

(51) International Patent Classification:
G06F 21/60 (2013.01) H04L 9/40 (2022.01)

(21) International Application Number:
PCT/US2024/035530

(22) International Filing Date:
26 June 2024 (26.06.2024)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
18/344,349 29 June 2023 (29.06.2023) US

(71) Applicant: AMAZON TECHNOLOGIES, INC.
[US/US]; PO Box 81226, Seattle, Washington 98108-1226 (US).

(72) Inventors: NOLTON, David M.; Ashburn, Virginia 20146 (US). TROCHE, Jose; Gaithersburg, Maryland 20877 (US). WEATHERLY, Richard M.; Vienna, Virginia 22180 (US). BOLLING, Robert Hill; North Chesterfield, Virginia 23235 (US).

(74) Agent: LEEGE, Brian F. et al.; Nicholson DeVos Webster & Elliott LLP, 99 Almaden Blvd., Suite 575, San Jose, California 95113 (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, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, 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, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, 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, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:
— with international search report (Art. 21(3))

(54) Title: CRYPTOGRAPHIC MESSAGING DEVICE

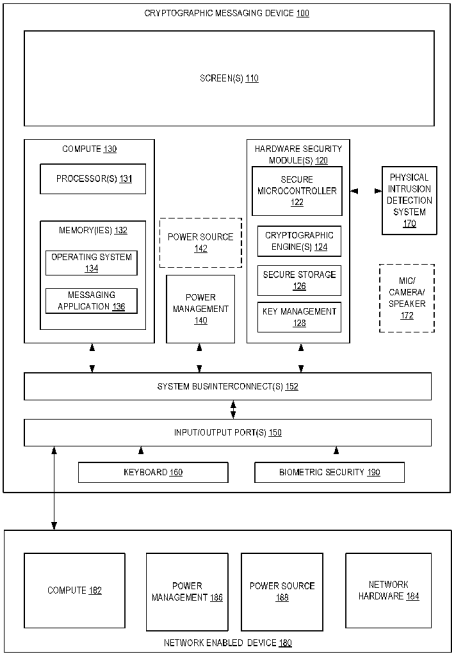


FIG. 1

(57) Abstract: Techniques for cryptographic messaging are described. In some examples, cryptographic messaging is enabled on a device that at least includes an input/output port configured to receive an encrypted message from a coupled external device; a hardware security module (HSM) configured to decrypt the encrypted message, wherein the HSM is to include storage to store at least one private key to be used to decrypt the encrypted message; and a screen to display contents of the decrypted message.

CRYPTOGRAPHIC MESSAGING DEVICE

BACKGROUND

[0001] Customer enterprises working in sensitive domains experience a lack of visibility, delay, and team stress when responding to unscheduled, time-critical issues. For example, teams wishing to discuss the details of a sensitive operational problem may be required to do so in a physical location. Waiting for team members to arrive at the location delays issue resolution. Because some details cannot be shared over public networks due to security issues, it is hard to determine who has relevant insights on the problem at hand. Consequently, often more team members are asked to be present than is necessary.

[0002] Delay and wasted effort would be avoided if leaders had the ability to exchange a few lines of sensitive information with the team before determining who needs to be in the location for in-depth collaboration and who can contribute remotely. Informed decisions about who needs to travel and who contributes remotely optimizes team effectiveness and enhances team member satisfaction.

[0003] Distributed teams have embraced cellphone-based text messaging as an effective collaboration tool. Unfortunately, security sensitive customers, however, do not trust cellphone hardware, system software, nor applications thereof.

BRIEF DESCRIPTION OF DRAWINGS

[0004] Various examples in accordance with the present disclosure will be described with reference to the drawings, in which:

[0005] FIG. 1 illustrates examples of a cryptographic messaging device.

[0006] FIG. 2 illustrates examples of a screen of a cryptographic messaging device.

[0007] FIG. 3 illustrates examples of using cryptographic messaging devices.

[0008] FIG. 4 illustrates examples of using cryptographic messaging devices.

[0009] FIG. 5 illustrates examples of a provider network and device messaging service thereof.

[0010] FIG. 6 is a flow diagram illustrating operations of a method for using a cryptographic device according to some examples.

[0011] FIG. 7 is a flow diagram illustrating operations of a method for enrolling a cryptographic messaging device according to some examples.

[0012] FIG. 8 is a flow diagram illustrating operations of a method for generating a message to be received according to some examples.

[0013] FIG. 9 is a flow diagram illustrating operations of a method for handling a message according to some examples.

[0014] FIG. 10 illustrates an example provider network environment according to some examples.

[0015] FIG. 11 is a block diagram of an example provider network that provides a storage service and a hardware virtualization service to customers according to some examples.

[0016] FIG. 12 is a block diagram illustrating an example computer system that can be used in some examples.

DETAILED DESCRIPTION

[0017] The present disclosure relates to methods, apparatus, systems, and non-transitory computer-readable storage media for secure communications. In some examples, a cellphone (or other internet access device) is minimally used to relay secure communications to/from a coupled cryptographic communications device.

[0018] Existing solutions to sending and receiving cryptographic communications generally rely on software applications running on general purpose machines (phones, tablets, personal computers, etc.). For example, a popular way to have cryptographic communications is to use an encrypted messaging program installed and running on a smartphone. Typically, these applications make their cryptographic libraries available in open source and provide client and server implementations. These resources will guide the device design and encourage trust by security accreditors. Unfortunately, these software solutions still require running on a device that is not, by default, secure at all times. That is messages are input in the clear into the software which the operating system of the device has access to. If there is malware or other malicious software on that device these messages may be intercepted and made available to a bad actor.

[0019] Examples detailed herein are for a cryptographic communications device that is isolated from its means of external communication. That is the device is to be coupled (e.g., using a cable) to a cellphone (or other internet access device) and the networking capabilities of the cellphone (or other internet access device) are to be used to transmit messages that are encrypted before they are provided to the cellphone (or other internet access device). As such, the operating system or

other software on the cellphone (or other internet access device) never has access to any unencrypted data of the cryptographic communications device.

[0020] The cryptographic communications device allows a user to compose, display, send, and/or receive cryptographic messages (the message content may be text, audio, video, image, a combination thereof, etc.) For example, the cryptographic communications device can take in clear text from its own dedicated keyboard (hardware or virtual), encrypt the clear text as a part of an encrypted message, and use the cellphone's networking capabilities to transmit the encrypted message. Similarly, encrypted messages are received via the cellphone's networking capabilities, decrypted by the cryptographic communications device, and then presented on a display of the cryptographic communications device. Clear text messages never pass through the cellphone, nor are clear text messages transmitted or received via public networks. The cryptographic communications device uses pair-wise, end-to-end, cryptographic technology. In some examples, encrypted audio, image, and/or audio/video including messages are supported. Similar to text messages, these messages are not passed through the cellphone in the clear.

[0021] In some examples, the cryptographic communications device is attached to the back of a cellphone, in the form of a protective case, and provides an independent physical interface on which to compose and display text messages. The cryptographic communications device contains a cryptographic capability to create secure channels between pairs of communicating users. The cryptographic communications device obtain electrical power and/or internet access and nothing else. The connection may be made by a cable such as a universal serial bus (USB) or Lightning® cable.

[0022] In some examples, each cryptographic communications device is registered with a service (e.g., a provider network service) to register the device and/or users. Users may be assigned to one or more groups if segregation of user communication is desired. Groups can be used, for example, to enforce isolation or to bind communication to specific classification levels such as Confidential or Secret. Cryptographic material is used to irrevocably associate a user's name, biometric characteristic, and/or a set of authorized groups with each cryptographic communications device.

[0023] FIG. 1 illustrates examples of a cryptographic messaging device. The cryptographic messaging device 100 allows for a use to send and receive encrypted and signed messages through insecure mobile phones, networks, devices, servers, and services. Only the receiving device is able to decrypt and verify the signature of the message which protects the integrity and secrecy of the message's contents.

[0024] The cryptographic messaging device 100 includes compute resources 130 such as a one or more processors 131 to execute a messaging application 136 running on an operating system 134 stored in memor(ies) 132. The messaging application 136 utilizes one or more hardware security modules (HSMs) 120 to send and receive encrypted and signed messages. In some examples, the messaging application 136 is stored in secure storage 126 of the HSM 120 when not in use. In some examples, the messaging application 136 is executed in a secure enclave. In some examples, messages handled by the messaging application 136 are ephemeral and once read are erased. In some examples, messages are manually deletable.

[0025] Cryptographic operations are executed inside the HSM 120 which uses and protects a private key used to sign messages. A public key of a recipient is used to generate a shared key that is used to encrypt signed messages. In some examples, one or more cryptographic engines 124 generate keys. Examples of cryptographic ciphers that are supported may include but are not limited to: Elliptic Curve (ECC), ECDSA, ECDH, AES-256, SHA256, SHA3 (KECCAK), and/or post-quantum encryption algorithms. In some examples, the , one or more cryptographic engines 124 utilize a physical unclonable function (PUF). In some examples, public keys are provided offline (e.g., not using a messaging service).

[0026] The HSM 120 generates and stores keys (e.g., private and shared keys) in secure storage 126 in a tamper resistant manner using key management 128. In some examples, the generated keys are never exposed outside of the HSM.

[0027] In some examples, a secure microcontroller 122 is used to supervise device multifactor identity/authentication and/or physical security, run the key management 128, and/or isolate the secure elements (key management 128, secure storage 126) from the compute resources 130.

[0028] In some examples, the HSM 120 is used to monitor for symptoms of physical tampering using a physical intrusion detection system 170 which may do one or more of: monitor power quality to detects anomalies such as brown-out events, detect shock and orientation change events using one or more accelerometers, and/or use perimeter integrity circuits to detect breaks in a defined wire loops/mesh. In some examples, biometric security 190 such as a fingerprint reader is used to access the cryptographic messaging device 100.

[0029] The cryptographic messaging device 100 is locked down such that it cannot communicate wirelessly (via Wi-Fi®, Bluetooth®, etc.) and can only attach via cable to its host device. However, when the message leaves the cryptographic messaging device 100, it is already encrypted and travels encrypted all the time until it reaches a destination device, which contains the only private key able to decrypt the message. Additionally, the cryptographic messaging

device 100 locks itself if its operating system 134 or applications such as messaging application 136 is modified or tampered with.

[0030] In some examples, the cryptographic messaging device 100 includes one or more screen(s) 110 to allow a user to use the messaging application 136 and/or other applications. The one or more screen(s) 110 are a touch screen in some examples. In some examples, the one or more screen(s) 110 provide a virtual keyboard for input. In some examples, a physical keyboard 160 is used for input.

[0031] In some examples, the cryptographic messaging device 100 includes one or more of a microphone, camera (still and/or video), and/or a speaker. The microphone and/or camera may be used as input for the messaging application 136.

[0032] In some examples, the cryptographic messaging device 100 includes a power management component 140 to control power to the compute resources 130, HSM 120, etc. In some examples, power is obtained using one or more input/output (I/O) port(s) 150 when the cryptographic messaging device 100 is coupled to a network enabled device 180. That is the cryptographic messaging device 100 may not have its own power source 142 and may get power from an external power source 188. I/O port(s) 150 may be one or more of universal serial bus (USB) ports, Lightning ® ports, etc.

[0033] In some examples, one or more of a system bus and/or interconnect(s) 152 are used to couple various aspects of the cryptographic messaging device 100.

[0034] In some examples, a network enabled device 180 provides power (via power manager 186 and power source 188) and/or network access (via networking hardware 184 such as cellular or satellite communication hardware). Note that the network can be a network other than the internet. In some examples, compute resources 182 includes one or more processors, memory, and an OS. In some examples, the OS controls network access and that aspect of the OS is accessed by the cryptographic messaging device 100. In other examples, the cryptographic messaging device 100 bypasses the OS to use the networking hardware 184.

[0035] FIG. 2 illustrates examples of a screen of a cryptographic messaging device such as cryptographic messaging device 100. As shown, a graphical user interface (GUI) 201 allows for a user to have a “conversation” 211 with another party that also has a cryptographic messaging device. In this example, the conversation is between Alice and Bob. Each of the messages in this conversation are encrypted on the user’s cryptographic messaging device and then sent to the other

party to be decrypted and displayed. In some examples, a virtual keyboard 221 allows for text to be entered, files to be shared, etc.

[0036] FIG. 3 illustrates examples of using cryptographic messaging devices. As shown, a cryptographic messaging device 100(A) (user “Alice”) sends an encrypted message via a coupled to a network enabled device 180(A) to cryptographic messaging device 100(B) (user “Bob”) and cryptographic messaging device 100(C) (user “Caleb”) through network enabled device 180(B) and network enabled device 180(C) respectively. In this example, the encrypted message is sent directly without an intermediate messaging service. In some examples, Elliptic Curve Diffie-Hellman techniques are used to create unique, secure channels between all pairs of communicants. That is Alice has a completely separate and independently derived key set for her exchanges between Bob and Caleb. Bob and Caleb decrypt the messages at least in part using their internally stored key(s).

[0037] FIG. 4 illustrates examples of using cryptographic messaging devices. As shown, a cryptographic messaging device 100(A) (user “Alice”) sends an encrypted message via a coupled to a network enabled device 180(A) to cryptographic messaging device 100(B) (user “Bob”) and cryptographic messaging device 100(C) (user “Caleb”) through network enabled device 180(B) and network enabled device 180(C) respectively. In this example, the encrypted message is sent through an intermediate messaging service. In some examples, Elliptic Curve Diffie-Hellman (ECDH) techniques are used to create unique, secure channels between all pairs of communicants. That is Alice has a completely separate and independently derived key set for her exchanges between Bob and Caleb. Bob and Caleb decrypt the messages at least in part using their internally stored key(s).

[0038] In some examples, messages are sent to/from the provider network 400 using hypertext transfer protocol secure (HTTPS). In some examples, messages are sent using MQ Telemetry Transport (MQTT). MQTT uses a publish/subscribe communication model where the client that sends (publishes) a message is decoupled from the receiver (subscriber) device via a third party. When the connection from a subscribing client to a broker is broken, then the broker will buffer messages and push them out to the subscriber when it is back online. When the connection from the publishing client to the broker is disconnected without notice, then the broker can close the connection and send subscribers a cached message with instructions from the publisher. An MQTT broker acts as a go-between for the clients who are sending messages and the subscribers who are receiving those messages.

[0039] In some examples, messages are sent/received using a different transfer protocol such as Constrained Application Protocol (CoAP), Advanced Message Queuing Protocol (AMQP), Simple/Streaming Text Oriented Messaging Protocol (STOMP), Mosquitto, Simple Media Control Protocol (SMCP), etc.

[0040] In some examples, the provider network 400 includes a device messaging service to handle messages. The provider network 400 (or, “cloud” provider network) provides users with the ability to use one or more of a variety of types of computing-related resources such as compute resources (e.g., executing virtual machine (VM) instances and/or containers, executing batch jobs, executing code without provisioning servers), data/storage resources (e.g., object storage, block-level storage, data archival storage, databases and database tables, etc.), network-related resources (e.g., configuring virtual networks including groups of compute resources, content delivery networks (CDNs), Domain Name Service (DNS)), application resources (e.g., databases, application build/deployment services), access policies or roles, identity policies or roles, machine images, routers and other data processing resources, etc. These and other computing resources can be provided as services, such as a hardware virtualization service that can execute compute instances, a storage service that can store data objects, etc. The users (or “customers”) of provider networks 400 can use one or more user accounts that are associated with a customer account, though these terms can be used somewhat interchangeably depending upon the context of use. Users can interact with a provider network 400 across one or more intermediate networks (e.g., the internet) via one or more interface(s), such as through use of application programming interface (API) calls, via a console implemented as a website or application, etc. An API refers to an interface and/or communication protocol between a client and a server, such that if the client makes a request in a predefined format, the client should receive a response in a specific format or initiate a defined action. In the cloud provider network context, APIs provide a gateway for customers to access cloud infrastructure by allowing customers to obtain data from or cause actions within the cloud provider network, enabling the development of applications that interact with resources and services hosted in the cloud provider network. APIs can also enable different services of the cloud provider network to exchange data with one another. The interface(s) can be part of, or serve as a front-end to, a control plane of the provider network 400 that includes “backend” services supporting and enabling the services that can be more directly offered to customers.

[0041] As described herein, one type of service that a provider network may provide may be referred to as a “managed compute service” that executes code or provides computing resources for its users in a managed configuration. Examples of managed compute services include, for

example, an on-demand code execution service, a hardware virtualization service, a container service, or the like.

[0042] An on-demand code execution service (referred to in various examples as a function compute service, functions service, cloud functions service, functions as a service, or serverless computing service) can enable users of the provider network 400 to execute their code on cloud resources without having to select or manage the underlying hardware resources used to execute the code. For example, a user can use an on-demand code execution service by uploading their code and use one or more APIs to request that the service identify, provision, and manage any resources required to run the code. Thus, in various examples, a “serverless” function can include code provided by a user or other entity – such as the provider network itself – that can be executed on demand. Serverless functions can be maintained within the provider network by an on-demand code execution service and can be associated with a particular user or account or can be generally accessible to multiple users/accounts. A serverless function can be associated with a Uniform Resource Locator (URL), Uniform Resource Identifier (URI), or other reference, which can be used to invoke the serverless function. A serverless function can be executed by a compute resource, such as a virtual machine, container, etc., when triggered or invoked. In some examples, a serverless function can be invoked through an application programming interface (API) call or a specially formatted HyperText Transport Protocol (HTTP) request message. Accordingly, users can define serverless functions that can be executed on demand, without requiring the user to maintain dedicated infrastructure to execute the serverless function. Instead, the serverless functions can be executed on demand using resources maintained by the provider network 400. In some examples, these resources can be maintained in a “ready” state (e.g., having a pre-initialized runtime environment configured to execute the serverless functions), allowing the serverless functions to be executed in near real-time.

[0043] FIG. 5 illustrates examples of a provider network and device messaging service thereof. The device messaging service 501 allows for user devices to communicate with each other, allows for registration of users and devices, stores and provides a public key for registered cryptographic devices, and provides the infrastructure for devices to send and pull (encrypted) messages.

[0044] A device and/or user registry 511 stores information about cryptographic devices such as identification information, owner (user), contact information, etc. In some examples, users are registered and verified by a trusted human agent. If verified, a unique user identifier is generated. The trusted agent, in some examples, obtains the user’s biometric data and biometric features, customer contact information, and user ID are added to a secure user directory of the registry 511.

In some examples, public keys (e.g., generated by certificate service 509) for each device are stored in the registry 511.

[0045] A user's ID is associated with a unique serial number of a cryptographic messaging device. In some examples, the registration of a cryptographic messaging device causes an encrypted operating system image to be loaded into the device. A key to decrypt the operating system image is loaded into the device's HSM. This key is used to decrypt the operating system image each time the device boots up. The customer's biometric features (when used) are entered into the device HSM.

[0046] The HSM is triggered to generate a set of private keys. The private keys cannot be extracted from the HSM and will be erased by the HSM, along with all identifying information such as the biometric features, if the HSM detects an attempt to physically compromise the device. Authentication (e.g., biometrics, camera, pin code, password, etc.) plus the HSM physical compromise detection mitigates the risk of smash-and-grab attacks. Additionally, there is no customer message data retained at rest. The only data retained when the device is powered down are the keys stored in the HSM and the encrypted operating system image.

[0047] Users can resign voluntarily or involuntarily. At any moment and for any reason, the administrator of the device messaging service 501 can revoke the privileges granted to a specific device and delete the owning user's information from the registry 511. This isolates the device from the device messaging service 501 and all other user devices, as would be required if a cryptographic messaging device was lost or stolen. For a planned user resignation, a trusted human agent would collect the device and isolate it from all other users and the device messaging service 501. The trusted agent could recycle the device by zeroing its HSM, checking the device for evidence of physical tampering, and then re-registering it to a new user.

[0048] An authentication/authorization component 503 verifies the identities of devices. This authentication may be a server authentication (allowing devices to ensure they are communicating with the right service) and/or a client authentication where devices authenticate themselves. In some examples, X.509 certificates are used. Authorization determines if the devices is allowed to access messages, etc.

[0049] A device gateway 505 handles connections and communications for devices. In some examples, the device gateway 505 is the message broker for a publish/subscribe model. Messages to be delivered are stored in message storage 513. In some examples, the storage is encrypted (thus the messages themselves are encrypted multiple times). In some examples, the device

messaging service 501 generates an out-of-band message to alert a receiving device that there is a message being stored for the receiving device.

[0050] A rules engine 507 allows for devices to interact with other services of the provider network 400. Rules are analyzed and actions are performed.

[0051] A certificate service 509 allows for the generation of keys/certificates to be used.

[0052] In some examples, the device messaging service 501 does not have the ability to decrypt any messages. In some examples, the device messaging service 501 stores a message until a cryptographic messaging device connects to the service and receives it but does not keep it longer than that (if so configured). The device messaging service 501 supports multiple organizations each with isolated sets of cryptographic devices.

[0053] To send a message using the device messaging service 501, a user looks up the recipient in the registry 511. The sender's device connects to the recipient device by way of the device messaging service 501. Once connected, the sender's device and the recipient's device derive unique cryptographic material for a secure, post-compromise safe (e.g., using a double ratchet algorithm), pair-wise conversation. This ensures that the device messaging service 501, eavesdroppers, nor man-in-the-middle attackers can observe or compromise the conversation.

[0054] FIG. 6 is a flow diagram illustrating operations of a method for using a cryptographic device according to some examples. Some or all of the operations (or other processes described herein, or variations, and/or combinations thereof) are performed under the control of one or more computer systems configured with executable instructions, and are implemented as code (e.g., executable instructions, one or more computer programs, or one or more applications) executing collectively on one or more processors. The code is stored on a computer-readable storage medium, for example, in the form of a computer program comprising instructions executable by one or more processors. The computer-readable storage medium is non-transitory. In some examples, one or more (or all) of the operations are performed by one or more of a cryptographic messaging device, messaging service, etc. of the other figures.

[0055] In some examples, a cryptographic messaging device is enrolled with a provider network service at 602. This enrollment allows for the cryptographic messaging device to send and/or receive messages.

[0056] FIG. 7 is a flow diagram illustrating operations of a method for enrolling a cryptographic messaging device according to some examples. Some or all of the operations (or other processes described herein, or variations, and/or combinations thereof) are performed under the control of

one or more computer systems configured with executable instructions, and are implemented as code (e.g., executable instructions, one or more computer programs, or one or more applications) executing collectively on one or more processors. The code is stored on a computer-readable storage medium, for example, in the form of a computer program comprising instructions executable by one or more processors. The computer-readable storage medium is non-transitory. In some examples, one or more (or all) of the operations are performed by one or more of a cryptographic messaging device, messaging service, etc. of the other figures.

[0057] An identifier to the cryptographic messaging device is assigned at 702. This identifier is used to uniquely identify the device and is associated with a particular user. As such, messages directed to a user can be routed to the correct device. However, even if messages were routed incorrectly, the wrong receiving device will not have the required information to decrypt the message.

[0058] A hardware security module of the cryptographic messaging device is engaged to generate and store a private key and, in some examples, generate a certificate service request corresponding to a public key for the device and/or user at 704.

[0059] The certificate service request is received, and a certificate authority is made to create and sign a certificate at 706. For example, certificate service 509 is invoked.

[0060] The certificate is attached to the cryptographic messaging device at 708. That is the certificate is attached to the device in the registry 511. The certificate is also sent to the cryptographic messaging device to be stored by the cryptographic messaging device at 710. As such, the cryptographic messaging device has a unique certificate to authenticate its identity.

[0061] At least one security policy is attached to the certificate (and device) at 712. The policy allows for the device to send and receive messages.

[0062] At some point in time, an encrypted message is received from the cryptographic messaging device at 604. In some examples, this occurs after at least the sending cryptographic messaging device has been authenticated at 603. This message is to be sent to a desired recipient. FIG. 8 is a flow diagram illustrating operations of a method for generating a message to be received according to some examples. Some or all of the operations (or other processes described herein, or variations, and/or combinations thereof) are performed under the control of one or more computer systems configured with executable instructions, and are implemented as code (e.g., executable instructions, one or more computer programs, or one or more applications) executing collectively on one or more processors. The code is stored on a computer-readable storage

medium, for example, in the form of a computer program comprising instructions executable by one or more processors. The computer-readable storage medium is non-transitory. In some examples, one or more (or all) of the operations are performed by one or more of a cryptographic messaging device, messaging service, etc. of the other figures.

[0063] A selection of, or input of, a recipient identifier is received at 802. For example, a user ID is selected. The selection of a recipient establishes a communication channel. In some examples, ECDH key exchange techniques are used to create the channel. For example, this selection causes each user device to generate a random ECC pair from their public and private keys, the public keys are exchanged via the messaging service, and a shared key is calculated by the devices based on the user's private key and the received public key. The shared key is the same key for both users. In some examples, the certificate for each device contains the public key for that device.

[0064] A message to send (text, image, audio, video, etc.) is received within the sending cryptographic messaging device at 804. The cryptographic messaging device signs the message with the private key using the cryptographic messaging device's HSM at 806.

[0065] The signed message then encrypted with the secret shared key at 812. Note that message may also include a "new" public key that the sender will use in the future. That is a ratcheting process may be followed such that the comprise of one key will not compromise all messages.

[0066] The encrypted, signed message is sent to the messaging service for delivery to the recipient at 814.

[0067] The encrypted message from the cryptographic messaging device is routed to a receiving device at 606. In some examples, this occurs after at least the receiving cryptographic messaging device has been authenticated at 605. This may happen as the receiving device connects to the messaging service, at a scheduled time, etc. FIG. 9 is a flow diagram illustrating operations of a method for handling a message according to some examples. Some or all of the operations (or other processes described herein, or variations, and/or combinations thereof) are performed under the control of one or more computer systems configured with executable instructions, and are implemented as code (e.g., executable instructions, one or more computer programs, or one or more applications) executing collectively on one or more processors. The code is stored on a computer-readable storage medium, for example, in the form of a computer program comprising instructions executable by one or more processors. The computer-readable storage medium is non-transitory. In some examples, one or more (or all) of the operations are performed by one or more of a cryptographic messaging device, messaging service, etc. of the other figures.

[0068] A message is received by a cryptographic messaging device at 902. For example, a message is received from the messaging service detailed herein. In some examples, a Transport Layer Security (TLS) socket of the messaging service is listened to for a message.

[0069] A shared key is calculated at 904. In particular, a HSM of the cryptographic messaging device calculates a shared key from its stored private key and the sender's public key that is a part of the message or provided by the messaging service. Note that ratcheting may also be used.

[0070] The message is decrypted using the shared key at 906 and a signature of the decrypted message is verified at 908 to prove the identity of the sender. When the identity is verified and the message unaltered, the content of the message is presented to a user of the messaging service at 910. Note that in some examples once a message is received it is deleted from the messaging service.

[0071] FIG. 10 illustrates an example provider network (or "service provider system") environment according to some examples. A provider network 1000 can provide resource virtualization to customers via one or more virtualization services 1010 that allow customers to purchase, rent, or otherwise obtain instances 1012 of virtualized resources, including but not limited to computation and storage resources, implemented on devices within the provider network or networks in one or more data centers. Local Internet Protocol (IP) addresses 1016 can be associated with the resource instances 1012; the local IP addresses are the internal network addresses of the resource instances 1012 on the provider network 1000. In some examples, the provider network 1000 can also provide public IP addresses 1014 and/or public IP address ranges (e.g., Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) addresses) that customers can obtain from the provider 1000.

[0072] Conventionally, the provider network 1000, via the virtualization services 1010, can allow a customer of the service provider (e.g., a customer that operates one or more customer networks 1050A-1050C (or "client networks") including one or more customer device(s) 1052) to dynamically associate at least some public IP addresses 1014 assigned or allocated to the customer with particular resource instances 1012 assigned to the customer. The provider network 1000 can also allow the customer to remap a public IP address 1014, previously mapped to one virtualized computing resource instance 1012 allocated to the customer, to another virtualized computing resource instance 1012 that is also allocated to the customer. Using the virtualized computing resource instances 1012 and public IP addresses 1014 provided by the service provider, a customer of the service provider such as the operator of the customer network(s) 1050A-1050C can, for example, implement customer-specific applications and present the customer's

applications on an intermediate network 1040, such as the Internet. Other network entities 1020 on the intermediate network 1040 can then generate traffic to a destination public IP address 1014 published by the customer network(s) 1050A-1050C; the traffic is routed to the service provider data center, and at the data center is routed, via a network substrate, to the local IP address 1016 of the virtualized computing resource instance 1012 currently mapped to the destination public IP address 1014. Similarly, response traffic from the virtualized computing resource instance 1012 can be routed via the network substrate back onto the intermediate network 1040 to the source entity 1020.

[0073] Local IP addresses, as used herein, refer to the internal or “private” network addresses, for example, of resource instances in a provider network. Local IP addresses can be within address blocks reserved by Internet Engineering Task Force (IETF) Request for Comments (RFC) 1918 and/or of an address format specified by IETF RFC 4193 and can be mutable within the provider network. Network traffic originating outside the provider network is not directly routed to local IP addresses; instead, the traffic uses public IP addresses that are mapped to the local IP addresses of the resource instances. The provider network can include networking devices or appliances that provide network address translation (NAT) or similar functionality to perform the mapping from public IP addresses to local IP addresses and vice versa.

[0074] Public IP addresses are Internet mutable network addresses that are assigned to resource instances, either by the service provider or by the customer. Traffic routed to a public IP address is translated, for example via 1:1 NAT, and forwarded to the respective local IP address of a resource instance.

[0075] Some public IP addresses can be assigned by the provider network infrastructure to particular resource instances; these public IP addresses can be referred to as standard public IP addresses, or simply standard IP addresses. In some examples, the mapping of a standard IP address to a local IP address of a resource instance is the default launch configuration for all resource instance types.

[0076] At least some public IP addresses can be allocated to or obtained by customers of the provider network 1000; a customer can then assign their allocated public IP addresses to particular resource instances allocated to the customer. These public IP addresses can be referred to as customer public IP addresses, or simply customer IP addresses. Instead of being assigned by the provider network 1000 to resource instances as in the case of standard IP addresses, customer IP addresses can be assigned to resource instances by the customers, for example via an API provided by the service provider. Unlike standard IP addresses, customer IP addresses are allocated to

customer accounts and can be remapped to other resource instances by the respective customers as necessary or desired. A customer IP address is associated with a customer's account, not a particular resource instance, and the customer controls that IP address until the customer chooses to release it. Unlike conventional static IP addresses, customer IP addresses allow the customer to mask resource instance or availability zone failures by remapping the customer's public IP addresses to any resource instance associated with the customer's account. The customer IP addresses, for example, enable a customer to engineer around problems with the customer's resource instances or software by remapping customer IP addresses to replacement resource instances.

[0077] FIG. 11 is a block diagram of an example provider network environment that provides a storage service and a hardware virtualization service to customers, according to some examples. A hardware virtualization service 1120 provides multiple compute resources 1124 (e.g., compute instances 1125, such as VMs) to customers. The compute resources 1124 can, for example, be provided as a service to customers of a provider network 1100 (e.g., to a customer that implements a customer network 1150). Each computation resource 1124 can be provided with one or more local IP addresses. The provider network 1100 can be configured to route packets from the local IP addresses of the compute resources 1124 to public Internet destinations, and from public Internet sources to the local IP addresses of the compute resources 1124.

[0078] The provider network 1100 can provide the customer network 1150, for example coupled to an intermediate network 1140 via a local network 1156, the ability to implement virtual computing systems 1192 via the hardware virtualization service 1120 coupled to the intermediate network 1140 and to the provider network 1100. In some examples, the hardware virtualization service 1120 can provide one or more APIs 1102, for example a web services interface, via which the customer network 1150 can access functionality provided by the hardware virtualization service 1120, for example via a console 1194 (e.g., a web-based application, standalone application, mobile application, etc.) of a customer device 1190. In some examples, at the provider network 1100, each virtual computing system 1192 at the customer network 1150 can correspond to a computation resource 1124 that is leased, rented, or otherwise provided to the customer network 1150.

[0079] From an instance of the virtual computing system(s) 1192 and/or another customer device 1190 (e.g., via console 1194), the customer can access the functionality of a storage service 1110, for example via the one or more APIs 1102, to access data from and store data to storage resources 1118A-1118N of a virtual data store 1116 (e.g., a folder or "bucket," a virtualized

volume, a database, etc.) provided by the provider network 1100. In some examples, a virtualized data store gateway (not shown) can be provided at the customer network 1150 that can locally cache at least some data, for example frequently accessed or critical data, and that can communicate with the storage service 1110 via one or more communications channels to upload new or modified data from a local cache so that the primary store of data (the virtualized data store 1116) is maintained. In some examples, a user, via the virtual computing system 1192 and/or another customer device 1190, can mount and access virtual data store 1116 volumes via the storage service 1110 acting as a storage virtualization service, and these volumes can appear to the user as local (virtualized) storage 1198.

[0080] While not shown in FIG. 11, the virtualization service(s) can also be accessed from resource instances within the provider network 1100 via the API(s) 1102. For example, a customer, appliance service provider, or other entity can access a virtualization service from within a respective virtual network on the provider network 1100 via the API(s) 1102 to request allocation of one or more resource instances within the virtual network or within another virtual network.

Illustrative Systems

[0081] In some examples, a system that implements a portion or all of the techniques described herein can include a general-purpose computer system, such as the computer system 1200 (also referred to as a computing device or electronic device) illustrated in FIG. 12, that includes, or is configured to access, one or more computer-accessible media. In the illustrated example, the computer system 1200 includes one or more processors 1210 coupled to a system memory 1220 via an input/output (I/O) interface 1230. The computer system 1200 further includes a network interface 1240 coupled to the I/O interface 1230. While FIG. 12 shows the computer system 1200 as a single computing device, in various examples the computer system 1200 can include one computing device or any number of computing devices configured to work together as a single computer system 1200.

[0082] In various examples, the computer system 1200 can be a uniprocessor system including one processor 1210, or a multiprocessor system including several processors 1210 (e.g., two, four, eight, or another suitable number). The processor(s) 1210 can be any suitable processor(s) capable of executing instructions. For example, in various examples, the processor(s) 1210 can be general-purpose or embedded processors implementing any of a variety of instruction set architectures (ISAs), such as the x86, ARM, PowerPC, SPARC, or MIPS ISAs, or any other suitable ISA. In

multiprocessor systems, each of the processors 1210 can commonly, but not necessarily, implement the same ISA.

[0083] The system memory 1220 can store instructions and data accessible by the processor(s) 1210. In various examples, the system memory 1220 can be implemented using any suitable memory technology, such as random-access memory (RAM), static RAM (SRAM), synchronous dynamic RAM (SDRAM), nonvolatile/Flash-type memory, or any other type of memory. In the illustrated example, program instructions and data implementing one or more desired functions, such as those methods, techniques, and data described above, are shown stored within the system memory 1220 as messaging service code 1225 (e.g., executable to implement, in whole or in part, the device messaging service 501) and data 1226.

[0084] In some examples, the I/O interface 1230 can be configured to coordinate I/O traffic between the processor 1210, the system memory 1220, and any peripheral devices in the device, including the network interface 1240 and/or other peripheral interfaces (not shown). In some examples, the I/O interface 1230 can perform any necessary protocol, timing, or other data transformations to convert data signals from one component (e.g., the system memory 1220) into a format suitable for use by another component (e.g., the processor 1210). In some examples, the I/O interface 1230 can include support for devices attached through various types of peripheral buses, such as a variant of the Peripheral Component Interconnect (PCI) bus standard or the Universal Serial Bus (USB) standard, for example. In some examples, the function of the I/O interface 1230 can be split into two or more separate components, such as a north bridge and a south bridge, for example. Also, in some examples, some or all of the functionality of the I/O interface 1230, such as an interface to the system memory 1220, can be incorporated directly into the processor 1210.

[0085] The network interface 1240 can be configured to allow data to be exchanged between the computer system 1200 and other devices 1260 attached to a network or networks 1250, such as other computer systems or devices as illustrated in FIG. 1, for example. In various examples, the network interface 1240 can support communication via any suitable wired or wireless general data networks, such as types of Ethernet network, for example. Additionally, the network interface 1240 can support communication via telecommunications/telephony networks, such as analog voice networks or digital fiber communications networks, via storage area networks (SANs), such as Fibre Channel SANs, and/or via any other suitable type of network and/or protocol.

[0086] In some examples, the computer system 1200 includes one or more offload cards 1270A or 1270B (including one or more processors 1275, and possibly including the one or more network

interfaces 1240) that are connected using the I/O interface 1230 (e.g., a bus implementing a version of the Peripheral Component Interconnect - Express (PCI-E) standard, or another interconnect such as a QuickPath interconnect (QPI) or UltraPath interconnect (UPI)). For example, in some examples the computer system 1200 can act as a host electronic device (e.g., operating as part of a hardware virtualization service) that hosts compute resources such as compute instances, and the one or more offload cards 1270A or 1270B execute a virtualization manager that can manage compute instances that execute on the host electronic device. As an example, in some examples the offload card(s) 1270A or 1270B can perform compute instance management operations, such as pausing and/or un-pausing compute instances, launching and/or terminating compute instances, performing memory transfer/copying operations, etc. These management operations can, in some examples, be performed by the offload card(s) 1270A or 1270B in coordination with a hypervisor (e.g., upon a request from a hypervisor) that is executed by the other processors 1210A-1210N of the computer system 1200. However, in some examples the virtualization manager implemented by the offload card(s) 1270A or 1270B can accommodate requests from other entities (e.g., from compute instances themselves), and cannot coordinate with (or service) any separate hypervisor.

[0087] In some examples, the system memory 1220 can be one example of a computer-accessible medium configured to store program instructions and data as described above. However, in other examples, program instructions and/or data can be received, sent, or stored upon different types of computer-accessible media. Generally speaking, a computer-accessible medium can include any non-transitory storage media or memory media such as magnetic or optical media, e.g., disk or DVD/CD coupled to the computer system 1200 via the I/O interface 1230. A non-transitory computer-accessible storage medium can also include any volatile or non-volatile media such as RAM (e.g., SDRAM, double data rate (DDR) SDRAM, SRAM, etc.), read only memory (ROM), etc., that can be included in some examples of the computer system 1200 as the system memory 1220 or another type of memory. Further, a computer-accessible medium can include transmission media or signals such as electrical, electromagnetic, or digital signals, conveyed via a communication medium such as a network and/or a wireless link, such as can be implemented via the network interface 1240.

[0088] Various examples discussed or suggested herein can be implemented in a wide variety of operating environments, which in some cases can include one or more user computers, computing devices, or processing devices which can be used to operate any of a number of applications. User or client devices can include any of a number of general-purpose personal computers, such as desktop or laptop computers running a standard operating system, as well as

cellular, wireless, and handheld devices running mobile software and capable of supporting a number of networking and messaging protocols. Such a system also can include a number of workstations running any of a variety of commercially available operating systems and other known applications for purposes such as development and database management. These devices also can include other electronic devices, such as dummy terminals, thin-clients, gaming systems, and/or other devices capable of communicating via a network.

[0089] Most examples use at least one network that would be familiar to those skilled in the art for supporting communications using any of a variety of widely-available protocols, such as Transmission Control Protocol / Internet Protocol (TCP/IP), File Transfer Protocol (FTP), Universal Plug and Play (UPnP), Network File System (NFS), Common Internet File System (CIFS), Extensible Messaging and Presence Protocol (XMPP), AppleTalk, etc. The network(s) can include, for example, a local area network (LAN), a wide-area network (WAN), a virtual private network (VPN), the Internet, an intranet, an extranet, a public switched telephone network (PSTN), an infrared network, a wireless network, and any combination thereof.

[0090] In examples using a web server, the web server can run any of a variety of server or mid-tier applications, including HTTP servers, File Transfer Protocol (FTP) servers, Common Gateway Interface (CGI) servers, data servers, Java servers, business application servers, etc. The server(s) also can be capable of executing programs or scripts in response requests from user devices, such as by executing one or more Web applications that can be implemented as one or more scripts or programs written in any programming language, such as Java®, C, C# or C++, or any scripting language, such as Perl, Python, PHP, or TCL, as well as combinations thereof. The server(s) can also include database servers, including without limitation those commercially available from Oracle(R), Microsoft(R), Sybase(R), IBM(R), etc. The database servers can be relational or non-relational (e.g., “NoSQL”), distributed or non-distributed, etc.

[0091] Environments disclosed herein can include a variety of data stores and other memory and storage media as discussed above. These can reside in a variety of locations, such as on a storage medium local to (and/or resident in) one or more of the computers or remote from any or all of the computers across the network. In a particular set of examples, the information can reside in a storage-area network (SAN) familiar to those skilled in the art. Similarly, any necessary files for performing the functions attributed to the computers, servers, or other network devices can be stored locally and/or remotely, as appropriate. Where a system includes computerized devices, each such device can include hardware elements that can be electrically coupled via a bus, the elements including, for example, at least one central processing unit (CPU), at least one input

device (e.g., a mouse, keyboard, controller, touch screen, or keypad), and/or at least one output device (e.g., a display device, printer, or speaker). Such a system can also include one or more storage devices, such as disk drives, optical storage devices, and solid-state storage devices such as random-access memory (RAM) or read-only memory (ROM), as well as removable media devices, memory cards, flash cards, etc.

[0092] Such devices also can include a computer-readable storage media reader, a communications device (e.g., a modem, a network card (wireless or wired), an infrared communication device, etc.), and working memory as described above. The computer-readable storage media reader can be connected with, or configured to receive, a computer-readable storage medium, representing remote, local, fixed, and/or removable storage devices as well as storage media for temporarily and/or more permanently containing, storing, transmitting, and retrieving computer-readable information. The system and various devices also typically will include a number of software applications, modules, services, or other elements located within at least one working memory device, including an operating system and application programs, such as a client application or web browser. It should be appreciated that alternate examples can have numerous variations from that described above. For example, customized hardware might also be used and/or particular elements might be implemented in hardware, software (including portable software, such as applets), or both. Further, connection to other computing devices such as network input/output devices can be employed.

[0093] Storage media and computer readable media for containing code, or portions of code, can include any appropriate media known or used in the art, including storage media and communication media, such as but not limited to volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage and/or transmission of information such as computer readable instructions, data structures, program modules, or other data, including RAM, ROM, Electrically Erasable Programmable Read-Only Memory (EEPROM), flash memory or other memory technology, Compact Disc-Read Only Memory (CD-ROM), Digital Versatile Disk (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by a system device. Based on the disclosure and teachings provided herein, a person of ordinary skill in the art will appreciate other ways and/or methods to implement the various examples.

[0094] In the preceding description, various examples are described. For purposes of explanation, specific configurations and details are set forth in order to provide a thorough

understanding of the examples. However, it will also be apparent to one skilled in the art that the examples can be practiced without the specific details. Furthermore, well-known features can be omitted or simplified in order not to obscure the example being described.

[0095] Bracketed text and blocks with dashed borders (e.g., large dashes, small dashes, dot-dash, and dots) are used herein to illustrate optional aspects that add additional features to some examples. However, such notation should not be taken to mean that these are the only options or optional operations, and/or that blocks with solid borders are not optional in certain examples.

[0096] Reference numerals with suffix letters can be used to indicate that there can be one or multiple instances of the referenced entity in various examples, and when there are multiple instances, each does not need to be identical but may instead share some general traits or act in common ways. Further, the particular suffixes used are not meant to imply that a particular amount of the entity exists unless specifically indicated to the contrary. Thus, two entities using the same or different suffix letters might or might not have the same number of instances in various examples.

[0097] References to “one example,” “an example,” etc., indicate that the example described may include a particular feature, structure, or characteristic, but every example may not necessarily include the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same example. Further, when a particular feature, structure, or characteristic is described in connection with an example, it is submitted that it is within the knowledge of one skilled in the art to affect such feature, structure, or characteristic in connection with other examples whether or not explicitly described.

[0098] Moreover, in the various examples described above, unless specifically noted otherwise, disjunctive language such as the phrase “at least one of A, B, or C” is intended to be understood to mean either A, B, or C, or any combination thereof (e.g., A, B, and/or C). Similarly, language such as “at least one or more of A, B, and C” (or “one or more of A, B, and C”) is intended to be understood to mean A, B, or C, or any combination thereof (e.g., A, B, and/or C). As such, disjunctive language is not intended to, nor should it be understood to, imply that a given example requires at least one of A, at least one of B, and at least one of C to each be present.

[0099] As used herein, the term “based on” (or similar) is an open-ended term used to describe one or more factors that affect a determination or other action. It is to be understood that this term does not foreclose additional factors that may affect a determination or action. For example, a determination may be solely based on the factor(s) listed or based on the factor(s) and one or more additional factors. Thus, if an action A is “based on” B, it is to be understood that B is one factor

that affects action A, but this does not foreclose the action from also being based on one or multiple other factors, such as factor C. However, in some instances, action A may be based entirely on B.

[00100] Unless otherwise explicitly stated, articles such as “a” or “an” should generally be interpreted to include one or multiple described items. Accordingly, phrases such as “a device configured to” or “a computing device” are intended to include one or multiple recited devices. Such one or more recited devices can be collectively configured to carry out the stated operations. For example, “a processor configured to carry out operations A, B, and C” can include a first processor configured to carry out operation A working in conjunction with a second processor configured to carry out operations B and C.

[00101] Further, the words “may” or “can” are used in a permissive sense (i.e., meaning having the potential to), rather than the mandatory sense (i.e., meaning must). The words “include,” “including,” and “includes” are used to indicate open-ended relationships and therefore mean including, but not limited to. Similarly, the words “have,” “having,” and “has” also indicate open-ended relationships, and thus mean having, but not limited to. The terms “first,” “second,” “third,” and so forth as used herein are used as labels for the nouns that they precede, and do not imply any type of ordering (e.g., spatial, temporal, logical, etc.) unless such an ordering is otherwise explicitly indicated. Similarly, the values of such numeric labels are generally not used to indicate a required amount of a particular noun in the claims recited herein, and thus a “fifth” element generally does not imply the existence of four other elements unless those elements are explicitly included in the claim or it is otherwise made abundantly clear that they exist.

[00102] At least some embodiments of the disclosed technologies can be described in view of the following clauses:

1. A computer-implemented method comprising:
 - receiving, from a messaging service of a provider network, an encrypted message from a first cryptographic messaging device at a second cryptographic messaging device via an attached network-enabled device, wherein the second cryptographic messaging device does not include networking capabilities;
 - calculating a shared key from a public key of the first cryptographic messaging device and a private key of the second cryptographic messaging device using a hardware security module (HSM) of the second cryptographic messaging device;
 - decrypting the encrypted message using the shared key;
 - verifying a signature of the decrypted message; and
 - displaying contents of the decrypted message.

2. The computer-implemented method of clause 1, wherein the private key of the second cryptographic messaging device is stored in a hardware security module of the second cryptographic messaging device.
3. The computer-implemented method of any one of clauses 1-2, wherein the public key is managed by the messaging service of a provider network.
4. An apparatus comprising:
 - an input/output port configured to receive an encrypted message from a coupled external device;
 - a hardware security module (HSM) configured to decrypt the encrypted message, wherein the HSM is to include storage to store at least one private key to be used to decrypt the encrypted message; and
 - a screen to display contents of the decrypted message.
5. The apparatus of clause 4, wherein power for the apparatus is to be provided from the external device.
6. The apparatus of any one of clauses 4-5, wherein the encrypted message is to be deleted from a message delivery service of a provider network upon the apparatus receiving the encrypted message.
7. The apparatus of any one of clauses 4-6, wherein the apparatus is to use an Elliptic Curve Diffie-Hellman key exchange with a sending device.
8. The apparatus of any one of clauses 4-7, wherein the encrypted message is to be received directly from a sending device.
9. The apparatus of any one of clauses 4-8, wherein the encrypted message is to be received from a message delivery service of a provider network.
10. The apparatus of any one of clauses 4-9, wherein the HSM is to include at least one cryptographic engine and the storage is secure storage.
11. The apparatus of any one of clauses 4-10, further comprising:
 - a physical intrusion detection system to monitor for symptoms of physical tampering of the apparatus.

12. The apparatus of any one of clauses 4-11, further comprising:
a processor; and
a memory coupled to the processor, the memory to store at least an operating system and
a messaging application.
13. The apparatus of any one of clauses 4-12, wherein the encrypted message is to include one
or more of text, a document, an image, a video file, or an audio file.
14. The apparatus of any one of clauses 4-13, further comprising:
a physical keyboard to input a message.
15. The apparatus of any one of clauses 4-14, wherein the encrypted message is to be deleted
from the apparatus after a period of time has expired.
16. The apparatus of any one of clauses 4-15, further comprising:
a user authentication mechanism.
17. A system comprising:
a first one or more electronic devices to implement a network-enabled device; and
a second one or more electronic devices to implement a cryptographic messaging device,
the cryptographic messaging device comprising:
an input/output port to receive an encrypted message using the network-enabled
device, wherein the network-enabled device is to provide network
capabilities for the cryptographic messaging device,
a hardware security module (HSM) to decrypt the encrypted message, wherein the
HSM is to include storage to store at least one private key to be used to
decrypt the encrypted message, and
a screen to display contents of the decrypted message.
18. The system of clause 17, wherein the network-enabled device is a cellular device having
an operating system and networking hardware to couple to a telecommunications network.
19. The system of any one of clauses 17-18, further comprising:
a third one or more electronic devices to implement a messaging service of a provider
network, the messaging service to provide the encrypted message to the
cryptographic messaging device via the network-enabled device.

20. The system of any one of clauses 17-19, wherein the cryptographic messaging device is to use an Elliptic Curve Diffie-Hellman key exchange with a sending device.

[00103] The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. It will, however, be evident that various modifications and changes can be made thereunto without departing from the broader scope of the disclosure as set forth in the claims.

CLAIMS

WHAT IS CLAIMED IS:

1. A computer-implemented method comprising:
receiving, from a messaging service of a provider network, an encrypted message from a first cryptographic messaging device at a second cryptographic messaging device via an attached network-enabled device, wherein the second cryptographic messaging device does not include networking capabilities;
calculating a shared key from a public key of the first cryptographic messaging device and a private key of the second cryptographic messaging device using a hardware security module (HSM) of the second cryptographic messaging device;
decrypting the encrypted message using the shared key;
verifying a signature of the decrypted message; and
displaying contents of the decrypted message.
2. The computer-implemented method of claim 1, wherein the private key of the second cryptographic messaging device is stored in a hardware security module of the second cryptographic messaging device.
3. The computer-implemented method of any one of claims 1-2, wherein the public key is managed by the messaging service of a provider network.
4. An apparatus comprising:
an input/output port configured to receive an encrypted message from a coupled external device;
a hardware security module (HSM) configured to decrypt the encrypted message, wherein the HSM is to include storage to store at least one private key to be used to decrypt the encrypted message; and
a screen to display contents of the decrypted message.
5. The apparatus of claim 4, wherein power for the apparatus is to be provided from the external device.
6. The apparatus of any one of claims 4-5, wherein the encrypted message is to be deleted from a message delivery service of a provider network upon the apparatus receiving the encrypted message.

7. The apparatus of any one of claims 4-6, wherein the apparatus is to use an Elliptic Curve Diffie-Hellman key exchange with a sending device.
8. The apparatus of any one of claims 4-7, wherein the encrypted message is to be received directly from a sending device.
9. The apparatus of any one of claims 4-8, wherein the encrypted message is to be received from a message delivery service of a provider network.
10. The apparatus of any one of claims 4-9, wherein the HSM is to include at least one cryptographic engine and the storage is secure storage.
11. The apparatus of any one of claims 4-10, further comprising:
 - a physical intrusion detection system to monitor for symptoms of physical tampering of the apparatus.
12. A system comprising:
 - a first one or more electronic devices to implement a network-enabled device; and
 - a second one or more electronic devices to implement a cryptographic messaging device, the cryptographic messaging device comprising:
 - an input/output port to receive an encrypted message using the network-enabled device, wherein the network-enabled device is to provide network capabilities for the cryptographic messaging device,
 - a hardware security module (HSM) to decrypt the encrypted message, wherein the HSM is to include storage to store at least one private key to be used to decrypt the encrypted message, and
 - a screen to display contents of the decrypted message.
13. The system of claim 12, wherein the network-enabled device is a cellular device having an operating system and networking hardware to couple to a telecommunications network.
14. The system of any one of claims 12-13, further comprising:
 - a third one or more electronic devices to implement a messaging service of a provider network, the messaging service to provide the encrypted message to the cryptographic messaging device via the network-enabled device.

15. The system of any one of claims 12-14, wherein the cryptographic messaging device is to use an Elliptic Curve Diffie-Hellman key exchange with a sending device.

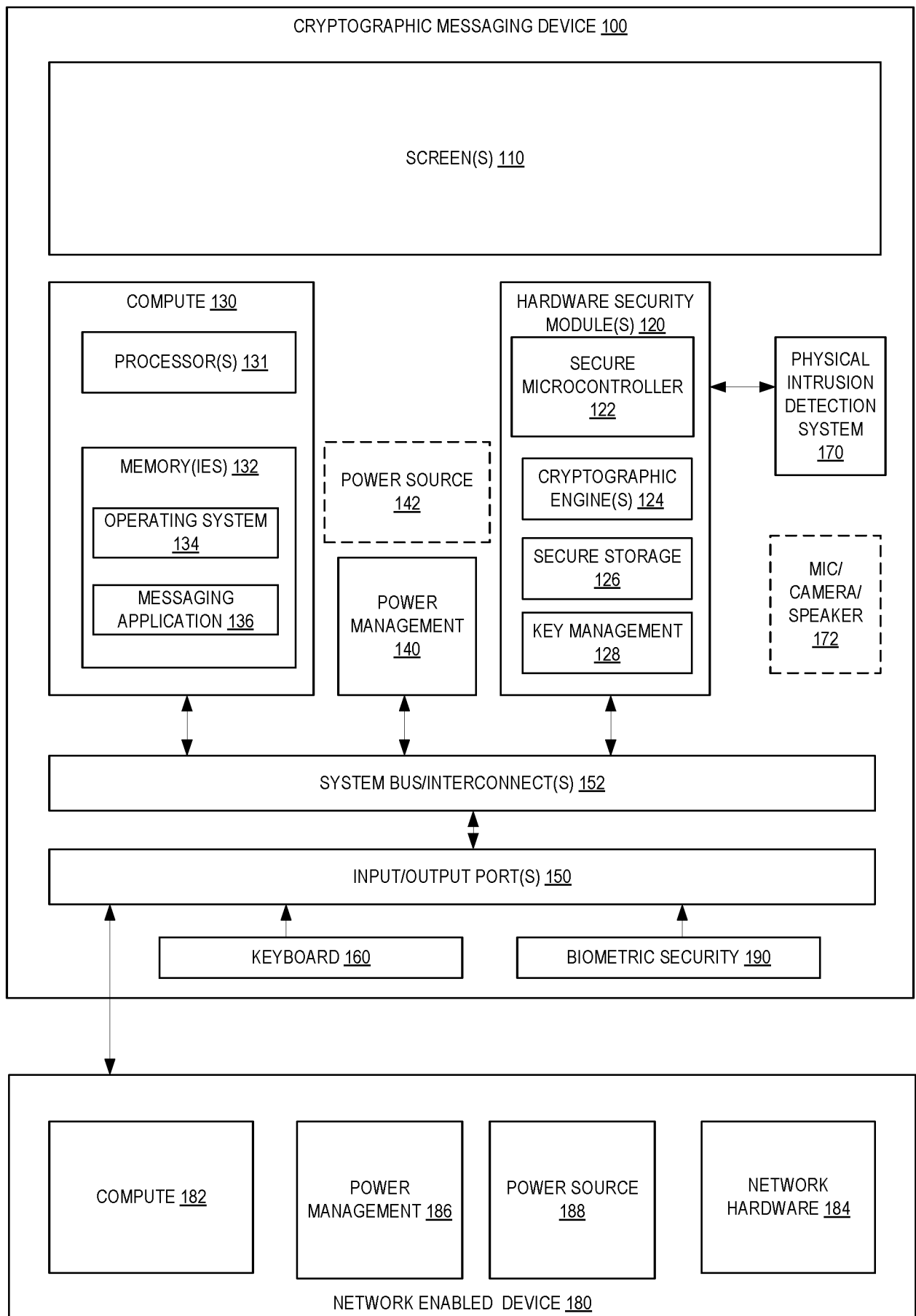


FIG. 1

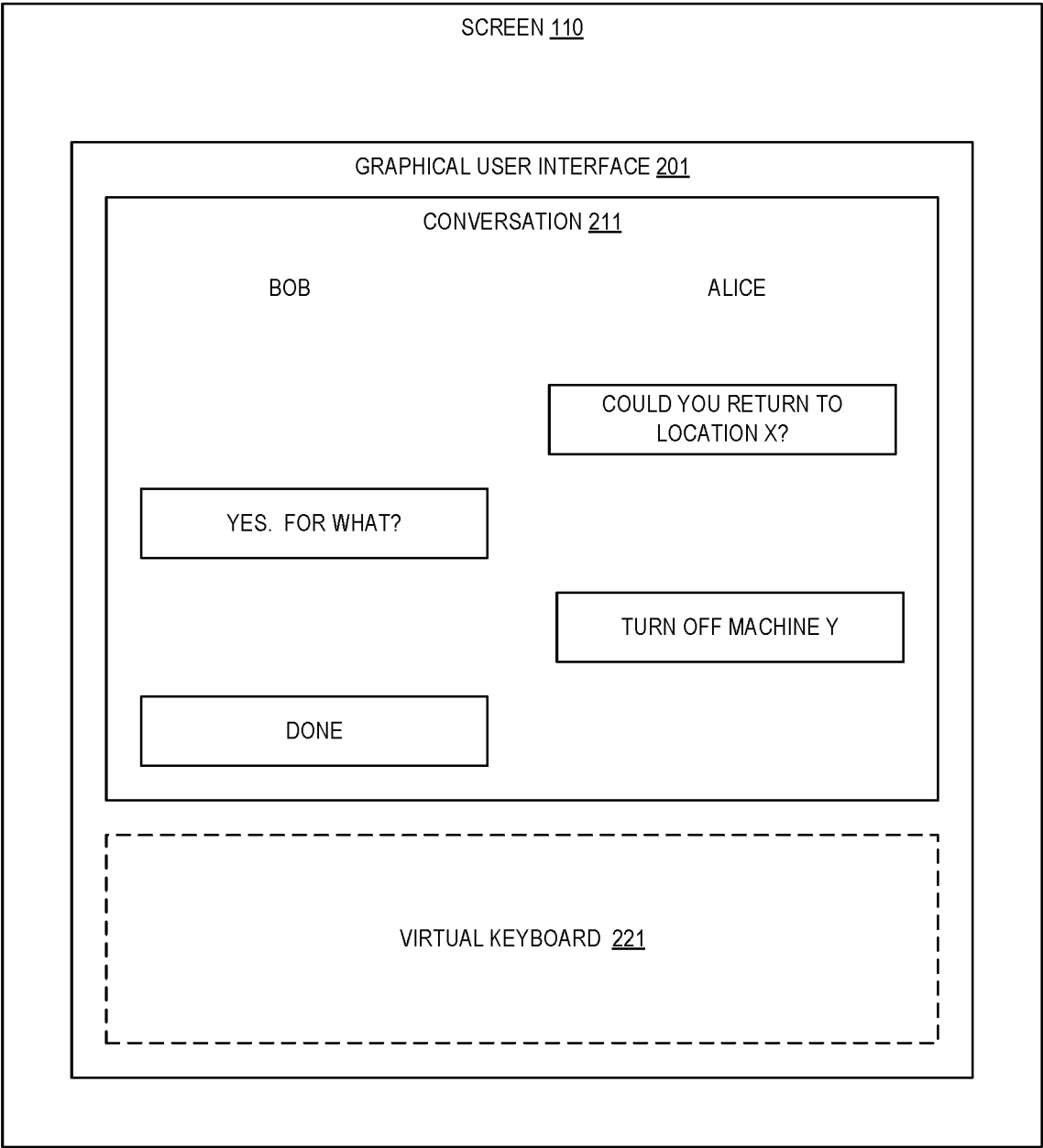


FIG. 2

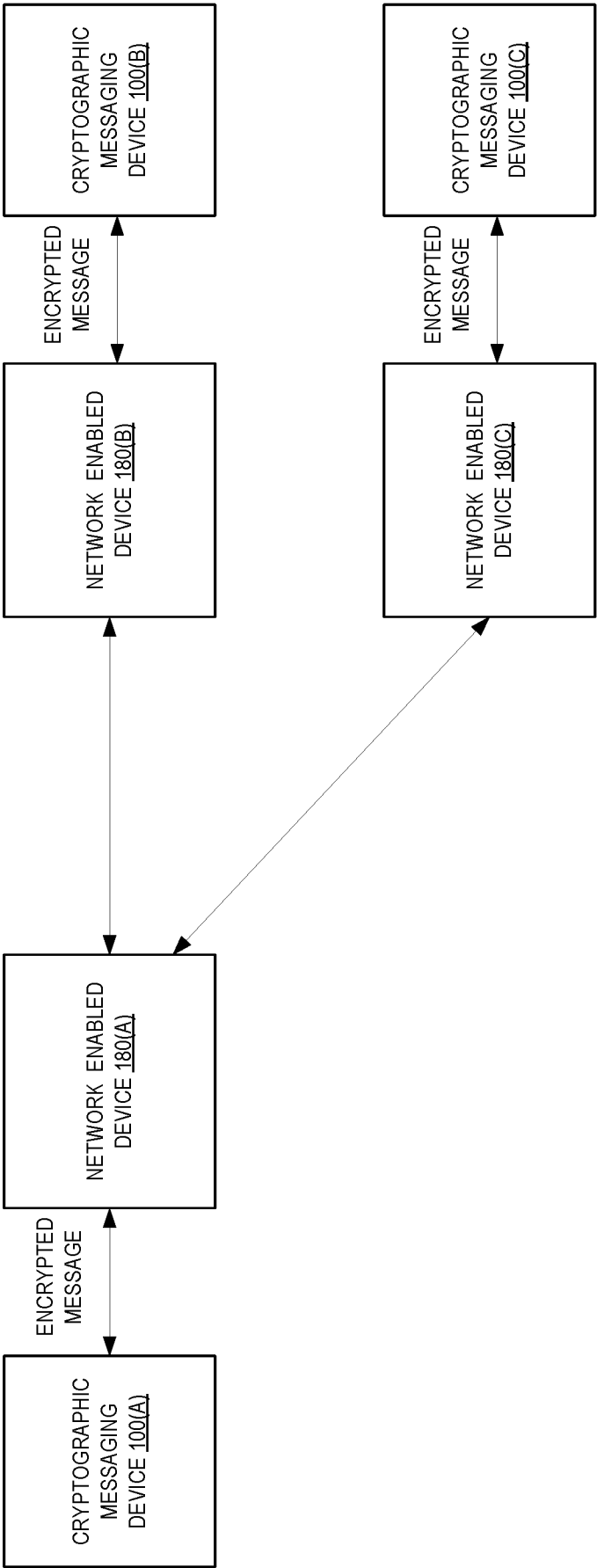


FIG. 3

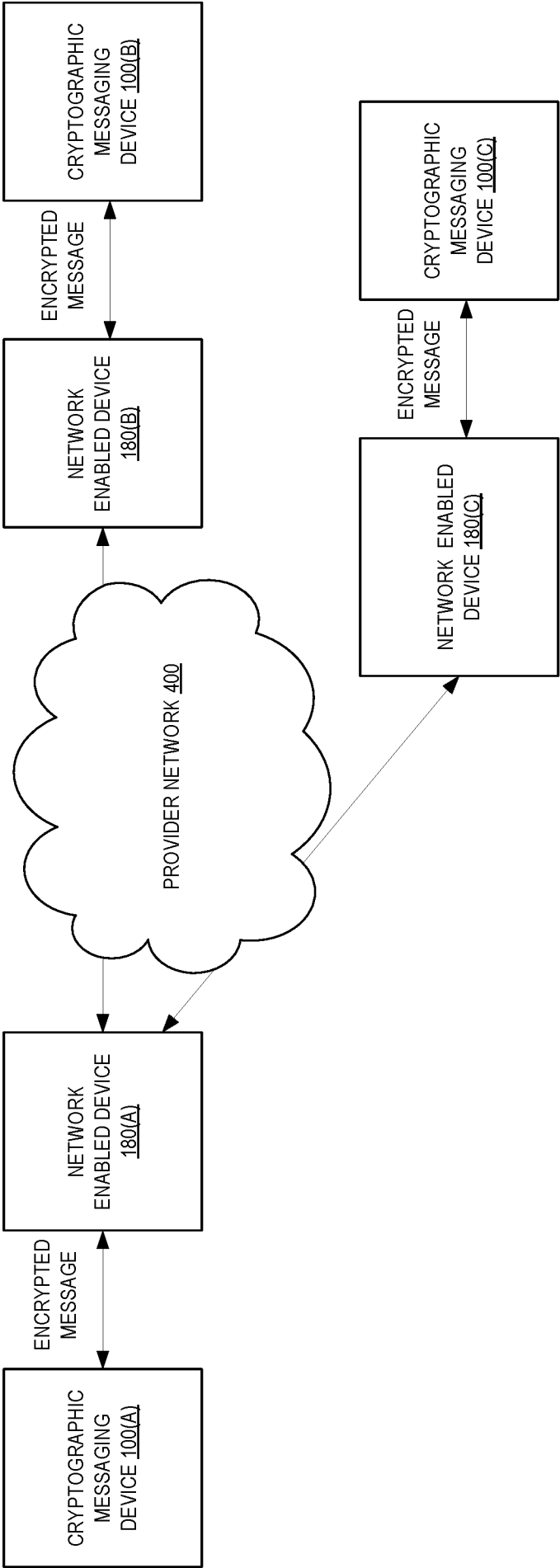


FIG. 4

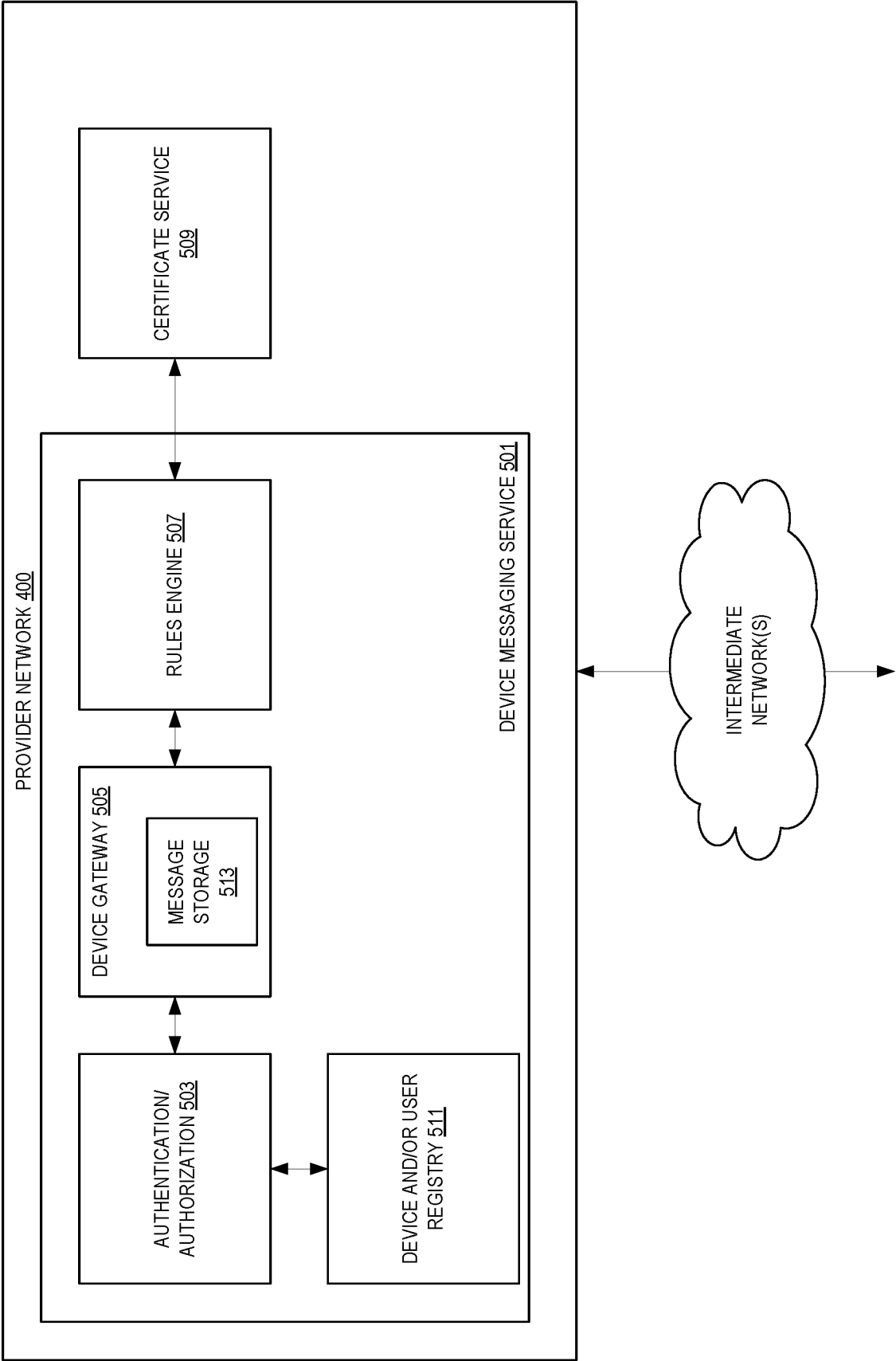
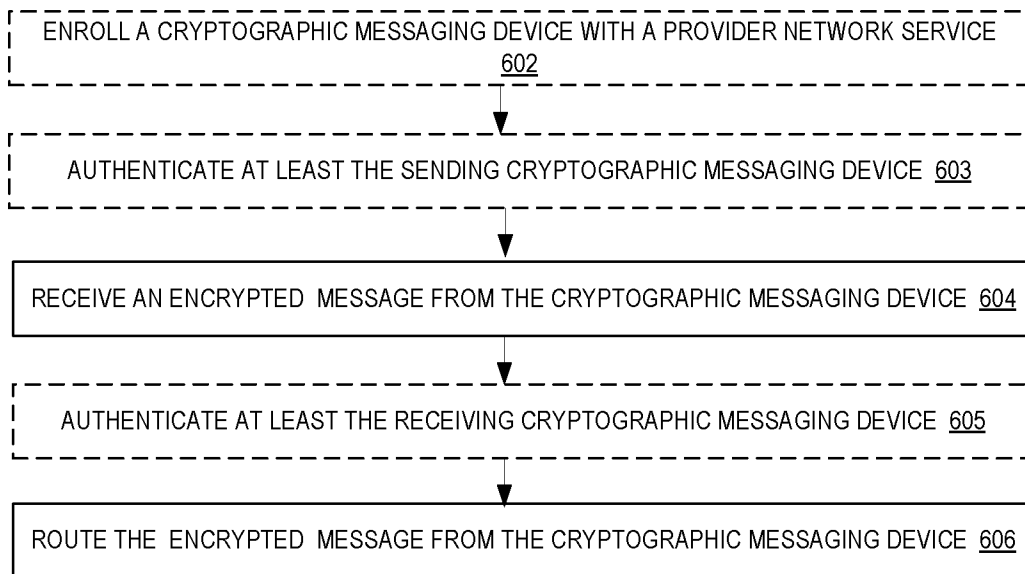
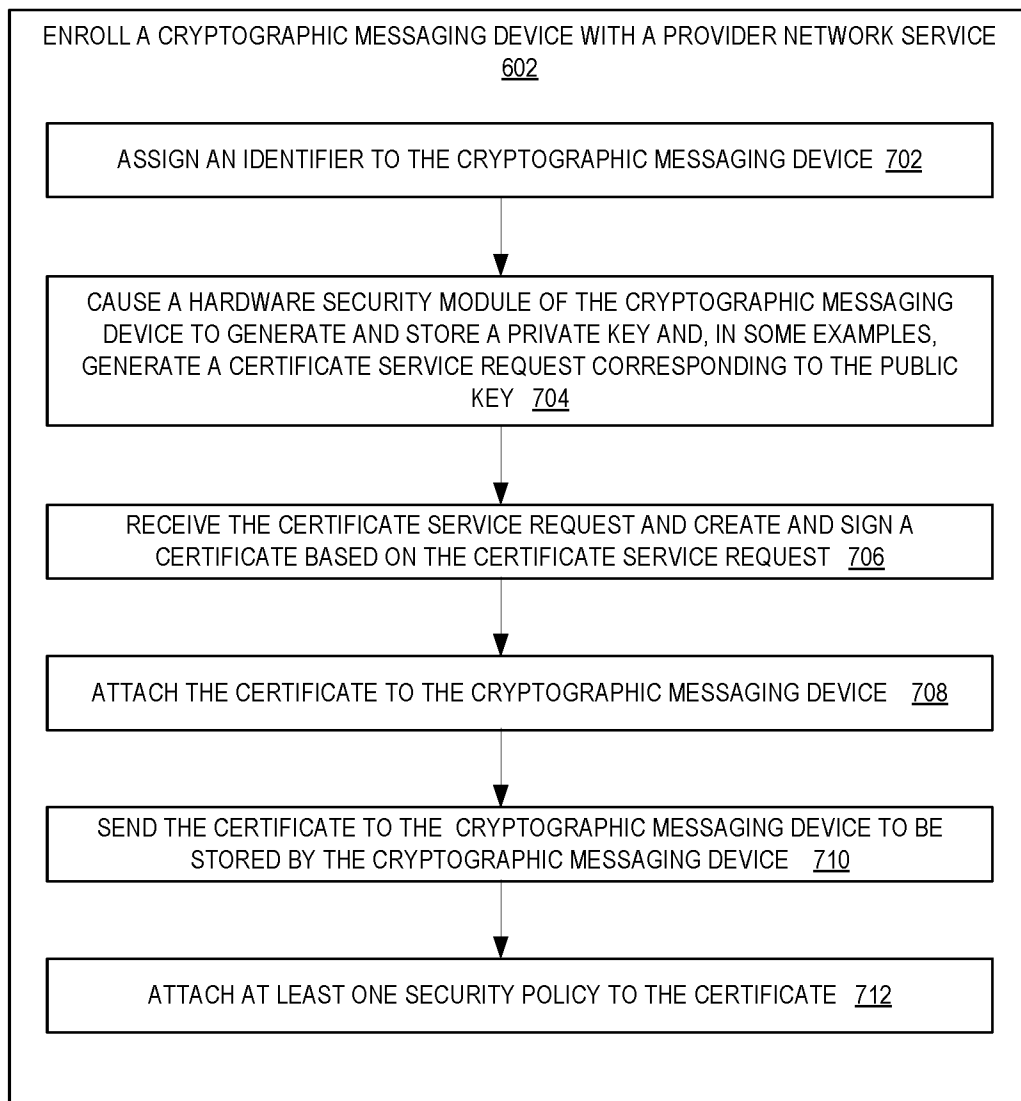
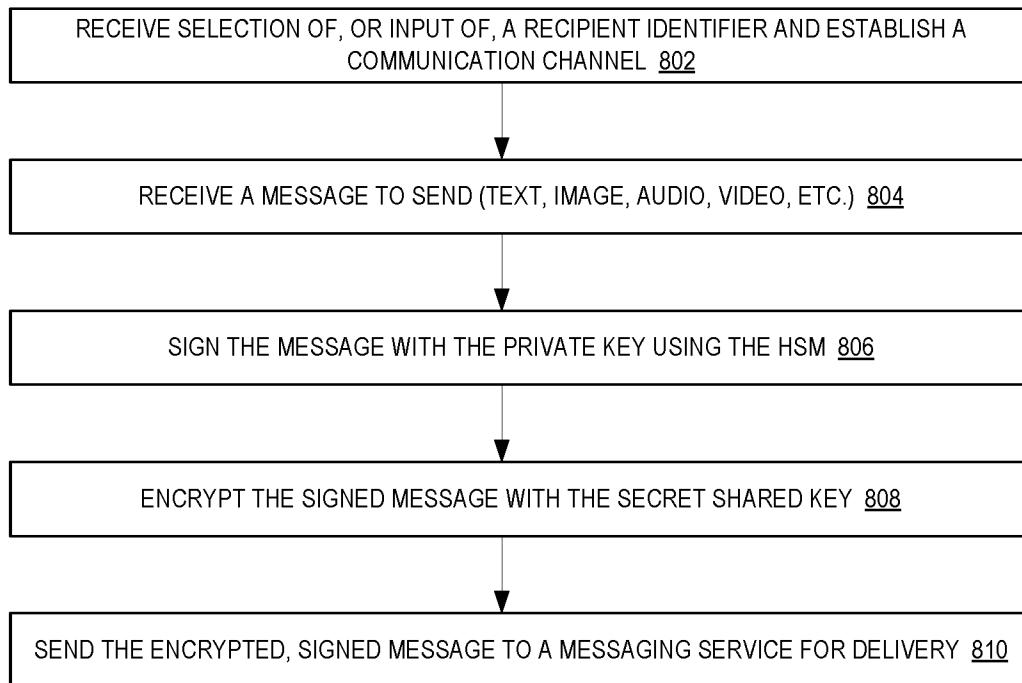
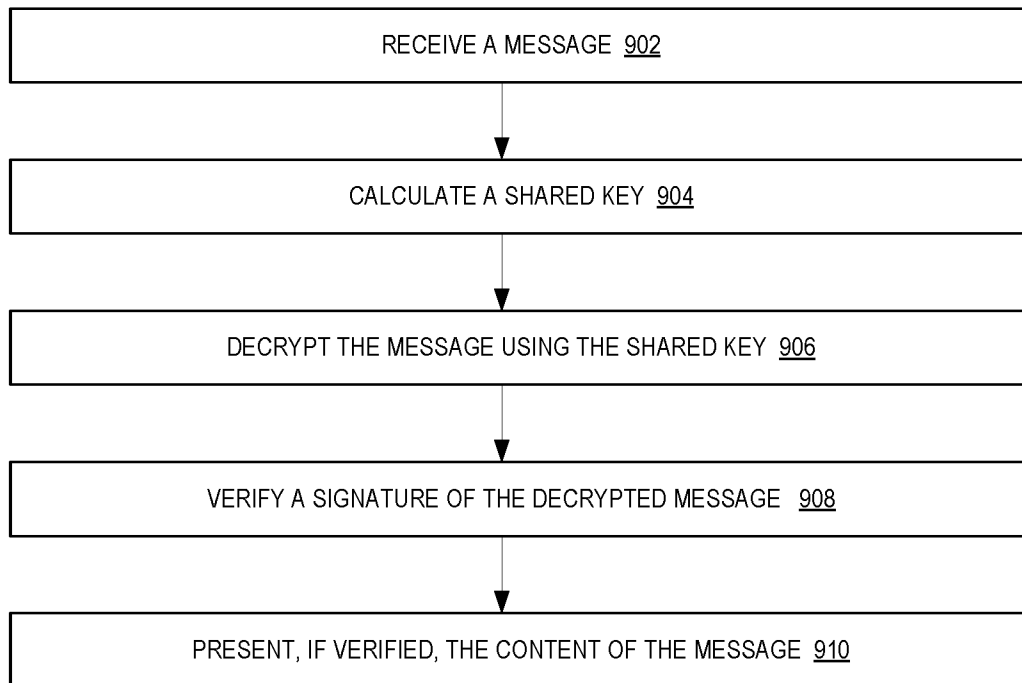


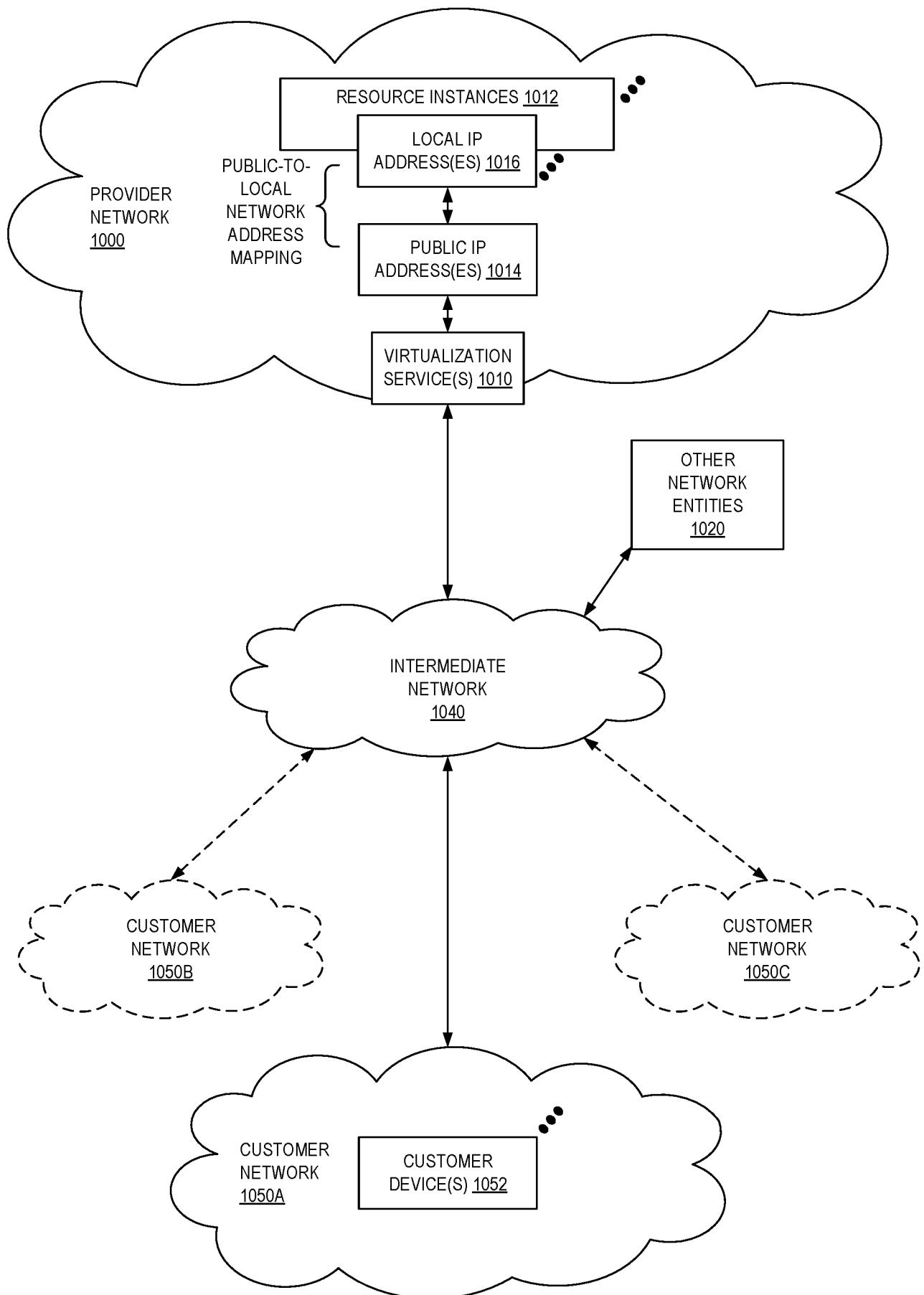
FIG. 5

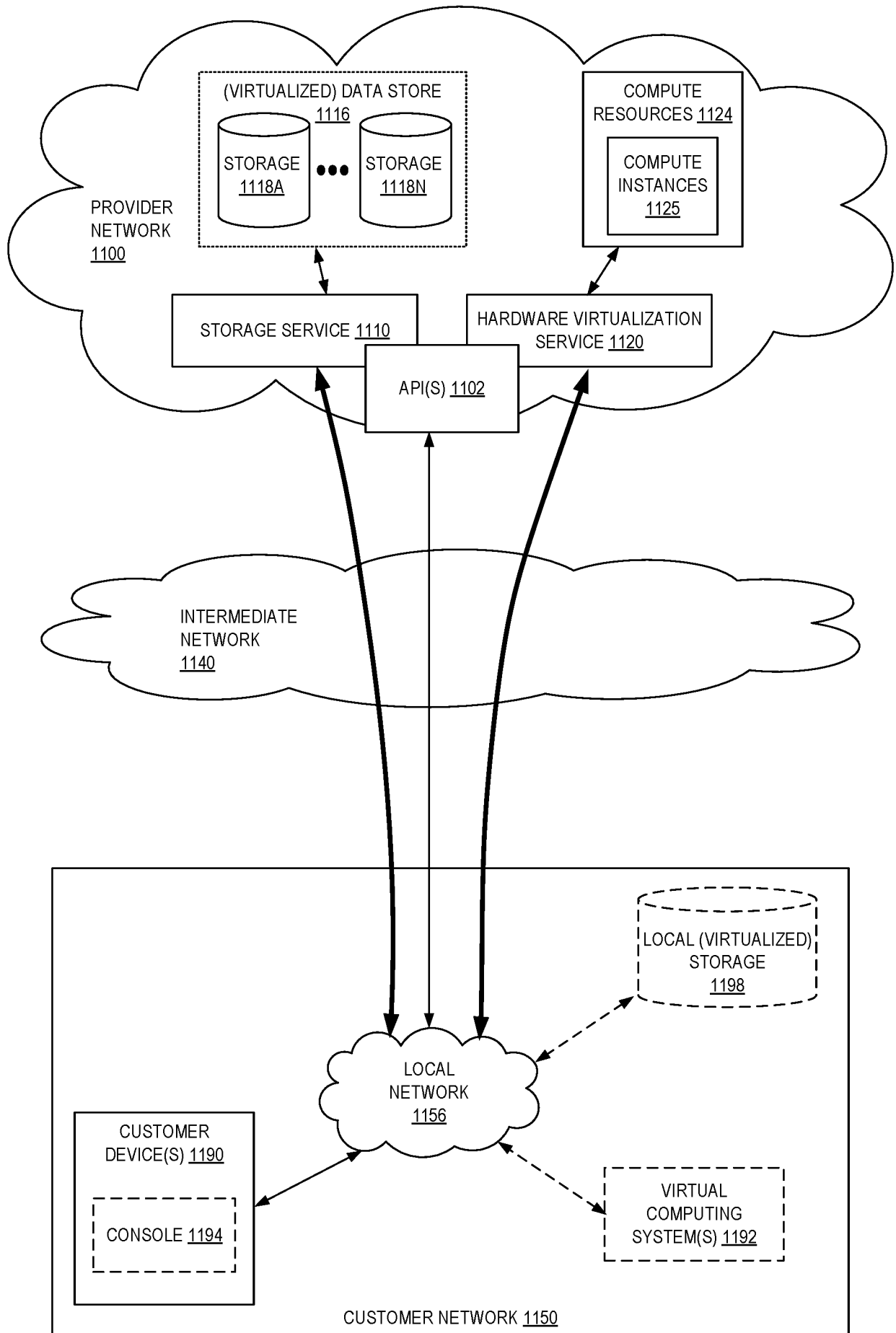
**FIG. 6**

**FIG. 7**

**FIG. 8**

**FIG. 9**

**FIG. 10**

**FIG. 11**

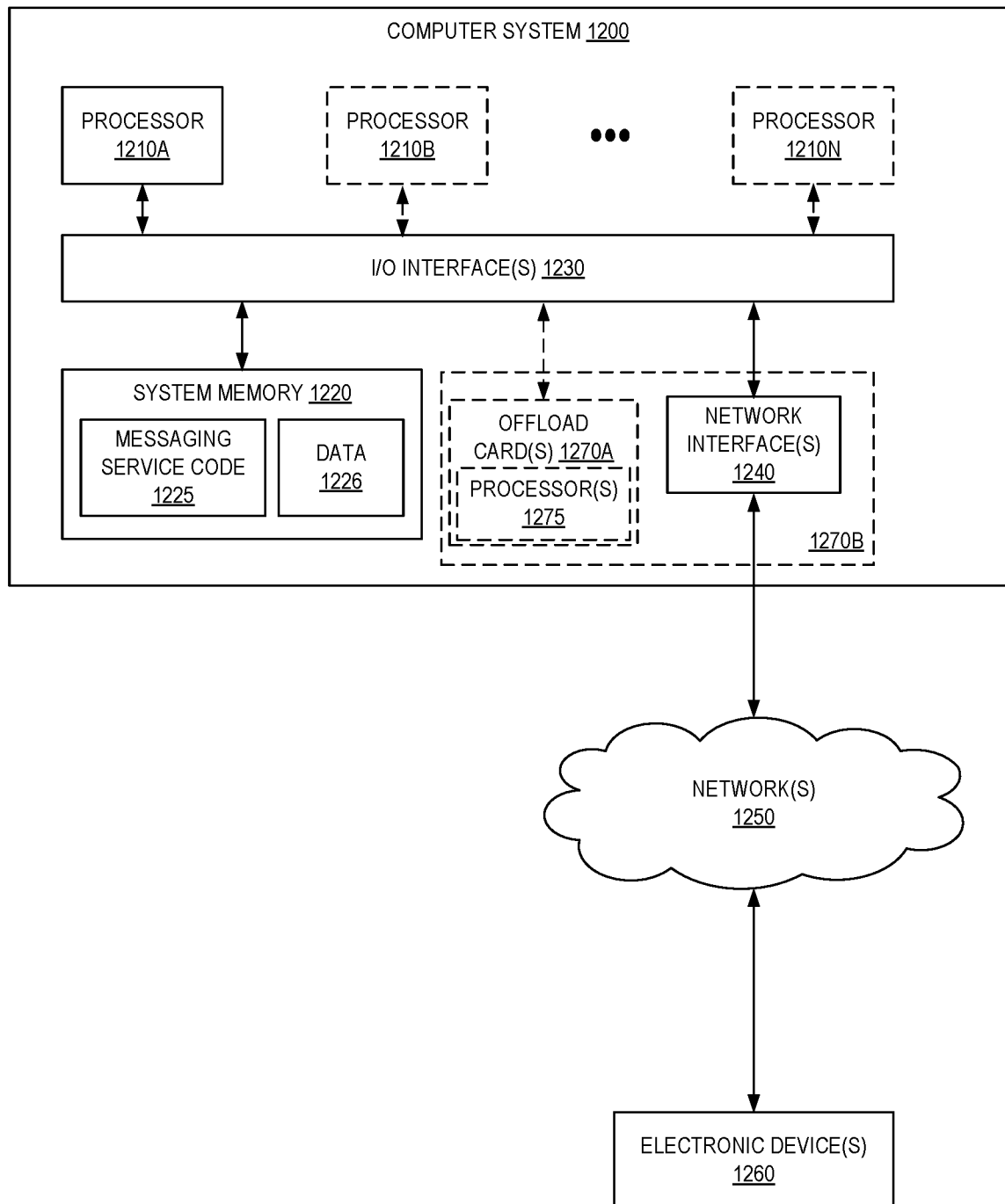


FIG. 12

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2024/035530

A. CLASSIFICATION OF SUBJECT MATTER INV. G06F21/60 H04L9/40 ADD.		
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) G06F 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		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2016/309331 A1 (MOON HWA SHIN [KR]) 20 October 2016 (2016-10-20)	4 - 15
A	paragraphs [0058] - [0064]; figures 1-6 -----	1 - 3
X	Rohde & Schwarz: "TopSec Mobile Abhörsicher telefonieren", , 1 February 2014 (2014-02-01), XP055665000, Retrieved from the Internet: URL:https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_common_library/dl_brochures_and_datasheets/pdf_1/TopSec_Mobile_bro_de_3606-6492-11_v0300.pdf [retrieved on 2020-02-04]	1, 3 - 15
Y	pg. 1, pg. 4, second paragraph, pg. 5, pg. 9 ----- - / - -	2
☒ 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 application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 27 September 2024		Date of mailing of the international search report 09/10/2024
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer Dingel, Janis

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2024/035530

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2016/065376 A1 (SMITH NED M [US] ET AL) 3 March 2016 (2016-03-03)	4 - 15
Y	paragraphs [0029] - [0032], [0047] -	2
A	[0050]; figures 1-8 -----	1, 3
X	US 2022/247729 A1 (HSU YUAN LIOU [TW] ET AL) 4 August 2022 (2022-08-04)	4 - 15
A	paragraphs [0030] - [0032], [0037], [0039] - [0044]; figures 1-6 -----	1 - 3

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2024/035530

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2016309331 A1	20-10-2016	KR 20160123558 A US 2016309331 A1	26-10-2016 20-10-2016
US 2016065376 A1	03-03-2016	CN 106576043 A EP 3186918 A1 US 2016065376 A1 WO 2016032591 A1	19-04-2017 05-07-2017 03-03-2016 03-03-2016
US 2022247729 A1	04-08-2022	CN 114513345 A EP 4037250 A1 JP 2022117456 A TW 202231014 A US 2022247729 A1	17-05-2022 03-08-2022 10-08-2022 01-08-2022 04-08-2022