive the encrypted credentials and convert the encrypted credentials into plaintext form in order for the device to load and use the plaintext credentials with a network.

A network can use DPPv1.0 in order to connect devices to the network using a responder. The security for conducting a device provisioning protocol according to DPPv1.0 can depend on the security of a responder, since as envisioned by DPPv1.0 the responder (i) records a responder bootstrap private key and (ii) conducts key exchanges with the responder bootstrap private key. However, a responder for conducting a DPP may be outside the control of a network and may also be an insecure device. A responder could be a "rooted" mobile phone operated by a hacker or also a "rouge" access point, as a couple of examples, and many other examples exist as well. Or, the responder could simply operated by a computing device that is insecure or operates with security policies that do not meet the standards of the network. A need exists in the art for an insecure responder to conduct a device provisioning protocol that is compatible with DPPv1.0, while simultaneously keeping (i) the responder bootstrap private key and (ii) cryptographic operations with the responder bootstrap private key under the control or within the security policies of a network providing credentials to the device.

Petitioner Samsung Ex-1004, 1391

Many other examples exist as well for needs in the art to support efficient yet secure device configuration or provisioning, and the above are examples are just a few and intended to be illustrative instead of limiting.

Petitioner Samsung Ex-1004, 1392

SUMMARY

Methods and systems are provided for a networked responder proxy to conduct a device provisioning protocol (DPP) with an initiator operating in a device. The device provisioning protocol can support the Device Provisioning Protocol specification version 1.0 and subsequent versions from the WiFi Alliance™. Conducting a DPP with a device can transfer a set of network credentials to the device, in order for the device to connect with a network using the network credentials. The set of network credentials can also include a network configuration, where the network configuration includes settings to use with a WiFi radio in the device, in order for the WiFi radio to communicate with other nodes with the credentials, including an access point which can provide connectivity to an IP network for the device.

A device can operate an initiator, which periodically automatically transmits DPP authentication request messages until receiving a DPP authentication response from a responder proxy. The initiator can record and operate an initiator configuration, specifying a frequency band to utilize, a wireless networking technology to implement (e.g. 802.11 ac, 802.11ah, 4G LTE, 5G, etc.) and a channel list. The wireless networking technology for an initiator and a responder proxy can support a wide-area network (WAN) as well, such as 4G and 5G networks associated with public land mobile networks. The initiator can record an initiator bootstrap public key, an initiator bootstrap private key, and, when supporting mutual authentication with a responder, a responder bootstrap public key. The PKI keys can be associated with a selected set of cryptographic parameters, where the parameters specify settings to use for cryptographic operations by the initiator. The initiator can also include a random number generator, a PKI key pair generation algorithm, a secure hash algorithm, an elliptic curve Diffie Hellman key exchange algorithm, a symmetric ciphering algorithm, a digital signature algorithm, and an asymmetric ciphering algorithm.

A network can operate an access point, where the access point operates with a set of credentials. The credentials can include an identifier for the network such an SSID, a pairwise master key (PMK) or a pre-shared key (PSK), a pairwise master key identifier (PMKID), and also a configuration. Credentials for other authentication schemes between a device and an access point could be supported as well, including EAP-TLS, EAP-PSK, EAP-AKA, EAP-AKA', etc. The access point can support wireless networking standards in the series of 802.11 standards. A network can also operate a plurality of access points. The network can operate or be associated with a Device Provisioning Protocol (DPP) server, and a device database. The network can be connected to an IP network, which can comprise the globally routable public Internet. Different servers within a network can be connected using an Intranet, a virtual private network (VPN), Internet Protocol Security (IPSEC), or similar techniques to secure communication between the servers. The access point for a

Petitioner Samsung Ex-1004, 1393

network can also comprise a enodeb for 4G wireless technology or a gnodeb for 5G wireless networking technology. The credentials with 4G networking technology can include at least a key K and the credentials with 5G networking technology can include at least a key Kausf.

The network can operate or communicate with the device database. The DPP server in the network can also communicate with the device database. The device database can record information for a plurality of devices before initiators in the device conduct the device provisioning protocol with a responder proxy. The device database can record the device identity, an initiator identity, which could be in the form of an initiator media access control (MAC) address, and a key table. The key table in a device database can record for each device an initiator bootstrap public key, a responder bootstrap public key, and a responder bootstrap private key. Recorded data within a device database can be obtained by a network (i) upon manufacturing from a device manufacturer, or (ii) before distribution of a device to the location of the network's access point.

The DPP server for a network can operate a server database, where the server database records information for a plurality of devices connecting with the network. The server database can record the device identity, media access control (MAC) addresses for devices and initiators, a responder mode, a selected set of cryptographic parameters for each device, a responder bootstrap private key, the initiator bootstrap public key, the responder bootstrap public key, a responder ephemeral public key, and responder ephemeral private key, a public key exchange (PKEX) key, a first, second, and third derived shared secret keys for an initiator. Depending on the responder mode, not all PKI keys and secret shared keys from the previous sentence may be recorded in the server database. The DPP server can also include a set cryptographic algorithms, where the cryptographic algorithms can include a random number generator, a PKI key pair generation algorithm, a secure hash algorithm, an elliptic curve Diffie Hellman key exchange algorithm, a symmetric ciphering algorithm, a digital signature algorithm, and an asymmetric ciphering algorithm. The cryptographic algorithms within a DPP server can be compatible with the equivalent set of cryptographic algorithms in the responder and also the initiator.

A responder can be a computing device that includes a WiFi radio and can be associated with a network. The responder can operate the WiFi radio with a responder configuration in order for the responder to send and receive messages with the initiator. The responder can be located in sufficient proximity with the initiator so that the WiFi radio for the initiator and the WiFi radio for the responder can communicate using 802.11 standards. The responder can operate a DPP application, where the DPP application can also include the set of cryptographic algorithms, which are compatible with cryptographic algorithms in both the initiator and the DPP server. The responder can use an IP network in order to communicate with the network via the IP network.

In a first responder mode, the responder can establish a secure, mutually authenticated connection with a DPP server operating in a network. The DPP server can send configuration data to

Petitioner Samsung Ex-1004, 1394

the responder in preparation for the responder operating as a responder proxy to receive incoming DPP authentication requests from at least one initiator. The responder can conduct a responder configuration step after receiving the responder configuration data from the DPP server. An initiator can receive and/or record a responder bootstrap public key and conduct a series of steps to transmit a DPP authentication request, potentially as a broadcast message on a WiFi channel on which the device operating the responder listens. The initiator can record an initiator key table and an initiator DPP configuration to create and transmit a series of DPP authentication request messages on different channels within the initiator DPP configuration. The responder can receive at least one DPP authentication request message transmitted by the initiator using a channel in a responder DPP configuration. The responder can forward data from the DPP authentication request received from the initiator to the DPP server through the secure session.

Continuing in the first responder mode, the DPP server can then conduct a series of steps in order to process data in the Device Provisioning Protocol (DPP) authorization request received by the responder operating as a responder proxy. The DPP server can derive a responder ephemeral PKI key pair. The DPP server can query a device database for data for the device using either the initiator media access control (MAC) address or the hash value of the initiator bootstrap public key from the DPP authentication request message. The DPP server can conduct a first responder elliptic curve Diffie-Hellman (ECDH) key exchange with the initiator bootstrap public key and the derived responder private key to derive a key k1. The DPP server can use the derived key k1 to decrypt ciphertext from the DPP authentication message and read a plaintext initiator authentication value.

Continuing in the first responder mode, the DPP server can conduct a second responder ECDH key exchange with the received initiator ephemeral public key and the derived responder private key to derive a key k2. The DPP server can conduct a third responder ECDH key exchange with the initiator bootstrap public key and a combination of the derived responder private key and the responder bootstrap private key in order to derive a key ke. The DPP server can derive a responder nonce and calculate a responder authentication value using the received and decrypted initiator authentication value. The DPP server can encrypt the responder authentication value into a first responder ciphertext using the derived key ke. The DPP server can (i) encrypt the first responder ciphertext and the responder nonce and the initiator nonce using the derived key k2 into (ii) a second responder ciphertext. The DPP server can send the responder a message with the second responder ciphertext and the derived responder ephemeral public key. The responder operating as a responder proxy can send the initiator a DPP authentication response message, which includes the second responder ciphertext and the responder ephemeral public key. Note that the responder operating as a responder proxy does not need to record the responder bootstrap private key.

Continuing in the first responder mode, the initiator can receive the DPP authentication response from the responder operating as a responder proxy and perform a series of steps to decrypt

Petitioner Samsung Ex-1004, 1395

the received second responder ciphertext. The initiator can conduct an initiator ECDH key exchange using the responder ephemeral public key and the initiator ephemeral private key in order to derive the key k2. The initiator can conduct a second initiator key exchange using (i) a combination of the responder bootstrap public key and the responder ephemeral public key and (ii) the initiator bootstrap private key and in order to derive the key ke. The initiator can decrypt the second responder ciphertext using the derived key k2. The initiator can read the plaintext responder nonce and the first responder ciphertext. The initiator can decrypt the first responder ciphertext using the derived key ke. The initiator can read the plaintext responder authentication value from the first responder ciphertext. The initiator can calculate the responder authentication value, and if the received responder authentication value matches the calculated responder authentication value, then the responder is authenticated.

The initiator can calculate an initiator authentication value using at least the received and decrypted responder nonce. The initiator can encrypt the initiator authentication value using the key ke, thereby creating an initiator ciphertext. The initiator can send a DPP authentication confirm message to the responder with the initiator ciphertext. The responder operating as a responder proxy can forward the initiator ciphertext to the DPP server through the secure session. The DPP server can decrypt the initiator ciphertext using the derived key ke and read the plaintext initiator authentication value. The DPP server can calculate the same initiator authentication value, and if the received, decrypted initiator authentication value matches the calculated authentication value then the initiator is authenticated with the DPP server and the responder proxy.

After the mutual authentication, the responder can begin operating as a configurator and the initiator can begin operating as an enrollee. The initiator can generate an enrollee nonce and also a configuration attribute, where the configuration attribute describes WiFi capabilities for the device. The initiator can create a second initiator ciphertext by encrypting the enrollee nonce and the configuration attribute with the derived key ke. The initiator can send a DPP configuration request to the responder, where the DPP configuration request can include the second initiator ciphertext. The responder operating as a responder proxy can send the second initiator ciphertext to the DPP server.

The DPP server can receive the second initiator ciphertext from the responder operating as a responder proxy and decrypt the second initiator ciphertext using the derived key ke, in order to read the plaintext enrollee nonce and the configuration attribute. Using the configuration attribute, the DPP server can select a set of network credentials for the device operating the initiator. The DPP server can encrypt the received enrollee nonce and the network credentials using the derived key ke, creating a third responder ciphertext. The DPP server can send the third responder ciphertext to the responder operating as a responder proxy. The responder can forward the third responder ciphertext to the initiator in a DPP configuration response message.

The initiator can receive the DPP configuration response message and decrypt the third responder ciphertext using the derived key ke in order to read the plaintext network credentials. The

Petitioner Samsung Ex-1004, 1396

initiator can pass the network credentials to the device, which can apply the credentials with the WiFi radio for the device. The device can subsequently connect with a network access point using the received network credentials. The network access point can provide connectivity to an IP network for the device. The DPP server and the device can record that the DPP session is successfully completed.

These as well as other aspects and advantages will become apparent to those of ordinary skill in the art by reading the following detailed description, with reference where appropriate to the accompanying drawings.

Petitioner Samsung Ex-1004, 1397

BRIEF DESCRIPTION OF THE DRAWINGS

Various exemplary embodiments are described herein with reference to the following drawings, wherein like numerals denote like entities.

Figure 1a is a graphical illustration of an exemplary system, where a network operates a responder and communicates with an initiator in a device, in order to conduct a device provisioning protocol, in accordance with exemplary embodiments;

Figure 1b is a graphical illustration of a device provisioning protocol for authentication of a responder and configuration of an initiator, in accordance with conventional technology;

Figure 1c is a graphical illustration of a device provisioning protocol for mutual authentication of an initiator and a responder, and configuration of an initiator, in accordance with conventional technology;

Figure 1d is an illustration of an exemplary initiator DPP configuration and initiator key database, with tables recording exemplary data for a plurality of bootstrap responder public keys and bootstrap initiator private keys, in accordance with exemplary embodiments;

Figure 1e is an illustration of an exemplary responder bootstrap public keys and initiator bootstrap PKI keys recorded by a plurality of devices, in accordance with exemplary embodiments;

Figure 1f is a graphical illustration of hardware, firmware, and software components for a responder, including a responder configuration step, in accordance with exemplary embodiments;

Figure 2a is an illustration of an exemplary device database, with tables for a device database recording exemplary data, in accordance with exemplary embodiments;

Figure 2b is a flow chart illustrating exemplary steps for an initiator to use an initiator DPP configuration and initiator key database in order to automatically connect with a responder proxy, in accordance with exemplary embodiments;

Figure 3a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device database, a DPP server, a responder proxy, and an initiator, in accordance with exemplary embodiments;

Figure 3b is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a DPP server, a responder proxy, and an initiator, in accordance with exemplary embodiments;

Figure 3c is an illustration of an exemplary server database, with tables for a server database recording exemplary data, in accordance with exemplary embodiments;

Figure 3d is a flow chart illustrating exemplary steps for creating and verifying a digital signature using PKI keys, parameters, and data input, in accordance with exemplary embodiments;

Petitioner Samsung Ex-1004, 1398

Figure 3e is a flow chart illustrating exemplary steps for using asymmetric ciphering in order to encrypt credentials using a public key and decrypting credentials using the corresponding private key, in accordance with exemplary embodiments;

Figure 4a is a flow chart illustrating exemplary steps for conducting a key exchange using PKI keys in order to derive a shared secret key, and for using the derived shared secret key to encrypt and decrypt data, in accordance with exemplary embodiments;

Figure 4b is a flow chart illustrating exemplary steps for conducting a key exchange using PKI keys in order to derive a shared secret key, and for using the derived shared secret key to encrypt and decrypt data, in accordance with exemplary embodiments;

Figure 4c is a flow chart illustrating exemplary steps for conducting a key exchange using PKI keys in order to derive a shared secret key, and for using the derived shared secret key to encrypt and decrypt data, in accordance with exemplary embodiments;

Figure 4d is a flow chart illustrating exemplary steps for conducting key exchanges using PKI keys in order to derive a first and a second shared secret keys, where ECC point addition is performed on output from a key exchange algorithm in order to derive a third shared secret key;

Figure 5a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device database, a DPP server, a responder proxy, and an initiator, in accordance with exemplary embodiments;

Figure 5b is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device database, a DPP server, a responder proxy, and an initiator, in accordance with exemplary embodiments;

Figure 6 is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a DPP server, a responder proxy, and an initiator, in accordance with exemplary embodiments;

Figure 7 is a graphical illustration of an exemplary system, where an access point with an initiator communicates with a responder proxy using a hosted device provisioning protocol, in accordance with exemplary embodiments; and,

Figure 8 is a graphical illustration of an exemplary system, where an access point operates as a responder proxy for a device with an initiator, in order transfer a set of network credentials to the device, in accordance with exemplary embodiments.

Petitioner Samsung Ex-1004, 1399

DETAILED DESCRIPTION

Figure 1a

Figure 1a is a graphical illustration of an exemplary system, where a network operates a responder and communicates with an initiator in a device, in order to conduct a device provisioning protocol, in accordance with exemplary embodiments. The system 100 can include a network 107 and a device 106. Network 107 can operate a WiFi access point 105, a DPP server 103, and a device database 104. The nodes within network 107 can communicate via an IP network 113. Device 106 can be a computing device operating an initiator 101 and a WiFi radio 106r. Device 106 can include a device identity 106i for a network 107 to track or identify a device 106 across a plurality of different devices 106. Or, in exemplary embodiments a media access control (MAC) address MAC.I 106n for device 106 can be used as a device identity.

Responder proxy 102 can operate as a responder and send and receive messages with device 106 operating an initiator 101 according to a device provisioning protocol (DPP) 191 as specified in the "Device Provisioning Protocol Specification version 1.0" (DPPv1.0) from the WiFi Alliance ™, which is hereby incorporated by reference in its entirety. Subsequent and related versions of a DPP 191 could be supported as well by initiator 101 in device 106 and a responder proxy 102. A summary of the message flows between responder proxy 102 and device 101 according to the DPP 191 are depicted and described in connection with Figure 1b and Figure 1c below, as well as subsequent figures herein. The message flows for a DPP 191 between responder proxy 102 and initiator 101 could be via a WiFi data link 192, where the WiFi data link 192 operated by (i) initiator 101 can use an initiator DPP configuration 130 and (ii) responder proxy 102 can use a responder DPP configuration 131. Responder proxy 102 can be associated with a tag 102v, where tag 102v can record a responder bootstrap public key Br 102a.

Tag 102v can be in the form of a QR code or NFC tag, which could be read by device 106 in order to obtain the responder bootstrap public key Br 102a "out of band". Or, in some embodiments tag 102v can be optionally omitted, where an initiator 101 can receive or record responder bootstrap public key Br 102a via other means, such as during manufacturing or distribution of device 106 where a device 106 could record a plurality of different responder bootstrap public keys Br 102a in a responder public key table 101ta depicted and described in connection with Figure 1d below.

Responder proxy 102 could operate a WiFi radio as depicted, where the WiFi radio could record a media access control (MAC) address MAC.R 102n. In exemplary embodiments, a wireless device could operate the responder proxy 102, where the wireless device could comprise a mobile handset, a smartphone, a wireless router, a tablet computer, and/or a wireless gateway. Other

Petitioner Samsung Ex-1004, 1400

possibilities exist as well for the embodiments of a wireless device operating a responder proxy 102 without departing from the scope of the present disclosure. MAC.R 102n for responder 102 could comprise a string of octets to identify responder 102 with any of a WiFi access point 105, device 106, network 107, and DPP server 103. In order to acquire the data to function as a responder according to a DPP 191 and DPPv1.0, responder 102 in Figure 1a can communicate and transmit/receive data from network 107 via IP network 113. Additional details regarding the electrical components within a responder 102 or WiFi access point 105 operating a responder 102 are depicted and described in connection with Figure 1f below. The responder proxy 102 can also include a set cryptographic algorithms, where the cryptographic algorithms can include a random number generator, a PKI key pair generation algorithm, a secure hash algorithm, an elliptic curve Diffie Hellman key exchange algorithm, a symmetric ciphering algorithm, a digital signature algorithm, and an asymmetric ciphering algorithm. The cryptographic algorithms within a responder proxy 102 can be compatible with the equivalent set of cryptographic algorithms in the initiator 101 and also the DPP server 103.

Embodiments of the present disclosure contemplate both (i) some of the functionality of a responder in the DPP specification version 1.0 operates within responder proxy 102 depicted in Figure 1a, and (ii) other functionality of a responder according to the DPP protocol can be performed or conducted by DPP server 103. In other words, (x) from the perspective of device 106, responder proxy 102 can operate as a responder according to the DPP Specification, but (y) in the present disclosure some of the functionality and data for a responder as contemplated by the DPP Specification remains in the DPP server 103. Depictions and descriptions below in the present disclosure may utilize the term "responder 102" or configured "responder 102' ", which can also comprise a "responder proxy 102" depicted in Figure 1a. As contemplated herein, a "responder proxy 102' " can be equivalent to "responder 102'."

This configuration of operating a responder proxy 102 in a system 100 (along with proper operation of a network 107) can address and solve the needs in the art described above. A responder proxy 102 can communicate with DPP server 103 via IP network 113. Although a responder proxy 102 is depicted as operating in WiFi access point 105 in Figure 1a, other embodiments of the present disclosure contemplate that different WiFi capable devices than access point 105 could operate a responder proxy 102, such as a mobile phone 701 depicted in Figure 7 below. Other possibilities exist as well for a WiFi capable device to operate a responder proxy 102 without departing from the scope of the present disclosure. In addition, although the use of a WiFi radio is depicted in Figure 1a and other figures herein, other wireless networking technologies and radios besides WiFi could be used with responder proxy 102 and device 106 with initiator 101. As one example, (i) a responder proxy could operate with a wireless wide area network (WAN) radio such as a "gnodeb" with a 5G network and (ii) device 106 with initiator 101 could operate a radio module supporting 5G networking standards.

Petitioner Samsung Ex-1004, 1401

IP network 113 could be either a Local Area Network (LAN) or a Wide Area Network (WAN), or potentially a combination of both. IP network 113 could comprise a packet switched network supporting either IEEE 802.11 (WiFi) standards and/or IETF standards for IP networking such as IPv4 or IPv6. Responder 102 within access point 105 and network 107 could also utilize a variety of WAN wireless technologies to communicate, including Low Power Wide Area (LPWA) technology, 3rd Generation Partnership Project (3GPP) technology such as, but not limited to, 3G, 4G Long-Term Evolution (LTE), or 4G LTE Advanced, NarrowBand – Internet of Things (NB-IoT), LTE Cat M, proposed 5G networks, and other examples exist as well. A wired responder proxy 102 can connect to the IP network 113 via a wired connection such as, but not limited to, an Ethernet, a fiber optic, or a Universal Serial Bus (USB) connection (not shown). In some exemplary embodiments, access to an IP network 113 for responder proxy 102 can be provided by access point 105, where access point 105 may also record and operate with a set of network credentials 109.

IP network 113 could also be a public or private network supporting Internet Engineering Task Force (IETF) standards such as, but not limited to, such as, RFC 786 (User Datagram Protocol), RFC 793 (Transmission Control Protocol), and related protocols including IPv6 or IPv4. A public IP network 113 could utilize globally routable IP addresses, and a private IP network 113 could utilize private IP addresses which could also be referred to as an Intranet. Other possibilities for IP Network 113 exist as well without departing from the scope of the disclosure. Although not depicted in Figure 1a, elements within a system 100 could each operate or be associated with a firewall, such that packets from IP network 113 are filtered or restricted in order to secure the communication between the nodes.

Device 106 can comprise a computing device with a WiFi radio for transmitting and receiving data such as via WiFi data link 192. Device 106 can include a WiFi radio 106r. Radio 106r and radio 102i for responder proxy 102 can also support other wireless technologies besides WiFi, including 4G and 5G networks. WiFi radio 106r can operate with a media access control (MAC) address MAC.I 106n. Responder 102 can read the value MAC.I 106n from messages transmitted by device 106 over WiFi link 192. Device 106 can operate an initiator 101 and, after a configuration step 106a, also operate a set of network WiFi credentials 109 that are compatible with an access point 105. Device 106 could be powered via any of (i) traditional "wall power" potentially with an AD/DC adapter, (ii) a battery which may be periodically recharged or replaced, (iii) power over a wired LAN connection such as "power over Ethernet", and other possibilities exist as well. A device 106 can record a certificate cert.device 106m, which can record a device public key, which can be associated with a device private key SK.device 106s also recorded in device 106. For the system depicted in Figure 1a, device 106 operating an initiator 101 can have a roll of an enrollee, and responder proxy 102 can operate with a roll of a configurator.

The initiator 101 in device 106 can record an initiator key database 101t, which could comprise a database such as SQLlite, or alternatively simply a text file of keys and identities, and other

Petitioner Samsung Ex-1004, 1402

possibilities exist as well for the form and structure of keys recorded in an initiator 101. In exemplary embodiments, the initiator key database 101t can be stored in a secure processing environment or a secure enclave for a CPU operating in device 106. The initiator key database 101t can optionally include an initiator bootstrap public key Bi 101a, an initiator bootstrap private key bi 101b (for embodiments supporting a mutual authentication 142 depicted in Figure 1c below), a responder bootstrap public key Br 102a, a set of cryptographic parameters 199a for the PKI keys, and optionally a public key exchange protocol (PKEX) shared secret key 198. The initiator 101 in device 106 can comprise a process or running software program operated by a processor and recorded in memory for device 106. Although not depicted in Figure 1a, an initiator 101 can also include a random number generator, a PKI key pair generation algorithm, a secure hash algorithm, an elliptic curve Diffie Hellman key exchange algorithm, and a symmetric ciphering algorithm, a digital signature algorithm, and an asymmetric ciphering algorithm. The cryptographic algorithms within an initiator 101 can be compatible with the equivalent set of cryptographic algorithms in a responder proxy 102 and also DPP server 103.

A network 107 can operate an access point 105, where the access point operates with a set of credentials 109. The credentials 109 can include an identifier for the network such an SSID.network-AP 109a, at least one pairwise master key (PMK) PMK.network-AP 109b or a pre-shared key (PSK), a pairwise master key identifier (PMKID) (not shown), and also a configuration Config.network-AP 109c. The access point can support wireless networking standards in the series of 802.11 standards. Credentials 109 for other authentication schemes between a device and an access point could be supported as well, including EAP-TLS, EAP-PSK, EAP-AKA, EAP-AKA', etc. A network 107 can also operate a plurality of access points 105. In addition, access point 105 can comprise a plurality of different WiFi radios to collectively function as one access point 105, such as with "mesh" WiFi networks. Or, access point 105 can operate as a single WiFi router. The network 107 can include or be associated with a Device Provisioning Protocol (DPP) server 103, and a device database 104. Different servers within the network 107 can be connected using an Intranet, a virtual private network (VPN), Internet Protocol Security (IPSEC), or similar techniques to secure communication between the servers.

The network 107 can operate or communicate with the device database 104. The DPP server 103 in the network 107 can also communicate with the device database 104. The device database 104 can record information for a plurality of devices 106 before an initiator 101 for the device 106 conducts the device provisioning protocol 191 with the responder proxy 102. The device database 104 can record an identity for device 106, which could be in the form of either (i) an initiator media access control (MAC) address MAC.I 106n (depicted) or a hash value 101aa for the initiator bootstrap public key Bi 101a (where the hash value 101aa is not shown in Figure 1a but is shown in Figure 1b and other Figures below).

Petitioner Samsung Ex-1004, 1403

A device database 104 can record at least one responder bootstrap public key 102a with a corresponding responder bootstrap private key 102b, and optionally an initiator bootstrap public key 101a for embodiments supporting a mutual authentication depicted in Figure 1c and other Figures below. Recorded data within a device database 104 can be obtained by a network 107 from either (i) upon manufacturing from a device manufacturer, or (ii) before distribution of a device to the location of the network's access point. Some data in a device database 104 could also be acquired by a responder 102 or a network 107 prior to the start of a DPP 191. For example, the derivation and recording of responder bootstrap private key br 102b can be independent of recording data in a database 104 for a device 106 (such as MAC.I 106n, key Bi 101a, etc.). Device database 104 can also record a set of cryptographic parameters 199a associated with the PKI keys for responder 102 and initiator 101. Additional exemplary data for a device database 104 is depicted and described in connection with Figure 2a below.

The DPP server 103 for a network 107 can operate a server database 103a, where the server database 103a records information for a plurality of devices 106 and a plurality of responders proxies 102 connecting with the network 107. A server database 103a can also record data for initiators 101 in devices 106, as depicted and described in connection with Figure 3c below. Depending on a responder mode 306a in Figures 2a and 3c below, only selected sets of PKI keys and secret shared keys for initiators 101 may be recorded in the server database 103a. The DPP server 103 can also include a set cryptographic algorithms, where the cryptographic algorithms can include a random number generator, a PKI key pair generation algorithm, a secure hash algorithm, an elliptic curve Diffie Hellman key exchange algorithm, a symmetric ciphering algorithm, a digital signature algorithm, and an asymmetric ciphering algorithm. The cryptographic algorithms within a DPP server 103 can be compatible with the equivalent set of cryptographic algorithms in the initiator 101 and also the responder proxy 102.

Note that the security of communications between a device 106 and an access point 105 using credentials 109 may generally depend on the security of a configuration step 106a, which in turn may rely on a device provisioning protocol 191. Further, a manufactured device 106 may not have the credentials 109 or configuration in order to communicate with an access point 105 without successfully completing a DPP 191 and a configuration step 106a (or other manual or cumbersome means) in order for device 106 to securely receive and use network credentials 109. After a DPP 191 and a configuration step 106a described in the present disclosure and in multiple Figures below, device 106 can communicate with access point 105 using credentials 109.

Network 107 can also be associated with an external network 108, which can comprise a network similar to network 107. External network 108 could operate a plurality of access points 105, and in exemplary embodiments although the device provisioning protocol 191 is conducted with network 107, the credentials 109 transferred to device 106 in a configuration step 106a may belong to

Petitioner Samsung Ex-1004, 1404

the external network 108, and the device 106 could connect with an access point operated by the external network 108 using the credentials 109 instead of connecting with access point 105 operated by network 107. For these embodiments, network 107 could receive the credentials 109 for device 106 as asymmetrically encrypted credentials 109' (described in Figure 3b below) from the external network 108.

Figure 1b

Figure 1b is a graphical illustration of a device provisioning protocol for authentication of a responder and configuration of an initiator, in accordance with conventional technology. Figure 1b depicts a summary of the WiFi Device Provisioning Protocol (DPP) specification, version 1.0 which was published on April 9, 2018. The summary depicted in Figure 1b highlights recorded bootstrap PKI keys, derived ephemeral PKI keys, and messages transmitted and received between an initiator 101 and a responder 102*. The use of keys and messages for a DPP 191 can accomplish both (i) authenticating the responder 102* and (ii) transfer a configuration object to initiator 101 where the configuration object could comprise a set of network access credentials 109 from Figure 1a. Responder 102* is depicted with a "*" and has several differences than a responder proxy 102 depicted in Figure 1a and subsequent Figures below, although some functionality is shared between a responder 102* in Figure 1b and responder proxy 102 herein. Both responder 102* and responder proxy 102 (from Figure 1a and other Figures below) can send and receive the full and complete set of a DPP messages for a Device Provisioning Protocol 191. As contemplated herein, a responder 102* can comprise a responder operating according to the conventional technology depicted in Figure 1b and Figure 1c, while a responder 102 or responder 102' or responder proxy 102' can (i) operate as a proxy for DPP 191, and (ii) record and operate on different PKI keys than those depicted for responder 102* in Figure 1b.

In other words, responder 102* and a responder proxy 102 can send equivalent messages such that a initiator 101 (i) would process the messages in the same manner and (ii) would not normally be able to detect to discern whether a responder proxy 102 or an responder 102* was transmitting and receiving messages for the Device Provisioning Protocol 191. In the present disclosure, responder proxy 102 (not depicted in Figure 1b, but depicted in Figure 1a, Figure 1f, Figure 3a, etc.) can send and receive DPP messages 122, 123, etc. with initiator 101 as an responder according to the DPP Specification 1.0. Responder 102* or a responder proxy 102 can be a process, program, firmware, or software application operating in a computing device with a WiFi compatible radio, in order to communicate with an initiator 101 in a device 106 via a WiFi

Petitioner Samsung Ex-1004, 1405

data link 192 (in Figure 1a). Exemplary additional configurations for an initiator 101 and a responder proxy 102 within computing devices is depicted and described in connection with Figures 7 and 8 below, and other possibilities exist as well without departing from the scope of the present disclosure.

Other differences between a responder 102* and a responder proxy 102 can exist in the present disclosure. For a system 100 with a responder proxy 102 in Figure 1a and Figures below, the recordation of PKI keys, PKI key derivation, and cryptographic processes such as symmetric encryption and decryption can be distributed between a responder proxy 102 and a DPP server 103. However, a responder 102* can internally record the keys necessary for symmetric key exchange, key derivation, and symmetric encryption and decryption, as contemplated by DPP specification version 1.0. The distributed storage and operation with PKI keys for a system that includes a responder proxy 102 is depicted and described in more detail in connection with Figure 3a, Figure 3b, Figure 3c, Figure 5a, Figure 5b, and Figure 6 below. Additional details for internal components, memory structures, values, and operation of a responder proxy 102 in order to provide equivalent functionality as responder 102* for an initiator 101 are described in Figures below.

Initiator 101 can record ephemeral keys and derive symmetric encryption keys in order to authenticate the responder 102* and receive the network access credentials 109 according to a Device Provisioning Protocol 191. For a system with responder authentication 141, an initiator 101 can record a media access control (MAC) address MAC.I 106n. MAC.I 106n can (i) be associated with a WiFi radio operating in device 106 that also operates initiator 101 and (ii) comprise a string of octets in order to uniquely identify initiator 101 and/or device 106 within a wireless network. MAC.I 106n can also comprise an identity for device 106 with network 107 in Figure 1a. With current, convention technology, MAC.I 106n can comprise a string of 6 octets, although other possibilities exist as well for a MAC.I 106n. Initiator 101 can also operate a key pair generation function 101y in order to derive ephemeral PKI keys such as an initiator ephemeral public key Pi 101c and an initiator ephemeral private key pi 101d. As contemplated herein, an ephemeral PKI key can also be referred to as a "protocol" key. In other words, key Pi 101c can be referred to as either (i) an initiator protocol public key Pi 101c or (ii) an initiator ephemeral public key Pi 101c, which can both represent the same numeric value, number, or sequence of digits.

A key pair generation function 101y or 102y could use (i) a random number generator in order to derive the PKI private key and (ii) a selected set of cryptographic parameters 199a (depicted in Figure 1a above and Figure 2a below and other Figures herein) in order to specify an

Petitioner Samsung Ex-1004, 1406

elliptic curve cryptography (ECC) curve name, key length, key formatting, etc. As contemplated herein, the algorithms used with ECC PKI keys in the present disclosure can be compatible and substantially conform with ECC algorithms and keys as specified in (i) the IETF Request for Comments (RFC) 6090 titled "Fundamental Elliptic Curve Cryptography Algorithms", and (ii) IETF RFC 5915 titled "Elliptic Curve Private Key Structure", and also subsequent and related versions of these standards.

With a random number generated by responder 102* or a responder proxy 102, a key pair generation function 102y, a derived ephemeral responder private key pr 102r can be used to derive a corresponding ephemeral initiator public key Pr 102p. The two derived PKI keys can be used in conjunction for subsequent operations such as an Elliptic Curve Diffie Hellman key exchange and also other cryptographic algorithms such as a key derivation function and a digital signature algorithm. As contemplated in the present disclosure, the use of a capital letter as the first character for a PKI key can represent a public key, the use of a lower case letter as the first character for a PKI key can represent a private key. As contemplated in the present disclosure, the second letter for a PKI key can represent the entity the key is associated with or belongs to (e.g. "r" for responder 102 and "i" for initiator 101). Further, the letter "B" or "b" as the first character for a PKI key can represent a recorded bootstrap key, which could be static or relatively static over the lifetime of a responder or initiator, and the letter "P" or "p" the first character for a PKI key can represent an ephemeral key, which can also be referred to as a protocol key.

Responder 102* in Figure 1b and Figure 1c can include functionality to operate as a responder according to the DPP Specification version 1.0 for the purpose or transmitting and receiving messages according to the DPP protocol 191. For a DPP 191, a responder 102* can record a responder bootstrap public key Br 102a, a responder bootstrap private key 102b, a MAC address MAC.R 102n, and also operate a key pair generation algorithm 102y. A responder bootstrap public key Br 102a and responder bootstrap private key br 102b recorded in responder 102* could comprise an ECC key pair selected and formatted to be compatible with the set of parameters 199a for initiator 101 (in Figure 1a above). Ephemeral bootstrap PKI keys for responder 102* could also be formatted and selected with a compatible set of cryptographic parameters 199a for Pi 101c and pi 101d. Bootstrap PKI keys for responder 102* could be recorded in responder 102* or nonvolatile memory for a WiFi enabled computing device operating a responder 102*. With the conventional technology depicted in Figure 1b, the bootstrap PKI keys could be recorded in responder 102* during manufacturing or a configuration of responder 102* before receiving the DPP authentication request message 122. MAC address MAC.R 102n

Petitioner Samsung Ex-1004, 1407

can comprise an identity of responder 102* or the computing device operating responder 102* on a WiFi network, and can be similar to MAC.I 106n for an initiator 101 as described above.

An initiator 101 and a responder 102* can conduct a Device Provisioning Protocol (DPP) 191. Again, as contemplated herein, an exemplary Device Provisioning Protocol 191 according to conventional technology could comprise the "Device Provisioning Protocol Specification" version 1.0 as published by the WiFi Alliance ™. In message 121, an initiator 101 can receive the responder bootstrap public key Br 102a for responder 102*. The "out of band" transfer of Br 102a in message 121 could comprise several different possible methods for initiator 101 to receive Br 102a, which are also described in the DPP specification 1.0, such as via reading a QR code on the device operating responder 102*, reading a near-field communications tag, using a Bluetooth wireless connection to read Br 102a from responder 102*, and other possibilities exist as well. The data could be read via a tag 102v for responder 102 depicted and described in connection with Figure 1a. A DPP 191 also contemplates that associated information for responder 102* could be transferred in message 121, such as an identity for the WiFi enabled computing device operating responder 102*, MAC.R 102n, in addition to Br 102a and parameters 199a in message 121, as well as a responder DPP configuration 131 as depicted in Figure 1a.

Initiator 101 can receive Br 102a from message 121 and derive PKI keys Pi 101c and pi 101d. Initiator 101 can conduct an Elliptic Curve Diffie Hellman key exchange algorithm 401a as depicted and described below in key exchange 314 step for Figure 4a in order to derive a first shared secret key with responder 102*. Initiator 101 can derive a random number comprising an initiator nonce and also determine a set of capabilities for the initiator, such as a configurator or enrollee as specified in DPP specification version 1.0. Initiator 101 can encrypt the initiator nonce and the initiator capabilities using the derived shared secret key as depicted and described in connection with an encryption step 315 in Figure 4a below. Initiator 101 can send the derived public key Pi 101c and a ciphertext with the encrypted data in a message 122. Although not depicted in Figure 1b, initiator 101 can also send a hashed value of responder bootstrap public key Br 102a in message 122.

As specified in DPP specification version 1.0, message 122 could also include a hashed value for initiator bootstrap public key Bi 101a. But, for a responder authentication 141 depicted in Figure 1b, the hashed value of Bi 101a may optionally not be calculated or sent in message 122 (since responder 102* may not record the corresponding initiator bootstrap public key Bi 101a for a responder authentication 141 as depicted in Figure 1b). The hash algorithm on a bootstrap public key could comprise the SHA-256 hash algorithm according to IETF RFC 5754 titled "Using SHA2 Algorithms with Cryptographic Message Syntax". Other hash algorithms could be used as

Petitioner Samsung Ex-1004, 1408

well and specified in a selected set of cryptographic parameters 199 as depicted and described in connection with Figure 2b below, such as SHA-3, SHA-384, etc. Message 122 can comprise a Device Provisioning Protocol Authentication Request according to DPP specification version 1.0. Additional details for a message 122 are provided in section 6.2.2 of DPP specification version 1.0.

Responder 102* can receive message 122 and follow the steps for a responder according to the section 6.2.3 of DPP specification version 1.0, in order to generate a Device Provisioning Protocol Authentication Response, which could comprise message 123. Responder 102* can conduct an Elliptic Curve Diffie Hellman key exchange algorithm 401a as depicted and described below in key exchange 319 step for Figure 4a in order to mutually derive a first shared secret key with initiator 101. Responder 102* can decrypt the initiator nonce and the initiator capabilities using the derived first shared secret key as depicted and described in connection with an decryption step 320 in Figure 4a below. Responder 102* can use the key pair generation algorithm 102y in order to derive an ephemeral responder public key Pr 102p and a corresponding ephemeral private key 102r. Responder 102* can use the derived responder ephemeral private key pr 102r and the received initiator ephemeral public key Pi 101c in order to conduct a key exchange algorithm 314b as depicted below in Figure 4b and derive a second secret shared key. The first mutually derived symmetric key can comprise k1 and the second mutually derived symmetric key can comprise k2 according to the DPP specification version 1.0. Responder 102* can derive a random number comprising a responder nonce and also determine a set of capabilities for the responder, such as a configurator as specified in DPP specification version 1.0.

Responder 102* could also generate a responder authentication value, which could comprise a hash value over at least the initiator nonce received in message 122 and the generated responder nonce. Responder 102* could use the first shared secret key and the second shared secret key in order to derive a third shared secret key. Responder 102* could encrypt the responder authentication value with the third shared secret key and also then include the encrypted responder authentication value with the responder nonce and responder capabilities in a ciphertext. The ciphertext could be encrypted with the second shared key, as depicted and described in connection with an encryption step 315b in Figure 4b below. Responder 102* could send the derived ephemeral responder public key Pr 102p and the ciphertext in a message 123.

Initiator 101 can receive message 123 and record or store the received ephemeral responder public key Pr 102p. Initiator 101 can follow the steps in section 6.2.4 of DPP specification version 1.0 in order to process message 123 and confirm the authentication response from responder 102* in message 123. Initiator 101 can mutually derive the second shared secret

Petitioner Samsung Ex-1004, 1409

key using the received ephemeral responder public key 102p and the derived initiator ephemeral private key 101d, using a key exchange algorithm 319a as depicted in Figure 4b. Initiator 101 can decrypt the ciphertext in message 123 using the second mutually derived secret shared key and a decryption step 320b in Figure 4b. Initiator 101 can derive the third secret shared key using at least the first mutually derived secret shared key and the second mutually derived secret shared key. Initiator 101 can use the third mutually derived secret shared key to decrypt the encrypted responder authentication value using a decryption step 328 in Figure 4c. Initiator 101 can calculate a value for the responder authentication, and if the calculated value for the responder authentication matches the received value for the responder authentication, then responder 102* can be considered authenticated.

Although not depicted in Figure 1b, after authenticating responder 102*, initiator 101 can send an authentication confirm message, which could comprise a DPP authentication confirm message according to section 6.2.4 of the DPP specification version 1.0. The authentication confirm message can also include an initiator authentication value, which could be encrypted with the third secret shared key. The responder 102* can receive the encrypted authentication confirm value and decrypt the value and also internally calculate the initiator authentication value. If the received initiator authentication value and the calculated authentication confirm values are equal, then the initiator is authenticated with the responder 102*. Note that for a responder authentication 141 in Figure 1b, the authentication of initiator 101 by responder 102* is weak and only comprises verifying that initiator 101 records the bootstrap public key Br 102a for responder 102*. Also, although not depicted in Figure 1b, responder 102* could have the capabilities of a configurator and receive a DPP configuration request message from initiator 101, which could take the role of an enrollee. The DPP configuration request message can be according to section 6.3.2 of the DPP specification version 1.0, and can include an enrollee nonce, which could be a random number generated by the enrollee/initiator.

As depicted in Figure 1b, responder 102* can then send a configuration object in a ciphertext in message 124. Message 124 could comprise a DPP configuration response message according to section 6.3.3 of the DPP specification version 1.0. Responder 102* can record the configuration object before sending message 124. In the present disclosure, the configuration object comprises the network credentials 109 depicted in Figure 1a above. In other words, a configuration object can be configuration information for a device 106 with initiator 101 to use in order to connect with other devices via a wireless network. Configuration object could contain a list of network identifiers, device identifiers, RF band and channel information, configuration parameters, shared keys, PKI keys, names, passwords, group temporal keys, a shared secret 198

Petitioner Samsung Ex-1004, 1410

for a PKEX key exchange, a pairwise master key (PMK) and PMK identity, etc. An exemplary list of possible values for a configuration object is depicted in table 14 of DPP specification version 1.0. In message 124, the network credentials 109 as a configuration object can be sent in a ciphertext using the third mutually derived secret shared key. As depicted in Figure 1b, the ciphertext can include both the enrollee nonce and the network credentials 109. Initiator 101 can receive message 124, decrypt the ciphertext using the third mutually derived secret shared key, and record the network credentials 109. Device 101 can then apply the network credentials 109 and communicate with access point 105 using the received network credentials 109.

Figure 1c

Figure 1c is a graphical illustration of a device provisioning protocol for mutual authentication of an initiator and a responder, and configuration of an initiator, in accordance with conventional technology. Figure 1c depicts a summary of the WiFi Device Provisioning Protocol (DPP) specification, version 1.0 which was published on April 9, 2018, supporting a mutual authentication 142 by both initiator 101 and responder 102*. The summary depicted in Figure 1c highlights recorded bootstrap PKI keys, derived ephemeral PKI keys, and messages transmitted and received between an initiator 101 and a responder 102*. Many of (i) the PKI keys for initiator 101 and responder 102*, and (ii) the messages transmitted between the nodes are equivalent to those depicted and described in connection with Figure 1b. This description of Figure 1c will focus upon the differences in order for initiator 101 and responder 102* to mutually authenticate.

In order to support a mutual authentication between initiator 101 and responder 102*, initiator 101 can record an initiator bootstrap public key Bi 101a and an initiator bootstrap private key bi 101b. Although initiator bootstrap public key Bi 101a is depicted within initiator 101, another entity besides initiator 101 could record the public key and make it available to other nodes such as a responder 102*. As contemplated by the DPP Specification, the initiator bootstrap keys can be recorded with initiator 101 before conducting a DPP 191 authentication and configuration with responder 102*. For example, the bootstrap PKI keys for the initiator could be recorded during manufacturing or distribution of the initiator, or loaded by an initiator user before conducting the depicted communications with responder 102*. Bootstrap public keys for the initiator can support a selected set of cryptographic parameters 199a, where the selected set of cryptographic parameters 199a can match the parameters used by keys in responder 102*. The DPP specification version 1.0 identifies the minimal set of cryptographic parameters in order to ensure compatibility, which can comprise the "set A" for a set of cryptographic parameters 199 depicted in Figure 2a below.

Petitioner Samsung Ex-1004, 1411

In order to conduct a mutual authentication 142 in addition to the responder authentication 141 depicted in Figure 1c, a responder 102* should have access to the initiator bootstrap public key Bi 101a. In message 126, a responder 102* can receive the initiator bootstrap public key Bi 101a for initiator 101. The "out of band" transfer of Bi 101a in message 126 could comprise several different possible methods for responder 102* to receive Bi 101a, which are also described in the DPP specification 1.0, such as via reading a QR code, reading a near-field communications tag, using a Bluetooth wireless connection to read Bi 101a from initiator 101, and other possibilities exist as well, including using WiFi direct to first establish a "peer-to-peer" WiFi connection. A DPP 191 also contemplates that associated information for initiator 101 could be transferred in message 126, such as a device identity for initiator 101, MAC.I 106n or hash value 101aa, in addition to Bi 101a. As depicted in Figure 1c, the responder 102* can also record the initiator bootstrap public key Bi 101a, which can be used to authenticate the initiator 101 as described below.

For a mutual authentication 142 between device 101 and responder 102*, the two nodes can exchange the same or equivalent messages as message 122 and message 123, which were also described above in connection with Figure 1b. For mutual authentication, the third shared encryption key ke in DPP specification version 1.0 can be derived by a responder 102* using a key exchange step 321 and the received initiator bootstrap public key Bi 101a as depicted and described in connection with Figure 4c below. The use of initiator bootstrap public key Bi 101a for the key exchange step 321 is described in section 6.2.3 of the DPP Specification version 1.0. The difference between the third mutually derived secret shared key ke for a mutual authentication 142 and a responder authentication 141 is the inclusion of initiator bootstrap public key Bi 101a in the key exchange algorithm 401c and key derivation function 402c in Figure 4c. The first mutually derived secret shared key can comprise k1 and the second mutually derived secret shared key can comprise k2. The third mutually derived secret shared key ke is used by the responder 102* to encrypt the responder authentication value as depicted in an encryption step 322 in Figure 4c. The responder authentication value can include a random number "initiator nonce" or I-nonce from the initiator 101 in message 122. The responder authentication value can comprise the value "R-auth" in step 3 of section 6.2.3 of DPP specification version 1.0.

Initiator 101 can also mutually derive the third secret shared encryption key ke using the initiator bootstrap private key bi 101b and key exchange step 327 depicted and described below in connection with Figure 4c. Initiator 101 can subsequently decrypt the responder authentication value using the mutually derived third secret shared encryption key ke and a decryption step 328 as depicted and described in connection with Figure 4c below. Initiator 101 can calculate the same value for the responder authentication value as described in section 6.2.4 of DPP specification version 1.0. If the received, decrypted responder authentication value matches the calculated responder authentication value, then the responder 102* is authenticated.

Petitioner Samsung Ex-1004, 1412

The initiator 101 can calculate an initiator authentication value using at least the received responder nonce in message 123 and encrypt the initiator authentication value using the mutually derived third secret shared key ke (from a key exchange step 327) and send the initiator authentication value in a message (not shown in Figure 1c). The responder 102* can (i) decrypt the encrypted initiator authentication value using the third mutually derived secret shared key ke (from a key exchange step 327) and (ii) also calculate the initiator authentication value. If the calculated initiator authentication value by the responder 102* matches the received, decrypted initiator authentication value then the initiator 101 is authenticated with the responder 102* for a mutual authentication. The initiator 101 as enrollee can then receive a configuration object in a message 124. Message 124 can include network credentials 109 encrypted with the third mutually derived secret shared key ke. The initiator 101 can receive message 124 and pass the decrypted network credentials 109 to device 106, which can then apply the network credentials 109 in order to communicate with network access point 105.

Figure 1d

Figure 1d is an illustration of an exemplary initiator DPP configuration and initiator key database, with tables recording exemplary data for a plurality of bootstrap responder public keys and bootstrap initiator private keys, in accordance with exemplary embodiments. Although Figure 1b and Figure 1c above depicted and described conventional technology for an initiator 101, data for an initiator 101 depicted in Figure 1d and used by initiator 101 with a responder proxy 102 is not contemplated by conventional technology. An initiator 101 communicating with a responder proxy 102 can record and operate an initiator key database 101t, where an initiator key database 101t was also depicted and described above in connection with Figure 1a. An initiator key database 101t can include a responder public key table 101ta and an initiator key table 101tb.

Responder public key table 101ta can record a plurality of bootstrap responder public keys Br 102a for a plurality of different networks 107. The present disclosure with the use of a responder proxy 102' supports the recording and operation of the corresponding a bootstrap responder private key br 102b solely within a DPP server 103 and/or a device database 104 as depicted in Figure 1a. This represents a significant difference from the conventional technology depicted in Figure 1b and Figure 1c, which contemplates that a responder 102* would operate and record bootstrap responder private key br 102b. By recording responder bootstrap private key br 102b in secure servers within network 107, the corresponding responder bootstrap public key Br 102a can be "reused" or distributed among a plurality of devices 106 during a pre-configuration step or during manufacturing or distribution, before an end user receives the device 106. The reuse of responder bootstrap public key Br 102a can comprise shared keys 102z as depicted in Figure 2a. In other words, the responder

Petitioner Samsung Ex-1004, 1413

bootstrap public key Br 102a recorded in a device 106 before the device 106 conducts a DPP 191 can comprise a pre-shared bootstrap responder public key.

In addition, a manufacturer or distributor of device 101 may not know which network 107 among a plurality of possible different networks 107 that an end user for device 106 may wish to connect device 106. Since the responder bootstrap public key Br 102a comprises a public key and can be shared, a network 107 could send the responder bootstrap public key Br 102a to the manufacturer or distributor of device 106, where the responder bootstrap public key Br 102a could be recorded in a plurality of devices 106 in nonvolatile memory. Network 107 could securely record the corresponding responder bootstrap private key br 102b in a server database 103a or a device database 104. A manufacturer or distributor of device 106 could collect the publicly shared responder bootstrap public keys Br 102a from a plurality of different networks 107 and record the resulting plurality of different responder bootstrap public keys Br 102a in a responder public key table 101ta. The responder public key table 101ta could also be recorded in a plurality of devices. In this manner, an (a) exemplary device 106 depicted in Figure 1a could record the appropriate responder bootstrap public key Br 102a for a network 107 depicted in Figure 1a before (b) the device 106 is received by an end user.

The use of a responder public key table 101ta simplifies conducting a DPP 191 between responder 102' and initiator 101, since a separate "out of band" transfer of responder bootstrap public key Br 102a in a step 121 of Figure 1b and Figure 1c could be omitted or is not required by device 106 recording key Br 102a for a responder proxy 102'. A responder proxy 102' can operate with potentially many different keys Br 102a recorded by initiator 101, since responder 102' can forward messages such as DPP 122 to DPP server 103 for processing by DPP server 103 (and DPP server 103 could record and operate on the corresponding private key br 102b).

As depicted in Figure 1d, responder public key table 101ta for an initiator 101 can include data or values for a responder bootstrap public key Br identity 101z, a responder network 107a name or identity, a responder bootstrap public key Br 102a, a secure hash value 102aa over the key Br 102a, and a set of parameters 199 for the public key Br 102a. A responder public key table 101ta could record a plurality of data for a plurality of network 107a names with corresponding key Br 102a. In exemplary embodiments, at least one responder bootstrap public key Br 102a is recorded in a table 101ta for initiator 101. In exemplary embodiments data for an initiator key database 101t can be recorded in protected nonvolatile memory within device 106 and accessible only to initiator 101 or root or privileged processes operating in a device 105.

In exemplary embodiments, an initiator 101 for a device 106 could record a plurality of responder bootstrap public keys Br 102a in a responder public key table 101ta. Reasons could include the use of different keys Br 102a for different purposes, such as a first key Br 102a for a responder authentication 141 and a second key Br 102a for a mutual authentication 142. Or, a third key Br 102a could be recorded or mapped in a tag 102v from responder 102' and received in a message 121.

Petitioner Samsung Ex-1004, 1414

Responder 101x can designate the use of select keys Br 102a with a higher level of trust, with the following exemplary results (A) and (B). For (A), device 106 could provide additional privileges for a configuration step 106a for an initiator 102' or initiator 102* that uses the second key Br 102a (e.g. a mutual authentication 142 with the second key Br 102a could require responder 102' to have key Bi 101a, and release of key Bi 101a could be under the control of device 106 or recorded only in a device database 104 in a network 107). As one example of additional privileges, a set of credentials 109 delivered to device 106 using the second key Br 102a with mutual authentication 142 using key Bi 101a could allow the installation of a new certificate authority root certificate for device 106. As one example of normal or other privileges, a set of credentials 109 delivered to device 106 using the first key Br 102a could deny the installation of a new certificate authority root certificate for device 106.

As depicted in Figure 1d, initiator key table 101tb in an initiator key database 101t can record values for an initiator bootstrap key identity 101w, a set of parameters 199 for the initiator bootstrap private key bi 101b, the initiator bootstrap public key Bi 101a, a secure hash value 101aa over the key Bi 101a, and the initiator bootstrap private key bi 101b. In exemplary embodiments, an initiator key database 101t can include at least one initiator bootstrap public key Bi 101a in order to support a mutual authentication 142 depicted in Figure 1c. One use of an initiator key table 101tb is also depicted and described in connection with Figure 2b below.

As depicted in Figure 1d, initiator key table 101tb can record data for a plurality of different initiator bootstrap PKI keys. The initiator bootstrap PKI keys could be different according to the set of parameters 199. In other words, a first key bi 101b could be for parameters 199 of "A" with a key length of 256 bits, and a second key bi 101b could be for parameters 199 of "B" with an exemplary length of 384 bits. Different elliptic curves could be specified by parameters 199 for different keys in an initiator key database 101t. In exemplary embodiments where initiator 101 can provide bootstrap public key Bi 101a "in band" to responder 102', such as, but not limited to, initiator 101 using a PKEX key exchange, then table 101tb could record a plurality of different key bi 101b and Bi 101a for the same parameters 199, and initiator 101 could select and deprecate different keys bi 101b and Bi 101a over time. One use of an initiator key table 101tb is also depicted and described on connection with Figure 2b below.

As depicted in Figure 1a, an initiator DPP configuration 130 for initiator 101 is depicted in Figure 1d. An initiator DPP configuration 130 can comprise a collection of data and values recorded in nonvolatile memory for WiFi radio 106r. Initiator DPP configuration 130 can be similar or compatible with a responder DPP configuration 131 as depicted and described in connection with Figure 1f below. Initiator DPP configuration 130 can record values for a device supported configuration 130z, a global operating class 130a and a channel list 130b. Global operating class 130a can specify a WiFi technology and frequency band for operating WiFi radio 106r, and can represent the capabilities of the radio in a manufactured device 106. Global operating class 130a can support

Petitioner Samsung Ex-1004, 1415

exemplary class values and numbers in Appendix E of the 802.11ac specification, and other possibilities exist as well without departing from the scope of the present disclosure. Channel list 130b can comprise a list of channels for initiator 101 to transmit DPP authentication request messages 122. An initiator 101 in an unconfigured device 106' can rotate through a list of Global operating classes and channel numbers in an channel list 130b and transmit a series of DPP authentication request messages 122 on each combination of global operating class 130a and channel number from a channel list 130b in order to send message 122 to a responder 102' operating with a responder configuration 131.

Figure 1e

Figure 1e is an illustration of an exemplary responder bootstrap public keys and initiator bootstrap PKI keys recorded by a plurality of devices, in accordance with exemplary embodiments. Figure 1e depicts PKI keys recorded for an exemplary three different devices 106, although a system 100 and other systems herein could operate with potentially millions or more devices 106. The data depicted for each device in Figure 1e can comprise exemplary data for a responder public key table 101ta and an initiator key table 101tb within an initiator key database 101t, which were depicted and described in connection with Figure 1d above.

The first two entries for responder bootstrap public keys Br 102a for device 106 (1) and device 106(2) in Figure 1e depicts the same alphanumeric values for base85 binary to text encoding for an exemplary responder bootstrap public keys Br 102a in a first device 106(1) and a second device 106(1). Likewise, the second two entries for responder bootstrap public key Br 102a for device 106 (1) and device 106(2) in Figure 1e depicts the same alphanumeric values for base85 binary to text encoding for an exemplary second responder bootstrap public keys Br 102a in first device 106(1) and a second device 106(1). The depiction of these keys Br 102a illustrates the use of shared keys 102z for a plurality of different devices 106. Although only two devices are depicted with shared keys 102z, many more devices could also record the same shared keys for Br 102a. Each of the shared keys 102z is associated with a different network 107, identified with an exemplary network name 107a. In this manner, a plurality of different devices 106 can record and use the same value for a responder bootstrap public key Br 102a.

The same values for shared keys 102z across different devices 106 could be recorded in device 106 during manufacturing or before distribution to end users of device 106. In this manner, devices 106 could be received by end users in a "partially configured" yet secure state, such that a device 106 could use the recorded keys Br 102a with a responder proxy 102', where a responder proxy 102' does not operate or record the corresponding responder bootstrap private key br 102b. As depicted and described in connection with Figures 3a, 3b, 5a, and 5b below, a DPP server 103 could store and

Petitioner Samsung Ex-1004, 1416

operate with the corresponding responder bootstrap private key br 102b and thus the key br 102b can remain secured and not distributed out or sent to a responder proxy 102'. In this manner, a DPP 191 can be conducted between initiator 101 and a responder proxy 102' securely, where responder proxy 102' could operate on a device with a lower level of security or trust than DPP server 103. In some exemplary embodiments, responder proxy 102' could operate within an insecure device. Although not depicted in Figure 1e or Figure 1d, an entry for a key Br 102a could also record a "trust level" or "privilege level", such that a configuration object received from a DPP 191 using the key Br 102a with the corresponding privilege level could be authorized by device 106 to perform tasks or operations or record data with a higher level of privilege or trust.

In addition, the use of the same values for shared keys 102z across different devices 106 supports simplified yet secure automatic configuration of devices using a DPP 191, where WiFi access points 105 could operate as responder proxies 102'. As depicted and described in connection with Figure 1f below, a WiFi router operating as a responder proxy 102' could potentially connect with at least one network 107 and DPP server 103 (or perhaps a plurality of DPP servers 103 for different networks 107). Responder proxy 102' could use the hash value 102aa for the responder bootstrap public key Br 102a to determine a network 107 identified by the hash value 102aa (see responder network tables 102t below in Figure 1f). In this manner, responder proxy 102' could route data from a DPP authentication request message 122 to the correct DPP server 103 which operates and records (or can obtain from a device database 104 for network 107) the corresponding responder bootstrap private key br 102b in order to conduct the full series of messages for a device provisioning protocol 191.

In exemplary embodiments, responder bootstrap public keys Br 102a can also comprise a unique key such as a unique key 102vv recorded in a tag 102v for responder 102. Thus exemplary embodiment also support the use of a responder bootstrap public key Br 102a that is not shared across multiple initiators 101. In exemplary embodiments, and as depicted in Figure 1e, the initiator bootstrap private key bi 101b and corresponding initiator bootstrap public key Bi 101a can be different for different devices 106. In this manner, a device 106 can also be identified by the hash value 101aa (depicted in Figure 1d) over the initiator bootstrap public key Bi 101a by responder 102' or network 107. As depicted in Figure 1e, an initiator 101 can record a different initiator bootstrap private key bi 101b and corresponding initiator bootstrap public key Bi 101a for each set of supported parameters 199a. In exemplary embodiments, the initiator bootstrap private key bi 101b and corresponding initiator bootstrap public key Bi 101a can be recorded in a secure nonvolatile memory within device 106 for initiator 101 upon device manufacturing or during device distribution.

Petitioner Samsung Ex-1004, 1417

Figure 1f

Figure 1f is a graphical illustration of hardware, firmware, and software components for a responder, including a responder configuration step, in accordance with exemplary embodiments. Figure 1f is illustrated to include several components that can be common within a responder 102. Responder 102 may consist of multiple electrical components in order to both (i) communicate with a network 107 and (ii) transmit and receive messages with an initiator 101 according to a device provisioning protocol 191. Responder 102 can undergo a responder configuration step 127 in order to configure the components in a responder 102 to function as the enhanced responder or responder proxy as contemplated in the present disclosure, such that a responder bootstrap private key br 102b can remain recorded with a network 107 as depicted in Figure 1a above. In addition, although a responder 102 in Figure 1a is depicted as a separate component in a network access point 105, a network access point 105 could function and operate with the components of a responder 102 or responder 102' in Figure 1f. In other words, a network access point 105 or another wireless computing device with a WiFi radio 102x could comprise a responder 102 or responder 102' as contemplated herein. Responder 102 and configured responder 102' could also operate with a radio 102x that supports wireless networking technology other than WiFi, such as radio 102x comprising a radio for 4G LTE or 5G networks, and other possibilities exist as well.

In exemplary embodiments and as depicted in Figure 1f, responder 102 can include a responder identity 102i, a processor 102c (depicted as "CPU 102c"), random access memory (RAM) 102d, an operating system (OS) 102e, storage memory 102f (depicted as "nonvolatile memory 102f"), a WiFi radio 102x, a system bus 102j, and a wireless wide area network (WAN) interface 102h. Figure 1d depicts responder 102 as both a default responder 102 and a configured responder 102'. The default responder 102 can comprise the state of responder 102 before a configuration step 127 and responder 102' can comprise the state of responder 102' after the configuration step 127. As depicted, both a default responder 102 and a configured responder 102' can contain the same hardware components such as CPU 102c, RAM 102d, etc., where a difference between responder 102 and 102' can be in both (i) data recorded in storage memory 102f and a change in the operation of WiFi radio 102x to listen for incoming DPP messages 122.

Responder identity 102i could comprise a preferably unique alpha-numeric or hexadecimal identifier for responder 102, such as an Ethernet or WiFi MAC address, an International Mobile Equipment Identity (IMEI), an owner interface identifier in an IPv6 network, a serial number, or other sequence of digits to uniquely identify each of the many different possible units for responder 102 for a network 107. A system 100 depicted in Figure 1a could operate with a plurality of different responders 102. Responder identity 102i can preferably be recorded in a non-volatile memory or written directly to hardware in responder 102 by (i) a responder manufacturer upon device

Petitioner Samsung Ex-1004, 1418

manufacturing, or (ii) a pre-configuration step for responder 102 before starting the sequence of steps and message flows depicted in Figure 3a below.

The CPU 102c can comprise a general purpose processor appropriate for typically low power consumption requirements for a responder 102, and may also function as a microcontroller. CPU 102c can comprise a processor for responder 102 such as an ARM® based processor or an Intel® based processor such as belonging to the Atom or MIPS family of processors, and other possibilities exist as well. CPU 102c can utilize bus 102j to fetch instructions from RAM 102d and operate on the instruction. CPU 102c can include components such as registers, accumulators, and logic elements to add, subtract, multiply, and divide numerical values and record the results in RAM 102d or storage memory 102f, and also write the values to an external interface such as WiFi radio 102x or WAN interface 102h.

RAM 102d may comprise a random access memory for responder 102. RAM 102d can be a volatile memory providing rapid read/write memory access to CPU 102c. RAM 102d could be located on a separate integrated circuit in responder 102 or located within CPU 102c. The RAM 102d can include data recorded in responder 102 for the operation when collecting and communicating DPP messages 122 through 124 with initiator 101 depicted in Figures 1b, 1c, 3a, etc. The system bus 102j may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures including a data bus. System bus 102j connects components within responder 102 as illustrated in Figure 1f, such as transferring electrical signals between the components illustrated. Responder 102 can include multiple different versions of bus 102j to connect different components, including a first memory bus 102j between CPU 102c and RAM 102d, and a second system bus 102j between CPU 102c and WiFi radio 102x, which could be an I2C bus, an SPI bus, PCIe bus, or similar data busses.

The operating system (OS) 102e can include Internet protocol stacks such as a User Datagram Protocol (UDP) stack, Transmission Control Protocol (TCP) stack, a domain name system (DNS) stack, a TLS stack, a datagram transport layer security (DTLS) stack, etc. The operating system 101e may include timers and schedulers for managing the access of software/firmware to hardware resources within responder 102, including CPU 102c and WiFi radio 102x. The operating system shown of 102e can be appropriate for a low-power device with more limited memory and CPU resources (compared to a server such as DPP server 103, device database 104, etc.). Example operating systems 102e for a responder 102 includes Linux, Android® from Google®, IOS from Apple®, Windows® 10 IoT Core, or Open AT® from Sierra Wireless®. Additional example operating systems 102e for a responder 102 include eCos, uC/OS, LiteOs, Contiki, OpenWRT, Raspbian, and other possibilities exist as well without departing from the scope of the present disclosure. Although depicted as a separate element within responder 102 in Figure 1f, OS 102e may reside in RAM 102d and/or storage memory 102f during operation of responder 102.

Petitioner Samsung Ex-1004, 1419

Storage memory 102f (or "nonvolatile memory 102f") within responder 102 can comprise a nonvolatile memory for storage of data when responder 102 is powered off. Memory 102f may be a NAND flash memory or a NOR flash memory, and other possibilities exist as well without departing from the scope of the present disclosure. Memory 102f can record firmware and/or software for responder 102. Memory 102f can record long-term and non-volatile storage of data or files for responder 102. In an exemplary embodiment, OS 102e is recorded in memory 102f when responder 102 is powered off, and portions of memory 102f are moved by CPU 102c into RAM 102d when responder 102 powers on. Memory 102f (i) can be integrated with CPU 102b into a single integrated circuit (potentially as a "system on a chip"), or (ii) operate as a separate integrated circuit or a removable card or device, such as a removable SD card. Memory 102f may also be referred to as "device storage" and can include exemplary file systems of FAT16, FAT 32, NTFS, ext3, ext4, UDF, or similar file systems.

As depicted in Figure 1f, nonvolatile memory 102f for a default responder 102 may also contain a secret key SK0.responder 102s, a certificate cert.responder 102m, and a certificate cert.CA.root 102w. A secret key SK0.responder 102s in a nonvolatile memory 102f could comprise the private key for a PKI key pair, where the corresponding public key could be recorded in a certificate cert.responder 102m. A secret key SK0.responder 102s and the corresponding public key in a certificate 102m could comprise values to use with asymmetric public key infrastructure (PKI) cryptography, and could utilize exemplary algorithms based on either Rivest Shamir Adleman (RSA) or elliptic curve cryptography (ECC). For use of ECC algorithms, parameters within certificate 102m (and other certificates herein) can specify elliptic curve names (e.g. NIST P-256, sect283k1, sect283r1, sect409k1, sect409r1, etc.). The same ECC parameters can be used with bootstrap PKI keys in the form of parameters 199a. Further, elliptic curves that do not depend on parameters specified by NIST could be utilized as well, such as Curve22519 or FourQ. For use of RSA algorithms, parameters within certificate 102m can specify a modulus and other associated values for using an RSA PKI key pair. For either asymmetric algorithms RSA or ECC for a PKI key pair herein, parameters in a certificate such as cert.responder 102m can specify key lengths, a duration for key validity, uses of the PKI key pair such as for key derivation or signatures, encoding rules, message digest algorithms, etc. Parameters in certificate 102m (and other certificates herein) may also include identifying information associated with the PKI key pair such as a sequence number, a serial number, a certificate revocation list or URL, and a domain name of the server associated with the PKI key pair.

Certificate cert.CA.root 102w can comprise a root certificate for responder 102 to authenticate a server certificate received from DPP server 103 or other servers or nodes in a network 107, where these other nodes for network 107 are depicted and described in connection with Figure 1a, Figure 3a, and other Figures below. After authenticating a server certificate received by responder 102 or 102' with cert.CA.root 102w, responder 102 can conduct additional authentication steps contemplated by

Petitioner Samsung Ex-1004, 1420

secure protocol standards such as Transport Layer Security (TLS), Datagram Transport Layer Security (DTLS), IPSec, etc. Root certificate cert.CA root 102w can comprise the list or a subset of the list of included root certificates from the Mozilla Foundation with Mozilla projects, where the aggregate list of community approved root certificates and associated parameters is in the widely distributed file certdata.txt from the Mozilla web site.

Nonvolatile memory 102f in a configured responder proxy 102' can include a responder network table 102t. Responder network table 102t could be received from a network 107 for a configuration step 127 in a message 305 depicted below in Figure 3a. Or, table 102t could be loaded by a user of responder 102' or potentially recorded during a pre-configuration of responder 102 or responder 102' before the sequence of message flows and steps in Figure 3a below. Responder network table 102t can record and map a DPP server 103 that is associated with a responder bootstrap public key hash value 102aa. The responder bootstrap public key hash value 102aa can be associated with a network 102ab for responder bootstrap public key Br 102a (e.g. the network equivalent to network 107 that records the responder bootstrap private key br 102b to process an incoming DPP authentication request message 122). The configured responder proxy 102' can use the responder network table 102t to route data from incoming DPP messages 122 with the responder bootstrap public key hash value 102aa to a selected DPP server 103. Different DPP servers 103 could be associated with different networks 107, where a network 107 can comprise a network for H(Br) 102aa, as depicted for a responder network table 102t in Figure 1f. Although not depicted in Figure 1f, a responder network table 102t could also include a set of cryptographic parameters 199a associated with the responder bootstrap public key hash value 102aa, including the hash algorithm 199e associated with the responder bootstrap public key hash value 102aa.

In some exemplary embodiments, the responder network table 102t with a plurality of responder bootstrap public key hash values 102aa could be omitted from a memory of responder 102, and a DPP server 103 or network 107 could record the values. A network 107 or DPP server 103 could use a hash value 102aa received in a message 309 (depicted and described below in connection with Figure 3a) in order to select a DPP server 103. In other exemplary embodiments, a responder nework table 102t could include at least one value for a responder bootstrap public key hash value 102aa.

In other words, a plurality of different devices 106 with initiators 101 may use different responder bootstrap public keys Br 102a, where the hash value 102aa would be received in an incoming DPP message 122. The devices 106 could be manufactured or pre-configured with at least one key Br 102a, and the manufacturer could operate or be associated with a DPP server 103 in order to support and conduct a DPP 191 with device 106 and send a configuration object in a DPP configuration response message 124 from Figure 1c. Note that a configuration object in a message 124 could include data other than credentials 109 (and may optionally omit credentials 109), where

Petitioner Samsung Ex-1004, 1421

the configuration object could include configuration parameters for device 106. Responder 102' can use the received hash value 102aa to select a DPP server 103 in order to further process the message and support a DPP 191, where the network 102ab could be different than a network 107 operating a network access point 105. In exemplary embodiments a network 102ab operating a DPP server with responder bootstrap private key br 102b can be selected using the hash value 102aa over key responder bootstrap public key Br 102a.

A device 106 with initiator 101 could use (i) a first DPP 191 with a first Br 102a to obtain network credentials 109 with network 107 for AP 105, and (ii) a second DPP 191 with a second Br 102a (with the same responder proxy 102') to obtain configuration parameters with network 102ab for device 106. The first DPP 191 could be with a first DPP server 103 and first network 102ab, and the second DPP 191 with a second DPP server 103 and a second network 102ab. The use of responder network table 102t by responder 102' can determine which incoming DPP message 122 is associated with which network 102ab and DPP server 103. In other words, a responder 102' can conduct multiple sessions of DPP 191 with different keys Br 102a, and use a responder network table 102t to determine which DPP messages 122 are associated with which DPP server 103 (via hash value 102aa). Other possibilities exist as well for a responder 102' to use a responder network table 102t in order to process different incoming DPP authentication request messages 122 without departing from the scope of the present disclosure. In exemplary embodiments, a selected DPP server 103 from a responder network table 102t can provide credentials 109 for a DPP configuration response 124, using the steps below such as those depicted in Figure 3a, 3b, Figure 6, etc.

Responder 102 and 102' can include a WiFi radio 102x to communicate wirelessly with networks such as device 101 depicted and described in Figure 1a above, including via WiFi data link 192. WiFi radio 102x could connect with an antenna in order to transmit and receive radio frequency signals. Responder 102 and 102' can operate a WAN interface 102h in order to communicate with network 107, and WAN interface can include a WAN MAC address 102q. WAN interface 102h can comprise either a wireless connection, such as with a 4G LTE network, a 5G network, or can comprise a wired connection such as digital subscriber line (DSL), coaxial cable connection, or Ethernet connections, and other possibilities exist as well without departing from the scope of the present disclosure.

As depicted in Figure 1f, a WiFi radio 102x in a configured responder 102' can record or operate with a responder configuration 131. The values for responder DPP configuration 131 could also specify a version of the 802.11 standards to utilize, such as, but not limited to, 802.11n, 802.11ac, 802.11ah, 802.11ax, or related and subsequent versions of these standards. The responder DPP configuration 131 can be compatible with the initiator DPP configuration 130 of a target WiFi device 101 with responder 102'. The responder configuration 131 could be obtained during an authentication step for responder 102, such as depicted below in Figure 3a (in a message 305), although responder

Petitioner Samsung Ex-1004, 1422

configuration 131 could be obtained at other times as well, such as possibly being recorded in a WiFi radio 102x before responder 102 begins sending and receiving data through a WiFi link 192.

The responder DPP configuration 131 can include a responder operating class 131a which can be selected from the plurality of potential global WiFi operating classes in table E-4 of IEEE standard specification 802.11ac, which is incorporated by reference herein. The present disclosure contemplates that a responder 102' communicates with multiple initiators 101 either concurrently, or over time. The responder configuration 131 can also include Responder DPP Channel Authentication List 131b, which can comprise a list of channels for responder 102' to receive DPP authentication request 122 messages in order to establish initial connectivity with device 106 and initiator 101. In exemplary embodiments, responder configuration 131 for responder 102' provides data for a responder 102 to receive data such as message 122 from initiator 101, where initiator 101 may operate with different possible values for operating classes or transmitting channels.

A responder configuration 131 can include other data for responder 102' to receive a DPP message from an initiator 101 as well, such as specifying the use of broadcast or unicast for receiving DPP messages, timer values for retry of sending DPP 191 messages, timer values for timeouts for sending and receiving messages, as well as counters for a maximum number of retries on sending messages. In other words, without responder configuration 131, a responder 102 may not normally be able to (i) select the proper WiFi operating channel, and (ii) receive broadcast data from initiator 101. For embodiments where a Device DPP Channel Authentication List 131b is unknown or not available, then responder 102' can periodically listen for incoming DPP 122 messages over time as broadcast messages on all available channels for a radio 102x.

WiFi radio 102x can include standard radio components such as RF filters, RF amplifiers, a clock, and phased loop logic (PLL). WiFi radio 102x can support protocols and specifications according to the IEEE 802.11 family of standards. For example, if WiFi radio 102x operates according to 802.11n standards, WiFi radio 102i could operate at a radio frequency of 2.4 or 5 GHz. In other embodiments where WiFi radio 102i supports 802.11ac standards, then WiFi radio 102x could also support radio frequencies of 0.054 through 0.79 GHz in order to operate in "white space" spectrum. Other possibilities exist as well for WiFi radio to support wireless LAN standards or 802.11 standards and radio frequencies, without departing from the scope of the present disclosure. WiFi radio 102x can access a nonvolatile memory 102f in order to record the depicted configuration values of responder configuration 131 where the nonvolatile memory could be within WiFi radio 102x, or WiFi radio 102x could access memory 102f via bus 102j in order to record and read the values. Device 106 can operate a WiFi radio 106r that is compatible with WiFi radio 102x.

As depicted in Figure 1f, a WiFi configured responder 102' can include configuration data resulting from a configuration step 127, in order for responder 102' to conduct a device provisioning protocol 191 with a device 101. A nonvolatile memory 102f for a WiFi configured responder 102'

Petitioner Samsung Ex-1004, 1423

can include data recorded for a default responder 102, with the addition of recording a DPP application program 102g. DPP application program 102g (or "DPP application 102g") could comprise a software package for OS 102e in order for responder 102' to conduct the series of steps and messages with DPP server 103 and initiator 101 depicted below in Figure 3a, Figure 3b, Figure 4a, etc. DPP application 102g can include cryptographic algorithms in order to support key exchanges, symmetric encryption and decryption, as well as signature verification and asymmetric encryption and decryption. As one example, a DPP application 102g could perform the function of a key derivation 102y from Figure 1b, and other possibilities exist as well.

Responder 102 may also optionally include user interface 102k which may include one or more devices for receiving inputs and/or one or more devices for conveying outputs. User interfaces are known in the art and may be simple for many devices such as a few LED lights or and LCD display, and thus user interfaces are not described in detail here. User interface 102k could comprise a touch screen if responder 102 has sophisticated interaction with user 102a. Responder 102 can optionally omit a user interface 102k, if no user input or display is required for conducting a device provisioning protocol 191 between responder 102 and device 106 with initiator 101. Although not depicted in Figure 1f, responder 102 can include other components to support operation, such as a clock, power source or connection, antennas, etc. User interface 102k could also comprise a web server operating in responder 102 and responder 102', such that a user could access the user interface 102k with a web browser and view or control depicted data stored in memory for a responder 102 or configured responder 102'. Other possibilities exist as well for hardware and electrical components operating in responder 102 or responder 102' without departing from the scope of the present disclosure.

Although not depicted in Figure 1f, the various servers shown above in Figure 1a such as DPP server 103 and device database 104 can include equivalent internal electrical components depicted for a responder 102 in order to operate as servers. The servers in Figure 1a could include a processor similar to CPU 102c, with primary differences for the processor in a server being increased speed, increased memory cache, an increased number and size of registers, with the use of a 64 bits for datapath widths, integer sizes, and memory address widths, etc., In contrast, an exemplary 32 bit datapaths could be used for CPU 102c in responder 102 (although CPU 102c could also use 64 bit wide datapath widths if responder 102 comprises a mobile phone such as a smart phone). For embodiments where responder 102 comprises a home WiFi router, then a CPU 102c in responder 102 could comprise an exemplary 32 bit processor, although other possibilities exist as well.

Similarly, RAM in a server could be a RAM similar to RAM 102d in responder 102, except the RAM in a server could have more memory cells such as supporting exemplary values greater than an exemplary 8 gigabytes, while RAM 102d in responder 102 could support fewer memory cells such as less than an exemplary 4 gigabytes. Non-volatile memory for storage in a server in Figure 1a could comprise disks, "solid state drives" (SSDs) or "storage area networks" (SAN) for a server. For a WAN

Petitioner Samsung Ex-1004, 1424

interface in a server equivalent to WAN interface 102h for responder 102', in exemplary embodiments, a server in Figure 1a could use a physical, wired LAN interface such as a high-speed Ethernet or fiber optic connection. In exemplary embodiments, a device 106 with an initiator 101 can include and operate with hardware and electrical components similar or equivalent to the components for a responder 102' depicted in Figure 1f.

Figure 2a

Figure 2a is an illustration of an exemplary device database, with tables for a device database recording exemplary data, in accordance with exemplary embodiments. A device database 104 depicted and described above in connection with Figure 1a can record data for (i) an unconfigured responder 102 or (ii) a configured responder 102' and a DPP server 103 to conduct a device provisioning protocol 191 with a device 106 operating an initiator 101. A database 104 could record a plurality of tables for a plurality of devices 106 and/or responders 102', such as the depicted values for a key table 104k, and a set of cryptographic parameters 199. Other possibilities exist as well for the organization, tables, and recorded data within a device database 104 without departing from the scope of the present disclosure. Data within device database 104 could be encrypted using a symmetric key. As depicted in Figure 1a, a network 107 could operate and record a device database 104, although another entity besides network 107 could also operate and record the device database 104, such as a device manufacturer or a device owner. For embodiments where a plurality of different networks 107 may communicate with a device 106 through responder 102' over time, each of the different networks 107 can operate a device database 104.

Device database 104 can record sets of ID.device 106i, MAC.I 106n for the WiFi MAC address of the device 106 operating the initiator 101, responder identity 102i, a responder mode 306a, cryptographic parameters 199, and a secret shared key 198 for a public key exchange protocol (PKEX). These values within a device database could be recorded in a device 106 and/or initiator 101 upon or after manufacturing of a device 106 or distribution of device 106 and before device 106 conducts a DPP 191 with responder 102'. The exemplary data in device database 104 could be obtained from the manufacturer or device distributor. For embodiments where some data is not previously recorded in a device database 104 before an initiator 101 sends a message 122 as depicted below in Figure 3a, then a network 107 operating device database 104 could query the device 106 manufacturer or a device 106 distributor in order to receive and record values for a device database 104, and other possibilities exist as well for the source of device 106 data within a device database 104. Note that key 198 for PKEX can be optionally omitted for some devices, with the depiction of "NA".

ID.device 106i can comprise an identity for device 106i, and could be a serial number or identification number or string for device 106. The two identity values shown for a device 101 in a

Petitioner Samsung Ex-1004, 1425

device database 104 allows a DPP server 103 or network 107 to map or convert between ID.device 106i and MAC.I 106n, and also convert one value into another. MAC.I 106n can comprise the MAC address used by an initiator 101 for conducting a device provisioning protocol 191. In other words, MAC.I 106n can be used by a responder 102' or network 107 in order to identify that an incoming DPP authentication request message 122 is from device 106 (and not a fraudulent or imposter device 106). For embodiments where key Br 102a comprises a shared key 102z, MAC.I 106n can also be used by network 107, database 104, DPP server 103, or responder 102' to verify the message 122 is from device 106 (since only device 106 with MAC.I 106n would normally send the message 122). A message 122 would have a source MAC address, which could be checked against MAC.I 106n that was recorded in a device database 104. Further, the hash value 101aa from a message 122 could be checked against a hash value calculated by device database 104 or DPP server 103 over the recoded initiator bootstrap public key Bi 101a. Additional information for verifying an identity for device 106 is provided for a step 312 below in Figure 3a.

In some exemplary embodiments, MAC.I 106n can be omitted from a device database 104 or otherwise not available to a network 107, and in this case the MAC.I 106n can be recorded in a database 104 after successful receipt of a DPP authentication confirm message 123a depicted in Figure 3b below. In other words, some devices 101 with initiators 106 may change or randomize the MAC.I 106n over time, and the MAC.I 106n stored in a device database 104 can comprise the value of MAC.I 106n used by device 106 to send a DPP authentication request message 122. A network 107 with a device database 104 could operate a plurality of different responders 102'. The responder identity 102i of a responder 102' communicating with a device 106 could be recorded in a device database 104, although in some exemplary embodiments the responder identity 102i could be optionally omitted. Responder identity 102i could be omitted or blank in a device database 104 until after a message 122 is received by a responder 102' and forwarded to device database 104 via network 107.

A device database 104 could also specify a responder mode 306a (in Figure 3a and Figure 3c below) for responder 102' to operate with initiator 101, where different responder modes could comprise a first responder mode as depicted in Figure 3a, where DPP server 103 records and operates on (a) responder bootstrap private key br 102b and (b) ephemeral PKI keys for responder 102' and (c) all derived secret shared keys (as depicted in the first row of Figure 3c). A second responder mode 306a could comprise DPP server 103 recording and operating on responder bootstrap private key br 102b, but not derived responder ephemerals private key pr 102r for responder 102' (e.g. as depicted in Figure 5a). A third responder mode 306a could comprise DPP server 103 recording and operating on responder bootstrap private key br 102b and derived responder ephemeral private key pr 102r, but not derived secret shared keys for encryption such as keys k1, k2, and ke (e.g. as depicted in Figure 5b).

Petitioner Samsung Ex-1004, 1426

In some exemplary embodiments, a device 106 could comprise a higher value device 106 or require a greater level of security, so the security level of a DPP 191 could be selected using a responder mode 306a. In addition, the responder mode 306a could be selected based on the security level of a responder 102', such that a first responder mode 306a such as in Figure 3a and Figure 3b is selected by network 107 for a responder 102' with a lower level of security. A second responder mode 306a such as depicted in Figure 5b is selected by a network for a responder 102' with a higher level of security, such as a gnodeb in a 5G network or a secured enterprise WiFi router. Other reasons for the selection of a responder mode exist as well without departing from the scope of the present disclosure.

Sets of cryptographic parameters 199 in device database 104 can represent different sets of cryptographic parameters supported by responder 102' and/or initiator 106 in order to conduct a device provisioning protocol 191. For example, some devices 106 may use a first set of cryptographic parameters with PKI keys such as bootstrap and ephemeral keys, and other devices may use a second set of cryptographic parameters, which could be specified by cryptographic parameters 199. Note that devices 101 can optionally support more than one set of cryptographic parameters, as depicted for device with exemplary identity ID.device 106i of "20003", where the device can operate with either a set "A" of cryptographic parameters 199 or a set "B" of cryptographic parameters.

PKEX key 198 could be used with a public key exchange protocol as specified in IETF draft document "Public Key Exchange", which is hereby incorporated by reference in its entirety. In some exemplary embodiments, a PKEX key 198 can be omitted, such as for the example data depicted with an "NA" in Figure 2b. The values depicted of X1, X2, etc. for a PKEX key 198 could comprise the secret key for each device 106 identified by MAC.I 106n. For the exemplary database 104 depicted in Figure 2a and other tables herein, a numeric value following a constant name can depict a subsequent or different instance or numeric value for the constant. In other words, "X1" and "X2" for a PKEX key 198 can represent different values of "X" for the PKEX. PKEX key 198 could be used as depicted and described in Figure 3a with a PKEX session 312e in order to query device 106 for the initiator bootstrap public key Bi 101a if a database 104 does not record the key Bi 101a (or if network 107 cannot obtain key Bi 101a via other means via an IP network 113).

Device database 104 can record a key table 104k, which could comprise exemplary keys as depicted. A key table 104k can include responder bootstrap public key Br 102a, responder bootstrap private key br 102b, initiator bootstrap public key Bi 101a, a secure hash value for responder bootstrap public key 102aa, and a secure hash value for initiator bootstrap public key 101aa. Similarly, values of "Br-1" and "Br-2" depict different values for a responder bootstrap public key 102a, but exemplary values of "Br-1" and then a second "Br-1" can depict that the same value or number may be recorded as the responder bootstrap public key Br 102a in different devices 106 each with an initiator 101. The use of the same value for a responder bootstrap public key Br 102a across different devices 106 can

Petitioner Samsung Ex-1004, 1427

comprise a shared responder bootstrap public key 102z, which is also depicted and described in connection with Figure 1e above.

Responder bootstrap public key Br 102a in a device database 104 can represent the bootstrap public key for responder 102'. Br 102a was also depicted and described above in connection with Figure 1a, Figure 1b, and Figure 1e above. In exemplary embodiments, a device database 104 can record a plurality of different initiator bootstrap public keys Bi 101a, where each key for a set of parameters 199 can be associated with different initiators 101. As depicted in Figure 2b, for some embodiments different initiators 101 could also share the same numeric value for an initiator bootstrap public key Bi 101a (e.g. depicted devices "20001" and "20002" can both use an initiator bootstrap public key Bi 101a of "Bi-1"). Also note that a DPP 191 can be conducted between a responder 102' and initiator 101 without recording an initiator bootstrap public key Bi 101a in a device database 104 or DPP server 103 or responder 102', which could comprise a responder only authentication 141 as depicted in Figure 1b. For these embodiments, the depicted value for Bi 101a can be "NA" as shown in Figure 2a.

In a key table 104k in device 104, different initiator bootstrap public keys Bi 101a can be associated with a different selected set of parameters 199a. The manufacturer or entity configuring initiator 101 in device 106 may not know which of a plurality of different possible sets of cryptographic parameters 199 may be utilized by an initiator 101 for conducting a DPP 191 in order to be compatible with a responder 102'. For example, DPP specification version 1.0 identifies several possible sets of cryptographic parameters 199 for bootstrap keys that could be utilized in a DPP 191, such as the different values for cryptographic parameters shown in Appendix A of DPP specification version 1.0. As depicted in Figure 2a, an initiator bootstrap public key Bi 101a of "Bi-5 (A)" can represent an instance or number for Bi 101a which uses the selected set "A" of cryptographic parameters 199 (e.g. curve p256 with SHA-256), while initiator bootstrap public key Bi 101a of "Bi-5 (D)" can represent an instance of Bi 101a which uses the selected set "D" of cryptographic parameters 199 (e.g. curve 25519 with SHA-3) Similar designations and uses of different cryptographic parameters 199 with different responder PKI keys can be recorded in a device database 104 as well. In some exemplary embodiments, the values depicted for responder keys could be omitted from a device database 104 and recorded in another server such as DPP server 103.

Responder bootstrap public key Br 102a and corresponding responder bootstrap private key br 102b in a device database 104 can represent the bootstrap public key and the bootstrap private key for a responder proxy 102'. Br 102a and br 102b were depicted and described above in connection with Figures 1a through Figure 1e above. In exemplary embodiments, a device provisioning protocol 191 using a responder 102' and a DPP server 103 can utilize a plurality of different responder bootstrap PKI keys, and different values for the responder bootstrap PKI keys Br 102a and br 102b can be associated with different initiators 101. As discussed above in connection with Figure 1a and 1b, an

Petitioner Samsung Ex-1004, 1428

initiator 101 could record or obtain a responder bootstrap public key Br 102a, and subsequently the value Br 102 recorded for the initiator 101 could also be stored in a device database 104.

The responder bootstrap public key Br 102a and corresponding responder bootstrap private key br 102b could be stored by a device database 104 concurrently with the initiator bootstrap public key Bi 101a, where the initiator bootstrap public key Bi 101a could be (i) received from a device manufacturer or device distributor, or (ii) obtained by a PKEX session 312e (in Figure 3a), or (iii) responder 102' scanning a QR code or bar code for initiator 101 (and sending the key Bi 101a to device database 104), and other possibilities exist as well for the source of key Bi 101a in a database 104 for device 106 with MAC.I 106n. Or, in exemplary embodiments a device manufacturer or a device distributor could operate a device database 104 and subsequently record the exemplary keys and values depicted of (i) Bi 101a for initiator 101 in device 106 and (ii) recorded Br 102a in device 106, but not (iii) br 102b, which could be recorded by a device database 104 for a network 107 operating or associated with responder 102'. As depicted in Figure 2a, some responders 102' could share a common responder bootstrap public key Br 102a, which could be associated with the same responder bootstrap private key br 102b.

The sharing of responder bootstrap public key Br 102a among multiple devices 106 with initiators 101 (e.g. "shared key 102z") can simplify the manufacturing and distribution of devices by using a common responder bootstrap public key Br 102a for some embodiments. For the present disclosure, the potential tradeoff of greater simplicity for reduced security (by reusing a responder bootstrap public key Br 102a in multiple devices 106) can be avoided, since the responder bootstrap private key br 102b may (i) preferably not be sent to the responder 102', and (ii) remain recorded in and operated by a secured DPP server 103.

Yet, with the embodiments contemplated herein, the full set of device provisioning protocol messages 191 can be sent and received with an initiator 101 by responder proxy 102'. The full set of DPP 191 messages can be transmitted and received by a responder 102' with initiator 101 using either responder authentication 141 or mutual authentication 142. Embodiments for the present disclosure depict and describe the responder bootstrap private key br 102b as residing within a DPP server 103 when a responder 102' conducts a DPP 191 (as depicted and described in connection with Figure 3a, Figure 3b, Figure 5a, etc. below). Consequently in exemplary embodiments and a difference with conventional technology depicted in Figure 1b and Figure 1c, (i) the responder bootstrap private key br 102a does not need to be recorded by a responder 102' (but br 102a is recorded for a conventional responder 102* in Figure 1b and Figure 1c), and (ii) a responder bootstrap public key Br 102a can be commonly shared between multiple initiators 101 in different devices 106 (e.g. "shared key 102z").

Thus, the present disclosure resolves the potential tradeoff of security versus simplification by supporting manufacturing and distribution of devices 106 with initiators 101 with commonly shared responder bootstrap public key Br 102a (also depicted in Figure 1e as "shared key 102z"). Additional

Petitioner Samsung Ex-1004, 1429

details will be provided in Figures below. The responder bootstrap private key br 102b can be recorded in a device database 104 and subsequently sent to a DPP server 103, but preferably not sent to responder 102', and responder 102' can still conduct a device provisioning protocol 191 for initiator 101 using mutual authentication 142 as depicted in Figure 1c. In exemplary embodiments, a responder bootstrap public key Br 102a is recorded inside an initiator 102, while a device database 104 can record both the responder bootstrap public key Br 102a and the responder bootstrap private key br 102b, as depicted in Figure 2a.

Cryptographic parameters 199 can specify sets of cryptographic parameters that are supported by responder 102' when using bootstrap PKI keys. Cryptographic parameters 199 can be recorded in a device database 104. Cryptographic parameters 199 can record a collection of cryptographic algorithms or specifications such as a set identifier 199a, a key length 199b, an ECC curve name 199c, a hash algorithm 199e, a key length for a symmetric ciphering algorithm 199f, and a nonce length 199g. Asymmetric encryption parameters 199h can specify values and parameters to use with an asymmetric encryption algorithm 351a depicted and described in connection with Figure 3e below. Parameters 199i can specify encoding rules, padding, key lengths, selected algorithms, curve names, use of deterministic mode, and other values or fields necessary to utilize a signature algorithm 338b depicted and described in connection with Figure 3d below. In exemplary embodiments, parameters 199i can specify ASN.1 syntax notation (abstract syntax notation) with distinguished encoding rules (DER). Parameters 199i can also specify the use of either (i) ephemeral PKI keys or (ii) bootstrap PKI keys for use with a digital signature algorithm.

As contemplated herein, when a selected set of cryptographic parameters such as using the words or description "parameters 199a" or "cryptographic parameters 199a" can specify a row of parameters or values in a set of cryptographic parameters 199, such that the collection of values in the row can be used with a device provisioning protocol 191. Set identifier 199a can be an identity for a row or set of values for cryptographic parameters 199. For example, set "A" can comprise cryptographic suite 1 as specified in section 3.2.3 of DPP specification version 1.0. Key length 199b can be the length of keys in bits for PKI keys used with a device provisioning protocol with an initiator 101 in a device 106. ECC Curve name 199b can be a name for an ECC curve used with PKI keys for a device provisioning protocol with an initiator 101 in a device 106.

Hash algorithm 199e in cryptographic parameters 199 can be the name of a secure hash algorithm, such as the exemplary SHA-256 algorithm depicted, which may also be referred to as "SHA-2". Hash algorithm 199e can be used in a key derivation function (e.g. 402a, 402b, etc. below) and also to generate a signature based on a received nonce. A key length for a symmetric ciphering algorithm 199f can specify the length of a symmetric key in bits to use with a symmetric ciphering algorithm. The depicted algorithm in Figure 2a for algorithm 199f is "AES-SIV", although other symmetric ciphering algorithms could be specified in a set of cryptographic parameters as well. In

Petitioner Samsung Ex-1004, 1430

other words, an additional, different column in cryptographic parameters 199, such as an exemplary "symmetric algorithm" 199z (not depicted) could specify the symmetric ciphering algorithm to utilize with a responder for a device provisioning protocol. Nonce length 199g can specify the length in bits for nonces or random numbers generated by both initiator 102' and responder 101x, including the length in bits of authentication values, where the nonces can be used for generating signatures and encryption keys.

Figure 2b

Figure 2b is a flow chart illustrating exemplary steps for an initiator to use an initiator DPP configuration and initiator key database in order to automatically connect with a responder proxy, in accordance with exemplary embodiments. An initiator 101 can use the exemplary data recorded in an initiator DPP configuration 130 and initiator key database 101t in order to automatically connect with a responder proxy 102'. The steps depicted in Figure 2b allow a device 106 with initiator 101 to locate or find a compatible responder proxy 102' in an automated manner, such that no manual intervention is required by a device user to conduct a DPP 191 with subsequent configuration step 106a in order to load a device with credentials 109. This provides benefits over a conventional DPP 191, where additional manual intervention is required for operating a device with initiator 101 as a configurator and obtaining a bootstrap responder public key Br 102a via an "out of band" method, or loading a device 106 with a PKEX key 198.

The processes and operations, described below with respect to all of the logic flow diagrams and flow charts may include the manipulation of signals by a processor and the maintenance of these signals within data structures resident in one or more memory storage devices. For the purposes of this discussion, a process can be generally conceived to be a sequence of computer-executed steps leading to a desired result.

These steps usually require physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, compared, or otherwise manipulated. It is convention for those skilled in the art to refer to representations of these signals as bits, bytes, words, information, elements, symbols, characters, numbers, points, data, entries, objects, images, files, or the like. It should be kept in mind, however, that these and similar terms are associated with appropriate physical quantities for computer operations, and that these terms are merely conventional labels applied to physical quantities that exist within and during operation of the computer.

It should also be understood that manipulations within the computer are often referred to in terms such as listing, creating, adding, calculating, comparing, moving, receiving, determining, configuring, identifying, populating, loading, performing, executing, storing etc. that are often

Petitioner Samsung Ex-1004, 1431

associated with manual operations performed by a human operator. The operations described herein can be machine operations performed in conjunction with various input provided by a human operator or user that interacts with the device, wherein one function of the device can be a computer.

In addition, it should be understood that the programs, processes, methods, etc. described herein are not related or limited to any particular computer or apparatus. Rather, various types of general purpose machines may be used with the following process in accordance with the teachings described herein.

The present disclosure may comprise a computer program or hardware or a combination thereof which embodies the functions described herein and illustrated in the appended flow charts. However, it should be apparent that there could be many different ways of implementing the disclosure in computer programming or hardware design, and the disclosure should not be construed as limited to any one set of computer program instructions.

Further, a skilled programmer would be able to write such a computer program or identify the appropriate hardware circuits to implement the disclosed disclosure without difficulty based on the flow charts and associated description in the application text, for example. Therefore, disclosure of a particular set of program code instructions or detailed hardware devices is not considered necessary for an adequate understanding of how to make and use the disclosure. The inventive functionality of the claimed computer implemented processes will be explained in more detail in the following description in conjunction with the remaining Figures illustrating other process flows.

Further, certain steps in the processes or process flow described in all of the logic flow diagrams below must naturally precede others for the present disclosure to function as described. However, the present disclosure is not limited to the order of the steps described if such order or sequence does not alter the functionality of the present disclosure. That is, it is recognized that some steps may be performed before, after, or in parallel other steps without departing from the scope and spirit of the present disclosure.

The processes, operations, and steps performed by the hardware and software described in this document usually include the manipulation of signals by a CPU or remote server and the maintenance of these signals within data structures resident in one or more of the local or remote memory storage devices. Such data structures impose a physical organization upon the collection of data stored within a memory storage device and represent specific electrical or magnetic elements. These symbolic representations are the means used by those skilled in the art of computer programming and computer construction to most effectively convey teachings and discoveries to others skilled in the art.

At step 201, device 106 can be powered on and activate a WiFi radio 106r. A step 201 can include device 106 loading a bootloader program and an operating system for device 106, as well as starting a process or operating firmware to perform the steps of an initiator 101. At step 202, device 106 can load initiator DPP configuration 130 and initiator key database 101t from nonvolatile memory,

Petitioner Samsung Ex-1004, 1432

where DPP configuration 130 and initiator key database 101t were depicted and described above in connection with Figure 1d.

At step 203, device 106 can select a set of parameters 199 and then all recorded responder bootstrap public keys 102a and at least one initiator bootstrap private key bi 101b for the parameters 199 from an initiator key database 101t. A hash value 101aa for the corresponding initiator bootstrap public keys Bi 101a can be selected or processed in a step 203 as well. In addition, the initiator bootstrap public key Bi 101a can be selected in a step 203.

At step 204, for each set of parameters 199, the initiator 101 in device 106 can generate an ephemeral PKI key pair comprising initiator ephemeral private key pi 101d and corresponding initiator ephemeral public key Pi 101c. Initiator 101 in device 106 can then process or generate a DPP authentication request message 122 for each responder bootstrap public key Br 102a selected in step 203. As depicted in Figure 3a below, the series of steps for processing a message 122 in a step 204 can comprise steps 313 through 316.

At step 205, initiator 101 operating in device 101 can send (i) a DPP authentication request message 122 on the current operating WiFi channel 130b within an operating class 130a for (ii) each responder bootstrap public key Br 102a selected in a step 203. For the first instance of step 205, the first WiFi channel 130b and associated operating class 130a can be the default channel 130b and class 130a selected by device 106 for WiFi radio 106r upon power up in step 201. A step 205 can also comprise sending each DPP message 122 for each key Br 102a multiple times with retry timers, such as waiting a few seconds between sending each DPP message 122. In exemplary embodiments, each DPP message 122 in a step 205 is sent as a broadcast message on the current channel 130b using MAC.I 106n. In exemplary embodiments, for a step 205 each DPP message 122 for each key Br 102a is sent an exemplary 3 to 5 times. Note that for a step 205, multiple different DPP messages 122 for different keys Br 102a can be sent concurrently in order to speed the process for responder 102' operating with responder DPP configuration 131 and a compatible network 107 for key Br 102a to receive the message 122 via WiFi link 192.

At step 206, initiator 101 or device 106 can determine if a DPP authentication response message 123 for a transmitted DPP authentication request message 122 has been received. If a DPP authentication response message 123 has been received, then initiator 101 and device 106 can conclude the step 200 in order to establish communications with a responder 102 or responder proxy 102' (by receiving a DPP authentication response message 123). If a DPP authentication response message 123 has not been received, then initiator 101 and device 106 can proceed to step 207.

At step 207, initiator 101 or device 106 can determine if device 106 has cycled through all available channels 130b on all available operating classes 130a for WiFi radio 106r operating in device 106. If device 106 has not cycled through all available channels 130b on all available operating classes 130a, then device 106 can (i) increment a channel number 130b from the recorded DPP configuration

Petitioner Samsung Ex-1004, 1433

130 or (ii) if device 106 is already operating at the highest channel number 130b in an operating class 130a then change the operating class 130a for radio 106r and implement the lowest channel number in the new operating class 130a, and then return to step 205 to retransmit the collection of DPP messages 122 for each key Br 102a on the new channel number.

At step 207, if device 106 has cycled through all available channels 130b on all available operating classes 130a, then device 106 can (i) store data from a step 204 in nonvolatile memory for access at a later step 200, and (ii) conclude step 200 and enter a period of sleep or dormancy. After a period of time, device 106 could then wake and repeat the series of steps in a step 200 in order to receive a DPP authentication response 123 message from a responder proxy 102'. Note that upon conducting a step 200 again at a later time, for a step 204 device 106 or initiator 101 could read (a) the data previously calculated or derived from a prior step 204 in (b) nonvolatile memory to access data for sending DPP messages 122 again, instead of re-calculating all the steps in 204.

Note that a step 200 can also be conducted for each set of parameters 199 supported by device 106, such as using a first step 200 with a first bootstrap private key bi 101b with a first set of parameters 199, and then a second step 200 concurrently with a second bootstrap private key bi 101b with a second set of parameters 199. Using the step 200 depicted for a device 106 operating an initiator 101 can allow device 106 to automatically receive a DPP authentication response message 123 from a responder proxy 102' when device 106 is powered on and with range of a responder proxy 102' for a responder bootstrap public key Br 102a (where responder 102' communicates with a DPP server 103 that securely records the responder bootstrap private key br 102b). A step 200 could be conducted with a responder 102* from Figure 1b or Figure 1c as well, such that the use of a responder proxy 102' is not required to use a step 200.

Figure 3a

Figure 3a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device database, a DPP server, a responder proxy, and an initiator, in accordance with exemplary embodiments. System 300a can include a device database 104, DPP server 103, responder proxy 102', and initiator 101 operating in device 106. Responder proxy 102' can communicate with DPP server 103 via IP network 113 as depicted in Figure 1a. As contemplated herein, a responder proxy 102' depicted in Figure 3a and subsequent Figures below may also be referred to herein as responder 102', and a responder 102' and responder proxy 102' may be equivalent. In other words, a responder 102' may operate as a responder with some functions and operations for DPP 191 by proxy with DPP server 103.

Before starting steps and message flows depicted in Figure 3a, the device 106 with initiator 101 can record responder bootstrap public keys Br 102a depicted for an initiator key database 101t

Petitioner Samsung Ex-1004, 1434

from Figure 1d, where initiator 101 records at least one key Br 102a. The key Br 102a for a initiator 101 could be received in an "out of band" message 121 from Figure 1b and Figure 1c, or the key Br 102a could be recorded in a device 106 during manufacturing or distribution. Initiator key database 101t for device 106 could also record a plurality of responder bootstrap public keys Br 102a in a table 101ta from Figure 1d. For embodiments supporting mutual authentication, then device 106 can also record at least one initiator bootstrap public key Bi 101a and a corresponding initiator bootstrap private key bi 101b, such as in a table 101tb from Figure 1d. A device 106 with initiator 101 can also operate with at least one device supported configuration 130z from a DPP configuration 130 in Figure 1d. Exemplary PKI keys for initiator 101 in device 106 could also comprise the exemplary keys depicted in Figure 1e. Device database 104 can record data depicted and described in connection with Figure 2a.

Responder 102' can be different than responder 102* in Figure 1b and Figure 1c, since (a) some PKI and cryptographic computations (plus recording of PKI keys) that would normally be performed by a responder 102* as contemplated by DPP specification version 1.0 can (b) be performed by DPP server 103 and thus responder 102' is different than an responder 102*. For system 300a, DPP server 103 and device database 104 may establish a secure session setup 302a, which could comprise establishing a secure communications link between the two servers using protocols such as TLS, IPSec, a virtual private network (VPN), a secure shell (SSH), or similar networking, transport, or application layer technologies in order to establish secure communications between DPP server 103 and database 104. Secure session setup 302a can utilize a certificate cert.DPPS 103c for DPP server 103 and a certificate for database 104 in order to provide mutual authentication and mutual key derivation for a symmetric encryption key in secure session 302a.

At step 303, responder 102 can be powered on and begin operating a WiFi radio 102x with a responder configuration 131 as depicted in Figure 1f. At step 303, responder 102 can operate as an unconfigured responder 102 in Figure 1f, where the subsequent responder configuration step 127 depicted in Figure 3a can convert responder 102 into a configured responder 102'. In other words, responder 102' can comprise a responder 102 that has undergone a configuration step 127. At step 303, responder 102 can use a first MAC address MAC.R 102n to transmit or send and receive wireless data via WiFi link 192. At step 303, responder 102 can use a second, network MAC address MAC.N 102q in order to send and receive data with an IP network 113 and network 107 including DPP server 103. Or, in some exemplary embodiments, the same physical interface and MAC address could be used for both MAC.R 102n and MAC.N 102q.

DPP server 103 and responder 102 can establish a secure session 302b, where secure session 302b could comprise a session using TLS, DTLS, IPSec, a VPN, and other possibilities exist as well. Responder 102 can previously record a URL or domain name to reach DPP server 103. Responder 102 can also use a certificate for DPP server 103 comprising cert.DPPS 103c recorded in responder

Petitioner Samsung Ex-1004, 1435

102 in order to locate and connect with DPP server 103. As depicted in Figure 3a, secure session 302b could also comprise DPP server 103 receiving and using a responder certificate cert.responder 102m in order to authenticate responder 102. Alternatively, DPP server 103 could authenticate a user of responder 102 via a user ID and password entered by user of responder 102 into a user interface 102k for responder 102. In some exemplary embodiments, authentication of a responder 102 or a user of responder 102 for a secure session 302b could be omitted, such as if responder 102 does not record a cert.responder 102m. Secure session 302b could be established using cert.DPPS 103c, similar to a web site providing HTTPS security based on a certificate for the web site (i.e. the session 302b can authenticate the server and secure the session based on the certificate from the server). In exemplary embodiments, responder 102 can authenticate the cert.DPPS 103c using parent certificates up to a recorded root certificate in cert.root 102w from Figure 1f.

After secure session 302b has been established between DPP server 103 and responder 102, DPP server 103 can conduct a step 304. Note that a responder identity 102i for responder 102 could be received by DPP server 103 during setup of secure session 302b. At step 304, DPP server 103 can select a DPP application 102g for responder 102', where DPP application 102g can comprise software or firmware for an unconfigured responder 102 to load and operate in order to function as a configured responder 102' DPP application 102g was depicted and described in Figure 1f. DPP server 103 could select DPP application 102g based on the OS 102e for responder 102, which could be communicated and received by DPP server 103 in setup of secure session 302b. DPP server 103 could obtain DPP application 102g from another server in network 107 or external networks. At step 304, DPP server 103 could also select or process a responder network table 102t for responder 102, where responder network table 102t was depicted and described in connection with Figure 1f. Responder network table 102t can select which DPP server 103 a configured responder 102' should forward an incoming DPP authentication request message 122 based on the responder bootstrap public key hash value 102aa. Although a DPP server 103 is depicted in Figure 3a as conducting secure session 302b and a step 304 and then a step 306, another server associated with network 107 could perform those functions and steps for a DPP server 103.

At step 306, DPP server 103 can select the responder mode 306a of operation for responder 102' to conduct a DPP 191 for receipt and processing of incoming DPP authentication request messages 122 from an initiator 101. The selected mode of operation for responder 102' can comprise a mode 306a or a responder mode 306a, which could be recorded in a server database 103a as depicted and described in connection with Figure 3c below. The selected mode 306a in a step 306 can specify the distribution of PKI keys and related processing between DPP server 103 and responder 102'. The selected mode 306a in a step 306 can determine the subsequent series of steps and message flows between DPP server 103 and responder 102'. The selected mode 306a in a step 306 could be based on the security established or mutual authentication or server authentication in secure session 302b.

Petitioner Samsung Ex-1004, 1436

As one example, if secure session 302b used a certificate cert.DPPs 103c but not a certificate cert.responder 102m (e.g. secure session 302b is less secure or not mutually authenticated or responder 102 is not securely authenticated), then selected mode 306a can comprise a first mode depicted in Figure 3a and Figure 3b, where DPP server 103 records and operates on responder ephemeral private key pr 102r and also all secret shared keys k1 314a, k2 314c, and ke 327a. As a second example, if secure session 302b is mutually authenticated (e.g. used a certificate cert.DPPs 103c and a certificate cert.responder 102m), then a selected mode 306a in a step 306 could comprise a second mode 306a for responder 102' to operate as depicted in Figure 5a, where responder 102' records and operates on responder ephemeral private key pr 102r (and DPP server 103 does not record or operate on the key pr 102r).

As another example, a selected mode 306a in a step 306 could also depend on an identity for device 106, such as MAC.I 106n. For embodiments where the selected mode 306a of operation of a responder 102' depends on the identity of a device 106, then a step 306 to select the mode could be conducted after receipt of a DPP message 122 which would include MAC.I 106n in order to identify the device 106. In other words, in some exemplary embodiments, a preferred responder mode 306a for a device 106 could depend on the device 106, such as if device 106 was a higher-value piece of equipment such as an automobile compared to a potentially lower value networked sensor. For these embodiments, the responder mode 306a for responder 102' could be specified in a device database 104, and for these embodiments the selected mode 306a for responder 102' to operate could be conducted by a step 306 after the receipt of message 312a with data from device database 104.

A first mode 306a could (a) specify that DPP server 103 records all public and private bootstrap and ephemerals keys for responder 102', plus the public keys Bi 101a and Pi 101e for initiator 101 such that (b) resulting cryptographic operations and ECDH key exchanges and symmetric ciphering reside within DPP server 103 and results are transmitted to responder 102'. This first mode 306a could represent the message flows and steps in Figure 3a and Figure 3b, with the recording of keys depicted in Figure 3c below. A second mode 306a could specify that DPP server records and operates on the responder bootstrap private key br 102b, but responder 102' records and operates on the responder ephemeral private key pr 102r, and the second mode 306a is depicted in Figure 5a below. A third responder mode 306a could specify that both responder 102' and DPP server 103 record and operate on responder ephemeral private key pr 102r, such that (b) resulting cryptographic operations and key exchanges are shared between DPP server 103 and responder 102'. This third responder mode 306a could represent the message flows and steps depicted in Figure 5b, with the recording of keys depicted in Figure 3c below.

In exemplary embodiments, all the first, second, and third responder modes 306a specify responder bootstrap private key br 102b residing within DPP server 103 and not shared with responder 102'. By not sharing responder bootstrap private key br 102b with responder 102' (which could be a

Petitioner Samsung Ex-1004, 1437

device with much less security than DPP server 103), the responder bootstrap private key br 102b can remain secured within network 107. These embodiments where responder bootstrap private key br 102b remains in network 107 (e.g. in DPP server 103 and device database 104) support securely sharing the same responder bootstrap public key Br 102a potentially across a plurality of devices 106 with initiators 101 (as well as responder 102' recording key Br 102a). Sharing the same responder bootstrap public key Br 102a across multiple devices supports recording the key Br 102a in a device 106 either (i) during manufacturing or distribution of device 106, or (ii) at a time before any association between device 106 and network access point 105 is known. This simplifies and enables the use of a device provisioning protocol 191 across multiple devices 106 with mutual authentication, where the responder bootstrap public key Br 102a can comprise a shared key 102z. Or, in other embodiments the responder bootstrap public key Br 102a may not be a shared key 102z, but rather unique for a responder 102' (such as a unique responder bootstrap public key Br 102a recorded in a tag 102v), and security is still enhanced because responder 102' may not record or operate on responder bootstrap private key br 102b.

In other words, without the technology described in the present disclosure, a device provisioning protocol with mutual authentication can essentially shift the problem of (a) securely mutually authenticating a device and providing credentials 109 to a device 106 to a new, but related problem of (b) securely providing a responder bootstrap private key br 102b to an responder 102*, where an initiator should record the responder bootstrap public key Br 102a . The first, second, and third modes 306a can solve this shifted/new problem since (i) the responder bootstrap private key 102b can permanently reside in a network 107 and does not need to be sent to the responder 102', and (ii) an initiator can be configured to record key Br 102a before receipt of device 106 by a user (since the same key Br 102a can be written to many different devices). However, with the technology in the present disclosure, the responder 102' can still transmit and receive the full set of a device provisioning protocol 191 messages with an initiator 101 as specified in DPP specification version 1.0.

A fourth mode 306a could specify that responder 102' records all of the responder bootstrap and ephemeral PKI keys, as well as the initiator 101 public keys, such that resulting cryptographic operations and key exchanges reside in responder 102'. This fourth mode 306a could represent the message flows and steps depicted in Figure 1b or Figure 1c, and in this case responder 102' could comprise an initiator 102*. Other possibilities exist as well without departing from the scope of the present disclosure. Additional details for the different responder modes 306a selected in a step 306, with the resulting different allocation or recording of PKI keys and derived symmetric ciphering keys are described below in Figure 3b for a server database 103a.

After a step 306, DPP server 103 can send responder 102 a message 305, where message 305 can include a responder identity 102i, responder network table 102t, responder DPP application 102g, and a responder configuration 131. Note that the responder configuration 131 in message 305 can

Petitioner Samsung Ex-1004, 1438

represent an updated responder configuration to replace or update a previous responder configuration 131 recorded by responder 102. DPP application 102g could comprise a link or URL for responder 102 to download and install DPP application 102g (comprising such as less than an exemplary 100 bytes), or the full software DPP application 102g for responder 102' to operate, which could comprise more than 1 megabyte of data. Responder identity 102i can be useful for a device such as access point 105 operating a responder 102', since in exemplary embodiments an access point 105 could operate more than one responder 102', and the different responders 102' operating in an access point 105 or other device could be identified using the responder identity 102i. In some exemplary embodiments, the responder mode 306a could also be sent in a message 305 from DPP server 103. In other exemplary embodiments, responder mode 306a for responder 102' can also be sent below in a message 312b, after an identity for device 106 such as MAC.I 106n is obtained from a subsequent message 122.

Responder 102 using responder identity 102i in system 300a can receive message 305 and process the data. Responder 102 can download and install the DPP application 102g from a message 305. Responder 102 can load and apply both responder network table 102t and the received responder configuration 131, which could specify radio frequencies or channels to utilize, such as class 131a and channels 131b. Responder 102 can conduct the responder configuration step 127 depicted and described in connection with Figure 1f in order to begin operating as a configured responder 102'. At step 307, responder 102' can begin listening for incoming DPP authentication request messages 122 using WiFi radio 102x and operating with responder configuration 131 in order to communicate with a WiFi radio 106r operating in device 106 with initiator 101, where the communications can use a WiFi data link 192. In exemplary embodiments, DPP application 102g may be (i) included or distributed along with an operating system 102e running on responder 102' or (ii) acquired at a different time than a message 305, and thus from a separate download of DPP application 102g. In exemplary embodiments, WiFi radio 102x operating in responder 102' can listen for message 122 either broadcast or unicast messages or datagrams from device 106 with initiator 101 when using responder configuration 131. In exemplary embodiments, initiator 101 transmits DPP messages 122 using the same or compatible configuration as responder configuration 131.

Device 106 operating initiator 101 can conduct the steps 200 depicted and described in connection with Figure 2b above. In summary, at step 200 in Figure 3a, device 106 may power on from an unpowered or "deep sleep" state. Power could be provided from a battery or converted "wall power". Device 106 could (i) load the equivalent or initiator version of information contained in responder configuration 131 using a bootloader program and (ii) power on WiFi radio 106r. Device 106 can load database 101t for initiator 101 with at least one responder bootstrap public key Br 102a. As a portion of a step 200, device 106 can conduct a step 204 as depicted in Figure 2b. Step 204 can comprise steps 313, 314, 315, and 316. Note that step 200 can comprise a cycle or loop through responder bootstrap public keys Br 102a recorded in database 101t so that initiator 101 send a DPP

Petitioner Samsung Ex-1004, 1439

authentication request message 122 for each responder bootstrap public key Br 102a until initiator 101 successfully receives a DPP authentication response message 123.

A step 313 can comprise initiator 101 deriving an initiator ephemeral private key pi 101d and an initiator ephemeral public key Pi 101c. Step 313 can use a random number generator to produce a random number, which could be input into a key pair generation algorithm using the set of cryptographic parameters 199a. The resulting PKI key values from conducting a step 313 can comprise the initiator ephemeral private key pi 101d and the initiator ephemeral public key Pi 101c. The data for a step 313 comprising derived ephemeral PKI keys for initiator 101 can be recorded in memory of device 106, such that the ephemeral PKI keys can be derived one time until a DPP authentication response message 123 is received from a responder such as responder 102'.

Step 314 can comprise initiator 101 conducting a first initiator key exchange step 314 using an Elliptic Curve Diffie Hellman (ECDH) key exchange 401a, as depicted and described in connection with Figure 4a below. Data input for a step 314 can comprise the initiator ephemeral private key pi 101d and the selected responder bootstrap public key Br 102a from a step 203. Data output from a step 314 can comprise a key k1 314a as described in Figure 4a below. Step 314 can comprise the key exchange and symmetric key derivation step in DPP specification version 1.0 section 6.2.2 in order to for an initiator to obtain key k1 314a. Note that step 314 by initiator 101 can be conducted for each recorded responder bootstrap public key Br 102 in a database 101t. The data for a step 314 can be stored in memory for device 106 such that keys k1 314a for each key Br 102a can be stored and reused for a subsequent "loop" in a step 200 until initiator 101 receives a DPP authentication response message 123, thereby reducing the number of computations and energy required by device 106.

At step 315, initiator 101 can generate a random number for an initiator nonce 317a using parameters 199a. As used herein and throughout the present disclosure, the words "random number" can also mean a pseudo random number, since random numbers generated by nodes may not be truly random. However, their "randomness" may be sufficient in order to securely conduct the cryptographic operations contemplated herein. Initiator 101 can also determine the initiator capabilities, which could comprise an enrollee for the initiator as depicted in Figure 3a. In other embodiments, the responder 102' could have initiator capabilities as an enrollee (where the initiator 101 could be the configurator). A step 315 can comprise encrypting the initiator nonce 317a and initiator capabilities using the derived key k1 from step 314. An encryption step 315 step is further depicted and described in connection with Figure 4a below.

Note that step 315 by initiator 101 can be conducted for (i) each recorded responder bootstrap public key Br 102a in a database 101t with (ii) a corresponding key k1 314a recorded from a step 314. A single initiator nonce 317a can be reused across all encryption steps 315 and the output of all encryption steps 315 comprising ciphertext 315a can be recorded in memory of device 106, such that an encryption step 315 needs only to be conducted once by initiator 101 for each key Br 102a in a

Petitioner Samsung Ex-1004, 1440

database 101t (by reusing initiator ephemeral private key pi 101d and key k1 314a). In other words, initiator 101 can record one ciphertext 315a for each responder bootstrap public key Br 102a in a database 101t, and a subsequent "loop" or generation of a DPP authentication request message 122 (such as transmitting on a different channel 130b) can reuse the ciphertext 315a recorded for the key Br 102a, key pi 101d, and initiator nonce 317a. In this manner, the number of computations and energy use by device 106 can be reduced, by reusing a previously recorded ciphertext 315a when re-transmitting a DPP authentication request message 122 for the same key Br 102a at a later time.

At step 316, initiator 101 can calculate secure hash value 101aa for initiator bootstrap public key Bi 101a and secure hash value 102aa for responder bootstrap public key Br 102a, using a secure hash algorithm 199e. The values are depicted in Figure 3a as "H(Bi)" and "H(Br)", respectively. Initiator 101 can then transmit using radio 106r and channel 130b a DPP authentication request message 122, which can include the hash value 101aa for the initiator bootstrap public key Bi 101a and the hash value 102aa the responder bootstrap public key Br 102a, the derived initiator ephemeral public key Pi 101c, and ciphertext 315a, where ciphertext 315a is encrypted with key k1 314a and includes an initiator nonce 317a and the initiator capabilities.

In exemplary embodiments and as depicted and described in connection with Figure 2b in a step 205 for a step 200 depicted in Figure 3a, a DPP authentication request message 122 for each recorded responder bootstrap public key Br 102a can be transmitted by initiator 101 on channel 130b. In exemplary embodiments, each DPP message 122 can be transmitted as a broadcast message on channel 130b, since initiator 101 may not know a responder 102' MAC address. As depicted for a step 200 above in Figure 2b, initiator 101 may need to repeat messages 122 for each key Br 102a multiple times on several different channels in a channel list 130b, since (i) responder 102' may be listening on a different channel in a responder channel list 131b, and (ii) responder 102' may be using only one of the bootstrap responder public keys Br 102a from a database 101t which corresponds to the key Br 102a recorded and operated by responder 102'.

Although a message 122, message 123, etc. as depicted herein shows key values of "k1", "ke", etc., the depiction of the keys is to illustrate that ciphertext is encrypted with the corresponding key by the use of brackets. In other words, although the key values are shown with the messages, the actual key values are not transmitted inside or with the message in exemplary embodiments. The equivalent syntax is used in DPP specification version 1.0.

Responder 102' can receive a DPP authentication request message 122 in a step 308a and process the message. For a step 308a, responder 102' can compare (i) hash values 102aa for H(Br 102a) received in message 122 with (ii) the recorded hash values 102aa in a responder network table 102t. Or, responder 102' can record the responder bootstrap public key Br 102a and calculate a hash value and compare with the hash value received in a message 122. If (a) the hash value 102aa for H(Br 102a) in message 122 matches or are equal to a recorded or calculated hash value 102aa, then

Petitioner Samsung Ex-1004, 1441

(b) for a step 308b responder 102' can conduct a key validation step on received initiator ephemeral public key Pi 101c described two paragraphs below. Otherwise, responder 102' can continue to monitor or listen for new, different incoming messages 122 where the hash value 102aa would equal a recorded or calculated hash value 102aa for a responder bootstrap public key Br 102a in responder 102.

An identity for device 106 and initiator 101 could also comprise the secure hash value over the initiator bootstrap public key Bi 101a, which can be value 101aa as depicted for message 122 in Figure 3a. In other words, as contemplated throughout the present disclosure, an identity for device 106 and/or initiator 101 can comprise the hash value 101aa over the initiator bootstrap public key Bi 101a. For example, a device database 104 (Fig 2a), a server database 103a (Fig 3c), or responder 102' could record and use hash value 101aa as an identity for device 106 or initiator 101 in order to select keys for conducting key exchange and encryption/decryption steps. The hash value 101aa could be either (i) instead of or (ii) in addition to MAC.I 106n or device identity ID.device 106i in order to identify and track device 106 with initiator 101 across a potential plurality of devices 106 in systems 100, 300, 500, etc.

A step 308b could comprise responder 102' performing the steps for an ECC Full Public-Key Validation Routine in section 5.6.2.3.2 of FIPS publication SP 800-56A (revision 2) for the received initiator ephemeral public key Pi 101c. Alternatively, step 308b can comprise responder 102' performing the steps ECC Partial Public-Key Validation Routine in section 5.6.2.3.3 of the same FIPS publication. Other example steps within a public key validation step 308 can comprise (i) verifying the public key is not at the "point of infinity", and (ii) verifying the coordinates of the point for the public key are in the range [0, p-1], where p is the prime defining the finite field. Other possibilities exist as well for evaluating and validating a received public key is cryptographically secure in a public key validation step 308b, without departing from the scope of the present disclosure. As contemplated in the present disclosure, a responder 102', a DPP server 103, or an initiator 101 can conduct a public key validation step 308b each time a public key is received. In addition, a responder network table 102t can include a set of parameters 199a for each entry corresponding to a key Br 102a, and the set of parameters 199a could be used for key validation steps in a step 308b.

A step 308c can comprise responder 102' selecting a DPP server 103 based on the received hash value 102aa for responder bootstrap public key Br 102a in message 122. A responder 102' could communicate with a plurality of different DPP servers 103 or networks 107, and a key Br 102a could be associated with different DPP servers 103. For example, different networks 107 or DPP servers 103 could record and operate with the responder bootstrap private key br 102b in order to conduct or support a DPP 191 with initiator 101. The responder 102' could use the hash value 102aa and a responder network table 102t as depicted in Figure 1f in order to determine which DPP server 103 should receive data from the incoming DPP authentication request message 122. Figure 3a depicts

Petitioner Samsung Ex-1004, 1442

hash value 102aa for DPP server 103 in network 107 from Figure 1a, although hash value 102aa could be for a different network 107 in exemplary embodiments. In other words, DPP server 103 could comprise a depicted "DPP 103 -1" for responder network table 102t in Figure 1f, and a different hash value 102aa for Br 102a in message 122 could be for a different DPP server 103 comprising a depicted value of "DPP 103 -3" in responder network table 102t in Figure 1f. A step 308c can also include responder 102' starting a timer in order to count the seconds between receiving message 122 and sending upcoming message 123. Additional details regarding the timer from a step 308c are described in a step 324 below.

For some exemplary embodiments, a responder network table 102t as depicted in Figure 1f above could (i) omit the storing or recording of a plurality of responder bootstrap public key hash values 102aa and (ii) include at least one value or identity or domain name for a DPP server 103. Thus, although "DPP server 103" is depicted for a responder network table 102t in Figure 1f, the value recorded could be for an identity or domain name for a DPP server 103, such that responder 102' could forward data received in a message 122 to the DPP server 103 in a subsequent message 309. For a step 308c, a responder 102' could select the DPP server 103 and an identity or domain name for DPP server 103 from the responder network table 102t for sending a subsequent message 309 with data received in DPP authentication request 122. For these embodiments where responder network table 102t omits the storing or recording of a plurality of responder bootstrap public key hash values 102aa, then a DPP server 103 could record a table that stores a plurality of responder bootstrap public key hash values 102aa associated with a plurality of DPP servers 103. DPP server 103 could take actions after receiving a message 309 in order to select another DPP server 103 associated with the responder bootstrap public key hash values 102aa received in a message 309 (for embodiments where the received responder bootstrap public key hash value 102aa from a message 309 does not match the DPP server 103 that receives the message 309).

After steps 308a, 308b, and 308c, responder 102' can send DPP server 103 a message 309 via the secure session 302b. Message 309 can include a responder identity 102i, an identity for initiator 101 and/or device 101 comprising MAC.I 106n, and the data contained in DPP authentication request message 122 received by responder 102'. Note that MAC.I 106n could be observed by responder 102' as the source MAC address for message 122 received via WiFi link 192. Data from message 122 in message 309 can include secure hash value 101aa for initiator bootstrap public key Bi 101a and secure hash value 102aa for responder bootstrap public key Br 102a, the initiator ephemeral public key Pi 101c, and ciphertext 315a. Although not depicted in Figure 3a, a message 309 could also include the responder bootstrap public key Br 102a, for embodiments where responder 102' also records key Br 102a. DPP server 103 can receive message 309 and process the data, including recording the received initiator ephemeral public key Pi 101c in a server database 103a, which is also depicted in Figure 3c

Petitioner Samsung Ex-1004, 1443

below. Before taking steps to process the received ciphertext 315a, DPP server 103 could require additional PKI keys, if they are not already recorded in a server database 103a.

For some embodiments, DPP server 103 can store the equivalent of a responder network table 102t depicted in Figure 1f (and responder 102' may not record the plurality of responder bootstrap public key hash values 102aa for different DPP servers 103). For these embodiments, after receipt of a message 309, DPP server 103 could use the received value of responder bootstrap public key hash value 102aa from a message 309 to query or obtain a different, second DPP server 103 associated with the responder bootstrap public key hash value 102aa. DPP server could forward the data from the message 309 to the different, second DPP server 103. Note that the different, second DPP server 103 could be associated with a different network 107. The different, second DPP server 103 could received the message 309, and subsequently the different, second DPP server 103 could conduct the subsequent steps for a DPP server 103 depicted in Figure 3a and Figure 3b, such as sending a message 312b, conducting the depicted steps of 318 through 315b, sending a message 313 to responder 102', etc.

For embodiments where DPP server 103 does not already record PKI keys for responder 102' and initiator 101, DPP server 103 can send device database 104 a message 310 in order to query for keys and information in order to support a DPP 191 between responder 102' and initiator 101. DPP server 103 can send message 310 through secure session 302a. Message 310 can include an identity for device 101, which could be MAC.I 106n and secure hash value 101aa for initiator bootstrap public key Bi 101a and secure hash value 102aa for responder bootstrap public key Br 102a.

Device database 104 can receive message 310. Device database 104 in step 312 can conduct a query of device database 104 for data associated with the received data from an initiator comprising and identity of MAC.I 106n, hash value 102aa and hash value 101aa. Exemplary data for device database 104 was depicted and described in connection with Figure 2a above. Not all data in the depicted message 310 is required to conduct a query in a step 312, and a hash value 101aa for Bi 101a could be omitted as one example. Or, in exemplary embodiments, the MAC.I 106n may not be recorded in a device database 104, but data for a device 106 could be found using hash value 101aa for Bi 101a. Thus, in exemplary embodiments at least one of MAC.I 106n and hash value 101aa for Bi 101a is received by device database 104 in a message 310, and the at least one MAC.I 106n and hash value 101aa for Bi 101a is recorded in a device database 104.

In exemplary embodiments, a query in a step 312 can select a responder bootstrap private key br 102b, responder bootstrap public key Br 102a, optionally either (i) initiator bootstrap public key Bi 101a or (ii) PKEX key 198 for device 106 from a device database 104. For embodiments where initiator bootstrap PKI keys for device 106 are updated over time, then the most recently updated initiator bootstrap public key Bi 101a can be selected by a database query in a step 312. Additional data can be selected for device 106 and/or initiator 101 in a query at step 312 as well, including a set of parameters 199a to use with bootstrap PKI keys and ephemeral keys and/or a responder mode 306a

Petitioner Samsung Ex-1004, 1444

for DPP server 103 and responder 102' to use when conducting a DPP 191 with initiator 101. Note that a responder mode 306a could be selected for responder 102' in either a step 306 or a query in step 312. The query in step 312 can return values for a message 312a from device database 104 to DPP server 103.

The values in a message 312a can comprise data for a device identity of MAC.I 106n (or hash value 101aa) and the values can include responder bootstrap public key Br 102a, initiator bootstrap public key Bi 101a, responder bootstrap private key br 102b, a device identity ID.device 101i, a selected set of cryptographic parameters 199a for the PKI keys, and optionally PKEX key 198. MAC.I 106n or hash value 101aa can be included in a message 312a so that DPP server 103 can track which of a possible plurality of initiators 101 and devices 106 that message 312a is associated with. Although not depicted for a message 312a, a message 312a could also include a responder mode 306a for device 106 using MAC.I 106n.

DPP server 103 can receive the message 312a and record the data received in a server database 103a. DPP server 103 can (i) use a device identity such as MAC.I 106n or hash value 101aa in order to select or update a responder mode 306a previously selected in a step 306, or (ii) conduct a step 306 at this time in order to select a responder mode 306a recorded for responder 102' in a server database 103a, where a server database 103a is depicted and described in connection with Figure 3c below. As depicted in Figure 3a, DPP server 103 can then send message 312b to responder 102', where message 312b can include (i) the selected responder mode 306a of operation for responder 102' and DPP server 103 to use when conducting a DPP 191 with initiator 101, (ii) a selected set of cryptographic parameters 199a for use with PKI keys and a device provisioning protocol, and (iii) optionally a PKEX key 198 for conducting a public key exchange with initiator 101. In exemplary embodiments, the PKEX key 198 can be included in message 312a for embodiments where device database 104 does not record an initiator bootstrap public key Bi 101a for device 106 with MAC.I 106n. For the responder mode 306a depicted in Figure 3a and Figure 3b, message 312b can comprise a signal that DPP server 103 derives the responder ephemeral private key 102r. Message 312b can also optionally include a device identity such as MAC.I 106n or hash value 101aa in order to associate the data in message 312b with a particular initiator 101, since responder 102' could potentially communicate with a plurality of different devices 106. For some exemplary embodiments, DPP server 103 can send message 312b after establishment of secure session 302b and before DPP server 103 receives a message 309.

For some embodiments, DPP server 103 can (i) store the equivalent of a responder network table 102f depicted and described in connection with Figure 1f, (ii) may not store or operate with the responder bootstrap private key br 102b for responder 102', and (iii) select a different, second DPP server 103 after the receipt of a message 309 as described six paragraphs above. For these embodiments, a message 312b can (i) include an identity or domain name for the different, second

Petitioner Samsung Ex-1004, 1445

DPP server 103 and (ii) omit depicted data of responder mode 306a and parameters 199 and PKEX key 198. For these embodiments, responder 102' (i) could receive the message 312b and use the received identity or domain name of the different, second DPP server 103 in the message 312b to (ii) resend the message 309 to the different, second DPP server 103 received in the message 312b. Responder 102' and the different, second DPP server 103 could establish a second secure session 302b. For these embodiments, after resending the message 309 to the different, second DPP server 103, the subsequent steps and messages depicted in Figures 3a and 3b can apply for the different, second DPP server 103 with the responder 102' (where the different, second DPP server 103 operates as the depicted DPP server 103 in Figure 3a and Figure 3b). For example, the different, second DPP server 103 could send a message 312b to responoder 102', conduct the depicted steps of 318 through 315b, send the message 313 to responder 102', etc.

Responder 102' can receive message 312b from DPP server 103 and conduct a step 312c. In step 312c, responder 102' can use and record the responder mode 306a in order to conduct subsequent steps for responder 102' and messages with DPP server 103 for device 106 with initiator 101. Note that a responder mode 306a could specify the operation of responder 102' with device 106, such that responder 102' could operate with a first responder mode 306a for a first device 106 and a second responder mode 306a with a second device 106. The depicted subsequent steps and messages for responder 102' in Figure 3a with the responder mode 306a depicted as received in a message 312b can comprise a first responder mode 306a depicted as "Fig. 3" in server database 103a. For embodiments where a responder network table 102t includes cryptographic parameters 199a associated with the responder bootstrap public key hash value 102aa, then a message 312b could optionally omit parameters 199a within message 312b, and responder 102' can determine the cryptographic parameters 199a based on the hash value 102aa received from initiator 101 in a message 122.

For embodiments where PKEX key 198 is received in a message 312b (e.g. embodiments where device database 104 may not previously record initiator bootstrap public key Bi 101a before step 312), then responder 102' can conduct a public key exchange protocol exchange 312d with initiator 101 using the PKEX key 198 depicted in Figure 3a. A PKEX 312d can comprise the PKEX protocol as specified in section 5.6 of DPP specification version 1.0. In this manner, responder 102' can receive the key Bi 101a if not previously recorded by a device database 104. Although not depicted in Figure 3a, upon conclusion of a PKEX 312d (or otherwise obtaining an initiator bootstrap public key Bi 101a such as an "out of band" message 126 in Figure 1c), responder 102' can forward the key Bi 101a to DPP server 103 after receipt of the initiator bootstrap public key Bi 101a from a step 312d. The use of a PKEX protocol and key 198 can also be optionally omitted in some embodiments, if initiator bootstrap public key Bi 101a can be recorded by a network 107 prior to a step 312c.

Petitioner Samsung Ex-1004, 1446

As depicted in Figure 3a, the collection of steps and messages in Figure 3a, from step 200 for device 106 through the optional message 312d can be grouped together into a step 301. Step 301 with the collective group of sub-steps and messages depicted for a step 301 can also be performed in other embodiments depicted in Figure 5a and Figure 5b below. Subsequent messages and steps depicted in Figure 3a starting after step 301 can represent a first responder mode 306a for DPP server 103 and responder 102', where the allocation of recording PKI keys and symmetric keys from key exchanges for the first mode 306a are depicted below in Figure 3c. The first responder mode 306a can be depicted in Figure 3c with the label "Fig. 3" is depicted in Figure 3a and Figure 3b.

For the first responder mode 306a selected in a step 306, DPP server 103 can then conduct the series of depicted steps 318 through 315b in Figure 3a in order to generate data for responder 102' to send a DPP authentication response message 123. At step 318, DPP server 103 can also generate two random numbers using a random number generator. The first random number can be used as a responder ephemeral private key pr 102d. The responder ephemeral private key pr 102p can be input into an ECC key pair generation algorithm in order to output a responder ephemeral public key Pr 102p. The set of cryptographic parameters 199a used for the responder ephemeral PKI key pair can match the recorded parameters 199a used with the bootstrap public keys such as responder bootstrap public key Br 102a. The second random number generated by DPP server 103 can be used as a responder nonce 318b. DPP server 103 can also determine responder capabilities 319b, which could comprise the capabilities for a configurator. The responder ephemeral private key pr 102d can also be referred to as responder protocol private key pr 102d, and responder ephemeral public key Pr 102c can be referred to as responder protocol public key Pr 102c. The second random number in a step 318 can comprise a responder nonce value 318b, which can be used in subsequent messages and calculation of authentication values.

DPP server 103 can conduct a first responder key exchange step 319, where the details for a key exchange step 319 are depicted in Figure 4a below. In summary, the received initiator ephemeral public key Pi 101c and the responder bootstrap private key br 102b can be input into an ECDH key exchange in order to mutually derive a shared secret key M 410. The shared secret key M 410 can be input into a key derivation function 402a in order for DPP server 103 to derive mutually shared key k1 314a. A DPP server 103 can then conduct a decryption step 320 as depicted in Figure 4a below. DPP server 103 can input the received ciphertext 315a from a message 122 into a symmetric ciphering algorithm 405b and key k1 314a using parameter 199f in order to read the plaintext 403a, which could comprise the initiator nonce 317a and the initiator capabilities. In exemplary embodiments, the responder 102' can have capabilities of a configurator, and DPP server 103 can select initiator capabilities or an enrollee in a step 320. Steps 319 and steps 320 can also comprise the first four paragraphs of section 6.2.3 of DPP specification version 1.0.

Petitioner Samsung Ex-1004, 1447

DPP server 103 can then conduct a second responder key exchange step 314b, where the details for a key exchange step 314b are depicted and described in connection with Figure 4b below. In summary, the initiator ephemeral public key Pi 101c received in message 122 can be input into an ECDH key exchange algorithm 401a along with the responder ephemeral private key pr 102r derived in a step 318 above in order to output a secret shared key N 411. The value for shared secret key N 411 can be recorded and used in the derivation of a key ke 327a in a third key exchange step 321 below. For a step 314b, the value for the shared secret key N 411 can be input into a key derivation function 402b in order to derive a mutually shared symmetric key k2 314c.

DPP server 103 can conduct a third responder key exchange step 321, where the details for a key exchange step 321 are depicted in Figure 4c below. In summary, both (i) the derived responder ephemeral private key pr 102r from a step 318 with the recorded responder bootstrap private key br 102b, and (ii) the initiator bootstrap public key Bi 101a can be input into an ECDH key exchange algorithm 401c in order to output a secret shared key L 412. For embodiments where a DPP responder authentication 141 is conducted by DPP server 103 without the use of an initiator bootstrap public key Bi 101a, then calculation of a secret shared key L 412 can be optionally omitted in a step 321. Step 321 for DPP server 103 can then comprise a key derivation function 402c using the mutually derived secret shared keys M 410, N 411, and optionally L 412 in order to mutually derive secret shared key ke 327a. An example for a key exchange step 321 is described in point 2 of section 6.2.3 of DPP specification version 1.0 on page 48.

DPP server 103 can then conduct an encryption step 322, where the details for an encryption step 322 are depicted and described in connection with Figure 4c below. Step 322 can also comprise DPP server 103 creating a value R-auth 404, where R-auth 404 can comprise a hash of the initiator nonce 317a from plaintext 403a in step 320, the second random number R-nonce 318b from a step 318 above, and public PKI keys. The value for R-auth 404 in a step 322 can comprise the value in point 3 of section 6.2.3 of DPP specification version 1.0. An encryption step 322 can comprise DPP server 103 (a) encrypting the value for R-auth 404 using the mutually derived key ke 327a from the third key exchange step 321 above, where (b) the output of encryption step 322 can comprise ciphertext 324a.

DPP server 103 can then conduct an encryption step 315b, where the details for an encryption step 315b are depicted and described in connection with Figure 4b below. Ciphertext 324a from an encryption step 322 above can also be encrypted again for an encryption step 315b. The symmetric ciphering key for encryption step 315b can comprise symmetric key k2 314c. Encryption step 315b, encryption step 322, and decryption step 320 can use a common symmetric ciphering algorithm 405a, which could comprise the Advanced Encryption Standard with Synthetic Initialization Vectors (AES-SIV) (and deciphering algorithm) also with a common set of symmetric ciphering parameters 199f. Symmetric ciphering parameters 199f can also specify the use of a block chaining mode such as cipher block chaining (CBC), counter mode (CTR), or Galois/Counter mode (GCM) and other possibilities

Petitioner Samsung Ex-1004, 1448

exist as well. In addition, symmetric ciphering parameters 199f could specify a mode for message authentication, which could comprise a CMAC mode as specified in NIST publication SP-800-38B. In exemplary embodiments, a symmetric ciphering algorithm 405a can comprise the AES-SIV algorithm as specified in IETF RFC 5297. The output from an encryption step 315b can be ciphertext 315c, as depicted in Figure 4b below.

After the encryption step 315b, DPP server 103 can send responder 102' a message 313, where message 313 can include an identity for device 106 such as MAC.I 106n, hash value 102aa for key Br 102a, hash value 101aa for key Bi 101a, the responder ephemeral (or protocol) public key Pr 102p from a step 318 and a ciphertext 315c. The ciphertext 315c can comprise the encrypted data depicted and described in connection with Figure 4b below from the encryption step 315b. As depicted in Figure 3a, ciphertext 315c can be encrypted with key k2 314c and include the responder nonce 318c, the initiator nonce 317a, the responder capabilities (as an exemplary configurator in Figure 3a), and a ciphertext 324a. In other words, ciphertext 324a can comprise data that is encrypted twice, where the first encryption with a key ke 327a can be in a step 322 as depicted in Figure 4c below, and the second encryption step can be with key k2 314c. Ciphertext 324a can include a responder authentication value 404, where the responder authentication value 404 is encrypted by DPP server 103 with key ke 327a.

Responder 102' can receive message 313 and conduct a step 324 to process the message. Note that message 313 and other messages between responder 102' and DPP server 103 can be sent and received via secure session 302b. Responder 102' can conduct a step 324. For a step 324, responder 102' can record the received responder ephemeral public key Pr 102p in a responder network table 102t along with the identity of device 106 comprising potentially MAC.I 106n. In other words, a responder network table 102t can be updated to include information received from network 107 (and DPP server 103) regarding devices 106 communication through network 107 via responder 102'. Step 324 can also include responder 102' conducting a key validation step 308b on the received key Pr 102p from message 313 using parameters 199a from a message 312b above. In exemplary embodiments, if message 313 has been received after more than an exemplary 10 seconds after the start of the timer in step 308c (e.g. maximum exemplary time between message 122 and message 123), then step 324 comprises responder 102' caching the data from message 313 in memory 102f until responder 102' receives a retry message 122 again. Upon responder 102' receiving the message 122 again (this time after message 313 has been received with data from message 313 cached in memory 102f), then responder 102' can send the cached data from message 313 recorded in the step 324.

Using the data received from DPP server 103, responder 102' can then send initiator 101 a DPP authentication response message 123. In exemplary embodiments, message 123 is sent as a unicast message from (i) the responder MAC address MAC.R 102n (which received message 122) to (ii) the initiator MAC address MAC.I 106n (which sent message 122), over (iii) WiFi link 192. As

Petitioner Samsung Ex-1004, 1449

depicted in Figure 3a, message 123 can include hash values of the bootstrap public keys H(Br 102a), H(Bi 101a), the responder ephemeral public key Pr 102p, and the ciphertext 315c. The ciphertext 315c can have encrypted values for (i) R-nonce 318b, (ii) I-nonce 317a, (iii) R-capabilities 318c, and (iv) ciphertext 324a. The ciphertext 315c can be encrypted with mutually derived symmetric encryption key k2 314c. Ciphertext 324a can be encrypted with the mutually derived symmetric key ke 327a and can include a plaintext value for R-auth 404. As described above, R-auth 404 can comprise a hash value over an initiator nonce 317a received and the responder nonce 318b.

As depicted in Figure 3a, initiator 101 can receive DPP Authentication Response message 123. Although not depicted in Figure 3a, in exemplary embodiments, initiator 101 can conduct a step 308b to process and validate the received responder ephemeral public key Pr 102p, such as evaluating that the received key in message 123 is securely compatible with key pi 101d and key bi 101b, using the same steps as a key validation step 308c. In other words, a key validation step 308c by responder 102' may not be as complete as a key validation step 308b by initiator 101, since initiator 101 can compare the received key Pr 102p from a message 123 with data from private keys pi 101d and key bi 101b using parameters 199a for the keys.

Initiator 101 can receive DPP authentication response message 123 from responder 102' and conduct a series of steps in order to process the message and reply to the responder 102'. At step 325, initiator 101 can record the validated, received responder ephemeral public key Pr 102p in a responder public key table 101ta, as depicted in Figure 1d above. In other words, although Figure 1d does not depict a column for key Pr 102p, the column and data could be added for a responder as the data is received. Initiator ephemeral private key pi 101d and initiator ephemeral public key Pi 101c could be recorded in a responder public key table 101ta as well, when the keys can be utilized. Any insertion of key Pr 102p, pi 101d and Pi 101c (for responder 102' identified by key Br 102a into table 101ta preferably has the data deleted upon conclusion of a DPP 191 and receipt and installation of credentials 109 as a configuration object for device 106 with initiator 101.

At step 319a, initiator 101 can conduct a second initiator key exchange step 319a, where initiator 101 can input the received, validated responder ephemeral public key Pr 102p and the derived initiator ephemeral private key pi 101d into an ECDH key exchange algorithm 401b in order to calculate a shared secret key N 411. The second server key exchange step 319a is also depicted and described in connection with Figure 4b below. Shared secret key N 411 can also be used by initiator 101 in a step 327 below. Initiator 101 can input shared secret key N 411 into a key derivation function 402b in order to mutually derive the symmetric encryption key k2 314c.

At step 320b, initiator 101 can use the derived key k2 314c from a step 319a in order to conduct a decryption step 320b on received ciphertext 315c. A decryption step 320b on received ciphertext 315c using key k2 314c is depicted an described in connection with Figure 4b below. Initiator 101 can read the plaintext 403b values from decryption step 320b. The plaintext values 403b can include

Petitioner Samsung Ex-1004, 1450

the initiator nonce 317a from a message 122 and message 309 above. The plaintext values 403b can also include a second ciphertext 324a (e.g. ciphertext 324a can be inside ciphertext 315c). Initiator 101 verifies that the initiator nonce 317a received in message 123 and decrypted in step 320b is equal to the initiator nonce 317a value sent in message 122 above. If the two initiator nonce 317a values are not equal, then initiator 101 can abort the DPP procedure and subsequent steps.

At step 327, initiator 101 can conduct a third initiator key exchange step 327, where initiator 101 can determine the secret shared key L 412 by inputting (i) the an ECC point addition of the responder bootstrap public key Br 102a and the responder ephemeral public key Pr 102p, and (ii) the initiator bootstrap private key bi 101b into an ECDH key exchange algorithm 401c. This third server key exchange step 327 is depicted and described in connection with Figure 4b below. The shared secret key L 412 along with share secret keys M 410 and N 411 can be used with a key derivation function 402c in order to output the mutually derived symmetric ciphering key ke 327a.

At step 328, initiator 101 can conduct a decryption step 328 on the ciphertext 324a using the key ke 327a. A decryption step 328 is depicted and described in connection with Figure 4c below. Initiator 101 can use the decryption step 328 in order to read the plaintext value of R-auth 404. R-auth 404 can comprise a hash 199e value over at least a combination of initiator nonce 317a and the responder nonce 318b. At step 328, initiator 101 can then also internally calculate the same value for R-auth 404, and if the internally calculated value and the received, decrypted value are equal, then the responder 102' can be consider authenticated with initiator 101. The calculation of R-auth 404 by initiator 101 can be according to section 6.2.3 of DPP specification version 1.0.

Initiator 101 can then conduct a step 322b, where step 322b can comprise calculating an initiator authentication value I-auth 404a. Initiator authentication value 404a can be similar to responder authentication value R-auth 404, except the values of initiator nonce 317a can be sequenced before a responder nonce 318b inside a hash 199e calculation over the two nonces. An initiator 101 can then encrypt the initiator authentication value 404a using an encryption step 322 using key ke 327a, as depicted in Figure 4c below. In other words, an encryption step 322 in Figure 4c depicts the encryption of R-auth 404, and encryption step 322b can comprise a initiator 101 encrypting the initiator authentication value I-auth 404a using the key ke 327a derived in a step 327. The resulting ciphertext with the encrypted initiator nonce valued 404a from an encryption step 322b can comprise ciphertext 324b. As depicted in Figure 3a, the collection of steps for an initiator 101 of the depicted steps 325, 319a, through 322b can comprise a step 331. A step 331 in summary allows initiator 101 to process and verify the received DPP authentication response message 123. As depicted in Figure 3a, additional steps for a responder 102' in initiator 101, and DPP server 103 to conduct in order to complete a device provisioning protocol session are depicted in Figure 3b below.

Petitioner Samsung Ex-1004, 1451

Figure 3b

Figure 3b is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a DPP server, a responder proxy, and an initiator, in accordance with exemplary embodiments. Before starting steps and message flows depicted in Figure 3b, the device 106 with initiator 101, responder 102', and DPP server 103 can complete the series of steps and message flows depicted above in Figure 3a. System 300b can include a DPP server 103, responder proxy 102', and initiator 101 operating in device 106. Responder proxy 102' can communicate with DPP server 103 via IP network 113 as depicted in Figure 1a. Responder proxy 102' can communicate with initiator 101 via WiFi link 192.

After completing the series of steps for a step 331 in Figure 3a, initiator 101 can send responder 102' a DPP authentication confirm message 123a. The data and steps to determine and calculate a DPP authentication confirm message 123a can be compatible with section 6.2.4 of DPP specification version 1.0. Note that although the DPP authentication confirm message 123a can be equivalent to the DPP authentication confirm message from section 6.2.4 of DPP specification version 1.0, an important difference can be the operation of responder 102' in Figure 3b versus a responder as contemplated by DPP specification version 1.0. In the present disclosure as depicted in Figure 3b, responder 102' does not record the responder bootstrap private key br 102b. However, using the technology depicted in Figure 3a and Figure 3b, a responder 102' can (i) send and receive messages that are fully compatible with a DPP protocol 191 (including DPP specification version 1.0), but (ii) DPP server 103 can record and operate on key br 102b. In other words, a initiator 101 may not "know" that a responder 102' operates as either (i) an responder 102* from Figure 1b or Figure 1c, or (ii) an responder 102' from Figure 3a and Figure 3b since the messages transmitted and received by the initiator 101 from responder 102' can be fully compatible with a DPP 191.

Other differences besides the location of recording and operating on key br 102b can apply as well for the responder mode 306a depicted in Figure 3a and Figure 3b, such as key pr 102r also remains in DPP server 103 and not recorded or operated on by responder 102'. The recording and operation of responder private keys br 102b and key pr 102r by DPP server 103 can comprise a first responder mode 306a, where responder mode 306a was selected by DPP server 103 or network 107 in a step 306 and communicated to responder 102' in a message 312b.

Responder 102' can listen for either (i) a broadcast message with the hash 199e values for the bootstrap public keys 102aa and 101aa, or (ii) a unicast message to MAC.R 102n from MAC.I 106n for initiator 101 operating in device 106. Responder 102' can receive the DPP authentication confirm message 123a and process the message. DPP authentication confirm message 123a can be processed by initiator 101 using a step 331 as depicted in Figure 3a. DPP authentication confirm message 123a

Petitioner Samsung Ex-1004, 1452

can include an encrypted value for initiator authentication value 404a as a ciphertext 324b, where ciphertext 324b is encrypted with derived key ke 327a in a step 322b.

Upon receipt of DPP authentication confirm message 123a from device 106, responder 102' can forward data in the message to DPP server 103 via secure session 302b. Responder 102' can send DPP server 103 a message 329, where message 329 includes an identity for device 106 (which could be the depicted value of MAC.I 106n) as well as ciphertext 324b. Although not depicted for a message 329 in Figure 3b, ciphertext 324b can also include the hash values for the bootstrap public keys of hash value 102aa and/or hash value 101aa.

DPP server 103 can receive message 329 and record data from the message 329 and process the data. DPP server 103 can use the identity for device 106 comprising MAC.I 106n (or another device identity such as ID.device 106i or hash value 101aa sent in a message 329). DPP server 103 can first conduct a decryption step 328b, where decryption step 328b can use the same steps as a decryption step 328 as depicted in Figure 4c below, with a difference being decryption step 328b operates on the ciphertext 324b containing the encrypted initiator authentication value I-auth 404a instead of the R-auth 404 value. After a step 328b, DPP server 103 can read the plaintext initiator authentication value I-auth 404a.

In step 330, DPP server 103 can compare the decrypted value for I-auth 404a with a calculated value for I-auth 404a and if the two values match then the initiator 101 can be considered by DPP server 103 to be authenticated. The calculation of I-auth 404a for DPP server 103 in a step 330 can comprise a calculation of a secure hash value over at least the responder nonce 318b and bootstrap and ephemeral public keys, with an exemplary calculation for I-auth 404a shown in the sixth paragraph of page 50 of DPP specification version 1.0. Other calculations of an initiator authentication value over the responder nonce 318b are possible as well, without departing from the scope of the present disclosure.

In a step 333, responder 102' can select a set of network credentials 109 for device 106 using initiator 101. The set of network credentials 109 could comprise the exemplary set of credentials 109 depicted and described in connection with Figure 1a. The set of network credentials 109 can comprise an SSID.network-AP 109a, a pairwise master key (PMK) PMK.network-AP 109b, and a set of configuration data comprising config.network-AP 109c. The value PMK.network-AP 109b can also include an identifier for a PMK such as a PMKID. Or, for legacy WiFi connections that utilize WPA2 and similar or earlier standards, PMK.network-AP 109b can comprise a value for a shared secret key such as a PSK. The set of network credentials 109 can comprise values for device 106 to authenticate and encrypt data with network access point 105 in order for device 106 to connect with network access point 105. The network credentials 109 could support use of WiFi link 192 with WPA3 standards from the WiFi Alliance ™ in exemplary embodiments. Note that in some exemplary embodiments, a step 333 for selecting credentials 109 for device 106 can be omitted by responder 102' and the step

Petitioner Samsung Ex-1004, 1453

333 can be performed by DPP server 103 or another server within network 107, such as a server running a RADIUS server, or performing an extensible authentication protocol (EAP) based authentication with device 106.

In a step 334, for embodiments where responder 102' conducts the step 333 to select credentials 109, responder 102' can optionally encrypt the credentials 109 with an asymmetric ciphering algorithm using a public key for initiator 101 or device 106. The use of an asymmetric ciphering algorithms and associated public keys is depicted below in a step 334 in Figure 3e. Responder 102' can send DPP server 103 a message 335, where the message 335 includes an identity for device 106 such as MAC.I 106n, and a ciphertext 334a, where the ciphertext includes the asymmetrically encrypted network credentials 109'. The encryption key depicted in Figure 3b for ciphertext 334a is initiator ephemeral public key Pi 101p, although a different public key for device 106 or initiator 101 could be used as well, such as the initiator bootstrap public key or a public key in certificate cert.device 106m. The device certificate cert.device 106m could be (i) stored in a device database 103d and (ii) sent to DPP server 103 in a message 312a in Figure 3a, and then (ii) sent from DPP server 103 to responder proxy 102' in a message 312b in Figure 3a.

The "optional" notation for a message 335 depends on the source of credentials 109 in a system 300b. If a device operating responder 102' (such as an exemplary access point 105) is responsible for generating or operating credentials 109 (such as a single access point in a user's home), then responder 102' can be the source of the credentials 109 and send the message 335. If a network 107 is responsible for generating or operating credentials 109 (such as in a corporate or campus network of LAN access points 105), including operating EAP-based authentication via RADIUS or similar standards, then the message 335 can be optionally omitted since DPP server 103 could obtain the network credentials 109 from another source in a network 107 besides the device operating responder 102'.

After transmitting the DPP authentication confirm message 123a, in step 332, initiator 101 can take on a roll of enrollee, such that initiator 101 can prepare to send a configuration attribute 330a. Configuration attribute 330a can be used to specify that initiator 101 can take the roll of enrollee and is expecting a set of network credentials 109 from responder 102'. A configuration attribute 330a can be formatted according to section 6.3.4 of DPP specification version 1.0. For the exemplary embodiment depicted in Figure 1a, device 106 with initiator 101 would prefer a role of WiFi client (or "sta") for a configuration attribute 330a, but other possibilities exist as well. For the exemplary embodiment depicted in Figure 7 below a responder 102' operating within an access point 105 would prefer a role of WiFi access point (or "ap") for a configuration attribute 330a.

In step 332, initiator 101 can also generate a third random number, for an enrollee nonce E-nonce 332a. E-nonce 332a and other nonce values in Figure 3a such as the initiator nonce I-nonce 317a and responder nonce R-nonce 318b can be processed using nonce parameters 199g. Initiator 101 can then conduct an encryption step 322c, where encryption step 322c can comprise (i) an encryption

Petitioner Samsung Ex-1004, 1454

step 322 using key ke 327a, and (ii) the plaintext to be encrypted can be at least the E-nonce 332a and the configuration attribute 330a. The output of an encryption step 322c can be a ciphertext 324c. Initiator 101 can then send responder 102' a DPP Configuration Request message 123b, where the configuration request 123b can include the ciphertext 324c, where the plaintext in ciphertext 324c can include E-nonce 332a and the configuration attribute 330a. DPP Configuration Request message 123b can be processed by initiator 101 according to section 6.3.3 of DPP specification version 1.0.

Responder 102' can receive DPP Configuration Request message 123b via connection 192 and using DPP application 102g. As contemplated herein, a DPP application 102g can perform the steps for responder 102' depicted in Figure 3a, such as transmitting/receiving DPP 191 messages using a first MAC.R 102n for connection 192 and transmitting/receiving messages with DPP server 103 using MAC.N 102q via access network 112. DPP application 102g could also perform the depicted steps 308b above to validate the received initiator ephemeral public key Pi 101c. For embodiments where responder 102' uses a WAN radio 102h to connect with IP network 113 and DPP server 103, then DPP application 102g could also use a different identity than MAC.N 102q to identify the physical interface of WAN radio 102h, such as an IMEI, or other possibilities exist as well. For other embodiments MAC.R 102n could also potentially be the same as MAC.N 102q, and other possibilities exist as well for the identification of physical network interfaces used by responder 102' or a device operating responder 102' without departing from the scope of the present disclosure.

Responder 102' can send DPP server 103 a message 336 via secure connection 302b, where message 336 can include the ciphertext 324c. Message 336 can also include an identity for device 106, such as, but not limited to, MAC.I 106n or ID.device 106i, since both DPP server 103 and responder 102' can communicate with a plurality of different devices 106 over time, and DPP server 103 may prefer to know which ciphertext 324c is associated with which device 106 in order to process the ciphertext 324c. As depicted in Figure 3b, message 336 can include the encrypted values for E-nonce 332a and configuration attribute 330a.

DPP server 103 can receive message 336 and process the message 336. DPP server 103 can conduct a decryption step 328c using decryption step 328 depicted in Figure 4c below. A difference between decryption step 328 and decryption step 328c is that decryption step 328c uses the ciphertext 324c as input into the symmetric decryption algorithm 405b. DPP server 103 can use the previously calculated key ke 327a in order to read the plaintext within ciphertext 324c. In a step 328c, DPP server 103 can record the plaintext E-nonce 332a and configuration attribute 330a.

In step 337, DPP server 103 can process the received configuration attribute 330a. For the exemplary embodiment depicted in Figure 1a, configuration attribute 330a can specify that device 106 with initiator 101 may have a role of WiFi client. For a step 337, DPP server 103 can select a set of network credentials 109 for device 106 using initiator 101 in order to create a configuration object. The configuration object in a step 337 can be compatible with a configuration object in section 6.3.6

Petitioner Samsung Ex-1004, 1455

of DPP specification version 1.0. For embodiments where DPP server 103 received a message 335 with an asymmetrically encrypted set of credentials 109' from responder 102', then in a step 337 DPP server 103 can select and use the credentials 109' as the data for a configuration object. If encrypted credentials 109' are selected by DPP server 103 in a step 337, then the credentials 109' after decryption by device 106 using a decryption step 339 can comprise a configuration object per section 6.3.6 of DPP specification version 1.0. As contemplated herein, a reference to DPP specification version 1.0 can also include subsequent and related versions of the standard that would be substantially compatible with version 1.0.

Continuing at step 337, for embodiments where DPP server 103 did not receive a message 335 from responder 102', then DPP server 103 can be responsible for obtaining credentials 109, such as if network 107 is responsible for generating or creating the network credentials 109 for device 106 (such as if access point 105 uses EAP-based authentication and other possibilities exist as well). Note that DPP server 103 could also receive an asymmetrically encrypted set of network credentials 109' from another server or external network 108 in a step 337, where DPP server 103 or network 107 sends an identity for device 106 such as Bi 101a or H(BI) 101aa to the external network 108 and the external network 108 can use the identity 106 to conduct a step 334 for credential 109'. Or, in a step 337, DPP server 103 could receive plaintext network credentials 109 from another server in network 107 or generate the network credentials 109 using data recorded for access point 105 in a server database 103a. Other possibilities exist as well for the source of a set of network credentials 109 for DPP server 103 in a step 337 without departing from the scope of the present disclosure.

Continuing at step 337, DPP server 103 can process a configuration object for network credentials 109, and the configuration object can include an SSID value of SSID.network-AP 109a, a pairwise master key value of PMK.network-AP 109b, and configuration values of config.network-AP 109c. The value of PMK.network-AP 109b can also include a value for PMKID (e.g. PMK identifier), such that PMK.network-AP 109b is used by each of a client and an access point with the PMKID value. The configuration values of config.network-AP 109c in a configuration object could also specify required supporting data for operation of device 106 with an access point 105 after a configuration step 106a, such as an operating class 131a, channel list 131b, an authentication mode for device 106 to operate, which could be PSK, PMK, SAE, EAP, also with any required supporting data and/or identities in order to use the authentication mode. The configuration values of config.network-AP 109c in a configuration object could also include a passphrase or pre-shared key values of PSK for legacy WPA2 networks. In other words, credentials 109 for other authentication schemes between a device and an access point could be supported as well, including EAP-TLS, EAP-PSK, EAP-AKA, EAP-AKA', etc.

As contemplated herein, a set of network credentials 109 can include either (i) network credentials 109 for a selected primary access network, and/or (ii) network credentials 109 for both a

Petitioner Samsung Ex-1004, 1456

selected primary access network and a selected backup access network. Or, in another exemplary embodiment, although a single configuration object of network credentials 109 is depicted in Figure 3b, DPP server 103 could process (i) a first set of network credentials 109 for selected primary access network into a first configuration object and then (ii) a second set of network access credentials 109 for selected backup access network into a second configuration object. At the conclusion of a step 337, DPP server 103 can record and process a configuration object comprising a set of plaintext network credentials 109 or asymmetrically encrypted set of network credentials 109' for device 106 operating initiator 101, where device 106 can be identified by an identity such as ID.device 106i or MAC.I 106n or H(Bi) 101aa.

For a step 338 in Figure 3b, DPP server 103 can conduct a signature creation step 338 as depicted and described in connection with Figure 3d below. The configuration object with network credentials 109 (or encrypted credentials 109') can be signed using the recorded responder ephemeral private key pr 102r, corresponding to public key Pr 102p sent in message 325. The digital signature by DPP server 103 in Figure 3b can comprise a digital signature 338c. Or, in another embodiment the configuration object with network credentials 109 can be signed using the recorded responder bootstrap private key 102b, where initiator 101 could verify the signature 338c with the bootstrap public key Br 102a. Note that some exemplary embodiments contemplate that DPP server 103 receives (i) credentials 109 or asymmetrically encrypted credentials 109' from (ii) another server in network 107 or an external network 108, and a step 338 can comprise DPP server 103 creating a digital signature 338c for the (i) credentials 109 or asymmetrically encrypted credentials 109' from (ii) another server in network 107 or an external network 108. The steps to create and verify a signature with ECC PKI keys can be according to FIPS publication 186-4, and other possibilities exist as well. The use of a digital signature 338c for credentials 109 or 109' is optional in exemplary embodiments, and for some embodiments a digital signature 338c could be omitted (as well as signature creation step 338 and signature verification step 340).

At step 332d, DPP server 103 can conduct an encryption step 332d using an encryption step 322 depicted and described in connection with Figure 4c below. Encryption step 332d can be performed using key ke 327a for device 106. For encryption step 332d, the plaintext values of (i) the E-nonce 332a and (ii) the configuration object comprising network credentials 109 (or asymmetrically encrypted credentials 109') can be input into a symmetric encryption algorithm 405a in order to generate a ciphertext 324d. Note that digital signature 338c can be included in an encryption step 332d as well, although digital signature 338c could also optionally not be included in a ciphertext 324d and sent in a subsequent message 341 and 124 external to ciphertext 324d. As depicted in Figure 3b, the collection of steps 328c through 322d can comprise a step 349.

DPP server 103 can then send responder 102' a message 341, where message 341 includes the ciphertext 324d. Message 338 can also include (i) an identity for device 106 such as MAC.I 106n or

Petitioner Samsung Ex-1004, 1457

ID.device 106i, and (ii) a signature 338c from a signature creation step 338. Message 338 can be sent via secure connection 302b, as depicted in Figure 3b. Although signature 338c is depicted as within ciphertext 324d for a message 341, signature 338c could be external to ciphertext 324d in a message 341.

Responder 102' using DPP application 102g can receive message 341 via secure connection 302b. Responder 102' can use an optional identity for device 106 in a message 341 in order to determine which of a potential plurality of devices 106 is associated with ciphertext 324d in order to forward the data to device 106. Responder 102' can then send initiator 101 a DPP configuration response message 124 via connection 192, where message 124 can include the ciphertext 324d, and the ciphertext 324d can include the E-nonce 332a, the signature 338c, and either network credentials 109 or asymmetrically encrypted network credentials 109'. If network credentials 109' are sent in ciphertext 324d, then the network credentials 109' can comprise a set of "double encrypted" network credentials of (i) asymmetrically encrypted network credentials 109' and (ii) then encrypted again in a ciphertext 324d using the key ke 327a.

Initiator 101 can receive message 124 from responder 102' with ciphertext 324d. Initiator 101 can conduct a decryption step 328d using decryption step 328 depicted in Figure 4c below. A difference between decryption step 328 and decryption step 328d is that decryption step 328d uses the ciphertext 324d as input into the symmetric decryption algorithm 405b. Initiator 101 can use the previously calculated key ke 327a in order to read the plaintext within ciphertext 324d. Initiator 101 can confirm that plaintext in ciphertext 324d includes the previously sent E-nonce 332a. Initiator 101 can extract a plaintext value for (i) network credentials 109 or (ii) asymmetrically encrypted network credentials 109' using symmetric decryption algorithm 405b and key ke 327a. In other words, with credentials 109' decryption step 328d removes the first layer of "double encryption" where the first layer was encrypted with key ke 327a, and the "plaintext" after decryption step 328d for credentials 109' can comprise the asymmetrically encrypted network credentials 109' (e.g. with the first layer of encryption with key ke 327a has been removed).

Initiator 101 can conduct a signature verification step 340 as depicted and described in connection with Figure 3d above in order to verify signature 338c received in message 124. Initiator 101 can use either (i) responder ephemeral bootstrap public key Pr 102p or (ii) responder bootstrap public key Br 102a in order to conduct a signature verification step 340 on digital signature 338c. If plaintext network credentials 109 are included in ciphertext 324d, then initiator 101 can read the plaintext values for credentials 109. If asymmetrically encrypted network credentials 109' are included in ciphertext 324d, then initiator 101 or device 106 operating initiator 101 can conduct an asymmetric decryption step 339 depicted below in Figure 3e in order to convert credentials 109' into plaintext credentials 109.

Petitioner Samsung Ex-1004, 1458

After reading the plaintext network credentials 109 and optionally verifying a signature using recorded public keys 102p or 102a, device 106 can complete the configuration step 106a by loading the network credentials 109 and begin operating a WiFi radio 106r with the network credentials 109. In other words, the WiFi data link 192 depicted in Figure 1a can be converted into a fully authenticated and encrypted WiFi connection using network credentials 109 and standards such as WPA2, WPA3, and also subsequent or related standards where WiFi nodes such as clients or access points use a compatible set of network credentials 109. For example, device 106 could begin scanning for SSID.network-AP 109a on a channel in Config.network-AP 109c, and then send access point 105 a probe request. In this manner, device 106 can obtain connectivity to network access point 105.

Figure 3c

Figure 3c is an illustration of an exemplary server database, with tables for a server database recording exemplary data, in accordance with exemplary embodiments. A server database 103a depicted and described above in connection with Figure 1a and Figure 3a can record data for a DPP server 103 to work with a responder 102' in order for the responder 102' to conduct a device provisioning protocol 191 with a device 106 operating an initiator 101. A server database 103a could record a plurality of tables for a plurality of devices 106 and a plurality of responders 102', including the depicted device credentials table 345. Other possibilities exist as well for the organization, tables, and recorded data within a server database 103a without departing from the scope of the present disclosure. Data within server database 103a could be encrypted using a symmetric key. As depicted in Figure 1a, a DPP server 103 operating in a network 107 could operate and record a server database 103a.

Server database 103a can record a device identity for device 106 of MAC.I 106n, a responder identity which could be MAC address MAC.N 102q, a responder mode 306a comprising a mode of operation for the DPP server 103 and responder 102', a selected set of Cryptographic Parameters 199a, a Responder Ephemeral Public Key Pr 102p, a Responder Ephemeral Private Key pr 102r, an Initiator Ephemeral Public Key Pi 101c, Bootstrap keys 399, an optional PKEX Key 198, a Shared Secret k1 314a, a Shared Secret k2 314c, and a Shared Secret ke 327a. Different identities for device 106 and responder 102' could be recorded in a server database 103a as well, such as ID.device 106i or an identity for a responder 102' that is different than a MAC address for responder 102', such as a serial number or unique identifier for responder 102'. In exemplary embodiments, hash value 101aa over the initiator bootstrap public key Bi 101a can be used for a device identity in a server database 103a.

For exemplary embodiments, not all depicted values may be recorded in a server database 103a, as shown by the depicted value of "-", which can mean that the value is not recorded or stored in a server database 103a. A responder 102' could record and operate on the value depicted as "-" for

Petitioner Samsung Ex-1004, 1459

a server database 103a in Figure 3c for some responder modes 306a. For example, with the row in server database 103a depicted with a mode "Fig. 5a", the responder 102' could derive, record, and operate on the responder ephemeral public key Pr 102p, and DPP server 103 may not need or record the key. Similarly, with the row in server database 103a depicted in Figure 3c as a mode of "Fig. 5a", the responder 102' could derive or operate on shared secret keys k1 314a, k2 314c and ke 327e, while server database 103a would not need to record or operate on the shared secret keys in order for responder 102' with a mode 306a of "Fig. 5a" to conduct a DPP 191 with initiator 101.

Note that in alternative exemplary embodiments, a server database 103a could record the values M 410, N 411, and L 412 respectively instead of key k1 314a, k2 314c, and/or ke 327e, and either server 103 or responder 102' could use the key derivation function 402a, 402b, and/or 402c, respectively in order to convert the values M, N, and L into keys k1, k2, and ke as depicted in Figures 4a through 4c. In exemplary embodiments, the time that a device shared secret key or a derived ephemeral PKI key, or a bootstrap key is recorded in a server database 103a can be minimized to the minimum time necessary to conduct a DPP 191. For example, shared secret keys and ephemeral PKI keys recorded in a server database 103a can be deleted once a configuration step 106a has been completed for a device 101, such as at the time DPP server 103 receives a confirmation that device 106 has begun using the credentials 109 with a network access point 105.

Responder mode 306a can specify an operating mode for responder 102' and DPP server 103, such that a sequence of steps and transmitting/receiving PKI and/or shared secret keys and/or related supporting messages can be specified by the responder mode 306a. A responder mode 306a could also be referred to as a "server mode" 306a, since the mode specifies the operation of a DPP server 103. Or, responder mode 306a could be referred to as a "mode 306a", since both steps for responder 102' and DPP server 102 are affected by the mode. Other terms could be used to refer to the selected mode as well, such as a "method", "procedure", or a "process". Figure 2a above also described a first, second, and third operating responder modes 306a. A first mode is depicted as "Fig. 3" in Figure 3c and can comprise the mode 306a depicted in Figure 3a and Figure 3b. Note that all depicted ephemeral PKI keys and bootstrap PKI keys, as well as all depicted shared secret keys can be recorded in a server database 103a for a responder 102' for the first mode. In other words, when conducting a DPP 191, responder 102' and DPP server 103 could have all the depicted keys recorded in a server database 103a for the first mode. This first mode for an initiator mode 306a can be preferred for embodiments where (i) responder 102' may have a lower level of trust with network 107, or (ii) responder 102' could have a lower level of processing resources, or (iii) an owner for device 106 or a network 107 may prefer that responder bootstrap private key br 102b is not shared externally to network 107, (iv) regulatory requirements where device 106 could comprise a high value or sensitive piece of equipment, or (v) responder 102' conducts a secure session 302b without a responder certificate cert.responder 102m. Other possibilities exist as well for the reasons different modes 306a may be

Petitioner Samsung Ex-1004, 1460

used in a server database 103a to specify the steps for a responder 102' and DPP server 103 to conduct a DPP 191 with device 101.

A second responder mode 306a is depicted as "Fig. 5a" in Figure 3c, where (a) the steps and message flows in Figure 5 for responder 102' and DPP server 103 can comprise (b) the second mode. Note that all relevant bootstrap PKI keys 399 are recorded in a server database 103a for the second responder mode 306a depicted in Figure 5a. The second mode 306a can also omit (i) the initiator ephemeral public key Pi 101c, (ii) and the responder ephemeral private key pr 102r from being recorded in a server database 103a. Note that secret shared keys can optionally not be recorded in a server database 103a for the second mode depicted in Figure 5a. This exemplary embodiment of a second responder mode 306a may be preferred for cases where (i) responder 102' can handle more of the cryptographic operations (e.g. reducing DPP server 103 processing load), but (ii) a network 107 could prefer that the responder bootstrap private key br 102b is not transmitted to or recorded in responder 102'. A third responder mode 306a is depicted as "Fig. 5b" in Figure 3c, where (a) the steps and message flows in Figure 5b for responder 102' and DPP server 103 can comprise (b) the third mode.

A fourth responder mode 306a in Figure 3c can comprise responder 102' operating as a responder 102* depicted in Figure 1b and Figure 1c, and in this case responder 102* can record the keys shown for the third mode 306a in Figure 3b (depicted as "102*). A DPP server 103 can still provide the bootstrap keys 399 shown to responder 102* in this third mode of operation, or the responder 102* could obtain the bootstrap keys 399 from other means, such as "out of band" messages 121 and 126 in Figure 1c. In addition, a responder 102* could obtain the bootstrap keys "in band" (e.g. via WiFi link 192) using PKEX key 198. In exemplary embodiments, a fourth responder mode 306a where responder 102' operates as a responder 102* as depicted in Figure 1b and Figure 1c (e.g. responder 102' would record responder bootstrap private key br 102b), responder 102 can conduct a step 604 as depicted in Figure 6 in order to obtain credentials 109 from a DPP server 103. In other words, although the conventional technology depicted for a responder 102* in Figure 1b and Figure 1c contemplates sending credentials 109 from responder 102* to initiator 101, the conventional technology does not suggest responder 102* obtains the credentials from a DPP server 103. A step 604 as depicted and described below in connection with Figure 6 could be used by a responder 102' with the fourth responder mode 306a in order for responder 102' (operating as a responder 102*) in order to obtain credentials 109 or credentials 109' from a DPP server 103.

Bootstrap keys 399 in a server database 103 can record the bootstrap keys depicted in Figure 3c. For embodiments where a device 106 can support multiple different sets of cryptographic parameters 199a, a server database 103a can record the selected set of cryptographic parameters 199a for communication between a responder 102' and device 106 with initiator 101. The keys for "bootstrap keys 399" can be according to the selected set of parameters 199a. In other words, a device

Petitioner Samsung Ex-1004, 1461

database 104 in Figure 2a above can record all versions of the bootstrap PKI keys for all supported cryptographic parameters 199a for a device 106. However, since a set of cryptographic parameters 199a can be selected for a DPP 191 (e.g. based on data received from initiator 101 in message 122 such as key Pi 101c or hash value 102aa), then the bootstrap keys 399 for that selected set of cryptographic parameters 199a can be recorded in a server database 103a.

A server database 103a can also record a pre-shared PKEX key 198, if PKEX is optionally supported by initiator 101. PKEX key 198 can be used by responder 102' to securely transfer bootstrap public keys in band (i.e. via connection 192 in Figure 1a and Figure 3a). The use of a PKEX protocol to exchange the public keys is described in section 5.6 of DPP specification version 1.0. For embodiments where responder 102' uses PKEX key 198 with initiator 101, the responder bootstrap private key br 102b preferably continues to reside within a network 107 and can be in a server database 103a. Embodiments can omit the use of PKEX key 198, as depicted with the value of "NA" for the key in Figure 3c. Derived, shared secret keys with initiator 101 can also be recorded in a server database 103a. The derived shared secret keys could comprise key k1 314a, key k2 314c, and key ke 327e. The steps to derive and use these keys are depicted and described below in Figure 4a, Figure 4b, and Figure 4c, respectively. As discussed above, some keys may reside in server database 103a, and other keys may reside within a responder 102' (depicted with a "-"), depending on the responder mode 306a.

The present disclosure contemplates other implementations of a device provisioning protocol besides the DPP specification version 1.0. These other implementations could have different mutually derived shared secret keys than those depicted in Figure 3c, and the different mutually derived shared secret keys could be recorded in a server database 103a as well. For example, different key derivation functions 402 could be used to derive a secret shared key than that specified in DPP specification version 1.0. In addition, other or different PKI keys for a responder 102' and/or initiator 101 could be recorded in a server database 103a. For example, the DPP specification version 1.0 could be extended to explicitly include a DPP server 103 public and private key, in order to extend support with a DPP server 103, and in these embodiments the DPP server 103 public and private keys could be included in a server database 103. Other possibilities exist as well for a server database 103a to record PKI keys and shared secret keys without departing from the scope of the present disclosure.

A server database 103a could also record a device credentials table 345. Device credentials table 345 can record a device identity ID.device 106i, MAC.I 106n, a set of primary network credentials 350a, and a set of backup network credentials 350b. The set of primary network credentials 350a can comprise the set of credentials (plus configuration) for a device 106 to use with a selected primary network, such as an exemplary network access point 105 in Figure 1a. The set of backup network credentials 350b can comprise the set of credentials (plus configuration) for a device 106 to use with a selected backup network. The credentials in a device credentials table 345 could be obtained

Petitioner Samsung Ex-1004, 1462

from during a step 333 or a step 337 as depicted and described above in connection with Figure 3b. Note that a DPP server 103 could conduct a step 337 in order to select the network credentials 109, including receiving the credentials 109 from an external network 108. DPP server 103 could receive and record the asymmetrically encrypted network credentials 109' in a device credentials table 345.

For example, with the first device depicted in a device credentials table 345, the credentials in device table for the first device depicted could comprise a set of plaintext credentials 109 depicted in Figure 1a. The depicted designation of "109-1" can identify the specific set of plaintext credentials 109 for a primary network 350a device 106 with ID.device 106i. The device in the second row can use the same plaintext network credentials 109. The credentials stored for the primary network for the third device can comprise depicted credentials "109' -1", which can mean asymmetrically encrypted network credentials 109' are recorded. The asymmetrically encrypted network credentials 109' could be received from an external network 108 (or possibly just securely recorded by a network 107), and only the third device could feasibly decrypt the credentials 109' using a step 339 and a recorded private key for the device 106.

In exemplary embodiments, the asymmetrically encrypted credentials 109' are encrypted using Elgamal encryption and either (i) the initiator bootstrap public key Bi 101a or (ii) the initiator ephemeral public key Pi 101c. The DPP server 103 could send either (i) the initiator bootstrap public key Bi 101a or (ii) the initiator ephemeral public key Pi 101c (along with an identity for device 106 such as ID.device 106i or MAC.I 106n) to a server that conducts an asymmetric encryption step 334 on plaintext credentials 109. The DPP server 103 could receive the encrypted credentials 109' and record them for device 106 with ID.device 106i in a device credentials table 345. Additional details for receiving credentials from an external network 108, and recording the encrypted credentials 109' in a server database 103 are depicted and described below in connection with Figure 3b, depicted as credentials 109'.

Note the present disclosure contemplates that (a) credentials 109 or 109' could utilize wireless networking technology other than WiFi or 802.11 based technology. For example, surrounding wireless networks for device 101 could use exemplary potential 5G wireless technology or LPWAN networking technology potential wireless networks for device 106. Thus, credentials 109 or 109' in a server database 103a and used by device 106 can support networking technology other than WiFi. As one example, an initiator 101 could be operated by an Ethernet interface on a computer or server, and an Ethernet switch could operate a responder 102'. The flow of messages for a DPP 191 protocol could be conducted via a physical wired Ethernet link instead of a WiFi link 192. In this manner, the device operating an initiator 101 could be authenticated and configured by an Ethernet switch operating the responder 102'. As one example, the values and configuration in a dynamic host configuration protocol (DHCP) could be sent as credentials 109 or credentials 109' in a message 124 from an Ethernet switch operating a responder 102'.

Petitioner Samsung Ex-1004, 1463

In another exemplary embodiment, a wireless WAN network, such as a 5G or 6G wireless network according to future 3GPP or ETSI standards, could implement support for the device provisioning protocol contemplated herein. A base station or base transceiver station for the wireless network could operate as a responder 102' and a user device desiring to connect with the network could operate as an initiator 101. The user device and base station could conduct a DPP 191 and the base station could send credentials 109 or credentials 109' for the 5G or 6G wireless network to the user device in a DPP message 124 after successful authentication using DPP. The credentials 109 or 109' for either a wired Ethernet network or a wireless WAN network could be recorded for a device 106 or a plurality of devices 106 in a server database 103a. The use of a primary and backup network for device 101 can provide increased reliability for the operation of device 106, such that if the primary access network for credentials 350a is not available then device 101 could use the selected backup network with credentials 350b in order for device 106 to maintain connectivity to an IP network 113.

Figure 3d

Figure 3d is a flow chart illustrating exemplary steps for creating and verifying a digital signature using PKI keys, parameters, and data input, in accordance with exemplary embodiments. Nodes in a system 100 and other systems herein can create and verify digital signatures using the steps depicted for a signature creation step 338 and a signature verification step 340. In exemplary embodiments, a DPP server 103 or a responder proxy 102 could conduct a signature creation step 338 and an initiator 101 operating in a device 106 can conduct a signature verification step 340. In Figure 3d, signature creation 338 can use the sub-steps of obtaining a message to sign, calculating a message digest 338a, using a private key such as responder ephemeral private key pr 102r, using a signature algorithm 338b, inputting parameters 199i, and calculating a resulting signature 338c. The steps and sub-steps depicted for DPP server creating signature 338 in Figure 3d can also be used by other nodes, including signature creation 338d for a responder 102' to create a signature 338c, where responder 102' creates a signature 338c for network credentials 109 received from a DPP server 103 in Figure 6 below.

Signature algorithm 338a and a corresponding signature verification algorithm for a signature verification step 340 below could comprise an RSA-based digital signature algorithm (DSA) or an ECC based elliptic curve digital signature algorithm (ECDSA), and other possibilities exist as well for signature algorithm 338a and the corresponding signature verification algorithm without departing from the scope of the present disclosure. When using a DSA or ECDSA algorithm in non-deterministic mode for a signature creation 338, a value of "k" or "r", which could comprise a random number can be associated with the digital signature 338c. When using a DSA or ECDSA in deterministic mode for preferred exemplary embodiments, such as specified in IETF RFC 6979 and

Petitioner Samsung Ex-1004, 1464

titled "Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)", which are hereby incorporated by reference, then the requirement for a separately transmitted random number with digital signature 338c (such as value "k" or "r") can be optionally omitted, such that "r" can be deterministically calculated based on the message to sign. In exemplary embodiments, device 106 and servers such as DPP sever 103 can utilize deterministic ECDSA without also sending a random number along with a digital signature, although the value "r" from the deterministic mode could be sent with the digital signature 338c. In other words, a value can be sent with the digital signature 338c that is deterministically derived and associated with the message to sign. In other exemplary embodiments, a random number can be generated a derived value for the random number such as "r" sent with digital signature 338c (e.g. not in deterministic mode).

For a signature creation 338 step, the exemplary message to sign comprises either plaintext network credentials 109 or asymmetrically encrypted network credentials 109'. The message to sign values can be transmitted to the initiator 101 in an encrypted format, such as shown for message 124 in Figure 3b. The message to sign values can be input into a message digest algorithm 338a, which could comprise a standard algorithm such as SHA-256, SHA-3, or similar algorithms. In exemplary embodiments, the message digest algorithm 338a can be specified by a hash value 199e in a selected set of parameters 199a from parameters 199. The output of message digest algorithm 338a can be input along with parameters 199i and a private key such as pr 102d into signature algorithm 338b. Parameters 199i can specify encoding rules, padding, key lengths, selected algorithms, curve names, and other values or fields necessary to utilize a signature algorithm 338b. Both a signature creation step 338 and a signature verification step 340 use the same or equivalent values for parameters 199i. A private key used in a signature creation step 338 can comprise a private key held and recorded by a network 107 where initiator 101 can record the corresponding public key.

Figure 3d depicts the private key as responder ephemeral private key pr 102r, although alternatively the private key could be responder bootstrap private key br 102b or a different compatible private key recorded by network 107. The selection of (i) key pr 102r or key br 102b for a signature creation step 338 and (ii) key Pr 102p or key Br 102a for a signature verification step 340 can be specified in a set of cryptographic parameters 199a, or simply included in software for DPP server 103 and initiator 101. For embodiments where signature 338 uses a private key of responder bootstrap private key br 102b, then initiator 101 could record and use responder bootstrap public key Br 102a. For embodiments where signature 338 uses a private key for network 107, then initiator 101 could record a certificate for the network 107 and use the public key from the certificate for a signature verification step 340.

Signature verification 340 can comprise a step using the sub-steps of (i) obtaining a message to verify, (ii) calculating a message digest 338a for the message to verify, (iii) using a public key corresponding to the private key, such as responder ephemeral public key Pr 102p for responder

Petitioner Samsung Ex-1004, 1465

ephemeral private key pr 102r, (iv) using a signature verification algorithm corresponding to signature creation algorithm 338b, (v) inputting parameters 199i and received signature 338c into the signature verification algorithm, and (vi) determining a pass/fail. If the signature 338c received matches a calculated signature using the public key and message to verify, then the received signature 338c passes or is validated or verified. If the signature 338c does not match the calculated signature in a step 340, then the signature 338c is considered to fail or not be verified. In exemplary embodiments, a device 106 operating an initiator 101 can conduct a signature verification step 340 for a set of credentials 109 (or encrypted credentials 109') before accepting and applying the credentials in a configuration step 106a.

Figure 3e

Figure 3e is a flow chart illustrating exemplary steps for using asymmetric ciphering in order to encrypt credentials using a public key and decrypting credentials using the corresponding private key, in accordance with exemplary embodiments. An encryption 334 step could be performed by (i) a responder 102' as depicted and described in connection with Figure 3b above in order to create a ciphertext 334a, (ii) an external network 108 (different than network 107 operating DPP server 103), where the external network 108 could be the source of a plaintext set of credentials 109 for device 106. The purpose of an encryption 334 step can be to convert a plaintext set of network credentials 109 into the ciphertext 334a, which can comprise and be depicted as an asymmetrically encrypted set of network credentials 109' in Figure 3b, Figure 3c, Figure 6, etc. An exemplary reason that an asymmetrically encrypted set of network credentials 109' may be preferred for receipt by and operation inside DPP server 103 is that DPP server 103 is that a network access point 105 or an external network 108 may prefer that DPP server 103 or responder 102' not be able to read asymmetrically encrypted set of network credentials 109', while device 106 could decrypt and read the plaintext set of network credentials 109'.

An encryption step 334 can use an asymmetric encryption algorithm 351a, where asymmetric encryption algorithm 351a can be based on either RSA or ECC algorithms. The use of an asymmetric encryption algorithm 351a and an asymmetric decryption algorithm 351b with RSA based keys is described in Internet Engineering Task Force Request for Comments (RFC) 8017, which is titled "PKCS #1: RSA Cryptography Specifications Version 2.2", which is hereby incorporated by reference. The use of asymmetric encryption algorithm 351a and an asymmetric decryption algorithm 351b with elliptic curve cryptography can be accomplished with ElGamal asymmetric encryption, as summarized in the Wikipedia article for ElGamal encryption from April 3, 2018, which is herein incorporated by reference. Other possibilities exist as well for the use of ECC algorithms for

Petitioner Samsung Ex-1004, 1466

asymmetric encryption, including IEEE 1363a standards and ISO/IEC standard 18033-2. Other possibilities exist as well without departing from the scope of the present disclosure.

For an encryption step 334, the initiator ephemeral public key Pi 101c can be input into an asymmetric encryption algorithm 351a. As depicted, the plaintext values can be input as well, along with a set of parameters 199h. Parameters 199h can be a subset of parameters 199 depicted and described in connection with Figure 2a above, and can include a set of asymmetric encryption parameters 199h. Note that where initiator 101 or responder 102' is depicted as recording or using a set of parameters 199a, the set of parameters 199a can specify a selected row in a set of parameters 199, such that a single value for parameters 199a can identify all values for 199a through 199h in a set of cryptographic parameters 199. Parameters 199h can specify key lengths, encoding rules, byte orders, use of point compression, and other values necessary to operate an asymmetric ciphering algorithms 351a and 351b. The output of an asymmetric encryption algorithm 351a can be ciphertext 334a.

In a different, but related, exemplary embodiment, the public key used in an encryption step 334 can comprise a public key from certificate cert.device 106m as depicted for device 106 in Figure 1a. The public key from certificate cert.device 106m may be more accessible to an external network 107 or a node than key Pi 101c. The parameters 199h used for a step 334 in this different embodiment could be recorded in the certificate cert.device 106m. Cert.device 106m could be publicly shared by a device manufacturer or device distributor along with a device identity in the certificate such as ID.device 106i or MAC.I 106n. For this different, but related embodiment of encryption step 334, the secret key for device 106 or initiator 101 to utilize in a decryption step 339 below can comprise the secret key SK.device 106s.

As depicted in Figure 3e, and encryption 334 step could also comprise an embodiment of encrypting a plaintext symmetric key using asymmetric encryption, such that the symmetric key could be used with a symmetric ciphering algorithms 405a and 405b such as AES, AES-SIV, or Triple Data Encryption Standard (3DES) or Blowfish, or related algorithms. In this embodiment, (i) the symmetric key for encryption network credentials 109 could comprise a random number derived by responder 102' (or a node for the owner of credentials 109), and (ii) ciphertext 334a could comprise two parts, where the first part is the encrypted symmetric key and the second part is a symmetric encryption of network credentials 109. In other words, for this embodiment where (a) a symmetric key for encrypting/decrypting credentials 109 is asymmetrically encrypted with algorithm 351a, then (b) network credentials 109 would be symmetrically encrypted/decrypted using the asymmetrically encrypted/decrypted symmetric key.

For a decryption step 339, the initiator ephemeral private key pi 101d for initiator 101 in device 106 can be input into an asymmetric decryption algorithm 351b. The initiator ephemeral private key pi 101d can be the secret key corresponding to the public key Pi 101c used with the asymmetric

Petitioner Samsung Ex-1004, 1467

encryption algorithms 351a. Asymmetric decryption algorithm 351b can correspond to asymmetric encryption algorithm 351a, such that if an ECC curve with a given set of parameters 199h is used with asymmetric encryption algorithm 351a, then the same ECC curve with the same or equivalent set of parameters 199h can be used with asymmetric decryption algorithm 351b. The output of an asymmetric decryption algorithm 351b can be the plaintext, as depicted in Figure 3e.

In addition, although credentials 109 are depicted as decrypted in a decryption step 339, an asymmetric decryption algorithm 351b could be used to read a plaintext symmetric key for a symmetric ciphering algorithm, where the plaintext credentials 109 could be read by inputting the plaintext symmetric key and encrypted credentials 109 into a symmetric ciphering algorithm 405b. Standards such as IEEE 1363a standards and ISO/IEC standard 18033-2 describe using asymmetric algorithms and public keys to encrypt symmetric keys, and asymmetric decryption algorithms and private keys to decrypt the asymmetrically encrypted symmetric keys, where (i) the plaintext and ciphertext can be converted using the symmetric key, and then (ii) a second plaintext comprising credentials 109 could be read using the symmetric key and a symmetric ciphering algorithm. In other words, as described in two paragraphs above, the ciphertext 334a could comprise two parts, where (i) the first part comprises a symmetric key asymmetrically decrypted by 351b, and then (ii) the second part comprises the symmetrically encrypted credentials 109, where the symmetrically encrypted credentials 109 are decrypted with the asymmetrically decrypted (e.g. plaintext) symmetric key.

Note that encryption step 334 and decryption step 339 can be used with other plaintext values and ciphertext values as contemplated herein, such as (i) a device 106 asymmetrically encrypting in a step 334 a configuration attribute 330a for an external network 108 using a public key for the external network, and then (ii) the external network 107 asymmetrically decrypting in a step 339 the configuration attribute 330a using a private key. Other possibilities exist as well without departing from the scope of the present disclosure.

Figure 4a

Figure 4a is a flow chart illustrating exemplary steps for conducting a key exchange using PKI keys in order to derive a shared secret key, and for using the derived shared secret key to encrypt and decrypt data, in accordance with exemplary embodiments. Exemplary steps for a responder 102' and an initiator 101 to mutually derive keys can comprise (i) a key exchange 319 step for DPP server 103 or responder 102' (depending on the responder mode 306a) to derive a symmetric encryption key k1 314a and (ii) a key exchange 314 step for a initiator 101 to derive the same symmetric encryption key k1 314a. Exemplary steps for a initiator 101 to encrypt data and responder 102' or DPP server 103 (depending on a mode 306a) to decrypt data can comprise (a) an encryption step 315 for initiator 101 to utilize the symmetric encryption key k1 314a to convert plaintext to ciphertext, and (b) a decryption

Petitioner Samsung Ex-1004, 1468

320 step for responder 102' or DPP server 103 (depending on mode 306a) to utilize the symmetric encryption key k1 314a to convert the ciphertext received from initiator 101 into plaintext.

The use of the steps for a key exchange 314, a key exchange 319, encryption 315, and decryption 320 were also depicted and described in connection with Figure 3a and Figure 3b above. Additional detail regarding the use of these steps will be provided herein. Further, although the steps are depicted specifically for the use of particular keys and plaintext/ciphertext combinations in Figure 4a, the steps illustrated can be used with different keys and plaintext/ciphertext combinations. For example, Figure 4b below depicts the same functionality used by the nodes as in Figure 4a, but with (i) different PKI keys input to mutually derive a different symmetric encryption key, (ii) a different key derivation function and (iii) different plaintext encrypted and different ciphertext decrypted.

A key exchange 314 step for initiator 101 to derive a symmetric encryption key 314a can utilize a selected set of cryptographic parameters 199a as depicted and described in connection with Figure 1d and Figure 1e above. As depicted in Figure 4a, a key exchange algorithm 401a can receive input both of a responder bootstrap public key Br 102a and an initiator ephemeral private key pi 101d. The key exchange algorithm 401a could comprise a Diffie Hellman key exchange (DH), an Elliptic Curve Diffie Hellman key exchange (ECDH), and other possibilities exist as well without departing from the scope of the present disclosure. A key exchange algorithm 401a can support either PKI keys based on elliptic curves or RSA algorithms, although support of elliptic curves may be preferred in some exemplary embodiments due to their shorter key lengths and lower processing requirements.

A summary of ECDH as a key exchange algorithm 401a is included in the Wikipedia article titled "Elliptic Curve Diffie-Hellman" from March 9, 2018, which is herein incorporated by reference. An exemplary embodiment of key exchange algorithm 401a could comprise a "One-Pass Diffie-Hellman, C(1, 1, ECC CDH)" algorithm as described in section 6.2.2.2 on page 81 of the National Institute of Standards and Technology (NIST) document "NIST SP 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" from March, 2007 which is hereby incorporated by reference its entirety. Other key exchange algorithms in NIST SP 800-56A could be utilized as well for a key exchange algorithm 401a in Figure 4a, Figure 4b, and Figure 4c without departing from the scope of the present disclosure. Example calculations for an ECDH key exchange for a key exchange algorithm 401a are shown on pages 88 and 89 of DPP specification version 1.0, where a selected set of parameters 199a can use the ECC curve P-256 with PKI key lengths of 256 bits.

Other algorithms to derive a shared symmetric encryption key using public keys and private keys may also be utilized in a key exchange algorithm 401a (and 401b in Figure 4b and 401c in Figure 4c), such as, but not limited to, the American National Standards Institute (ANSI) standard X-9.63. Cryptographic parameters 199a can also include information for using a key exchange algorithm 401a and a key derivation function 402a in order to derive a commonly shared symmetric encryption key

Petitioner Samsung Ex-1004, 1469

k1 314a. As contemplated herein, the terms "selected set of cryptographic parameters 199a" and "cryptographic parameters 199a", and "parameters 199a" can be equivalent. Parameters 199a input into a key exchange algorithm 401a can include a time-to-live for a key k1 314a that is derived, a supported point formats extension, where the supported point formats extension could comprise uncompressed, compressed prime, or "compressed char2" formats, as specified in ANSI X-9.62. In other words, an ECC keys input into a key exchange algorithm 401a may have several different formats and a set of parameters 199a can be useful to specify the format.

As depicted in Figure 4a, the output of a key exchange algorithm 401a, such as an ECDH key exchange, can comprise a secret shared key M 410. The secret shared key M 410 can be input into both (i) a first key derivation function 402a in order to calculate the key k1 314a and (ii) a second key derivation function 402c in order to calculate a symmetric encryption key ke 327a as depicted in Figure 4c below. Key derivation function 402a could use a hash algorithm 199e from parameters 199a, where an input into the hash algorithm 199e can include the value M 410, in addition to other data such as public keys. As contemplated herein, a secret shared key can also be referred to as a value, since the secret shared key can be a numeric value or a number in the form of a sequence of digits. In DPP specification version 1.0 a key derivation function to determine key k1 314a is depicted on the fourth line of page 48.

Note that in some exemplary embodiments, such a second responder mode 306a depicted and described in connection with Figure 5a below, (i) a key exchange algorithm 401a can be processed by a DPP server 103 and a key derivation function 402a can be processed by an responder 102'. In other words, for some embodiments the key exchange 314b step can be shared or distributed between a DPP server 103 and an responder 102'. Other possibilities exist as well for a key derivation function 402a without departing from the scope of the present disclosure. For example, with other embodiments besides those within DPP specification version 1.0, the output of a key derivation function 402a can be both a symmetric encryption key k1 314a and a message authentication code (MAC) key, where a MAC key can function to verify message integrity of ciphertexts generated with the symmetric encryption key 314a.

As depicted in Figure 4a, an encryption 315 step could be utilized with the derived symmetric encryption key k1 314a in order to convert plaintext 403a into ciphertext 315a. An encryption 315 step can utilize a symmetric encryption algorithm 405a, where symmetric encryption algorithm 405a could comprise an algorithm according to the Advanced Encryption Standard (AES), Blowfish, and Triple Data Encryption Standard (3DES), and other possibilities exist as well. As depicted in Figure 4a, a symmetric encryption algorithm 405a can comprise the AES-SIV algorithm as specified in IETF RFC 5297. Parameters 199f could specify settings for a symmetric ciphering algorithm 405a as well, such as key length and a block cipher modes of ECB, CBC, OFB, and CFB. Parameters 199f may

Petitioner Samsung Ex-1004, 1470

also specify an authenticated encryption mode such as CCM (NIST SP800-38C), GCM (NIST SP800-38D), and other examples exist as well.

Symmetric encryption algorithm 405a can accept input of (i) plaintext 403a, (ii) symmetric encryption key k1 314a, (iii) parameters 199f, and (iv) optionally an initialization vector (not shown in Fig. 4a and not required for AES-SIV), and output ciphertext 315a. Although not depicted in Figure 4a, and encryption 405a step can output a message authentication code (MAC), where the MAC can be calculated with the MAC key from a key derivation function 402a. In exemplary embodiments the use of a MAC and MAC key, as well as an initialization vector can be optionally omitted, such as with algorithm 405a. The exemplary listed values for plaintext and ciphertext in step 315 in Figure 4a were described in connection with depicted step 315 in Figure 3a.

Key exchange 319 step for a responder 102' or DPP server 103 (depending on a mode 306a) depicted in Figure 4a can correspond to key exchange 319 in Figure 3a. Key exchange 319 step can comprise inputting or using the initiator ephemeral Public key Pi 101c (from message 122) and the responder bootstrap private key br 102b into key exchange algorithm 401a, which can comprise the same or equivalent key exchange algorithm 401a depicted and described in connection with key exchange step 314 described above. Other elements or algorithms within a key exchange step 319 can be equivalent to a key exchange step 314 above, including the use of shared parameters 199a. In this manner, responder 102' or DPP server 103 (depending on a mode 306a) and initiator 101 can securely derive the same symmetric encryption key k1 314a as depicted in Figure 4a.

For embodiments with a first mode 306a depicted in Figure 3a, DPP server 103 can conduct key exchange step 319 and for embodiments with a second mode 306a depicted in Figure 5a, a responder 102' can conduct key derivation function 402a and DPP server 103 can conduct key exchange algorithm 401a. For embodiments where DPP server conducts a key exchange step, such as step 401a in Figure 4a, or other key exchange steps such as 401a in Figure 4b below, etc., the appropriate keys for input into the key exchange algorithm can be selected from a database (e.g. 103a for DPP server) or table (e.g. table 102u) using an identity for device 106 received in message 122. In exemplary embodiments, the identity of device 106 can comprise either MAC.I 106n or the hash value 101aa over the initiator bootstrap public key Bi 101a.

A decryption 320 step allows responder 102' or DPP server 103 (depending on mode 306a) to convert the ciphertext 315a received in a message 122 from Figure 3a into plaintext 403a. Decryption 320 step can utilize a symmetric decryption algorithm 405b, which could comprise the same algorithm used in symmetric encryption algorithm 405a except the algorithm being used for decryption instead of encryption. For example, if symmetric encryption algorithm 405a in Figure 4a comprises an AES-SIV encryption, then symmetric decryption algorithm 405b can comprise an AES-SIV decryption. Note that the same values are input into symmetric decryption algorithm 405b as symmetric encryption algorithm 405a, such as symmetric encryption key k1 314a and parameters 199f in order to convert

Petitioner Samsung Ex-1004, 1471

ciphertext 315a back into plaintext 403a. Responder 102' or DPP server 103 (depending on the mode 306a) can the read and process plaintext 403a after a decryption 320 step. For example, with the first mode 306a depicted in Figure 3a and Figure 3b, DPP server 103 can perform decryption step 320 to read plaintext 403a. With the second mode 306a depicted in Figure 5, responder 102' can perform decryption step 320 to read plaintext 403a.

For embodiments where a MAC code is received with ciphertext 315a, initiator 101 can also verify the MAC code using a MAC key from a key derivation function 402a in order to verify the integrity of ciphertext 315a. Note that AES-SIV can have a message integrity check built into the algorithm, and thus a separate MAC code and MAC key can be optionally omitted in some exemplary embodiments. In exemplary embodiments, the successful decryption of a ciphertext into a plaintext using decryption algorithms 405b supports mutual authentication between the two nodes, since successful decryption can only take place when the two nodes record the depicted private keys used for the depicted key exchange.

Figure 4b

Figure 4b is a flow chart illustrating exemplary steps for conducting a key exchange using PKI keys in order to derive a shared secret key, and for using the derived shared secret key to encrypt and decrypt data, in accordance with exemplary embodiments. Exemplary steps for (i) an responder 102' or a DPP server 103 (depending on the mode 306a) and (ii) a an initiator 101 to mutually derive keys can comprise (a) a key exchange 314b step for DPP server 103 or responder 102' (depending on the mode 306a) to derive a symmetric encryption key k2 314c and (b) a key exchange 319a step for a initiator 101 to derive the same symmetric encryption key k2 314c. Exemplary steps for a initiator 101 to decrypt data and responder 102' (or DPP server 103) to encrypt data can comprise (i) an encryption 315b step for responder 102' or DPP server 103 to utilize the symmetric encryption key k2 314c to convert plaintext to ciphertext, and (ii) a decryption 320b step for initiator 101 to utilize the symmetric encryption key k2 314c to convert the ciphertext received from responder 102' into plaintext. The use of the steps for a key exchange 314a, a key exchange 319a, encryption 315b, and decryption 320b were also depicted and described in connection with Figure 3a above.

Note that steps in Figure 4b and the steps in Figure 4a can share some algorithms and values, and the descriptions for the algorithms and values in Figure 4a can be applicable for Figure 4b. For example, a same key exchange algorithm 401a depicted and described in Figure 4a can also be used in Figure 4b. The key exchange algorithm 401a can comprise an ECDH key exchange. The set of parameters 199a depicted and described in Figure 4a can also be used in Figure 4b. Further, a symmetric encryption algorithm 405a and a symmetric decryption algorithm 405b can be the same in

Petitioner Samsung Ex-1004, 1472

Figure 4a and Figure 4b. A set of parameters 199f for a symmetric encryption algorithm 405a and a symmetric decryption algorithm 405b can be the same in Figure 4a and Figure 4b.

A responder 102' or DPP server 103 (depending on a mode 306a) can conduct a key exchange step 314b. In a key exchange step 314b, the initiator ephemeral public key Pi 101c received in message 122 from Figure 3a can be input into an ECDH key exchange algorithm 401a along with the responder ephemeral private key pr 102r derived in a step 318 in Figure 3a in order to output a secret shared key N 411. The value for shared secret key N 411 can be recorded and also used in the derivation of a key ke 327a in a key exchange step 321 below in Figure 4c. For a step 314b, the value for the shared secret key N 411 can be input into a key derivation function 402b in order to derive a mutually shared symmetric key k2 314c. In exemplary embodiments which support DPP specification version 1.0, a key derivation function 402b can comprise the key derivation function for key k2 in point 2 of section 6.2.3 of DPP specification version 1.0. For a first responder mode 306a depicted in Figure 3a, a DPP server 103 can conduct key exchange step 314b and for a second responder mode 306a depicted in Figure 5a, a responder 102' can conduct key exchange step 314b.

An initiator 101 can conduct a key exchange step 319a. At step 319a, a received, validated responder ephemeral public key Pr 102p and a derived initiator ephemeral private key pi 101d can be input into an ECDH key exchange algorithm 401a in order to calculate a shared secret key N 411. Shared secret key N 411 can also be recorded and used in a step 327 below. Initiator 101 can input shared secret key N 411 into a key derivation function 402b in order to mutually derived the symmetric encryption key k2 314c. In exemplary embodiments which support DPP specification version 1.0, a key derivation function 402b can comprise the key derivation function for key k2 in section 6.2.4 of DPP specification version 1.0.

Responder 102' or DPP server 103 (depending on a mode 306a) can conduct an encryption step 315b. The use for an encryption step 315b for a DPP server 103 in a first responder mode 306a is depicted and described in connection with Figure 3a above. The use for an encryption step 315b for responder 102' for a second responder mode 306a is depicted and described in connection with Figure 5a below. Ciphertext 324a can comprise a ciphertext from an encryption step 322 below in Figure 4c. In other words, a responder 102' or DPP server 103 can conduct the encryption step 322 below and also key exchange step 321 below in Figure 4c before conducting the encryption step 315b. As depicted in Figure 4b, the ciphertext 324a can also be encrypted again for an encryption step 315b. The symmetric ciphering key for encryption step 315b can comprise symmetric key k2 314c from a key exchange step 314b in two paragraphs above. The output from an encryption step 315b can be ciphertext 315c, as depicted in Figure 4b. The plaintext 403b for an encryption step 315b can comprise the depicted values of ciphertext 324a, I-nonce 317a, R-Capabilities 318c, and R-nonce 318b.

A decryption 320b step can be performed by an initiator 101. A decryption 320b step converts the ciphertext 315c received in a message 123 from Figure 3a into plaintext 403b. Decryption 320b

Petitioner Samsung Ex-1004, 1473

step can utilize a symmetric decryption algorithm 405b, which could comprise the same algorithm used in symmetric encryption algorithm 405a except the algorithm being used for decryption instead of encryption. Note that the same values (other than ciphertext) are input into symmetric decryption algorithm 405b as symmetric encryption algorithm 405a, such as symmetric encryption key k2 314c and parameters 199f in order to convert ciphertext 315c back into plaintext 403b. Initiator 101 can read and process plaintext 403b after a decryption 320b step. In exemplary embodiments, the successful decryption of a ciphertext into a plaintext using decryption algorithms 405b supports mutual authentication between the two nodes, since successful decryption can only take place when the two nodes record the depicted private keys used for the depicted key exchange.

Figure 4c

Figure 4c is a flow chart illustrating exemplary steps for conducting a key exchange using PKI keys in order to derive a shared secret key, and for using the derived shared secret key to encrypt and decrypt data, in accordance with exemplary embodiments. Exemplary steps for (i) an responder 102' or a DPP server 103 (depending on the mode 306a) and (ii) an initiator 101 to mutually derive keys can comprise (a) a key exchange 321 step for DPP server 103 or responder 102' (depending on the mode 306a) to derive a symmetric encryption key ke 327a and (b) a key exchange 327 step for a initiator 101 to derive the same symmetric encryption key ke 327a. Note that in some exemplary embodiments, such a second responder mode 306a depicted and described in connection with Figure 5a below, (i) a key exchange algorithm 401c can be processed by a DPP server 103 a key derivation function 402c can be processed by an responder 102'. In other words, for some embodiments the key exchange 321 step can be shared or distributed between a DPP server 103 and an responder 102'.

Exemplary steps in Figure 4c for (i) a responder 102' or DPP server 103 to encrypt data and initiator 101 to decrypt data can comprise (i) an encryption 322 step for responder 102' or DPP server 103 to utilize the symmetric encryption key ke 327a to convert plaintext to ciphertext, and (ii) a decryption 328 step for initiator 101' to utilize the symmetric encryption key ke 327a to convert the ciphertext received from responder 102' into plaintext. The use of the steps for a key exchange 321, a key exchange 327, encryption 322, and decryption 328 were also depicted and described in connection with Figure 3a and Figure 3b above.

Note that steps in Figure 4c and the steps in Figure 4a can share some algorithms and values, and the descriptions for the algorithms and values in Figure 4a can be applicable for Figure 4c. For example, the key exchange algorithm 401c can comprise an ECDH key exchange. The set of parameters 199a depicted and described in Figure 4a can also be used in Figure 4c. Further, a symmetric encryption algorithm 405a and a symmetric decryption algorithm 405b can be the same in

Petitioner Samsung Ex-1004, 1474

Figure 4a and Figure 4c. A set of parameters 199f for a symmetric encryption algorithm 405a and a symmetric decryption algorithm 405b can be the same in Figure 4a and Figure 4c.

For a first responder mode 306a depicted in Figure 3a and Figure 3b, a DPP server 103 can conduct a key exchange step 321. For a second responder mode 306a depicted in Figure 5a, a key exchange step 321 can be shared between DPP server 103 and responder 102', such that DPP server 103 conducts a key exchange algorithm 401c and responder 102' conducts a key derivation function 402c. In a key exchange step 321, (i) the initiator ephemeral public key Pi 101c received in message 122 from Figure 3a can be input into an ECDH key exchange algorithm 401c along with (ii) the responder ephemeral private key pr 102r and the responder bootstrap private key br 102b.

An ECDH key exchange algorithm 401c can use (pr 102r + br 102b) modulo q, as specified in DPP specification version 1.0 in point 2 of section 6.2.3, so algorithm 401c for initiator 101 can be slightly different than algorithm 401a for initiator 101 above in Figure 4a and Figure 4b. The output of key exchange algorithm 401c can be a shared secret key L 412. The value L 412 can be input into a key derivation function 402c in order to calculate a shared secret key ke 327a. Note that for a key derivation function 402c, the input can include all of key M 410 from a step 319, key N from a step 314b, and the derived key L 412. As contemplated herein, a key derivation function 402a, 402b, and 402c can also use other data mutually shared between (i) initiator 101 and (ii) responder 102' or DPP server 103 (depending on a mode 306a). In exemplary embodiments which support DPP specification version 1.0, a key derivation function 402c can comprise the key derivation function for key ke 327a in point 2 of section 6.2.3 of DPP specification version 1.0.

An initiator 101 can conduct a key exchange step 327. At step 327, (i) a combination of a recorded responder bootstrap public key Br 102a and received responder ephemeral public key Pr 102p, and (ii) the recorded initiator bootstrap private key bi 101b can be input into an ECDH key exchange algorithm 401c in order to calculate a shared secret key L 412. The recorded responder bootstrap public key Br 102a and received responder ephemeral public key Pr 102p can be combined via elliptic curve point addition. Shared secret key L 412 can be input into a key derivation function 402c, along with shared secret key M 410 from a step 314 and shared secret key N 411 from a step 319a in order to mutually derive the symmetric encryption key ke 327a. In exemplary embodiments which support DPP specification version 1.0, a key derivation function 402c can comprise the key derivation function for key ke 327e in section 6.2.4 of DPP specification version 1.0.

Responder 102' or DPP server 103 (depending on a mode 306a) can conduct an encryption step 322, where the use for an encryption step 322 are depicted and described in connection with Figure 3a above and also Figure 4b above. Plaintext 403c in a step 322 can comprise the responder authentication value 404, where value R-auth 404 can comprise a hash 199e that includes at least the initiator nonce 317a, the second random number R-nonce 318b from a step 318 above in Figure 3a, and public PKI keys such as responder bootstrap public key 102a and responder ephemeral public key

Petitioner Samsung Ex-1004, 1475

102c. The value for R-auth 404 in a step 322 can comprise the value in point 3 of section 6.2.3 of DPP specification version 1.0. The symmetric ciphering key for encryption step 322 can comprise symmetric key ke 327a from a key exchange step 321.

The output from an encryption step 322 can be ciphertext 324a, as depicted in Figure 4c. Note that ciphertext 324a in a step 322 can also be encrypted again in a step 315b above in Figure 4b. In other words, a responder 102' or DPP server 103 (depending on the mode 306a) can conduct the encryption step 322 herein for Figure 4c (along with the key exchange 321 herein in Figure 4b) before conducting the encryption step 315b. For a first mode 306a representing an embodiment depicted in Figure 3a, DPP server 103 can conduct encryption step 322. For a second mode 306a representing an embodiment depicted in Figure 5a, a responder 102' can conduct encryption step 322.

A decryption step 328 can be performed by initiator 101. A decryption 328 step converts the ciphertext 324a received in a message 123 from Figure 3a into plaintext 403c. Decryption 328 step can utilize a symmetric decryption algorithm 405b, which could comprise the same algorithm used in symmetric encryption algorithm 405a except the algorithm being used for decryption instead of encryption. Note that the same values are input into symmetric decryption algorithm 405b as symmetric encryption algorithm 405a in Figure 4c, such as symmetric encryption key ke 327a and parameters 199f in order to convert ciphertext 324a back into plaintext 403c. Initiator 101 or device 106 can the read and process plaintext 403c after a decryption 328 step. In exemplary embodiments, the successful decryption of a ciphertext into a plaintext using decryption algorithms 405b supports mutual authentication between the two nodes, since successful decryption can only take place when the two nodes record the depicted private keys used for the depicted key exchange. In other words, only the nodes could derive key ke 327a in Figure 4c, and thus data that is successfully encrypted by one node and decrypted by the other node using key ke 327a would confirm the nodes are authenticated, since only the nodes recording the private keys depicted in Figure 4c would be able to derive keys ke 327a.

As depicted and described in connection with Figure 3a, a responder 102' or DPP server 103 can also conduct both an encryption step 322 and a decryption step 328. The steps for DPP server 103 to conduct a decryption step 328 for a first responder mode 306a can comprise step 328b and step 328c as depicted and described in Figure 3a. When responder 102' or DPP server 103 conducts decryption step 328 using symmetric encryption key ke 327a, the ciphertext and plaintext will comprise different values than those depicted in Figure 4c. For example, in Figure 3a a DPP server 103 with a first responder mode 306a can use a decryption step 328b (e.g. step 328 but with different ciphertext than that depicted in Figure 4c) in order to decrypt the encrypted initiator authentication value 404a into a plaintext authentication value 404a. Also, in Figure 5a, a responder 102' for a second responder mode 306a can use a decryption step 328b (e.g. step 328 but with different ciphertext 324e than that depicted in Figure 4c) in order to decrypt the encrypted initiator authentication value 404a.

Petitioner Samsung Ex-1004, 1476

Further, an initiator 101 can use a decryption step 328d to decrypt network credentials 109 from a responder 102'.

Note that in some exemplary embodiments, the "plaintext" network credentials 109' (for the purposes of a step 328d in Figure 5a) can comprise network credentials 109' that can also be encrypted with an asymmetric ciphering algorithm such as Elgamal using a step 351a in Figure 3e above. In other words, network credentials 109' can comprise credentials that are "double encrypted" with both (i) a symmetric key 327a (which can be decrypted by initiator 101 using a decryption step 328d, where decryption step 328d uses the depicted decryption 328 in Figure 4c), and then (ii) an asymmetric ciphering algorithm 351b such as Elgamal from a step 339. In other words, the output of a decryption step 328d could be input into an asymmetric decryption algorithm 351b, so that the input into asymmetric decryption algorithm 351b could be decrypted by initiator 101 using either (i) initiator ephemeral private key pi 101d or (ii) initiator bootstrap private key bi 102b.

Figure 4d

Figure 4d is a flow chart illustrating exemplary steps for conducting key exchanges using PKI keys in order to derive a first and a second shared secret keys, where ECC point addition is performed on output from a key exchange algorithm in order to derive a third shared secret key. The exemplary key exchange step 421 can be used by a DPP server 103 in order to derive a key L(a) 421a. The exemplary key exchange step 422 can be used by a responder 102' in order to derive a key L(b) 422a. A key derivation step 423 can combine key L(a) 421a and key L(b) 422b in order to calculate or derive the key L 412, using ECC point addition on key L(a) 421a and key L(b) 422b. In this manner, key L 412 can be calculated through the use of two different key exchange steps 421 and 422 in some exemplary embodiments such as those depicted and described in connection with Figure 5a below, instead of the single key exchange step 321 depicted in Figure 4c.

Key exchange 421 can use an ECDH key exchange algorithm 401a, where the input comprises the initiator bootstrap public key Bi 101a and the responder bootstrap private key br 102b. The set of mutually agreed parameters 199a between with an initiator 101 can be used with a key exchange step 421. The output of ECDH key exchange algorithm 401a can comprise key L(a) 421a. In exemplary embodiments, where DPP server 103 conducts the key exchange step 421, the responder bootstrap private key br 102b can remain in DPP server 103 or within a network 107 and does not need to be shared with responder 102'. In addition, although a DPP server 103 is described herein for a Figure 4d as performing the key exchange step 421, another server on an external network 108 could (i) record the responder bootstrap private key br 102b and (ii) perform a key exchange step 421, and the output of key exchange step 421 in the form of key L(a) 421a could be sent to a DPP server 103 or

Petitioner Samsung Ex-1004, 1477

responder 102', such as in a message 503 depicted in Figure 5a below. Or, DPP server 103 could be operated by an external network 108 and communicate with a responder 102' via a secure session 302b. Key L(a) 421a could comprise a shared secret key, although in exemplary embodiments an initiator 101 may not need to calculate the corresponding key L(a) 421a directly, but rather initiator 101 can calculate a shared secret key L 412 using step 327 as depicted in Figure 4c.

Key exchange 422 can use an ECDH key exchange algorithm 401a, where the input comprises the initiator bootstrap public key Bi 101a and the responder ephemeral private key pr 102r. The set of mutually agreed parameters 199a between with an initiator 101 can be used with a key exchange step 422. The output of ECDH key exchange algorithm 401a can comprise key L(b) 422a. In exemplary embodiments, where responder 102' conducts the key exchange step 422, the responder does need access to the responder bootstrap private key br 102b, which can remain in DPP server 103 or within a network 107 and does not need to be shared with responder 102'. Key L(b) 422a could comprise a shared secret key, although in exemplary embodiments an initiator 101 may not need to calculate the corresponding key L(b) 421a directly, but rather initiator 101 can calculate a shared secret key L 412 using a step 327 as depicted in Figure 4c.

A key derivation step 423 can (i) combine the output of key exchange steps 421 and 422 in order to calculate or derived the shared secret key L 412 and then (ii) perform a key derivation function step 402c on the derived or calculated shared secret key L 412 in order to determine or calculate shared secret key ke 327a, which can be also mutually derived by initiator 101, where initiator 101 uses the key exchange step 327 depicted and described in connection with Figure 4c. In exemplary embodiments, a responder 102' can conduct the key derivation step 423 using (i) the value L(a) 421a received from a DPP server 103 (where receipt of L(a) 421a by responder 102' can be in a message 503 as shown in Figure 5a below), and (ii) the value or key L(b) 422b output from a key exchange step 422 in the paragraph above.

Key derivation step 423 can comprise two primary steps. A first step in key derivation 423 can comprise an ECC point addition 424 on the value L(a) 421a and the value L(b) 422a. The result of the ECC point addition will be equal to the value L 412, where the value L 412 could also be calculated without ECC point addition 424 using a step 321. Exemplary calculations for an ECC point addition 424 can comprise the calculations shown for point addition in the Wikipedia article for "Elliptic Curve Point Multiplication" dated May 15, 2018, which is herein incorporated by reference in its entirety. As depicted in Figure 4c, the calculation of L 412 using an ECC point addition 424 over L(a) 421a and L(b) 422a will equal the value for L 412 calculated by a key exchange algorithm 401c from a step 321 in Figure 4c. A second step in key derivation step 423 as depicted in Figure 4d can comprise a key derivation function step 402c using input from ECC point addition step 424 (e.g. value L 412 output from step 424), where the output of key derivation function step 402c can comprise key ke 327a, which is also depicted in Figure 4c above.

Petitioner Samsung Ex-1004, 1478

By conducting a key derivation step 423 as depicted in Figure 4c (where DPP server 103 conducts the calculations using the responder bootstrap private key br 102b), (i) responder 102' can calculate key ke 327a without recording or operating on the responder bootstrap private key br 102b, and (ii) the responder bootstrap public key Br 102a could comprise a shared key 102z as depicted in Figure 2a across multiple different devices 106. Although the responder 102' can receive the key or value L(a) 421a, the responder bootstrap private key br 102b remains secure, because responder 102' cannot feasibly calculate or determine the responder bootstrap private key br 102b using a received key L(a) 421a. In other words, the use of a key derivation step 423 can allow a responder bootstrap public key Br 102a to be securely shared between multiple devices 106 in order to conduct a DPP 191, since a responder 102' can conduct a DPP 191 as contemplated herein without recording, receiving, or operating on a responder bootstrap private key 102b.

Exemplary data and numbers can be provided to demonstrate the calculations for key exchange steps 421, key exchange step 422, and key derivation step 423 using an ECC point addition 424. The exemplary data can comprise decimal numbers for the example data and calculations listed in "Test vectors for DPP Authentication using P-256 for mutual authentication" on pages 88 and 89 of the DPP specification version 1.0. Parameters 199a can comprise the elliptic curve 199c of "secp256r1" with key lengths 199b of 256 bit long keys.

The responder bootstrap private key 102b can comprise the exemplary following number, and may be recorded in a DPP server 103:

38358416135251014160802731750427376395128366423455574545250035236739593908128

The responder ephemeral private key 102r can comprise the exemplary following number, and may be recorded by a responder 102':

111991471310604289774359152687306247761778388605764559848869154712980108827301

The initiator bootstrap public key Bi 101a can comprise the following exemplary values with X and Y numbers (or "coordinates") of:

X:

61831688504923817367484272103056848457721601106987911548515219119661140991966

Y:

43682127411605262630763685096978902757372085459561282092692249825509082694 4

Petitioner Samsung Ex-1004, 1479

Key exchange step 421 for an ECDH algorithm key exchange 401a can input the initiator bootstrap public key Bi 101a and the responder bootstrap private key br 102b (both with numbers above) in order to calculate a key L(a) 421a. An exemplary number or value for key L(a) 421a from the values above using parameters 199a can be:

X:
114900471986805225153115909625996714820294170643513373033139066428057435731 19
Y:
279339665602382047312450979433990845238094818334347544097236049703660820218 55

Key exchange step 422 for an ECDH algorithm key exchange 401a can input the initiator bootstrap public key Bi 101a and the responder ephemeral private key pr 102r (both with numbers above) in order to calculate a key L(b) 422a. An exemplary number or value for key L(b) 422a from the values above using parameters 199a can be:

X:
789447196517742066982505887015825706335031829034153942430065294811891581946 50
Y:
112277127029246845818349358288374891402018204245360629120510863823245894452 37

An ECC point addition 424 for the above two derived points L(a) 421a (from keys Bi 101a and br 102b) and L(b) 422b (from keys Bi 101a and pr 102r) will result in the following point that also equals L 412.

X:
113734500629065545557893524064610113740858966831672649615565042035695230713090
Y:
689614296913074291667967608810956893480888757713349706445933063883757419652 62

Note that key L 412 can also be derived by initiator 101 from a key exchange step 327 using ECDH key exchange algorithm 401c. ECDH key exchange algorithm 401c can comprise initiator 101 calculating L from keys Br 102b, Pr 102p, and bi 101b, represented by (Br 102b + Pr 102p) * bi 101b. Using hexadecimal representations of the above keys Br 102b, Pr 102p, and bi 101b from the same keys values in DPP specification version 1.0 and exemplary numbers above, the X,Y numbers or point or "coordinates" derived by initiator 101 with algorithm 401c for shared secret L on page 89 equals:

Petitioner Samsung Ex-1004, 1480

X: fb737234 c973cc3a 36e64e51 70a32f12 089d198c 73c2fd85 a53d0b28 2530fd02
Y: 9876c937 b642cce8 f604bef0 c24ce0da d2c27867 08a7575a f940a58c fe8ca3ce

The hexadecimal representation of shared secret L using keys Br 102b, Pr 102p, and bi 101b by initiator 101 equals the decimal representation of shared secret key L 412 from the ECC point addition 424 by a responder 102', where ECC point addition 424 used as input key L(a) 421a and key L(b) 422a.

After an ECC point addition 424, a key derivation step 423 in Figure 4c can input the shared secret key L 412 output from the ECC point addition 424 into a key derivation function 402c. The key derivation function 402c can comprise the same key derivation function 402c as depicted and described in connection with Figure 4c above. The key derivation function 402c can also use input of shared secret key M 410 from Figure 4a in a step 319 and shared secret key N 411 from a step 314b. The output of the key derivation function 402c can be shared secret key ke 327a. An exemplary key derivation function 402c could comprise the key derivation function in point 2 of page 48 of DPP specification version 1.0. Note that initiator 101 can mutually derive the same value or number or "coordinate" for shared secret key ke 327e using a key exchange step 321. As contemplated herein, the use of the words or description "shared secret key" can also mean "shared secret". Thus, "shared secret key M 410" can be the same as "shared secret M 410", "shared secret key N 411" can be the same as "shared secret N 411", etc.

Figure 5a

Figure 5a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device database, a DPP server, a responder proxy, and an initiator, in accordance with exemplary embodiments. System 500a can include a device database 104, DPP server 103, responder proxy 102', and initiator 101 operating in device 106. Before starting steps and message flows depicted in Figure 5a, the initiator, DPP server, and device database depicted for system 500a in Figure 5a may previously complete exemplary message flows and steps depicted for a step 301 in Figure 3a above. Responder 102' can use data received from initiator 101 in a message 122 from step 301 (such as initiator ephemeral public key Pi 101c) in order to conduct the steps and process the messages depicted for a responder 102' in Figure 5a.

Responder proxy 102' can communicate with DPP server 103 via IP network 113, also as depicted in Figure 3a above. As contemplated herein, a "responder proxy 102' " depicted in Figure 5a may also be referred to as responder 102', and a responder 102' and responder proxy 102' may be equivalent. A difference between system 500a in Figure 5a and system 300 in Figure 3a can be that (i) responder ephemeral private key pr 102r, with corresponding (ii) responder ephemeral public key

Petitioner Samsung Ex-1004, 1481

Pr 102p are derived and recorded by a responder 102' instead of a DPP server 103. In exemplary embodiments depicted in Figure 5a the responder bootstrap private key br 102b can remain recorded in a DPP server 103 and not exposed to responder 102'. In exemplary embodiments depicted in Figure 5a the responder ephemeral private key pr 102r can remain recorded in responder 102' and not exposed to DPP server 103. The exemplary embodiment depicted in Figure 5a can comprise a second mode 306a of operation for responder 102' and DPP server 103, as depicted and described in connection with Figure 3c above.

At step 502, DPP server 103 can select a mode 306a for both responder 102' and DPP server 103. Note that the mode 306a selected is not normally observable to an initiator 101, since the full and complete set of DPP 191 messages between initiator 101 and responder 102' can be transmitted and received between the two nodes. In other words, (a) the different distribution of PKI and shared secret keys in a server database 103a and responder 102' for a first mode 306a in Figure 3a and a second mode 306a in Figure 5a (b) would not normally be observable to an initiator 101. The recording of PKI and shared secret keys in a server database is depicted in a server database 103a in Figure 3c, where the value "-" can mean that the key is recorded and/or operated by responder 102'. Step 502 can also comprise DPP server 103 selecting a responder bootstrap public key Br 102a and a responder bootstrap private key br 102b and initiator bootstrap public key Bi 101a for device 106 using MAC.I 106n, although a different value in server database 103a could be used as well, such as device identity ID.device 106i or hash value 101aa. Selecting keys Br 102a and br 102b in a step 502 can use the parameters 199a received in a message 312a in Figure 3a, in order to select keys that are compatible with received initiator ephemeral public key Pi 101c from a message 122 in Figure 3a. Also, although a step 502 is depicted in Figure 5a, a step 502 could be used in other Figures and embodiments herein, include embodiments depicted in Figure 3a and Figure 5b.

DPP server 103 can then conduct an ECDH key exchange step 401a using as input (i) the responder bootstrap private key br 102b and (ii) the initiator ephemeral public key Pi 101c received in a message 122 from Figure 3a during a step 301. This ECDH key exchange step 401a is depicted and described in connection with a key exchange step 319 depicted and described in connection with Figure 4a. The output of step 401a can be shared secret M 410. DPP server 103 can send (i) shared secret M 410 along with a device identity such as MAC.I 106n (or ID.device 106i or hash value 101aa) to (ii) responder 102' in a message 503. The use of a device identity in message 503 can be helpful for embodiments where responder 102' communicates with a plurality of devices 106 either simultaneously or in sequence.

Responder 102' can receive message 503 with shared secret M 410 and conduct a series of steps in order to prepare a DPP authentication response message 123. A step 318 in Figure 5a can comprise the same steps 318 as performed by DPP server 103 in Figure 3a, except responder 102' performs the actions in Figure 5a instead of DPP server 103 performing the actions. In summary for

Petitioner Samsung Ex-1004, 1482

a responder 102' in Figure 5a, responder 102' can derive two random numbers using a random number generator. The first random number can be used as a responder ephemeral private key pr 102r and the second random number can be used as a responder nonce 318b. Parameters 199a can specify the format and length of key pr 102p and nonce 318b in a step 318 for responder 102. Responder 102' can also derive a responder ephemeral public key Pr 102p using the derived key pr 102r. DPP server 103 can also determine responder capabilities 319b, which could comprise the capabilities for a configurator.

Responder 102' can process a key derivation function 402a using the received shared secret M 410 from a message 503. The key derivation function 402a can be equivalent to a function 402a depicted and described in connection with Figure 4a. Shared secret M 410 can be input into key derivation function 402a, and the output can be shared secret key k1 314a. Responder 102' can then perform a decryption step 320 using the derived key k1 314a, where a decryption step 320 is depicted and described in connection with Figure 4a. Decryption step 320 can be performed by responder 102' on the ciphertext 315a received in a DPP authentication request message 122. After a decryption step 320, responder 102' can read plaintext 403a with the values of the initiator nonce 317a and the initiator capabilities. For the embodiment depicted in Figure 5a, the initiator 101 can have the capabilities of an enrollee, although other possibilities exist as well without departing from the scope of the present disclosure.

Responder 102' can then conduct a second responder key exchange step 314b, where the details for a key exchange step 314b are depicted and described in connection with Figure 4b above. In summary, the initiator ephemeral public key Pi 101c received in message 122 can be input into an ECDH key exchange algorithm 401a along with the responder ephemeral private key pr 102r derived in a step 318 above in order to output a secret shared key N 411. The value for shared secret key N 411 can be recorded and used in the derivation of a key ke 327a in a key derivation function 402c below by responder 102'. For a step 314b, the value for the shared secret key N 411 can be input into a key derivation function 402b in order to derive a mutually shared symmetric key k2 314c. As depicted in Figure 5a, the collection of steps and messages beginning with a step 502 by DPP server 103 though key exchange step 314b can comprise a step 501, where a step 501 can also be used by a DPP server 103 and responder 102' in Figure 5b below.

In Figure 5a, a DPP server 103 can then conduct or calculate an ECDH key exchange step 421, where key exchange step 421 was depicted and described in connection with Figure 4d above. In summary, the initiator bootstrap public key Bi 101a and the responder bootstrap private key br 102b can be input into an ECDH key exchange algorithm 401d using a set of parameters 199a. The output of key exchange step 421 can comprise a value L(a) 421a. Note that step 421 can be performed before a step 501 or concurrently with a step 501. DPP server 103 can then send responder 102' a message 504, where message 504 include an identity for device 106 such as MAC.1 106n, the initiator bootstrap

Petitioner Samsung Ex-1004, 1483

public key Bi 101a, and the calculated value L(a) 421a from a key exchange step 421. Message 504 can be sent via secure session 302b established during a step 301 from Figure 3a.

Responder 102' can receive message 504 and begin performing a series of steps in order to process the data and prepare a DPP authentication response message 123. Responder 102' can conduct a key exchange step 422 in order to calculate a value L(b) 422a. In summary, the responder ephemeral private key pr 102r can be input into an ECDH key exchange algorithm 401a along with the initiator bootstrap public key Bi 101a, using a set of parameters 199a in order to calculate the value or key L(b) 422a using a key exchange step 422. Key Bi 101a can be received by responder 102' in message 504. A key exchange step 422 for responder 102'was depicted and described in connection with Figure 4d above. Responder 102' can then conduct a key derivation step 423 from Figure 4d above in order to determine or derived a secret shared key ke 327a.

A key derivation step 423 for responder 102' was depicted and described in connection with Figure 4d above. In summary, responder 102' can perform an elliptic curve point addition step 424 for the two values or points of L(a) 421a and L(b) 422b. As contemplated herein, the output of a ECDH key exchange function such as 401a or 401c can comprise a point value comprising an X coordinate or value and a Y coordinate or value, and when the output is described herein as a key or a value, the key or value can comprise either (i) the X value only for use in a key derivation function such as 402a, 402b, or 402c, or (ii) the X and Y values together for other calculations such as an ECC point addition step 424. The combination of the points L(a) 421a and L(b) for an ECC point addition step 424 can comprise the point or value L 412, as depicted and described in connection with a key derivation step 423 in Figure 4d. Key derivation step 423 can then input the value L 412, M 410 from message 503, and N 411 from step 314b into key derivation function 402c in order output symmetric ciphering key ke 327a.

Responder 102' can then perform an encryption step 322, where encryption step 322 was depicted and described in connection with Figure 4c above. Step 322 can also include responder 102' creating a value R-auth 404, where R-auth 404 can comprise a hash of the initiator nonce 317a from plaintext 403a in step 320, the second random number R-nonce 318b from a step 318 above, and public PKI keys. The value for R-auth 404 in a step 322 can comprise the value in point 3 of section 6.2.3 of DPP specification version 1.0. An encryption step 322 can comprise responder 102' (a) encrypting the value for R-auth 404 using the mutually derived key ke 327a from the key derivation step 423 above, where (b) the output of encryption step 322 can comprise ciphertext 324a.

Responder 102' can then conduct an encryption step 315b, where the details for an encryption step 315b are depicted and described in connection with Figure 4b above. Ciphertext 324a from an encryption step 322 in the paragraph above can also be encrypted again for an encryption step 315b. The symmetric ciphering key for encryption step 315b can comprise symmetric key k2 314c from a

Petitioner Samsung Ex-1004, 1484

key exchange step 314b. The output from an encryption step 315b can be ciphertext 315c, as depicted in Figure 4b above.

Responder 102' can conduct a step 324 before sending a message 123. For a step 324, responder 102' can record the derived responder ephemeral public key Pr 102p and responder ephemeral private key pr 102r and initiator bootstrap public key Bi 101a in an initiator table 102u from Figure 1f along with the identity of device 106 comprising potentially MAC.I 106n. In other words, an initiator table 102u can be updated to include information or data (i) received from network 107 (and DPP server 103) regarding devices 106 communication through network 107 via responder 102', and (ii) derived by responder 102' when performing key exchange steps 401a and 401c and key derivation steps 402a, 402b, and 402c. In other words, an initiator table 102u can include columns for the values of M 410, N 411, L 412, L(a) 421a, L(b) 422a, key k1 314a, key k2 314c, and key ke 327a for each device 106 with an identity such as MAC.I 106n. The recording of values for a device 106 or a plurality of devices 106 in an initiator table 102 can be equivalent to the values recorded for a server database 103a depicted and described in connection with Figure 3c above. In this manner, responder 102' can track keys and values for devices 106 since responder 102' may communicate with a plurality of different devices 106 and initiators 101 over time. In exemplary embodiments, responder 102' derives a new responder ephemeral PKI key pair for each unique DPP authentication request message 122 received. Data or values stored by responder 102' pertaining to derived encryption keys (e.g. k1, k2, and ke) and ephemeral PKI keys (e.g. Pr 102p and pr 102r) are preferably erased from an initiator table 102u upon successful completion of a configuration step 106a by device 106.

Responder 102' can then send initiator 101 a DPP authentication response message 123. In exemplary embodiments, message 123 is sent as a unicast message from (i) the responder MAC address MAC.R 102n (which received message 122) to (ii) the initiator MAC address MAC.I 106n (which sent message 122), over (iii) WiFi link 192. As depicted in Figure 5a, message 123 can include hash values of the bootstrap public keys H(Br 102a), H(Bi 101a), the responder ephemeral public key Pr 102p, and the ciphertext 315c. The ciphertext 315c can have at least encrypted values for (i) R-nonce 318b, (ii) I-nonce 317a, (iii) R-capabilities 318c, and (iv) ciphertext 324a. The ciphertext 315c can be encrypted with mutually derived symmetric encryption key k2 314c. Ciphertext 324a can be encrypted with the mutually derived symmetric key ke 327a and can include a plaintext value for R-auth 404. As described above, R-auth 404 can comprise a hash value over at least an initiator nonce 317a received and the responder nonce 318b.

Initiator 101 can receive DPP authentication response message 123 from responder 102' and conduct a series of steps in order to process the message and reply to the responder 102'. Initiator 101 can conduct a step 331, and a step 331 can comprise the series of steps 325, 319a, 320b, 327, 328, and 322b. The series of steps 331 for an initiator 101 to process a DPP authentication response 123 were depicted and described in connection with Figure 3a above. After completing the series of steps for a

Petitioner Samsung Ex-1004, 1485

step 331 in Figure 5a, initiator 101 can send responder 102' a DPP authentication confirm message 123a. The data and steps to determine and calculate an DPP authentication confirm message 123a for a step 331 can be compatible with section 6.2.4 of DPP specification version 1.0. Note that although the DPP authentication confirm message 123a can be equivalent to the DPP authentication confirm message from section 6.2.4 of DPP specification version 1.0, an important difference can be the operation of responder 102' in Figure 5a versus a responder as contemplated by DPP specification version 1.0. In the present disclosure as depicted in Figure 5a, responder 102' does not record the responder bootstrap private key br 102b.

Responder 102' can listen for either (i) a broadcast message with the hash 199e values for the bootstrap public keys 102aa and 101aa, or (ii) a unicast message to MAC.R 102n from MAC.I 106n for initiator 101 operating in device 106. Responder 102' can receive the DPP authentication confirm message 123a and process the message. DPP authentication confirm message 123a can include an encrypted value for initiator authentication value 404a as a ciphertext 324b, where ciphertext 324b is encrypted with derived key ke 327a in a step 322b within step 331. Upon receipt of DPP authentication confirm message 123a from device 106, responder 102' can first conduct a decryption step 328b, where decryption step 328b can use the same steps as a decryption step 328 as depicted in Figure 4c above, with a difference being decryption step 328b operates on the ciphertext 324b containing the encrypted initiator authentication value I-auth 404a instead of the R-auth 404 value. After a step 328b, responder 102' can read the plaintext initiator authentication value I-auth 404a.

In step 330, responder 102' can compare the decrypted value for I-auth 404a with a calculated value for I-auth 404a and if the two values match then the initiator 101 can be considered by responder 102' to be authenticated. The calculation of I-auth 404a for responder 102' in a step 330 can comprise a calculation of a secure hash value over the responder nonce 318b and bootstrap and ephemeral public keys, with an exemplary calculation for I-auth 404a shown in the sixth paragraph of page 50 of DPP specification version 1.0. Other calculations of an initiator authentication value over the responder nonce 318b are possible as well, without departing from the scope of the present disclosure.

After authentication step 330 is completed, in a step 505, responder 102' can determine the source of network credentials 109 or encrypted network credentials 109' for initiator 101. In a step 505, responder 102' can use the selected responder mode 306a as depicted and described in connection with Figure 3c. For embodiments where a network 107, an external network 108, or DPP server 103 can be the source of network credentials 109 or encrypted network credentials 109' (as specified in a responder mode 306a), then responder 102' can use the series of steps and messages in Figure 6 in order to (i) receive network credentials 109 or encrypted network credentials 109' from a DPP server 103 and (ii) send the credentials to initiator 101. For other embodiments where a responder 102' already records the network credentials 109 for initiator 101 and device 106 (such as responder 102' uses network credentials 109 to connect with network access point 105), then responder 102' can read

Petitioner Samsung Ex-1004, 1486

the locally stored credentials 109 and send the credentials 109 to initiator 101 in a DPP configuration response message 124 in Figure 6 below. Responder 102' can use a step 349 as depicted in Figure 3b in order to send locally stored credentials 109 to initiator 101, where responder 102' performs the actions instead of a DPP server 103. As depicted in Figure 5a, the collection of steps and messages beginning with responder 102' sending message 123 through step 505 can comprise a step 506. Step 506 can subsequently be used by a responder 102' for a different responder mode 306a depicted in Figure 5b below.

Figure 5b

Figure 5b is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device database, a DPP server, a responder proxy, and an initiator, in accordance with exemplary embodiments. System 500b can include a device database 104, DPP server 103, responder proxy 102', and initiator 101 operating in device 106. Before starting steps and message flows depicted in Figure 5b, the initiator, DPP server, and device database depicted for system 500b in Figure 5b may previously complete exemplary message flows and steps depicted for a step 301 in Figure 3a above. Responder proxy 102' can communicate with DPP server 103 via IP network 113, also as depicted in Figure 3a above.

A difference between system 500b in Figure 5b and system 500a in Figure 5a can be that (i) responder ephemeral PKI keys pr 102r and Pr 102p are derived and recorded by a responder 102' (instead of a DPP server 103 in Figure 3a), (ii) responder 102' sends responder ephemeral private key pr 102r to DPP server 103, and (iii) DPP server 103 derives key L 412 instead of responder 102' in Figure 5a. In exemplary embodiments depicted in Figure 5b the responder bootstrap private key br 102b can remain recorded in a DPP server 103 and not exposed to responder 102'. The exemplary embodiment depicted in Figure 5b can comprise a third mode 306a of operation for responder 102' and DPP server 103, as depicted and described in connection with Figure 3c above.

As depicted in Figure 5b, responder 102' and DPP server 103 can conduct a step 501, where a step 501 is depicted and described in connection with Figure 5a above. Responder 102' can use data received from initiator 101 in a message 122 from step 301 in order to conduct step 501. In summary, a step 501 in Figure 5b can comprise, (i) a step 502 for DPP server 103 to select a mode 306a for both responder 102' and DPP server 103, (ii) DPP server 103 conducting an ECDH key exchange to calculate the value or key M 410, (iii) responder 102' receiving the key M 410 from DPP server 103, (iv) responder 102' deriving the responder ephemeral PKI keys of pr 102r and Pr 102p, (v) responder 102' performing a key derivation step 402a with M 410 in order to calculate key k1 314a, (vi)

Petitioner Samsung Ex-1004, 1487

responder 102' decrypting ciphertext in message 122 using key k1 314a, and (vii) responder 102' conducting a key exchange step 320 to calculate key k2 314c.

Note that the selected responder mode 306a in a step 501 is not normally observable to an initiator 101, since the full and complete set of DPP 191 messages between initiator 101 and responder 102' can be transmitted and received between the two nodes in Figure 5b. In other words, the different distribution of PKI and shared secret keys in a server database 103a and responder 102' for the responder mode 306a used for Figure 5b would not normally be observable to an initiator 101. The recording of PKI and shared secret keys in a server database is depicted in a server database 103a in Figure 3c, where the value "-" can mean that the key is recorded and/or operated by responder 102'.

Upon completion of a step 501 in Figure 5b, responder 102' can send DPP server 103 a message 507 via secure session 302b established during step 301 from Figure 3a. Message 507 can include (i) an identity for device 106 and initiator 101, such as MAC.I 106n or hash value 101aa, and (ii) the derived responder ephemeral private key pr 102r. DPP server 103 can receive message 507 and use the received device identity to select the initiator bootstrap public key Bi 101a and responder bootstrap private key br 102b from a server database 103a. DPP server can use the received responder ephemeral private key pr 102r from message 507 and the selected data in order to conduct an ECDH key exchange step 401c as depicted and described in connection with Figure 4c above. The output of a step 401c can be key L 412.

After deriving key L 412, DPP server 103 can then send responder 102' a message 508, where message 508 can include an identity for device 106 such as MAC.I 106n (or hash value 101aa), the initiator bootstrap public key Bi 101a (from a server database 103a), and the derived key L 412 from a step 402c. Responder 102' can receive message 508 and record and process the data. Responder 102' can conduct a key derivation function 402c using (i) the received key L 412, (ii) the received key M 310 from a message 503 in a step 501, and (ii) the key N 311 derived by responder 102 in a step 314b in step 501. The output of a key derivation function 402c can be key ke 327a, as depicted and described for a key exchange step 402c in Figure 4c.

Responder 102' can then calculate a responder authentication value 404 and conduct an encryption step 322 using the plaintext 403c as depicted and described for a step 322 in Figure 4c. The output of an encryption step 322 can comprise ciphertext 324a. Responder 102' can then assemble plaintext 403b (which includes ciphertext 324a in the plaintext values for plaintext 403b) and conduct an encryption step 315b, where encryption step 315b is depicted and described in connection with Figure 4b above. Responder 102' can conduct a step 324 before sending a message 123. For a step 324, responder 102' can record the derived responder ephemeral public key Pr 102p and responder ephemeral private key pr 102r and initiator bootstrap public key Bi 101a in an initiator table 102u from Figure 1f along with the identity of device 106 comprising potentially MAC.I 106n.

Petitioner Samsung Ex-1004, 1488

Responder 102' and initiator 101 can then conduct a step 506, where the series of steps and messages for a step 506 were depicted and described for a responder 102' and initiator 101 in Figure 5a above. In summary for a step 506 depicted above in a Figure 5a, a responder 102' can send initiator 101 a DPP authentication response message 123 via WiFi link 192. Initiator 101 can process the DPP authentication response message 123 using a step 331, which was depicted and described in connection with Figure 3a above. Initiator 101 can then send a DPP authentication confirm message 123a and responder 102' can receive the message 123a via WiFi link 192. Responder 102' can decrypt ciphertext 324b from message 123a using a decryption step 328b in order to read plaintext initiator authentication value 404a. In step 330 of a step 506, responder 102' can compare the decrypted value for I-auth 404a with a calculated value for I-auth 404a and if the two values match then the initiator 101 can be considered by responder 102' to be authenticated.

After authentication step 330 in step 506 is completed, in a step 505 of a step 506, responder 102' can determine the source of network credentials 109 or encrypted network credentials 109' for initiator 101. For embodiments where a network 107, an external network 108, or DPP server 103 can be the source of network credentials 109 or encrypted network credentials 109' (as specified in a responder mode 306a), then responder 102' can use the series of steps and messages in Figure 6 in order to (i) receive network credentials 109 or encrypted network credentials 109' from a DPP server 103 and (ii) send the credentials to initiator 101. For other embodiments where a responder 102' already records the network credentials 109 for initiator 101 and device 106 (such as responder 102' uses network credentials 109 to connect with network access point 105), then responder 102' can read the locally stored credentials 109 and send the credentials 109 to initiator 101 in a DPP configuration response message 124 in Figure 6 below. Responder 102' can use a step 349 as depicted in Figure 3b in order to send locally stored credentials 109 to initiator 101, where responder 102' performs the actions instead of a DPP server 103.

Figure 6

Figure 6 is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a DPP server, a responder proxy, and an initiator, in accordance with exemplary embodiments. System 600 can include a DPP server 103, responder proxy 102', and initiator 101 operating in device 106. Before starting steps and message flows depicted in Figure 6, DPP server 103, responder 102', and initiator 101 can previously complete exemplary message flows and steps depicted for the nodes in either Figure 5a or Figure 5b. System 600 depicts the transfer of credentials 109 or 109' from responder 102' when responder 102' calculates and operates with symmetric ciphering key ke 327a. Although the source of credentials 109 or 109' is depicted as network 107 with DPP server 103 in Figure 6, the source of credentials 109 or 109' could be either

Petitioner Samsung Ex-1004, 1489

DPP server 103 or responder 102'. A difference between system 600 and system 300b in Figure 3b is that for system 300b, DPP server 103 calculates, records, and operates with key ke 327a for use with initiator 101 in device 106.

After transmitting or sending DPP authentication confirm message 123a in a step 506 as depicted and described in connection with Figure 5a and Figure 5b, initiator 101 can conduct a step 332. Initiator 101 can take on a roll of enrollee, such that initiator 101 can prepare to send a configuration attribute 330a. Configuration attribute 330a can be used to specify that initiator 101 can take the roll of enrollee and is expecting a set of network credentials 109 from responder 102', with associated data for responder 102' to prepare or format a set of network credentials 109.

In step 332, initiator 101 can also generate a random number, for an enrollee nonce E-nonce 332a. E-nonce 332a and other nonce values can be processed using nonce parameters 199g. Initiator 101 can then conduct an encryption step 322c, where encryption step 322c can comprise (i) an encryption step 322 using key ke 327a, and (ii) the plaintext to be encrypted can be the E-nonce 332a and the configuration attribute 330a. The output of an encryption step 322c can be a ciphertext 324c, which is also described for a step 322c in Figure 3b above. Initiator 101 can then send via WiFi link 192 to responder 102' a DPP Configuration Request message 123b, where the configuration request 123b can include the ciphertext 324c. The plaintext in ciphertext 324c can include E-nonce 332a and the configuration object 330. DPP Configuration Request message 123b can be processed by initiator 101 according to section 6.3.3 of DPP specification version 1.0.

Responder 102' can receive DPP Configuration Request message 123b via connection 192 and using DPP application 102g. Responder 102' can conduct a decryption step 328c using decryption step 328 depicted in Figure 4c above. A difference between decryption step 328 and decryption step 328c is that decryption step 328c uses the ciphertext 324c as input into the symmetric decryption algorithm 405b. Responder 102' can use the previously calculated key ke 327a in order to read the plaintext within ciphertext 324c. Responder 102' can record the plaintext E-nonce 332a and configuration attribute 330a.

Responder 102' can then conduct a step 601 in order to determine the source of credentials 109 for initiator 101 and device 106. For embodiments where responder 102' previously records a valid set of network credentials 109 for use with network access point 105 depicted in Figure 1a, then responder 102' could select the recorded network access credentials 109 for use as a configuration object for initiator 101 and device 106. In other words, for embodiments where responder 102' previously connects with network access point 105 (or responder 102' could operate within network access point 105), then responder 102' could record a valid set of network credentials 109 in order to successfully connect with network access point 105. For these embodiments, responder 102' could select the recorded network access credentials 109 for transfer to initiator 101 and used by initiator 101 and device 106 in order to also connect with access point 105 using the same set of network

Petitioner Samsung Ex-1004, 1490

credentials 109 recorded by responder 102'. Responder 102' can read the locally stored credentials 109 and send the credentials 109 to initiator 101 in a DPP configuration response message 124 in Figure 6. Responder 102' can use a step 349 as depicted in Figure 3b in order to send locally stored credentials 109 to initiator 101, where responder 102' performs the actions instead of a DPP server 103.

For other embodiments, such as responder 102' does not previously record a set of network credentials 109 for use by initiator 101 and device 106, then in a step 601 responder 102' could determine a query to DPP server 103 or network 107 is required for the network credentials 109 or encrypted network credentials 109' for initiator 101 and device 106. The determination of a source for network credentials (e.g. either responder 102' or DPP server 103 could also be specified in a responder mode 306a, such that a first responder mode 306a selects responder 102' as the source of credentials 109 and a second responder mode 306a selects DPP server 103 as the source of credentials 109 or encrypted credentials 109'). Other possibilities exist as well for a step 601 by responder 102' in Figure 6 to determine a source of credentials 109 for device 106 without departing from the scope of the present disclosure.

For embodiments where step 601 by responder 102' determines that network 107 or DPP server 103 is the source of network credentials 109 for device 106, responder 102' can then send DPP server 103 a message 602 through secure session 302b (depicted and described in Figure 3a), where message 602 include an identity for device 106 such as MAC.I 106n (or device identity 106i or hash value 101aa) and the plaintext configuration attribute 330a. DPP server 103 can receive message 602 and can conduct a step 337 in order to obtain network credentials 109 or asymmetrically encrypted credentials 109'. A step 337 for DPP server 103 was depicted and described in connection with Figure 3b above, and the same step can be used by DPP server 103 for system 600 in Figure 6 (for embodiments where step 601 by responder 102' determines that network 107 or DPP server 103 is the source of network credentials 109 for device 106).

Continuing with embodiments where step 601 by responder 102' determines that network 107 or DPP server 103 is the source of network credentials 109 for device 106, DPP server 103 can then send responder 102' a message 603, where message 603 includes (i) an identity for device 106 such as MAC.I 106n or ID.device 106i or hash value 101aa, and (ii) a set of network credentials 109 or asymmetrically encrypted network credentials 109'. Asymmetrically encrypted credentials 109' could be in message 603 if DPP server 103 obtained the credentials outside of or external to DPP server 103, such as an external network 108 (where external network 108 may prefer that only device 106 be able to read the plaintext network credentials 109 inside asymmetrically encrypted network credentials 109'). A difference between message 603 in Figure 6 and message 341 in Figure 3b can be (i) the message 603 is not encrypted with key ke 327a, and (ii) the message 603 may not include a digital signature 338c. Or, for other embodiments, then a system 600 could include the use of a digital

Petitioner Samsung Ex-1004, 1491

signature creation step 338 as depicted and described in connection with Figure 3b and Figure 3d and the message 603 and subsequent message 124 can include the digital signature 338c. Responder 102' can receive message 603 and process the data. Note that message 603, step 337, and message 603 can be omitted for embodiments where step 601 determines that responder 102' is the source of credentials 109 for device 106.

For embodiments where message 603 does not include a digital signature 338c as depicted in Figure 6, responder 102' can then record the network credentials 109 or asymmetrically encrypted network credentials 109' and conduct a signature creation step 338. Note that the source of credentials 109 can be either responder 102' or DPP server 103, depending on the source selected in step 601. In exemplary embodiments, asymmetrically encrypted network credentials 109' can be from DPP server 103 or potentially an external network 108. For a step 338 in Figure 6, responder 102' can conduct a signature creation step 338 as depicted and described in connection with Figure 3d above. The configuration object with network credentials 109 (or encrypted credentials 109') can be signed using the recorded responder ephemeral private key pr 102r, corresponding to public key Pr 102p sent in message 325 in Figure 3a above. The digital signature by responder 102' in Figure 3b can comprise a digital signature 338c. Or, in another embodiment the configuration object with network credentials 109 can be signed using the recorded responder bootstrap private key br 102b recorded by DPP server 103 and digital signature 338c included in a message 603, where initiator 101 could verify the signature 338c with the bootstrap public key Br 102a.

At step 332d, responder 102' can conduct an encryption step 332d using an encryption step 322 depicted and described in connection with Figure 4c above. Encryption step 332d can be performed using key ke 327a for device 106 and initiator 101 from an initiator table 102u. For encryption step 332d, the plaintext values of (i) the E-nonce 332a and (ii) the configuration object comprising network credentials 109 (or asymmetrically encrypted credentials 109') can be input into a symmetric encryption algorithm 405a in order to generate a ciphertext 324d. As depicted, the collection of steps and messages from 602 through 322d can comprise a step 604. In some exemplary embodiments using a fourth responder mode as described in Figure 3c where responder 102 operates as a responder 102* using conventional technology in Figure 1b and Figure 1c, then a responder 102* could use the step 604 in order to obtain the credentials 109 or 109' from a DPP server 103.

Responder 102' can then send initiator 101 a DPP configuration response message 124 via connection 192, where message 124 can include the ciphertext 324d, and the ciphertext 324d can include the E-nonce 332a, the signature 338c, and either network credentials 109 or asymmetrically encrypted network credentials 109'. If network credentials 109' are sent in ciphertext 324d, then the network credentials 109' can comprise a set of "double encrypted" network credentials of (i) asymmetrically encrypted network credentials 109' and (ii) then encrypted again in a ciphertext 324d using the key ke 327a.

Petitioner Samsung Ex-1004, 1492

Initiator 101 can receive message 124 from responder 102' with ciphertext 324d. Initiator 101 can conduct a decryption step 328d using decryption step 328 depicted in Figure 4c above. A difference between decryption step 328 and decryption step 328d is that decryption step 328d uses the ciphertext 324d as input into the symmetric decryption algorithm 405b. Initiator 101 can use the previously calculated key ke 327a from a step 331 in order to read the plaintext within ciphertext 324d. Initiator 101 can confirm that plaintext in ciphertext 324d includes the previously sent E-nonce 332a. Initiator 101 can extract a plaintext value for (i) network credentials 109 or (ii) asymmetrically encrypted network credentials 109' using symmetric decryption algorithm 405b and key ke 327a. In other words, with credentials 109' decryption step 328d removes the first layer of "double encryption" where the first layer was encrypted with key ke 327a, and the "plaintext" after decryption step 328d for credentials 109' can comprise the asymmetrically encrypted network credentials 109' (e.g. with the first layer of encryption with key ke 327a has been removed).

Initiator 101 can conduct a signature verification step 340 as depicted and described in connection with Figure 3d above in order to verify signature 338c received in message 124. Initiator 101 can use either (i) responder ephemeral bootstrap public key Pr 102p or (ii) responder bootstrap public key Br 102a in order to conduct a signature verification step 340 on digital signature 338c. The use of either (i) responder ephemeral bootstrap public key Pr 102p or (ii) responder bootstrap public key Br 102a can be specified in a set of parameters 199 or as part of asymmetric encryption parameters 199h or digital signature parameters 199i. Note that if responder bootstrap public key Br 102a is used to verify digital signature 338c in a step 340, then DPP server 103 would create digital signature 338c in a step 338 with responder bootstrap private key br 102b. If plaintext network credentials 109 are included in ciphertext 324f, then initiator 101 can read the plaintext values for credentials 109. If asymmetrically encrypted network credentials 109' are included in ciphertext 324d, then initiator 101 or device 106 operating initiator 101 can conduct an asymmetric decryption step 339 depicted above in Figure 3e in order to convert credentials 109' into plaintext credentials 109.

After reading the plaintext network credentials 109 and optionally verifying a signature using a recorded public key 102p or 102a, device 106 can complete the configuration step 106a by loading the network credentials 109 and begin operating a WiFi radio 106r with the network credentials 109. In other words, the WiFi data link 192 depicted in Figure 1a can be converted into a fully authenticated and encrypted WiFi connection using network credentials 109 and standards such as WPA2, WPA3, and also subsequent or related standards where WiFi nodes such as clients or access points use a compatible set of network credentials 109. For example, device 106 could begin scanning for SSID.network-AP 109a on a channel in Config.network-AP 109c (which can be included in credentials 109 or 109'), and then send access point 105 a probe request. In this manner, device 106 can obtain connectivity to network access point 105 through a configuration step 106a.

Petitioner Samsung Ex-1004, 1493

Figure 7

Figure 7 is a graphical illustration of an exemplary system, where an access point with an initiator communicates with a responder proxy using a hosted device provisioning protocol, in accordance with exemplary embodiments. The present disclosure contemplates that the systems and methods for using a responder 102' with a DPP server 103 can support a device provisioning protocol 191 in order to configure a network access point 105'. System 700 in Figure 7 can include an unconfigured network access point 105', a mobile phone 701, and a network 107. Unconfigured network access point 105' can include an initiator 101, where initiator 101 can have the same or equivalent functionality as an initiator 101 depicted and described above in Figure 1a, Figure 1b, Figure 1c, Figure 3a, Figure 3b, Figure 5a, Figure 5b, and Figure 6.

Initiator 101 can include a responder bootstrap public key Br 102a, an initiator bootstrap private key bi 101b, and an initiator bootstrap public key Bi 101a. PKI keys could be recorded in access point 105' before access point 105' is installed at an operating location, such as during manufacturing or distribution of unconfigured access point 105'. Initiator 101 operating in unconfigured access point 105' can have responder capabilities of an enrollee. Before installation of access point 105' at an operating location, initiator 101 could record a responder public key table 101ta (depicted and described in connection with Figure 1d and also Figure 1e) with a plurality of responder bootstrap public keys Br 102a for a plurality of different networks 107.

In exemplary embodiments, the responder public key table 101ta could be loaded into access point 105' during manufacturing or distribution of access point 105' or a configuration step for access point 105' before conducting DPP 191. As contemplated herein, an access point 105 or access point 105' can comprise a "WiFi router", where one interface is a WiFi radio similar to 106r or 102r and the other interface is a wired connection such as Ethernet, DSL, or a coaxial cable. For some exemplary embodiments where a radio 102x uses wireless technology other than WiFi, then access point 105' could comprise a "gnodeb" for a 5G network, and "enodeb" for a 4G LTE network, and other possibilities exist as well for the wireless technology used by a radio 102x in access point 105'.

Unconfigured access point 105' can power up and conduct a step 200 from Figure 2a and Figure 3a in order to periodically transmit DPP authentication request messages 122 using different recorded responder bootstrap public keys (combined with possibly different initiator bootstrap PKI keys) from an initiator key database 101t in order to "search" or "scan" for a compatible responder to communicate with. When mobile phone 701 is taken to the vicinity or within WiFi range of AP 105' and activates DPP application 102g and responder 102', then responder 102' can receive the DPP authentication request message 122 and begin processing the message if mobile phone 701 records a matching hash value 102aa over at least the hash over responder bootstrap public key Br 102a recorded in memory, such as in a responder network table 102t depicted in Figure 1f.

Petitioner Samsung Ex-1004, 1494

Mobile phone 701 can comprise a smart phone or a mobile handset, such as a mobile phone manufactured by Apple® or running an Android based operating system. Mobile phone 701 could also comprise another mobile and networked computing device, such as a laptop or a tablet. Mobile phone 701 could include a responder 102', where responder 102' could have the same or equivalent functionality as responder 102' depicted and described in connection with Figure 1a, Figure 1d, Figure 1f, Figure 3a, etc. In exemplary embodiments, responder 102' can operate as an application for mobile phone 701 downloaded from a repository of mobile phone applications, such as an "app store" or "play store", or possibly a private application. In other words, embodiments herein contemplate that a mobile phone 701 could comprise the physical entity operating or processing a responder 102' and the software or firmware instructions for operation of responder 102' could comprise a DPP application 102g. Other physical arrangements are possible as well, without departing from the scope of the present disclosure. Mobile phone 701 can have both a WiFi radio 102x and a WAN radio 102h. Mobile phone 701 can communicate with unconfigured access point 105' via the WiFi radio 102x and a network 107 and a DPP server 103 via the WAN radio 102h and an IP network 113. Responder 102' for mobile phone 701 can have capabilities of a configurator.

Network 107 can operate a DPP server 103, where a DPP server 103 in Figure 7 can have the functionality of a DPP server 103 depicted and described in connection with Figures 1a, Figure 3a, Figure 3b, Figure 3c, Figure 5a, Figure 5b, and Figure 6. Although not depicted in Figure 7, a network 107 can also operate other supporting servers and/or running processes depicted in Figure 1a, Figure 2a, etc., such as a device database 104. Network 107 could be connected to mobile phone 107 via the IP network 113 depicted in Figure 1a and using a secure session 302b from Figure 3a.

As depicted in Figure 7, mobile phone 107 using responder 102' could conduct the series of messages for a device provisioning protocol 191 with initiator 101 operating in unconfigured access point 105'. The series of messages for a device provisioning protocol 191 could comprise the messages 122, 123, 123a, 123b, and 124 as depicted and described in connection with Figures 1a, 3a, 3b, 5a, 5b, and 6 above. In order for responder 102' to send and receive the device provisioning protocol messages 191 depicted in Figure 7, responder 102' transmits and receives a series of messages with DPP server 103 in network 107, where the series of messages could comprise a hosted device provisioning protocol 702. For a first mode 306a, representing embodiments depicted in Figure 3a (where responder bootstrap and ephemeral secret keys remain in DPP server 103), the series of messages for a hosted device provisioning protocol 702 could comprise the messages between responder 102' and DPP server 103 that are depicted in Figure 3a and Figure 3b (e.g. messages 302b, 305, 309, 312b, etc.).

For a second mode 306a, representing embodiments depicted in Figure 5a where the responder bootstrap private key br 102b can remain in DPP server 103 (but DPP server 103 does not operate on responder ephemeral private key pr 102r), the series of messages for a hosted device provisioning

Petitioner Samsung Ex-1004, 1495

protocol 702 could comprise the messages between responder 102' and DPP server 103 that are depicted in Figure 5a (e.g. messages in step 301 and then 503, 504, etc.). For a third mode 306a, representing embodiments depicted in Figure 5b where (i) the responder bootstrap private key br 102b remains in DPP server 103 and (ii) responder 102' derives key pr 102r and sends the key to DPP server 103, the series of messages for a hosted device provisioning protocol 702 could comprise the messages between responder 102' and DPP server 103 that are depicted in Figure 5b (e.g. messages in step 301 and then message 507, 508, etc.).

Upon completion of a device provisioning protocol 191 and a hosted device provisioning protocol 702, unconfigured network access point 105' can conduct a configuration step 106a to load a set of network credentials 109 and begin operating with the credentials 109, as depicted in Figure 7. The configuration attribute in a message 123b can specify that access point 105' is to be configured as an access point (or with a value of "ap"). The configuration step 106a can convert the unconfigured network access point 105' into an operating network access point 105, such that WiFi clients (or "sta") can connect with network access point 105 using a compatible set of network credentials 109. As depicted in Figure 7, (i) a mobile phone 701 could also operate as a responder 102' with a roll of a configurator and (ii) the access point 105' operating with a roll of enrollee.

As depicted in Figure 7, mobile phone 701 could also conduct a device provisioning protocol 191 using a hosted device provisioning protocol 702 with DPP server 103 in order to transfer a set of compatible network credentials 109 to device 106, such that device 106 could connect with network access point 105. The compatible network credentials 109 for device 106 can specify that device 106 operates in a WiFi client mode with network access point 105. Other possibilities exist as well for the operation of a responder 102' to support the transfer of credentials to devices operating as initiators 106 without departing from the scope of the present disclosure.

Figure 8

Figure 8 is a graphical illustration of an exemplary system, where an access point operates as a responder proxy for a device with an initiator, in order transfer a set of network credentials to the device, in accordance with exemplary embodiments. The present disclosure contemplates embodiments where a network access point 105 can operate as a responder 102' for a device 106 operating as an initiator 101, in order for the network access point 105 to transfer network credentials 109 to the device 106. System 800 in Figure 8 can include an unconfigured device 106', a network access point 105, and a network 107.

Network access point 105 in a system 800 can operate with a set of network access credentials 109 and also operate as a responder 102'. Note that network access point 105 as a WiFi router can simultaneously support other devices routing WiFi data through network access point 105

Petitioner Samsung Ex-1004, 1496

concurrently with network access point 105 operating as a WiFi router. In other words, the present disclosure contemplates that when a radio such as WiFi radio 102x operates for a responder 102' listening for incoming DPP authentication request messages 122, the radio 102x can concurrently support regular WiFi communication according to standard protocols such as WPA2, WPA3, etc. Network access point 105 can connect with a network 107 via an IP network 113 and a secure session 302b as depicted and described in connection with Figure 3a. Network 107 can operate a DPP server 103, where the function and operation of a DPP server 103 is depicted and described above in connection with Figure 1a, Figure 2a, Figure 3a, etc. Although not depicted in Figure 8, network 107 could also operate a device database 104 and DPP server 103 could include a server database 103b. Network access point 105 could comprise a WiFi router and operate a WiFi radio and communicate with a plurality of devices 106 that have already performed a configuration step 106a or a manual configuration to connect the configured devices 106. For some exemplary embodiments where a radio 102x uses wireless technology other than WiFi, then access point 105' could comprise a "gnodeb" for a 5G network, and "enodeb" for a 4G LTE network, and other possibilities exist as well for the wireless technology used by a radio 102x in access point 105'.

Network access point 105 can have the internal electrical components equivalent to a responder 102' depicted and described above in connection with Figure 1f, including a CPU, RAM, nonvolatile memory, and a system bus. For the embodiment depicted in Figure 8, the responder 102' in network access point 105 could comprise software, firmware, or a running process within network access point 105. Responder 102' in network access point 105 could have the same or equivalent functionality as responder 102' depicted and described in connection with Figure 1a, Figure 1f, Figure 3a, Figure 3b, Figure 5a, Figure 5b, and Figure 6. Network access point 105 could operate with capabilities as a configurator 801, as depicted in Figure 8.

An unconfigured device 106' could be a device with a WiFi radio 106r and operate an initiator 101. The electrical components inside device 106' could be similar to the electrical components depicted for a responder 102' in Figure 1f, such as including a CPU, RAM, nonvolatile memory, an operating system, and a system bus. Initiator 101 in unconfigured device 106' could have the same or equivalent functionality as a initiator 101 depicted and described above in Figure 1a, Figure 1b, Figure 1c, Figure 1d, Figure 1e, Figure 3a, etc.. Unconfigured device 106' can have capabilities as an enrollee 802.

Unconfigured device 106' can power up and conduct a step 200 from Figure 2b and Figure 3a in order to periodically transmit DPP authentication request messages 122 using different recorded responder bootstrap public keys (combined with possibly different initiator bootstrap PKI keys) from an initiator key database 101t in Figure 1d in order to "search" or "scan" for a compatible responder to communicate with. When unconfigured device 106' is powered on or taken to the vicinity or within WiFi range of AP 105, then responder 102' in WiFi access point 105 can receive the DPP

Petitioner Samsung Ex-1004, 1497

authentication request message 122 from unconfigured device 106' and begin processing the message 122 if WiFi access point 105 records a matching hash value 102aa over the hash over responder bootstrap public key Br 102a recorded in memory, such as in a responder network table 102t depicted and described in connection with Figure 1f.

As depicted in Figure 8, network access point 105 with responder 102' could conduct the series of messages for a device provisioning protocol 191 with initiator 101 operating in unconfigured device 106'. The series of messages for a device provisioning protocol 191 could comprise the messages 122, 123, 123a, 123b, and 124 as depicted and described in connection with Figure 3a, Figure 3b, Figure 5a, Figure 5b, and Figure 6. In order for responder 102' in WiFi access point 105 to send and receive the device provisioning protocol messages 191 depicted in Figure 8, responder 102' (or AP 105 operating the responder 102') transmits and receives a series of messages with DPP server 103 in network 107, where the series of messages could comprise a hosted device provisioning protocol 702. The series of messages for a hosted device provisioning protocol 702 were depicted and described in connection with Figure 7 above. Note that the steps and messages for conducting a hosted device provisioning protocol 702 can depend on a responder mode 306a, where the responder mode 306a can be selected by a DPP server 103 in a step 306. The present disclosure also contemplates that a responder mode 306a for a hosted device provisioning protocol 702 can be selected by a responder 102'. For these embodiments, where mode 306a is selected by a responder 102', a selected responder mode 306a by the responder 102' could be transmitted from responder 102' to DPP server 103 in a message 305 depicted and described in connection with Figure 3a above.

The combined (i) device provision protocol 191 between network access point 105 and unconfigured device 106' and (ii) hosted device provisioning protocol 702 between network access point 105 and DPP server 103 can securely transfer a set of compatible network access credentials 109 to device 106'. Upon completion of a device provisioning protocol 191 and a hosted device provisioning protocol 702, unconfigured device 106' can conduct a configuration step 106a to (i) load a compatible set of network credentials 109 for network access point 105 and (ii) begin operating with the credentials 109, as depicted in Figure 8. The configuration step 106a can convert the unconfigured device 106' into a configured and operating device 106, such that device 106 can conduct a WiFi connection setup step 803 using the received set of compatible network credentials 109. The WiFi connection setup step 803 could use the series of steps and messages as specified by standards such as WPA2 and WPA3 with credentials 109, and other possibilities exist as well for a WiFi connection setup step 803 without departing from the scope of the present disclosure.

Conclusion

Various exemplary embodiments have been described above. Those skilled in the art will understand, however, that changes and modifications may be made to those examples without departing from the scope of the claims.

Petitioner Samsung Ex-1004, 1498

CLAIMS

1. A method for conducting a device provisioning protocol (DPP), the method performed by a responder, the method comprising:

a) establishing a secure session with a DPP server using at least a certificate for the DPP server;

b) receiving, from an initiator, a DPP authentication request, wherein the DPP authentication request includes at least an initiator ephemeral public key, a hash value for a responder bootstrap public key, and a first ciphertext;

c) sending, to the DPP server, at least the initiator ephemeral public key, the hash value, and the first ciphertext;

d) receiving, from the DPP server, a responder ephemeral public key and a second ciphertext, wherein the second ciphertext is encrypted using at least a responder ephemeral private key corresponding to the responder ephemeral public key, and wherein the second ciphertext includes at least a responder nonce and a responder authentication value;

e) sending, to the initiator, a DPP authentication response, wherein the DPP authentication response includes at least the received responder ephemeral public key and the second ciphertext; and

f) receiving, from the DPP server, a third ciphertext, wherein the third ciphertext includes network credentials for the initiator, and wherein the third ciphertext is encrypted using at least (i) the responder ephemeral private key, and (ii) the responder bootstrap private key.

2. The method of claim 1, wherein the DPP server derives the responder ephemeral private key and the responder ephemeral public key using a set of cryptographic parameters, and wherein the set of cryptographic parameters are selected using at least the hash value.

Petitioner Samsung Ex-1004, 1499

3. The method of claim 2, wherein the hash value, the set of cryptographic parameters, and the responder bootstrap private key are stored in a device database, and wherein the DPP server uses the hash value to query the device database for the set of cryptographic parameters and the responder bootstrap private key.

4. The method of any one of claims 1 to 3, wherein the responder sends the hash value for the responder bootstrap public key to the DPP server before step b), and wherein the responder stores the hash value before a step a).

5. The method of any one of claims 1 to 4, further comprising, after step a) and before step d), receiving a responder mode from the DPP server, wherein the responder mode signals the DPP server to derive the responder ephemeral private key.

6. The method of any one of claims 1 to 5, wherein the second ciphertext includes the responder authentication value as a fourth ciphertext, and wherein the fourth ciphertext is encrypted by the DPP server using at least (i) the responder ephemeral private key, and (ii) the responder bootstrap private key.

7. The method of claim 6, wherein the fourth ciphertext is encrypted using at least an initiator bootstrap public key, and wherein a network operating the DPP server stores the initiator bootstrap public key.

8. The method of any one of claims 1 to 7, wherein the first ciphertext is encrypted with a symmetric ciphering key, and wherein the symmetric ciphering key is derived by the initiator using the responder bootstrap public key and an initiator ephemeral private key corresponding to the received initiator ephemeral public key.

9. The method of any one of claims 1 to 8, wherein the DPP server conducts an elliptic curve Diffie-Hellman key exchange with the responder ephemeral private key and the initiator ephemeral public key in order to derive a symmetric ciphering key for the second ciphertext.

Petitioner Samsung Ex-1004, 1500

10. The method of any one of claims 1 to 9, wherein the DPP server conducts an elliptic curve Diffie-Hellman key exchange with the responder ephemeral private key, the responder bootstrap private key, and an initiator bootstrap public key in order to derive a symmetric ciphering key for the third ciphertext.

11. The method of claim 10, wherein the DPP authentication request includes a second hash value for the initiator bootstrap public key, wherein the responder sends to the DPP server the second hash value in a step c), and wherein the DPP server selects the initiator bootstrap public key for the initiator using the second hash value.

12. The method of any one of claims 1 to 11, wherein the responder comprises a responder proxy, and wherein the responder uses a WiFi radio with a responder configuration in order to receive the DPP authentication request message from the initiator.

13. The method of any one of claims 1 to 12, futher comprising after step e) and before step f), receiving, from the initiator, a DPP authentication confirm message with an encrypted initiator authentication value, wherein the responder sends to the DPP server at least the encrypted initiator authentication value.

14. The method of claim 13, wherein the DPP server decrypts the encrypted initiator authentication value using at least (i) the responder ephemeral private key, and (ii) the responder bootstrap private key, and wherein the DPP server compares the received initiator authentication value with a calculated initiator authentication value.

15. The method of any one of claims 1 to 14, further comprising:

recording a responder network table, wherein the responder network table includes at least the hash value; and

using the hash value and the responder network table to select the DPP server.

Petitioner Samsung Ex-1004, 1501

16. A method for conducting a device provisioning protocol (DPP), the method performed by a responder, the method comprising:

a) recording a set of cryptographic parameters and a responder network table;

b) receiving a DPP authentication request from an initiator, wherein the DPP authentication request includes at least an initiator ephemeral public key and a hash value for a responder bootstrap public key;

c) using the hash value and the responder network table to select a DPP server;

d) sending at least the initiator ephemeral public key and the hash value to the selected DPP server;

e) deriving (i) a responder ephemeral private key, (ii) a corresponding responder ephemeral public key, and (iii) a responder nonce using the set of cryptographic parameters;

f) receiving an initiator bootstrap public key and a secret key from the DPP server;

g) conducting an elliptic curve Diffie-Hellman (ECDH) key exchange using the derived responder ephemeral private key and the received initiator bootstrap public key in order to derive a first shared secret key;

h) conducting an elliptic curve point addition with the secret key and the derived first shared secret key in order to derive a second shared secret key; and

i) encrypting at least the responder nonce using the derived second shared secret key.

17. The method of claim 16, wherein the responder comprises a responder proxy, and wherein the responder establishes a secure session with the DPP server using at least a certificate for the responder.

18. The method of claim 16 or 17, further comprising (i) transmitting the encrypted responder nonce to the initiator in a DPP authentication response message, and (ii) receiving from the initiator a DPP authentication confirm message.

Petitioner Samsung Ex-1004, 1502

19. The method of claim 18, further comprising (i) receiving from the initiator a DPP configuration request after receiving the DPP authentication confirm message, and (ii) transmitting a DPP configuration response message, wherein the DPP configuration response message includes network credentials for the initiator.

20. The method of claim 19, wherein the responder stores the network credentials.

21. The method of claim 19, futher comprising (i) sending at least a configuration attribute from the DPP configuration request to the DPP server, and (ii) receiving the network credentials from the DPP server before sending the DPP configuration response message.

22. The method of claim 21, wherein the network credentials received from the DPP server comprise asymmetrically encrypted network credentials.

23. The method of any one of claims 16 to 22, wherein the responder conducts the ECDH key exchange and the elliptic curve point addition using the set of cryptographic parameters.

24. The method of any one of claims 16 to 23, wherein the responder network table includes at least (i) a plurality of hash values for a plurality of responder bootstrap public keys and (ii) a list of DPP servers.

25. The method of any one of claims 16 to 24, wherein the derived second shared secret key comprises a key "L", wherein the responder derives a symmetric ciphering key using a key derivation function and at least the key "L", and wherein the responder encrypts at least the responder nonce using the symmetric ciphering key.

26. The method of any one of claims 16 to 25, wherein the DPP authentication request includes a first ciphertext of at least an initiator nonce, and wherein the first ciphertext is encrypted with a symmetric ciphering key, and wherein the symmetric ciphering key is derived by the initiator using the responder bootstrap public key and an initiator ephemeral private key corresponding to the initiator ephemeral public key.

Petitioner Samsung Ex-1004, 1503

27. A method for conducting a device provisioning protocol (DPP), the method performed by a responder, the method comprising:

a) recording a set of cryptographic parameters and a responder network table, wherein the responder network table includes at least (i) a plurality of hash values for a plurality of responder bootstrap public keys and (ii) a list of DPP servers;

b) receiving, from an initiator, a DPP authentication request, wherein the DPP authentication request includes at least an initiator ephemeral public key, a hash value for a responder bootstrap public key, and a first ciphertext;

c) using the hash value and the responder network table to select a DPP server;

d) sending the initiator ephemeral public key, the hash value for a responder bootstrap public key, and the first ciphertext to the selected DPP server;

e) receiving, from the DPP server, a responder ephemeral public key and a second ciphertext, wherein the second ciphertext is encrypted using at least a responder ephemeral private key corresponding to the responder ephemeral public key, and wherein the second ciphertext includes at least a responder nonce and a responder authentication value;

f) sending, to the initiator, a DPP authentication response, wherein the DPP authentication response includes at least the received responder ephemeral public key and the second ciphertext; and

g) receiving, from the DPP server, a third ciphertext, wherein the third ciphertext includes network credentials for the initiator, and wherein the third ciphertext is encrypted using at least (i) the responder ephemeral private key, and (ii) the responder bootstrap private key.

28. The method of claim 27, wherein the responder comprises a responder proxy, and wherein the responder establishes a secure session with the DPP server using at least a certificate for the DPP server.

Petitioner Samsung Ex-1004, 1504

29. The method of claim 27 or 28, wherein the DPP server derives the responder ephemeral private key and the responder ephemeral public key using the set of cryptographic parameters, and wherein the set of cryptographic parameters are selected by the DPP server using at least the hash value.

30. The method of claim 29, wherein the hash value, the set of cryptographic parameters, and the responder bootstrap private key are stored in a device database, and wherein the DPP server uses the hash value to query the device database for the set of cryptographic parameters and the responder bootstrap private key.

31. The method of any one of claims 27 to 30, wherein the responder sends the hash value for the responder bootstrap public key to the DPP server before step b), and wherein the responder network table includes the hash value.

32. The method of any one of claims 27 to 31, further comprising, after step a) and before step e), receiving a responder mode from the DPP server, wherein the responder mode signals the DPP server to derive the responder ephemeral private key.

33. The method of any one of claims 27 to 32, wherein the first ciphertext is encrypted with a symmetric ciphering key, and wherein the symmetric ciphering key is derived by the initiator using the responder bootstrap public key and an initiator ephemeral private key corresponding to the initiator ephemeral public key.

34. The method of any one of claims 27 to 33, wherein the DPP server derives the responder ephemeral private key, and wherein the DPP server conducts an elliptic curve Diffie-Hellman key exchange with the responder ephemeral private key and the initiator ephemeral public key in order to derive a symmetric ciphering key for the second ciphertext.

35. The method of any one of claims 27 to 34, wherein the second ciphertext includes the responder authentication value as a fourth ciphertext, and wherein the fourth ciphertext is encrypted by the DPP server using at least (i) the responder ephemeral private key, and (ii) the responder bootstrap private key.

Petitioner Samsung Ex-1004, 1505

36. The method of claim 35, wherein the fourth ciphertext is encrypted using at least an initiator bootstrap public key, and wherein a network operating the DPP server stores the initiator bootstrap public key.

37. The method of any one of claims 27 to 36, wherein the DPP server conducts an elliptic curve Diffie-Hellman key exchange with the responder ephemeral private key, the responder bootstrap private key, and an initiator bootstrap public key in order to derive a symmetric ciphering key for the third ciphertext.

38. The method of claim 37, wherein the DPP authentication request includes a second hash value for an initiator bootstrap public key, wherein the responder sends to the DPP server the second hash value in a step d), and wherein the DPP server selects the initiator bootstrap public key for the initiator using the second hash value.

39. The method of any one of claims 27 to 38, wherein the responder comprises a responder proxy, and wherein the responder uses a WiFi radio with a responder configuration in order to receive the DPP authentication request message from the initiator.

40. The method of any one of claims 27 to 39, futher comprising after step f) and before step g), receiving, from the initiator, a DPP authentication confirm message with an encrypted initiator authentication value, wherein the responder sends to the DPP server at least the encrypted initiator authentication value.

41. The method of claim 40, wherein the DPP server decrypts the encrypted initiator authentication value using at least (i) the responder ephemeral private key, and (ii) the responder bootstrap private key, and wherein the DPP server compares the received initiator authentication value with a calculated initiator authentication value.

Petitioner Samsung Ex-1004, 1506

42. A system for supporting a device provisioning protocol (DPP), the system comprising:

a wireless device for operating a DPP application, wherein the DPP application comprises at least, in part, a responder, for establishing a secure session with a DPP server using at least a certificate for the DPP server, for storing at least one hash value for a responder bootstrap public key, for using a responder configuration to communicate over a WiFi link with an initiator, for using the WiFi link to receive a DPP authentication request message with at least (i) an initiator ephemeral public key and (ii) the hash value, for receiving an encrypted set of network credentials from the DPP server for the initiator, and for sending over the WiFi link the encrypted set of network credentials to the initiator in a DPP configuration response message;

a database for recording (i) a responder bootstrap private key and (ii) a set of cryptographic parameters for the responder bootstrap private key;

a DPP server for receiving the initiator ephemeral public key and the hash value from the wireless device, for querying the database using the hash value in order to receive the responder bootstrap private key, and for conducting an elliptic curve Diffie-Hellman (ECDH) key exchange using at least (i) the received initiator ephemeral public key, (ii) the received responder bootstrap private key, and (iii) the set of cryptographic parameters in order to derive a shared secret;

a processor in the DPP server for deriving a symmetric ciphering key using at least the derived shared secret, and for encrypting the set of network credentials using the symmetric ciphering key and the set of cryptographic parameters; and

a LAN interface in the DPP server for using the secure connection (i) to send the hash value to the wireless device before the wireless device receives the DPP authentication request and (ii) to send the encrypted set of network credentials to the initiator.

43. The system of claim 42, further comprising the DPP server for selecting a responder mode for the wireless device, and for sending the responder mode to the wireless device.

Petitioner Samsung Ex-1004, 1507

44. The system of claim 42 or 43, wherein the wireless device operates as a responder proxy, and wherein the wireless device does not receive and does not operate with the responder bootstrap private key.

45. The system of any one of claims 42 to 44, further comprising a network access point for authenticating with the set of network access credentials, and for providing connectivity with an IP network to the initiator after authenticating with the set of network access credentials.

46. The system of any one of claims 42 to 45, further comprising the DPP server for deriving (i) a responder ephemeral public key for a DPP authentication response message after receiving the hash value and (ii) a ciphertext of a responder authentication value, and wherein the ciphertext is encrypted using at least the derived shared secret, wherein the DPP server sends the responder ephemeral public key and the ciphertext to the wireless device before sending the encrypted set of network credentials.

47. The system of any one of claims 42 to 46, wherein the DPP server uses the responder bootstrap private key with a plurality of different initiator bootstrap public keys, wherein the plurality of different initiator bootstrap public keys are associated with a plurality of different initiators.

48. A wireless device for supporting a device provisioning protocol (DPP), the wireless device comprising:

a WAN radio for establishing a secure session with a server, and for receiving from the server (i) a responder configuration, (ii) a first hash value of a responder bootstrap public key, and for sending to the server a first ciphertext for a DPP authentication request message from an initiator;

a WiFi radio for operating with the responder configuration, for receiving from the initiator the DPP authentication request message with (i) the first ciphertext, (ii) the first hash value, and (iii) an initiator ephemeral public key, and for transmitting to the initiator a DPP authentication response message with a second ciphertext and a responder ephemeral public key;

Petitioner Samsung Ex-1004, 1508

a nonvolatile memory for storing (i) a responder network table with at least the first hash value and an identity for the server, and (ii) a DPP application for conducting the DPP over the WiFi radio using the responder configuration;

a processor for selecting the identity from the responder network table using the first hash value, for validating the initiator ephemeral public key, for encrypting the initiator ephemeral public key and the first ciphertext for the secure session, and for storing in volatile memory a third ciphertext from the server, wherein the third ciphertext includes data for a DPP authentication confirm message; and

a data bus for electrically connecting the WAN radio, WiFi radio, nonvolatile memory, and processor, and for transferring from the WAN radio to the WiFi radio a fourth ciphertext of network credentials for the initiator.

49. The wireless device of claim 48, wherein the wireless device (i) does not store and (ii) does not receive a responder bootstrap private key corresponding to the responder bootstrap public key.

50. The wireless device of claim 48 or 49, wherein the wireless device comprises a responder for the DPP, and wherein the DPP authentication request message is received by the wireless device as a broadcast message over a WiFi link.

51. The wireles device of any one of claims 48 to 50, wherein the server records a responder bootstrap private key corresponding to the responder bootstrap public key, wherein the WiFi radio receives a second hash value of an initiator bootstrap public key with the DPP authentication request, and wherein the WAN radio sends the second hash value to the server.

52. The wireless device of any one of claims 48 to 51, wherein the third ciphertext is encrypted with a symmetric ciphering key, and wherein the symmetric ciphering key is derived by the server using an elliptic curve Diffie Hellman (ECDH) key exchange and at least (i) the responder bootstrap private key, and (ii) the initiator ephemeral public key.

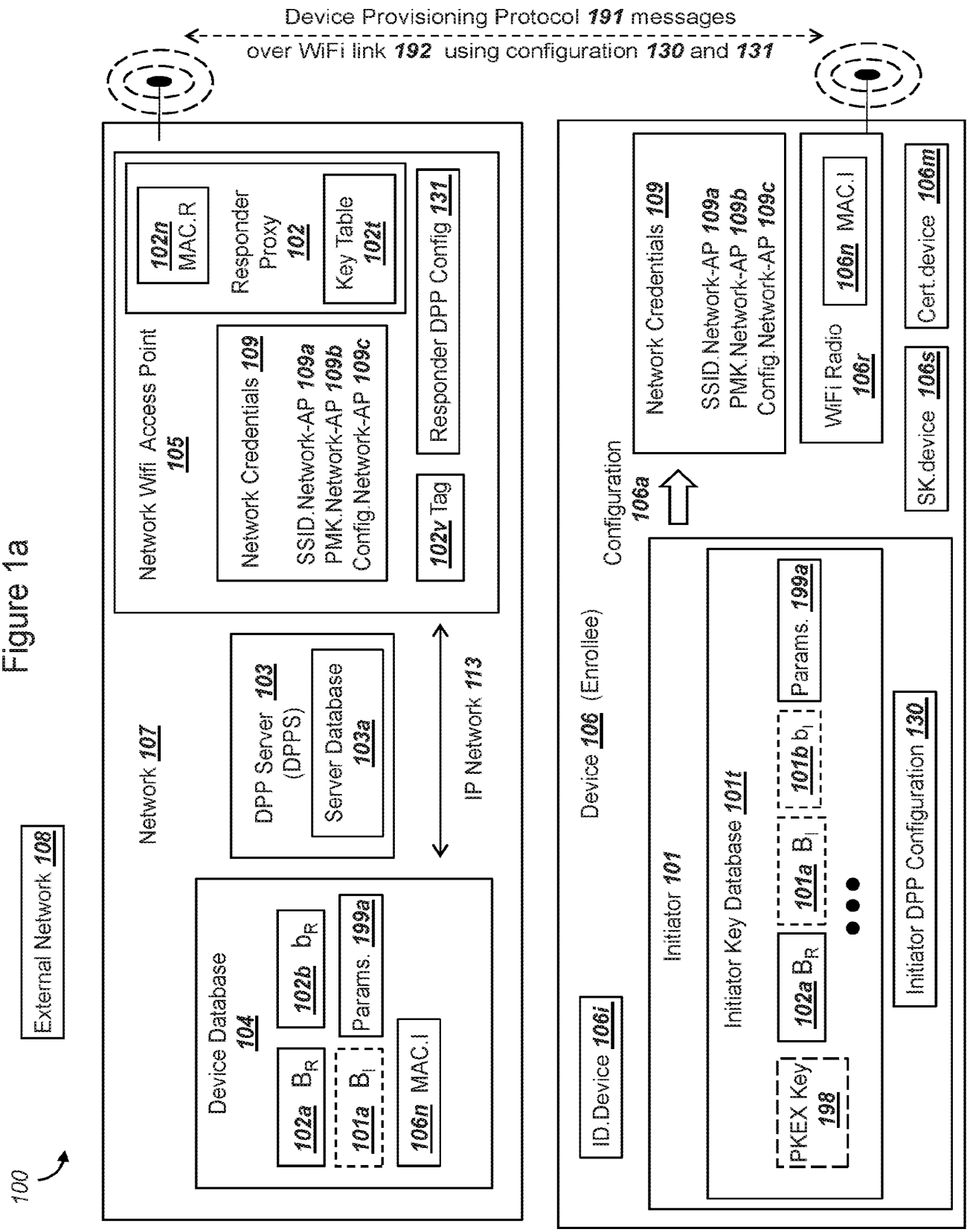
Petitioner Samsung Ex-1004, 1509

53. The wireless device of claim 52, wherein the symmetric ciphering key is derived using an initiator bootstrap public key.

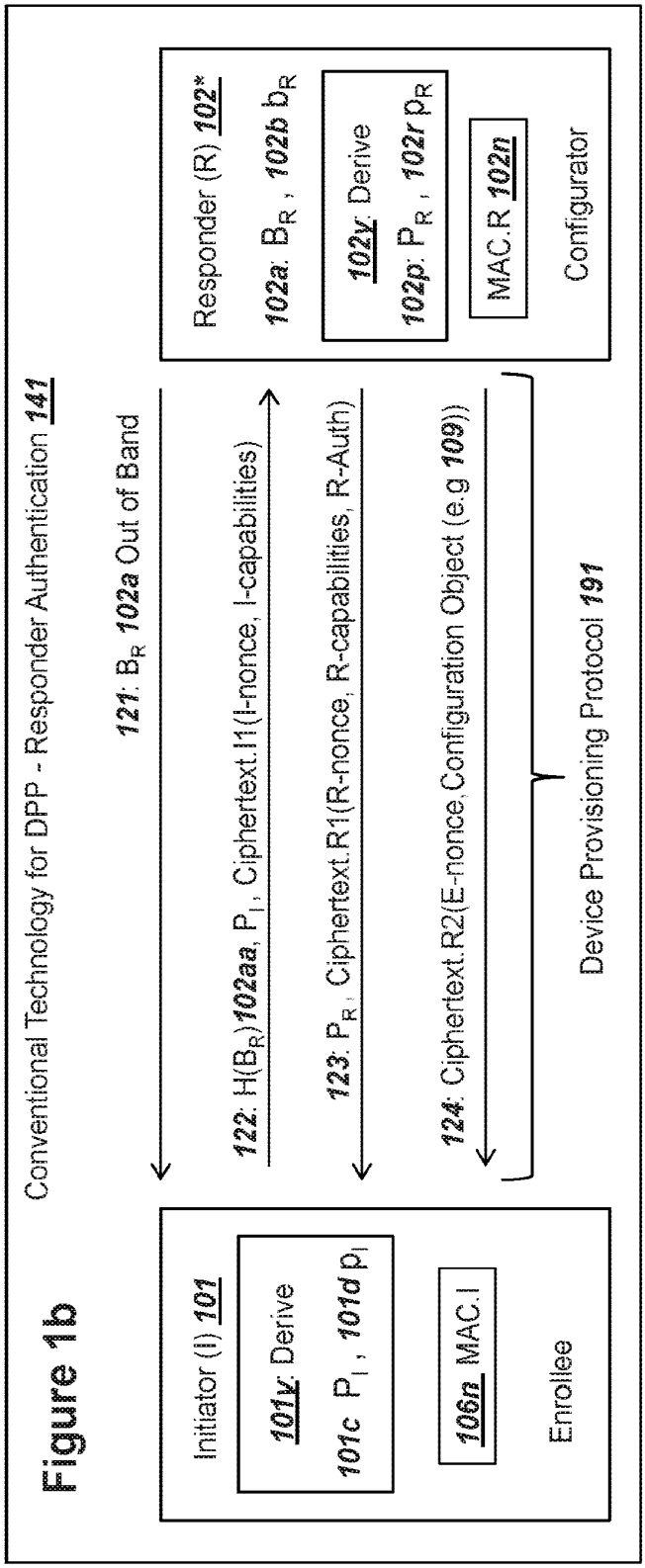
54. The wireless device of any one of claims 48 to 53, wherein the WiFi radio transmits the DPP authentication confirm message that includes the third ciphertext, wherein the WiFi radio receives a DPP configuration request message after the WiFi radio transmits the DPP authentication confirm message, wherein the WiFi radio transmits a DPP configuration response message with the fourth ciphertext to the initiator after the WiFi radio receives the DPP configuration request message.

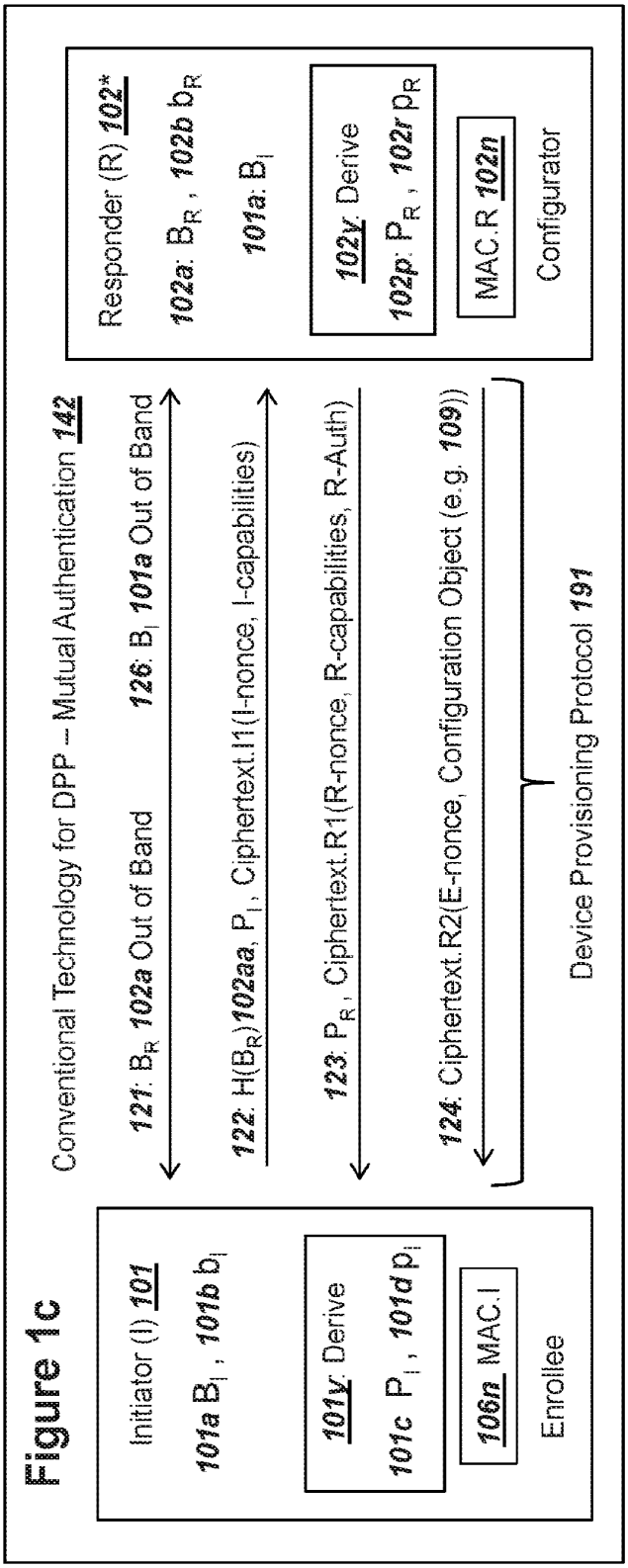
Petitioner Samsung Ex-1004, 1510

Figure 1a



Petitioner Samsung Ex-1004, 1511





Petitioner Samsung Ex-1004, 1512

Figure 1d

Initiator Key Database **1011**

Responder Public Key Table **101 ta**

Br Key ID 101z	Responder Network 107a	Br 102a	H(Br) 102aa	Params 199
1	Netgear	102a - 1	H(102a - 1)	A
2	Linksys	102a - 2	H(102a - 2)	A
3	Google	102a - 3	H(102a - 3)	A
4	Apple	102a - 4	H(102a - 4)	A
5	ATT	102a - 5	H(102a - 5)	A
6	Comcast	102a - 6	H(102a - 6)	C
7	Network 107 (Param A)	102a - 7	H(102a - 7)	A
8	Network 107 (Param B)	102a - 8	H(102a - 8)	B

Initiator Key Table **101 tb**

Bi Key ID 101w	Params 199a	Bi 101a	H(Bi) 101aa	bi 101b
1	A	101a-1	H(101a-1)	101b-1
2	B	101a-2	H(101a-2)	101b-2
3	C	101a-3	H(101a-3)	101b-3

DPP Configuration **130**

Device Supported Config 130z	Global Operating Class 130a	Channel List 130b
1	83 (2.4 Ghz)	1 - 9
2	122 (5 Ghz)	100, 108, 116
3	60 Ghz	1, 2

Figure 1e

Device 106 (1)

Initiator 101 (1)

Br ID 102z	Responder Network 107a	Br 102a	
1	Netgear	)M[a2D.*2Br6xZf[n}lAM(HF*c2GW)ouc2tQ_[qW	Shared Responder Bootstrap Public key 102z
2	Linksys	#w-?dRL7fz=UpUP^515;}gN1d9hTMcp.?!5tag2M	
3	Network 107	+.K8Ka%pq0WXPNQtH4jI.LbMfKk?74?)d)rY^^#;	

Bi ID 101w	Params 199a	Bi 101a	Bi 101b
1	A (p256)	.:3$.YGqL/[?=^-Or.GC&J3?;wdRaE?Aa[ZEUh!5	%LR!rFb}Lm9C.^Dp>+FaZzH=T[okv/p9g/4U*W.m
2	B (p 224)	!wED/$)a!ugv!<b$l^RWcX5-{I^mXwjmN-/	#Z;q0.S1#v!zgHG?!eQ]xU3yds:=-r@F(U@

Device 106 (2)

Initiator 101 (2)

Br ID 102z	Responder Network 107a	Br 102a	
1	Netgear	)M[a2D.*2Br6xZf[n}lAM(HF*c2GW)ouc2tQ_[qW	Shared Responder Bootstrap Public key 102z
2	Linksys	#w-?dRL7fz=UpUP^515;}gN1d9hTMcp.?!5tag2M	
3	Network 107	&^GB?%c.N^hztm69];wIfTjNvWSDqs^i]sxHr&]9	←Unique Responder Bootstrap Public Keys 102vv

Bi ID 101w	Params 199a	Bi 101a	Bi 101b
1	A (p256)	!}Cz:{RdlKu!q<ZE9tZ%6oe/r=#fW&-f.>Cw#Yr	)q}FE3nLL)&p]M4y7vC{:SrSPbUMAwDr:sO]$xI8B
2	B (p 224)	1B<B[]BDOfySr!?Nc2lf_+bau]!m=hj16FN	)+v]<avf!cphLvi0FemQT*%+64BK2vZC?r+

Device 106 (3)

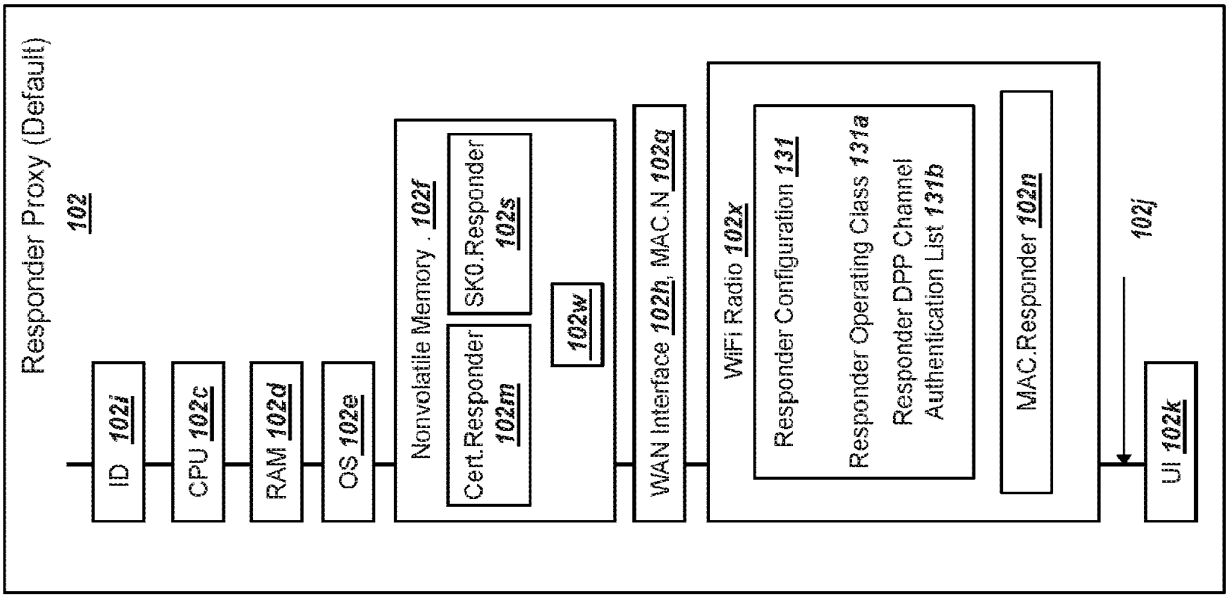
Initiator 101 (3)

Br ID 102z	Responder Network 107a	Br 102a	
1	Netgear	)M[a2D.*2Br6xZf[n}lAM(HF*c2GW)ouc2tQ_[qW	←Unique Responder Bootstrap Public Keys 102vv
2	Linksys	#w-?dRL7fz=UpUP^515;}gN1d9hTMcp.?!5tag2M	
3	Network 107	&LIS3R..Vn?o2v@yqq]eC+Tf0xDE5r%)o8(FbO(<	

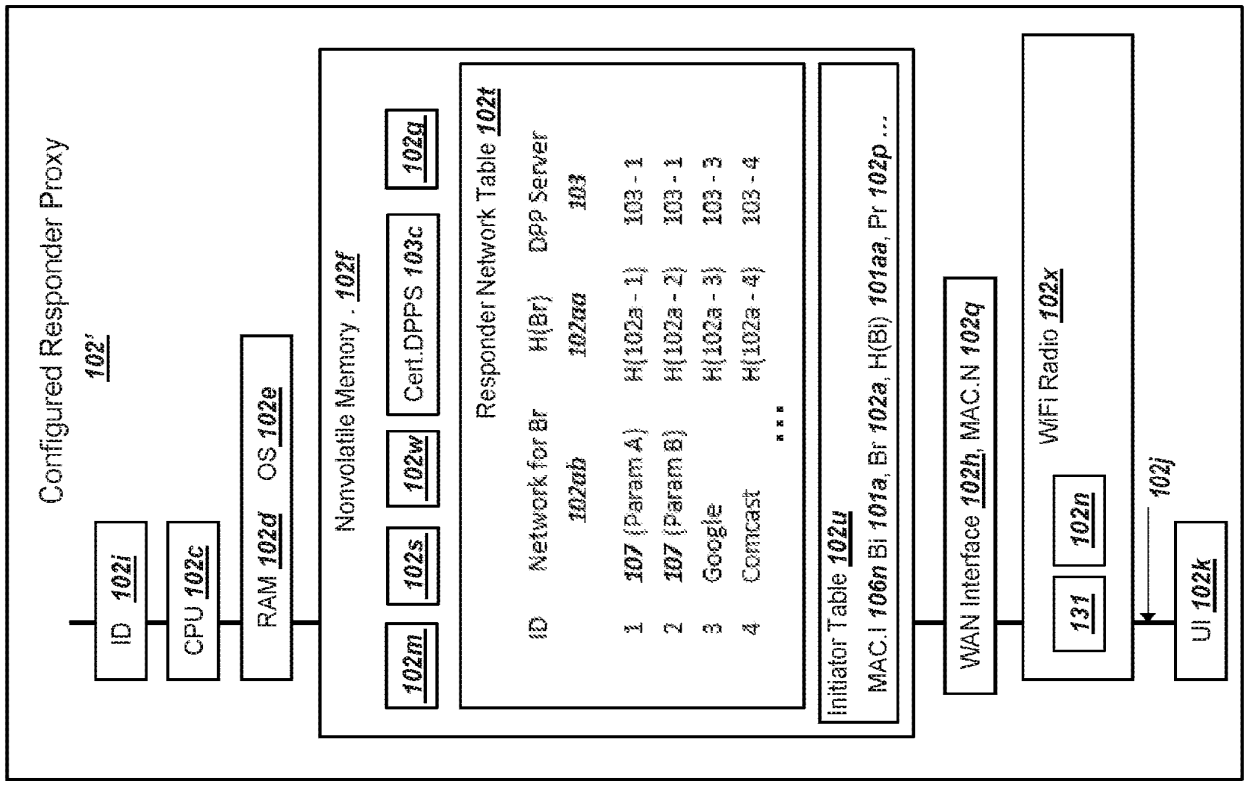
Bi ID 101w	Params 199a	Bi 101a	Bi 101b
1	A (p256)	&kPXOd7Apzkh04d^>!^k;1Q(ZP.Nb+kv45faq-f%	)5Jn&^K7@#>yVT}!rGJS|QDH4X6xo<kBCDDHHkI^
2	B (p 224)	!NfbvO9U=>!zM7*rd^BcDtv1rUI/!2:v?>q	*.j>fDtvy.?tp}K.z6?myP1b3=#]z40tdCt

Petitioner Samsung Ex-1004, 1514

Figure 1f



Responder Configuration 127



Petitioner Samsung Ex-1004, 1515

Figure 2a

Device Database **104**

Device Index	Device Identity	MAC Initiator	Responder Identity	Responder Mode	Cryptog. Params	PKEK Key	Responder Bootstrap Public Key (Br)	Responder Bootstrap Private Key (br)	Initiator Bootstrap Public Key (Bi)	Key Table **104k** H(Br)	H(Bi)
	106i	***106n***	***102i***	***306a***	***199***	***198***	***102a***	***102b***	***101a***	***102aa***	***101aa***
1	20001	38607720A	6cf049080	Fig. 3	A	NA	Br-1 (Shared	br-1	Bi-1	H(Br-1)	H(Bi-1)
2	20002	386077209	6cf049080	Fig. 5a	A	NA	Br-1 Keys	br-1	Bi-1	H(Br-1)	H(Bi-1)
3a	20003	386077400	6cf049080	102"	A	X3	Br-1 ***102z***)	br-1	NA	H(Br-1)	-
3b	20003	386077400	6cf049080	102"	B	X3	Br-3(B)	br-3(B)	NA	H(Br-3(B))	-
4	20004	3860773cc	1c6f654d2	Fig. 3	C	X4	Br-4(C)	br-4(C)	Bi-4(C)	H(Br-4(C))	H(Bi-4(C))
5a	20005	386077a09	1c6f654d2	Fig. 5b	A	NA	Br-5(A)	br-5(A)	Bi-5(A)	H(Br-5(A))	H(Bi-5(A))
5d	20006	386077a09	1c6f654d2	Fig. 5	D	NA	Br-5(D)	br-5(D)	Bi-5(D)	H(Br-5(D))	H(Bi-5(D))

●●●

Cryptographic Parameters ***199***

Set	Key Length	Curve	Hash	AES-SIV Key Length	Nonce Length	Asymmetric Encryption Parameters
199a	***199b***	***199c***	***199e***	***199f***	***199g***	***199h***
A	256	p256	SHA-256	256	128	Elgamal, p, g
B	384	p384	SHA-384	384	192	Elgamal, p, g
C	512	secp521r1	SHA-512	512	256	RSA
D	384	Curve25519	SHA-3	384	192	Elgamal, p, g

●●●

Petitioner Samsung Ex-1004, 1516

Figure 2b

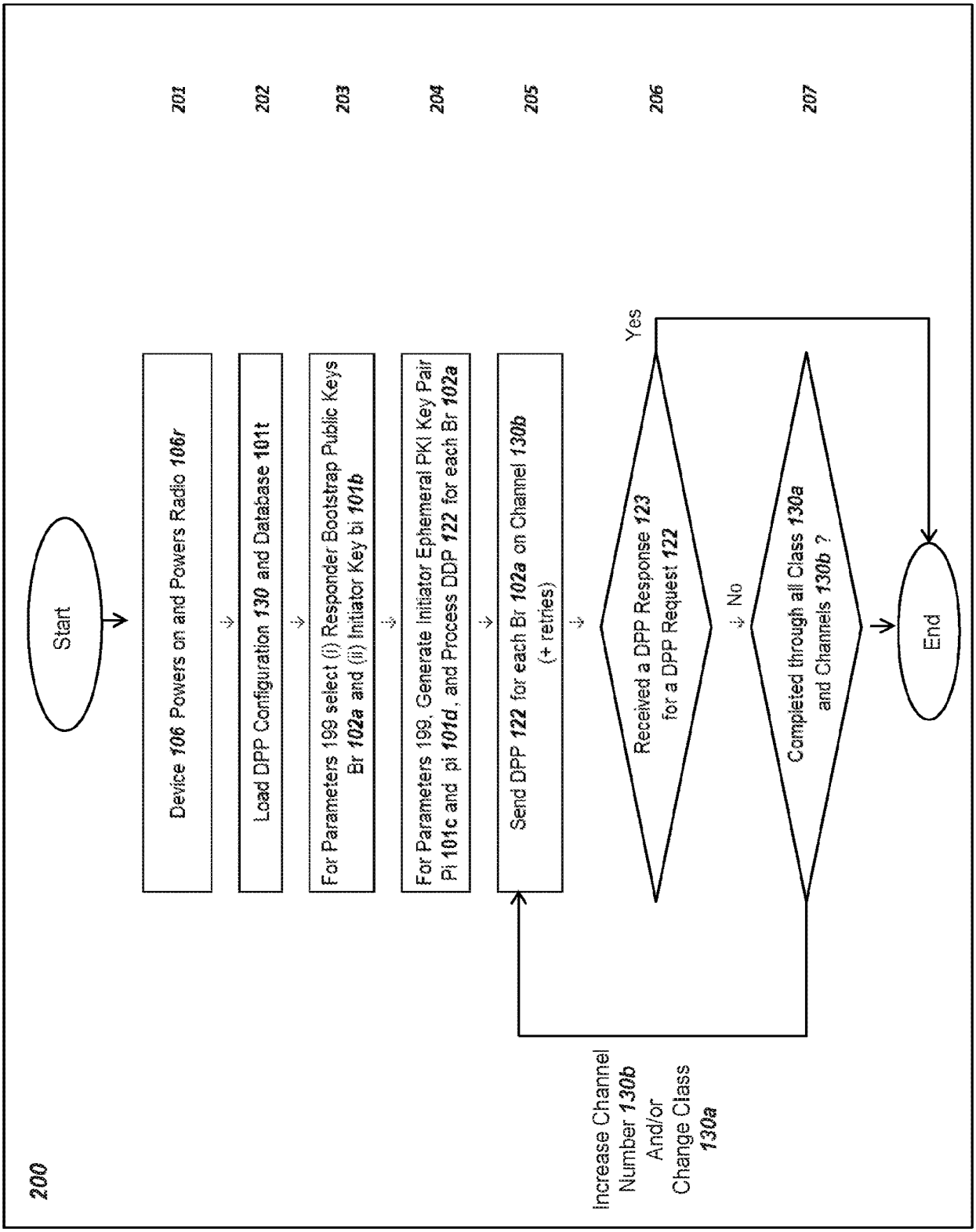
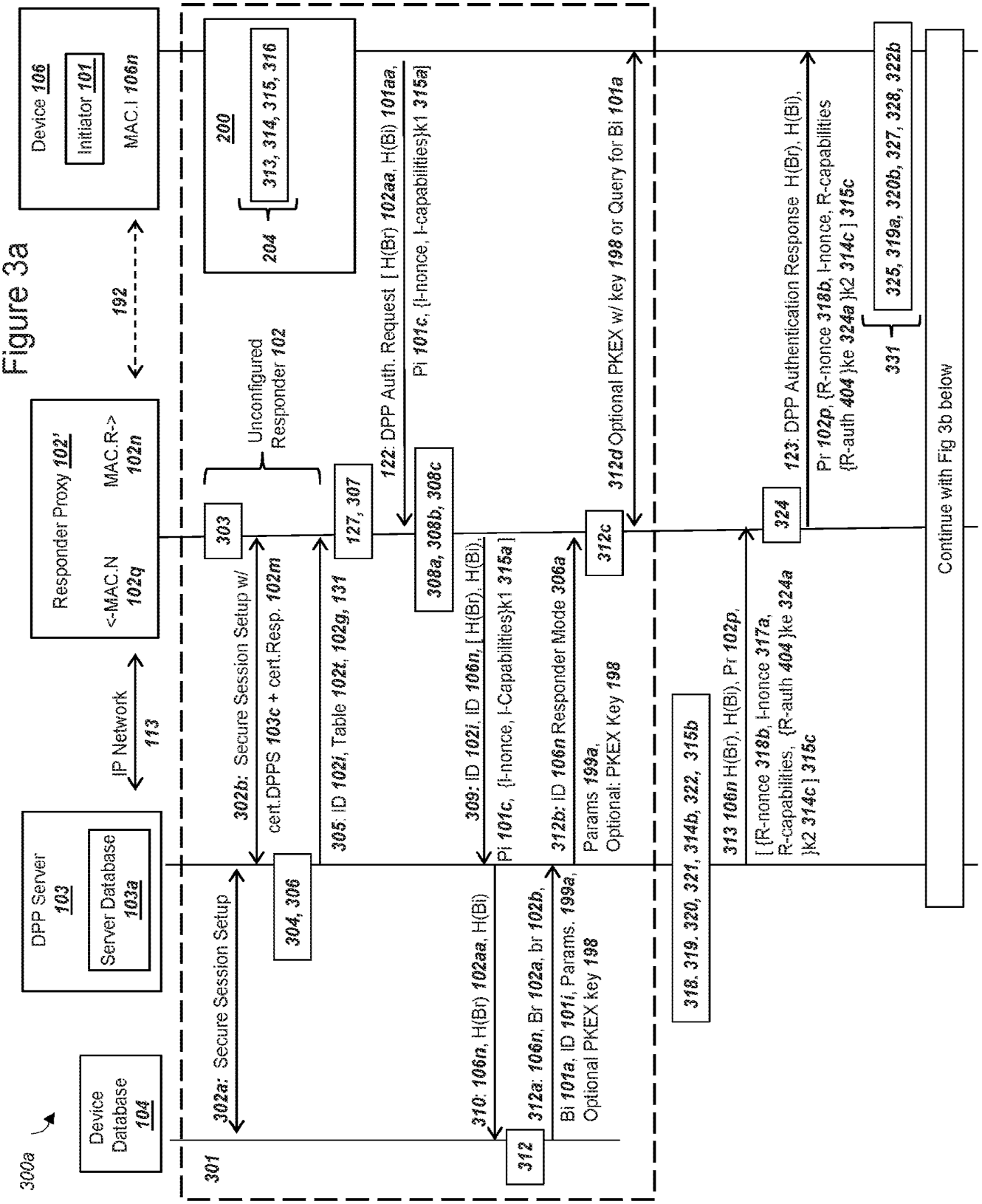
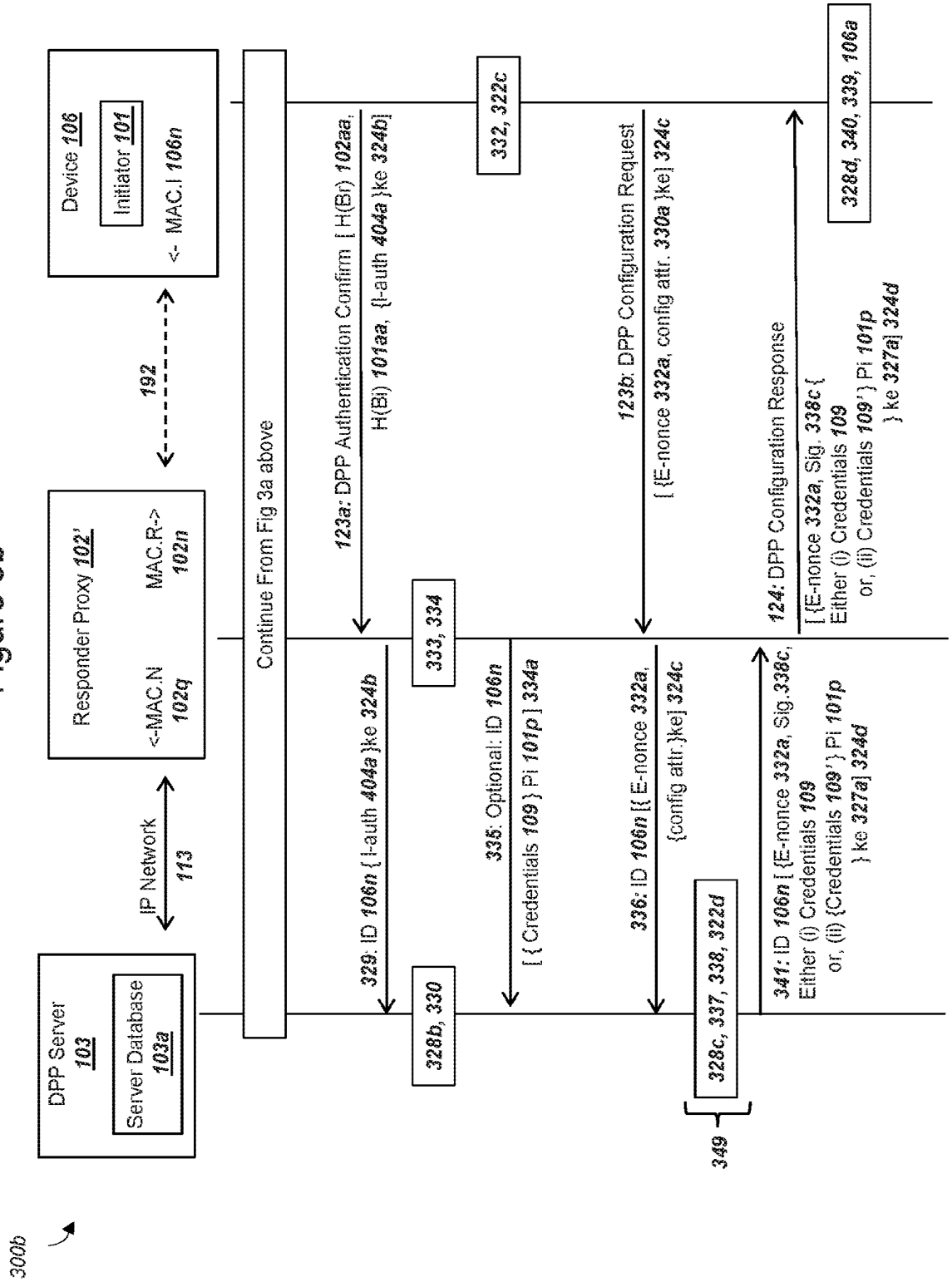

200
Start
201
Device 106 Powers on and Powers Radio 106r
202
Load DPP Configuration 130 and Database 101t
203
For Parameters 199 select (i) Responder Bootstrap Public Keys Br 102a and (ii) Initiator Key bi 101b
204
For Parameters 199, Generate Initiator Ephemeral PKI Key Pair Pi 101c and pi 101d, and Process DDP 122 for each Br 102a
205
Send DPP 122 for each Br 102a on Channel 130b (+ retries)
206
Received a DPP Response 123 for a DPP Request 122
Yes
No
207
Completed through all Class 130a and Channels 130b ?
End
Increase Channel Number 130b And/or Change Class 130a

Figure 3a



Petitioner Samsung Ex-1004, 1518

Figure 3b



Petitioner Samsung Ex-1004, 1519

Figure 3c

Server Database ***103a***

MAC Initiator ***106n***	Responder MAC ***102n***	Responder Mode ***306a***	Cryptog. Params ***199a***	Responder Ephem. Public Key ***102p***	Responder Ephem. Private Key ***102r***	Initiator Ephem. Public Key ***101c***	Bootstrap keys ***399***	PKEX Key ***198***	Shared Secret k1 ***314a***	Shared Secret k2 ***314c***	Shared Secret ke ***327a***
3860772OA	6cf049080	Fig. 3	A	Pr-1	pr-1	Pi-1	Br-1, br-1, Bi-1	X1	k1-1	k2-1	ke-1
386077208	4a508b99	Fig. 5a	A	-	-	Pi-2	Br-1, br-1, Bi-2		-	-	-
386077209	1c6f654d2	Fig. 5b	A	-	pr-2	Pi-2	Br-1, br-1, Bi-3	NA	-	-	-
386077400	7071bce29	102'	B	-	-	-	{Br-4, br-4, Bi-4} B	NA	-	-	-
3860773cc	7071bce29	Fig. 3	C	Pr-4	pr-4	Pi-4	{Br-5, br-5, Bi-5} C	X4	k1-4	k2-4	ke-4

● ● ●

Device Credentials Table ***345***

Device Identity ***106i***	MAC. Initiator ***106n***	Primary Network Credentials ***350a***	Backup Network Credentials ***350b***
20001	3860772OA	***109 - 1***	***109 - 3***
20002	386077209	***109 - 1***	***109 - 4***
20003	386077400	***109' - 1***	***109' - 3***
20004	3860773cc	***109' - 2***	***109' -4***
20005	386077a09	***109 - 2***	***109 - 5***

Figure 3d

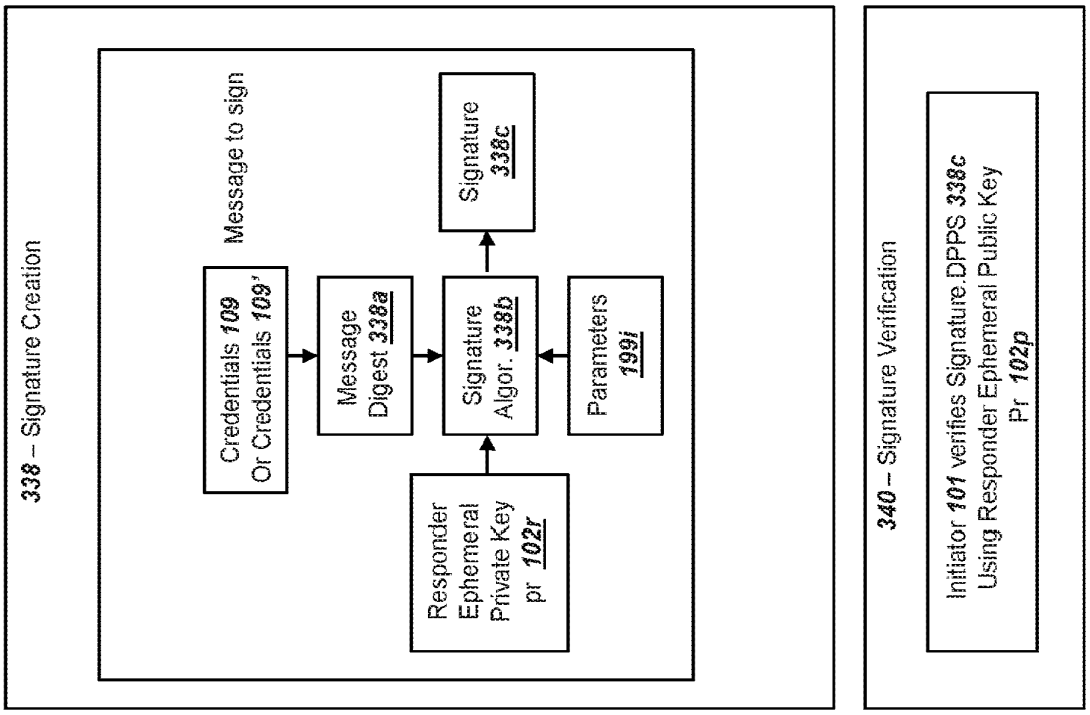
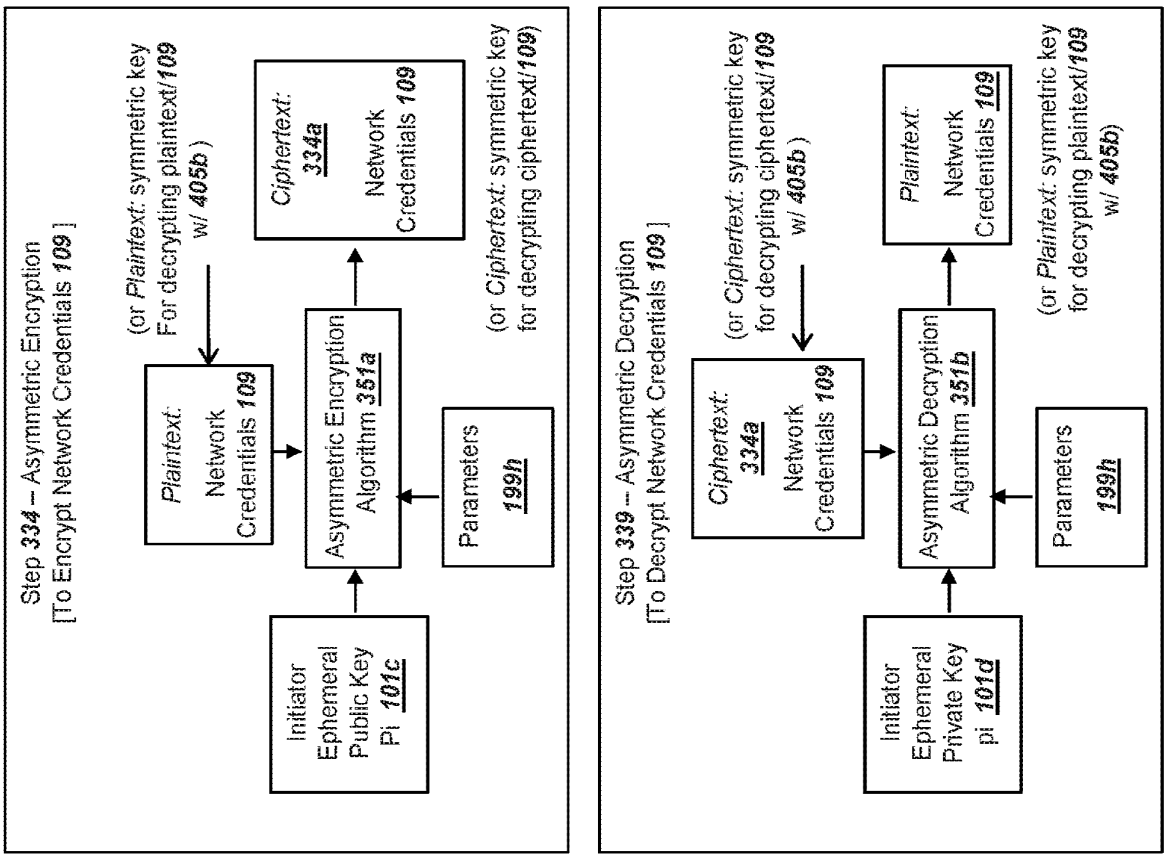
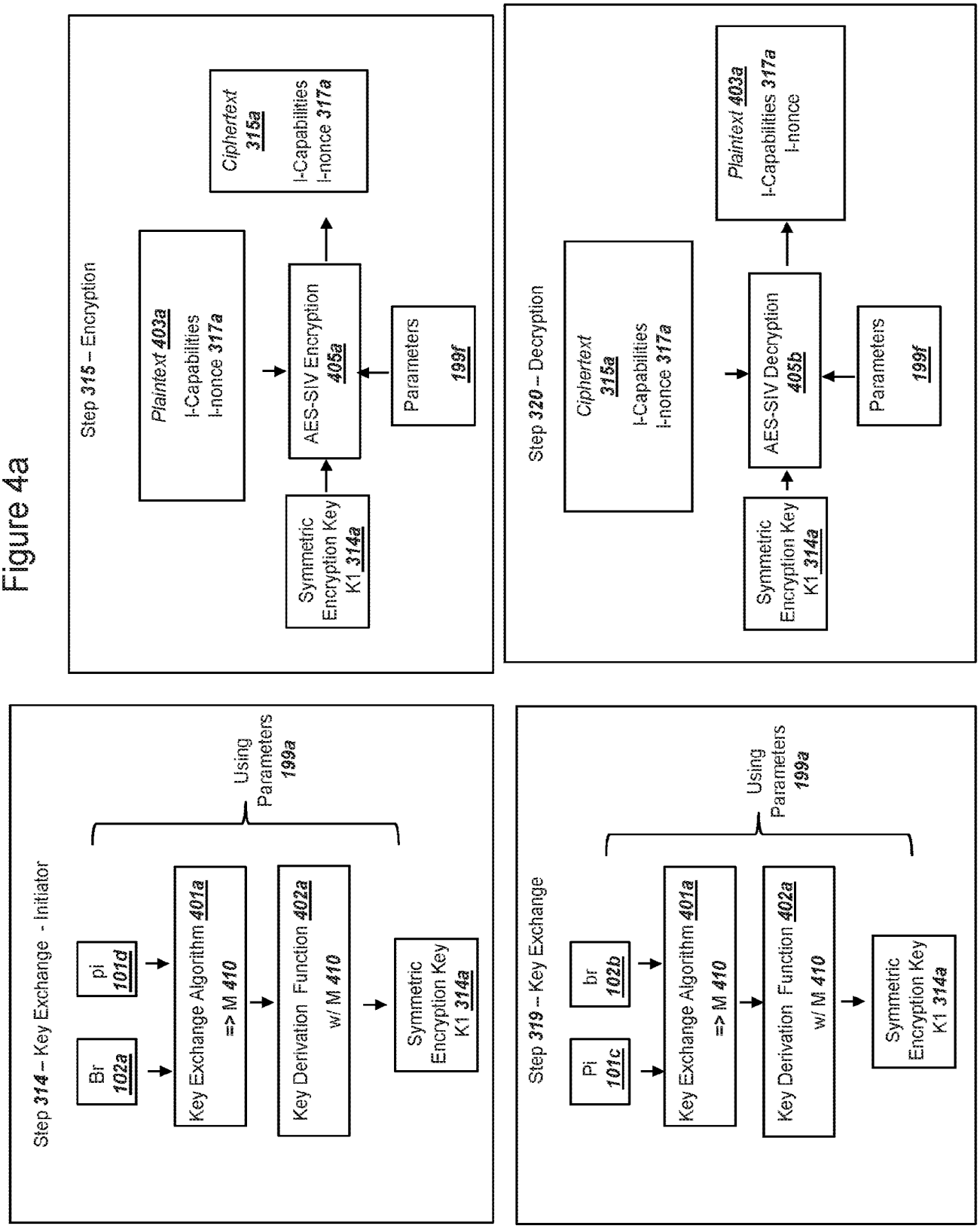


Figure 3e



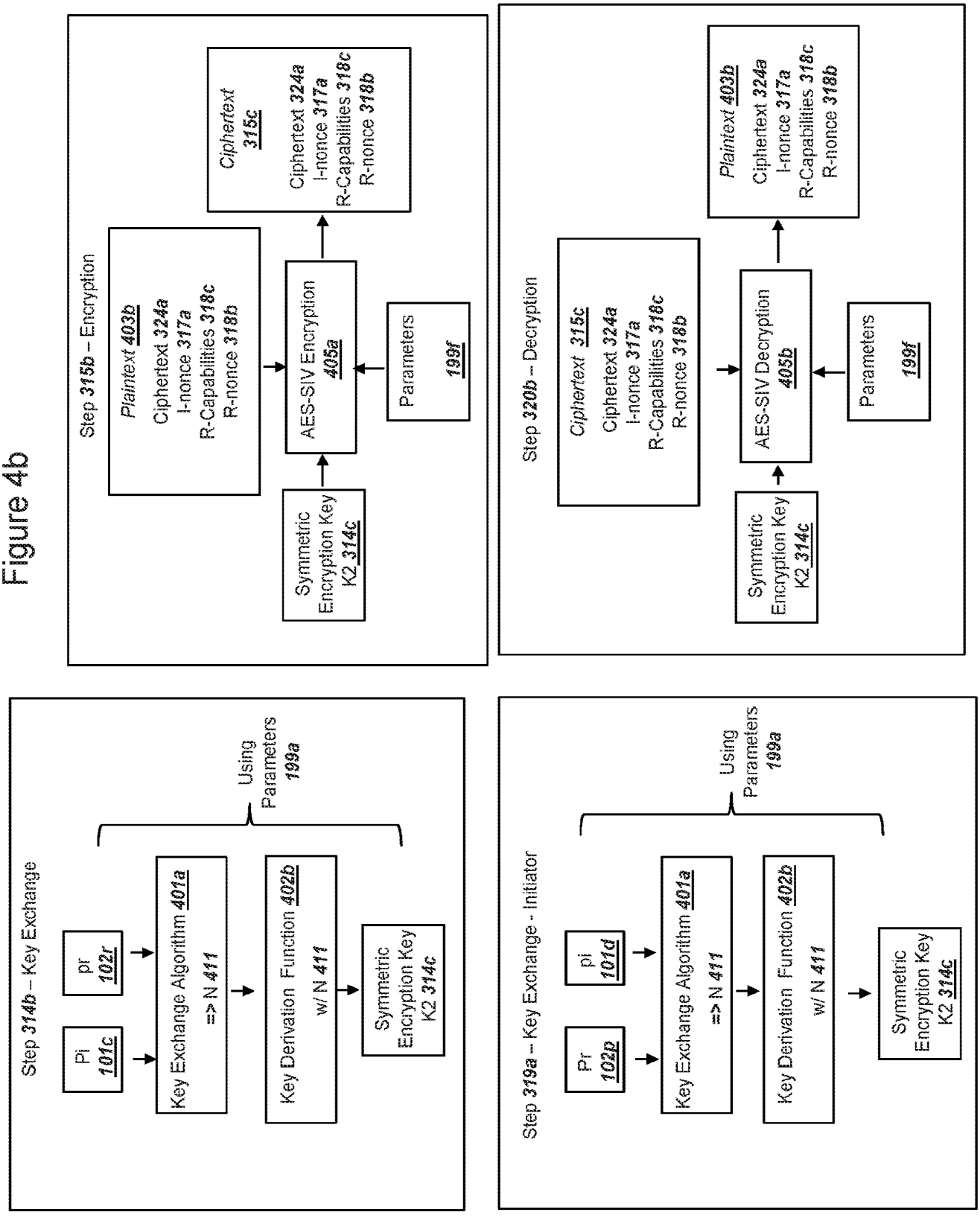
Petitioner Samsung Ex-1004, 1521

Figure 4a



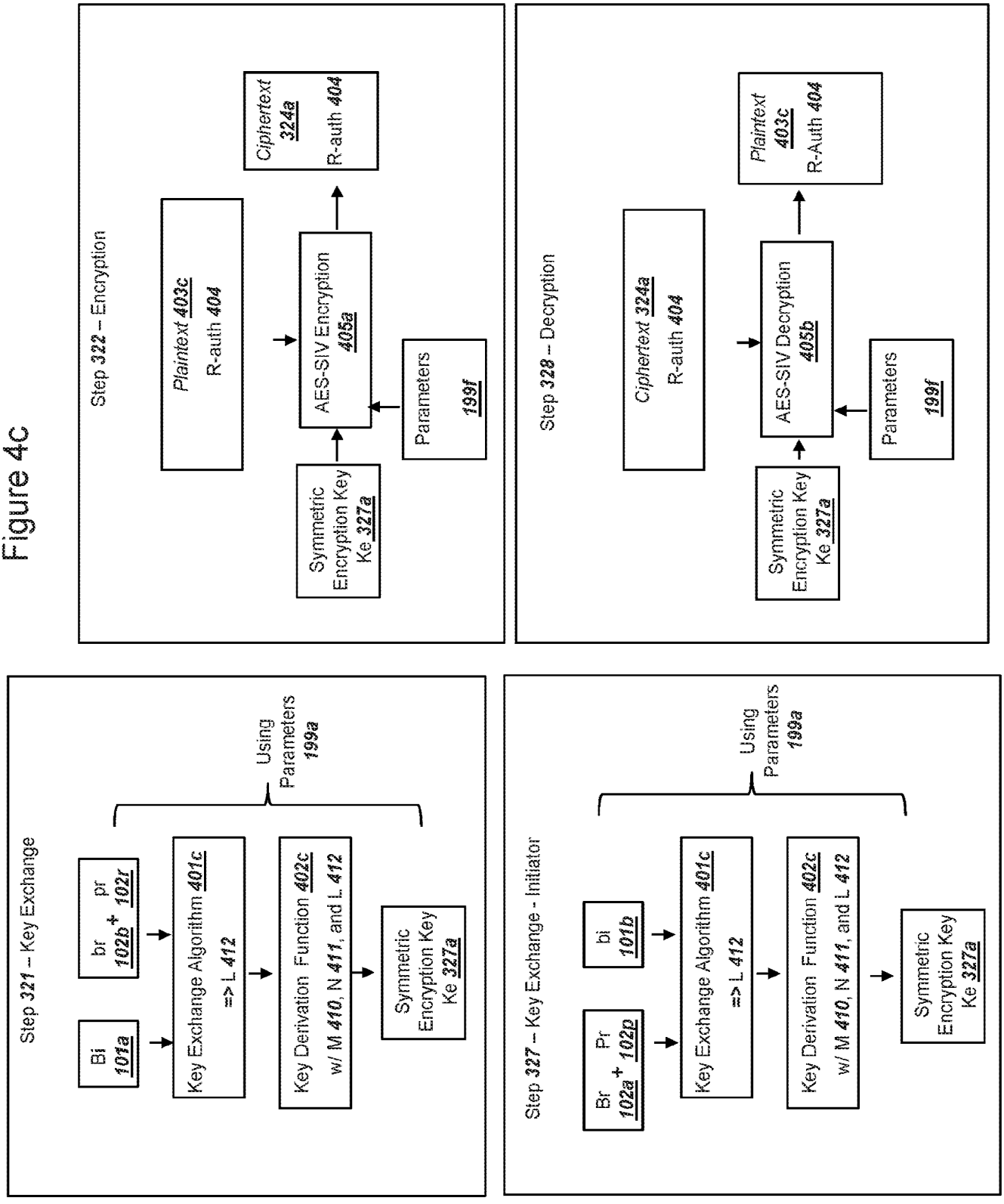
Petitioner Samsung Ex-1004, 1522

Figure 4b



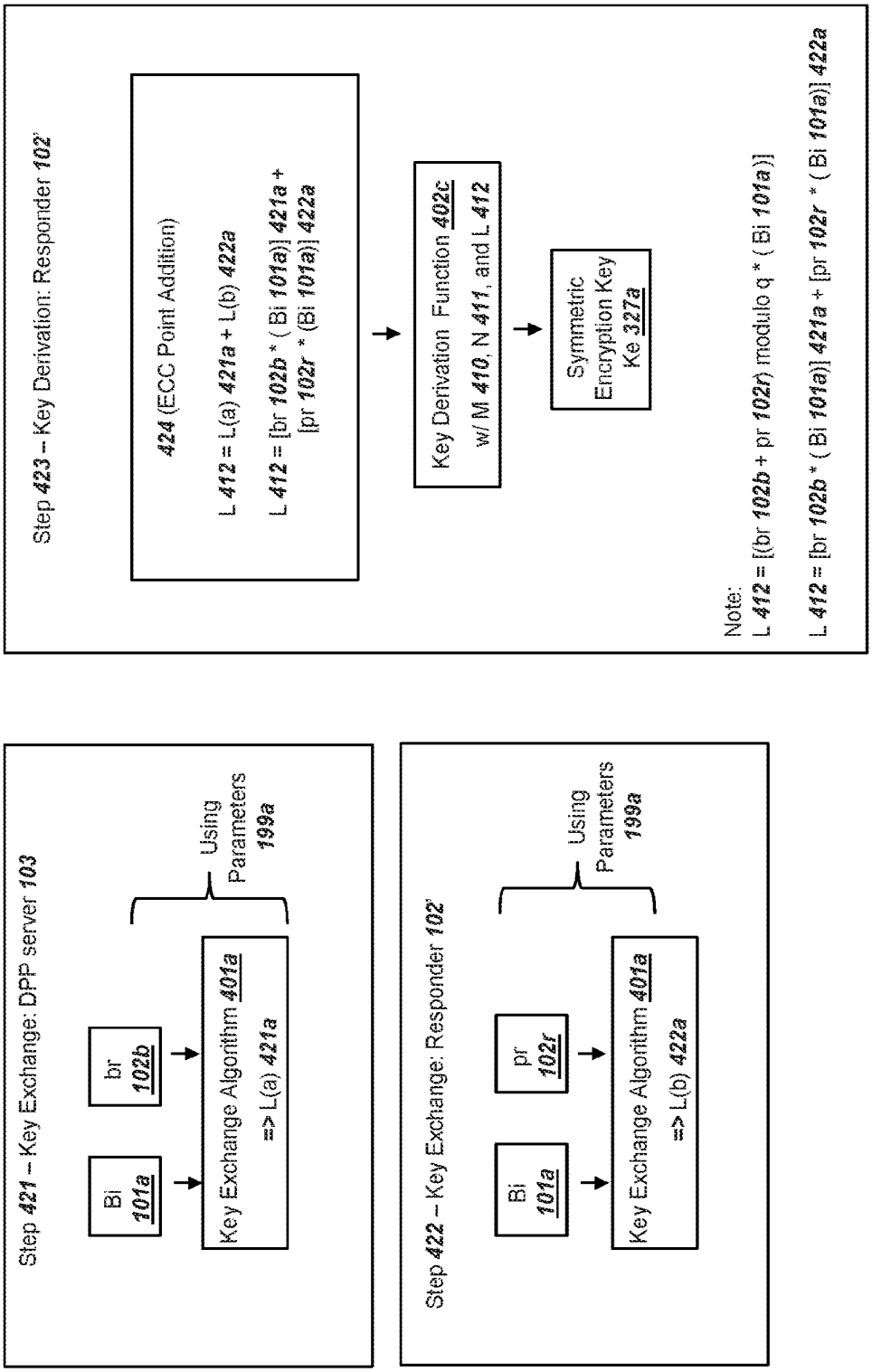
Petitioner Samsung Ex-1004, 1523

Figure 4c



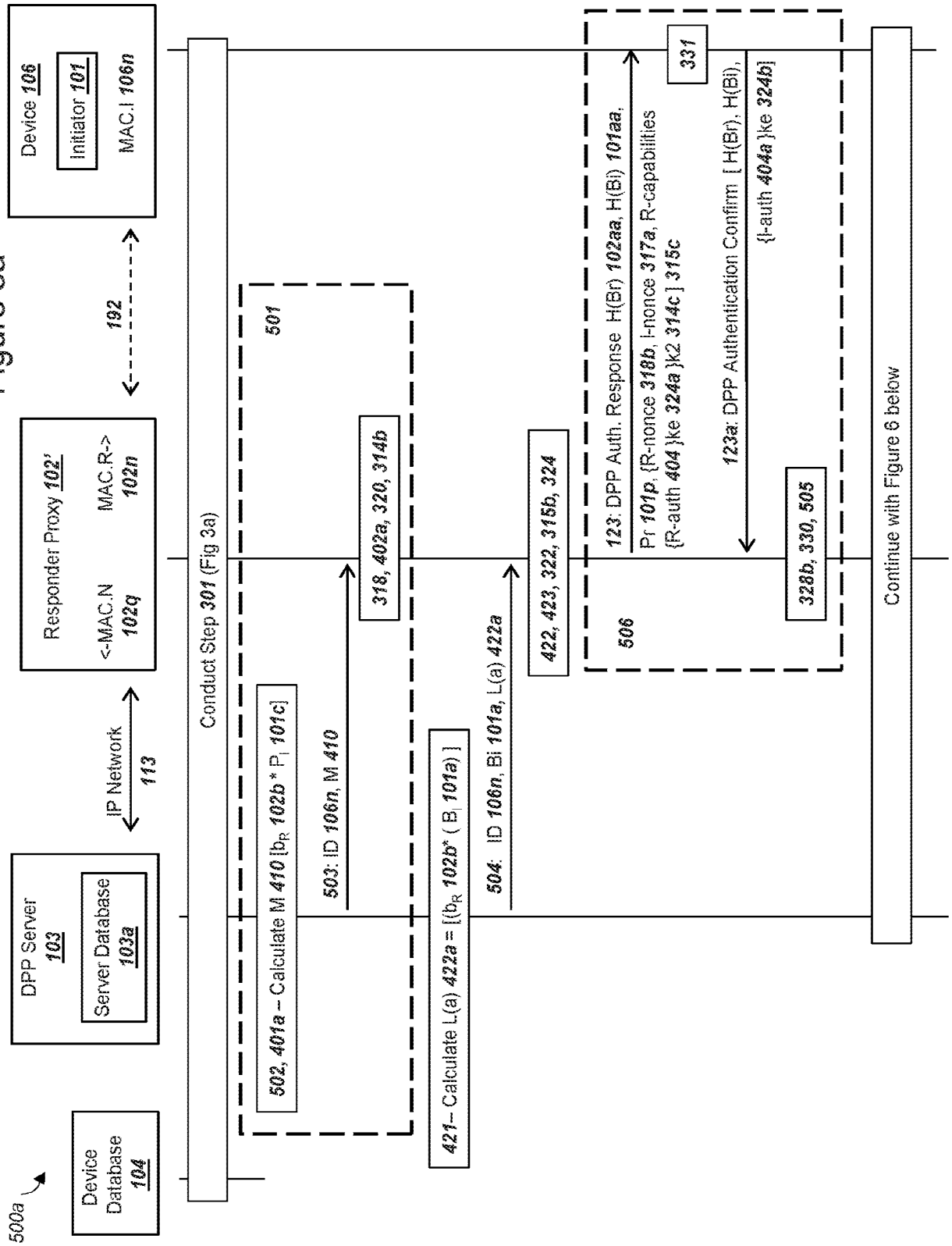
Petitioner Samsung Ex-1004, 1524

Figure 4d



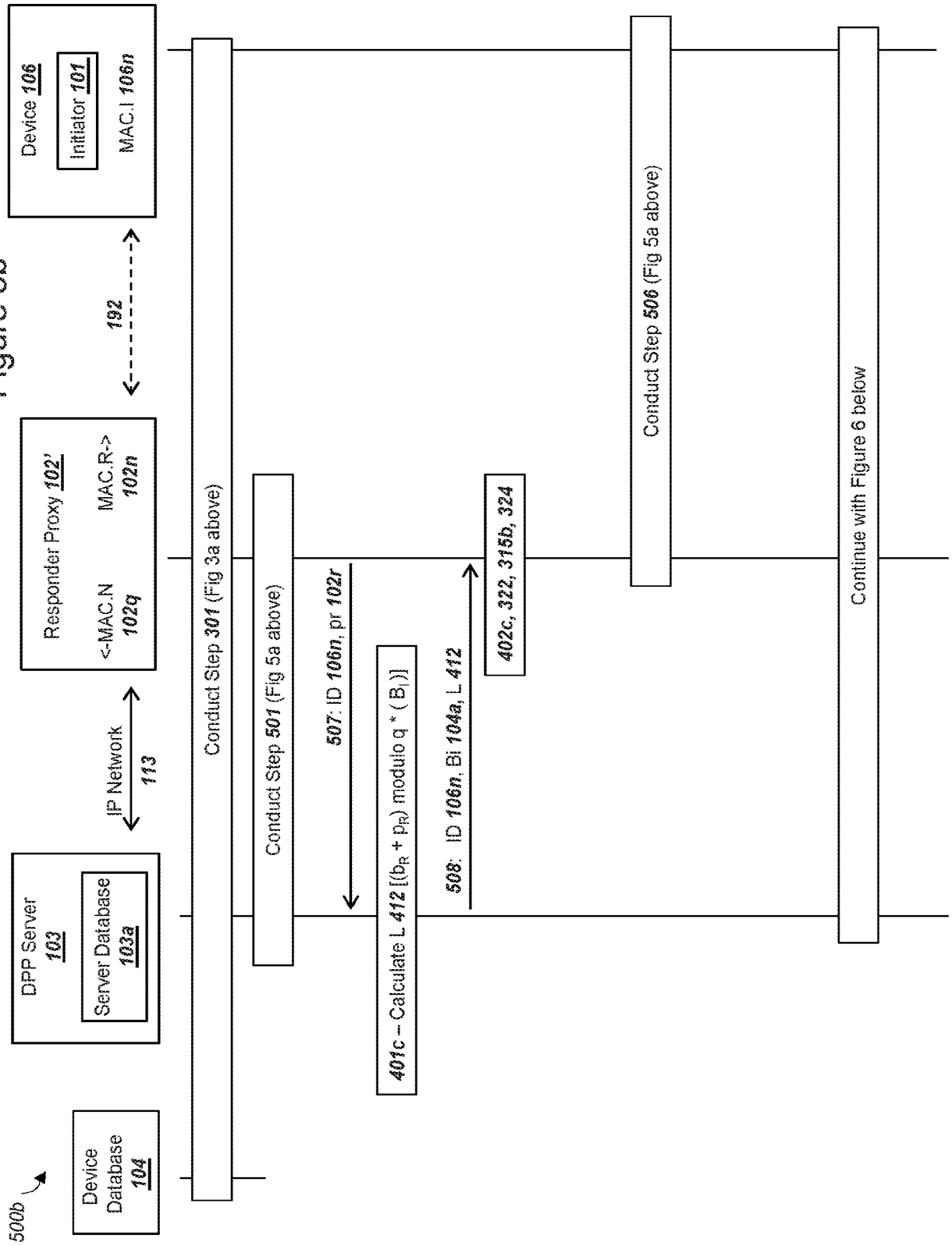
Petitioner Samsung Ex-1004, 1525

Figure 5a



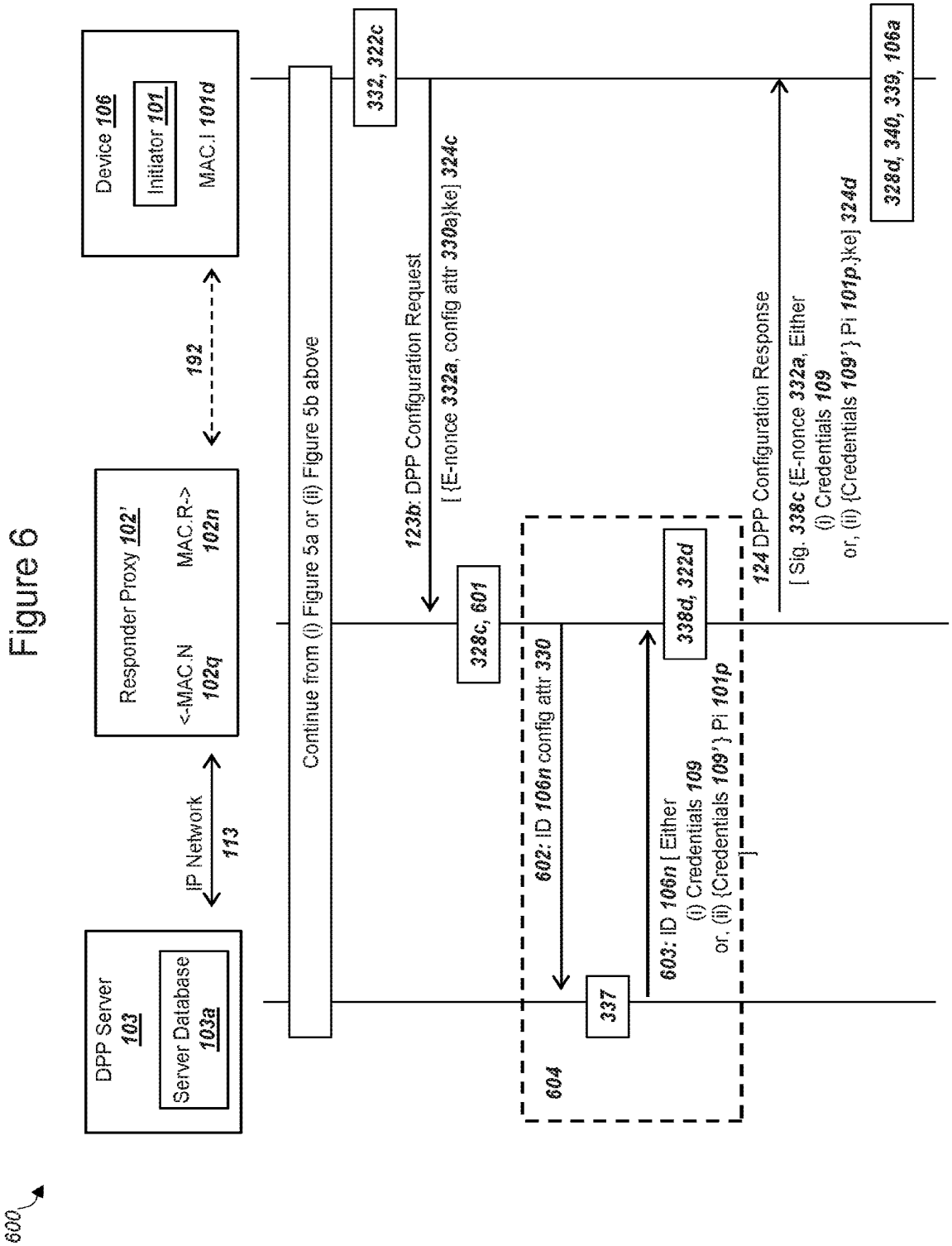
Petitioner Samsung Ex-1004, 1526

Figure 5b



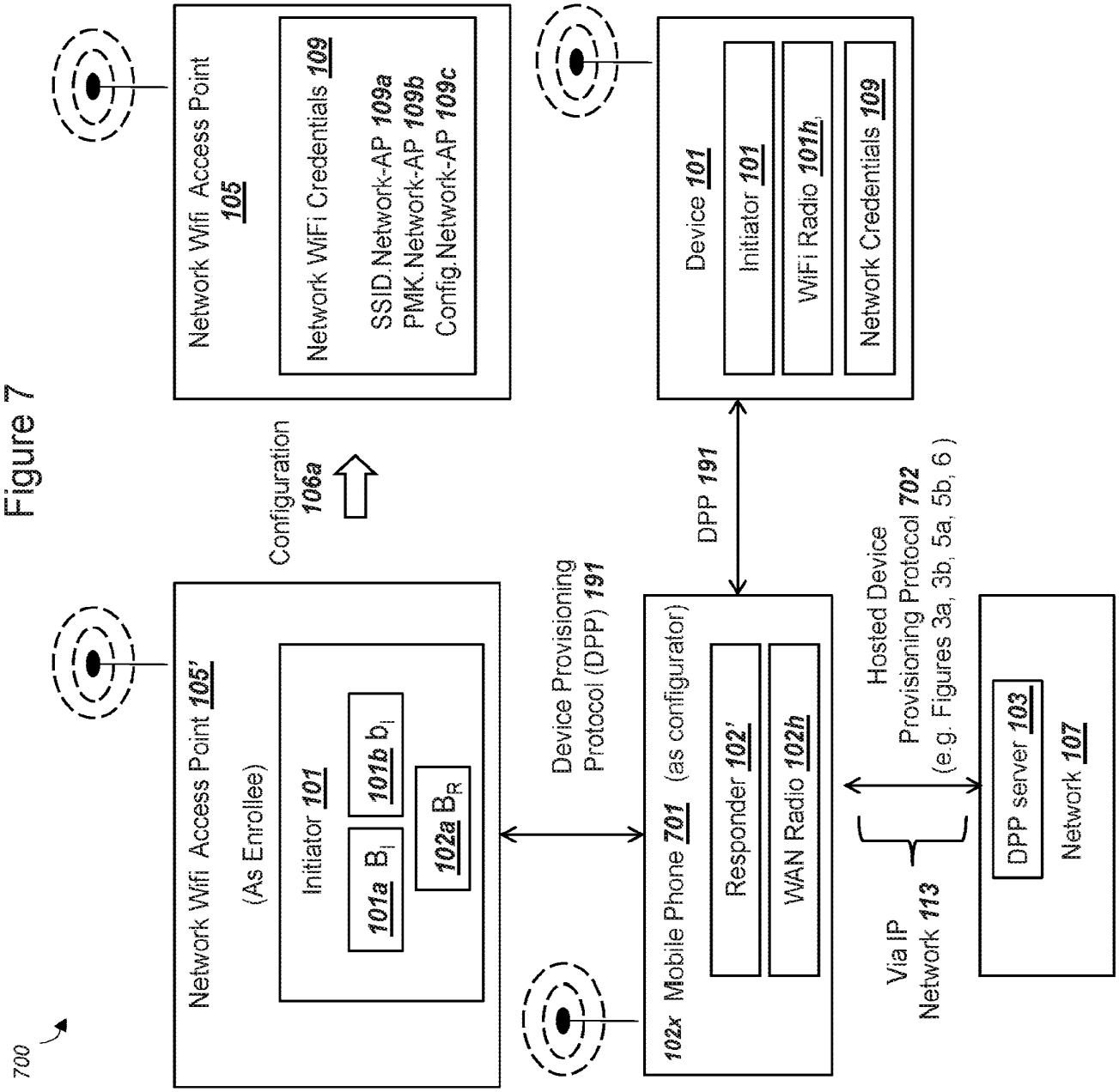
Petitioner Samsung Ex-1004, 1527

Figure 6



Petitioner Samsung Ex-1004, 1528

Figure 7



Petitioner Samsung Ex-1004, 1529

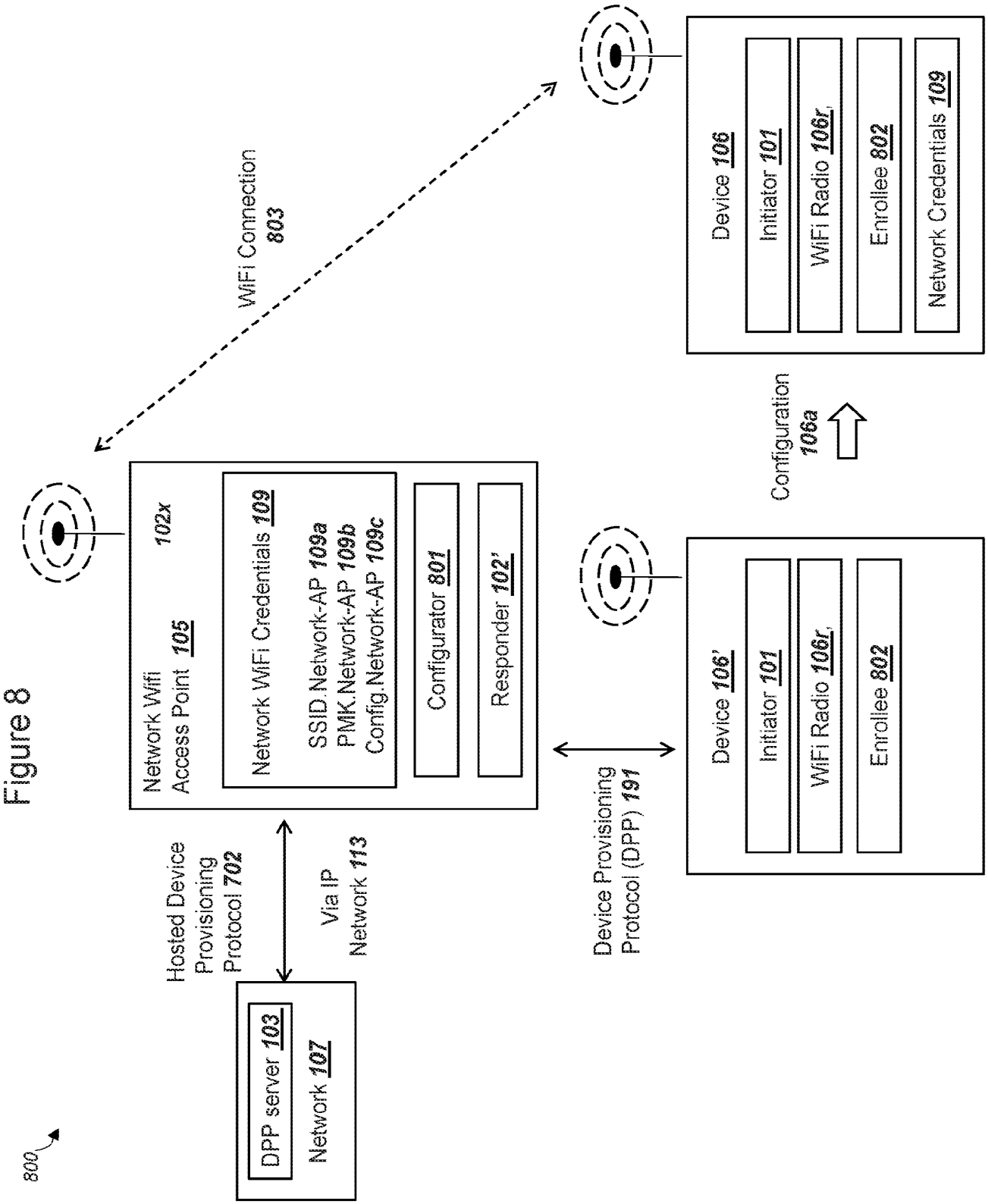
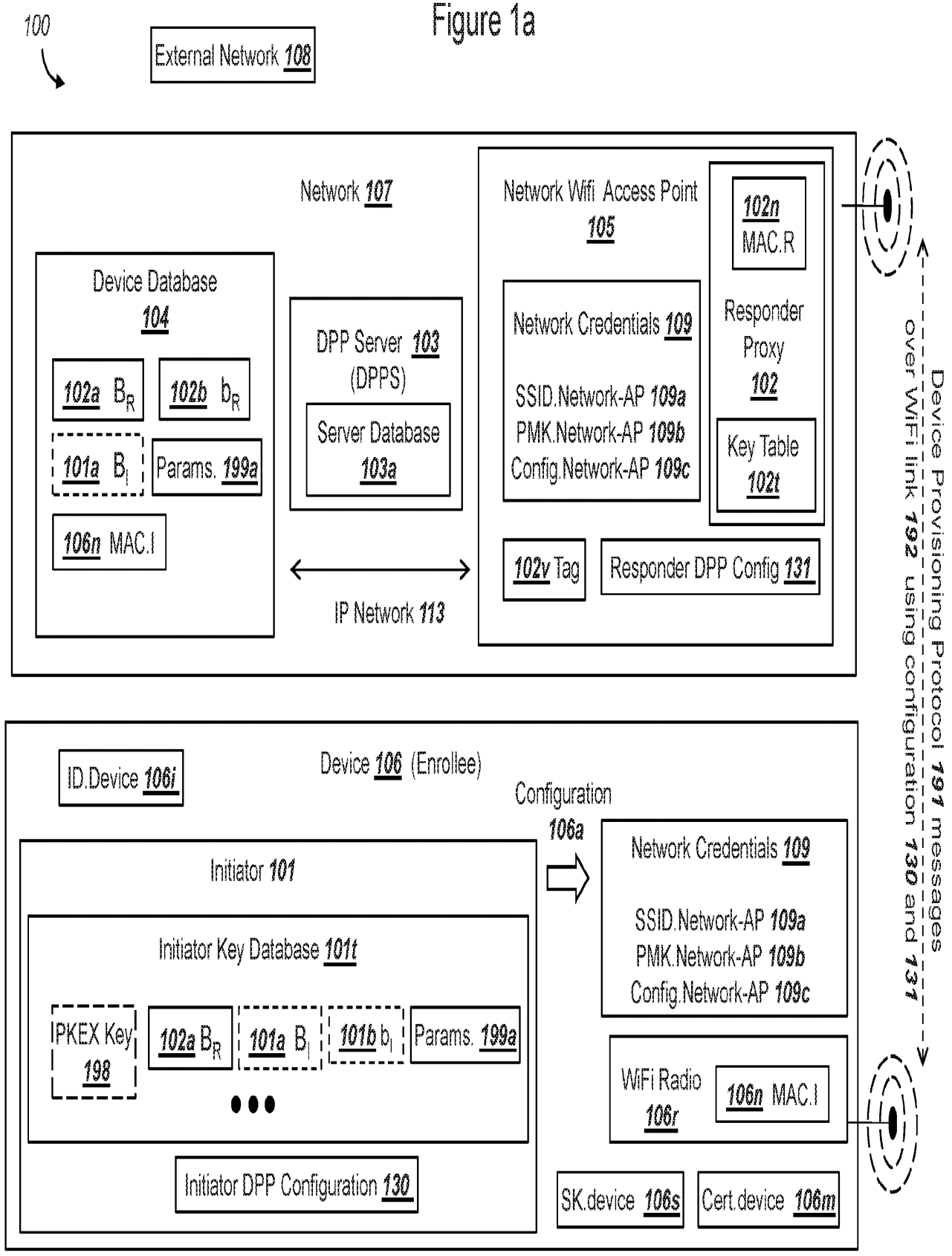


Figure 8

Petitioner Samsung Ex-1004, 1530

Figure 1a



Petitioner Samsung Ex-1004, 1531

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC PAYMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
16/453,682	08/05/2021 04:38:23 PM ET	63121/0184

Title of Invention

Embedded Universal Integrated Circuit Card Supporting Two-Factor Authentication

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	2186	FILED BY	Victoria Gilmore
PATENT CENTER #	60272997	AUTHORIZED BY	Albert Boardman
CUSTOMER #	1912	FILING DATE	06/26/2019
CORRESPONDENCE ADDRESS	-	FIRST NAMED INVENTOR	John A. Nix

Payment Information

PAYMENT METHOD	PAYMENT TRANSACTION ID	PAYMENT AUTHORIZED BY
CARD / 1044	E202185G39513012	Victoria Gilmore

FEE CODE	DESCRIPTION	ITEM PRICE($)	QUANTITY	ITEM TOTAL($)
2801	REQUEST FOR CONTINUED EXAMINATION (RCE) - 1ST REQUEST (SEE 37 CFR 1.114)	680.00	1	680.00
			TOTAL AMOUNT:	$680.00

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 January 2021 (14.01.2021)

WIPO | PCT

(10) International Publication Number
WO 2021/007235 A1

(51) International Patent Classification:
H04L 29/06 (2006.01) *H04L 9/30* (2006.01)

(21) International Application Number:
PCT/US2020/041048

(22) International Filing Date:
07 July 2020 (07.07.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/871,298 08 July 2019 (08.07.2019) US

(72) Inventor; and
(71) Applicant: **NIX, John, A.** [US/US]; 807 Davis St, Unit 2207, Evanston, IL 60201 (US).

(74) Agent: **THYMIAN, Marcus, J.**; McDonnellBoehnenHulbert & Berghoff LLP, 300 South Wacker Drive, Chicago, IL 60606 (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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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))*

WO 2021/007235 A1

(54) **Title:** EAP-TLS AUTHENTICATION WITH CONCEALED USER IDENTITIES AND WIRELESS NETWORKS

(57) **Abstract:** A device, mobile operator, network, and a device provider can exchange messages for EAP-TLS authentication. The network can include an authentication server Junction (AUSF). A device and a device provider can record both a device certificate and a device provider certificate. The network can receive an encrypted identity for the device and forward The identity to the device provider. The device provider can send tire device certificate and the device provider certificate to the network. The network can (i) receive a "client hello", (ii) select a network public key and private key, and (hi) send a certificate signing request to the device provider with the network public key, and (hr) receive a network certificate verified by the device provider certificate. The network can receive the device certificate from the device in a TLS handshake and mutually authenticate with the device using the received network certificate and the device certificate.

Petitioner Samsung Ex-1004, 1534

EAP-TLS AUTHENTICATION WITH CONCEALED USER IDENTITIES AND WIRELESS NETWORKS

RELATED APPLICATION

The present application claims priority to and hereby incorporates by reference the entirety of U.S. Provisional Patent Application No. 62/871298, filed on July 8, 2019.

BACKGROUND

Technical Field

The present systems and methods relate to conducting an extensible authentication protocol - transport layer security (EAP-TLS) authentication with concealed subscriber identities for wireless networks, and more particularly to storing and using keys and certificates with a device, a network, and a device provider in order to improve efficiency, increase flexibility, and enhance security of the EAP-TLS authentication.

Description of Related Art

The use of EAP-TLS authentication has been widely adopted over the past decade within enterprise wireless networks using WiFi standards such as 802.11. Standards for EAP-TLS authentication include the Internet Engineering Task Force (IETF) Request for Comments (RFC) 5216 titled "The EAP-TLS Authentication Protocol", which is herein incorporated by reference is its entirety. Since approximately 2017, the European Technical Standards institute (ETSI) has begun publishing standards for 5* generation (5G) wireless wide-area networks (WAN), such as described in TS 133 501 V15.1.0 titled "5G; Security architecture and procedures for 5G System", which is herein incorporated by reference in its entirety. Unlike prior 4th generation wireless WAN networks, 5G standard networks can include the support of authentication of devices using EAP-TLS, as described in Annex B of TS 133 501. EAP-TLS authentication enables the use of certificates for mutual authentication of a device and a wireless network. Previous wireless WAN standards for authentication such as with 4G LTE, 3G, etc. have depended on a pre-shared secret key recorded in a device and a secure server within the network. The significant changes with 5G networks removes some previous limitations, such as requiring the use of a pre-shared secret key, but also creates additional needs in the art in order to efficiently support EAP-TLS authentication for thousands or more different types of devices connecting with hundreds or more different wireless networks globally.

The inclusion of support for EAP-TLS in 5G networks can ease the support for wireless device authentication and secure connectivity for many applications, where a pre-shared secret key such as a key K within a subscriber identity module (SIM) or embedded SIM may no longer be required. Most commercial arrangements for device owners or device users bind the use of the

Petitioner Samsung Ex-1004, 1535

pre-shared secret key K within a device to a particular home wireless network that also records and derives keys from the same pre-shared secret key. 5G networks also include the use of a subscription concealed identifier (SUCI) as a partially encrypted form of a subscription permanent identifier (SUM). 5G networks as specified in TS 133 501 also include tire use of a network access identifier (NAI) according to IETF RFC 7542 titled "The Network Access Identifier", which is herein incorporated by reference in its entirety.

In addition, in 2018 the IETF updated TLS version 1.2 to TLS version 1.3 in IETF RFC 8446 titled "The Transport Layer Security (TLS) Protocol Version 1.3", which is herein incorporated by reference in its entirety. ETSI published standards as of May 2019 such as TS 133 501 for 5G networks do not address the use of TLS version 1.3 with EAP-TLS authentication. A need exists in the art for devices, wireless networks, devices, and device providers to support an EAP-TLS authentication using TLS version 1.3 with the encrypted or partially encrypted subscription concealed identifiers as described in TS 133 501 Annex B, where the corresponding subscription permanent identifiers can use tire NAI format.

Although the use of PKI certificates and corresponding private keys for EAP-TLS authentication can avoid the logistical and provisioning complexities of using a pre-shared secret key for authentication (such as with 4G and 5G AKA authentication), the use of EAP-TLS authentication can create a new series of problems or needs in the art for the many different devices and wireless networks winch use EAP-TLS authentication. These problems may be avoided or more easily addressed with existing implementations of EAP-TLS authentication with corporate or enterprise WiFi networks, for example, because the enterprise can control both the device and the wireless network. The enterprise can ensure (i) a root certificate or certificate issuer certificate stored in tire wireless network can be used to authenticate the device and (ii) and a root certificate or certificate issuer certificate stored in the device can be used to authenticate the wireless network, winch is often required for mutual authentication using EAP-TLS standards. In addition, TLS authentication and PKI certificates and secret keys can support many different sets of cryptographic algorithms and cryptographic parameters, and an enterprise deploying EAP-TLS authentication currently can control the device certificates and the wireless network certificates to ensure the PKI certificates support compatible sets of cryptographic parameters and algorithms.

However, for widespread and even global deployments of 5G networks and devices, there can be hundreds or more different device manufacturers, device providers, and wireless networks, where each could use different combinations of preferred cryptographic algorithms, parameters, and/or certificate issuers. In addition, (A) tire device manufacturers or device providers storing device default PKI certificates in devices with device default cryptographic algorithms and cryptographic parameters may riot support (B) the network PKI certificates and network supported

Petitioner Samsung Ex-1004, 1536

cryptographie algorithms and cryptographic parameters. For many applications, no pre-existing relationship may exist between a device manufacturer and a wireless network before device distribution to end users and the device's first connection to a 5G wireless network. In other words, the relatively dosed environment of an enterprise configuring devices for connection o a Wi-Fi network with EAP-TLS authentication is much less complex since the misprise controls the device and network before the devices connect to the enterprise Wi-Fi. For 5G networks, the wireless network operator will typically not control tire device before devices connect to foe 5G network.

As one example of complexify for EAP-TLS authentication for 5G networks, a device, using EAP-TLS authentication and a subscription concealed identifier, may not record a certificate chain or root certificate that can be used to authenticate a network certificate received from the network in a TLS handshake for EAP-TLS authentication. A need exists in the art for a wireless network to support EAP-TLS authentication for any one of hundreds or more of different devices types using concealed identities from hundreds or more different device providers. A need exists in the art, where the device uses a subscription concealed identifier, to support verification of a certificate received from any one of hundreds or more different wireless networks, where no pre-existing relationship may exist between the device owner and the wireless network.

For example with the hundreds or more different 5G wireless networks globally and with a manufactured or installed set of certificate issuer certificates or root certificates, a device could receive a server certificate from an authentication server function (AUSF) that does not chain or may not be able to be verified through a root certificate or certificate issuer certificate recorded by the device. In addition, the server certificate may use ciyptograpiiie algorithms and parameters that are different than those used by a certificate issuer certificate stored within the device. The device may not have access to an IP network without first authenticating with a wireless network. This potentially creates significant problems, where the device does not have access to the IP network o receive (i) a supporting certificate for verifying the server certificate from (ii) a server other than through the wireless network where a connection is attempted. In other words, the supporting certificate may be required in order to verify the server certificate received from fire wireless network, but the device may not be able to receive the supporting certificate from an IP network without first successfully completing an EAP-TLS authentication.

Similarly, (i) the wireless network may control or establish a first certificate drain for a server certificate used by an authentication server function (AUSF), (ii) a device provider may control or establish a second certificate chain within a device before distribution to and end user for verifying tire server certificate, and (in) the server certificate from a wireless network may not be verifiable by the second certificate chain. The problem of (iii) is compounded by the thousands

Petitioner Samsung Ex-1004, 1537

or more potential combinations of different certificate issuer certificates and supported cryptographic parameters for required certificates to successfully complete an EAP-TLS authentication. A need exists in the art for ensuring a device can authenticate a server certificate from a wireless network using both (i) a certificate issuer certificate recorded in the device before distribution, and (ii) cryptographic algorithms and parameters recorded hi the device before distribution that would support verifica tion of the server certificate.

A need exists in the art for a device to support EAP-TLS mutual authentication, without requiring a device user to load a server certificate or supporting certificate issuer certificate in order tor the device to conduct mutual authentication with a wireless network. A need exists in the art for a device to record compatible cryptographic algorithms and cryptographic parameters for a server certificate without requiring a device user to load compatible cryptographic algorithms and cryptographic parameters that can be required in order for the device to use or verify the server certificate.

A need exists in the art for a device to operate with at least one certificate authority different than certificate authorities for authenticating web sites during web browsing or establishing TLS sessions with general servers. Some common root certificates or certificate issuer certificates may be essential for devices to properly work with millions of different servers, but the device may prefer to prevent those certificates from being used by the device to authenticate a 5G network server certificate. As one example, hackers could easily, successfully, and authoritatively obtain a valid certificate from 'Let's Encrypt" for an example domain name of "5g-eap-hacker.com". If the device is based on Android f om 2020, then the device would store a root certificate or intermediate certificate for "Let's Encrypt".

The hackers could (i) set up a rouge gnode b, and (ii) AUSF using the valid "Let's Encrypt" certificate for the example domain "5g-eap-hacker.com". The device could connect to fire girodeb, select use of EAP-TLS authentication, and receive the valid certificate and verify the certificate using the root certificate or intermediate certificate for "Let's Encrypt". The successful verification of the valid certificate for "5G-eap-hacker.com", and associated authentication steps for the device to authenticate the network, could result in the device believing the AUSF was authenticated, when in fact it was operated by potential hackers. A need exists in the art for a device provider to have control over the at least one certificate authority for authentication of an AUSF that is different than certificate authorities for authenticating web sites. A need exists in the art for a network to obtain, select, and use a certificate for the network that can be authenticated using the at least one certificate authority certificate stored in the device in order to for the device to authenticate the AUSF. Without access to an IP network (such as after successfully completing EAP-TLS authentication with the wireless network), the device may not be able to request a server certificate that can be verifiable

Petitioner Samsung Ex-1004, 1538

through an installed certificate chain in a device, and consequently mutual authentication using methods such as EAP-TLS may fail or require manual intervention or unnecessary interaction with a device user in order for the EAP-TLS authentication to succeed. Again, the above problems related to server certificate verification by a device and the use of different cryptographic algorithms and parameters for different server certificates received by a device is compounded by the fact that there could globally be hundreds of mobile network operators and thousands or more device providers.

A device provider may not know which of the hundreds of different mobile networks a particular device may try to connect with until the device is turned on by a user or first attempts to connect with a mobile network in a particular geographical region. Even if a particular geographical region is known, such as the United States, before a device attempts to connect to one of a handful of wireless networks, the preferred wireless network may not he known to a device provider before a device attempts to connect A need exists in the art for a device to receive a server certificate in a TLS handshake from a wireless network supporting EAP-TLS, where verification of the server certificate does not require intervention from an end user, in order to allow a device to mutually authenticate with EAP-TLS using a subscription concealed identifier with a plurality of different mobile network operators.

In addition, a device provider may control or establish a certificate chain for verifying a device certificate stored within a device, and (ii) art authentication server function may record a set of certificate issuer certificates and/or root certificates with cryptographic algorithms and cryptographic parameters that may not support the verification of the device certificate. Devices from hundreds or more different device providers can also use different certificate chains for the device certificates and also different sets of cryptographic algorithms and cryptographic parameters. Or, as explained 3 paragraphs above, a hacker with a device could easily obtain a valid certificate for use with a device from an important and valid certificate authority on tire Internet, such as "Let's Encrypt". Thus, a device with a valid certificate (at least verifiable through a "Let's Encrypt" intermediate or root certificate) may attempt to connect with a wireless network. A need exists in the art for devices with some valid certificates to remain unauthentleated with a network and require different device certificates for authentication with the network.

The request by the authentication server function for supporting certificate issuer certificates and/or cryptographic algorithms and cryptographic parameters for a received device certificate may take additional time, complexity, and potential security issues for securely receiving a supporting certificate issuer certificate and/or cryptographic algorithms and cryptographic parameters in order to properly *verify* a received device certificate. A need exists in the art for ensuring an authentication server function can securely and efficiently authenticate a device

Petitioner Samsung Ex-1004, 1539

certificate received through a wireless network, where tire device can utilize a subscription concealed identifier.

Further, a need exists in the art for a network to verify that a device certificate has not been revoked when conducting an EAP-TLS authentication. Although TLS and EAP-TLS standards support the online certificate status protocol (QSCP) a device may not have Internet access until after the completion of an EAP-TLS authentication. Without Inter et access, the device may not be able to generate QSCP stapling for the verification that the device certificate has not been revoked. Consequently, a need exists in the art for a network to verify that a received device certificate has not been revoked, where the device sending tire device certificate may not have access general access to the internet to communicate with other servers than a network or MNG until after the completion of an EAP-TLS handshake.

Many other examples exist as well for needs in the art for devices to securely and efficiently conduct EAP-TLS authentication with wireless networks, where the device uses a subscription concealed identifier, and the above are examples are just a few and intended to be illustrative instead of hunting.

SUMMARY

Methods and systems are provided for a device, a mobile network operator (LINO), a network, and a device provider to support EAP-TLS authentication, where the device can use a subscription concealed identifier. A system can include a plurality of devices, a mobile network operator, a network, and a device provider. The device can record a subscriber permanent identifier (SUPI), a device certificate, a device private key corresponding to the device public key in the device certificate, an elliptic curve integrated encryption scheme (ECIES) function, a static network public key for the ECIES function, MNO parameters, a device provider certificate with a device provider public key, and cryptographic algorithms and cryptographic parameters used with both the device provider certificate and the device certificate. The subscriber permanent identifier can comprise a network access identifier (NAI) supporting IETF RFC 7542, where the NAI can include a user name and a realm. The realm can comprise a first portion with a provider identity and a second portion with a network identity". For some embodiments, the user name can comprise an International Mobile Subscriber Identity (IMSI).

The device can be a computing device for connecting with wireless networks such as a 5G network and the device can include a radio. The device can include a processor, memory?, a user interface, and a bus connecting the components within an enclosure. The device could comprise a radio module within a larger device, such as a radio module within an automobile, industrial equipment, a mobile handset such as a smart phone, and other possibilities exist as well for physical

Petitioner Samsung Ex-1004, 1540

embodiments of the device. The device can conduct an EAP-TLS authentication through the MNO with the network.

The mobile network operator can comprise a wireless network with geographically distributed radios and antennas in order to wirelessly connect with the plurality of devices. The MNO can include a MNO identity in order to uniquely identify the MNO, such as including a mobile country code and a mobile network code. The MNO can receive messages from the device through a wireless network and forward the messages to the network. The MNO can receive messages from the network and forward the messages to the device through the wireless network. The MNO can establish a secure session through an IP network with the network.

The network can include a plurality of servers operating in a coordinate manner to support communications with a device. The network can include an authentication server junction (AUSF) and a network database. The network can include an identity comprising the network identity in the second portion of a realm for the device subscriber permanent identifier. The network can conduct an EAP-TLS authentication with the device through the MNO supporting 5G standard such as TS 133 501 V15.1.0. The network can establish a secure session with a device provider. The network can operate a network database for storing data associated with each of the plurality of different devices connecting with the network through the MNO. The network can record or operate an elliptic curve integrated encryption scheme (EQES) function tor decryption of a subscription concealed identifier (SUCI) and a static network private key for the EQES function.

The device provider can comprise a collection of servers for an entity that controls the device before the device connects with the MNO. The device provider can represent a device manufacturer, a device distributor, or a device owner for a plurality of different devices. The device provider can include an identity' comprising the provider identity' in the first portion of a realm for the device subscriber permanent identifier. The device provider can operate a device database storing data for the plurality of devices. The device provider can record the subscriber permanent identifier, a device certificate for the device, a device provider certificate, a device provider private key' for the public key' in toe device provider certificate, and cryptographic algorithms and cryptographic parameters used with at least the device provider certificate.

The device can generate the subscription concealed identifier (SUCI) using at least the SUPI and the EQES encryption function with toe static network public key. The device can use the MNO parameters to establish a radio resource connection with the MNO. The device can send the SUCI to the MNO, and tire MNO can use the plaintext second portion of the realm to select the network and send the SUCI to the network. For some embodiments, the device can send the MNO a realm portion of toe SUPI without a user name portion, and tire EQES encryption function operating on the user name portion can be omitted.

Petitioner Samsung Ex-1004, 1541

The network can use the plaintext second portion of the realm to select the static **provider** private key for the ECIES **function,** where the static **network** private key **corresponds** to the static network **public** key for the ECIES **function** stored and used by the **device** to generate the SUCI. **The** network can use **at** least the ECIES **function** and the static network private key for the ECIES **function** to decrypt the SUCI and read the SUPI. **The** network can send the **SUPI** to the device provider. For some embodiments, the network can send a certificate signing request **(CSR)** for a network static **public** key **either** (i) along with **the** SUPI (if the network determines a valid network certificate is not **current** or present for **a provider** domain in **a** SUCI or SUPI), or **(ii)** after the network and device provider establish a secure session.

The device **provider** can receive the SUPI from the network and use the SUPI **to** select a device **provider** certificate **and** a device certificate for the **device** from the device database. The device provider can send at least the **device provider certificate** and the device certificate **to** the network. The network can store the received device provider certificate and the device certificate along with tire **SUPI** and the SUCI in the network database. For embodiments where the user portion is omitted from the SUCI as sent by the device, then the device provider can send the network the **device provider certificate** and omit the device certificate. **Or,** the device premier could send a bulk set of device certificates to the network, without association to a specific SUCI as received by fee network. For **embodiments** where network sends a CSR **to** device provider before **network** receives a TLS **"client** hello" message **from** the **device,** then device provider **would** send network fee network certificate signed by **the** device premier before the network sends a command to fee **device** to start an EAP-TLS session.

The network can send an EAP-TLS start command to the device **through** the MNO. The network can receive a "client hello" and fee SUCI from the MNO, where fee "client hello" can support TLS version 1.3 as specified in RFC 8446. Tire "client hello" can include an ephemeral elliptic curve **cryptography (ECC)** public key for the device, where the device stores fee **corresponding** device ephemeral private key. The **AUSF** can generate a server ephemeral PKI key pail compatible with the received ECC public key from fee device, including a **server** ephemeral public key. The **network** can store the PKI keys in the **network** database. The network **can** use the SUCI received with the "client hello"' to select the device provider **certificate** for fee device **from** fee network database. The "client hello" can include a list of supported **cryptographic** parameters **for** fee device.

The network can record **a** set of **cryptographic** parameters and a set of **network** public keys and network private keys for each member of fee set of **cryptographic** parameters. The network can select a network public key and a network private key for a set of cryptographic parameters, where fee set of **cryptographic** parameters was **included** in the "client hello" message received f om

Petitioner Samsung Ex-1004, 1542

the device. The network can send a Certificate Signing Request (CSR) to the device provider with the selected network public key, along with the set of cryptographic parameters for the network public key. The message with the CSR can also include the SUPI for the device.

The device provider can receive tire CSR and verify the CSR. The device provider can select the device provider private key to use for generating a network certificate based on the SUPI, where die device provider can determine die corresponding device provider public key stored by a device using the SUPI and a query from a device provider database. In this manner, a device provider can ensure that a network certificate generated by the device provider for the network can be verified by the device using the device provider public key stored by the device. The device provider can generate a network certificate for the network public key using the device provider private key (possibly selected using the SLIP!) and a signature algorithm and parameters specified in the device provider certificate. The network certificate can include a digital signature of the network public key using the device provider private key. The device provider can send the network certificate to the network.

The network can receive the network certificate and process a "server hello" using at least the received network certificate and the corresponding selected network private key. Data processed by the AUSF for a "server hello" can include standard fields such as the server ephemeral public key, selected extensions, a certificate request, and the received network certificate. The data processed in the previous sentence can be included in a "Certificate Verify" step. The AUSF can use the network private key (which can correspond to the network public key in the network certificate) to create a digital signature for a "Certificate Verify"' portion of a "server hello" message. A device provider identity in the device provider certificate can comprise a named certificate authority in the extensions.

The network can generate a shared secret and a symmetric ciphering key using the device ephemera] public key received in the "client hello" and the derived server ephemeral private key with an elliptic curve Diffie-Hellman (ECDH) key exchange. The network can store the derived shared secret and the symmetric ciphering key with die SUCI for the device in the network database. The server can encrypt at least the selected extensions, a certificate request, and the network certificate and the "CertificateVerify" into a ciphertext using the derived symmetric ciphering key. The network can send the device a "server hello" message with the server ephemeral public key and the ciphertext through the MNO, where the message to the MNO can include the SUCI so that the MNO can route the message to the device.

The device can receive the "server hello" message and take steps to process the message. The device can generate the shared secret and die symmetric ciphering key using the server ephemeral public key received in the "server hello" and the derived device ephemeral private key

Petitioner Samsung Ex-1004, 1543

osing the ECDH key exchange. The device can decrypt the ciphertext using the symmetric ciphering key. The device can read the plaintext value for the network certificate. The device can confirm that the device provider public key stored by tire device is used for verification of the network certificate. The device can verify the digital signature of the network public key using the device provider public key recorded by the device. In this manner, the network certificate as used by the AUSF and MNO can be trusted by the device without requiring the device to verify a parent certificate chain for the network certificate, since the device previously securely records the device provider public key required to verify the network certificate. If the received network certificate does not contain a digital signature verified through the device provider public key previously securely stored by the device, then the device could verify' the received network certificate through a certificate issuer chain stored within the device.

In other words, the device does not need Internet connectivity in order to verify the received network certificate as a server certificate in the "server hello" message because the network certificate can be verified with data previously stored inside the device before connecting with the MNO. The selection by the device provider of the derice provider private key used for the digital signature in the network certificate (by generating the network certificate from the CSR) can ensure file network certificate can be verified by the device without the device needing connectivity to an IP network (in order to query for other intermediate verifying certificates), thereby solving some needs in the art identified above.

The device can also verify the "Certificate Verify" digital signature portion of the decrypted "saver hello" ciphertext using the public key with the received network certificate. In tills manner, the device can trust that the AUSF has access to the network private key corresponding to the network public key in the network certificate. Verification of the "CertifieateVeriiy" digital signature can comprise an authentication of the network and/or MNO with the device.

The device can use the network certificate and the user portion of the SUFI to select a device certificate stored in the device. For example, the device could record multiple derice certificates and select a device certificate compatible with the cryptographic parameters and algorithms within the network certificate. Or the device could select the device certificate based on the extensions within the decrypted ciphertext within the "server hello" message. In some embodiments, the device can select a device certificate using the device provider identify' as a named certificate authority within the extensions, and the selected device certificate can be verified with the device provider certificate (where the device provider certificate includes the device prorider identify). The device can generate a response to the "server hello", where the response includes the selected device certificate and a digital signature or "CertificateVeiify" from the device. The device could generate the "CertifcateVeriiy" using the device private key

Petitioner Samsung Ex-1004, 1544

cos esponding the device public key in the device certificate. The device can encrypt the response with tie derived symmetric ciphering key and send the response to the MNO. The MNO can forward the response wife the SUCI to fee network.

The netwoik/AUSF can use the SUCI received wife the encrypted response to select the symmetric ciphering key f om the network database. The netwoik/AUSF can decrypt the encrypted response from fee device to read the plaintext values of the device certificate and the digital signature for the device "CertifkateVerify". The AUSF can compare (i) the received device certificate front the device with (ii) the device certificate stored in the network database, where the stored device certificate in the network database could be received previously as described above from the device provider. If (i) fee received device certificate from the device and (ii) the received device certificate from the device premier are the same, then the device certificate can be trusted or considered verified. Or, if the device certificate received from the device is not fee same as the device certificate received from the device provider, then the AUSF cart conduct a certificate verification 6 r the device certificate received Horn the device through a chain of certificate issuer certificates. The AUSF can verify the device "CertificateVerify" in the response by generating a digital signature using the device public key from the device certificate, and if the received digital signature is equal to or the same as the generated digital signature, then the device can be considered authenticated with the network.

The AUSF can use (i) the shared secret key from the TLS ECDH key exchange stored in the network database for the device and (ii) the plaintext data from the "client hello" and the "server hello" in order to generate a TLS master secret according to fee steps in IETF RFC 8446. In exemplary embodiments, the TLS master secret can be derived using a hash-based key derivation function with input of at least the ephemeral public key from the device in the "client hello" message, fee server ephemeral public key from the "server hello" message, the shared secret, and fee network certificate. Additional mutually shared data between the device and the network can be used to derive a TLS master secret.

The AUSF can use the generated or derived TLS master secret to generate at least a value for an extended master secret key (EMSK) as specified in EAP-TLS standards, such as inputting the TLS master secret into a key derivation function that could include a secure hash algorithm. The AUSF can use a portion of the EMSK, such as the leading 256 bits from an EMSK as an anchor key or Kaiisf. The AUSF can send the anchor key along wife the SUPI for the device to fee MNO. The MNO can use the anchor key to derive subsequent keys required for secure wireless communications with the device.

The device can conduct the equivalent steps of using the shared secret key and the plaintext data from the "client hello" and fee "server hello" in order to generate the TLS master secret

Petitioner Samsung Ex-1004, 1545

according to the steps in IETF RFC 8446. The device can use the generated or derived TLS master secret to generate at least a value for EMSK as specified in EAP-TLS standards, such as inputting the TLS master secret into a key derivation function that could include a secure hash algorithm. The device can use a portion of the EMSK, such as the leading 256 bits from an EMSK as an anchor key or Kausf. The device can use the anchor key to derive subsequent keys required for secure wireless communications with the mobile network operator.

These as well as other aspects and advantages will become apparent to those of ordinary skill in the art by reading the following detailed description, with reference where appropriate to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Various exemplary embodiments are described herein with reference to the following drawings, wherein like numerals denote like entities.

Figure 1 is a graphical illustration of an exemplary system, where a device communicates data with a mobile network operator and a network in order to mutually authenticate with the network, in accordance with exemplary embodiments;

Figure 2 is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device, a mobile network operator, a network, and a device provider, in accordance with exemplary embodiments;

Figure 3 is an illustration of a device database and a network database with exemplary data stored, in accordance with exemplary embodiments;

Figure 4a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device, a mobile network operator, a network, and a device provider, in accordance with exemplary embodiments;

Figure 4b is an illustration of an exemplary set of cryptographic parameters and a device database, in accordance with exemplary embodiments;

Figure 5a is an illustration of an exemplary system where a network communicates with a plurality of devices and a plurality of device providers, in accordance with exemplary embodiments; and

Figure 5b is a graphical illustration of communication between a device, a plurality of networks, and a device provider in accordance with exemplary embodiments.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS OF THE INVENTION

Figure 1

Figure 1 is a graphical illustration of an exemplary system, where a device communicates data with a mobile network operator and a network in order to mutually authenticate with the network, in accordance with exemplary embodiments. The system 100 can include a device 101,

Petitioner Samsung Ex-1004, 1546

a mobile network operator 102, a network 103, a device provider 104, and certificate issuers 105. A system 100 can use multiple different external certificate issuers over time, depicted as 105-1 through 105-3. The depicted nodes or entities can communicate data 106 over an Internet Protocol (IP) network 107. Although a single device 101, a single mobile network operator 102, a single network 103, and a single device provider 104 are depicted in Figure 1, a system 100 can comprise a plurality of each of the depicted nodes or entities. A system 100 as depicted in Figure 1 can support EAP-TLS authentication for 5G wireless WAN networks, such as supporting EAP-TLS authentication as described in Annex B of TS 133 501 V15.1.0. Other wireless networking technology for device 101 and MNO 102 could be supported as well, such as WiFi.

Device 101 and mobile network operator 102 can utilize a variety of wireless wide area network (WAN) and wireless local area network (LAN) wireless and technologies to communicate data 106 between the nodes, including Low Power Wide Area (LPWA) technology, 3rd Generation Partnership Project (3GPP) technology such as, but not limited to, 3G, 4G Long-Term Evolution (LTE), or 4G LTE Advanced, NarrowBand - Internet of Things (NB-IoT), LTE Cat M, and 5G or subsequent wireless technologies. In addition, the wireless technology used by device 101 and mobile network operator 102 could support or implement wireless LAN technologies such as WiFi and the related series of standards from IEEE 802.11 standards, such as 802.11ac, 802.11 ax, etc. Other examples exist as well for wireless WAN technology and/or wireless LAN technology used by device 101 and mobile network operator 102 without departing from the scope of the present disclosure.

Mobile network operator 102, network 103, and device provider 104 can connect to the IP network 107 and communicate with each other via a wired connection such as, but not limited to, an Ethernet connection, or a fiber optic connection. In other words, mobile network operator 102 can connect to (i) device 101 through wireless technology and (ii) a network 103, a device provider 104, and a certificate issuer 105 using wired technology. IP network 107 could also be a public or private network supporting internet Engineering Task Force (IETF) standards such as, but not limited to, such as, RFC 786 (User Datagram Protocol), RFC 793 (Transmission Control Protocol), and related protocols including IPv6 or IPv4. A public IP network 107 could utilize globally routable IP addresses. A private IP network overlayed on IP network 107 could utilize private IP addresses which could also be referred to as an Intranet. Other possibilities for IP Network 107 and a private network between mobile network operator 102, network 103, and device provider 104 exist as well without departing from the scope of the disclosure.

Mobile network operator (MNO) 102 can include a plurality of radio-frequency (RF) access technologies (RAT) systems for supporting wireless communications with a plurality of devices 101 in a networked manner, including through the use of geographically dispersed antennas, radio

Petitioner Samsung Ex-1004, 1547

nodes, or towels. The RAT systems for MNO 102 can include a radio 102r for communicating with device 101, where radio 102r includes an antenna system and can operate through licensed radio spectrum. In exemplary embodiments, MNO 102 can operate a plurality of radios 102r which are connected to an IP network 107 in a secure manner, including connecting via a private IP network.

In exemplary embodiments, mobile network operator 102 can comprise traditional public land mobile networks providing data and voice services such as AT&T, Verizon, T-Mobile, Sprint, etc, and provide data communications sendees through a variety of radio access technologies. Further, although Figure 1 depicts an MNO 102 as communicating with device 101 with a radio 102r, another entity besides a mobile network operator could perform the function of MNO 102. For embodiments where radio 102r for device 101 uses WiFi technology, then MNO 102 could comprise an entity other than a PLMN, such as a service or network that operates a plurality of WiFi access points. The function of MNO 102 could also he conducted by a large enterprise with a collection of geographically distributed WiFi access points.

Each of mobile network operator (MNO) 102, network 103, and device provider 104 could operate a plurality" of servers in order to support tire communications and connectivity depicted in Figure 1 and additional Figures below. The exemplary data structures, values or numbers, and steps for a MNO 102, network 103, and device provider 104 depicted in Figure 1 and Figures below could be recorded and/or conducted by a collection of servers for each entity. Exemplary" servers for a mobile network operator 102, network 103, or device provider 104 in system 100 can be either different physical computers such as rack-mounted savers, or different logical or virtual servers or instances operating in a "cloud" configuration.

An exemplary server or collection of servers for MNO 102 can comprise a Security Anchor Function (SEAF) 102b, where a MNO 102 can include at least one SEAF 102b for establishing secure and authenticated communications with device 101 through at least one radio 102r. The SEAF 102b could comprise and operate according to the European Technical Standards Institute (ETSI) standard TS 133 501 V15.1.0 from August of 2018 and titled "5G; Security architecture and procedures for 5G System", which is hereby incorporated by reference in its entirety. In exemplary embodiments, the SEAF 102b for MNO 102 can receive an anchor key from network 103 in order to establish secure and authenticated communication with device 101, such as depicted and described for a message 237 below in Figure 2. In exemplary embodiments where Extensible Authentication Protocol - Transport Layer Security (EAP-TLS) in a system 100, then SEAF 102b could operate as specified in section 6.1.1.2 of TS 133 501 V15.1.0, where "the SEAF takes the role of pass-through authenticator."

Petitioner Samsung Ex-1004, 1548

Note that MNO 102 could use other wireless networking technologies besides 5G. Tims, although an SEAT 102b is depicted in Figure 1, a different collection of servers or an entity besides aSEAF 102b could perform the steps for a MNO 102 depicted in Figure 2 below, without departing from the scope of the present invention. In general, the function of an SEAF 102b as depicted and described in connection with Figure 2 below could be conducted by at least one server or process operating within a cloud computing environment, where the sewer or process may not be an SEAF 102b. For example, the sewer or process operating with equivalent functionality as an SEAF 102b in Figure 2 below could (i) receive a partially encrypted Network Access Identifier (NAI) such as a SUCI 101b from device 101 with an encrypted user name and a plaintext realm for the NAI, and (ii) use the plaintext realm to forward the SUCI 101b to network 103, where network 103 uses or is associated with at least a portion of the plaintext realm.

MNO 102 can include a NINO identity 102a, where MNO identity 102a can comprise the combination of a mobile country code and a mobile network code, according to 3GPP standards for 4G networks, 5G networks, etc. As one example, MNO identity 102a for an example network of AT&T could comprise an MCC number of 310 and a MNC number of 410, such that the NINO identity 102a can be 310410. Note that a MNO 102 can operate with multiple MNO identities 102a, and a MNO identity 102a could also comprise a Domain Name Sendee (DNS) name such as an exemplar}·' value of "att.net", and other possibilities exist for a NINO identity 102a without departing from tire scope of the present invention. In addition, a MNO 102 could operate with a plurality/ of MNO identities 102a.

MNO 102 can include a plurality of processors 102p in order to store and record data as well as communicate within a system 100 over an IP network 107 and a radio 102r. Processor 102p can comprise a general purpose processor appropriate for the computational requirements for a MNO 102, and may operate with multiple different processor cores, including field programmable gate arrays (FPGA). A processor 102p can comprise exemplar}·' ARM® based processors or an Intel® based processor such as belonging to the XEQN ® family of processors, and other possibilities exist as well. Processor 102p can utilize a data bus to fetch instructions from memory or storage within a sewer and operate on the instruction. A processor 102p can include components such as registers, accumulators, and logic elements to add, subtract, multiply, and divide numerical values and record the results in memory. In exemplar}' embodiments, at least one processor 102p within MNO 102 can be used with an SEAF 102b to conduct the steps for MNO 102 and send/receive messages for NINO 102 as depicted and described in connection with Figure 2 below. Although riot depicted in Figure 1, MNO 102 can include memory and at least one database in order to establish communications with a plurality of devices 101 through a plurality of radios 102r.

Petitioner Samsung Ex-1004, 1549

Device 101 can be a computing device for sending and receiving data using a radio 101r. Device 101 can take several different embodiments, such as a general purpose personal computer, a mobile phone or mobile handset based on the Android ® or Fuchsia from Google ® or the IOS operating system from Apple ®, a tablet a device with a sensor or actuator for the "Internet of Things", a module for "machine to machine" communications, a device that connects to a wireless Wide Area Network (WAN) operated by MNO 102, a router, and/or a server, and other possibilities exist as well for the embodiments of a device 101 without departing from the scope of the present invention.

The electrical components within device 101 can include a memory 101m, a processor 101p, a radio 101r, and a user interface 101u. Although not depicted in Figure 1, a data bits or a system bus could internally electrically connect the depicted components within a device 101. Additional components to support the operation of device 101 can include a battery to store electrical power, an antenna to transmit and receive RF signals, a sensor to collect data external or internal to the device 101, such as temperature, motion, position, pressure, etc. A device 101 could also include an actuator to convert electrical signals into physical actions, such as a motor for moving components, a relay for opening or closing a circuit, etc.

Memory 101m can comprise combinations of (i) volatile random access memory and nonvolatile memory. The volatile memory can include random access memory (RAM) for relatively fast read and write operations compared to nonvolatile memory, such as SRAM or DRAM. RAM for memory 101m could also include persistent RAM or nonvolatile RAM (NVRAM), such that data in a persistent RAM memory or nonvolatile RAM is stored when power is removed. Nonvolatile memory can include storage memory such as a Hash memory and record or store data when power is removed from device 101. In general, different forms and electrical components for memory 101m can be used without departing from the scope of the present disclosure. Processor 101p can comprise a central processing unit (CPU) or a "system on a chip" and be similar to a processor 102p for a MNO 102 described above, but with reduced capabilities for a device 101 compared to a processor 102p for MNO.

Device 101 may include radio 101r support radio-frequency (RF) communications with networks including a MNO 102 via standards such as GSM, UMTS, mobile WiMax, CDMA, LTE, LTE Advanced, 5G, and/or other mobile-network technologies. In a wireless configuration, the radio 101r may also provide connectivity to local networks such as 802.11 WLAN, Bluetooth, Zigbee, or an IEEE 802.15.4 network, among other possibilities. In exemplary embodiments, a radio 101r is connected to an antenna, which could be either internal to device 101 or external to device 101.

Petitioner Samsung Ex-1004, 1550

Note that device 101 may also optionally include user interface 101u which may include one or more devices for receiving inputs and/or one or more devices for conveying outputs. User interfaces are known in the art and thus user interfaces are not described in detail here. User interface 101u could comprise a touch screen if device 101 operates as a smartphone or mobile phone. Device 101 can optionally omit a user interface 101u, since no user input may be required for many M2M applications such as networked sensors, although a user interface 101u could be included with device 101. LED lights or a display of LEDs could also comprise a user interface 101u.

Device 101 can include a device identity comprising a subscription permanent identifier (SUPI) 101a. The device identity can comprise a string or number to uniquely identify device 101 with MNO 102, network 103, and device provider 104, and other nodes connected to the IP network 107. A device identity of a SUPI 101a can include a network access identifier (NAI) according to IETF RFC 7542, titled "The Network Access Identifier", which is herein incorporated by reference in its entirety. A NAI can consist of different portions, such as a user identity, and a realm. The user identity can comprise a number or a string to uniquely identity device 101 within a realm. A NAI for a SUPI 101a can have many forms, and with several examples for a NAI are included in RFC 7542. Other possibilities exist as well for a NAI as a SUPI 101a for an identity of device 101 without departing from the scope of the present invention. The use of a SUPI 101a in an NAI format as a device identity for device 101 is not required for some exemplary embodiments, and for some embodiments the SUPI 101a could be a single string with an embedded user identity and realm but without the "@" character standard for a NAI.

As depicted in Figure 1, in exemplary embodiments, a SUPI 101a can have a user identity and a realm, where the realm can have a prefix and a suffix. The prefix and the suffix can have multiple portions or more than one portion. In general, (i) a realm suffix can include a base string of a domain name associated with a network 103 (depicted as "network" for SUPI 101a in Figure 1), and (ii) a realm prefix can include a second string associated with a device provider 104 (depicted as "provider" for SUPI 101a). In exemplary embodiments, the realm prefix can include at least a portion of a provider identity 104a and the realm suffix can include at least a portion of a network identity 103a. Note that a realm suffix could also include data or a name for MNO 102, which could include a MNO ID 102a. Other embodiments of a SUPI 101a for a device 101 than that depicted in Figure 1 could include exemplary values of "user@network.provider" (e.g. a prefix first portion of the realm includes a name for the network 103 and a suffix second portion of the realm includes a name for the device provider 104). In other exemplary embodiments, the realm of a SUPI 101a can comprise solely a name or identity for device provider ID 104a, and for these

Petitioner Samsung Ex-1004, 1551

embodrments, MNG 102 and network 103 could have agreements and technical arrangements to route data 106 with fee realm of a provider ID 104a to device provider 104.

In preferred embodiments, a.device 101 with a user identity of"XXXXX" could have be associated wife a device provider wife fee name "Company", where fee company has a commercial agreement and technical agreement with a cloud service such as Amazon Web Sendees to operate the authentication server function (AUSF) 103b, such that devices 101 for the company (e.g. a device provider) could roam and/or be authenticated with a plurality of mobile network operators. The cloud sendee could have commercial and technical agreements with a plurality" of mobile network operators, including MNG 102 depicted in Figure 1. The cloud sendee could have an exemplary network ID 103a of aws.com. For this exemplary embodiment described in tins paragraph, the SUPI 101a for device 101 could comprise an example string of "XXXXX^ompany. aws.com". Note that an identity' ofMNO 102 such as MNO ID 102a may not need to be included in SUPI 101a, and MNO 102 can select network 103 based on the suffix of the realm (e.g. aws.com). In another embodiment, fee MNO 102 identity could be included in the realm, where the MNO ID 102a could be "at.com", and the SUPI 101b for device 101 could be "XXXXX@company.aws.att.com". Other possibilities exist as well without departing from the scope of the present invention.

Although not depicted in Figure 1, a NAI as a SUPI 101a could comprise a string where realm has an identity of the network 102a in the prefix and an identity of the device provider 103a in the suffix. Or, fee realm could comprise a single base string feat is unique for a combination of an identity' of device provider 104a and network 102a (such as a realm of "c0mpan5--aws.com"). Further, in some exemplary embodiments, the realm of a NAI for SUPI 101a could omit an identity device provider 104a and consist only of an identity' for network 103 of network ID 103a (where fee realm could be uniquely associated with a device provider 104). Or, the realm of a NAI for SUPI 101a could omit an MNO identity 102a and consist only of an identity' for network 103 of network ID 104a. In general, the realm of a SUPI 101a can include sufficient identifying information (such as at least a portion of network ID 103a and/or provider ID 104a) in order for data within a message 205 (below in Figure 2) from device 101 to be (i) received by' MNO 102 and forwarded to network 103, and (ii) received by network 103 and forwarded to device provider 104.

A device 101 can also include a subscription concealed identifier (SUCI) 101b. SUCI 101b can include an encrypted identity' of a user name or user identity', depicted as "{user}" in Figure 1 and fee plaintext realm, where exemplary' formats or contents of a realm were depicted and described in the two paragraphs above. A SUCI 101b can be derived from a SUPI 101a using an elliptic curve integrated encryption scheme (ECIES) 101j, which is described in addition detail below. In exemplary' embodiments, device 101 communicates with a MNO 102 using a SUCI 101b

Petitioner Samsung Ex-1004, 1552

as the identity for device 101 instead of the SUM 101a, in order for the user identity of device 101 to be encrypted.

Although "user" is the term depicted in Figure 1 and also for a NAI in RFC 7542 and related 3GPP specifications, the "user" portion of a NAI can comprise a string of digits or numbers or characters to uniquely identify device 101 for a device provider 104 and the "user" portion of a NAI within a SUPI 101a or SUCI 101b does not need to be associated with a user such as a person. In oilier words, a "user" within an identifier such as a NAI or a SUPI 101a can be an identify for a device. The plaintext for the realm in a SUCI 101b can be required is order for MNO 102 and/or network 103 to route data for a SUCI 101b forward (e.g. MNO 102 -> network 103 and network 103-> device provider 104). MNO 102 could use at least a first plaintext prefix with network ID 103a in the realm to select a network 102. In other words, in exemplary embodiments, the realm portion of a NAI for a SUCI 101b can remain plaintext, although the "user" portion of a NAI for a SUCI 101b can be encrypted.

Tire "user" portion of a NAI for a SUPI 101a could comprise a medium access control (MAC) address for a physical interface such as Ethernet or WiFi, an international mobile subscriber identify (IMSI) or international mobile equipment identify (IMEI) with 2G/3G/4G networks, and other possibilities exist as well for the user portion of a NAI as a SUPI 101a without departing from the scope of the present invention. In exemplary embodiments, SUPI 101a can be written to hardware hi device 101 and operate as a globally unique, long-term identity for device 101. Note that a device 101 could use several different SUPI 101a, such as a first SUPI 101a with a first network 103 and a second SUPI 101a with a second network 103, where the realm portion can be different for the first and second SUPI 101a and the different realm portions can include different network IDs 103a for the first and second network 103.

A certificate for device 101 can comprise eert.device 101c and can include the device identify comprising the SUPI 101a. For some embodiments, the device identify within a device certificate eert.device 101c can be a number, string, or value that is different than the SUPI 101a. As one example, a SUPI 101a could be for connecting with a wireless network 102, while a device identify in a cert.device 101c can be for communicating with a device provider 104. A certificate 101c for device 101 can include a device public key corresponding to the device private key SK.Device lOld, a set of cryptographic parameters for the device public key, the Identify of a certificate authority, and a digital signature for the certificate authority. A certificate 101c for device 101 and oilier nodes in a system 100 can comprise a certificate supporting X.509 v3 formats. The cryptographic parameters can specify parameters for both the device public key and also the digital signature from a certificate authority, and two different sets of cryptographic parameters could be used for each of the device public key and the digital signature. Cryptographic parameters

Petitioner Samsung Ex-1004, 1553

in a certificate can specify **algorithms** such as RSA or elliptic **curve** Glyptography (ECC) values, ECC **curve** names, key **lengths** in bits, **signature algorithms,** validity dates, encoding rales, etc. The **signature** of the certificate **authority** for **cert.device** 101c could **be** included iu **device certificate** 101c.

A node of system 100 **in Figure 1,** such as the **authentication** saver **function** (AUSF) 103b can use the **device** certificate cert.device **101c in** order to authenticate device **101** with **MNQ** 102 and network 103. Additional details **regarding** die use a certificate for **device** 101 of cert.device **101c** will be **described** below **in** Figure 2. Note **that** other nodes **in** a system 100 can record and operate with a certificate similar to cert.device 101c, except **with** identities for tire nodes, the public keys for the nodes, and certificate authority digital signatures for the public keys. A device 101 could record an use multiple different device certificates 101c (possibly **with** different device public keys **corresponding** to different device private keys anchor **cryptographic** parameters) **in** a database **101k,** where the device 101 selects the different **device** certificates **101c** based on the **MNO** 102 **and/or** die **network** 102 the device 101 **communicates with.**

A device 101 can record and use a device private key SK.device **101d** (e.g. "secret key"). The private key **can** correspond to the public key in the device certificate cert.device 101c. In **exemplary** embodiments, **device** 101 **internally** generates **a** random **number** for the device private key SK.device 101d and using **cryptographic** algorithms 101s and a set of **cryptographic parameters** 111a and generates the device public key in a certificate IGic. The private key SK.device 101d and the **corresponding** public key can be for the set of cryptographic **parameters within** the certificate **101c,** such as the values for an elliptic **curve** named **curve** or **cryptographic** algorithm **type** 111c in Figure 4b below. **The generation** of the public key for device certificate cert.device **101c** could be (i) during device 101 **manufacturing** or distribution **and** (ii) before device 101 sends **a** SUCI 101b to a MNO 102.

In some **exemplary** embodiments before **device** 101 establishes a radio **connection** with MNO 102 (such as **(hiring** a device configuration step), device **provider** 104 receives the public key **6** r SK.device 101d and signs the public key **with** a device provider secret key such as SK.Provider 104e in order to generate the **device certificate** cert device **101c.** Or, a device certificate **certdevice** 101c could be signed by a different certificate issuer than a certificate issuer operated by device **provider** 104. Note that **cryptographic** algorithms 101s and PKI **keys** for **device** 101 **can** support post-quantum cryptography (PQC) and support code-based, lattice-based, and related **cryptographic** algorithms. Example possible **algorithms** for **parameters** in device certificate cert.device **101c** include **those** described hi "Post-Quantum **Cryptography Standardization" from** Wikipedia dated **March** 12, 2019, which is **herein incorporated** by reference.

Petitioner Samsung Ex-1004, 1554

Device 101 can store or record a public key for the device provider 104 of PK-ECIES.Network 101e in order to conduct ECIES 101j encryption of the user name for a SUCI 101b. The public key can be associated with a set of cryptographic parameters ECIES Parameters 101f for using the public key (such as an ECC curve name), and the cryptographic parameters could be used with ECIES 101j function. In other words, ECIES Parameters 101f can specify a named curve associated with PK-ECIES.Network 101e. The network 103 and/or AUSF 103b can record a private key corresponding to PK-ECIES.Network 101e, where the private key can comprise SK-ECIES.Network 103s as depicted in Figure 1.

The value of PK-ECIES.Network 101e could be written to memory 101m during manufacturing or distribution or configuration of device 101, or securely before the time device 101 sends a SUCI 101b to a wireless network 102. Neste that PK-ECIES.Network 101e can comprise multiple different points on different ECC carves, such that different values of PK-ECIES.Network 101e cau be selected depending on the parameters ECIES Parameters 101f used by ECIES 101j. In other words, PK-ECIES.Network 101e can include a first public key for name curve P-256, a second public key for named curve 25519, where the selection of the first public key or second public key could be based on ECIES Parameters 101f.

MNO parameters 101g cau include a list of numbers or strings for values such as (i) allowed frequencies or frequency bands to scan, (ii) preferred access lists for roaming onto either wireless networks, (iii) criteria for a device 101 to select MNO 102, including in idle mode, (iv) support for emergency services, (v) supported languages or character encoding, (vi) codes to search for in beacons broadcast by a wireless network 102, (vii) parameters for a radio 101r to use when connecting to a wireless network 102, (viii) names or addresses for a server associated with a MNO 102 in order for a device 101 to conduct radio resource connect procedures, etc. A device 101 can record multiple sets of MNO parameters 101g for a plurality of different MNO 102, and a device 101 could select a MNO 102 from a plurality of available MNO 102 for radio 101r based on data within MNO parameters 101g.

Device 101 cau derive an ephemeral public key ePK.ECTES.Device 101hand an associated secret key or private key eSKLECIES-Device101i, using a random number, a key pair generation function, and fie set of ECIES Parameters 101f. Device 101 can utilize a random number generator similar such as hardware random number generator in order to derive fee private key eSK.ECIES.Device 101i and input the random number into the key pair generation function. A key pair generation function could be included in application software running on device 101, and exemplary software for key pair generation algorithm can comprise OpenSSL, oypto-H-, or Mozilla libraries. The resulting ephemeral public key ePK.ECIES.Device 101h can be sent to MNO 102 in a message 205 below in Figure 2, The resulting secret key or private key

Petitioner Samsung Ex-1004, 1555

eSK.ECIES.Device 101i can be input into the encryption function ECIES 10lj in order to convert a SUPI 101a into a SUCI 101b, where the user portion of an NAI for a SUPI 101a can be encrypted in a SUCI 101b. In some embodiments, the realm portion of the NAI in a SUCI 101b can remain plaintext, such feat wireless network 102 can read fee plaintext realm for fee SUCI 101b in order to select and forward fe e SUCI 101b to a network 103 and/or device provider 104.

Device 101 can include an encryption function ECIES 101j in order to convert a SUPI 101a into a SUCI 101b. The encryption function ECIES 10lj can comprise fee identity encryption scheme as described in Annex C: "Protection schemes for concealing the subscription permanent identifier" in ETSI standard TS 133.501 version 15.1.0. In summary, device 101 can conduct an elliptic curve Diffie-Hellman key exchange using private key eSK.ECIES.Device101i and public key PK-ECIES.Network 10le and parameters ECIES Parameters 10lf in order to generate an encryption key and a message authentication code (MAC) key. The user portion of SUPI 101a, or fee user name, can be input into the encryption function ECIES 10lj along with fee derived encryption key and MAC key in order to generate SUCI 101b which contains fee user name as a ciphertext (depicted as "{user}" in Figure 1 and Figure 2 below). A MAC code generated by ECIES 10Ij using fee MAC key can be used to verify message integrity for the SUCI 101b by device provider 104. The realm portion of SUPI 101a can remain as plaintext for SUCI 101b after encryption function ECIES 10lj.

Device 101 can record or store a certificate for device provider 104 comprising Cert.Provider 104c, where the certificate for device provider 104 can be similar to fee certificate for device 101 described above. Device 101 could also record a plurality of different device provider certificates Cert.provider 104c, where the different certificates 104c could use different provider IDs 104a and/or different sets of cryptographic parameters 111a" from parameters 111. The different provider IDs 104a and/or cryptographic parameters 111a" could be used to verify a digital signature 122 within a device provider certificate 104c. An identity' for device provider 104 recorded wife CertProvider 104c can comprise provider ID 104a. Cert.Provider 104c can also record a public key for device provider 104 comprising PKJ¾ovider 104ca.

Cryptographic parameters can he included in the certificate for device provider 104, such as an ECC named curve for PK.Provider 104ca, a time for validity·' of the certificate, and other information normally recorded in an X.509 v3 certificate. Note that in exemplary embodiments, fee certificate CertProvider 104c can be associated wife and generated by device provider 104 and not MNO 102 and network 103. As depicted and described below' in connection with Figure 2, exemplary embodiments contemplate that (i) device 101 can authenticate with network 103 and (ii) MNO 102 can obtain an anchor key for device 101 without device 101 receiving or storing a

Petitioner Samsung Ex-1004, 1556

certificate for MNO 102 and network 103 different than the stored or recorded device provider certificate Ceit.Provkler 104c .

hi this manner, the use of a device 101 in a system 100 can be significantly more flexible than conventional technology, since (i) a device provider 104 can control the device 101 before connecting to MNO 102 or selecting MNO 102 and (ii) device 101 does not need to receive a separate server certificate that chains to (or can be verified through) a previously recorded root certificate CertCI-root 105c. In other words, without the technology described herein, an MNO 102 and network 103 operating an AUSF 103b may not readily guarantee (or without user interaction for device 101) that device 101 can verify a server certificate received from AUSF 103b that can be verified with a recorded root certificate Cert.CI-root 105c.

The certificate Ceit.Provkler 104c can be signed by a certificate authority such as with a device provider certificate issuer certificate 104f, and device 101 can record a drain of certificate authority certificates up to a root certificate Cert.CI-root 105c for a certificate issuer 105. For example, as depicted in Figure 1, device 101 Ceit.Provider 104c could be signed by a certificate issuer for device provide 104 with a certificate comprising Cert.Provider-CI 104f. The public key tor the device provider certificate issuer in Cert.Provider-CI 104f could be signed by die certificate issuer 105-1. The certificates for device provider 104 including Ceit.Provider 104c and the supporting chain of certificates including CerLProvider-CI 104f and root certificate Ceit.CI-root 105c-1 could be written by device provider 104 during distribution or configuration of device 101 or before device 101 is received by an end user of device 101. The certificates for device provider 104 including Cert.Provider 104c and the supporting chain of certificates including Cert.Provider-CI 104f and root certificate Ceit.CI-root 105a could be stored by device 101 before device 101 sends data 106 to a MNO 102 in order to authenticate the device 101 with MNO 102.

For embodiments where the certificate Ceit.Provider 104c is written to device 101 or stored by device 101 before a device 101 connects with a MNO 102, thee a device 101 could also (i) receive CertNetwork 103c in a message from network 103 and (ii) check that tire received Cert.Network 103c can be verified with the device provider public key 104ca in the stored CertProvider 104c. hr this manner. Cert.Network 103c could be authenticated and verified without requiring device 101 to conduct a full set of certificate signature verification steps through parent certificates of CertNetwork 103c up to root certificate Cert.CI-root 105c stored within device 101. Note that when device 101 sends a message 205 below in Figure 2, device 101 may not have full access to the Internet 107 in order to conduct steps and receive messages to verify" a network certificate 103c with parent certificates that are not previously recorded in device 101 before sending the message 205.

Petitioner Samsung Ex-1004, 1557

In other words, without recording Cert.Provider 104c in memory before connecting to a MNO 102, then device 101 could receive a saver/network certificate from AUSF 103b that does not chain to (or be verified through) root certificate Cert.CI-root 105c. which can cause significant problems with conventional technology. The reason is device 101 may not have access to IP network 107 without first authenticating with AUSF 103b and may have no ability" to request a patent certificate from a network certificate such as Cert.Network 103c from AUSF 103b that chains to (or can be verified through) root certificate Cert.Cl-root 105c. Without (i) previously recording Cert.Provider 104c to verify a network certificate Cert.Network 103c and (ii) verifying a received Cert.Provider 104c through a root certificate CertCI-root lG5c-l (or some other certificate previously securely stored in device 101 before connecting with MNO 102), then device 101 may not feasibly be able to verify a received Cert.Network 103c and thus authenticate MNO 102 and network 103.

Device 101 can derive an ephemeral public key ePK.TLS .Device 101n and an associated secret key or private key eSK.TLS.Device 1Oiq, using a random number, a key pair generation function, and the set of cryptographic parameters for a TLS session. Device 101 can utilize a random number generator in order to derive private key eSK.TLS.Deviee 101q and input the random number into the key pair generation function. A key pair generation function could be included in application software running on device 101, and exemplary software for key pair generation algorithm can comprise OpenSSL, crypto-H-, or Mozilla libraries. The resulting ephemeral public key ePK.TLS.Device 101η can be sent to MNO 102 and network 103 in a message 218 below in Figure 2 in a TLS 1.3 "Client Hello" message.

Device 101 could use a set of cryptographic algorithms 101s in order to process and use ePK.TLS.Device 101n and eSK.TLS.Device 1Olq with TLS messages such as a "client hello", a "server hello", and subsequent TLS messages. Private key eSK.TL S.Device 101q and a public key for network 103 of ePK.TLS.Network 103n can be input into an ECDH key exchange algorithm within cryptographic algorithms 101s in order to mutually derive a shared secret key. The shared secret key can be used to generate or derive a symmetric ciphering key and a MAC key. Device 101 could use the PKI keys with cryptographic algorithms 101s and cryptographic parameters ill in order to mutually derive a symmetric ciphering key and a MAC key with network 102, as specified hi IETF RFC 8446 titled "The Transport Layer Security (TLS) Protocol Version 1.3", which is heroin incorporated by reference in its entirety. Note that device 101 and network 103 could also use the PKI keys to mutually derive a symmetric ciphering key and a MAC key using previous versions of TLS as well, such as TLS version 1.2,

Cryptographic algorithms 101s can include the steps and logic for processor 1Oip in device 101 to conduct in order for device 101 to securely communicate with MNO 102 and network 103.

Petitioner Samsung Ex-1004, 1558

Cryptographie algorithms **101s can** include **at** least **symmetric ciphering algorithms, a** random **number generator,** a key pah generation algorithm, digital signature algorithms, **asymmetric-**ciphering **algorithms,** and key exchange **algorithms. Cryptographic algorithms** 101s can also **support post-quantum cryptography,** such as the **exemplary types** 1 i 1c **for** parameters 111 depicted and described in **connection with Figure** 4b below. Cryptographic **algorithms 101s** can also include a key verification step for verifying that a received public key is valid for an **ECC curve. Cryptographic algorithms 101s** can use libraries available **from** example cryptographic suites such as OpenSSL, cry pto++: **BouncyCastie,** or **Mozilla,** and other possibilities exist as well without **departing** from **the** scope of the present **disclosure. Cryptographic** algorithms 101s can use inputs of keys such as public keys, private keys, author **symmetric keys** along with **cryptographic** parameters ill in order to for device 101 to process **cryptographic** data including ciphertext key exchanges, and digital signatures.

Cryptographic **parameters** 111 can specify values or **settings** tor (i) **conducting an** ECDH **or** ECDHE key exchange, (ii) mutually **deriving** a shared secret, (in) mutually deriving a symmetric ciphering key from the **shared** secret, **(iv) using** a **symmetric ciphering algorithm with** the symmetric ciphering key, and **(v)** using a digital signature algorithm. As **contemplated** herein, **cryptographic** parameters 111 may also be referred to as **parameters** 111. Each of device 101, MNO 102, and network 103 and device provider 104 can record at least one compatible subset of parameters 111a, 111a', or 111a'' within a set of cryptographic parameters 111. Parameters 111 can specify" values tor an **elliptic** curve **cryptography** (ECC) **curve** name, key **length,** key formatting (e,g. compressed or **uncompressed),** encoding rales, etc. Parameters 111 are also depicted and described in connection with Figure 4b below.

As contemplated herein, the parameters 111 and **cryptographic** algorithms 101s used with ECC PKI keys and a key exchange in the present **invention** can be compatible and substantially conform with ECC **algorithms** and keys as specified in (i) **the** IETF Request for Comments (RFC) 6090 titled "Fundamental Elliptic Curve Cryptography Algorithms", and (ii) IETF RFC 5915 titled "Elliptic **Curve** Private Key **Structure",** and also subsequent and related versions of these **standards.** Other possibilities exist as well for **cryptographic parameters** 111 without **departing** from the scope of the present invention.

For use of ECC **algorithms,** parameters 111 can specify" elliptic curve names such as, but not **limited** to NIST P-256, seet283kl , **sect283rl, sect409kl , seet409rl** , and other possibilities exist as well. **Further,** elliptic **curves** that do not depend on **curves currently** specified by the National Institute of Standards and **Technology** (NIST) could be utilized as well, such as, but not limited to, **Curve22S19, curve448,** or **FourQ.** Parameters 11i can **specify** domain parameters for nodes in system 100 and other systems herein to calculate values or numbers in a compatible manner, such

Petitioner Samsung Ex-1004, 1559

as common base point G for use with ECC PKI key pairs and a defining equation for an elliptic curve.

Network 103 can comprise a collection of servers and also operate as a cloud service. As depicted for system 100 in Figure 1, network 103 can communicate with device 101, MNO 102, and device provider 104 through IP network 107. Network 103 can include a network identity of network ID 103a, which could comprise a domain name, a name, or a string to uniquely identify network 103 in a system 100 and other systems herein. In exemplary embodiments, at least a portion of network ID 103a is included in the realm of device identify comprising a NAI for SUPI 101a.

Network 103 can include an authentication server function AUSF 103b as specified in ETSI TS 133 501 V15.1.0, as well as subsequent or related versions of ETSI standards. In general, an authentication server function can communicate with a security anchor function SEAF 102b from MNO 102. AUSF 103b can receive a SUCI 101b for device 101 from SEAF 102b, where MNO 102 receives SUCI 101b from device 101 via a wireless network and radio 102r. Additional details for the communication and operation of AUSF 103b for network 103 is provided below in Figure 2. In exemplary embodiments, AUSF 103b and device 101 can support EAP-TLS authentication, where EAP-TLS authentication for device 101 using a wireless network is described in Annex B of ETSI TS 133 501 V15.1.0. AUSF 103b can send and receive messages with device 101 through MNO 102 arid also send and receive messages with device provider 104 through IP network 107. Network 103 can include at least one processor 103p, where processor 103p can operate in a similar manner as a processor 102p described above for MNO 102. Processor 103p could be included in a server and also within AUSF 103b. A network 103 can operate with a plurality of processors 103p in order to perform the steps and communicate the messages depicted and described for a network 103 in subsequent figures herein.

Network 103 can derive an ephemeral public key ePK.TLS.Network 103n and an associated secret key or private key eSK.TLS.Network IG3q, using a random number, a key pair generation function, and the set of cryptographic parameters for a TLS session. Network 103 can utilize a random number generator such as hardware random number generator in order to derive the private key7 eSK.TLS.Network 103q and input the random number into the key pair generation function. A key pair generation function could be included in application software running in network 103. The PKI keys depicted and associated cryptographic algorithms and cryptographic parameters could be operated within AUSF 103b. The cryptographic algorithms and cryptographic parameters operated within AUSF 103b could be equivalent to cryptographic algorithms 101s and cryptographic parameters 111 used by device 101. Exemplary software for key pair generation algorithm can comprise OpenSSL, crypto-H-, or Mozilla libraries.

Petitioner Samsung Ex-1004, 1560

The **resulting ephemeral** public key ePK.TLS.Network **103n** can be sent **to MNO** 102 and device 101 **in** a message 227 below **in Figure** 2 **and other Figures** herein in a TLS 1.3 **"Server** Hello" message as key share **103n.** Network 103 could use a set of **cryptographic algorithms in** order **to** process and use ePK.TLS.Network **103n** and eSK.TLS.Network 103q with TLS messages such as a **"client hello",** a "server hello", and subsequent TLS messages. Private key eSK.TLS.Network **103q** and a public key **for device 101** of ePK.TLS.Device **101η** can be input into an **ECDH** key exchange algorithm in order to mutually derive a **symmetric ciphering key** and a MAC key. **AUSF** 103b could use the PKI keys with **cryptographic** algorithms and **cryptographic** parameters in order **to mutually** derive a symmetric ciphering key and **a** MAC key with device 101, as specified **in** IETF RFC 8446 titled "The Transport Layer Security" (TLS) Protocol **Version** 1.3", which is herein incorporated by reference in its entirety. Note that device 101 and network 103 could also use the PKI keys to mutually derive **a symmetric ciphering** key and a MAC key using **previous** versions of TLS as welt such as TLS version 1.2.

As depicted in Figure 1, **a** network 103 can also operate **a** network database 103d. A network database **103d** could comprise a collection of **servers** and storage memory for storing values associated with a **plurality** of devices in order to **support** die operation **of a** system 100. A network database 103d could store values, strings, or numbers for device 101 such as a SUCI 101b received **from** MNO 102 and a device provider certificate 104c **and/or device** certificate **101c** for device 101 **received from device provider** 104, and other data could be stored in a network database 103d as well. Exemplary- data for an exemplary **network** database 103d is depicted **and** describe **in connection with** Figure 3 below. Data within a network database **103d could** be recorded or stored as network 103 receives data for a device 101 **from** a **MNO** 102 and a device provider **104.** Subsequent communications after receipt of **the** data could use **a** network database **103d in** order to select stored values for communication with device 101.

Network 103 can store or record a secret key for the network 103 of SK-ECIES .Network 103s **in** order to conduct **ECIES 103j** decryption of the user **name in** a received SUCI 101b. The private key can be associated with a set of **cryptographic parameters** ECIES Parameters 101f for **using** the private key (such as an **ECC curve** name), and the **cryptographic** parameters could be **used** with ECIES **103j decryption function.** Note that **a network** 103 could use **(i)** a first **SK-**ECIES .Network 103s **with** a first network ID **103a** with ECIES **103j** decryption **function** and (ii) a second SK-ECIES.Network **103s** with a second network ID 103a with ECIES 103j decryption **function,** where the **different** values for network ID 103a and SK-ECIES .Network 103s could also be stored **within** a network database **103d** in Figure **3** below. In **exemplary** embodiments, a MAC code output by ECIES 103j Is calculated over tire **entire** SUCI **101b, including** an **encrypted** user name for device 101, and thus the integrity or authentication of a received plaintext realm within

Petitioner Samsung Ex-1004, 1561

the SUCI 101b can be verified by comparing the received MAC code for a SUCI 101b and a calculated M AC code output the ECIES 103j

ECIES Parameters **lOlf can specify** a named **curve** associated with private key SK-ECIES.Network **103s.** As described above, the device 101 **can** record **a** public key corresponding to SK-ECIES.Network 103s. where the public key for **device** 101 can comprise **PK-**ECIES.Network **lOle** as depicted in Figure 1. The value of PK-ECIES.Network **lOle** could be generated by Network 103 **using SK-ECIES.Network 103s** and ECIES Parameters **lOlf.** PK-ECIES.Networfc **lOle could provided to** device 101 before **device** 101 sends a SUCI **101b b MNO** 102. such as **during device manufacturing** or device distribution or **during** a secure session prior to device 101 sending a message 205 in Figure 2 below. Different values of SK-ECIES.Network 103s and corresponding ECIES Parameters lOlf can be selected by network 103 using (i) at least a portion of network ID 103a in **a** realm **within** a received SUCI **101b for** device 101 and **(ii)** a network database 103d. SK-ECIES.Network 103s can comprise a first private key for named curve **P-256 and a second public** key for named **curve** 25519. where the selection of the first private key or second private key could be based on ECIES **Parameters 101 f. which** could be associated with **the network ID 103a in a realm received in** a SUCI **101b for device** 101. Note that the transmission of a SUCI **101b by** device 10 1 can also include a specified set of ECIES parameters **lOlf** in order b Identify a named **curve** or parameters associated **with ePK.ECIES.Device lOlh** transmitted by device 101 along with the SUCI 101b. Different devices 101 could include the different network ID 103a names in the SUCI **101b,** such as a first set of devices using art exemplary value of "network-1" **in** the **realm** portion **of a** SUCI **101b and** a second set **of devices using** an exemplary value **of"netwoik-2"** in the realm portion of SUCI 101**b.** In **exemplary** embodiments, the network ID **103a** in **the** reahu portion of a SUCI 101b **can** be used to select **from** network database 103d the SK-ECIES.Network 103s for an ECIES **decryption function** 103j, such that different devices 101 can use different **corresponding PK-ECIES.Network lOle** for a SUPEECIES **encryption function** 1Oij by the different devices. The use of different network ID 103a names and different secret keys SK-ECIES.Network **103s** for network 103 could also be stored **in** a **network** database 103d in Figure 3 below.

Network 103 can include **a decryption** function ECIES **103j** in order to convert a SUCI 10 1b into a SUPI 101a. Tire **decryption function** ECIES **103j** can comprise the identify **decryption** scheme as described **in** Annex C: **"Protection** schemes for **concealing** the subscription **permanent** identifier" in ETSI **standard** TS TS 133.501 version **15.1.0.** In **summary,** device provider can conduct an elliptic **curve** Diffie-Helhnan key **exchange** using **(i)** the selected private key SK-ECIES.Network 103s (where key 103s can be selected based on ID **103a** in a SUCI reahu), (ii) **a** received device public key ePK.ECIES.Device **lOlh** received along **with** SUCI **1Oib**, and **(iii)**

Petitioner Samsung Ex-1004, 1562

parameters ECIES Parameters 101f in order to generate an encryption key and a message authentication code (MAC) key. The user portion of SUCI 101b, or the encrypted user name, can be input into the decryption function ECIES 103j along with the derived encryption key and MAC key in order to generate SUPI 101a which contains the user name as a plaintext. A MAC code generated by ECIES 103j using the MAC key can be used to verify message integrity for fee SUCI 101b by network 103. The realm portion of SUCI 101a can be received as plaintext and may preferably not be altered or changed by ECIES 103j for some exemplar;/ embodiments.

Network 103 can store a network certificate CertNetwork 103c, where (i) the public key in CertNetwork 103c can be digitally signed by a device provider 104 using SK.Provider 104e and (ii) fee corresponding public key for SK.Provider 104e of PK.Provider 104ca can be stored by device 104 before sending a SUCI 101b to NINO 102. A network 103 can also record or store a plurality of network certificates Cert.Network 103c for different network public keys and different parameters, as well as for signature by different device providers 104. CertNetwork 103c can include (i) a certificate issuer identify, which can comprise device provider identify Provider ID 104a, (ii) a server name identifier (SM) 103a for network 103 which can comprise a network identify 103a, (iii) a network public key PK.neiwork 103e, (iv) a set of parameters for the network public key of PK.Parameters .Network Ilia', and (v) a digital signature 122 over at least the network public key, where the digital signature 122 can be generated by device provider 104 using SK.Provider 104e for the PJCProvider 104ca stored by device 101. A network 103 can also store fee secret or private key corresponding to the public key 103e in Cert.Network 103c, and the private key can comprise network private key SKNetwork 103f. The steps for network 103 to obtain the network certificate cert.network 103c are depicted and described in connection wife Figure 2 and subsequent figures below. For some embodiments, network 103 could also store a network certificate 103c before device 101 connects with MNO 102.

Device provider 104 can comprise an entity or set of servers for authenticating or managing plurality of devices 101. Although depicted in Figure 1 and other Figures herein as "device provider 104", an entity such as a device owner, a device manufacturer, or a device distributor could record the data and perforin the functions of a "device provider 104". In general, a device provider 104 can comprise an entity or collection of servers which is the source for prosiding Cert.Provider 104c to device 101 before device 101 connects wife MNO 102, where CertProvider 104c can be used by device 101 b authenticate NINO 102 and wife AUSF 103b below in Figure 2 and subsequent Figures herein. As contemplated herein, a "device provider" may- also be referred to as a "provider". Consequently, a device provider secret key SK.Provider 104e can be referred to as a "provider secret key 104e", a device provider certificate Cert.Provider 104c can be referred to as a "provider certificate 104c", etc.

Petitioner Samsung Ex-1004, 1563

Although depicted as "device provider" in Figure 1 and subsequent Figures herein, the device provider 104 could be a device owner, a device distributor, or a device manufacturer. In general, a device provider has a sufficient level of ownership or control of device 101 before device 101 connects with MNO 102 such that device provider 104 or an entity associated with device provider 104 could record or store fee device identity SUPI 101a, and at least a certificate 101c for device 101 before device 101 connects with MNO 102. For some embodiments, device provider 104 could also be a device user, where the device user operates a plurality of different devices. As one example, device provider 104 could be a large enterprise responsible for hundreds or more devices that are geographically dispersed, and the enterprise may prefer to support the authentication and networking connectivity for device 101 using the steps and messages depicted herein.

Device provider 104 can include a provider identity of provider ID 104a. Provider ID 104a can comprise a name or unique identifier, including a domain name or at least a portion of a realm in a NAI, in order to uniquely identify device provider 104 in a system 100, since a system 100 could include a plurality of different device providers 104. At least a portion of the provider ID 104a can be included in fee SUPI 101a and SUCI 101b, such as within the realm portion of the SUPI 101a and SUCI 101b, as depicted in Figure 1. Although a single provider ID 104a is depicted in Figure 1, a device provider could use a plurality of different related provider ID 104a names or identities, such as "company-1", "company-2", etc. The use of different provider ID 104a names is depicted for in a device database 104d in Figure 3 below.

Device provider 104 can store a plurality of device Identities comprising SUPI 101a for a plurality of devices. Although depicted as external to device database 104d in Figure 1, fee values of SUPI 101a could be recorded within device database 104d along with fee unique user name portion for each of the devices. Note that for a given device 101 in a system 100, a user name may not be globally unique, but fee combination of user name and realm for a SUPI 101a cart be globally unique in preferred embodiments. For some embodiments, device provider 104 can also store the concealed identity" SUCI 101b received from a network 103 and originated by device 101 in order to conduct an ECIES decryption step to convert the SUCI 101b into a SUCI 101a for device 101. In other words, although a network 103 is depicted as conducting an ECIES decryption function 103j, for some embodiments a device provider 104 could conduct fee decryption function and convert a SUCI 101b into a SUPI 101a for a device 101.

Device provider 104 can also store or record a plurality of device certificates 101c corresponding to a plurality of devices in a system 100. The device provider 104 could be responsible for initial configuration of devices 101 or receive initial configuration data for devices 101, where the initial configuration of device 101 can include the generation of a device certificate

Petitioner Samsung Ex-1004, 1564

101c, and device provider 104 could store the device certificate 101c for each of the devices 101. Although depicted as external to a device database 101d, device provider 104 could store a plurality of device certificates in a device database 101d. By device provider 104 storing a device certificate 101c and providing the device certificate 101c to a network 103, the authentication by network 103 of device 101 can be simplified, as described below in Figure 2 (since network 103 could trust device provider 104 and receive the device certificate 101c from device provider 104, where network 103 may not trust device 101 when a certificate 101c is received from device 101).

A certificate for device provider 104 can comprise Cert.Provider 104c and can include the device provider identity comprising provider ID 104a. A certificate 104c for device provider 104 can include a public key corresponding to the device provider private key SK.Provider 104e, a set of cryptographic parameters for the device provider public key, the identity of a certificate authority', and a digital signature for the certificate authority, as well as a specified digital signature algorithm used for the digital signature. A certificate 104c for device provider 104 and oilier nodes in a system 100 can comprise a certificate supporting X.509 v3 formats, and other formats or structures for a device provider certificate Cert.Provider 104c and other certificates as contemplated herein are possible as well. The cryptographic parameters can specify parameters for both the device provider public key and also the digital signature from a certificate authority. Cryptographic parameters in a certificate can specify algorithms such as RSA or elliptic curve cryptography (ECC) values, ECC curve names, key lengths in bits, signature algorithms, validity dates, encoding rales, etc. Cryptographic parameters in a device provider certificate 104c and other certificates herein can also support algorithms for post-quantum cryptography, as depicted and described for parameters 111 in Figure 4b below. The signature of the certificate authority or certificate issuer for Cert.Provider 104c could be included in device provider certificate 104c.

Note that a device provider 104 could use a plurality of different device provider certificates 104c, and the device provider certificate 104c for a device could be selected from a device database 104d for a device using the SUPI 101a output from decryption function ECIES 103j and received Horn a network 103. In some exemplary embodiments, the device provider certificate Cert.Provider 104c is also recorded in a device 101 as depicted in Figure 1 before device 101 sends a SUCI 101b to MNO 102. In other exemplary embodiments, device 101 may not record Cert.Provider 104c before device 101 sends a SUCI 101b to MNO 102, and for these embodiments Cert.Provider 104c could be received an verified by device 101 in a TLS handshake, where device 101 could verify' Cert.Provider 104c using another "parent" certificate for Cert.Provider 104c such as certificate issuer 105 root certificate Cert.CI-root 105c-1.

A device provider 104 can record and use a device provider private key' SKihovider 104e (e.g. "secret key"). The private key can (i) correspond to the public key in the device provider

Petitioner Samsung Ex-1004, 1565

certificate Cert.Provider 104c and (ii) be used by device provider 104 in order to generate fee digital signature for Cert.Network 103c. In exemplary embodiments, device provider 104 (i) internally generates a random number for fee device provider private key SK.Provider 104e and using cryptographic algorithms 104s, (ii) generates fee device provider public key in a Cert.Provider 104c and fee set of cryptographic parameters within fee certificate Ceit.Provider 104c, such as fee values for an elliptic curve named curve and specification of digital signature algorithms. The generation of fee public key for device provider certificate Cert.Provider 104c could be before or during device 101 manufacturing or distribution and before device 101 sends a SUCI 101b to a MNO 102. In this manner, a device provider certificate Ceit.Provider 104c could be written to device 101 before device 101 sends a SUCI 101b to MNO 102.

In some exemplary embodiments, device provider 104 operates a parent or 'intermediate" certificate authority or certificate issuer in order to generate and sign Cert.Provider 104c. The public key for SK.Provider 104e in Cert.Provider 104c could be signed by a device provider secret key SK.Provider-CI (not shown in Figure 1) for Cert.Provider-CI 104f stored by a device 101. Note that device 101 could store the certificate issuer certificate Cert.Provider-CI 104f for the device provider certificate CeitProvider 104c, as depicted in Figure 1. The public key for CertProvider-CI 104f could be signed by SK.jRoot 105b-1 torn a certificate issuer 105-1, where the certificate authority/issuer signature in Cert.Provider-CI 104f can be verified with root certificate Cert.Cl-root 105c-1. Although a single level of a certificate issuer between device provider certificate Cert.Provider 104c and root certificate Cert.CI-root 105c-1 is depicted in Figure 1 within a device 101, a device provider 104 could operate multiple levels of intermediate certificates between device provider certificate Cert.Provider 104c and root certificate Ceit.CI-root 105c-1. As depicted in Figure 1, a system 100 could include multiple root certificate issuers 105, and device provider 104 could record multiple certificates Cert.Provider-CI 104f or Cert.Provider 104c signed by different root certificate issuers 105 wife different SK.Root 105b from fee different certificate issuers 105.

Cryptographic algorithms 104s can include the steps and logic for processor 104p in device provider 104 to conduct in order for device provider 104 to securely communicate with network 103. Cryptographic algorithms can include symmetric ciphering algorithms, a random number generator, a key pair generation algorithm, digital signature algorithms, asymmetric ciphering algorithms, and key exchange algorithms. Cryptographic algorithms can also include a key verification step for verifying feat a received public key is valid for an ECC curve. Cryptographic algorithms 104s can use libraries available from example cryptographic suites such as OpenSSL, crypto-H-, BouncyCastle, or Moziila, and other possibilities exist as well without departing from the scope of fee present disclosure. Cryptographic algorithms 104s can us inputs of keys such as public keys, private keys, and/or symmetric keys along with cryptographic parameters 104x in

Petitioner Samsung Ex-1004, 1566

order to for device provider 104 to process cryptographic data including ciphertext, key exchanges, and digital signatures.

Cryptographic parameters 111 can specify values or settings for (i) conducting an ECDH or ECDHE key exchange to obtain or derive a shared secret, (ii) mutually deriving a symmetric ciphering key using at least the derived shared secret, (iii) using a symmetric ciphering algorithm with the symmetric ciphering key, and (iv) using a digital signature algorithm. Cryptographic algorithms 104s for device provider 104 can correspond or be equivalent to cryptographic algorithms M1s for device 101, and cryptographic parameters 111 for device provider 104 can correspond or be equivalent to cryptographic parameters 111 for device 101.

Device provider 104 cun operate a plurality of servers with at least one processor 104p. A processor 104p can comprise a central processing unit (CPU) or a "system on a chip" and be similar to a processor 102p for a MNG 102 described above. Device provider user interface 104u can comprise a web-based interface for administrators of device provider 104 to manage the function and operation of device provider 104 as well as manage a plurality of devices 101. User interface 104n can support the addition of new devices in a system 100, the deletion or revocation of existing devices in a system 100, etc. User interface 104u can also allow changes in values and tables for a plurality of devices 101 in a device database 104d, where a device database 104d is depicted and described hi connection with Figure 3 below.

Device database 104d can record a plurality of values for a plurality of devices in order for device provider 104 to manage or operate communications with device 101 through a network 103. Device database 104d can record for each device exemplary values of SUPI 101a, device certificate 101c, the provider ID 104a used by device 101, a device provider certificate 104c used by a device 101, etc. Additional details regarding the data stored in a device database 104d are depicted and described in connection with Figure 3 below.

As depicted in Figure 1, device 103 can also store or operate a database 101k, which could also comprise a certificate database. The database 101k can include a plurality of certificates for device 101. An exemplary first certificate 101c-1 for device 101 could include a first device public key 101ca-1, a first set of parameters 111a-1 for the device public key, and a digital signature 122-1 over at least the first device public key 101ca-1, where the digital signature can be verified by a certificate such as Cert.Piovider 104d or a certificate issuer certificate such as Cert.Provider-CI 104f. Along with the first certificate can he stored a first device private key 101d-1 for the first device public key 101ca-1. Device 101 could include the same device identify of SUPI 101a in multiple certificates in database 101k. In addition, device 101 could include different values of SUPI 101a in different certificates in database 101k.

Petitioner Samsung Ex-1004, 1567

An exemplary second certificate 101c-2 for device 101 could include a second device public key 101ca-2, a second set of parameters 111a-2 for the device public key, and a digital signature 122-2 over at least the second device public key 101ca-2, where the digital signature can be verified by a certificate such as Cert.Provider 104d or a certificate issuer certificate such as Cert.Provider-CI 104f. Along with the second certificate can be stored a second device private key 101d-2 for the second device public key 101ca-2. In exemplary embodiments, a device 101 can select a device certificate 101c from a database 101k using the parameters 111a. Although not depicted for a database 101k, the certificates could also include identities for issuers, such as provider ID 104a, network ID 103a, etc,, and device 101 could select a device certificate 101c from the database 101k using the certificate issuer identity. For some embodiments, the device could select the device certificate 101c based on both the parameters 111a and the certificate issuer identity. The parameters 111a and certificate issuer identity' could be received in extensions within a "server hello" message 227 as depicted in Figure 2 below.

Although Figure 1 depicts MNO 102, network 103, and device provider 104 as separate nodes or entities, in some exemplary embodiments different entities or function of the nodes could be combined. In one embodiment, a MNO 102 and a network 103 could be combined, such that a MNO 102 controls or operates the network 103. In this embodiment, then a MNO 102 would operate both a SEAF 102b and an AUSF 103b. Or, a network 103 could control or operate a NINO 102, such that the network 103 controls or operates both a SEAT 102b and an AUSF 103b. The operation of a network 103 and device provider 104 could also be combined.

In addition, although a MNO 102 is depicted in Figure 1, the wireless networking technology, operated by MNO 102 with radios 102r could use WiFi based technology, (e.g. based on 802.11 standards) and a collection of geographically distributed WiFi access points. The WiFi access points could be controlled by network 103 and a security' anchor function 102b could be omitted. In another embodiment, the function and operation of a network 103 and a device provider 104 could combined, such that either (i) network 103 records the data and operates the function of a device provider 104, or (ii) a device provider 104 operates an AUSF 103b. Other possibilities exist as well for the combination of the data and functions for a MNO 102, network 103, and device provider 104 without departing from the scope of the present invention.

Figure 2

Figure 2 is a simplified message flow diagram illustrating an exemplary system wife exemplary' data sent and received by a device, a mobile network operator, a network, and a device provider, in accordance with exemplary embodiments. System 200 can include a device 101, a mobile network operator (NINO) 102, a network 103, and a device provider 104. The nodes or entities within system 200 were also depicted and described in connection with Figure 1 above.

Petitioner Samsung Ex-1004, 1568

wliere **Figure 2** depicts exemplary steps for the nodes and exemplary message flows between the nodes. Although a single device 101, **MNO** 102, network 103, and **device provider** 104 are depicted in a.system 200, a system 200 could include a **plurality** of each of the **depicted** nodes connected **via.** an IP network 107. In addition, data recorded for device 101 and **device provider** 104 depicted and described **above** in Figure 1 (other than ephemeral PKI keys) can be recorded in the device 101 and device provider 104 depicted in **Figure** 2 before the steps and message flows are conducted in Figure 2.

In order to support **authentication** of **devices** 101 **with** network 103, **a MNO** 102 could establish a secure session 201 with network 103. Secure session 201 could comprise **establishing** a **secure** communications link between the two **servers** using protocols such as **TLS,** IPSec, a virtual private network **(VPN),** a **secure** shell (SSH), or similar networking, transport, or application layer **technologies** hi order to **establish secure** communications **between** MNO 102 and network 103. Secure session 201 can utilize **certificates** for each of MNO 102 and **network** 103 hi order to provide **mutual** authentication and mutual key **derivation** for a **symmetric** encryption key in **secure** session 201. **Secure** session 201 can also be conducted over IP **network** 107, **although the secure** session 201 could also be established or conducted through a private network. Other possibilities exist as well for **establishing a** secure session 201 between MNO 102 and **network** 103 without departing from the scope of tire present disclosure.

In order **to** support authentication **of** devices 101 with **network** 103, a network 103 and device provider 104 could establish a **secure** session 202. **Secure** session 202 can be equivalent to **secure** session 201 described in tire **paragraph** above. Secure session 202 could use **certificates from both** network 103 and device provider 104 for mutual **authentication** and the secure derivation of **encryption** keys for the secure session. Note that subsequent data transmitted from device provider 104 to network 103 **through** the secure session 202 can be **trusted by network** 103 on the basis **of** the trust for secure session 202. For example, network 103 and AUSF **103b** could receive **a** device certificate **101c for** a device 101, where the device certificate **101c** may not be authenticated with a certificate chain and a root certificate recorded or used by network 103. However, network 103 could still **trust** the device **certificate** 101**c** since the device certificate 101c could also be received **through** secure session 202, where **secure** session 202 can be **mutually** authenticated using **certificates** for both network 103 and device provider 104. In some exemplary **embodiments,** both **secure** session 201 and **secure** session 202 are established using TLS 1.3.

At step 203, device 101 can power on and load firmware and **configuration** data in order to connect with wireless **network** 102. A step 203 can also include firmware or software in device 101 loading a RF module or radio 101r within device with **an** operating configuration such as MNO **parameters** 101g **in** Figure **1** (such as frequencies to **use,** wireless **technology to** use such as 5G or

Petitioner Samsung Ex-1004, 1569

WiFi 6, etc.}. Step 203 can include device 101 scanning for available mobile networks 102 and then selecting a preferred mobile network. In exemplary embodiments, device 101 scans for mobile network identities broadcast in system information blocks, there the mobile network identity' such as mobile network ID 102a matches the value for a mobile network in a realm for a SUPI 101a recorded by device 101.

In other words, device 101 could include multiple SUPI 101a with different realms for different networks 103 and device providers 104, and search for a MNO 102 in an RF beacon or signal that matches a realm for one of the multiple SUPI 101a. For other embodiments, the MNO ID 102 may not be included in a portion of a realm within a SUPI 101a, and a device 101 could select the MNO 102 using MNO parameters 101g. At the conclusion of a step 203, device 101 can select and store a SUPI 101a to use with a MNO 102. A step 203 can also include device 101 selecting MNO parameters 101g based on observed and available MNO 102 for a radio 101r in device 101. The selection of a SUPI 101a for device 101 h1 a step 203 can include the selection of (i) a device provider 104 provider ID 104a tor a realm of foe SUPI 101a and/or (ii) a selection of network identity' network ID 103a for a realm of SUPI I01a. The values for PK-ECIES.Network 101e and ECIES Parameters 101f can be selected in a step 203 based on the provider ID 104a for the SUPI 101a. As one example a SUPI 101a selected in a step 203 could comprise an exemplary' value of "XXXXX@conipany.aws.com", where XXXXX is the user name, "company" is the provider ID 104a and "aws.com" is the network ID 103a. Other possibilities for the selection of a SUPI 101a by device 101 from a plurality of SLIP! 101a stored by device 101 in a step 203 are possible as well, without departing from the scope of the present disclosure.

At step 204, device 101 can conduct a radio resource connection (RRC) request with MNO 102 using a radio 101r and the selected NINO parameters 101g. The RRC can transition a radio 101r in device 101 from an idle state to an active state. In the active state, device 101 can transmit messages or data to MNO 102. A step 204 can also include device 101 generating an ephemeral PKI key pair of ePK.EQES.Deviee 101h and eSK.ECIES.Device iOli At step IOlj' device 101 can conduct the ECIES IOlj encryption function with secret key' eSK.ECIESDevicel01i and public key PK-EOES.Network 101e in order to derive a symmetric ciphering key and an MAC key. The user name portion of a SUPI 101a can be input into the ECIES IOlj along with the symmetric key and tire MAC key' in order to output a first portion the SUCI 101b comprising an encrypted user name for tire device 101. .

Tire realm portion of a SUPI 101a can be input into the ECIES IOlj along with the MAC key in order to output a MAC code covering the realm portion the SUCI 101b, such that the realm portion can be verified using the MAC code. Or, a single MAC code for both the first portion of the SUCI 101b and the realm portion of the SUCI 101b could be generated or calculated in a step

Petitioner Samsung Ex-1004, 1570

204. As depicted in Figure 2, the combination of an encrypted user name, selected provider ID 104a for a realm, and/or die selected network ID 103a for a realm can comprise a value 204a. The value 204a can also include a MAC code using the MAC key from FOES 101j encryption function, such that a device provider 104 receiving and decrypting the value 204a comprising at least the SUCI 101b and die MAC code can verify fee integrity of fee value 204a.

Device 101 can then send a message 205 to the selected MNO 102 through (i) a wireless network operated by MNO 102 and (ii) the radio resource connection from a step 203, where the message 205 includes at least the public key ePK.ECIES.Deviee 101i and fee value 204a. MNO 102 and SEAF 102b can receive the message 205 and conduct a step 206 in order to process the message 205. At step 206, SEAF 102b can (i) read the plaintext realm for SUCI 101b in value 204a, and (ii) select die network 103 using fee network ID 103a in a realm for SUCI 101b. In a preferred exemplary embodiment as depicted in Figure 2, the suffix of the realm can comprise the network ID 103a, but in other embodiments the prefix of the realm could comprise the network ID 103a. Or, the entire reahn in a SUCI 101b received by MNO 102 could comprise the network ID 103a

Note that MNO 102 and network 103 could have prior commercial and technical agreements for MNO 102 to forward device 101 authentication data to network 103 and AUSF 103b. If MNO 102 has no commercial relationship wife network 103, or secure session 201 could not be established, or NINO 102 cannot forward fee data in message 205, then NINO 102 can send device 101 an error message wife an error code explaining the failure. Device 101 could then take corrective steps, such as selecting a different NINO 102 or different network 103 via a different SUPI 101a for device 101 in order to send a message 205 to the different MNO 102. Or, device 101 could select a different SUPI 101a in a step 204 again and send a new SUCI 101b with a different SUPI 101a to the same MNO 102 in order to conduct authentication with the MNO 102 and network 103.

MNO 102 can then send network 103 and AUSF 103b a message 207 through secure session 201, where (i) message 207 includes ePK.ECIES.Device lOlh and fee value 204a, and (ii) the value 204a includes SUCI 101b. The AUSF 103b for network 103 can receive the message 207. At step 208, the AUSF 103b can read and process the plaintext data in the realm for SUCI 101b in value 204a. At step 208, the AUSF 103b can select fee device provider 104 using the device provider ID 104a in the realm for SUCI 101b. In a preferred exemplary embodiment as depicted in Figure 2, the prefix of the realm can comprise the provider ID 104a.

Note that network 103 and device provider 104 could have prior commercial and technical agreements for network 103 to forward device 101 authentication data to device provider 104. If network 103 has no commercial relationship with device provider 104, or secure session 202 could

Petitioner Samsung Ex-1004, 1571

not be established, or network 103 cannot forward the data in message 207, then network 103 can seed MNO 102 an error message with an error code explaining the failure. MNO 102 could then send device 101 an error code explaining the failure, such that device 101 could take collective steps, such as selecting a different MNO 102 or different network 103 in order to send a message 205 to the different MNO 102. Upon a failure, a device 101 could also conduct a subsequent step 204 in order to select a different SUPI 101a and generate a new SUCI 101b for the new, different SUPI 101a to the same MNO 102 or a different MNO 102. Upon successful receipt of a SUCI 101b, a step 208 can include AUSF 103b storing SUCI 101b in a network database 103d for subsequent use in identifying device 101, such as when receiving a "client hello" message 218 below with a SUCI 101b added to message 218 by SEAF 102b.

Network 103 can then conduct a series of steps in order to process the message 207. At step 209, network 103 can read the plaintext network ID 103a in SUCI 101b and use network database 103d in order to select (i) secret key SK-ECIES .Network 103s and (ii) ECIES Parameters 101f. Within a network database 103d, a network ID 103a can he uniquely associated with secret key SK-ECIES .Network 103s and (ii) ECIES Parameters !Olf. Network 103 can then conduct a verification of the received ePK.ECIES.Device 10fh using tire ECIES Parameters i01f The verification step can comprise verifying that the received public key is on the curve specified by ECIES Parameters 101f, the public key value does not comprise tire "point at infinity", etc.

Network 103 can then conduct a step 103j' using ECIES decryption function 103j in order to convert the ciphertext user name in SUCI 101b into plaintext. ECIES 103j was depicted and described in connection with Figure 1 above, A step 103j' can also include deriving a MAC key and verifying a MAC code within value 204a in order to check the message integrity" of SUCI 101b. In csther words, device provider can securely- store a SUPI 101a for device 101 after step 103j' in a network database 103d. For some exemplary embodiments, a SUCI 101b could omit an encrypted user portion of the NAT, and the SUCI 101b in a message 205 and message 207 can comprise only the realm portion of the NAI. The MNO 102 can use the realm portion of the NAI, such as the network ID 103a in the realm, to select the network 103 and AUSF 103b. In this maimer, when the user is omitted from SUCI 101b, AUSF 103b can still send a TLS start command 215 to device 101.

Network 103 and AUSF 103b can then send a message 210 to device provider 104 through secure session 202, where message 210 can include (i) MNO ID 102a, and (ii) the value SUPI 101a. Note that the inclusion of NINO ID 102a can be useful for device provider 104 in a message 210, since (i) device 101 could connect with multiple different MNO 102, and (ii) network 103 could support multiple different MNO 102, and (iii) device provider 104 may need to know which MNO 102 is used by device 101 for billing or administrative purposes. In other words, without including

Petitioner Samsung Ex-1004, 1572

MNO ID 102a in message 210 then device provider 104 may not know which MNO 102 that device 101 attempts to connect with.

Although an ECIES deception function 103j is depicted in Figure 2 as being conducted by a network 103, in some exemplary embodiment a device provider 104 could conduct the ECIES decryption function 103j and network 103 would send device provider 104 a message 210 with the SUd 101b for a device 101. For these embodiments, then device provider 104 could store the values for secret key SK-ECIES.Network 103b and parameters ECIES Parameters 101f, as well as the ECIES decryption function 103j. In other words, although Figure 1 depicts values for secret key SK-ECIES.Netwoik 103b and parameters ECIES Parameters IGlf, as well as the ECIES decryption function 103j as within a network 103, a device provider 104 could store the data and perform a step 103j' in some exemplary embodiments. Device provider 104 could conduct tire step 103j' (instead of network 103 as depicted in Figure 2) to decrypt tire encrypted user portion within a SUCI 101b received from MNO 102 in order to generate a SUPI 101a for device 101. For the embodiment described in this paragraph, then a message 210 could include a SUCI 101b instead of a SUPI 101a, and device provider 104 could conduct the steps of 209 and 103j' instead of a network 103. For the embodiment described In tills paragraph, then the depicted key of "PK-ECIES.Network 101e" stored by device 101 can be associated with a device provider 103 instead of a network 103.

Device provider 104 can conduct a step 211 with the plaintext user name to query a device database 104d for data pertaining to device 101, and a step 211 can include verifying that device 101 with the SUPI 101a is authorized to connect with MNO 102 and/or network 103. In a step 211, device provider 104 can determine tire MNO 102 that device 101 attempts to connect with based on the MNO ID 102a received in a message 210 above. If device 101 is not authorized to connect with MNO 102, then device provider 104 could send an error message to network 103 (for forwarding to device 101) and not proceed to the additional steps depicted in Figure 2.

At step 212, device provider 104 can use the username from SUPI 101a from message 210 to query device database 1044 for the device certificate certdevice 101c and also the device provider certificate certprovider 104c recorded by device 101, as depicted and described above in connection with Figure 1.Note that a device certificate cert.device 101c can include the SUPI 101a as the device identity within die certificate. For embodiments where device 101 stores a plurality of certificates 101c, then device provider 104 could select the plurality of device certificates 101c from a device database 104d in a step 212, and device provider 104 could subsequently send network 103 the plurality of selected device certificates 101c.

For some exemplary embodiments, a step 212 could omit the selection of either device certificate 101c or device provider certificate 104c, and either certificate could be omitted in a

Petitioner Samsung Ex-1004, 1573

subsequent message such as message 213. For embodiments where a user name was omited from a message 205, then device provider 104 could receive a provider ID 104a from the realm for device 101 in a message 210 and select a device provider certificate certprovider 104c for the provider ID 104a in a step 212 and omit the selection of a device certificate certdevice 101c. Embodiments where a user name is omitted from a message 205 (or has a "null" value, etc.) are also depicted and described in connection with Figure 4a below.

A step 212 can also include device provider 104 collecting an OSCP stapling 212a through root certificate Cert.CI-iOot 105c for certificates cert.device 101c and certprovider 104c. OSCP stapling is described in the TLS 1.3 RFC 8446, and OSCP stapling 212a provides an updated signature from parent certificate issuers that the certificate has not been revoked. hi exemplary embodiments, OSCP stapling 212a is provided for certprovider 104c, such that device 101 could verify the OSCP stapling 212a using the certificate chain for certprovider 104c recorded by device 101. OSCP stapling 212a can be important for device 101 because device 101 may not have connectivity to IP network 107 without successfully conducting an EAP-TLS authentication with MNO 102, but connectivity to IP network 107 would normally he required to check certificate revocation lists without OSCP stapling 212a. In subsequent steps a network public key PKmetwork 103e can be in a network certificate verified using the device provider public key PKprovider 104ca, and both device 101 and network 103 can use OSCP stapling 212a of certprovider 104c in order to verify that certprovider 104c has not been revoked. In other words, without OSCP stapling 212a from a step 212, device 101 may not feasibly be able to check for certificate revocation for the device provider certificate certprovider 104c selected by device provider 104 in a step 212. Device 101 can receive certprovider 104c in a subsequent "server hello" message in Figure 2 along with OSCP stapling 212a.

Device provider 104 can then send network 103 and AUSF 103b a message 213 through secure session 202, where message 213 can include tire selected device provider certificate CertProvider 104c from a step 212, device certificate CertDeviee 101c (possibly w/ SUPI 101a) from a step 212, and OSCP stapling 212a for device provider certificate 104c and/or device certificate 101c. A SUPI 101a for device 101 can be in a device certificate CertDeviee 101c. For embodiments where the user name was omitted from a message 205 (but a realm with provider ID 104a associated with provider certificate certprovider 103c was in message 205), then message 213 can omit a device certificate certdeviee 104c. Network 103 and AUSF 103b can receive the message 213 and conduct a step 214 to store (i) device provider certificate Cert.Provider 104c and (ii) device certificate CertDeviee 101c for the SUPI 101a and (in) SUCI 101b used by device 101 in a network database 103d as depicted in Figure 3 below. In exemplary embodiments

Petitioner Samsung Ex-1004, 1574

Cert.Provider 104c includes a subset of cryptographic parameters 111a' and CerCDevke 101c includes a subset of cryptographic parameters 111a'.

Network 103 could operate a network database 103d similar to device database 104d, where an exemplary network database 103d is depicted in Figure 3 below. In a step 214, network 103 could store received certificates such as device provider certificate Cert.Provider 104c and device certificate Cert.Device 101c for a plurality of devices 101 along with the SUCI 101b for devices 101 received in messages 207 and SUFI 101a for devices 101 derived in a step 103j'. AUSF 103b can determine the SUPI 101a for device 101 based on the SUPI 101a within a device certificate 101c, and both the values for SUPI 101a and string or file for device certificate 101c could be stored in a network database 103d as depicted in Figure 2 below. Or, in another exemplary embodiment the SUPI 101a could also be sent external to a device certificate in a message 213. Network 103 and AUSF 103b can then use a network database 103d to select device provider certificate Cert.Provider 104c and device certificate Cert.Device 101c in subsequent steps, such as a step 219a using the SUPI 101a or SUCI 101b for a device 101. In a step 214, network 103 could also verify that a SUPI 101a received with a device certificate 101c in a message 213 matches the SUPI 101a derived in a step 103j' for a SUCI 101b received in a message 205.

In addition, although message 213 with Cert.Provider 104c and Cert.Device 101c is depicted as being sent by device provider 104 after the receipt of a message 210, device provider 104 could send CertProvider 104c and Cert.Device 101c to network 103 before a message 210 and after the establishment of secure session 202. Device provider 104 could send Cert.Provider 104c and Cert.Device 101c for a plurality" of different devices 101. By sending message 213 as depicted in Figure 2, then OSCP stapling 212a can be more current or more "fresh" for device 101. In addition, a network 103 could record a plurality of device provider certificates CertProvider 104c in a network database 103d before network 103 receives a message 213, and in this embodiment then a message 213 could include an identifier of a device provider certificate CertProvider 104c (such as a certificate identity') instead of the full certificate as a *.pem file or equivalent. Further, a network 103 could record a plurality of device certificates Ceit.Devke 104c in a network database 103d before network 103 receives a message 213, and in this embodiment then a message 213 could include an identifier of a device certificate Cert.Device 101c (such as a certificate identity') instead of the fill! certificate as a *.pem file or equivalent.

Further, although device provider certificate Cert.Provider 104c and device certificate Cert.Device 101c are depicted in Figure 2 as being sent in the same message 213, a message 213 could be sent in parts, such as the data being transmitted via a physical interface in device provider 104 separately. A device provider certificate CertProvider 104c or device certificate Cert.Device 101c could be selected by device provider at a later point in time than depicted in Figure 2, such as

Petitioner Samsung Ex-1004, 1575

device provider 104 sending device provider certificate Cert.Provider 104c and/or device certificate Cert.Devke 101c after the receipt of a message 220 from network 103. A step 211 to select the device provider certificate Cert.Provider 104c or device certificate Cert.Device 101c could be conducted after device provider 104 receives a message 220. Oilier possibilities for the sequence and timing for a device provider 104 to send a network 103 device provider certificate Cert.Provider 104c or device certificate Cert.Device 101c are possible as well without departing from the scope of the present disclosure.

Note that network 103 and AUSF 103b can trust CerLProvider 104c and Cert.Device 101c, since the certificates can be delivered through the secure session 202. Or, in a step 214 network 103 could conduct a certificate verification of Cert.Provider 104c and Cert.Device 101c through a root certificate, such as a Cert.Cl-root 105c. Note that Cert.Provider 104c and Cert.Devke 101c do not have to be verified through a root certificate stored by network 103, since the certificates are received through secure session 202, where secure session 202 can be established with other certificates for device provider 104 (and the oilier certificates for a secure session 202 could be verified through a different root certificate than the Cert.Cl-root 105c recorded by device 101 as depicted in Figure 1). A step 214 can also include AUSF 103b verifying that AUSF 103b supports the cryptographic parameters 111a' and/or 111a'* in the certificates received in a message 213. In other words, AUSF 103b can use a step 214 to verify that AUSF 103b can conduct a TLS 1.3 session using the parameters and certificates for Cert.Provider 104c and Cert.Device 101c received in a message 213.

At step 103q' AUSF 103b can generate an ephemeral private key for a TLS session with device 101 comprising eSK.TLS.Network 103p and the parameters within Cert.Provider 104c. In other words, AUSF 103b can read the named elliptic curve and key length from CerLProvider 104c received in message 213 and generate a private key for the named curve and key length in a step 103q'. The use and function of an ephemeral secret key eSK.TLS.Network 103q was also described above in Figure 1. AUSF 103b can generate the corresponding public key ePK.TLS.Network 103n. A step 103q' could also be conducted after receiving a "client hello" message 218. AUSF 103b can then send device 101 an "EAP-TLS Start" message 215, which is also depicted as step 6 in Figure B.2.1.1-1 of 3GPP TS 133.501 version 15.1.0. Note that message 215 is sent to NINO 102 and MNO 102 can transmit message 215 through the radio resource connection established by device 101 in step 204 above. Where MNO 102 functions as a "pass through" entity for messages between network 103 and device 101, then tire vertical line for MNO 102 in Figure 2 is omitted.

Device 101 can receive the "EAP-TLS" start message 215 can conduct a series of steps in order to generate the "client hello" message. Although TLS version 1.3 according to IETF RFC 8446 is depicted in Figure 2, subsequent and related version of a transport layer security protocol

Petitioner Samsung Ex-1004, 1576

couid be used as well between device 101 and an AUSF 103b operating in network 103. In a step 216, device 101 using processor 101p can select parameters for a subsequent "client hello" message 218 using the cryptographic parameters in the stored device provider certificate 104c associated with a device provider ID 104a used in the realm of a SUCI 101b transmitted by device 101 in a message 205. In tins manner, the set of cryptographic parameters for a "client hello" message 218 and a "server hello" message 227 can be reasonably assured to be compatible before the "client hello" message 218 is transmitted by device 101. In other words, in exemplary embodiments, a device 101 can store a device provider certificate 104c before device 101 sends a message 205, including storing the device provider certificate 104c dining a configuration of device 101 or a secure session for device 101 before device 101 sends a message 205.

The selected parameters in a step 216 can include at least a digital signature algorithm such as the elliptic curve digital signature algorithm as supported by TLS version 1.3 (or another TLS version for messages 218 and 227) and also a named elliptic curve supported by the TLS version in subsequent messages 218 and 227. hi exemplary embodiments, the set of cryptographic parameters selected from stored device provider certificate Cert.Provider 104c can be a subset of cryptographic parameters 111 for device 101. Or, in some exemplary embodiments device 101 could select the entire set or a larger subset of cryptographic parameters 111 in a step 216 in order to generate a "client hello" message 218. The selected set of cryptographic parameters for device 101 in a step 216 can include data for extensions 217a below. Device 101 could also use cryptographic algorithms 101s with the selected parameters from a step 216 in order to generate a "client hello" message 218. Device 101 could also store first list of cryptographic algorithms 101s and a second list of cryptographic parameters 111 for device 101 and generate an extensions 217a using the first list and tire second list

hi a step 101q', device 101 can generate device ephemeral private key eSK.TLS.Deviee 101q and tire corresponding device ephemeral public key ePK.TLS.De\ice 101n using the selected set of cryptographic parameters from a step 216 above. The generation of ePK.TLS.Deviee 101n and eSK/TLS.Device 101q were also previously described above for device 101 in Figure 1. Note that the ephemeral private and public key for device 101 and the ephemeral private and public key for network 103 can support post-quantum cryptographic algorithms, as specified by cryptographic algorithms and cryptographic parameters stored in the two nodes. At the conclusion of a step 101q', device 101 can conduct a step 217 in order to complete the processing or generation of the "client hello" message 218. The "client hello" message 218 generated in a step 217 can include a random number generated by device 101 using cryptographic algorithms 101s, a list of cipher suites supported by device 101 such as a symmetric ciphering algorithm and a cipher block chaining mode, the identity or data for a key derivation function including a hash function, and the list of

Petitioner Samsung Ex-1004, 1577

supported **extensions 217a** according to section 4.2 of IETF RFC 8446. In **exemplary embodiments,** the data for a "client **hello"** message 218 **can** be generated in a step 217 «sing the set of **cryptographic algorithms** 101s **in device** 101 **and parameters** 111a" **from** device provider certificate **Cert.Provider** 104c.

Device 101 can process extensions 217a, where the extensions **217a** can **either** (i) omit a certificate_authorities extension or **(ii)** include at. least **a certificate authority** specified by cert.provider 104c as stored **by** device 101 such as provider ID 104a as a **supported** certificate authority, or iin) the identity for the root certificate issuer 105-2, where device 101 includes a root **certificate** cert.root **lQ5c-l** as depicted **in** Figure 1 above. For some exemplary embodiments, a device **101 can** verify **a subsequent** certificate received **from AUSF** 103b and network 103 as cert.provider 104c by **simply** comparing if the received **certprovider** 104c Horn AUSF 103b matches the cert.provider 104c stored in memory 101m, as depicted in Figure 1 above. The previous sentence pertains to verifying a received cert.provider 104c from AUSF 103b is trusted, and a device 101 **would** use the **public** key in cert provider 104c in order to verify a **network** certificate cerfnetwork 103c from network 103.

In other words, the digital **signature in** a **"CertificateVerify"** segment **within** a **TLS handshake** with MNO 102 **and** AUSF 103b could be verified **by** a network certificate cert.network **103c,** and **a** digital **signature** in cert.network 103c could be **verified** by **the public key in** cert.provider **104c.** In this manner the verification through a certificate authority chain could be omitted in exemplary embodiments, thereby **simplifying mutual** authentication for device **101** and AUSF **103b/MNO** 102. Note **that** device 101 may not have access to IP network 107 other **than** sending messages **to** SEAF 102b, and **thus** device 101 may not have the **ability** to verify a server **certificate** In the form of a network certificate 103c from AUSF **103b (different than device provider** certificate 104c) that does not verify **through** an existing certificate or certificate authority **previously** stored in device 101, if the received **certificate** from AUSF 103b does not have a chain terminating with a stored root certificate 105a.

Note feat CertProvrder **104c and** device certificate Cert.DevieeiOlc can be stored by device 101 before device 101 sends "client hello" message 218, and also **in** a manner under the **control** of device **provider** 104 **(including** during **any** of device **configuration, device manufacturing,** or **device** distribution), and thus device provider 104 can take steps when processing **device** provider certificate Cet .Provider 104c and device certificate Cert.Device **101** c **b ensure the** specified **cryptographic** parameters **within** the certificates **are** compatible. Device **provider** certificate CerfProvider 104c and device certificate Cert.DevieeiOlc can have **compatible** or equivalent sets of **cryptographic** parameters (such as **specifying the** same digital **signature algorithm** and **mutually** supported or equivalent named elliptic **curves), and** in this **manner** device

Petitioner Samsung Ex-1004, 1578

101 and AUSF 103b can be reasonably assured that (i) the negotiation and use of cryptographic parameters in TLS messages such as messages 218 and 227 will be compatible and (ii) the negotiation of a TLS handshake in messages 218 and 227 will be successful.

Device provider can select or specify for device 101 cryptographic parameters including extensions in section 4.2 of TLS version 1.3 for cryptographic algorithms lOis (such as during a device configuration step to load device 101 with the data depicted and described in Figure 1). The selected or specified cryptographic parameters can be compatible with device provider certificate Cert.Provider 104c and device certificate Cert.Device 101c. It this manner, a TLS session can be reasonably assured to successfully he negotiated between device 101 and AUSF 103b before the series of steps and message flows depicted in Figure 2, including a message 218, a message 227, and a message 233.

In other words, through the use of (i) device provider certificate Cert.Provider 104c for verifying a network certificate cert.nefwork 103c for AUSF 103b (ii) cryptographic parameters 111a" supporting device provider certificate Cert.Provider 104c for AUSF 103b, a TLS session can be reasonably assured to be successful. This contrasts with conventional technology, where if AUSF 103b uses a network certificate that is not verified using CertProvider 104c stored in device 101 then the negotiation of the TLS session is more likely to fail from both (i) incompatible sets of cryptographic parameters or cryptographic algorithms, and (ii) the network certificate for AUSF 103b may not be able to be verified through a certificate chain in device 101 up to a stored root certificate Cert.Cl-root 105c,-1 and (iii) device 101 may not have access o IP network 107 in order to obtain additional certificates that could verify' a server certificate or network certificate eert.network 103c through a stored root certificate.

If a server or network certificate selected by AUSF 103b for use in a TLS session is verified by a certificate that is (i) not previously securely received and stored by device 101, and (ii) or verified through a certificate chain with a different root certificate than Cert.CI-root 105c stored by device 101, then device 101 may not be able to verify' the sever certificate selected by AUSF 103b (different than CerLProvrder 104c stored by device 101). Tire potential issues for AUSF 103b to select a server certificate verified by a different certificate than device provider certificate Cert.Provider 104c stored by device 101 are compounded by the likely scenarios where (i) hundreds of different AUSF 103b may be used globally with different cryptographic parameters and cryptographic algorithms used for the server certificate and also (ii) server certificates with different verifying certificate chains up to different root certificates.

The steps and message flows depicted in Figure 2 and subsequent figures herein, along with supporting data recorded as depicted in Figure 1, provide a solution to the significant number of issues that can arise when an AUSF 103b uses a server certificate verified by a certificate different

Petitioner Samsung Ex-1004, 1579

than device provider certificate Cert.Provider 104c. In summary, to solve the many potential issues and problems of an AUSF 103b using a server certificate verified by a certificate different than the device provider certificate Cert.Prowler 104c stored by device 101, (i) the device 101 can store the device provider certificate CertProvider 104c before sending a SUCI 101b in a message 205, (ii) the AUSF 103b can conduct the subsequent series of steps to securely obtain a server certificate (also referred to herein as a "network certificate") which can be verified by the device provider certificate Cert.Provider 104c from a device provider 104 in a message 213 and stored in device 101, and (in) the device 101 can use the device provider certificate Cert.Provider 104c to verify a saver certificate Cert.Network 103c when conducting a TLS session with device 101 as depicted h1 Figure 2.

Device 101 can then send the TLS "client hello" message 218 to MNO 102 and through SEAF 102b, where MNO 102 forwards the "client hello" message 218 to network 103 and AUSF 103b. The "client hello" message 218 through MNO 102 is also depicted as message 8 and 9 in Figure B.2.1.1-1 of 3GPP TS 133.501 version 15.1.0. A "client hello" message 218 can include a key share value of ePK.TLS.Deviee 10In, which is depicted in Figure 2. A "client hello" message 218 can include additional data for a "client hello" message as described in section 4.1.2 of IETF RFC 8446, including the data processed by device 101 in 216 and 217, such as selected extensions 217a. As depicted in Figure 2, message 218 can include supported certificate authorities 105-1 and 104a, an OSCP stapling request 108 for a server certificate from AUSF 103b, the supported parameters Ilia for device 101, and a server name indication (SNI) 103a. In exemplary embodiments, the SNI 103a value in a message 218 is also the network ID 103a in a realm for a SUCI 101b or SUP! 101a.

Note that MNO 102 and SEAF 102b can include a NausfyUEAuthentieation _Authenticate Request value with the message 218a, such that AUSF 103b can determine which device 101 the "client hello" message 218 is associated with. In exemplary embodiments and also as supported by 3GPP TS 133.501 version 15.1.0, the Naa¾f_UEAuthentication_Authenticate Request received with message 218a by AUSF 103b can include the SUCI 101b. Message 218a from MNO 102 to network 103 can comprise message 218 from device 101 to MNO 102. As depicted in Figure 2, the SEAF 102b can include the SUCI 101b with the "client hello" in message 218a sent to AUSF 103b that Includes the "client hello" message 218 from device 101.

AUSF 103b can receive message 218a and conduct a series of steps to process the "client hello". AUSF 103b can conduct a verification 219a of the received ePK.TLS.Device 101η using the named curve specified by cryptographic parameters within the "client hello" message 218 within extensions 217a. The verification step can comprise verifying that the received public key is on the curve specified by the received cryptographic parameters, the public key value does not

Petitioner Samsung Ex-1004, 1580

comprise the "point at infinity", etc. The verification step 219a can comprise the verification step within section 4.2.8.2 of IETF RFC 8446.

AUSF 103b or network 103 can then conduct a step 219b in order to select at least PK.Network 103e and PK.Params.Network Ilia/ &rau a network database 103d using SUCI 101b received along with the "client hello" from SEAF 102b. Thus, although device 101 may submit a plurality of different parameters 111a in a message 218, in exemplary embodiments network 103 and AUSF 103b can select a PK.network 103d for parameters 111a' from tire parameters 111a' in a device provider certificate 104c received in message 213. As noted above in a step 214 for AUSF 103b, the AUSF 103b can previously store device provider certificate Cert.Provider 104c, device certificate Cert.Device 101c, SUPI 101a, and SUCI 101b for device 101 in network database 103d after receiving the message 213 above from device provider 104.

A step 219b can also comprise AUSF 103b also selecting the device certificate Cert.Device 101c from a network database 103d using the SUCI 101b received along with the "client hello" from SEAF 102b, although the device certificate Cert.Devke 101c could he selected at a later time, such as after the receipt of a message 233 (such as if message 205 included a "null" value for a user identity of device 1010). An AUSF 103b can use a step 219b to select a device provider certificate 104c from a network database 103d using the SUCI 101b from message 218a for conducting an EAP-TLS authentication. The selected device provider certificate 104c from a step 219b can comprise the certificate for verifying a server certificate Cert.Network 103c for AUSF 103b to use when conducting an EAP-TLS authentication with device 101, such as the generated CerfNetwork 103c comprising the server certificate below in a "server hello" within message 227.

In exemplary embodiments, the provider ID 104a within the device provider certificate 104c selected from a step 219b can include the identity of a certificate authority' for network 103 to trse in extensions 219d for the subsequent "server hello" message 227 described below. In other words, both network 103 and device 101 can select a device provider certificate 104c with the certificate authority' identity' or name of provider ID 104a in order to successfully mutually authenticate. In tins manner, device 101 and network 103 can he reasonably assured that (i) device 101 can verify the network certificate 103c selected by network 103 using the provider ID 104a as a certificate authority in extensions 217a, and (ii) network 103 can verify the device certificate 101c selected by' tire device 101 using the provider ID 104a as a certificate authority in extensions 219d.

At step 219b, AUSF 103b can also further process the "client hello" by taking the steps specified in IETF RFC 8446, such as generating a saver random number, selecting extensions 219d according to section 4.2 of RFC 8446 based on a subset of extensions 217a received in the "client hello" message 218, generating a certificate request 219e according to section 4.3.2 of RFC

Petitioner Samsung Ex-1004, 1581