



(51) International Patent Classification:

G06F 21/60 (2013.01) H04L 9/18 (2006.01)
H04L 9/14 (2006.01) H04W 84/14 (2009.01)

(21) International Application Number:

PCT/US2016/040819

(22) International Filing Date:

1 July 2016 (01.07.2016)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

14/791,371 3 July 2015 (03.07.2015) US
14/791,373 3 July 2015 (03.07.2015) US

(71) Applicant: AFERO, INC. [US/US]; 4970 El Camino Real, Suite 210, Los Altos, California 94022 (US).

(72) Inventors: BRITT, Joe; 4970 El Camino Real, Suite 210, Los Altos, California 94022 (US). ZAKARIA, Omar; 4970 El Camino Real, Suite 210, Los Altos, California 94022 (US). ZIMMERMAN, Scott; 4970 El Camino Real, Suite 210, Los Altos, California 94022 (US).

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

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

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

[Continued on next page]

(54) Title: APPARATUS AND METHOD FOR ESTABLISHING SECURE COMMUNICATION CHANNELS IN AN INTERNET OF THINGS (IOT) SYSTEM

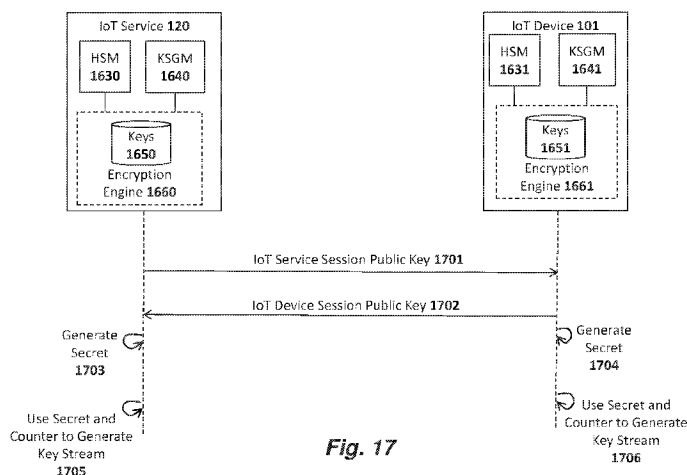


Fig. 17

(57) Abstract: An apparatus and method are described for secure communication between IoT devices and an IoT service. For example, one embodiment of a system comprises: an Internet of Things (IoT) service to establish communication with an IoT device through an IoT hub or a mobile user device; a first encryption engine on the IoT service comprising key generation logic to generate a service public key and a service private key; a second encryption engine on the IoT device comprising key generation logic to generate a device public key and a device private key; the first encryption engine to transmit the service public key to the second encryption engine and the second encryption engine to transmit the device public key to the first encryption engine; the first encryption engine to use the device public key and the service private key to generate a secret; the second encryption engine to use the service public key and the device private key to generate the same secret; and wherein once the secret is generated, the first encryption engine and the second encryption engine encrypt and decrypt data packets transmitted between the first encryption engine and the second encryption engine using the secret or using a data structure derived from the secret.



-
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) — before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))

Published:

- with international search report (Art. 21(3))

APPARATUS AND METHOD FOR ESTABLISHING SECURE
COMMUNICATION CHANNELS IN AN INTERNET OF THINGS (IoT) SYSTEM

BACKGROUND

Field of the Invention

[0001] This invention relates generally to the field of computer systems. More particularly, the invention relates to an apparatus and method for establishing secure communication channels in an IoT system.

Description of the Related Art

[0002] The "Internet of Things" refers to the interconnection of uniquely-identifiable embedded devices within the Internet infrastructure. Ultimately, IoT is expected to result in new, wide-ranging types of applications in which virtually any type of physical thing may provide information about itself or its surroundings and/or may be controlled remotely via client devices over the Internet.

[0003] IoT development and adoption has been slow due to issues related to connectivity, power, and a lack of standardization. For example, one obstacle to IoT development and adoption is that no standard platform exists to allow developers to design and offer new IoT devices and services. In order enter into the IoT market, a developer must design the entire IoT platform from the ground up, including the network protocols and infrastructure, hardware, software and services required to support the desired IoT implementation. As a result, each provider of IoT devices uses proprietary techniques for designing and connecting the IoT devices, making the adoption of multiple types of IoT devices burdensome for end users. Another obstacle to IoT adoption is the difficulty associated with connecting and powering IoT devices. Connecting appliances such as refrigerators, garage door openers, environmental sensors, home security sensors/controllers, etc, for example, requires an electrical source to power each connected IoT device, and such an electrical source is often not conveniently located.

[0004] Another problem which exists is that the wireless technologies used to interconnect IoT devices such as Bluetooth LE are generally short range technologies. Thus, if the data collection hub for an IoT implementation is outside the range of an IoT device, the IoT device will not be able to transmit data to the IoT hub (and vice versa). Consequently, techniques are needed which would allow an IoT device to provide data to an IoT hub (or other IoT device) which is out of range.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] A better understanding of the present invention can be obtained from the following detailed description in conjunction with the following drawings, in which:

[0006] **FIGS. 1A-B** illustrates different embodiments of an IoT system architecture;

[0007] **FIG. 2** illustrates an IoT device in accordance with one embodiment of the invention;

[0008] **FIG. 3** illustrates an IoT hub in accordance with one embodiment of the invention;

[0009] **FIG. 4A-B** illustrate embodiments of the invention for controlling and collecting data from IoT devices, and generating notifications;

[0010] **FIG. 5** illustrates embodiments of the invention for collecting data from IoT devices and generating notifications from an IoT hub and/or IoT service;

[0011] **FIG. 6** illustrates one embodiment of a system in which an intermediary mobile device collects data from a stationary IoT device and provides the data to an IoT hub;

[0012] **FIG. 7** illustrates intermediary connection logic implemented in one embodiment of the invention;

[0013] **FIG. 8** illustrates a method in accordance with one embodiment of the invention;

[0014] **FIG. 9A** illustrates an embodiment in which program code and data updates are provided to the IoT device;

[0015] **FIG. 9B** illustrates an embodiment of a method in which program code and data updates are provided to the IoT device;

[0016] **FIG. 10** illustrates a high level view of one embodiment of a security architecture;

[0017] **FIG. 11** illustrates one embodiment of an architecture in which a subscriber identity module (SIM) is used to store keys on IoT devices;

[0018] **FIG. 12A** illustrates one embodiment in which IoT devices are registered using barcodes or QR codes;

[0019] **FIG. 12B** illustrates one embodiment in which pairing is performed using barcodes or QR codes;

[0020] **FIG. 13** illustrates one embodiment of a method for programming a SIM using an IoT hub;

[0021] **FIG. 14** illustrates one embodiment of a method for registering an IoT device with an IoT hub and IoT service; and

- [0022] FIG. 15 illustrates one embodiment of a method for encrypting data to be transmitted to an IoT device;
- [0023] FIGS. 16A-B illustrate different embodiments of the invention for encrypting data between an IoT service and an IoT device;
- [0024] FIG. 17 illustrates embodiments of the invention for performing a secure key exchange, generating a common secret, and using the secret to generate a key stream;
- [0025] FIG. 18 illustrates a packet structure in accordance with one embodiment of the invention;
- [0026] FIG. 19 illustrates techniques employed in one embodiment for writing and reading data to/from an IoT device without formally pairing with the IoT device;
- [0027] FIG. 20 illustrates an exemplary set of command packets employed in one embodiment of the invention;
- [0028] FIG. 21 illustrates an exemplary sequence of transactions using command packets;
- [0029] FIG. 22 illustrates a method in accordance with one embodiment of the invention; and
- [0030] FIGS. 23A-C illustrate a method for secure pairing in accordance with one embodiment of the invention.

DETAILED DESCRIPTION

[0031] In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the embodiments of the invention described below. It will be apparent, however, to one skilled in the art that the embodiments of the invention may be practiced without some of these specific details. In other instances, well-known structures and devices are shown in block diagram form to avoid obscuring the underlying principles of the embodiments of the invention.

[0032] One embodiment of the invention comprises an Internet of Things (IoT) platform which may be utilized by developers to design and build new IoT devices and applications. In particular, one embodiment includes a base hardware/software platform for IoT devices including a predefined networking protocol stack and an IoT hub through which the IoT devices are coupled to the Internet. In addition, one embodiment includes an IoT service through which the IoT hubs and connected IoT devices may be accessed and managed as described below. In addition, one embodiment of the IoT platform includes an IoT app or Web application (e.g., executed on a client device) to access and configured the IoT service, hub and connected devices. Existing online retailers and

other Website operators may leverage the IoT platform described herein to readily provide unique IoT functionality to existing user bases.

[0033] **Figure 1A** illustrates an overview of an architectural platform on which embodiments of the invention may be implemented. In particular, the illustrated embodiment includes a plurality of IoT devices 101-105 communicatively coupled over local communication channels 130 to a central IoT hub 110 which is itself communicatively coupled to an IoT service 120 over the Internet 220. Each of the IoT devices 101-105 may initially be paired to the IoT hub 110 (e.g., using the pairing techniques described below) in order to enable each of the local communication channels 130. In one embodiment, the IoT service 120 includes an end user database 122 for maintaining user account information and data collected from each user's IoT devices. For example, if the IoT devices include sensors (e.g., temperature sensors, accelerometers, heat sensors, motion detectors, etc), the database 122 may be continually updated to store the data collected by the IoT devices 101-105. The data stored in the database 122 may then be made accessible to the end user via the IoT app or browser installed on the user's device 135 (or via a desktop or other client computer system) and to web clients (e.g., such as websites 130 subscribing to the IoT service 120).

[0034] The IoT devices 101-105 may be equipped with various types of sensors to collect information about themselves and their surroundings and provide the collected information to the IoT service 120, user devices 135 and/or external Websites 130 via the IoT hub 110. Some of the IoT devices 101-105 may perform a specified function in response to control commands sent through the IoT hub 110. Various specific examples of information collected by the IoT devices 101-105 and control commands are provided below. In one embodiment described below, the IoT device 101 is a user input device designed to record user selections and send the user selections to the IoT service 120 and/or Website.

[0035] In one embodiment, the IoT hub 110 includes a cellular radio to establish a connection to the Internet 220 via a cellular service 115 such as a 4G (e.g., Mobile WiMAX, LTE) or 5G cellular data service. Alternatively, or in addition, the IoT hub 110 may include a WiFi radio to establish a WiFi connection through a WiFi access point or router 116 which couples the IoT hub 110 to the Internet (e.g., via an Internet Service Provider providing Internet service to the end user). Of course, it should be noted that the underlying principles of the invention are not limited to any particular type of communication channel or protocol.

[0036] In one embodiment, the IoT devices 101-105 are ultra low-power devices capable of operating for extended periods of time on battery power (e.g., years). To conserve power, the local communication channels 130 may be implemented using a low-power wireless communication technology such as Bluetooth Low Energy (LE). In this embodiment, each of the IoT devices 101-105 and the IoT hub 110 are equipped with Bluetooth LE radios and protocol stacks.

[0037] As mentioned, in one embodiment, the IoT platform includes an IoT app or Web application executed on user devices 135 to allow users to access and configure the connected IoT devices 101-105, IoT hub 110, and/or IoT service 120. In one embodiment, the app or web application may be designed by the operator of a Website 130 to provide IoT functionality to its user base. As illustrated, the Website may maintain a user database 131 containing account records related to each user.

[0038] **Figure 1B** illustrates additional connection options for a plurality of IoT hubs 110-111, 190. In this embodiment a single user may have multiple hubs 110-111 installed onsite at a single user premises 180 (e.g., the user's home or business). This may be done, for example, to extend the wireless range needed to connect all of the IoT devices 101-105. As indicated, if a user has multiple hubs 110, 111 they may be connected via a local communication channel (e.g., Wifi, Ethernet, Power Line Networking, etc). In one embodiment, each of the hubs 110-111 may establish a direct connection to the IoT service 120 through a cellular 115 or WiFi 116 connection (not explicitly shown in **Figure 1B**). Alternatively, or in addition, one of the IoT hubs such as IoT hub 110 may act as a "master" hub which provides connectivity and/or local services to all of the other IoT hubs on the user premises 180, such as IoT hub 111 (as indicated by the dotted line connecting IoT hub 110 and IoT hub 111). For example, the master IoT hub 110 may be the only IoT hub to establish a direct connection to the IoT service 120. In one embodiment, only the "master" IoT hub 110 is equipped with a cellular communication interface to establish the connection to the IoT service 120. As such, all communication between the IoT service 120 and the other IoT hubs 111 will flow through the master IoT hub 110. In this role, the master IoT hub 110 may be provided with additional program code to perform filtering operations on the data exchanged between the other IoT hubs 111 and IoT service 120 (e.g., servicing some data requests locally when possible).

[0039] Regardless of how the IoT hubs 110-111 are connected, in one embodiment, the IoT service 120 will logically associate the hubs with the user and combine all of the

attached IoT devices 101-105 under a single comprehensive user interface, accessible via a user device with the installed app 135 (and/or a browser-based interface).

[0040] In this embodiment, the master IoT hub 110 and one or more slave IoT hubs 111 may connect over a local network which may be a WiFi network 116, an Ethernet network, and/or a using power-line communications (PLC) networking (e.g., where all or portions of the network are run through the user's power lines). In addition, to the IoT hubs 110-111, each of the IoT devices 101-105 may be interconnected with the IoT hubs 110-111 using any type of local network channel such as WiFi, Ethernet, PLC, or Bluetooth LE, to name a few.

[0041] **Figure 1B** also shows an IoT hub 190 installed at a second user premises 181. A virtually unlimited number of such IoT hubs 190 may be installed and configured to collect data from IoT devices 191-192 at user premises around the world. In one embodiment, the two user premises 180-181 may be configured for the same user. For example, one user premises 180 may be the user's primary home and the other user premises 181 may be the user's vacation home. In such a case, the IoT service 120 will logically associate the IoT hubs 110-111, 190 with the user and combine all of the attached IoT devices 101-105, 191-192 under a single comprehensive user interface, accessible via a user device with the installed app 135 (and/or a browser-based interface).

[0042] As illustrated in **Figure 2**, an exemplary embodiment of an IoT device 101 includes a memory 210 for storing program code and data 201-203 and a low power microcontroller 200 for executing the program code and processing the data. The memory 210 may be a volatile memory such as dynamic random access memory (DRAM) or may be a non-volatile memory such as Flash memory. In one embodiment, a non-volatile memory may be used for persistent storage and a volatile memory may be used for execution of the program code and data at runtime. Moreover, the memory 210 may be integrated within the low power microcontroller 200 or may be coupled to the low power microcontroller 200 via a bus or communication fabric. The underlying principles of the invention are not limited to any particular implementation of the memory 210.

[0043] As illustrated, the program code may include application program code 203 defining an application-specific set of functions to be performed by the IoT device 201 and library code 202 comprising a set of predefined building blocks which may be utilized by the application developer of the IoT device 101. In one embodiment, the library code 202 comprises a set of basic functions required to implement an IoT device

such as a communication protocol stack 201 for enabling communication between each IoT device 101 and the IoT hub 110. As mentioned, in one embodiment, the communication protocol stack 201 comprises a Bluetooth LE protocol stack. In this embodiment, Bluetooth LE radio and antenna 207 may be integrated within the low power microcontroller 200. However, the underlying principles of the invention are not limited to any particular communication protocol.

[0044] The particular embodiment shown in **Figure 2** also includes a plurality of input devices or sensors 210 to receive user input and provide the user input to the low power microcontroller, which processes the user input in accordance with the application code 203 and library code 202. In one embodiment, each of the input devices include an LED 209 to provide feedback to the end user.

[0045] In addition, the illustrated embodiment includes a battery 208 for supplying power to the low power microcontroller. In one embodiment, a non-chargeable coin cell battery is used. However, in an alternate embodiment, an integrated rechargeable battery may be used (e.g., rechargeable by connecting the IoT device to an AC power supply (not shown)).

[0046] A speaker 205 is also provided for generating audio. In one embodiment, the low power microcontroller 200 includes audio decoding logic for decoding a compressed audio stream (e.g., such as an MPEG-4/Advanced Audio Coding (AAC) stream) to generate audio on the speaker 205. Alternatively, the low power microcontroller 200 and/or the application code/data 203 may include digitally sampled snippets of audio to provide verbal feedback to the end user as the user enters selections via the input devices 210.

[0047] In one embodiment, one or more other/alternate I/O devices or sensors 250 may be included on the IoT device 101 based on the particular application for which the IoT device 101 is designed. For example, an environmental sensor may be included to measure temperature, pressure, humidity, etc. A security sensor and/or door lock opener may be included if the IoT device is used as a security device. Of course, these examples are provided merely for the purposes of illustration. The underlying principles of the invention are not limited to any particular type of IoT device. In fact, given the highly programmable nature of the low power microcontroller 200 equipped with the library code 202, an application developer may readily develop new application code 203 and new I/O devices 250 to interface with the low power microcontroller for virtually any type of IoT application.

[0048] In one embodiment, the low power microcontroller 200 also includes a secure key store for storing encryption keys for encrypting communications and/or generating signatures. Alternatively, the keys may be secured in a subscriber identify module (SIM).

[0049] A wakeup receiver 207 is included in one embodiment to wake the IoT device from an ultra low power state in which it is consuming virtually no power. In one embodiment, the wakeup receiver 207 is configured to cause the IoT device 101 to exit this low power state in response to a wakeup signal received from a wakeup transmitter 307 configured on the IoT hub 110 as shown in **Figure 3**. In particular, in one embodiment, the transmitter 307 and receiver 207 together form an electrical resonant transformer circuit such as a Tesla coil. In operation, energy is transmitted via radio frequency signals from the transmitter 307 to the receiver 207 when the hub 110 needs to wake the IoT device 101 from a very low power state. Because of the energy transfer, the IoT device 101 may be configured to consume virtually no power when it is in its low power state because it does not need to continually "listen" for a signal from the hub (as is the case with network protocols which allow devices to be awakened via a network signal). Rather, the microcontroller 200 of the IoT device 101 may be configured to wake up after being effectively powered down by using the energy electrically transmitted from the transmitter 307 to the receiver 207.

[0050] As illustrated in **Figure 3**, the IoT hub 110 also includes a memory 317 for storing program code and data 305 and hardware logic 301 such as a microcontroller for executing the program code and processing the data. A wide area network (WAN) interface 302 and antenna 310 couple the IoT hub 110 to the cellular service 115. Alternatively, as mentioned above, the IoT hub 110 may also include a local network interface (not shown) such as a WiFi interface (and WiFi antenna) or Ethernet interface for establishing a local area network communication channel. In one embodiment, the hardware logic 301 also includes a secure key store for storing encryption keys for encrypting communications and generating/verifying signatures. Alternatively, the keys may be secured in a subscriber identify module (SIM).

[0051] A local communication interface 303 and antenna 311 establishes local communication channels with each of the IoT devices 101-105. As mentioned above, in one embodiment, the local communication interface 303/antenna 311 implements the Bluetooth LE standard. However, the underlying principles of the invention are not limited to any particular protocols for establishing the local communication channels with the IoT devices 101-105. Although illustrated as separate units in **Figure 3**, the WAN

interface 302 and/or local communication interface 303 may be embedded within the same chip as the hardware logic 301.

[0052] In one embodiment, the program code and data includes a communication protocol stack 308 which may include separate stacks for communicating over the local communication interface 303 and the WAN interface 302. In addition, device pairing program code and data 306 may be stored in the memory to allow the IoT hub to pair with new IoT devices. In one embodiment, each new IoT device 101-105 is assigned a unique code which is communicated to the IoT hub 110 during the pairing process. For example, the unique code may be embedded in a barcode on the IoT device and may be read by the barcode reader 106 or may be communicated over the local communication channel 130. In an alternate embodiment, the unique ID code is embedded magnetically on the IoT device and the IoT hub has a magnetic sensor such as an radio frequency ID (RFID) or near field communication (NFC) sensor to detect the code when the IoT device 101 is moved within a few inches of the IoT hub 110.

[0053] In one embodiment, once the unique ID has been communicated, the IoT hub 110 may verify the unique ID by querying a local database (not shown), performing a hash to verify that the code is acceptable, and/or communicating with the IoT service 120, user device 135 and/or Website 130 to validate the ID code. Once validated, in one embodiment, the IoT hub 110 pairs the IoT device 101 and stores the pairing data in memory 317 (which, as mentioned, may include non-volatile memory). Once pairing is complete, the IoT hub 110 may connect with the IoT device 101 to perform the various IoT functions described herein.

[0054] In one embodiment, the organization running the IoT service 120 may provide the IoT hub 110 and a basic hardware/software platform to allow developers to easily design new IoT services. In particular, in addition to the IoT hub 110, developers may be provided with a software development kit (SDK) to update the program code and data 305 executed within the hub 110. In addition, for IoT devices 101, the SDK may include an extensive set of library code 202 designed for the base IoT hardware (e.g., the low power microcontroller 200 and other components shown in Figure 2) to facilitate the design of various different types of applications 101. In one embodiment, the SDK includes a graphical design interface in which the developer needs only to specify input and outputs for the IoT device. All of the networking code, including the communication stack 201 that allows the IoT device 101 to connect to the hub 110 and the service 120, is already in place for the developer. In addition, in one embodiment,

the SDK also includes a library code base to facilitate the design of apps for mobile devices (e.g., iPhone and Android devices).

[0055] In one embodiment, the IoT hub 110 manages a continuous bi-directional stream of data between the IoT devices 101-105 and the IoT service 120. In circumstances where updates to/from the IoT devices 101-105 are required in real time (e.g., where a user needs to view the current status of security devices or environmental readings), the IoT hub may maintain an open TCP socket to provide regular updates to the user device 135 and/or external Websites 130. The specific networking protocol used to provide updates may be tweaked based on the needs of the underlying application. For example, in some cases, where may not make sense to have a continuous bi-directional stream, a simple request/response protocol may be used to gather information when needed.

[0056] In one embodiment, both the IoT hub 110 and the IoT devices 101-105 are automatically upgradeable over the network. In particular, when a new update is available for the IoT hub 110 it may automatically download and install the update from the IoT service 120. It may first copy the updated code into a local memory, run and verify the update before swapping out the older program code. Similarly, when updates are available for each of the IoT devices 101-105, they may initially be downloaded by the IoT hub 110 and pushed out to each of the IoT devices 101-105. Each IoT device 101-105 may then apply the update in a similar manner as described above for the IoT hub and report back the results of the update to the IoT hub 110. If the update is successful, then the IoT hub 110 may delete the update from its memory and record the latest version of code installed on each IoT device (e.g., so that it may continue to check for new updates for each IoT device).

[0057] In one embodiment, the IoT hub 110 is powered via A/C power. In particular, the IoT hub 110 may include a power unit 390 with a transformer for transforming A/C voltage supplied via an A/C power cord to a lower DC voltage.

[0058] **Figure 4A** illustrates one embodiment of the invention for performing universal remote control operations using the IoT system. In particular, in this embodiment, a set of IoT devices 101-103 are equipped with infrared (IR) and/or radio frequency (RF) blasters 401-403, respectively, for transmitting remote control codes to control various different types of electronics equipment including air conditioners/heaters 430, lighting systems 431, and audiovisual equipment 432 (to name just a few). In the embodiment shown in **Figure 4A**, the IoT devices 101-103 are

also equipped with sensors 404-406, respectively, for detecting the operation of the devices which they control, as described below.

[0059] For example, sensor 404 in IoT device 101 may be a temperature and/or humidity sensor for sensing the current temperature/humidity and responsively controlling the air conditioner/heater 430 based on a current desired temperature. In this embodiment, the air conditioner/heater 430 is one which is designed to be controlled via a remote control device (typically a remote control which itself has a temperature sensor embedded therein). In one embodiment, the user provides the desired temperature to the IoT hub 110 via an app or browser installed on a user device 135. Control logic 412 executed on the IoT hub 110 receives the current temperature/humidity data from the sensor 404 and responsively transmits commands to the IoT device 101 to control the IR/RF blaster 401 in accordance with the desired temperature/humidity. For example, if the temperature is below the desired temperature, then the control logic 412 may transmit a command to the air conditioner/heater via the IR/RF blaster 401 to increase the temperature (e.g., either by turning off the air conditioner or turning on the heater). The command may include the necessary remote control code stored in a database 413 on the IoT hub 110. Alternatively, or in addition, the IoT service 421 may implement control logic 421 to control the electronics equipment 430-432 based on specified user preferences and stored control codes 422.

[0060] IoT device 102 in the illustrated example is used to control lighting 431. In particular, sensor 405 in IoT device 102 may be a photosensor or photodetector configured to detect the current brightness of the light being produced by a light fixture 431 (or other lighting apparatus). The user may specify a desired lighting level (including an indication of ON or OFF) to the IoT hub 110 via the user device 135. In response, the control logic 412 will transmit commands to the IR/RF blaster 402 to control the current brightness level of the lights 431 (e.g., increasing the lighting if the current brightness is too low or decreasing the lighting if the current brightness is too high; or simply turning the lights ON or OFF).

[0061] IoT device 103 in the illustrated example is configured to control audiovisual equipment 432 (e.g., a television, A/V receiver, cable/satellite receiver, AppleTV™, etc). Sensor 406 in IoT device 103 may be an audio sensor (e.g., a microphone and associated logic) for detecting a current ambient volume level and/or a photosensor to detect whether a television is on or off based on the light generated by the television (e.g., by measuring the light within a specified spectrum). Alternatively, sensor 406 may

include a temperature sensor connected to the audiovisual equipment to detect whether the audio equipment is on or off based on the detected temperature. Once again, in response to user input via the user device 135, the control logic 412 may transmit commands to the audiovisual equipment via the IR blaster 403 of the IoT device 103.

[0062] It should be noted that the foregoing are merely illustrative examples of one embodiment of the invention. The underlying principles of the invention are not limited to any particular type of sensors or equipment to be controlled by IoT devices.

[0063] In an embodiment in which the IoT devices 101-103 are coupled to the IoT hub 110 via a Bluetooth LE connection, the sensor data and commands are sent over the Bluetooth LE channel. However, the underlying principles of the invention are not limited to Bluetooth LE or any other communication standard.

[0064] In one embodiment, the control codes required to control each of the pieces of electronics equipment are stored in a database 413 on the IoT hub 110 and/or a database 422 on the IoT service 120. As illustrated in **Figure 4B**, the control codes may be provided to the IoT hub 110 from a master database of control codes 422 for different pieces of equipment maintained on the IoT service 120. The end user may specify the types of electronic (or other) equipment to be controlled via the app or browser executed on the user device 135 and, in response, a remote control code learning module 491 on the IoT hub may retrieve the required IR/RF codes from the remote control code database 492 on the IoT service 120 (e.g., identifying each piece of electronic equipment with a unique ID).

[0065] In addition, in one embodiment, the IoT hub 110 is equipped with an IR/RF interface 490 to allow the remote control code learning module 491 to "learn" new remote control codes directly from the original remote control 495 provided with the electronic equipment. For example, if control codes for the original remote control provided with the air conditioner 430 is not included in the remote control database, the user may interact with the IoT hub 110 via the app/browser on the user device 135 to teach the IoT hub 110 the various control codes generated by the original remote control (e.g., increase temperature, decrease temperature, etc). Once the remote control codes are learned they may be stored in the control code database 413 on the IoT hub 110 and/or sent back to the IoT service 120 to be included in the central remote control code database 492 (and subsequently used by other users with the same air conditioner unit 430).

[0066] In one embodiment, each of the IoT devices 101-103 have an extremely small form factor and may be affixed on or near their respective electronics equipment

430-432 using double-sided tape, a small nail, a magnetic attachment, etc. For control of a piece of equipment such as the air conditioner 430, it would be desirable to place the IoT device 101 sufficiently far away so that the sensor 404 can accurately measure the ambient temperature in the home (e.g., placing the IoT device directly on the air conditioner would result in a temperature measurement which would be too low when the air conditioner was running or too high when the heater was running). In contrast, the IoT device 102 used for controlling lighting may be placed on or near the lighting fixture 431 for the sensor 405 to detect the current lighting level.

[0067] In addition to providing general control functions as described, one embodiment of the IoT hub 110 and/or IoT service 120 transmits notifications to the end user related to the current status of each piece of electronics equipment. The notifications, which may be text messages and/or app-specific notifications, may then be displayed on the display of the user's mobile device 135. For example, if the user's air conditioner has been on for an extended period of time but the temperature has not changed, the IoT hub 110 and/or IoT service 120 may send the user a notification that the air conditioner is not functioning properly. If the user is not home (which may be detected via motion sensors or based on the user's current detected location), and the sensors 406 indicate that audiovisual equipment 430 is on or sensors 405 indicate that the lights are on, then a notification may be sent to the user, asking if the user would like to turn off the audiovisual equipment 432 and/or lights 431. The same type of notification may be sent for any equipment type.

[0068] Once the user receives a notification, he/she may remotely control the electronics equipment 430-432 via the app or browser on the user device 135. In one embodiment, the user device 135 is a touchscreen device and the app or browser displays an image of a remote control with user-selectable buttons for controlling the equipment 430-432. Upon receiving a notification, the user may open the graphical remote control and turn off or adjust the various different pieces of equipment. If connected via the IoT service 120, the user's selections may be forwarded from the IoT service 120 to the IoT hub 110 which will then control the equipment via the control logic 412. Alternatively, the user input may be sent directly to the IoT hub 110 from the user device 135.

[0069] In one embodiment, the user may program the control logic 412 on the IoT hub 110 to perform various automatic control functions with respect to the electronics equipment 430-432. In addition to maintaining a desired temperature, brightness level, and volume level as described above, the control logic 412 may automatically turn off

the electronics equipment if certain conditions are detected. For example, if the control logic 412 detects that the user is not home and that the air conditioner is not functioning, it may automatically turn off the air conditioner. Similarly, if the user is not home, and the sensors 406 indicate that audiovisual equipment 430 is on or sensors 405 indicate that the lights are on, then the control logic 412 may automatically transmit commands via the IR/RF blasters 403 and 402, to turn off the audiovisual equipment and lights, respectively.

[0070] **Figure 5** illustrates additional embodiments of IoT devices 104-105 equipped with sensors 503-504 for monitoring electronic equipment 530-531. In particular, the IoT device 104 of this embodiment includes a temperature sensor 503 which may be placed on or near a stove 530 to detect when the stove has been left on. In one embodiment, the IoT device 104 transmits the current temperature measured by the temperature sensor 503 to the IoT hub 110 and/or the IoT service 120. If the stove is detected to be on for more than a threshold time period (e.g., based on the measured temperature), then control logic 512 may transmit a notification to the end user's device 135 informing the user that the stove 530 is on. In addition, in one embodiment, the IoT device 104 may include a control module 501 to turn off the stove, either in response to receiving an instruction from the user or automatically (if the control logic 512 is programmed to do so by the user). In one embodiment, the control logic 501 comprises a switch to cut off electricity or gas to the stove 530. However, in other embodiments, the control logic 501 may be integrated within the stove itself.

[0071] **Figure 5** also illustrates an IoT device 105 with a motion sensor 504 for detecting the motion of certain types of electronics equipment such as a washer and/or dryer. Another sensor that may be used is an audio sensor (e.g., microphone and logic) for detecting an ambient volume level. As with the other embodiments described above, this embodiment may transmit notifications to the end user if certain specified conditions are met (e.g., if motion is detected for an extended period of time, indicating that the washer/dryer are not turning off). Although not shown in **Figure 5**, IoT device 105 may also be equipped with a control module to turn off the washer/dryer 531 (e.g., by switching off electric/gas), automatically, and/or in response to user input.

[0072] In one embodiment, a first IoT device with control logic and a switch may be configured to turn off all power in the user's home and a second IoT device with control logic and a switch may be configured to turn off all gas in the user's home. IoT devices with sensors may then be positioned on or near electronic or gas-powered equipment in the user's home. If the user is notified that a particular piece of equipment has been left

on (e.g., the stove 530), the user may then send a command to turn off all electricity or gas in the home to prevent damage. Alternatively, the control logic 512 in the IoT hub 110 and/or the IoT service 120 may be configured to automatically turn off electricity or gas in such situations.

[0073] In one embodiment, the IoT hub 110 and IoT service 120 communicate at periodic intervals. If the IoT service 120 detects that the connection to the IoT hub 110 has been lost (e.g., by failing to receive a request or response from the IoT hub for a specified duration), it will communicate this information to the end user's device 135 (e.g., by sending a text message or app-specific notification).

APPARATUS AND METHOD FOR COMMUNICATING DATA THROUGH AN INTERMEDIARY DEVICE

[0074] As mentioned above, because the wireless technologies used to interconnect IoT devices such as Bluetooth LE are generally short range technologies, if the hub for an IoT implementation is outside the range of an IoT device, the IoT device will not be able to transmit data to the IoT hub (and vice versa).

[0075] To address this deficiency, one embodiment of the invention provides a mechanism for an IoT device which is outside of the wireless range of the IoT hub to periodically connect with one or more mobile devices when the mobile devices are within range. Once connected, the IoT device can transmit any data which needs to be provided to the IoT hub to the mobile device which then forwards the data to the IoT hub.

[0076] As illustrated in **Figure 6** one embodiment includes an IoT hub 110, an IoT device 601 which is out of range of the IoT hub 110 and a mobile device 611. The out of range IoT device 601 may include any form of IoT device capable of collecting and communicating data. For example, the IoT device 601 may comprise a data collection device configured within a refrigerator to monitor the food items available in the refrigerator, the users who consume the food items, and the current temperature. Of course, the underlying principles of the invention are not limited to any particular type of IoT device. The techniques described herein may be implemented using any type of IoT device including those used to collect and transmit data for smart meters, stoves, washers, dryers, lighting systems, HVAC systems, and audiovisual equipment, to name just a few.

[0077] Moreover, the mobile device In operation, the IoT device 611 illustrated in **Figure 6** may be any form of mobile device capable of communicating and storing data. For example, in one embodiment, the mobile device 611 is a smartphone with an app

installed thereon to facilitate the techniques described herein. In another embodiment, the mobile device 611 comprises a wearable device such as a communication token affixed to a neckless or bracelet, a smartwatch or a fitness device. The wearable token may be particularly useful for elderly users or other users who do not own a smartphone device.

[0078] In operation, the out of range IoT device 601 may periodically or continually check for connectivity with a mobile device 611. Upon establishing a connection (e.g., as the result of the user moving within the vicinity of the refrigerator) any collected data 605 on the IoT device 601 is automatically transmitted to a temporary data repository 615 on the mobile device 611. In one embodiment, the IoT device 601 and mobile device 611 establish a local wireless communication channel using a low power wireless standard such as BTLE. In such a case, the mobile device 611 may initially be paired with the IoT device 601 using known pairing techniques.

[0079] Once the data has been transferred to the temporary data repository, the mobile device 611 will transmit the data once communication is established with the IoT hub 110 (e.g., when the user walks within the range of the IoT hub 110). The IoT hub may then store the data in a central data repository 413 and/or send the data over the Internet to one or more services and/or other user devices. In one embodiment, the mobile device 611 may use a different type of communication channel to provide the data to the IoT hub 110 (potentially a higher power communication channel such as WiFi).

[0080] The out of range IoT device 601, the mobile device 611, and the IoT hub may all be configured with program code and/or logic to implement the techniques described herein. As illustrated in **Figure 7**, for example, the IoT device 601 may be configured with intermediary connection logic and/or application, the mobile device 611 may be configured with an intermediary connection logic/application, and the IoT hub 110 may be configured with an intermediary connection logic/application 721 to perform the operations described herein. The intermediary connection logic/application on each device may be implemented in hardware, software, or any combination thereof. In one embodiment, the intermediary connection logic/application 701 of the IoT device 601 searches and establishes a connection with the intermediary connection logic/application 711 on the mobile device (which may be implemented as a device app) to transfer the data to the temporary data repository 615. The intermediary connection logic/application 701 on the mobile device 611 then forwards the data to the

intermediary connection logic/application on the IoT hub, which stores the data in the central data repository 413.

[0081] As illustrated in **Figure 7**, the intermediary connection logic/applications 701, 711, 721, on each device may be configured based on the application at hand. For example, for a refrigerator, the connection logic/application 701 may only need to transmit a few packets on a periodic basis. For other applications (e.g., temperature sensors), the connection logic/application 701 may need to transmit more frequent updates.

[0082] Rather than a mobile device 611, in one embodiment, the IoT device 601 may be configured to establish a wireless connection with one or more intermediary IoT devices, which are located within range of the IoT hub 110. In this embodiment, any IoT devices 601 out of range of the IoT hub may be linked to the hub by forming a "chain" using other IoT devices.

[0083] In addition, while only a single mobile device 611 is illustrated in **Figures 6-7** for simplicity, in one embodiment, multiple such mobile devices of different users may be configured to communicate with the IoT device 601. Moreover, the same techniques may be implemented for multiple other IoT devices, thereby forming an intermediary device data collection system across the entire home.

[0084] Moreover, in one embodiment, the techniques described herein may be used to collect various different types of pertinent data. For example, in one embodiment, each time the mobile device 611 connects with the IoT device 601, the identity of the user may be included with the collected data 605. In this manner, the IoT system may be used to track the behavior of different users within the home. For example, if used within a refrigerator, the collected data 605 may then include the identify of each user who passes by fridge, each user who opens the fridge, and the specific food items consumed by each user. Different types of data may be collected from other types of IoT devices. Using this data the system is able to determine, for example, which user washes clothes, which user watches TV on a given day, the times at which each user goes to sleep and wakes up, etc. All of this crowd-sourced data may then be compiled within the data repository 413 of the IoT hub and/or forwarded to an external service or user.

[0085] Another beneficial application of the techniques described herein is for monitoring elderly users who may need assistance. For this application, the mobile device 611 may be a very small token worn by the elderly user to collect the information in different rooms of the user's home. Each time the user opens the refrigerator, for

example, this data will be included with the collected data 605 and transferred to the IoT hub 110 via the token. The IoT hub may then provide the data to one or more external users (e.g., the children or other individuals who care for the elderly user). If data has not been collected for a specified period of time (e.g., 12 hours), then this means that the elderly user has not been moving around the home and/or has not been opening the refrigerator. The IoT hub 110 or an external service connected to the IoT hub may then transmit an alert notification to these other individuals, informing them that they should check on the elderly user. In addition, the collected data 605 may include other pertinent information such as the food being consumed by the user and whether a trip to the grocery store is needed, whether and how frequently the elderly user is watching TV, the frequency with which the elderly user washes clothes, etc.

[0086] In another implementation, the if there is a problem with an electronic device such as a washer, refrigerator, HVAC system, etc, the collected data may include an indication of a part that needs to be replaced. In such a case, a notification may be sent to a technician with a request to fix the problem. The technician may then arrive at the home with the needed replacement part.

[0087] A method in accordance with one embodiment of the invention is illustrated in **Figure 8**. The method may be implemented within the context of the architectures described above, but is not limited to any particular architecture.

[0088] At 801, an IoT device which is out of range of the IoT hub periodically collects data (e.g., opening of the refrigerator door, food items used, etc). At 802 the IoT device periodically or continually checks for connectivity with a mobile device (e.g., using standard local wireless techniques for establishing a connection such as those specified by the BTLE standard). If the connection to the mobile device is established, determined at 802, then at 803, the collected data is transferred to the mobile device at 803. At 804, the mobile device transfers the data to the IoT hub, an external service and/or a user. As mentioned, the mobile device may transmit the data immediately if it is already connected (e.g., via a WiFi link).

[0089] In addition to collecting data from IoT devices, in one embodiment, the techniques described herein may be used to update or otherwise provide data to IoT devices. One example is shown in **Figure 9A**, which shows an IoT hub 110 with program code updates 901 that need to be installed on an IoT device 601 (or a group of such IoT devices). The program code updates may include system updates, patches, configuration data and any other data needed for the IoT device to operate as desired by the user. In one embodiment, the user may specify configuration options for the IoT

device 601 via a mobile device or computer which are then stored on the IoT hub 110 and provided to the IoT device using the techniques described herein. Specifically, in one embodiment, the intermediary connection logic/application 721 on the IoT hub 110 communicates with the intermediary connection logic/application 711 on the mobile device 611 to store the program code updates within a temporary storage 615. When the mobile device 611 enters the range of the IoT device 601, the intermediary connection logic/application 711 on the mobile device 611 connects with the intermediary/connection logic/application 701 on the IoT device 601 to provide the program code updates to the device. In one embodiment, the IoT device 601 may then enter into an automated update process to install the new program code updates and/or data.

[0090] A method for updating an IoT device is shown in **Figure 9B**. The method may be implemented within the context of the system architectures described above, but is not limited to any particular system architectures.

[0091] At 900 new program code or data updates are made available on the IoT hub and/or an external service (e.g., coupled to the mobile device over the Internet). At 901, the mobile device receives and stores the program code or data updates on behalf of the IoT device. The IoT device and/or mobile device periodically check to determine whether a connection has been established at 902. If a connection is established, determined at 903, then at 904 the updates are transferred to the IoT device and installed.

EMBODIMENTS FOR IMPROVED SECURITY

[0092] In one embodiment, the low power microcontroller 200 of each IoT device 101 and the low power logic/microcontroller 301 of the IoT hub 110 include a secure key store for storing encryption keys used by the embodiments described below (see, e.g., **Figures 10-15** and associated text). Alternatively, the keys may be secured in a subscriber identify module (SIM) as discussed below.

[0093] **Figure 10** illustrates a high level architecture which uses public key infrastructure (PKI) techniques and/or symmetric key exchange/encryption techniques to encrypt communications between the IoT Service 120, the IoT hub 110 and the IoT devices 101-102.

[0094] Embodiments which use public/private key pairs will first be described, followed by embodiments which use symmetric key exchange/encryption techniques. In particular, in an embodiment which uses PKI, a unique public/private key pair is associated with each IoT device 101-102, each IoT hub 110 and the IoT service 120. In

one embodiment, when a new IoT hub 110 is set up, its public key is provided to the IoT service 120 and when a new IoT device 101 is set up, its public key is provided to both the IoT hub 110 and the IoT service 120. Various techniques for securely exchanging the public keys between devices are described below. In one embodiment, all public keys are signed by a master key known to all of the receiving devices (i.e., a form of certificate) so that any receiving device can verify the validity of the public keys by validating the signatures. Thus, these certificates would be exchanged rather than merely exchanging the raw public keys.

[0095] As illustrated, in one embodiment, each IoT device 101, 102 includes a secure key storage 1001, 1003, respectively, for securely storing each device's private key. Security logic 1002, 1304 then utilizes the securely stored private keys to perform the encryption/decryption operations described herein. Similarly, the IoT hub 110 includes a secure storage 1011 for storing the IoT hub private key and the public keys of the IoT devices 101-102 and the IoT service 120; as well as security logic 1012 for using the keys to perform encryption/decryption operations. Finally, the IoT service 120 may include a secure storage 1021 for securely storing its own private key, the public keys of various IoT devices and IoT hubs, and a security logic 1013 for using the keys to encrypt/decrypt communication with IoT hubs and devices. In one embodiment, when the IoT hub 110 receives a public key certificate from an IoT device it can verify it (e.g., by validating the signature using the master key as described above), and then extract the public key from within it and store that public key in its secure key store 1011.

[0096] By way of example, in one embodiment, when the IoT service 120 needs to transmit a command or data to an IoT device 101 (e.g., a command to unlock a door, a request to read a sensor, data to be processed/displayed by the IoT device, etc) the security logic 1013 encrypts the data/command using the public key of the IoT device 101 to generate an encrypted IoT device packet. In one embodiment, it then encrypts the IoT device packet using the public key of the IoT hub 110 to generate an IoT hub packet and transmits the IoT hub packet to the IoT hub 110. In one embodiment, the service 120 signs the encrypted message with its private key or the master key mentioned above so that the device 101 can verify it is receiving an unaltered message from a trusted source. The device 101 may then validate the signature using the public key corresponding to the private key and/or the master key. As mentioned above, symmetric key exchange/encryption techniques may be used instead of public/private key encryption. In these embodiments, rather than privately storing one key and

providing a corresponding public key to other devices, the devices may each be provided with a copy of the same symmetric key to be used for encryption and to validate signatures. One example of a symmetric key algorithm is the Advanced Encryption Standard (AES), although the underlying principles of the invention are not limited to any type of specific symmetric keys.

[0097] Using a symmetric key implementation, each device 101 enters into a secure key exchange protocol to exchange a symmetric key with the IoT hub 110. A secure key provisioning protocol such as the Dynamic Symmetric Key Provisioning Protocol (DSKPP) may be used to exchange the keys over a secure communication channel (see, e.g., Request for Comments (RFC) 6063). However, the underlying principles of the invention are not limited to any particular key provisioning protocol.

[0098] Once the symmetric keys have been exchanged, they may be used by each device 101 and the IoT hub 110 to encrypt communications. Similarly, the IoT hub 110 and IoT service 120 may perform a secure symmetric key exchange and then use the exchanged symmetric keys to encrypt communications. In one embodiment a new symmetric key is exchanged periodically between the devices 101 and the hub 110 and between the hub 110 and the IoT service 120. In one embodiment, a new symmetric key is exchanged with each new communication session between the devices 101, the hub 110, and the service 120 (e.g., a new key is generated and securely exchanged for each communication session). In one embodiment, if the security module 1012 in the IoT hub is trusted, the service 120 could negotiate a session key with the hub security module 1312 and then the security module 1012 would negotiate a session key with each device 120. Messages from the service 120 would then be decrypted and verified in the hub security module 1012 before being re-encrypted for transmission to the device 101.

[0099] In one embodiment, to prevent a compromise on the hub security module 1012 a one-time (permanent) installation key may be negotiated between the device 101 and service 120 at installation time. When sending a message to a device 101 the service 120 could first encrypt/MAC with this device installation key, then encrypt/MAC that with the hub's session key. The hub 110 would then verify and extract the encrypted device blob and send that to the device.

[00100] In one embodiment of the invention, a counter mechanism is implemented to prevent replay attacks. For example, each successive communication from the device 101 to the hub 110 (or vice versa) may be assigned a continually increasing counter value. Both the hub 110 and device 101 will track this value and verify that the value is

correct in each successive communication between the devices. The same techniques may be implemented between the hub 110 and the service 120. Using a counter in this manner would make it more difficult to spoof the communication between each of the devices (because the counter value would be incorrect). However, even without this a shared installation key between the service and device would prevent network (hub) wide attacks to all devices.

[00101] In one embodiment, when using public/private key encryption, the IoT hub 110 uses its private key to decrypt the IoT hub packet and generate the encrypted IoT device packet, which it transmits to the associated IoT device 101. The IoT device 101 then uses its private key to decrypt the IoT device packet to generate the command/data originated from the IoT service 120. It may then process the data and/or execute the command. Using symmetric encryption, each device would encrypt and decrypt with the shared symmetric key. If either case, each transmitting device may also sign the message with its private key so that the receiving device can verify its authenticity.

[00102] A different set of keys may be used to encrypt communication from the IoT device 101 to the IoT hub 110 and to the IoT service 120. For example, using a public/private key arrangement, in one embodiment, the security logic 1002 on the IoT device 101 uses the public key of the IoT hub 110 to encrypt data packets sent to the IoT hub 110. The security logic 1012 on the IoT hub 110 may then decrypt the data packets using the IoT hub's private key. Similarly, the security logic 1002 on the IoT device 101 and/or the security logic 1012 on the IoT hub 110 may encrypt data packets sent to the IoT service 120 using the public key of the IoT service 120 (which may then be decrypted by the security logic 1013 on the IoT service 120 using the service's private key). Using symmetric keys, the device 101 and hub 110 may share a symmetric key while the hub and service 120 may share a different symmetric key.

[00103] While certain specific details are set forth above in the description above, it should be noted that the underlying principles of the invention may be implemented using various different encryption techniques. For example, while some embodiments discussed above use asymmetric public/private key pairs, an alternate embodiment may use symmetric keys securely exchanged between the various IoT devices 101-102, IoT hubs 110, and the IoT service 120. Moreover, in some embodiments, the data/command itself is not encrypted, but a key is used to generate a signature over the data/command (or other data structure). The recipient may then use its key to validate the signature.

[00104] As illustrated in **Figure 11**, in one embodiment, the secure key storage on each IoT device 101 is implemented using a programmable subscriber identity module (SIM) 1101. In this embodiment, the IoT device 101 may initially be provided to the end user with an un-programmed SIM card 1101 seated within a SIM interface 1100 on the IoT device 101. In order to program the SIM with a set of one or more encryption keys, the user takes the programmable SIM card 1101 out of the SIM interface 500 and inserts it into a SIM programming interface 1102 on the IoT hub 110. Programming logic 1125 on the IoT hub then securely programs the SIM card 1101 to register/pair the IoT device 101 with the IoT hub 110 and IoT service 120. In one embodiment, a public/private key pair may be randomly generated by the programming logic 1125 and the public key of the pair may then be stored in the IoT hub's secure storage device 411 while the private key may be stored within the programmable SIM 1101. In addition, the programming logic 525 may store the public keys of the IoT hub 110, the IoT service 120, and/or any other IoT devices 101 on the SIM card 1401 (to be used by the security logic 1302 on the IoT device 101 to encrypt outgoing data). Once the SIM 1101 is programmed, the new IoT device 101 may be provisioned with the IoT Service 120 using the SIM as a secure identifier (e.g., using existing techniques for registering a device using a SIM). Following provisioning, both the IoT hub 110 and the IoT service 120 will securely store a copy of the IoT device's public key to be used when encrypting communication with the IoT device 101.

[00105] The techniques described above with respect to **Figure 11** provide enormous flexibility when providing new IoT devices to end users. Rather than requiring a user to directly register each SIM with a particular service provider upon sale/purchase (as is currently done), the SIM may be programmed directly by the end user via the IoT hub 110 and the results of the programming may be securely communicated to the IoT service 120. Consequently, new IoT devices 101 may be sold to end users from online or local retailers and later securely provisioned with the IoT service 120.

[00106] While the registration and encryption techniques are described above within the specific context of a SIM (Subscriber Identity Module), the underlying principles of the invention are not limited to a "SIM" device. Rather, the underlying principles of the invention may be implemented using any type of device having secure storage for storing a set of encryption keys. Moreover, while the embodiments above include a removable SIM device, in one embodiment, the SIM device is not removable but the IoT device itself may be inserted within the programming interface 1102 of the IoT hub 110.

[00107] In one embodiment, rather than requiring the user to program the SIM (or other device), the SIM is pre-programmed into the IoT device 101, prior to distribution to the end user. In this embodiment, when the user sets up the IoT device 101, various techniques described herein may be used to securely exchange encryption keys between the IoT hub 110/IoT service 120 and the new IoT device 101.

[00108] For example, as illustrated in **Figure 12A** each IoT device 101 or SIM 401 may be packaged with a barcode or QR code 1501 uniquely identifying the IoT device 101 and/or SIM 1001. In one embodiment, the barcode or QR code 1201 comprises an encoded representation of the public key for the IoT device 101 or SIM 1001.

Alternatively, the barcode or QR code 1201 may be used by the IoT hub 110 and/or IoT service 120 to identify or generate the public key (e.g., used as a pointer to the public key which is already stored in secure storage). The barcode or QR code 601 may be printed on a separate card (as shown in **Figure 12A**) or may be printed directly on the IoT device itself. Regardless of where the barcode is printed, in one embodiment, the IoT hub 110 is equipped with a barcode reader 206 for reading the barcode and providing the resulting data to the security logic 1012 on the IoT hub 110 and/or the security logic 1013 on the IoT service 120. The security logic 1012 on the IoT hub 110 may then store the public key for the IoT device within its secure key storage 1011 and the security logic 1013 on the IoT service 120 may store the public key within its secure storage 1021 (to be used for subsequent encrypted communication).

[00109] In one embodiment, the data contained in the barcode or QR code 1201 may also be captured via a user device 135 (e.g., such as an iPhone or Android device) with an installed IoT app or browser-based applet designed by the IoT service provider. Once captured, the barcode data may be securely communicated to the IoT service 120 over a secure connection (e.g., such as a secure sockets layer (SSL) connection). The barcode data may also be provided from the client device 135 to the IoT hub 110 over a secure local connection (e.g., over a local WiFi or Bluetooth LE connection).

[00110] The security logic 1002 on the IoT device 101 and the security logic 1012 on the IoT hub 110 may be implemented using hardware, software, firmware or any combination thereof. For example, in one embodiment, the security logic 1002, 1012 is implemented within the chips used for establishing the local communication channel 130 between the IoT device 101 and the IoT hub 110 (e.g., the Bluetooth LE chip if the local channel 130 is Bluetooth LE). Regardless of the specific location of the security logic 1002, 1012, in one embodiment, the security logic 1002, 1012 is designed to establish a secure execution environment for executing certain types of program code.

This may be implemented, for example, by using TrustZone technology (available on some ARM processors) and/or Trusted Execution Technology (designed by Intel). Of course, the underlying principles of the invention are not limited to any particular type of secure execution technology.

[00111] In one embodiment, the barcode or QR code 1501 may be used to pair each IoT device 101 with the IoT hub 110. For example, rather than using the standard wireless pairing process currently used to pair Bluetooth LE devices, a pairing code embedded within the barcode or QR code 1501 may be provided to the IoT hub 110 to pair the IoT hub with the corresponding IoT device.

[00112] **Figure 12B** illustrates one embodiment in which the barcode reader 206 on the IoT hub 110 captures the barcode/QR code 1201 associated with the IoT device 101. As mentioned, the barcode/QR code 1201 may be printed directly on the IoT device 101 or may be printed on a separate card provided with the IoT device 101. In either case, the barcode reader 206 reads the pairing code from the barcode/QR code 1201 and provides the pairing code to the local communication module 1280. In one embodiment, the local communication module 1280 is a Bluetooth LE chip and associated software, although the underlying principles of the invention are not limited to any particular protocol standard. Once the pairing code is received, it is stored in a secure storage containing pairing data 1285 and the IoT device 101 and IoT hub 110 are automatically paired. Each time the IoT hub is paired with a new IoT device in this manner, the pairing data for that pairing is stored within the secure storage 685. In one embodiment, once the local communication module 1280 of the IoT hub 110 receives the pairing code, it may use the code as a key to encrypt communications over the local wireless channel with the IoT device 101.

[00113] Similarly, on the IoT device 101 side, the local communication module 1590 stores pairing data within a local secure storage device 1595 indicating the pairing with the IoT hub. The pairing data 1295 may include the pre-programmed pairing code identified in the barcode/QR code 1201. The pairing data 1295 may also include pairing data received from the local communication module 1280 on the IoT hub 110 required for establishing a secure local communication channel (e.g., an additional key to encrypt communication with the IoT hub 110).

[00114] Thus, the barcode/QR code 1201 may be used to perform local pairing in a far more secure manner than current wireless pairing protocols because the pairing code is not transmitted over the air. In addition, in one embodiment, the same barcode/QR code 1201 used for pairing may be used to identify encryption keys to build

a secure connection from the IoT device 101 to the IoT hub 110 and from the IoT hub 110 to the IoT service 120.

[00115] A method for programming a SIM card in accordance with one embodiment of the invention is illustrated in **Figure 13**. The method may be implemented within the system architecture described above, but is not limited to any particular system architecture.

[00116] At 1301, a user receives a new IoT device with a blank SIM card and, at 1602, the user inserts the blank SIM card into an IoT hub. At 1303, the user programs the blank SIM card with a set of one or more encryption keys. For example, as mentioned above, in one embodiment, the IoT hub may randomly generate a public/private key pair and store the private key on the SIM card and the public key in its local secure storage. In addition, at 1304, at least the public key is transmitted to the IoT service so that it may be used to identify the IoT device and establish encrypted communication with the IoT device. As mentioned above, in one embodiment, a programmable device other than a "SIM" card may be used to perform the same functions as the SIM card in the method shown in **Figure 13**.

[00117] A method for integrating a new IoT device into a network is illustrated in **Figure 14**. The method may be implemented within the system architecture described above, but is not limited to any particular system architecture.

[00118] At 1401, a user receives a new IoT device to which an encryption key has been pre-assigned. At 1402, the key is securely provided to the IoT hub. As mentioned above, in one embodiment, this involves reading a barcode associated with the IoT device to identify the public key of a public/private key pair assigned to the device. The barcode may be read directly by the IoT hub or captured via a mobile device via an app or browser. In an alternate embodiment, a secure communication channel such as a Bluetooth LE channel, a near field communication (NFC) channel or a secure WiFi channel may be established between the IoT device and the IoT hub to exchange the key. Regardless of how the key is transmitted, once received, it is stored in the secure keystore of the IoT hub device. As mentioned above, various secure execution technologies may be used on the IoT hub to store and protect the key such as Secure Enclaves, Trusted Execution Technology (TXT), and/or Trustzone. In addition, at 803, the key is securely transmitted to the IoT service which stores the key in its own secure keystore. It may then use the key to encrypt communication with the IoT device. One again, the exchange may be implemented using a certificate/signed key. Within the hub

110 it is particularly important to prevent modification/addition/ removal of the stored keys.

[00119] A method for securely communicating commands/data to an IoT device using public/private keys is illustrated in **Figure 15**. The method may be implemented within the system architecture described above, but is not limited to any particular system architecture.

[00120] At 1501, the IoT service encrypts the data/commands using the IoT device public key to create an IoT device packet. It then encrypts the IoT device packet using IoT hub's public key to create the IoT hub packet (e.g., creating an IoT hub wrapper around the IoT device packet). At 1502, the IoT service transmits the IoT hub packet to the IoT hub. At 1503, the IoT hub decrypts the IoT hub packet using the IoT hub's private key to generate the IoT device packet. At 1504 it then transmits the IoT device packet to the IoT device which, at 1505, decrypts the IoT device packet using the IoT device private key to generate the data/commands. At 1506, the IoT device processes the data/commands.

[00121] In an embodiment which uses symmetric keys, a symmetric key exchange may be negotiated between each of the devices (e.g., each device and the hub and between the hub and the service). Once the key exchange is complete, each transmitting device encrypts and/or signs each transmission using the symmetric key before transmitting data to the receiving device.

APPARATUS AND METHOD FOR ESTABLISHING SECURE

COMMUNICATION CHANNELS IN AN INTERNET OF THINGS (IoT) SYSTEM

[00122] In one embodiment of the invention, encryption and decryption of data is performed between the IoT service 120 and each IoT device 101, regardless of the intermediate devices used to support the communication channel (e.g., such as the user's mobile device 611 and/or the IoT hub 110). One embodiment which communicates via an IoT hub 110 is illustrated in **Figure 16A** and another embodiment which does not require an IoT hub is illustrated in **Figure 16B**.

[00123] Turning first to **Figure 16A**, the IoT service 120 includes an encryption engine 1660 which manages a set of "service session keys" 1650 and each IoT device 101 includes an encryption engine 1661 which manages a set of "device session keys" 1651 for encrypting/decrypting communication between the IoT device 101 and IoT service 120. The encryption engines may rely on different hardware modules when performing the security/encryption techniques described herein including a hardware security module 1630-1631 for (among other things) generating a session public/private

key pair and preventing access to the private session key of the pair and a key stream generation module 1640-1641 for generating a key stream using a derived secret. In one embodiment, the service session keys 1650 and the device session keys 1651 comprise related public/private key pairs. For example, in one embodiment, the device session keys 1651 on the IoT device 101 include a public key of the IoT service 120 and a private key of the IoT device 101. As discussed in detail below, in one embodiment, to establish a secure communication session, the public/private session key pairs, 1650 and 1651, are used by each encryption engine, 1660 and 1661, respectively, to generate the same secret which is then used by the SKGMs 1640-1641 to generate a key stream to encrypt and decrypt communication between the IoT service 120 and the IoT device 101. Additional details associated with generation and use of the secret in accordance with one embodiment of the invention are provided below.

[00124] In **Figure 16A**, once the secret has been generated using the keys 1650-1651, the client will always send messages to the IoT device 101 through the IoT service 120, as indicated by Clear transaction 1611. "Clear" as used herein is meant to indicate that the underlying message is not encrypted using the encryption techniques described herein. However, as illustrated, in one embodiment, a secure sockets layer (SSL) channel or other secure channel (e.g., an Internet Protocol Security (IPSEC) channel) is established between the client device 611 and IoT service 120 to protect the communication. The encryption engine 1660 on the IoT service 120 then encrypts the message using the generated secret and transmits the encrypted message to the IoT hub 110 at 1602. Rather than using the secret to encrypt the message directly, in one embodiment, the secret and a counter value are used to generate a key stream, which is used to encrypt each message packet. Details of this embodiment are described below with respect to **Figure 17**.

[00125] As illustrated, an SSL connection or other secure channel may be established between the IoT service 120 and the IoT hub 110. The IoT hub 110 (which does not have the ability to decrypt the message in one embodiment) transmits the encrypted message to the IoT device at 1603 (e.g., over a Bluetooth Low Energy (BTLE) communication channel). The encryption engine 1661 on the IoT device 101 may then decrypt the message using the secret and process the message contents. In an embodiment which uses the secret to generate a key stream, the encryption engine 1661 may generate the key stream using the secret and a counter value and then use the key stream for decryption of the message packet.

[00126] The message itself may comprise any form of communication between the IoT service 120 and IoT device 101. For example, the message may comprise a command packet instructing the IoT device 101 to perform a particular function such as taking a measurement and reporting the result back to the client device 611 or may include configuration data to configure the operation of the IoT device 101.

[00127] If a response is required, the encryption engine 1661 on the IoT device 101 uses the secret or a derived key stream to encrypt the response and transmits the encrypted response to the IoT hub 110 at 1604, which forwards the response to the IoT service 120 at 1605. The encryption engine 1660 on the IoT service 120 then decrypts the response using the secret or a derived key stream and transmits the decrypted response to the client device 611 at 1606 (e.g., over the SSL or other secure communication channel).

[00128] **Figure 16B** illustrates an embodiment which does not require an IoT hub. Rather, in this embodiment, communication between the IoT device 101 and IoT service 120 occurs through the client device 611 (e.g., as in the embodiments described above with respect to **Figures 6-9B**). In this embodiment, to transmit a message to the IoT device 101 the client device 611 transmits an unencrypted version of the message to the IoT service 120 at 1611. The encryption engine 1660 encrypts the message using the secret or the derived key stream and transmits the encrypted message back to the client device 611 at 1612. The client device 611 then forwards the encrypted message to the IoT device 101 at 1613, and the encryption engine 1661 decrypts the message using the secret or the derived key stream. The IoT device 101 may then process the message as described herein. If a response is required, the encryption engine 1661 encrypts the response using the secret and transmits the encrypted response to the client device 611 at 1614, which forwards the encrypted response to the IoT service 120 at 1615. The encryption engine 1660 then decrypts the response and transmits the decrypted response to the client device 611 at 1616.

[00129] **Figure 17** illustrates a key exchange and key stream generation which may initially be performed between the IoT service 120 and the IoT device 101. In one embodiment, this key exchange may be performed each time the IoT service 120 and IoT device 101 establish a new communication session. Alternatively, the key exchange may be performed and the exchanged session keys may be used for a specified period of time (e.g., a day, a week, etc). While no intermediate devices are shown in **Figure 17** for simplicity, communication may occur through the IoT hub 110 and/or the client device 611.

[00130] In one embodiment, the encryption engine 1660 of the IoT service 120 sends a command to the HSM 1630 (e.g., which may be such as a CloudHSM offered by Amazon®) to generate a session public/private key pair. The HSM 1630 may subsequently prevent access to the private session key of the pair. Similarly, the encryption engine on the IoT device 101 may transmit a command to the HSM 1631 (e.g., such as an Atecc508 HSM from Atmel Corporation®) which generates a session public/private key pair and prevents access to the session private key of the pair. Of course, the underlying principles of the invention are not limited to any specific type of encryption engine or manufacturer.

[00131] In one embodiment, the IoT service 120 transmits its session public key generated using the HSM 1630 to the IoT device 101 at 1701. The IoT device uses its HSM 1631 to generate its own session public/private key pair and, at 1702, transmits its public key of the pair to the IoT service 120. In one embodiment, the encryption engines 1660-1661 use an Elliptic curve Diffie–Hellman (ECDH) protocol, which is an anonymous key agreement that allows two parties with an elliptic curve public–private key pair, to establish a shared secret. In one embodiment, using these techniques, at 1703, the encryption engine 1660 of the IoT service 120 generates the secret using the IoT device session public key and its own session private key. Similarly, at 1704, the encryption engine 1661 of the IoT device 101 independently generates the same secret using the IoT service 120 session public key and its own session private key. More specifically, in one embodiment, the encryption engine 1660 on the IoT service 120 generates the secret according to the formula $secret = IoT\ device\ session\ pub\ key * IoT\ service\ session\ private\ key$, where “*” means that the IoT device session public key is point-multiplied by the IoT service session private key. The encryption engine 1661 on the IoT device 101 generates the secret according to the formula $secret = IoT\ service\ session\ pub\ key * IoT\ device\ session\ private\ key$, where the IoT service session public key is point multiplied by the IoT device session private key. In the end, the IoT service 120 and IoT device 101 have both generated the same secret to be used to encrypt communication as described below. In one embodiment, the encryption engines 1660-1661 rely on a hardware module such as the KSGMs 1640-1641 respectively to perform the above operations for generating the secret.

[00132] Once the secret has been determined, it may be used by the encryption engines 1660 and 1661 to encrypt and decrypt data directly. Alternatively, in one embodiment, the encryption engines 1660-1661 send commands to the KSGMs 1640-1641 to generate a new key stream using the secret to encrypt/decrypt each data

packet (i.e., a new key stream data structure is generated for each packet). In particular, one embodiment of the key stream generation module 1640-1641 implements a Galois/Counter Mode (GCM) in which a counter value is incremented for each data packet and is used in combination with the secret to generate the key stream. Thus, to transmit a data packet to the IoT service 120, the encryption engine 1661 of the IoT device 101 uses the secret and the current counter value to cause the KSGMs 1640-1641 to generate a new key stream and increment the counter value for generating the next key stream. The newly-generated key stream is then used to encrypt the data packet prior to transmission to the IoT service 120. In one embodiment, the key stream is XORed with the data to generate the encrypted data packet. In one embodiment, the IoT device 101 transmits the counter value with the encrypted data packet to the IoT service 120. The encryption engine 1660 on the IoT service then communicates with the KSGM 1640 which uses the received counter value and the secret to generate the key stream (which should be the same key stream because the same secret and counter value are used) and uses the generated key stream to decrypt the data packet.

[00133] In one embodiment, data packets transmitted from the IoT service 120 to the IoT device 101 are encrypted in the same manner. Specifically, a counter is incremented for each data packet and used along with the secret to generate a new key stream. The key stream is then used to encrypt the data (e.g., performing an XOR of the data and the key stream) and the encrypted data packet is transmitted with the counter value to the IoT device 101. The encryption engine 1661 on the IoT device 101 then communicates with the KSGM 1641 which uses the counter value and the secret to generate the same key stream which is used to decrypt the data packet. Thus, in this embodiment, the encryption engines 1660-1661 use their own counter values to generate a key stream to encrypt data and use the counter values received with the encrypted data packets to generate a key stream to decrypt the data.

[00134] In one embodiment, each encryption engine 1660-1661 keeps track of the last counter value it received from the other and includes sequencing logic to detect whether a counter value is received out of sequence or if the same counter value is received more than once. If a counter value is received out of sequence, or if the same counter value is received more than once, this may indicate that a replay attack is being attempted. In response, the encryption engines 1660-1661 may disconnect from the communication channel and/or may generate a security alert.

[00135] **Figure 18** illustrates an exemplary encrypted data packet employed in one embodiment of the invention comprising a 4-byte counter value 1800, a variable-sized encrypted data field 1801, and a 6-byte tag 1802. In one embodiment, the tag 1802 comprises a checksum value to validate the decrypted data (once it has been decrypted).

[00136] As mentioned, in one embodiment, the session public/private key pairs 1650-1651 exchanged between the IoT service 120 and IoT device 101 may be generated periodically and/or in response to the initiation of each new communication session.

[00137] One embodiment of the invention implements additional techniques for authenticating sessions between the IoT service 120 and IoT device 101. In particular, in one embodiment, hierarchy of public/private key pairs is used including a master key pair, a set of factory key pairs, and a set of IoT service key pairs, and a set of IoT device key pairs. In one embodiment, the master key pair comprises a root of trust for all of the other key pairs and is maintained in a single, highly secure location (e.g., under the control of the organization implementing the IoT systems described herein). The master private key may be used to generate signatures over (and thereby authenticate) various other key pairs such as the factory key pairs. The signatures may then be verified using the master public key. In one embodiment, each factory which manufactures IoT devices is assigned its own factory key pair which may then be used to authenticate IoT service keys and IoT device keys. For example, in one embodiment, a factory private key is used to generate a signature over IoT service public keys and IoT device public keys. These signature may then be verified using the corresponding factory public key. Note that these IoT service/device public keys are not the same as the "session" public/private keys described above with respect to **Figures 16A-B**. The session public/private keys described above are temporary (i.e., generated for a service/device session) while the IoT service/device key pairs are permanent (i.e., generated at the factory).

[00138] With the foregoing relationships between master keys, factory keys, service/device keys in mind, one embodiment of the invention performs the following operations to provide additional layers of authentication and security between the IoT service 120 and IoT device 101:

A. In one embodiment, the IoT service 120 initially generates a message containing the following:

1. The IoT service's unique ID:
 - The IoT service's serial number;

- a Timestamp;
 - The ID of the factory key used to sign this unique ID;
 - a Class of the unique ID (i.e., a service);
 - IoT service's public key
 - The signature over the unique ID.
2. The Factory Certificate including:
 - A timestamp
 - The ID of the master key used to sign the certificate
 - The factory public key
 - The signature of the Factory Certificate
 3. IoT service session public key (as described above with respect to **Figures 16A-B**)
 4. IoT service session public key signature (e.g., signed with the IoT service's private key)

B. In one embodiment, the message is sent to the IoT device on the negotiation channel (described below). The IoT device parses the message and:

1. Verifies the signature of the factory certificate (only if present in the message payload)
2. Verifies the signature of the unique ID using the key identified by the unique ID
3. Verifies the IoT service session public key signature using the IoT service's public key from the unique ID
4. Saves the IoT service's public key as well as the IoT service's session public key
5. Generates the IoT device session key pair

C. The IoT device then generates a message containing the following:

1. IoT device's unique ID
 - IoT device serial number
 - Timestamp
 - ID of factory key used to sign this unique ID
 - Class of unique ID (i.e., IoT device)
 - IoT device's public key

- Signature of unique ID
- 2. IoT device's session public key
- 3. Signature of (IoT device session public key + IoT service session public key) signed with IoT device's key

D. This message is sent back to the IoT service. The IoT service parses the message and:

1. Verifies the signature of the unique ID using the factory public key
2. Verifies the signature of the session public keys using the IoT device's public key
3. Saves the IoT device's session public key

E. The IoT service then generates a message containing a signature of (IoT device session public key + IoT service session public key) signed with the IoT service's key.

F. The IoT device parses the message and:

1. Verifies the signature of the session public keys using the IoT service's public key
2. Generates the key stream from the IoT device session private key and the IoT service's session public key
3. The IoT device then sends a "messaging available" message.

G. The IoT service then does the following:

1. Generates the key stream from the IoT service session private key and the IoT device's session public key
2. Creates a new message on the messaging channel which contains the following:

- Generates and stores a random 2 byte value
- Set attribute message with the boomerang attribute Id (discussed below) and the random value

H. The IoT device receives the message and:

1. Attempts to decrypt the message
2. Emits an Update with the same value on the indicated attribute Id

I. The IoT service recognizes the message payload contains a boomerang attribute update and:

1. Sets its paired state to true
2. Sends a pairing complete message on the negotiator channel

J. IoT device receives the message and sets his paired state to true

[00139] While the above techniques are described with respect to an “IoT service” and an “IoT device,” the underlying principles of the invention may be implemented to establish a secure communication channel between any two devices including user client devices, servers, and Internet services.

[00140] The above techniques are highly secure because the private keys are never shared over the air (in contrast to current Bluetooth pairing techniques in which a secret is transmitted from one party to the other). An attacker listening to the entire conversation will only have the public keys, which are insufficient to generate the shared secret. These techniques also prevent a man-in-the-middle attack by exchanging signed public keys. In addition, because GCM and separate counters are used on each device, any kind of “replay attack” (where a man in the middle captures the data and sends it again) is prevented. Some embodiments also prevent replay attacks by using asymmetrical counters.

TECHNIQUES FOR EXCHANGING DATA AND COMMANDS WITHOUT FORMALLY PAIRING DEVICES

[00141] GATT is an acronym for the Generic Attribute Profile, and it defines the way that two Bluetooth Low Energy (BTLE) devices transfer data back and forth. It makes use of a generic data protocol called the Attribute Protocol (ATT), which is used to store Services, Characteristics and related data in a simple lookup table using 16-bit Characteristic IDs for each entry in the table. Note that while the “characteristics” are sometimes referred to as “attributes.”

[00142] On Bluetooth devices, the most commonly used characteristic is the devices “name” (having characteristic ID 10752 (0x2A00)). For example, a Bluetooth device may identify other Bluetooth devices within its vicinity by reading the “Name” characteristic published by those other Bluetooth devices using GATT. Thus, Bluetooth device have the inherent ability to exchange data without formally pairing/bonding the

devices (note that “paring” and “bonding” are sometimes used interchangeably; the remainder of this discussion will use the term “pairing”).

[00143] One embodiment of the invention takes advantage of this capability to communicate with BTLE-enabled IoT devices without formally pairing with these devices. Pairing with each individual IoT device would be extremely inefficient because of the amount of time required to pair with each device and because only one paired connection may be established at a time.

[00144] **Figure 19** illustrates one particular embodiment in which a Bluetooth (BT) device 1910 establishes a network socket abstraction with a BT communication module 1901 of an IoT device 101 without formally establishing a paired BT connection. The BT device 1910 may be included in an IoT hub 110 and/or a client device 611 such as shown in **Figure 16A**. As illustrated, the BT communication module 1901 maintains a data structure containing a list of characteristic IDs, names associated with those characteristic IDs and values for those characteristic IDs. The value for each characteristic may be stored within a 20-byte buffer identified by the characteristic ID in accordance with the current BT standard. However, the underlying principles of the invention are not limited to any particular buffer size.

[00145] In the example in **Figure 19**, the “Name” characteristic is a BT-defined characteristic which is assigned a specific value of “IoT Device 14.” One embodiment of the invention specifies a first set of additional characteristics to be used for negotiating a secure communication channel with the BT device 1910 and a second set of additional characteristics to be used for encrypted communication with the BT device 1910. In particular, a “negotiation write” characteristic, identified by characteristic ID <65532> in the illustrated example, may be used to transmit outgoing negotiation messages and the “negotiation read” characteristic, identified by characteristic ID <65533> may be used to receive incoming negotiation messages. The “negotiation messages” may include messages used by the BT device 1910 and the BT communication module 1901 to establish a secure communication channel as described herein. By way of example, in **Figure 17**, the IoT device 101 may receive the IoT service session public key 1701 via the “negotiation read” characteristic <65533>. The key 1701 may be transmitted from the IoT service 120 to a BTLE-enabled IoT hub 110 or client device 611 which may then use GATT to write the key 1701 to the negotiation read value buffer identified by characteristic ID <65533>. IoT device application logic 1902 may then read the key 1701 from the value buffer identified by characteristic ID <65533> and process it as

described above (e.g., using it to generate a secret and using the secret to generate a key stream, etc).

[00146] If the key 1701 is greater than 20 bytes (the maximum buffer size in some current implementations), then it may be written in 20-byte portions. For example, the first 20 bytes may be written by the BT communication module 1903 to characteristic ID <65533> and read by the IoT device application logic 1902, which may then write an acknowledgement message to the negotiation write value buffer identified by characteristic ID <65532>. Using GATT, the BT communication module 1903 may read this acknowledgement from characteristic ID <65532> and responsively write the next 20 bytes of the key 1701 to the negotiation read value buffer identified by characteristic ID <65533>. In this manner, a network socket abstraction defined by characteristic IDs <65532> and <65533> is established for exchanging negotiation messages used to establish a secure communication channel.

[00147] In one embodiment, once the secure communication channel is established, a second network socket abstraction is established using characteristic ID <65534> (for transmitting encrypted data packets from IoT device 101) and characteristic ID <65533> (for receiving encrypted data packets by IoT device). That is, when BT communication module 1903 has an encrypted data packet to transmit (e.g., such as encrypted message 1603 in **Figure 16A**), it starts writing the encrypted data packet, 20 bytes at a time, using the message read value buffer identified by characteristic ID <65533>. The IoT device application logic 1902 will then read the encrypted data packet, 20 bytes at a time, from the read value buffer, sending acknowledgement messages to the BT communication module 1903 as needed via the write value buffer identified by characteristic ID <65532>.

[00148] In one embodiment, the commands of GET, SET, and UPDATE described below are used to exchange data and commands between the two BT communication modules 1901 and 1903. For example, the BT communication module 1903 may send a packet identifying characteristic ID <65533> and containing the SET command to write into the value field/buffer identified by characteristic ID <65533> which may then be read by the IoT device application logic 1902. To retrieve data from the IoT device 101, the BT communication module 1903 may transmit a GET command directed to the value field/buffer identified by characteristic ID <65534>. In response to the GET command, the BT communication module 1901 may transmit an UPDATE packet to the BT communication module 1903 containing the data from the value field/buffer identified by characteristic ID <65534>. In addition, UPDATE packets may be transmitted

automatically, in response to changes in a particular attribute on the IoT device 101. For example, if the IoT device is associated with a lighting system and the user turns on the lights, then an UPDATE packet may be sent to reflect the change to the on/off attribute associated with the lighting application.

[00149] **Figure 20** illustrates exemplary packet formats used for GET, SET, and UPDATE in accordance with one embodiment of the invention. In one embodiment, these packets are transmitted over the message write <65534> and message read <65533> channels following negotiation. In the GET packet 2001, a first 1-byte field includes a value (0X10) which identifies the packet as a GET packet. A second 1-byte field includes a request ID, which uniquely identifies the current GET command (i.e., identifies the current transaction with which the GET command is associated). For example, each instance of a GET command transmitted from a service or device may be assigned a different request ID. This may be done, for example, by incrementing a counter and using the counter value as the request ID. However, the underlying principles of the invention are not limited to any particular manner for setting the request ID.

[00150] A 2-byte attribute ID identifies the application-specific attribute to which the packet is directed. For example, if the GET command is being sent to IoT device 101 illustrated in **Figure 19**, the attribute ID may be used to identify the particular application-specific value being requested. Returning to the above example, the GET command may be directed to an application-specific attribute ID such as power status of a lighting system, which comprises a value identifying whether the lights are powered on or off (e.g., 1 = on, 0 = off). If the IoT device 101 is a security apparatus associated with a door, then the value field may identify the current status of the door (e.g., 1 = opened, 0 = closed). In response to the GET command, a response may be transmitting containing the current value identified by the attribute ID.

[00151] The SET packet 2002 and UPDATE packet 2003 illustrated in **Figure 20** also include a first 1-byte field identifying the type of packet (i.e., SET and UPDATE), a second 1-byte field containing a request ID, and a 2-byte attribute ID field identifying an application-defined attribute. In addition, the SET packet includes a 2-byte length value identifying the length of data contained in an n-byte value data field. The value data field may include a command to be executed on the IoT device and/or configuration data to configure the operation of the IoT device in some manner (e.g., to set a desired parameter, to power down the IoT device, etc). For example, if the IoT device 101 controls the speed of a fan, the value field may reflect the current fan speed.

[00152] The UPDATE packet 2003 may be transmitted to provide an update of the results of the SET command. The UPDATE packet 2003 includes a 2-byte length value field to identify the length of the n-byte value data field which may include data related to the results of the SET command. In addition, a 1-byte update state field may identify the current state of the variable being updated. For example, if the SET command attempted to turn off a light controlled by the IoT device, the update state field may indicate whether the light was successfully turned off.

[00153] **Figure 21** illustrates an exemplary sequence of transactions between the IoT service 120 and an IoT device 101 involving the SET and UPDATE commands. Intermediary devices such as the IoT hub and the user's mobile device are not shown to avoid obscuring the underlying principles of the invention. At 2101, the SET command 2101 is transmitted from the IoT service to the IoT device 101 and received by the BT communication module 1901 which responsively updates the GATT value buffer identified by the characteristic ID at 2102. The SET command is read from the value buffer by the low power microcontroller (MCU) 200 at 2103 (or by program code being executed on the low power MCU such as IoT device application logic 1902 shown in **Figure 19**). At 2104, the MCU 200 or program code performs an operation in response to the SET command. For example, the SET command may include an attribute ID specifying a new configuration parameter such as a new temperature or may include a state value such as on/off (to cause the IoT device to enter into an "on" or a low power state). Thus, at 2104, the new value is set in the IoT device and an UPDATE command is returned at 2105 and the actual value is updated in a GATT value field at 2106. In some cases, the actual value will be equal to the desired value. In other cases, the updated value may be different (i.e., because it may take time for the IoT device 101 to update certain types of values). Finally, at 2107, the UPDATE command is transmitted back to the IoT service 120 containing the actual value from the GATT value field.

[00154] **Figure 22** illustrates a method for implementing a secure communication channel between an IoT service and an IoT device in accordance with one embodiment of the invention. The method may be implemented within the context of the network architectures described above but is not limited to any specific architecture.

[00155] At 2201, the IoT service creates an encrypted channel to communicate with the IoT hub using elliptic curve digital signature algorithm (ECDSA) certificates. At 2202, the IoT service encrypts data/commands in IoT device packets using the a session secret to create an encrypted device packet. As mentioned above, the session secret may be independently generated by the IoT device and the IoT service. At 2203,

the IoT service transmits the encrypted device packet to the IoT hub over the encrypted channel. At 2204, without decrypting, the IoT hub passes the encrypted device packet to the IoT device. At 2205, the IoT device uses the session secret to decrypt the encrypted device packet. As mentioned, in one embodiment this may be accomplished by using the secret and a counter value (provided with the encrypted device packet) to generate a key stream and then using the key stream to decrypt the packet. At 2206, the IoT device then extracts and processes the data and/or commands contained within the device packet.

[00156] Thus, using the above techniques, bi-directional, secure network socket abstractions may be established between two BT-enabled devices without formally pairing the BT devices using standard pairing techniques. While these techniques are described above with respect to an IoT device 101 communicating with an IoT service 120, the underlying principles of the invention may be implemented to negotiate and establish a secure communication channel between any two BT-enabled devices.

[00157] **Figures 23A-C** illustrate a detailed method for pairing devices in accordance with one embodiment of the invention. The method may be implemented within the context of the system architectures described above, but is not limited to any specific system architectures.

[00158] At 2301, the IoT Service creates a packet containing serial number and public key of the IoT Service. At 2302, the IoT Service signs the packet using the factory private key. At 2303, the IoT Service sends the packet over an encrypted channel to the IoT hub and at 2304 the IoT hub forwards the packet to IoT device over an unencrypted channel. At 2305, the IoT device verifies the signature of packet and, at 2306, the IoT device generates a packet containing the serial number and public key of the IoT Device. At 2307, the IoT device signs the packet using the factory private key and at 2308, the IoT device sends the packet over the unencrypted channel to the IoT hub.

[00159] At 2309, the IoT hub forwards the packet to the IoT service over an encrypted channel and at 2310, the IoT Service verifies the signature of the packet. At 2311, the IoT Service generates a session key pair, and at 2312 the IoT Service generates a packet containing the session public key. The IoT Service then signs the packet with IoT Service private key at 2313 and, at 2314, the IoT Service sends the packet to the IoT hub over the encrypted channel.

[00160] Turning to **Figure 23B**, the IoT hub forwards the packet to the IoT device over the unencrypted channel at 2315 and, at 2316, the IoT device verifies the signature

of packet. At 2317 the IoT device generates session key pair (e.g., using the techniques described above), and, at 2318, an IoT device packet is generated containing the IoT device session public key. At 2319, the IoT device signs the IoT device packet with IoT device private key. At 2320, the IoT device sends the packet to the IoT hub over the unencrypted channel and, at 2321, the IoT hub forwards the packet to the IoT service over an encrypted channel.

[00161] At 2322, the IoT service verifies the signature of the packet (e.g., using the IoT device public key) and, at 2323, the IoT service uses the IoT service private key and the IoT device public key to generate the session secret (as described in detail above). At 2324, the IoT device uses the IoT device private key and IoT service public key to generate the session secret (again, as described above) and, at 2325, the IoT device generates a random number and encrypts it using the session secret. At 2326, the IoT service sends the encrypted packet to IoT hub over the encrypted channel. At 2327, the IoT hub forwards the encrypted packet to the IoT device over the unencrypted channel. At 2328, the IoT device decrypts the packet using the session secret.

[00162] Turning to **Figure 23C**, the IoT device re-encrypts the packet using the session secret at 2329 and, at 2330, the IoT device sends the encrypted packet to the IoT hub over the unencrypted channel. At 2331, the IoT hub forwards the encrypted packet to the IoT service over the encrypted channel. The IoT service decrypts the packet using the session secret at 2332. At 2333 the IoT service verifies that the random number matches the random number it sent. The IoT service then sends a packet indicating that pairing is complete at 2334 and all subsequent messages are encrypted using the session secret at 2335.

[00163] Embodiments of the invention may include various steps, which have been described above. The steps may be embodied in machine-executable instructions which may be used to cause a general-purpose or special-purpose processor to perform the steps. Alternatively, these steps may be performed by specific hardware components that contain hardwired logic for performing the steps, or by any combination of programmed computer components and custom hardware components.

[00164] As described herein, instructions may refer to specific configurations of hardware such as application specific integrated circuits (ASICs) configured to perform certain operations or having a predetermined functionality or software instructions stored in memory embodied in a non-transitory computer readable medium. Thus, the techniques shown in the figures can be implemented using code and data stored and executed on one or more electronic devices (e.g., an end station, a network element,

etc.). Such electronic devices store and communicate (internally and/or with other electronic devices over a network) code and data using computer machine-readable media, such as non-transitory computer machine-readable storage media (e.g., magnetic disks; optical disks; random access memory; read only memory; flash memory devices; phase-change memory) and transitory computer machine-readable communication media (e.g., electrical, optical, acoustical or other form of propagated signals – such as carrier waves, infrared signals, digital signals, etc.). In addition, such electronic devices typically include a set of one or more processors coupled to one or more other components, such as one or more storage devices (non-transitory machine-readable storage media), user input/output devices (e.g., a keyboard, a touchscreen, and/or a display), and network connections. The coupling of the set of processors and other components is typically through one or more busses and bridges (also termed as bus controllers). The storage device and signals carrying the network traffic respectively represent one or more machine-readable storage media and machine-readable communication media. Thus, the storage device of a given electronic device typically stores code and/or data for execution on the set of one or more processors of that electronic device. Of course, one or more parts of an embodiment of the invention may be implemented using different combinations of software, firmware, and/or hardware.

[00165] Throughout this detailed description, for the purposes of explanation, numerous specific details were set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the invention may be practiced without some of these specific details. In certain instances, well known structures and functions were not described in elaborate detail in order to avoid obscuring the subject matter of the present invention. Accordingly, the scope and spirit of the invention should be judged in terms of the claims which follow.

CLAIMS

What is claimed is:

1. A system comprising:
 - an Internet of Things (IoT) service to establish communication with an IoT device through an IoT hub or a mobile user device;
 - a first encryption engine on the IoT service comprising key generation logic to generate a service public key and a service private key;
 - a second encryption engine on the IoT device comprising key generation logic to generate a device public key and a device private key;
 - the first encryption engine to transmit the service public key to the second encryption engine and the second encryption engine to transmit the device public key to the first encryption engine;
 - the first encryption engine to use the device public key and the service private key to generate a secret;
 - the second encryption engine to use the service public key and the device private key to generate the same secret; and
 - wherein once the secret is generated, the first encryption engine and the second encryption engine encrypt and decrypt data packets transmitted between the first encryption engine and the second encryption engine using the secret or using data structures derived from the secret.
2. The system as in claim 1 wherein the key generation logic comprises a hardware security module (HSM).
3. The system as in claim 1 wherein the data structures derived from the secret comprise a first key stream generated by the first encryption engine and a second key stream generated by the second encryption engine.
4. The system as in claim 3 further comprising a first counter associated with the first encryption engine and a second counter associated with the second encryption engine, the first encryption engine incrementing the first counter responsive to each data packet transmitted to the second encryption engine and the second encryption engine incrementing the second counter responsive to each data packet transmitted to the first encryption engine.

5. The system as in claim 4 wherein the first encryption engine generates the first key stream using a current counter value of the first counter and the secret and the second encryption engine generates the second key stream using a current counter value of the second counter and the secret.

6. The system as in claim 5 wherein the first encryption engine comprises an elliptic curve method (ECM) module to generate the first key stream using the first counter value and the secret and the second encryption engine comprises an ECM module to generate the second key stream using the first counter value and the first secret.

7. The system as in claim 5 wherein the first encryption engine encrypts a first data packet using the first key stream to generate a first encrypted data packet and transmits the first encrypted data packet to the second encryption engine along with a current counter value of the first counter.

8. The system as in claim 7 wherein the second encryption engine uses the current counter value of the first counter and the secret to generate the first key stream and uses the first key stream to decrypt the encrypted data packet.

9. The system as in claim 7 wherein encrypting the first data packet comprises XORing the first key stream with the first data packet to generate the first encrypted data packet.

10. The system as in claim 8 wherein the IoT device comprises a Bluetooth Low Energy (BTLE) communication interface to communicatively couple the IoT device to the IoT hub or the mobile user device, the IoT hub or the mobile user device communicatively coupled to the IoT service over the Internet.

11. A method comprising:
establishing communication between an Internet of Things (IoT) service and an IoT device through an IoT hub or a mobile user device;
generating a service public key and a service private key by key generation logic of a first encryption engine on the IoT service;

generating a device public key and a device private key by key generation logic of a second encryption engine on the IoT device;

transmitting the service public key from the first encryption engine to the second encryption engine and transmitting the device public key from the second encryption engine to the first encryption engine;

generating a secret using the device public key and the service private key;

generating the same secret using the service public key and the device private key; and

encrypting and decrypting data packets transmitted between the first encryption engine and the second encryption engine using the secret or using data structures derived from the secret.

12. The method as in claim 11 wherein the key generation logic comprises a hardware security module (HSM).

13. The method as in claim 11 wherein the data structures derived from the secret comprise a first key stream generated by the first encryption engine and a second key stream generated by the second encryption engine.

14. The method as in claim 13 wherein a first counter is associated with the first encryption engine and a second counter is associated with the second encryption engine, the first encryption engine incrementing the first counter responsive to each data packet transmitted to the second encryption engine and the second encryption engine incrementing the second counter responsive to each data packet transmitted to the first encryption engine.

15. The method as in claim 14 wherein the first encryption engine generates the first key stream using a current counter value of the first counter and the secret and the second encryption engine generates the second key stream using a current counter value of the second counter and the secret.

16. The method as in claim 15 wherein the first encryption engine comprises an elliptic curve method (ECM) module to generate the first key stream using the first counter value and the secret and the second encryption engine comprises an ECM

module to generate the second key stream using the first counter value and the first secret.

17. The method as in claim 15 wherein the first encryption engine encrypts a first data packet using the first key stream to generate a first encrypted data packet and transmits the first encrypted data packet to the second encryption engine along with a current counter value of the first counter.

18. The method as in claim 17 wherein the second encryption engine uses the current counter value of the first counter and the secret to generate the first key stream and uses the first key stream to decrypt the encrypted data packet.

19. The method as in claim 17 wherein encrypting the first data packet comprises XORing the first key stream with the first data packet to generate the first encrypted data packet.

20. The method as in claim 18 wherein the IoT device comprises a Bluetooth Low Energy (BTLE) communication interface to communicatively couple the IoT device to the IoT hub or the mobile user device, the IoT hub or the mobile user device communicatively coupled to the IoT service over the Internet.

21. An apparatus comprising:

a first local wireless communication module to establish local wireless communication channels with a second local wireless communication module;

a data structure managed by the first local wireless communication module, the data structure comprising a set of characteristics, each characteristic identified with an characteristic ID and having a value buffer associated therewith containing data associated with the characteristic;

the first wireless communication module to form a first network socket using the data structure by defining a first characteristic for receiving data packets transmitted by the second local wireless communication module and a second characteristic for transmitting outgoing data packets to the second local wireless communication modules;

wherein the incoming data packets are to be temporarily buffered in a first value buffer associated with the first characteristic and wherein the outgoing packets are to be temporarily buffered in a second value buffer associated with the second characteristic.

22. The apparatus as in claim 21 wherein the data structure comprises a table indexed by a characteristic ID and comprising a value field to identify each value buffer including the first value buffer associated with the first characteristic and the second value buffer associated with the second characteristic.

23. The apparatus as in claim 21 wherein the first and second value buffers each have a size of N bits.

24. The apparatus as in claim 23 wherein $N = 20$.

25. The apparatus as in claim 21 wherein the wireless communication module comprises a Bluetooth Low Energy (BTLE) communication module and wherein the data structure comprises a Generic Attribute Profile (GATT) data structure.

26. The apparatus as in claim 23 further comprising application logic to transmit a first data packet to the second wireless communication module by writing the first data packet, N bits at a time, to the second value buffer associated with the second characteristic.

27. The apparatus as in claim 26 wherein the second wireless communication module is to read the first data packet, N bits at a time, from the second value buffer associated with the second characteristic.

28. The apparatus as in claim 27 wherein to read the first data packet, the second wireless communication module is to transmit a first command to the first

wireless communication module including the characteristic ID of the second characteristic.

29. The apparatus as in claim 26 wherein the application logic is to receive a second data packet from the second wireless communication module by reading the second data packet, N bits at a time, from the first value buffer.

30. The apparatus as in claim 29 wherein to write the second data packet to the first value buffer, the second wireless communication module is to transmit a first command to the first wireless communication module including the characteristic ID of the first characteristic.

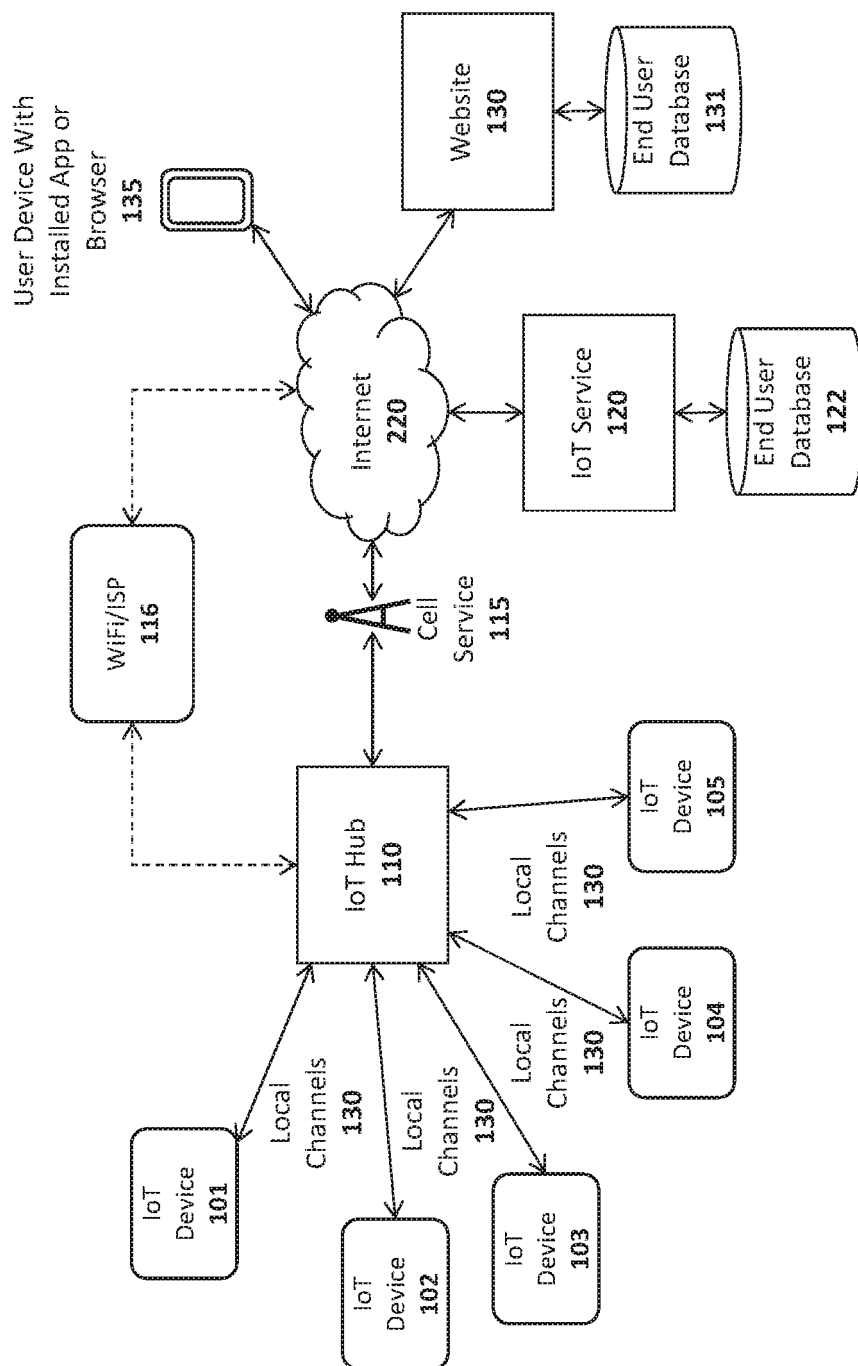


FIG. 1A

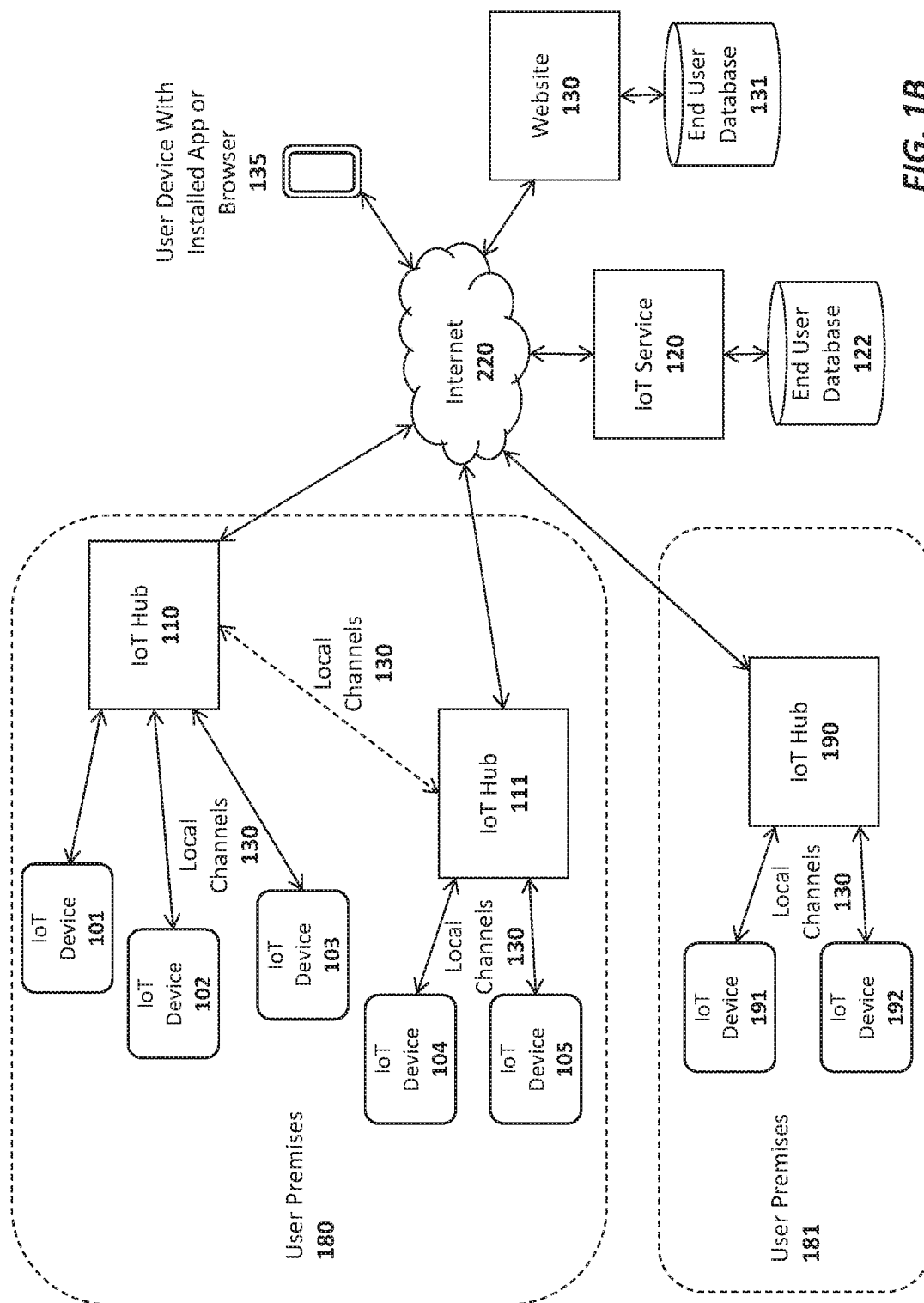


FIG. 1B

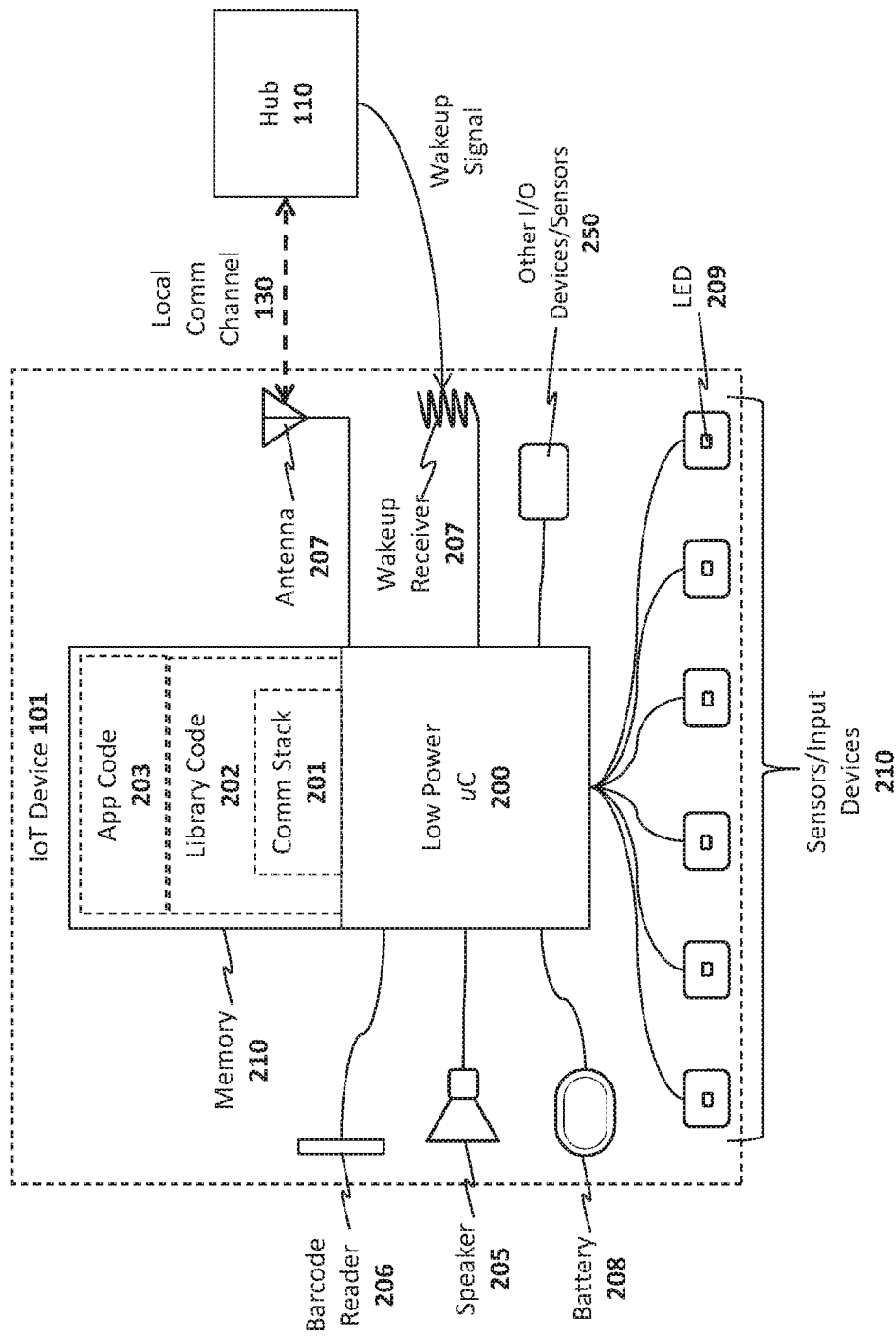


FIG. 2

4/30

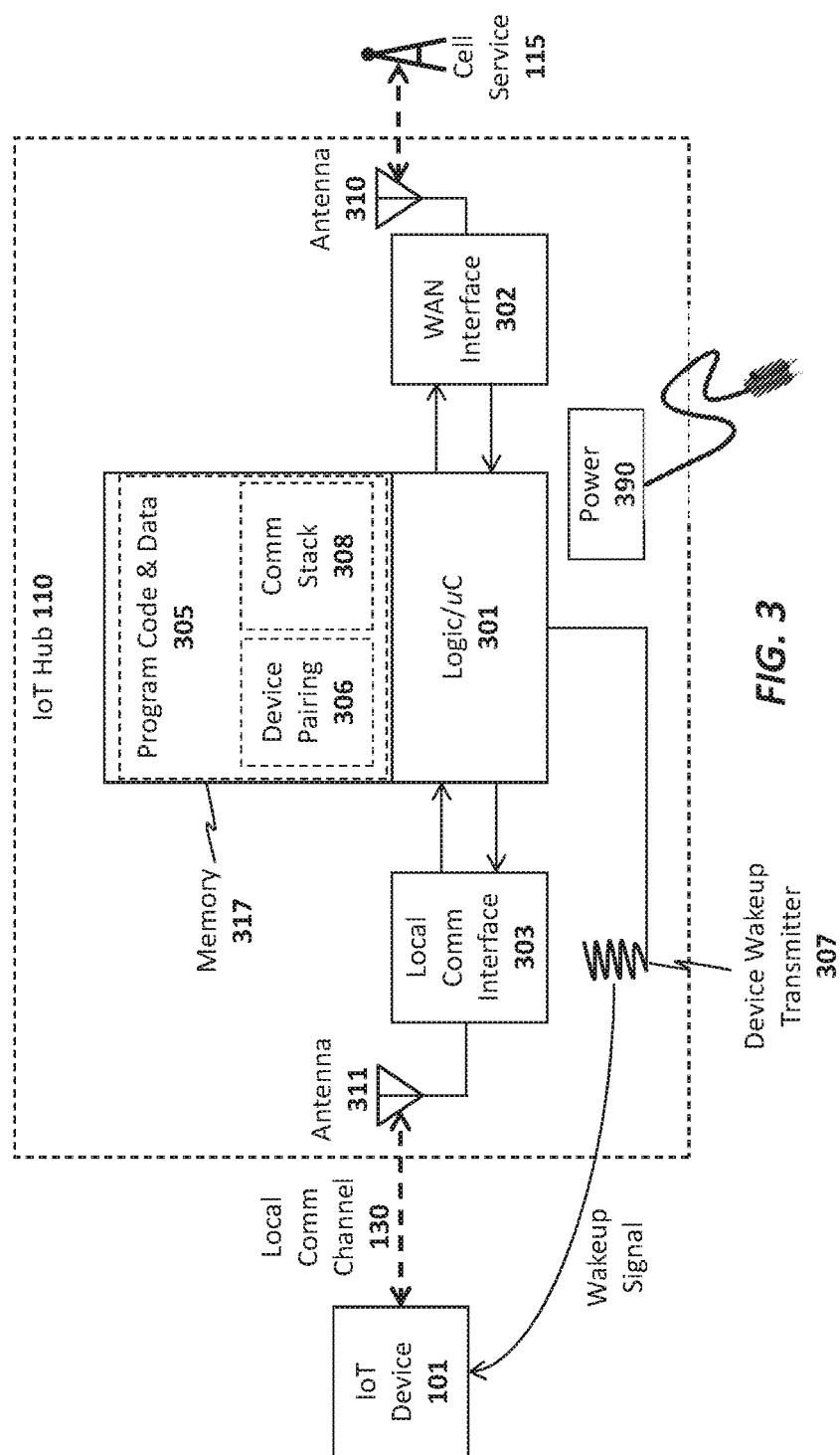


FIG. 3

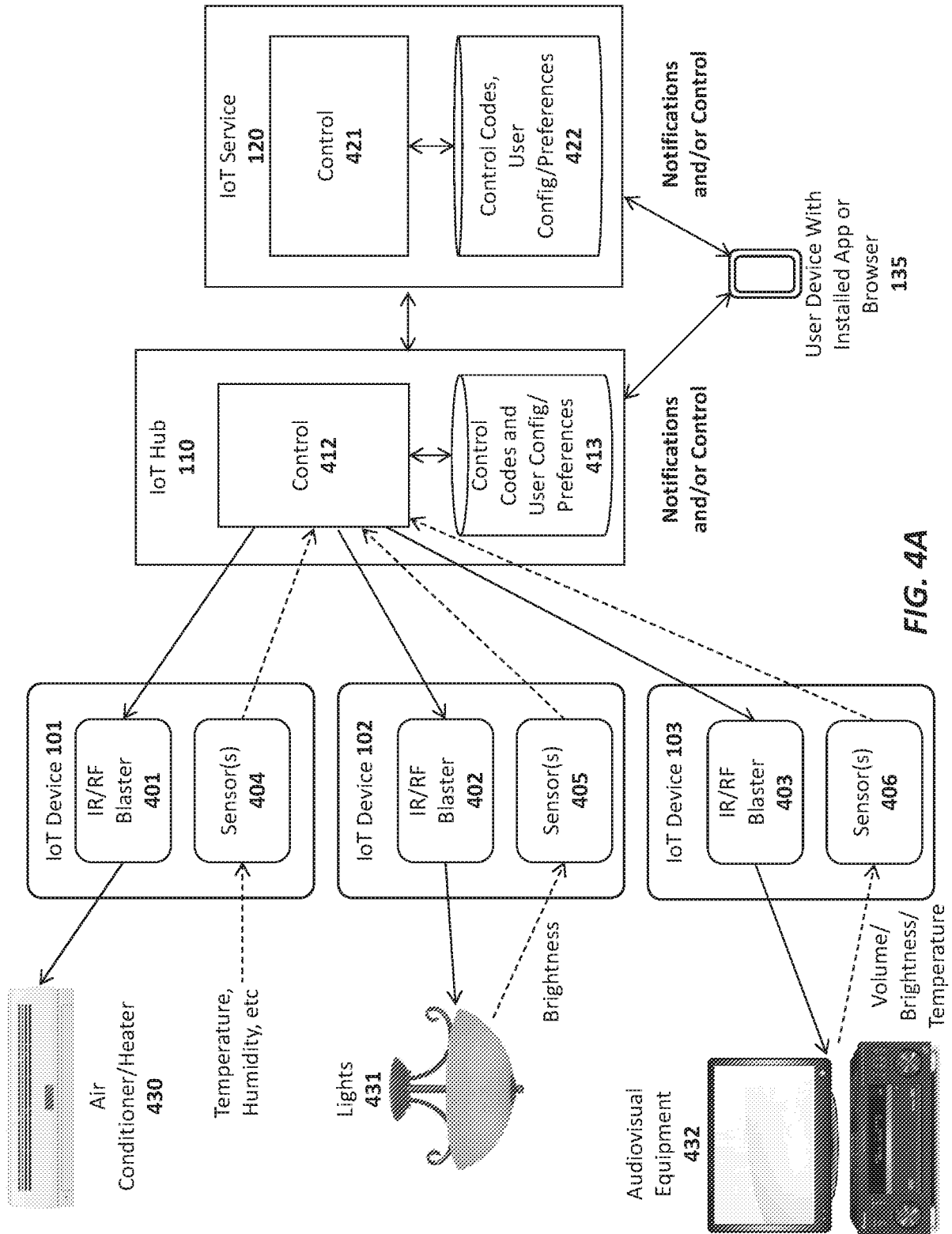


FIG. 4A

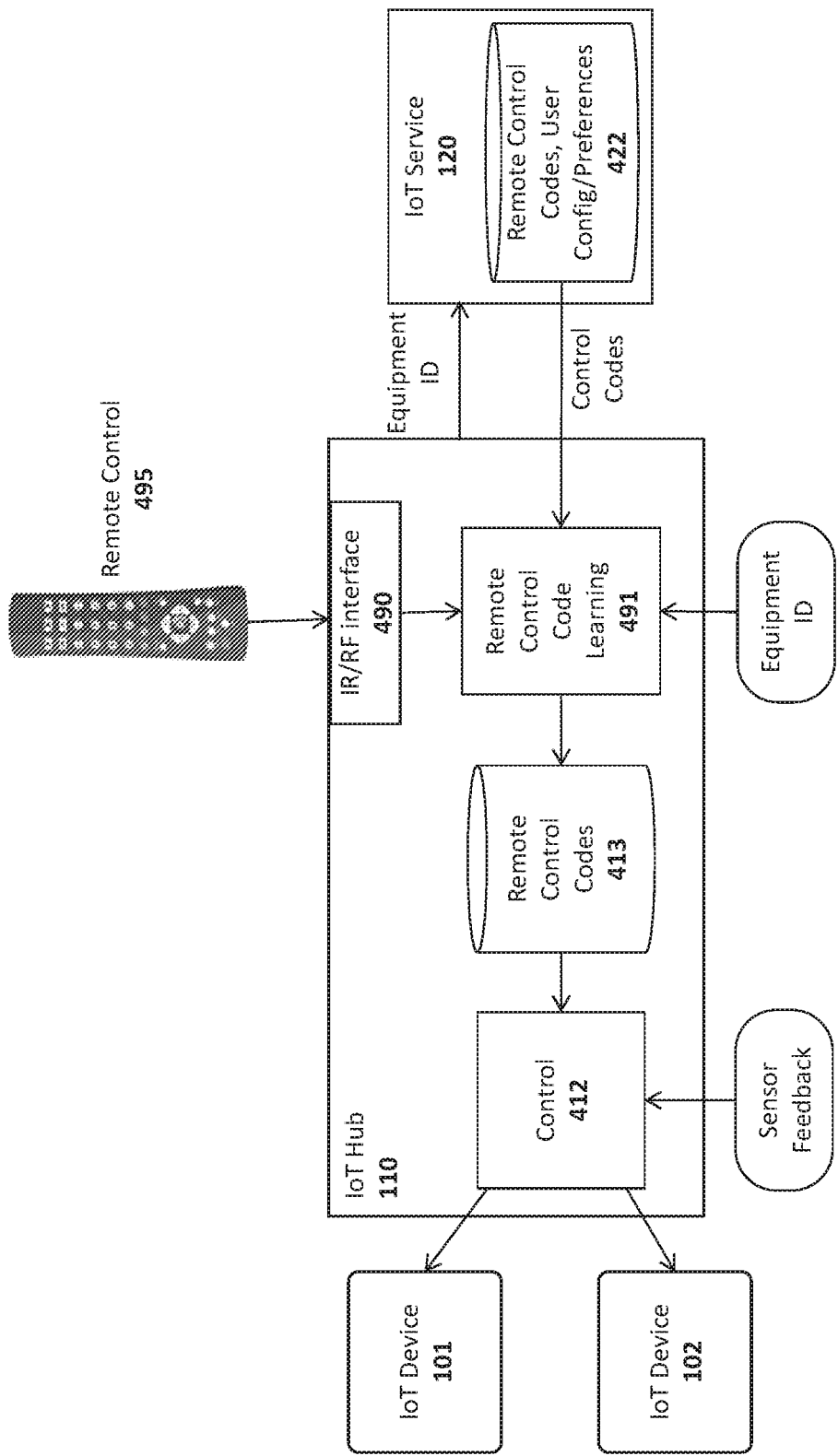


FIG. 4B

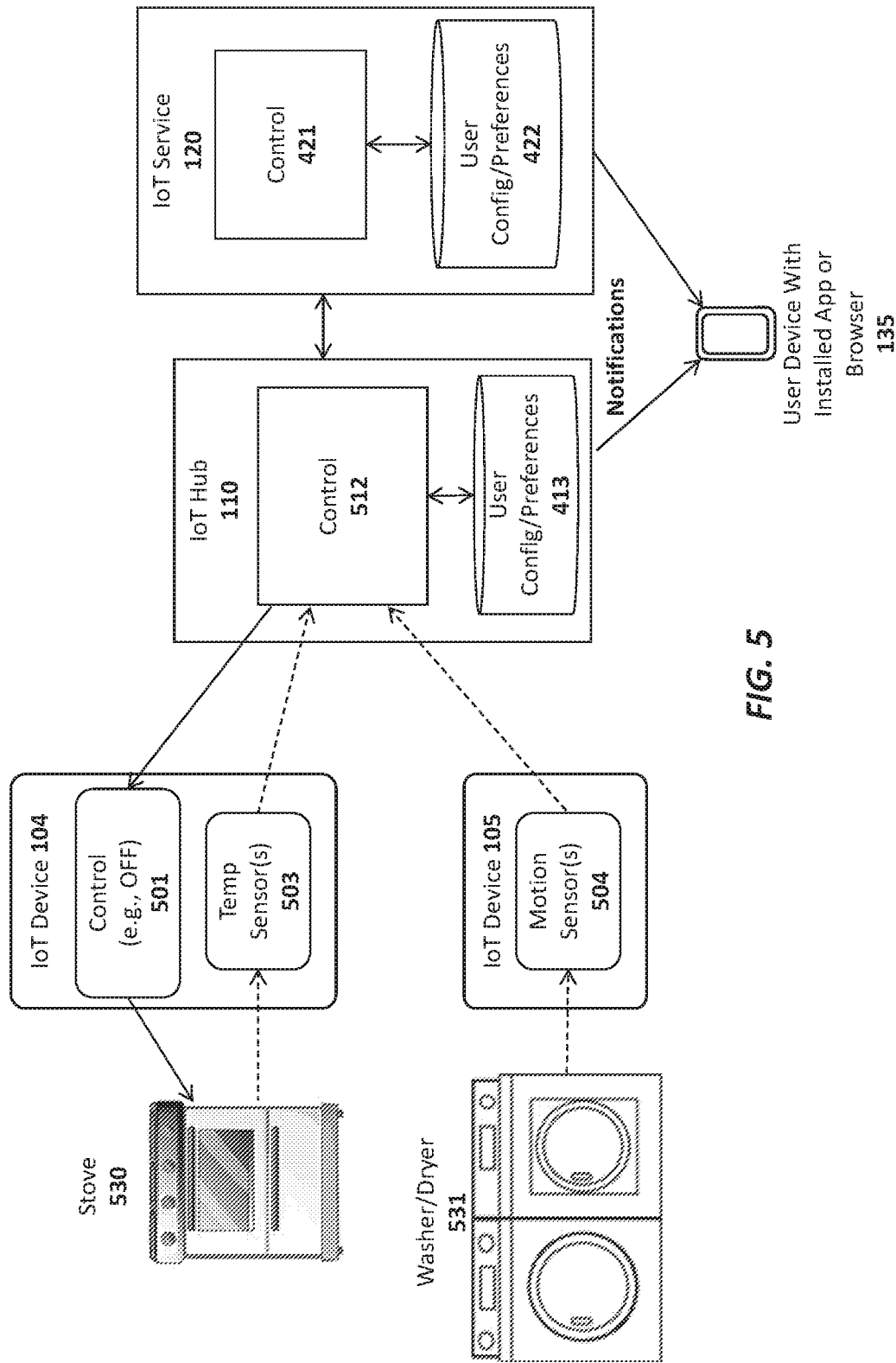


FIG. 5

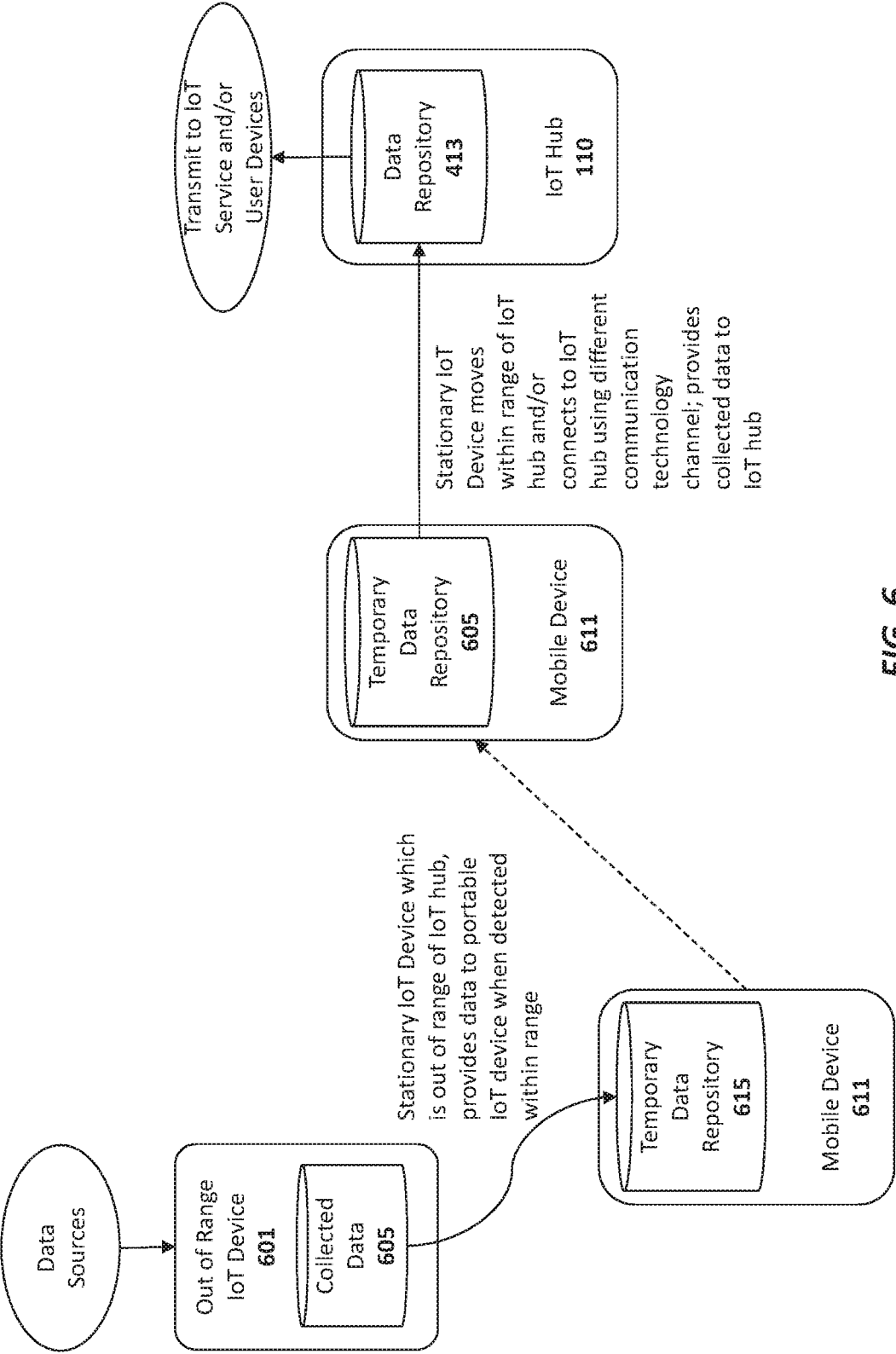


FIG. 6

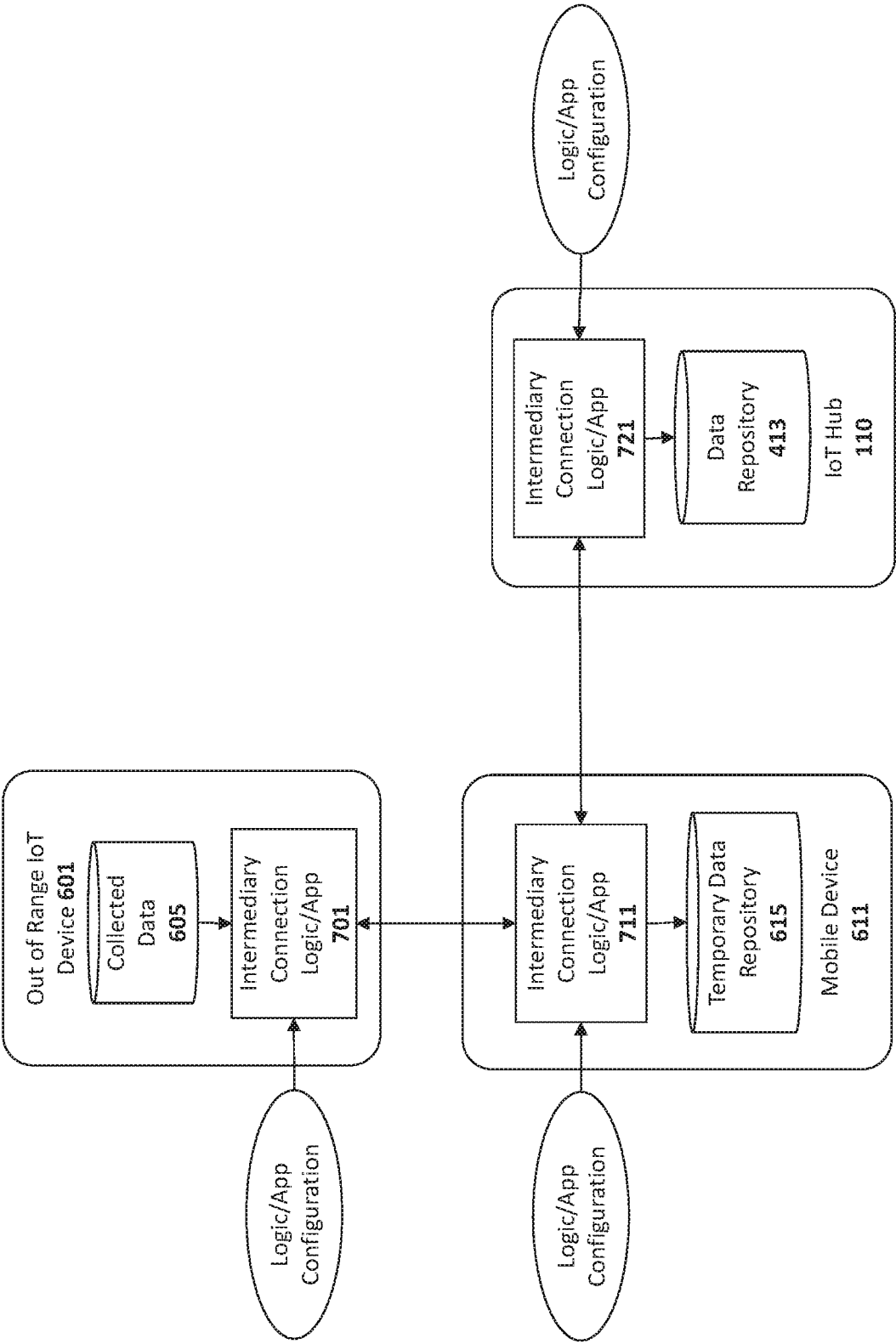
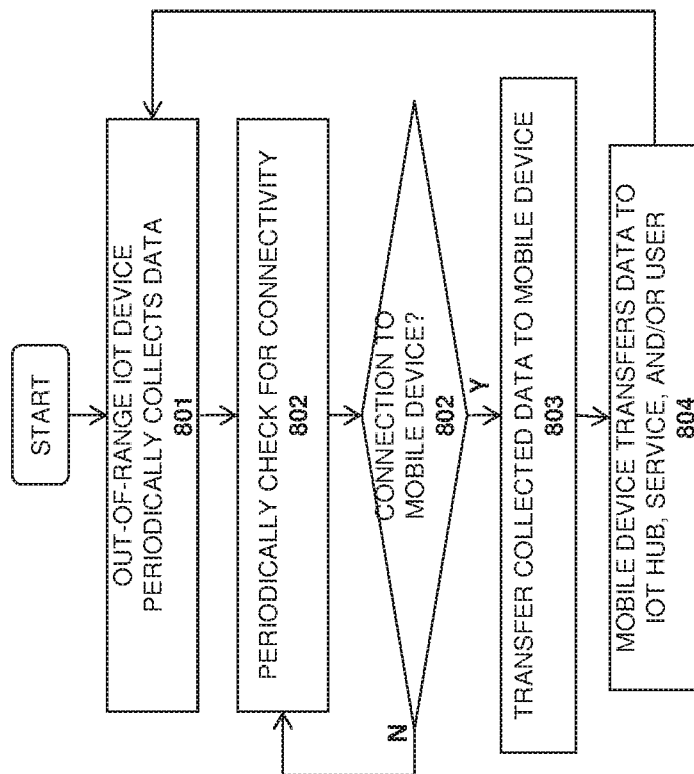


FIG. 7

10/30

**FIG. 8**

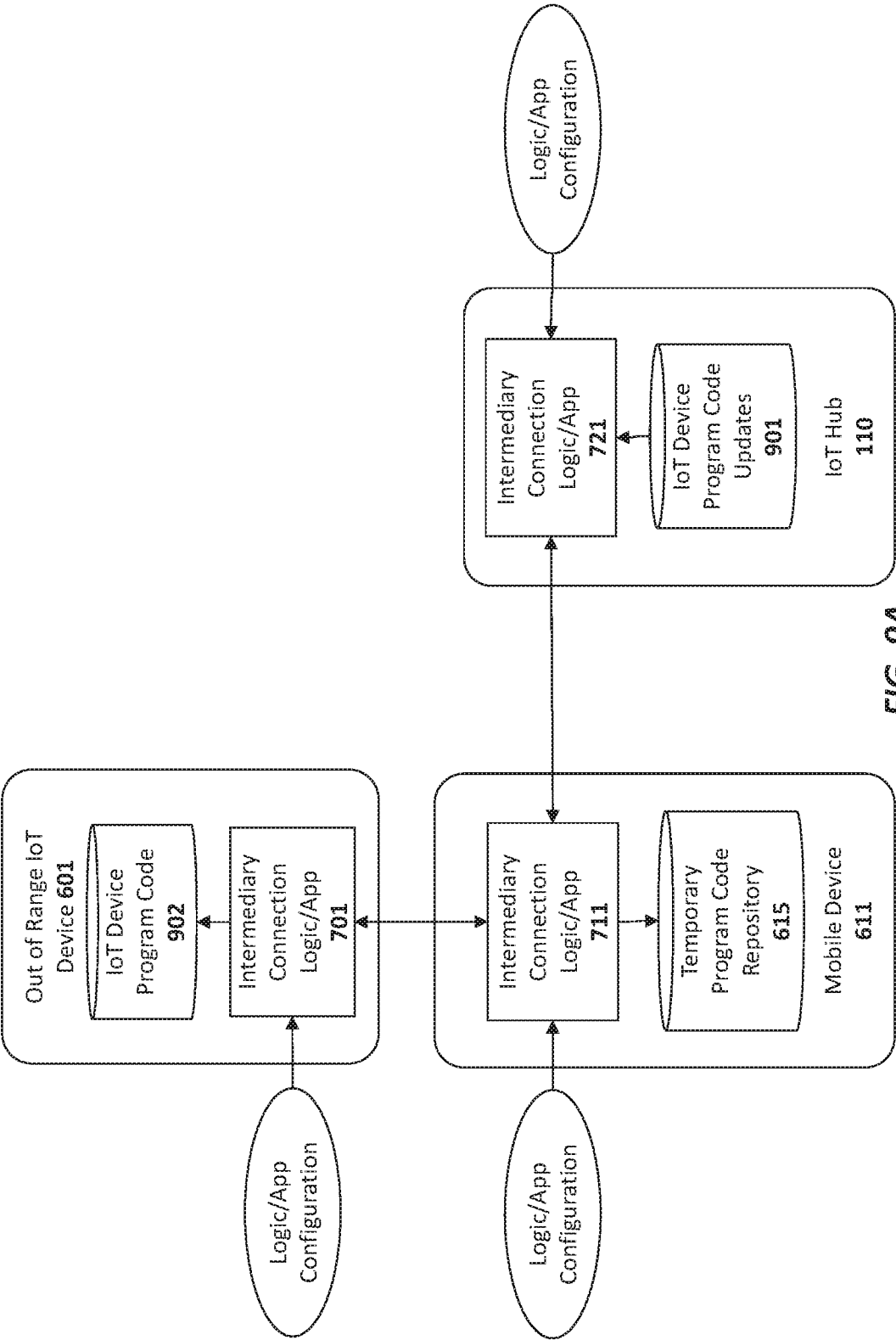
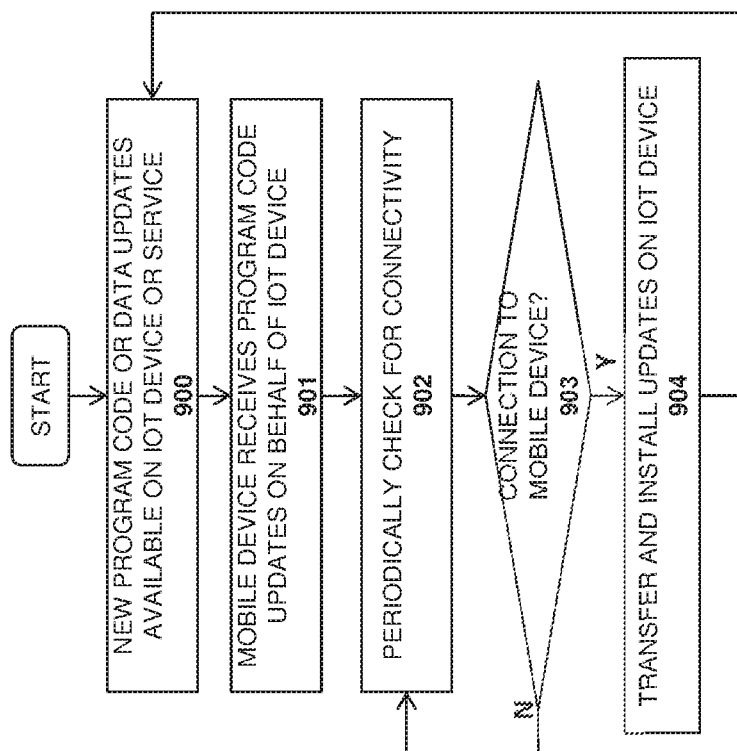


FIG. 9A

12/30

**FIG. 9B**

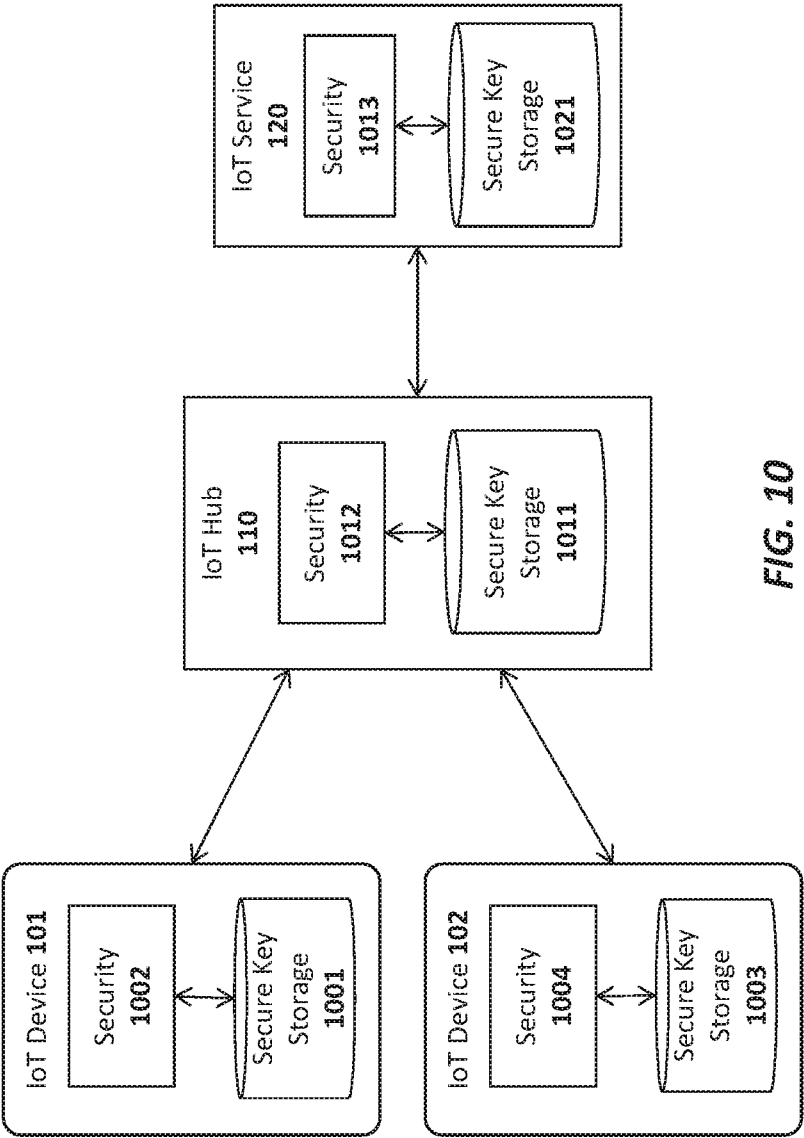


FIG. 10

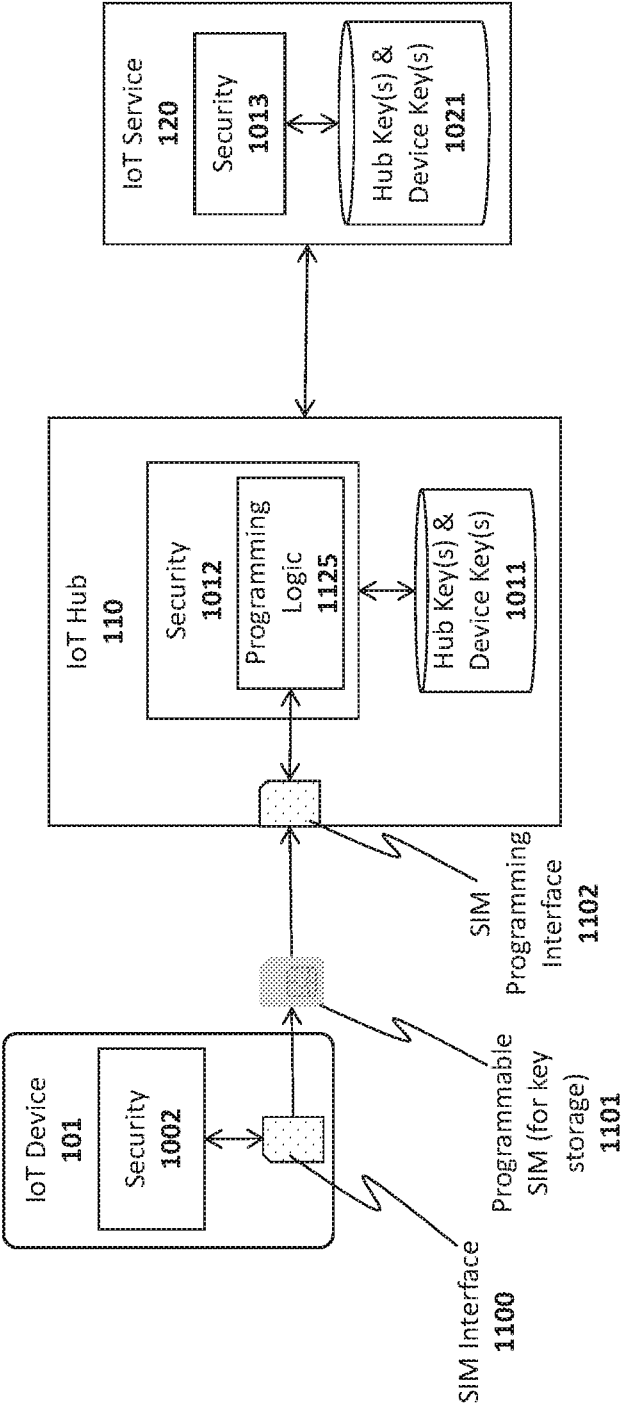


FIG. 11

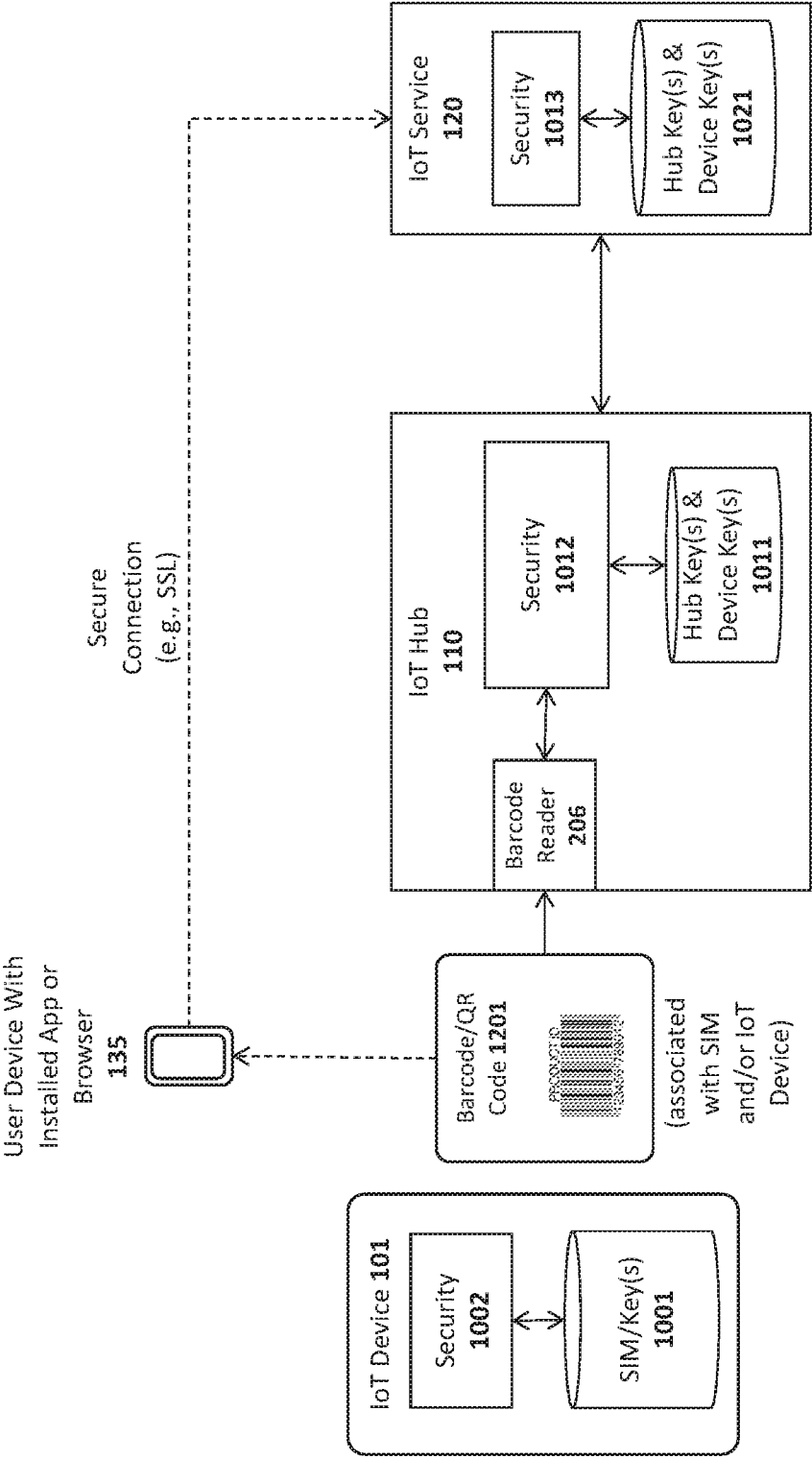


FIG. 12A

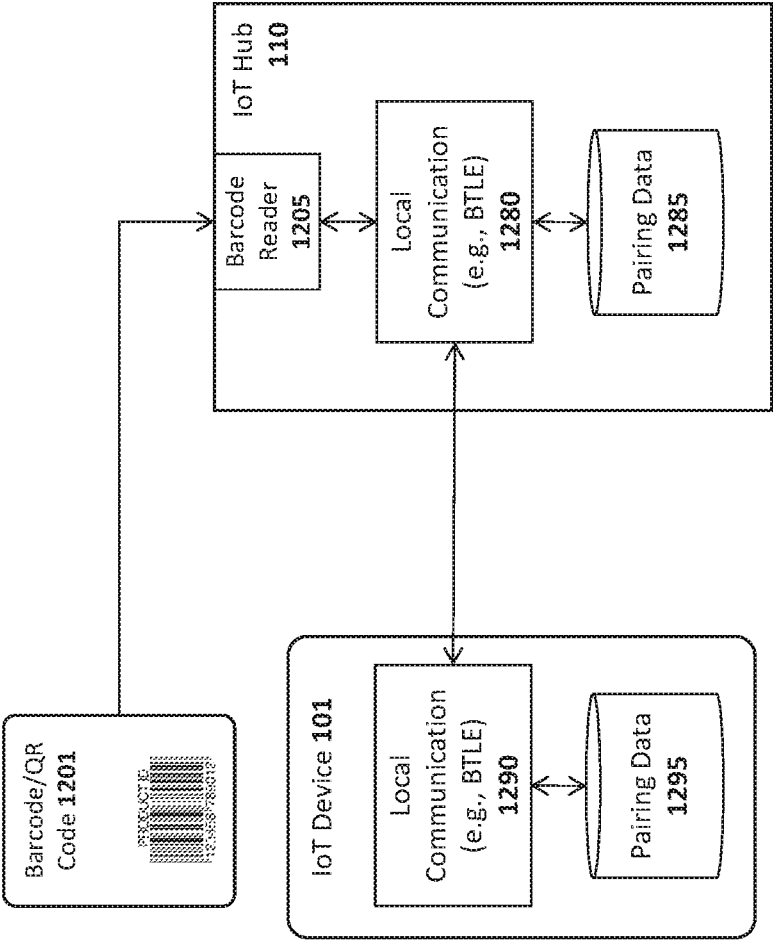


FIG. 12B

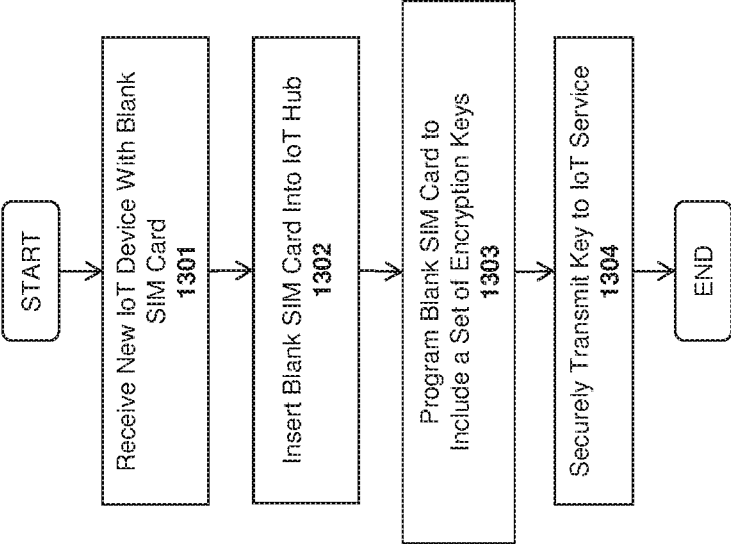
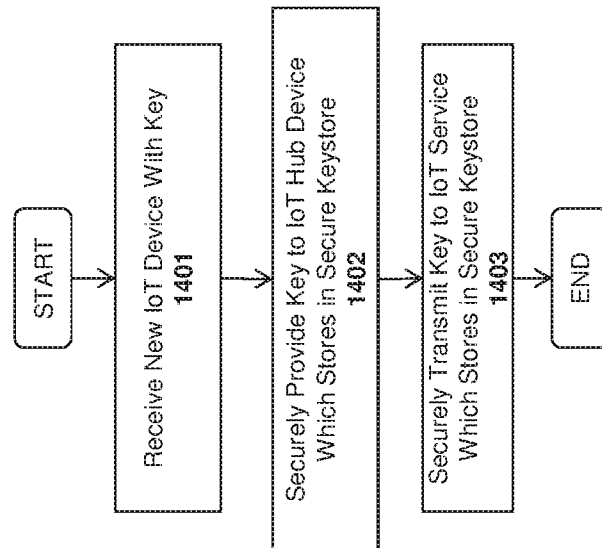


Fig. 13

**Fig. 14**

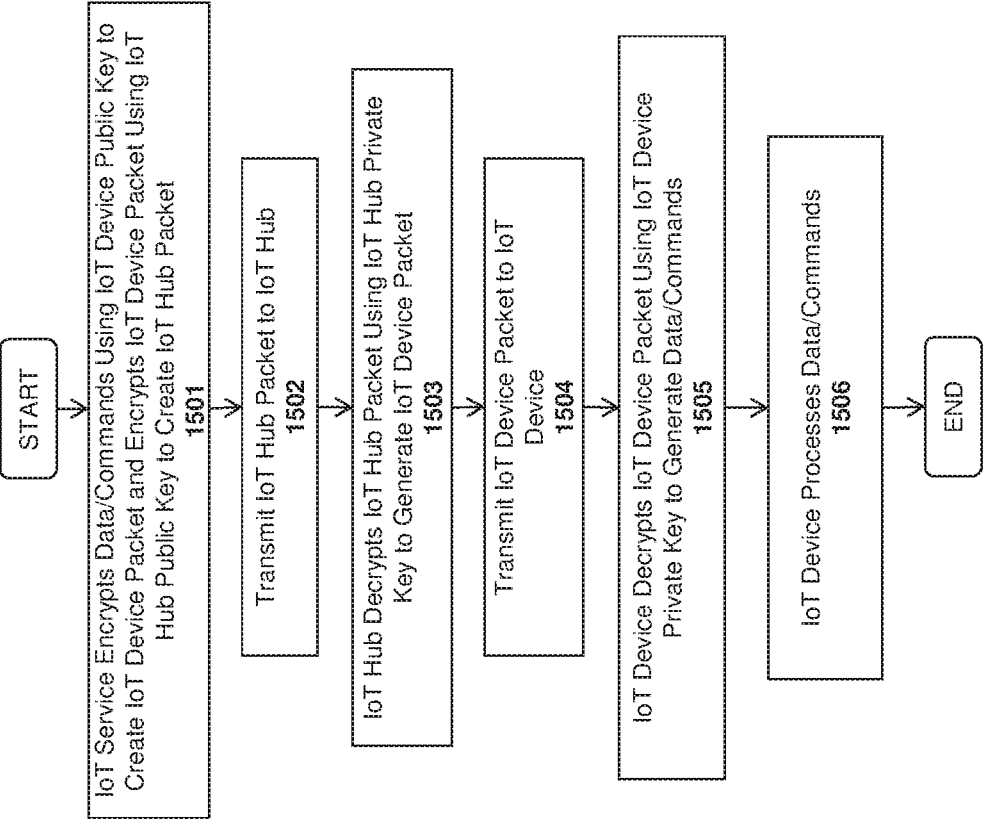


Fig. 15

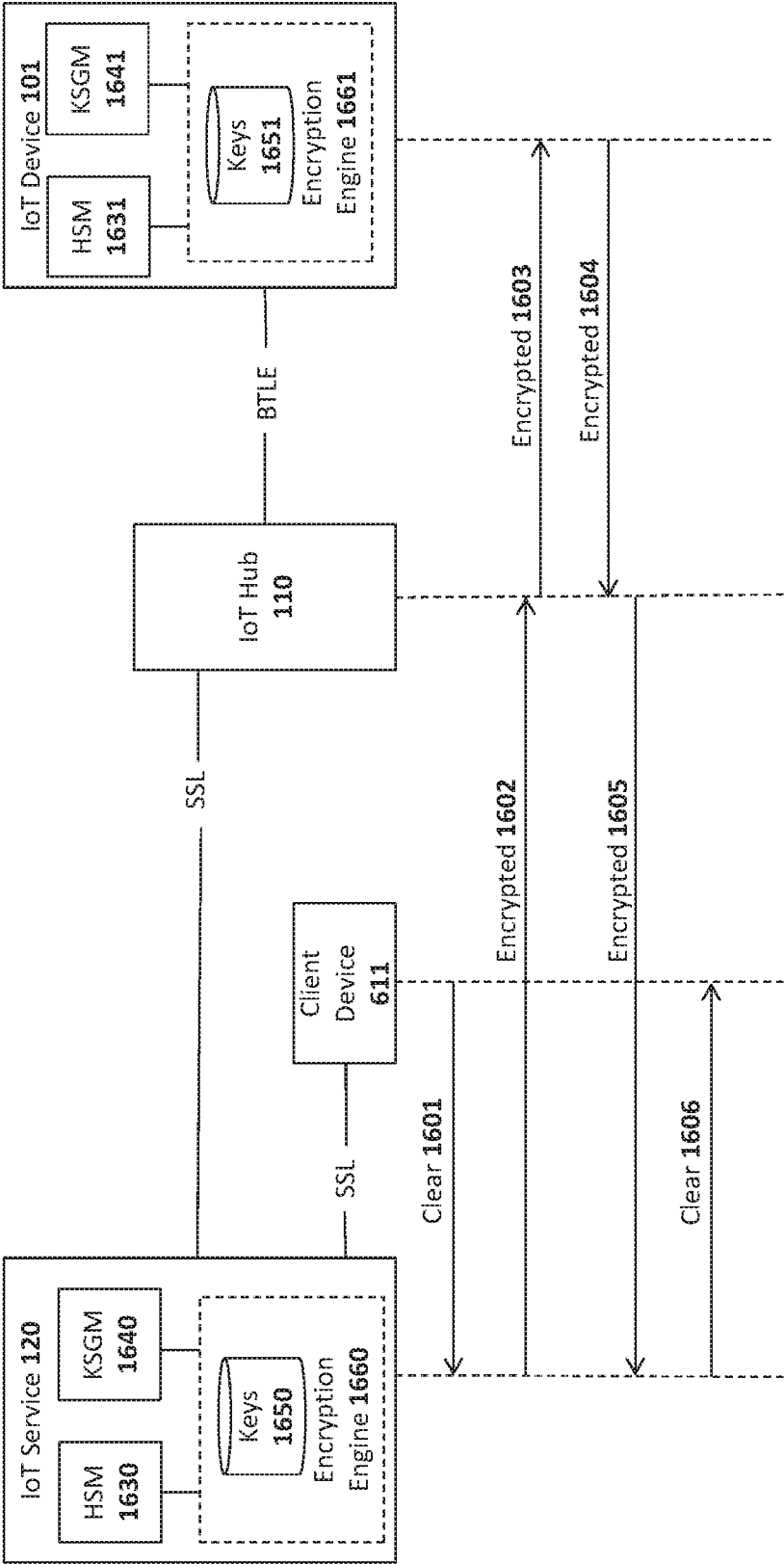


Fig. 16A

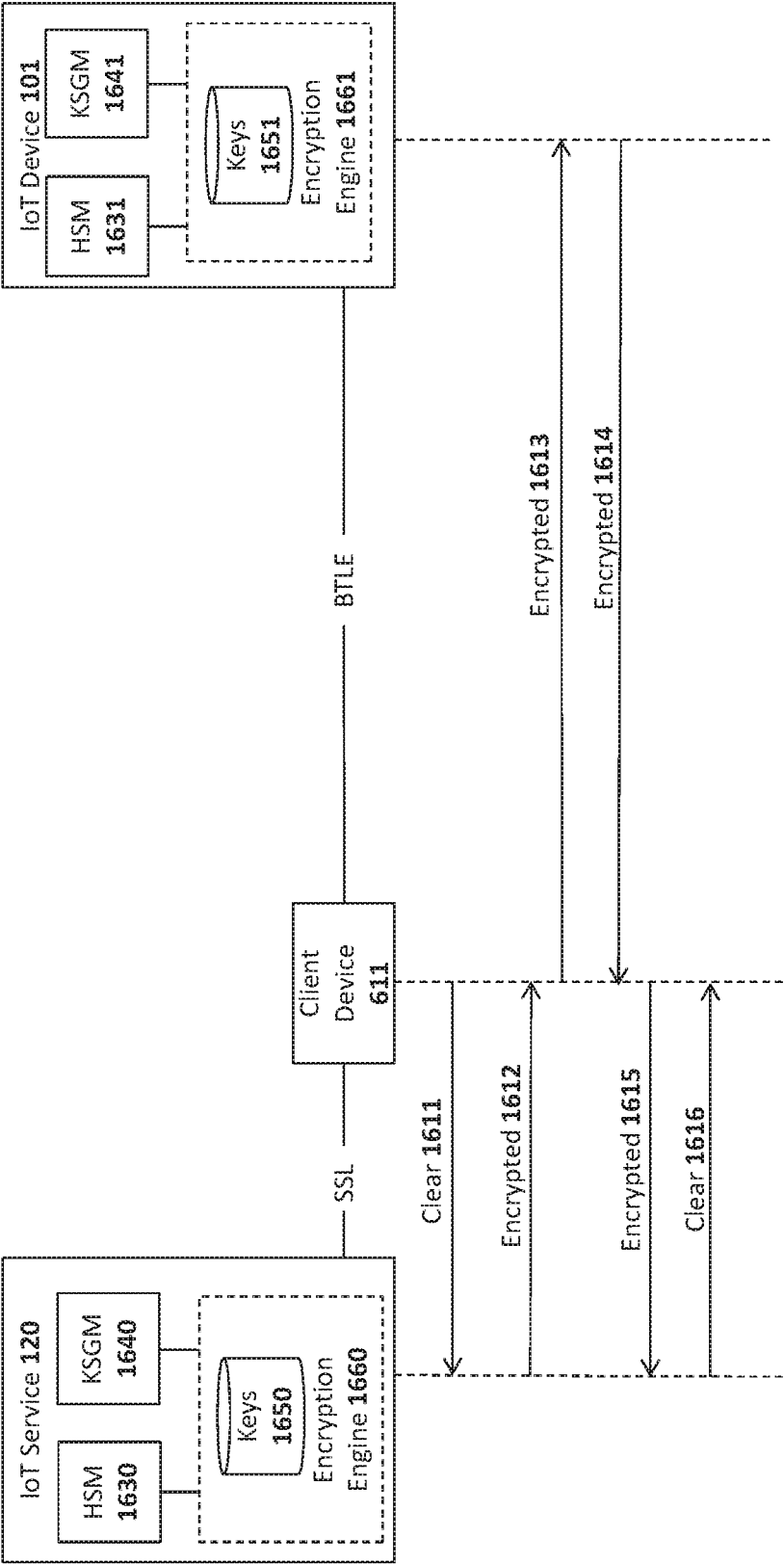


Fig. 16B

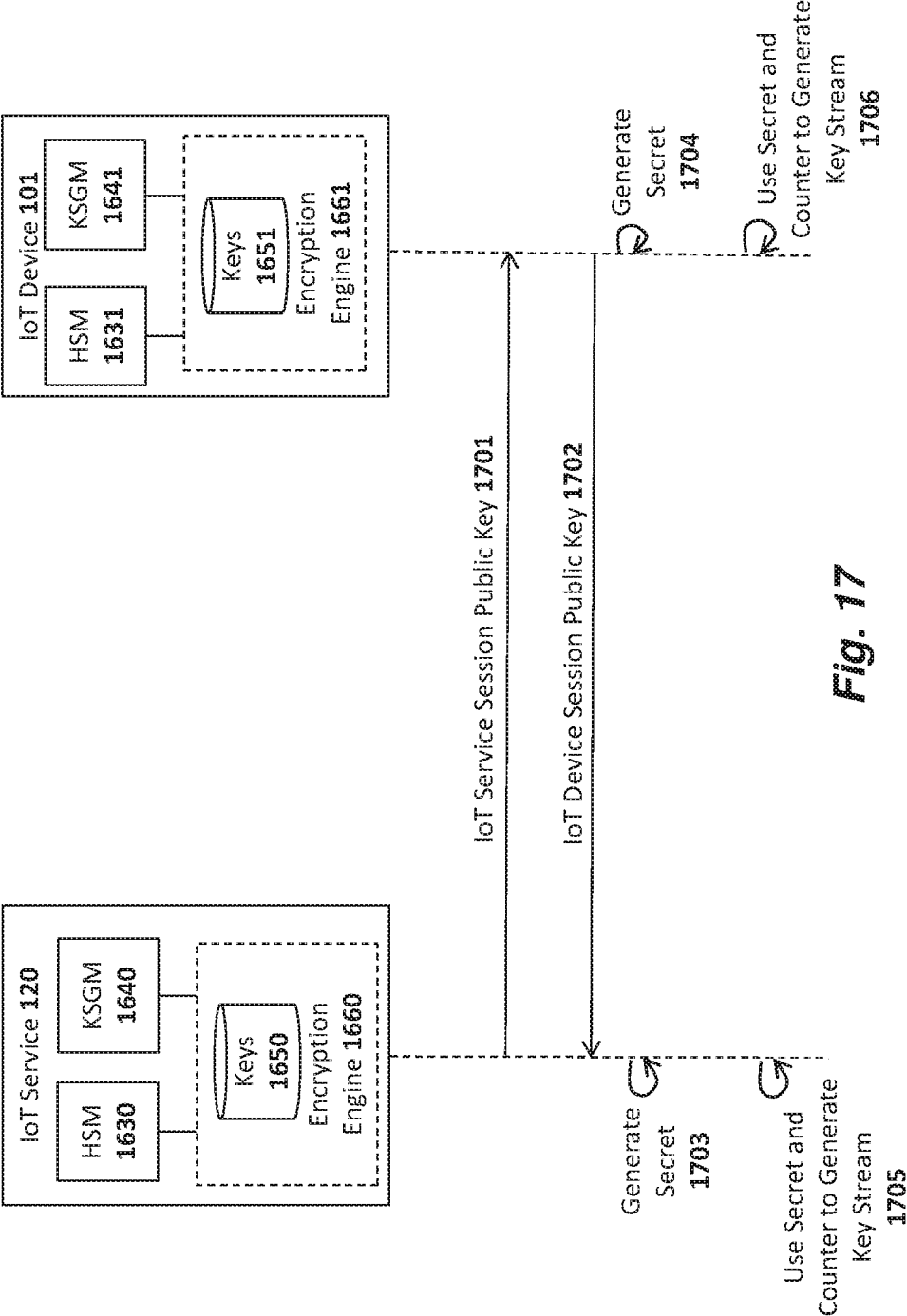


Fig. 17

4 bytes	N bytes	6 bytes
Counter 1800	Encrypted Data 1801	Tag 1802

Fig. 18

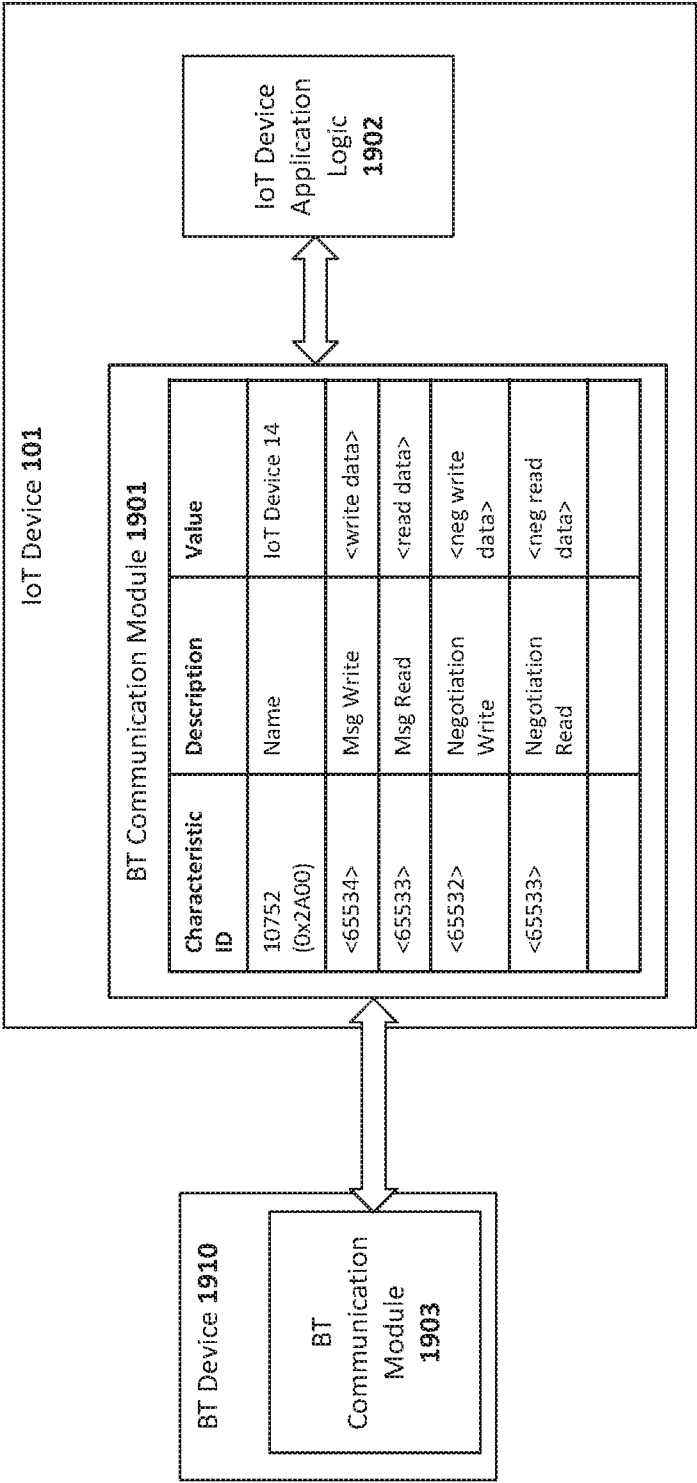


Fig. 19

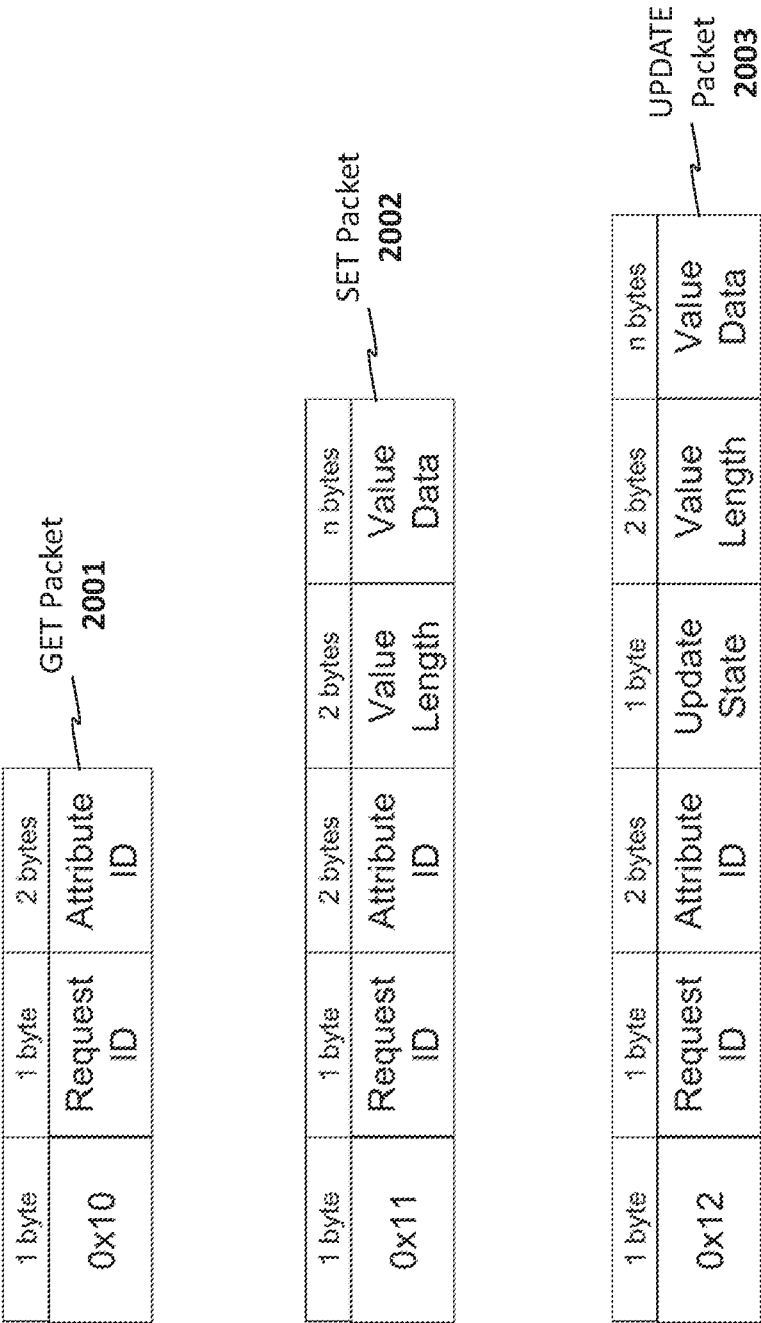


Fig. 20

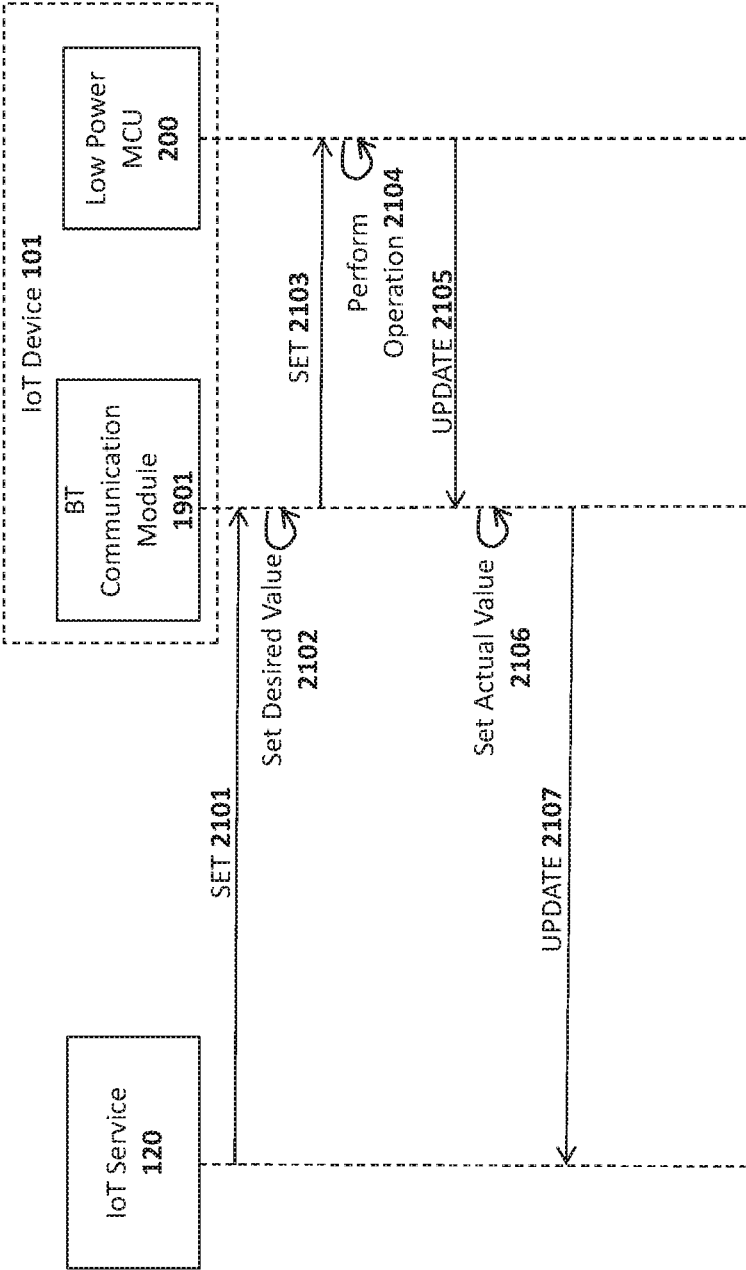


Fig. 21

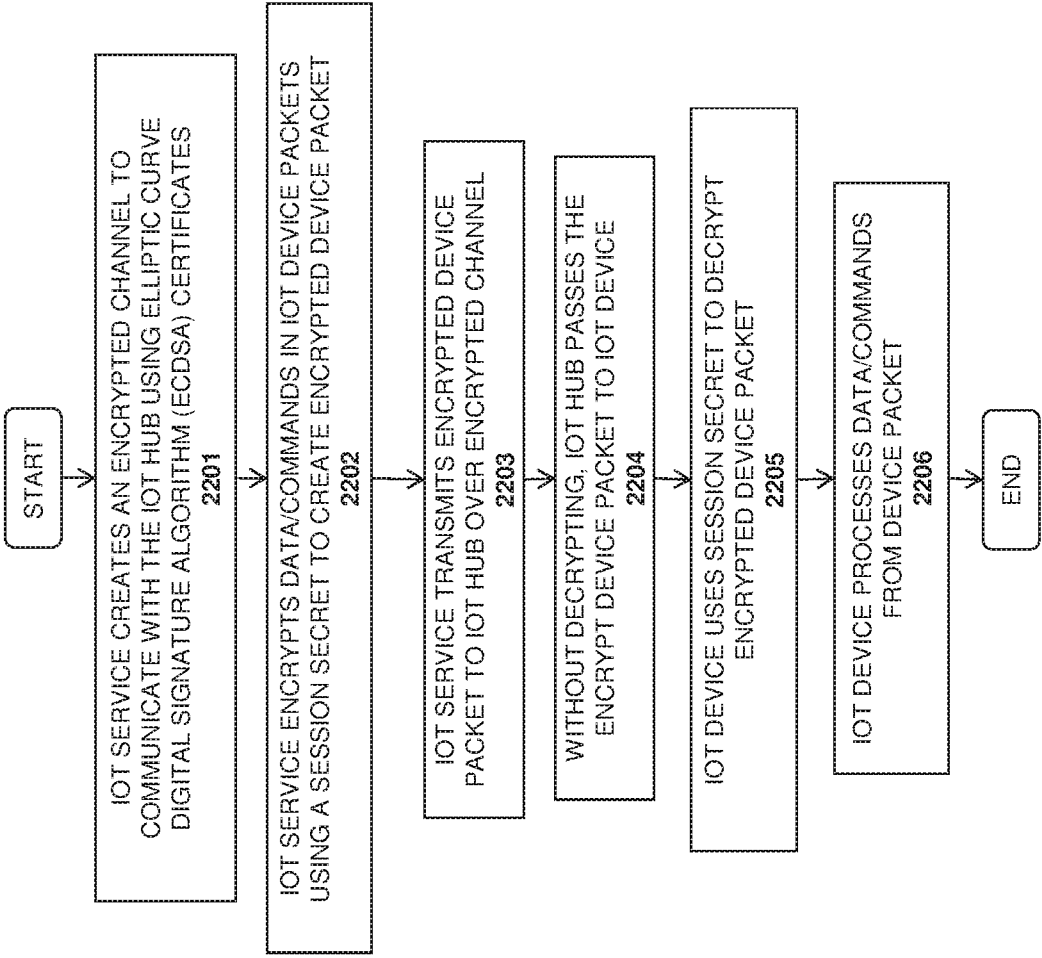
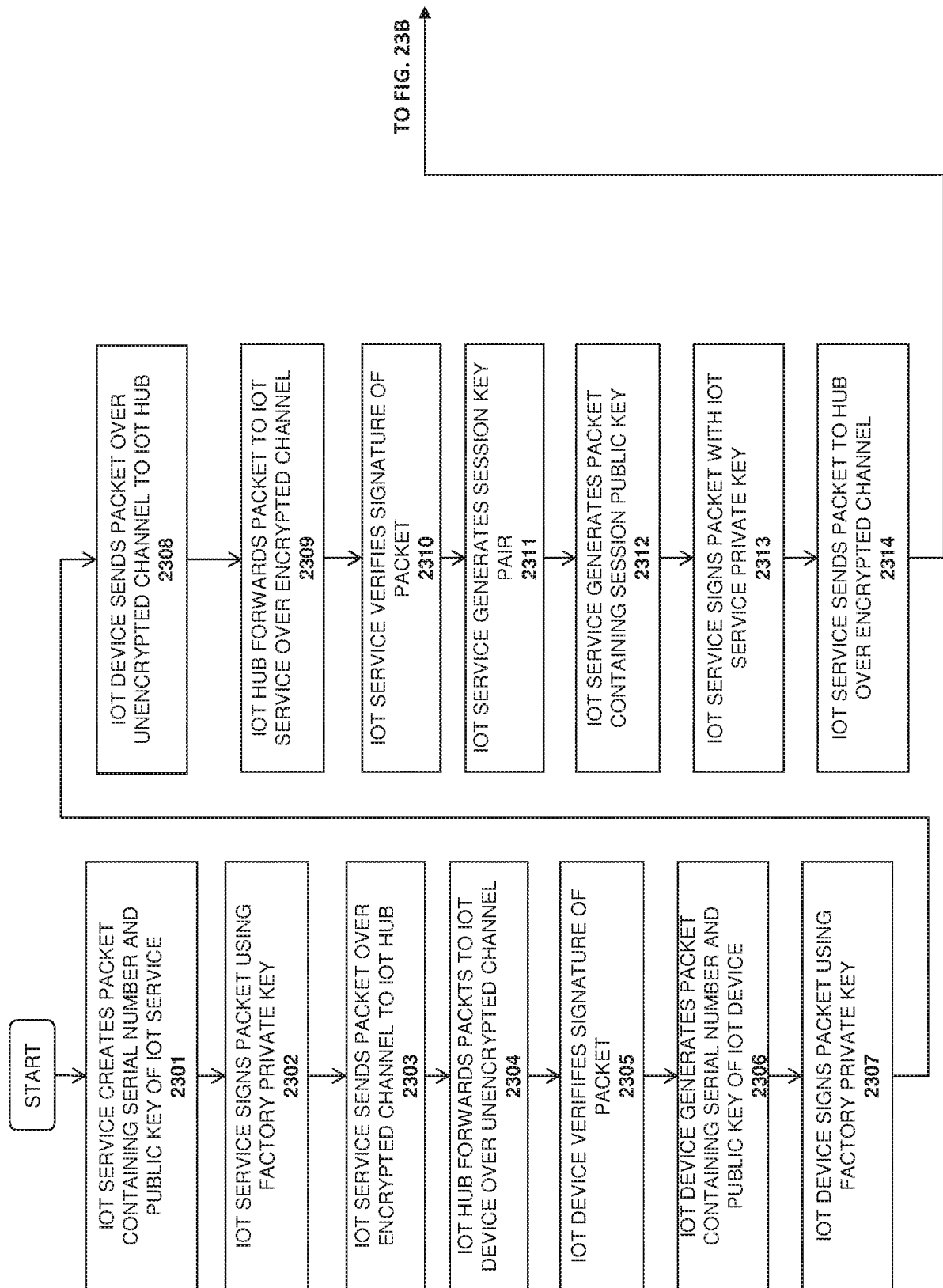
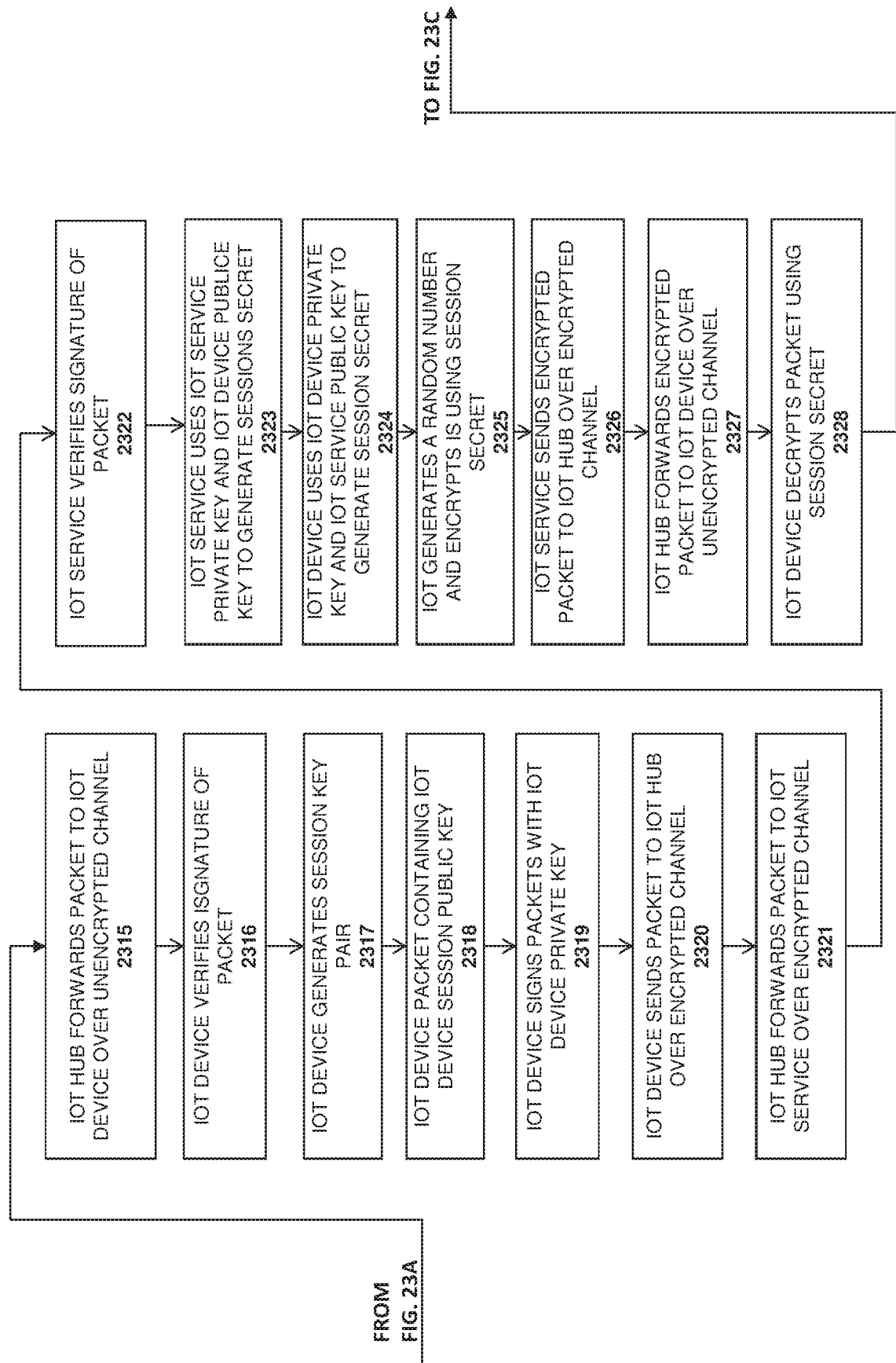


Fig. 22

28/30

**Fig. 23A**

29/30



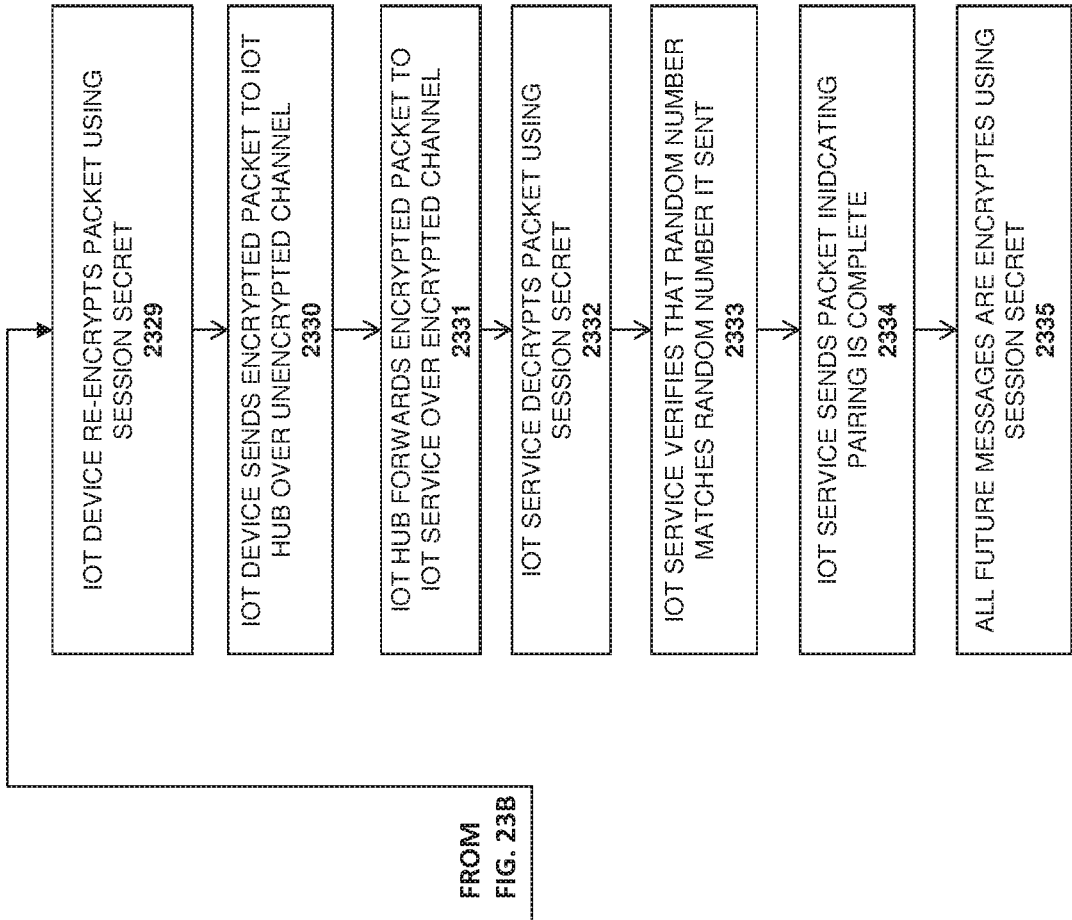


Fig. 23C

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US16/40819

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.:
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:

Group I: Claims 1-20; Group II: Claims 21-30

-Continued in extra sheet-

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.:
1-20

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.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US16/40819

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - G06F 21/60; H04L 99/08, 9/14, 9/18; H04W 84/14 (2016.01)

CPC - G06F 21/602, 21/645; H04L 99/0802, 9/0811, 9/0813, 9/0816, 9/14, 9/3066; 12/28, 29/06; H04W 84/14

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)

IPC(8): G06F 21/60; H04L 9/00, 9/08, 9/14, 9/18, 9/22, 9/28, 9/30; H04L 12/28, 29/06; H04W 8/18, 84/14 (2016.01) CPC: G06F 21/60, 21/602, 21/606, 21/645; H04L 9/08, 9/0802, 9/0811, 9/0813, 9/0816, 9/083, 9/14, 9/28, 9/3066; 12/28, 29/06; H04W 8/18, 84/14

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)

PatSeer (US, EP, WO, JP, DE, GB, CN, FR, KR, ES, AU, IN, CA, Other Countries (INPADOC), RU, AT, CH, TH, BR, PH); EBSCO; IEEE/IEEEExplore; Google/Google Scholar; Keywords: IoT, device, mobile, key, private, public, secret, server, service, host, router, hub, gateway, generate, derive, key stream, transmitter, receiver, sender, encryption, decryption, security, hardware

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2012/0331287 A1 (BOWMAN, R. et al.) 27 September 2012; figures 5, 7; paragraphs [0055], [0059].	1-20
Y	US 2014/0270166 A1 (QUALCOMM INCORPORATED) 18 September 2014; figure 2; paragraphs [0009], [0066].	1-20
Y	CN 1054693 A (TELEFONAKTIEBOLAGET LM ERICSSON) 18 September 1991; see machine translation.	3-10, 13-20
Y	US 2011/0200188 A1 (GHOUTI, L. et al.) 18 August 2011; paragraph [0218]; claim 4.	6, 16
Y	US 2015/0148989 A1 (QUALCOMM INCORPORATED) 28 May 2015; paragraphs [0044], [0072], [0129].	10, 20

☐ 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

30 August 2016 (30.08.2016)

Date of mailing of the international search report

31 OCT 2016

Name and mailing address of the ISA/

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, Virginia 22313-1450

Facsimile No. 571-273-8300

Authorized officer

Shane Thomas

PCT Helpdesk: 571-272-4300
PCT OSP: 571-272-7774

---Continued from Box No. III - Observations where unity of inventions is lacking---

This application contains the following inventions or groups of inventions which are not so linked as to form a single general inventive concept under PCT Rule 13.1. In order for all inventions to be examined, the appropriate additional examination fee must be paid.

Group I: Claims 1-20 are directed toward a method for encrypting communications between a mobile device and a service using public and private keys.

Group II: Claims 21-30 are directed toward an apparatus comprising a local communications channel and managing the data structure for the communications.

The inventions listed as Groups I-19 do not relate to a single general inventive concept under PCT Rule 13.1 because, under PCT Rule 13.2, they lack the same or corresponding special technical features for the following reasons:

The special technical features of Group I include a first encryption engine on the IoT service comprising key generation logic to generate a service public key and a service private key; a second encryption engine on the IoT device comprising key generation logic to generate a device public key and a device private key; the first encryption engine to transmit the service public key to the second encryption engine and the second encryption engine to transmit the device public key to the first encryption engine; the first encryption engine to use the device public key and the service private key to generate a secret; the second encryption engine to use the service public key and the device private key to generate the same secret; and wherein once the secret is generated, the first encryption engine and the second encryption engine encrypt and decrypt data packets transmitted between the first encryption engine and the second encryption engine using the secret or using data structures derived from the secret, which are not present in Group II.

The special technical features of Group II include a data structure managed by the first local wireless communication module, the data structure comprising a set of characteristics, each characteristic identified with an characteristic ID and having a value buffer associated therewith containing data associated with the characteristic; the first wireless communication module to form a first network socket using the data structure by defining a first characteristic for receiving data packets transmitted by the second local wireless communication module and a second characteristic for transmitting outgoing data packets to the second local wireless communication modules; wherein the incoming data packets are to be temporarily buffered in a first value buffer associated with the first characteristic and wherein the outgoing packets are to be temporarily buffered in a second value buffer associated with the second characteristic, which are not present in Group I.

The common technical feature shared by Groups I-II is an Internet of Things (IoT) service to establish communication with an IoT device through an IoT hub or a mobile user device.

However, this common feature is previously disclosed by US 2003/0039261 A1 (PURPURA). Purpura discloses an Internet of Things (IoT) service to establish communication with an IoT device through an IoT hub or a mobile user device (providing communications with an internet services server (IoT device) via a portable network mini-hub to a plurality of mobile network devices (mobile user device); paragraph [0020]).

Since the common technical feature is previously disclosed by the Purpura reference, this common feature is not special and so Groups I-II lack unity.



(12)发明专利申请

(10)申请公布号 CN 107710216 A

(43)申请公布日 2018.02.16

(21)申请号 201680038597.6

(22)申请日 2016.07.01

(30)优先权数据

14/791,371 2015.07.03 US

14/791,373 2015.07.03 US

(85)PCT国际申请进入国家阶段日

2017.12.29

(86)PCT国际申请的申请数据

PCT/US2016/040819 2016.07.01

(87)PCT国际申请的公布数据

W02017/007725 EN 2017.01.12

(71)申请人 阿费罗有限公司

地址 美国加利福尼亚州

(72)发明人 乔·布里特 奥马尔·扎卡里亚

斯科特·齐默尔曼

(74)专利代理机构 北京律盟知识产权代理有限公司 11287

代理人 林斯凯

(51)Int.Cl.

G06F 21/60(2006.01)

H04L 9/14(2006.01)

H04L 9/18(2006.01)

H04W 84/14(2006.01)

权利要求书3页 说明书25页 附图28页

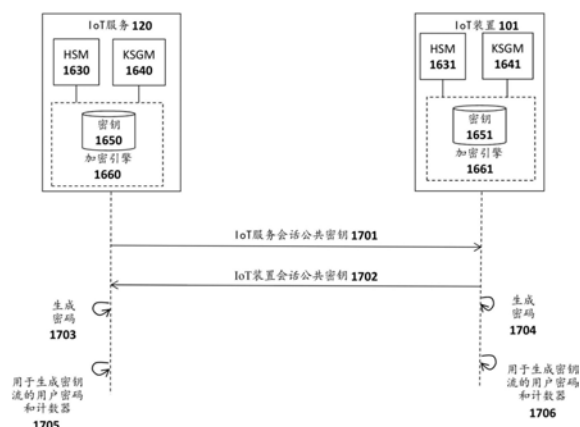
(54)发明名称

用于在物联网(IoT)系统中建立安全通信信道的设备和方法

(57)摘要

本发明描述了用于IoT装置和IoT服务之间的安全通信的设备和方法。例如,系统的一个实施方案包括:物联网(IoT)服务,所述物联网服务用于通过IoT中心或移动用户装置建立与IoT装置的通信;所述IoT服务上的第一加密引擎,所述第一加密引擎包括用于生成服务公共密钥和服务私有密钥的密钥生成逻辑;所述IoT装置上的第二加密引擎,所述第二加密引擎包括用于生成装置公共密钥和装置私有密钥的密钥生成逻辑;所述第一加密引擎用于将所述服务公共密钥传输至所述第二加密引擎并且所述第二加密引擎用于将所述装置公共密钥传输至所述第一加密引擎;所述第一加密引擎使用所述装置公共密钥和所述服务私有密钥来生成密码;所述第二加密引擎使用所述服务公共密钥和所述装置私有密钥来生成相同的密码;并且其中一旦生成所述密码,则所述第一加密引擎和所述第二加密引擎使

用所述密码或使用从所述密码得出的数据结构加密和解密在所述第一加密引擎和所述第二加密引擎之间传输的数据包。



1. 一种系统,包括:

物联网 (IoT) 服务,所述物联网服务用于通过IoT中心或移动用户装置建立与IoT装置的通信;

所述IoT服务上的第一加密引擎,所述第一加密引擎包括用于生成服务公共密钥和服务私有密钥的密钥生成逻辑;

所述IoT装置上的第二加密引擎,所述第二加密引擎包括用于生成装置公共密钥和装置私有密钥的密钥生成逻辑;

所述第一加密引擎用于将所述服务公共密钥传输至所述第二加密引擎并且所述第二加密引擎用于将所述装置公共密钥传输至所述第一加密引擎;

所述第一加密引擎使用所述装置公共密钥和所述服务私有密钥来生成密码;

所述第二加密引擎使用所述服务公共密钥和所述装置私有密钥来生成相同的密码;并且

其中一旦生成所述密码,则所述第一加密引擎和所述第二加密引擎使用所述密码或从所述密码得出的数据结构加密和解密在所述第一加密引擎和所述第二加密引擎之间传输的数据包。

2. 根据权利要求1所述的系统,其中所述密钥生成逻辑包括硬件安全模块 (HSM)。

3. 根据权利要求1所述的系统,其中从所述密码得出的所述数据结构包括由所述第一加密引擎生成的第一密钥流和由所述第二加密引擎生成的第二密钥流。

4. 根据权利要求3所述的系统,还包括与所述第一加密引擎相关联的第一计数器和与所述第二加密引擎相关联的第二计数器,所述第一加密引擎响应于传输到所述第二加密引擎的每个数据包而递增所述第一计数器,并且所述第二加密引擎响应于传输到所述第一加密引擎的每个数据包而递增所述第二计数器。

5. 根据权利要求4所述的系统,其中所述第一加密引擎使用所述第一计数器的当前计数器值和所述密码生成所述第一密钥流,并且所述第二加密引擎使用所述第二计数器的当前计数器值和所述密码生成所述第二密钥流。

6. 根据权利要求5所述的系统,其中所述第一加密引擎包括用于使用所述第一计数器值和所述密码生成所述第一密钥流的椭圆曲线方法 (ECM) 模块,并且所述第二加密引擎包括用于使用所述第一计数器值和第一密码生成所述第二密钥流的ECM模块。

7. 根据权利要求5所述的系统,其中所述第一加密引擎使用所述第一密钥流加密第一数据包以生成第一加密数据包,并且将所述第一加密数据包与所述第一计数器的当前计数器值一起传输到所述第二加密引擎。

8. 根据权利要求7所述的系统,其中所述第二加密引擎使用所述第一计数器的所述当前计数器值和所述密码生成所述第一密钥流,并且使用所述第一密钥流来解密所述加密数据包。

9. 根据权利要求7所述的系统,其中加密所述第一数据包包括将所述第一密钥流与所述第一数据包进行异或运算以生成所述第一加密数据包。

10. 根据权利要求8所述的系统,其中所述IoT装置包括蓝牙低功耗 (BTLE) 通信接口,以用于将所述IoT装置通信地耦接到所述IoT中心或所述移动用户装置,所述IoT中心或所述移动用户装置通过互联网通信地耦接到所述IoT服务。

11.一种方法,包括:

通过IoT中心或移动用户装置建立物联网(IoT)服务和IoT装置之间的通信;

通过所述IoT服务上的第一加密引擎的密钥生成逻辑来生成服务公共密钥和服务私有密钥;

通过所述IoT装置上的第二加密引擎的密钥生成逻辑来生成装置公共密钥和装置私有密钥;

将所述服务公共密钥从所述第一加密引擎传输到所述第二加密引擎,并且将所述装置公共密钥从所述第二加密引擎传输到所述第一加密引擎;

使用所述装置公共密钥和所述服务私有密钥来生成密码;

使用所述服务公共密钥和所述装置私有密钥来生成相同的密码;以及

使用所述密码或使用从所述密码得出的数据结构加密和解密在所述第一加密引擎和所述第二加密引擎之间传输的数据包。

12.根据权利要求11所述的方法,其中所述密钥生成逻辑包括硬件安全模块(HSM)。

13.根据权利要求11所述的方法,其中从所述密码得出的所述数据结构包括由所述第一加密引擎生成的第一密钥流和由所述第二加密引擎生成的第二密钥流。

14.根据权利要求13所述的方法,其中第一计数器与所述第一加密引擎相关联并且第二计数器与所述第二加密引擎相关联,所述第一加密引擎响应于传输到所述第二加密引擎的每个数据包而递增所述第一计数器,并且所述第二加密引擎响应于传输到所述第一加密引擎的每个数据包而递增所述第二计数器。

15.根据权利要求14所述的方法,其中所述第一加密引擎使用所述第一计数器的当前计数器值和所述密码生成所述第一密钥流,并且所述第二加密引擎使用所述第二计数器的当前计数器值和所述密码生成所述第二密钥流。

16.根据权利要求15所述的方法,其中所述第一加密引擎包括用于使用所述第一计数器值和所述密码生成所述第一密钥流的椭圆曲线方法(ECM)模块,并且所述第二加密引擎包括用于使用所述第一计数器值和所述第一密码生成所述第二密钥流的ECM模块。

17.根据权利要求15所述的方法,其中所述第一加密引擎使用所述第一密钥流加密第一数据包以生成第一加密数据包,并且将所述第一加密数据包与所述第一计数器的当前计数器值一起传输到所述第二加密引擎。

18.根据权利要求17所述的方法,其中所述第二加密引擎使用所述第一计数器的所述当前计数器值和所述密码生成所述第一密钥流,并且使用所述第一密钥流来解密所述加密数据包。

19.根据权利要求17所述的方法,其中加密所述第一数据包包括将所述第一密钥流与所述第一数据包进行异或运算以生成所述第一加密数据包。

20.根据权利要求18所述的方法,其中所述IoT装置包括蓝牙低功耗(BTLE)通信接口,以用于将所述IoT装置通信地耦接到所述IoT中心或所述移动用户装置,所述IoT中心或所述移动用户装置通过互联网通信地耦接到所述IoT服务。

21.一种设备,包括:

第一本地无线通信模块,所述第一本地无线通信模块用于建立与第二本地无线通信模块的本地无线通信信道;

由所述第一本地无线通信模块管理的数据结构,所述数据结构包括一组特征,每个特征用特征ID标识并且具有与其相关联的包含与所述特征相关联的数据的值缓冲区;

所述第一无线通信模块使用所述数据结构通过定义用于接收由所述第二本地无线通信模块传输的数据包的第一特征和用于向所述第二本地无线通信模块传输传出数据包的第二特征来形成第一网络套接字;

其中传入数据包将临时地在与所述第一特征相关联的第一值缓冲区中缓冲,并且其中所述传出数据包将临时地在与所述第二特征相关联的第二值缓冲区中缓冲。

22. 根据权利要求21所述的设备,其中所述数据结构包括由特征ID索引并且包括用于标识每个值缓冲区的值字段的表,所述值缓冲区包括与所述第一特征相关联的所述第一值缓冲区和与所述第二特征相关联的所述第二值缓冲区。

23. 根据权利要求21所述的设备,其中所述第一值缓冲区和所述第二值缓冲区各自具有N位的大小。

24. 根据权利要求23所述的设备,其中 $N=20$ 。

25. 根据权利要求21所述的设备,其中所述无线通信模块包括蓝牙低功耗 (BTLE) 通信模块,并且其中所述数据结构包括通用属性配置文件 (GATT) 数据结构。

26. 根据权利要求23所述的设备,还包括应用程序逻辑,以用于通过将第一数据包一次N位地写入与所述第二特征相关联的所述第二值缓冲区来将所述第一数据包传输到所述第二无线通信模块。

27. 根据权利要求26所述的设备,其中所述第二无线通信模块将从与所述第二特征相关联的所述第二值缓冲区一次N位地读取所述第一数据包。

28. 根据权利要求27所述的设备,其中为了读取所述第一数据包,所述第二无线通信模块将向所述第一无线通信模块传输包括所述第二特征的特征ID的第一命令。

29. 根据权利要求26所述的设备,其中所述应用程序逻辑将通过从所述第一值缓冲区一次N位地读取第二数据包来从所述第二无线通信模块接收所述第二数据包。

30. 根据权利要求29所述的设备,其中为了将所述第二数据包写入所述第一值缓冲区,所述第二无线通信模块将向所述第一无线通信模块传输包括所述第一特征的特征ID的第一命令。

用于在物联网 (IoT) 系统中建立安全通信信道的设备和方法

背景技术

技术领域

[0001] 本发明整体涉及计算机系统领域。更具体地讲,本发明涉及用于在IoT系统中建立安全通信信道的设备和方法。

[0002] 相关技术描述

[0003] “物联网”是指互联网基础架构内可唯一识别的嵌入式装置的互连。最终,IoT预计将导致新的种类广泛的应用程序,在这些应用程序中,几乎任何类型的物理事物可以提供关于其本身或其周围环境的信息和/或可以通过互联网上的客户端装置受到远程控制。

[0004] 由于与连接性、功率和标准化缺乏有关的一些问题,物联网的开发和采用一直比较缓慢。例如,IoT开发和采用所面临的一个障碍是,没有标准平台允许开发人员设计和提供新的IoT装置及服务。为了进入IoT市场,开发人员必须从头开始设计整个IoT平台,包括支持所需的IoT实现需要的网络协议和基础架构、硬件、软件和服务。因此,IoT装置的每一家提供商都使用专有技术来设计和连接IoT装置,这使得对最终用户而言,采用多种类型的IoT装置是一项繁重的工作。IoT采用所面临的另一个障碍是与IoT装置的连接和供电有关的困难。例如,连接诸如冰箱、车库门开关、环境传感器、家用安防传感器/控制器等的电器需要电源来为每个连接的IoT装置供电,并且这个电源通常定位不太方便。

[0005] 存在的另一个问题是用于互连IoT装置的无线技术(诸如蓝牙LE)通常是短程技术。所以,如果用于实施IoT的数据收集中心处于IoT装置的范围之外,则IoT装置将无法将数据传输到IoT中心(反之亦然)。因此,需要允许IoT装置将数据提供到范围之外的IoT中心(或其他IoT装置)的技术。

附图说明

[0006] 可结合下列附图从以下具体实施方式更好地理解本发明,其中:

[0007] 图1A至图1B示出了IoT系统架构的不同实施方案;

[0008] 图2示出了根据本发明的一个实施方案的IoT装置;

[0009] 图3示出了根据本发明的一个实施方案的IoT中心;

[0010] 图4A至图4B示出了本发明用于控制和收集来自IoT装置的数据并且生成通知的实施方案;

[0011] 图5示出了本发明用于收集来自IoT装置的数据并且生成来自IoT中心和/或IoT服务的通知的实施方案;

[0012] 图6示出了中间移动装置收集来自固定IoT装置的数据并将该数据提供给IoT中心的系统的一个实施方案;

[0013] 图7示出了本发明的一个实施方案中实施的中间连接逻辑;

[0014] 图8示出根据本发明的一个实施方案的方法;

[0015] 图9A示出了向IoT装置提供程序代码和数据更新的实施方案;

- [0016] 图9B示出了向IoT装置提供程序代码和数据更新的方法的实施方案；
- [0017] 图10示出了安全架构的一个实施方案的高级视图；
- [0018] 图11示出了其中使用用户身份模块 (SIM) 来在IoT装置上存储密钥的架构的一个实施方案；
- [0019] 图12A示出了其中使用条形码或QR代码来注册IoT装置的一个实施方案；
- [0020] 图12B示出了其中使用条形码或QR代码进行配对的一个实施方案；
- [0021] 图13示出了用于使用IoT中心对SIM进行编程的方法的一个实施方案；
- [0022] 图14示出了用于向IoT中心和IoT服务注册IoT装置的方法的一个实施方案；并且
- [0023] 图15示出了用于加密要发送给IoT装置的数据的方法的一个实施方案；
- [0024] 图16A至图16B示出了本发明用于在IoT服务和IoT装置之间加密数据的不同实施方案；
- [0025] 图17示出了本发明用于执行安全密钥交换、生成公共密码、并使用该密码生成密钥流的实施方案；
- [0026] 图18示出根据本发明的一个实施方案的数据包结构；
- [0027] 图19示出了在一个实施方案中所采用的在不与IoT装置正式配对的情况下向IoT装置写入数据/从IoT装置读取数据的技术；
- [0028] 图20示出了在本发明的一个实施方案中采用的一组示例性命令包；
- [0029] 图21示出了使用命令包的交易的示例性序列；
- [0030] 图22示出根据本发明的一个实施方案的方法；并且
- [0031] 图23A至图23C示出了根据本发明的一个实施方案的用于安全配对的方法。

具体实施方式

[0032] 在下面的描述中,出于解释的目的,本文陈述了许多特定细节以便透彻理解下文描述的本发明的实施方案。然而,本领域的技术人员将容易明白,可在没有这些特定细节中的一些的情况下实践本发明的实施方案。在其他情况下,为避免模糊本发明的实施方案的基本原理,已熟知的结构和装置以框图形式示出。

[0033] 本发明的一个实施方案包括一种物联网 (IoT) 平台,开发人员可以利用该平台来设计和构建新的IoT装置及应用程序。具体地讲,一个实施方案包括用于物联网装置的基础硬件/软件平台,所述基础硬件/软件平台包括预定义的网络协议栈和IoT中心,IoT装置通过该IoT中心连接到互联网。此外,一个实施方案包括一项IoT服务,可以通过该IoT服务如下所述地访问和管理IoT中心和连接的IoT装置。此外,IoT平台的一个实施方案包括用以访问和配置IoT服务、中心和连接的装置的IoT应用程序或Web应用程序(例如,在客户端装置上执行)。现有的在线零售商和其他网站运营商可以利用本文所述的IoT平台容易地为现有用户群提供独特的IoT功能。

[0034] 图1A示出了其上可以实现本发明的实施方案的架构平台的概览。具体地讲,图示实施方案包括多个IoT装置101至105,这些IoT装置通过本地通信信道130通信地连接到中央IoT中心110,该中央IoT中心本身通过互联网220通信地连接到IoT服务120。IoT装置101至105中的每个IoT装置最初可以(例如,使用下文描述的配对技术)与IoT中心110配对以使得能够实现本地通信信道130中的每个通信信道。在一个实施方案中,IoT服务120包括最终用

户数据库122,以用于维护从每个用户的IoT装置收集的用户账户信息和数据。例如,如果IoT装置包括传感器(例如,温度传感器、加速度计、热传感器、运动检测器等),则数据库122可以不断更新以存储由IoT装置101-105收集的数据。存储在数据库122中的数据然后可以由安装在用户装置135上的IoT应用程序和浏览器(或者经由台式计算机或其他客户端计算机系统)使最终用户和网络客户端(例如,订阅了IoT服务120的网站130)能够访问到。

[0035] IoT装置101至105可以配备有各种类型的传感器以收集关于自身及其周围环境的信息,并且经由IoT中心110将收集的信息提供给IoT服务120、用户装置135和/或外部网站130。IoT装置101至105中的一些IoT装置可以响应于通过IoT中心110发送的控制命令来执行指定的功能。下文提供由IoT装置101至105收集的信息以及控制命令的多种具体示例。在下文描述的一个实施方案中,IoT装置101是被设计为记录用户选择并将用户选择发送到IoT服务120和/或网站的用户输入装置。

[0036] 在一个实施方案中,IoT中心110包括蜂窝无线电装置,以经由诸如4G(例如,移动WiMAX、LTE)或5G蜂窝数据服务的蜂窝服务115建立到互联网220的连接。作为另外一种选择或除此之外,IoT中心110可以包括WiFi无线电装置,以通过将IoT中心110连接到互联网(例如,经由向最终用户提供互联网服务的互联网服务提供商)的WiFi接入点或路由器116建立WiFi连接。当然,应当注意,本发明的基本原理不限于任何特定类型的通信信道或协议。

[0037] 在一个实施方案中,IoT装置101至105是能够使用电池电量长时间(例如,数年)运行的超低功率装置。为了节省功率,可以使用低功率无线通信技术诸如蓝牙低功耗(LE)来实现本地通信信道130。在该实施方案中,IoT装置101至105中的每个IoT装置和IoT中心110配备有蓝牙LE无线电装置和协议栈。

[0038] 如上所述,在一个实施方案中,该IoT平台包括在用户装置135上执行以允许用户访问和配置所连接的IoT装置101至105、IoT中心110和/或IoT服务120的IoT应用程序或Web应用程序。在一个实施方案中,该应用程序或Web应用程序可以由网站130的运营者设计为向其用户群提供IoT功能。如图所示,该网站可以维护包含与每个用户相关的账户记录的用户数据库131。

[0039] 图1B示出了用于多个IoT中心110至111、190的附加连接选项。在该实施方案中,单个用户可以在单个用户驻地180(例如,用户的家或工作地点)处现场安装有多个中心110至111。可以这样做,以例如扩展连接所有IoT装置101至105所需的无线范围。如图所示,如果用户具有多个中心110、111,则它们可以经由本地通信信道(例如,Wifi、以太网、电力线网络等)连接。在一个实施方案中,中心110至111中的每个中心可以通过蜂窝连接115或WiFi连接116(图1B中未明确示出)来建立与IoT服务120的直接连接。作为另外一种选择或除此之外,IoT中心中的一个IoT中心诸如IoT中心110可以充当“主”中心,其向用户驻地180上的所有其他IoT中心诸如IoT中心111提供连接性和/或本地服务(如连接IoT中心110和IoT中心111的虚线所示)。例如,主IoT中心110可以是建立与IoT服务120的直接连接的唯一IoT中心。在一个实施方案中,只有“主”IoT中心110配备有蜂窝通信接口以建立与IoT服务120的连接。这样,IoT服务120和其他IoT中心111之间的所有通信将流经主IoT中心110。作为这个角色,主IoT中心110可以具有附加的程序代码,以对在其他IoT中心111和IoT服务120之间交换的数据执行过滤操作(例如,在可能时,本地服务于一些数据请求)。

[0040] 不管IoT中心110至111如何连接,在一个实施方案中,IoT服务120将逻辑地将中心

与用户相关联,并且将所有附接的IoT装置101至105组合在能经由已安装有应用程序135的用户装置访问的单个综合性用户界面(和/或基于浏览器的界面)下。

[0041] 在该实施方案中,主IoT中心110和一个或多个从属IoT中心111可以通过本地网络连接,该本地网络可以是WiFi网络116、以太网和/或使用电力线通信(PLC)网络(例如,其中网络的全部或部分通过用户的电力线运行)。另外,对于IoT中心110至111,IoT装置101至105中的每个IoT装置可以使用诸如WiFi、以太网、PLC或蓝牙LE的任何类型的本地网络信道与IoT中心110至111互连。

[0042] 图1B还示出了安装在第二用户驻地181处的IoT中心190。几乎无限数量的此类IoT中心190可以被安装和配置,以收集来自世界各地的用户驻地IoT装置191至192的数据。在一个实施方案中,可以为同一用户配置两个用户驻地180至181。例如,一个用户驻地180可以是用户的主要住宅,而另一个用户驻地181可以是用户的度假屋。在这种情况下,IoT服务120将逻辑地将IoT中心110至111、190与用户相关联,并将所有附接的IoT装置101至105、191至192组合在能经由已安装有应用程序135的用户装置访问的单个综合性用户界面(和/或基于浏览器的界面)下。

[0043] 如图2所示,IoT装置101的一个示例性实施方案包括用于存储程序代码和数据201至203的存储器210、以及用于执行程序代码和处理数据的低功率微控制器200。存储器210可以是诸如动态随机存取存储器(DRAM)的易失性存储器,或者可以是诸如闪存存储器的非易失性存储器。在一个实施方案中,非易失性存储器可用于永久存储,而易失性存储器可用于在运行时执行程序代码和数据。此外,存储器210可以集成在低功率微控制器200内,或者可以经由总线或通信结构连接到低功率微控制器200。本发明的基本原理不限于存储器210的任何特定实现方式。

[0044] 如图所示,所述程序代码可以包括定义将由IoT装置201执行的一组特定于应用程序的功能组的应用程序代码203、以及包括可由IoT装置101的应用程序开发人员利用的一组预定义构建块的库代码202。在一个实施方案中,库代码202包括实现IoT装置所需的一组基本功能,诸如用于使得能够实现每个IoT装置101和IoT中心110之间的通信的通信协议栈201。如上所述,在一个实施方案中,通信协议栈201包括蓝牙LE协议栈。在该实施方案中,蓝牙LE无线电装置和天线207可以被集成在低功率微控制器200内。然而,本发明的基本原理不限于任何特定的通信协议。

[0045] 图2中所示的具体实施方案还包括用以接收用户输入并向低功率微控制器提供该用户输入的多个输入装置或传感器210,所述低功率微控制器根据应用程序代码203和库代码202处理该用户输入。在一个实施方案中,输入装置中的每个输入装置包括用于向最终用户提供反馈的LED 209。

[0046] 另外,图示实施方案包括用于向低功率微控制器供电的电池208。在一个实施方案中,使用不可充电的纽扣电池。然而,在一个另选的实施方案中,可以使用集成的可充电电池(例如,通过将IoT装置连接到交流电源(未示出)来充电)。

[0047] 还提供用于产生音频的扬声器205。在一个实施方案中,低功率微控制器299包括用于解码压缩音频流(例如,诸如MPEG-4/高级音频编码(AAC)流)以在扬声器205上产生音频的音频解码逻辑。作为另外一种选择,低功率微控制器200和/或应用程序代码/数据203可以包括数字采样的音频片段,以在用户经由输入装置210输入选择时向最终用户提供言

语反馈。

[0048] 在一个实施方案中,基于IoT装置101针对其设计的特定应用程序,可以在IoT装置101上包括一个或多个其他/替代的I/O装置或传感器250。例如,可以包括环境传感器以测量温度、压力、湿度等。如果将IoT装置用作安防装置,则可以包括安防传感器和/或门锁开启器。当然,这些示例仅仅是为了举例说明的目的而提供的。本发明的基本原理不限于任何特定类型的IoT装置。事实上,考虑到配备有库代码202的低功率微控制器200的高可编程性质,应用程序开发人员可以容易地开发新的应用程序代码203和新的I/O装置250以对于几乎任何类型的IoT应用程序与低功率微控制器衔接。

[0049] 在一个实施方案中,低功率微控制器200还包括安全密钥存储装置,以用于存储用于加密通信和/或生成签名的加密密钥。作为另外一种选择,密钥可以被保护在用户识别模块(SIM)中。

[0050] 在一个实施方案中,包括唤醒接收器207以将IoT装置从几乎不消耗功率的超低功率状态中唤醒。在一个实施方案中,唤醒接收器207被配置为响应于从如图3所示的配置在IoT中心110上的唤醒发射器307接收到的唤醒信号来使IoT装置101退出该低功率状态。具体地讲,在一个实施方案中,发射器307和接收器207一起形成电谐振变压器电路,诸如特斯拉线圈。在操作中,当中心110需要将IoT装置101从极低功率状态唤醒时,能量经由射频信号从发射器307发送到接收器207。由于此能量传递,IoT装置101可以被配置成在处于低功率状态时几乎不消耗功率,因为它不需要连续地“监听”来自中心的信号(如使用允许装置通过网络信号被唤醒的网络协议的情况那样)。更确切地说,IoT装置101的微控制器200可以被配置为通过使用从发射器307以电学方式发送到接收器207的能量而在被有效地掉电之后被唤醒。

[0051] 如图3所示,IoT中心110还包括用于存储程序代码和数据305的存储器317、以及用于执行程序代码和处理数据的硬件逻辑301诸如微控制器。广域网(WAN)接口302和天线310将IoT中心110连接到蜂窝服务115。作为另外一种选择,如上所述,IoT中心110还可以包括本地网络接口(未示出),诸如WiFi接口(和WiFi天线)或以太网接口,用于建立局域网通信信道。在一个实施方案中,硬件逻辑301还包括安全密钥存储装置,以用于存储用于加密通信和生成/验证签名的加密密钥。作为另外一种选择,密钥可以被保护在用户识别模块(SIM)中。

[0052] 本地通信接口303和天线311建立与IoT装置101至105中的每个IoT装置的本地通信信道。如上所述,在一个实施方案中,本地通信接口303/天线311实施蓝牙LE标准。然而,本发明的基本原理不限于用于建立与IoT装置101至105的本地通信信道的任何具体协议。虽然在图3中被示出为单独的单元,但WAN接口302和/或本地通信接口303可以嵌入在与硬件逻辑301相同的芯片内。

[0053] 在一个实施方案中,程序代码和数据包括通信协议栈308,该协议栈可以包括分开的堆栈来用于通过本地通信接口303和WAN接口302进行通信。此外,可以将装置配对程序代码和数据306存储在存储器中以允许IoT中心与新的IoT装置配对。在一个实施方案中,为每个新的IoT装置101至105分配唯一的代码,在配对过程中将该唯一代码传送到IoT中心110。例如,该唯一代码可以嵌入在IoT装置上的条形码中,并且可以由条形码读取器106读取,或者可以通过本地通信信道130传送。在一个另选的实施方案中,该唯一ID代码被磁性地嵌入

在IoT装置上,并且IoT中心具有磁传感器诸如射频ID(RFID)或近场通信(NFC)传感器,以在IoT装置101在距离IoT中心110几英寸内移动时检测该代码。

[0054] 在一个实施方案中,一旦已经传送该唯一ID,IoT中心110就可以通过以下方式来验证该唯一ID:查询本地数据库(未示出)、执行散列来验证该代码是否可接受、并且/或者与IoT服务120、用户装置135和/或网站130通信来校验该ID代码。在一个实施方案中,一旦经校验,IoT中心110就和IoT装置101配对并且将配对数据存储在存储器317中(如上所述,该存储器可以包括非易失性存储器)。一旦配对完成,IoT中心110就可以与IoT装置101连接以执行本文所述的各种IoT功能。

[0055] 在一个实施方案中,运行IoT服务120的组织可以提供IoT中心110和基本硬件/软件平台,以允许开发人员容易地设计新的IoT服务。具体地讲,除了IoT中心110之外,还可以为开发人员提供软件开发工具包(SDK)来更新在中心110内执行的程序代码和数据305。另外,对于IoT装置101,SDK可以包括针对基础IoT硬件(例如,图2所示的低功率微控制器200和其他组件)设计的一组广泛的库代码202以有利于设计各种不同类型的应用程序101。在一个实施方案中,SDK包括开发人员仅需要在其中为IoT装置指定输入和输出的图形设计界面。已经为开发人员准备好了所有联网代码,包括允许IoT装置101连接到中心110和服务120的通信栈201。此外,在一个实施方案中,SDK还包括用于促进为移动装置(例如,iPhone和Android装置)设计应用程序的库代码基。

[0056] 在一个实施方案中,IoT中心110管理IoT装置101至105和IoT服务120之间的连续双向数据流。在需要实时更新到IoT装置101至105/从这些IoT装置更新的情况下(例如,在用户需要查看安防装置或环境读数的当前状态的情况下),IoT中心可以保持打开的TCP套接字以提供对用户装置135和/或外部网站130的定期更新。用于提供更新的具体联网协议可以根据底层应用程序的需求进行调整。例如,在某些情况下,如果连续双向流可能没有意义,则可以使用简单的请求/响应协议来在需要时收集信息。

[0057] 在一个实施方案中,IoT中心110和IoT装置101至105都可以通过网络自动升级。具体地讲,当IoT中心110有新的更新可用时,它可以自动地从IoT服务120下载并安装此更新。它可以首先将更新的代码复制到本地存储器中,运行并验证此更新,然后替换较旧的程序代码。类似地,当IoT装置101至105中的每一个有更新可用时,最初可以由IoT中心110下载这些更新并将其推送到IoT装置101至105中的每个装置。然后,每个IoT装置101至105可以采用类似于上述用于IoT中心的方式应用更新,并将更新的结果报告回IoT中心110。如果更新成功,则IoT中心110可以从其存储器中删除此更新,并记录安装在每个IoT装置上的最新的代码版本(例如,使得它可以继续检查每个IoT装置是否有新的更新)。

[0058] 在一个实施方案中,IoT中心110通过交流电源供电。具体地讲,IoT中心110可以包括具有变压器的供电单元390,其中变压器用于将通过交流电源线提供的交流电压转换为较低的直流电压。

[0059] 图4A示出了本发明使用IoT系统执行通用远程控制操作的一个实施方案。具体地讲,在该实施方案中,一组IoT装置101-103分别配备有红外(IR)和/或射频(RF)发射器401-403,用于发送远程控制代码以控制各种不同类型的电子设备,包括空调/加热器430、照明系统431和视听设备432(仅举几例)。在图4A所示的实施方案中,IoT装置101-103也分别配备有传感器404-406,以用于检测它们控制的装置的操作,如下所述。

[0060] 例如, IoT装置101中的传感器404可以是用于感测当前温度/湿度的温度和/或湿度传感器, 并且响应地基于当前所需的温度来控制空调/加热器430。在该实施方案中, 空调/加热器430是被设计为经由远程控制装置(通常是其本身具有嵌入其中的温度传感器的遥控器)来控制的空调/加热器。在一个实施方案中, 用户经由安装在用户装置135上的应用程序和浏览器向IoT中心110提供所需的温度。在IoT中心110上执行的控制逻辑412接收来自传感器404的当前温度/湿度数据, 并且响应地向IoT装置101发送命令, 以根据所需的温度/湿度来控制IR/Rf发射器401。例如, 如果温度低于所需温度, 则控制逻辑412可以经由IR/Rf发射器401向空调/加热器发送命令以升高温度(例如, 通过关闭空调或者打开加热器)。该命令可以包括存储在IoT中心110上的数据库413中的必要远程控制代码。作为另外一种选择或除此之外, IoT服务421可以实施控制逻辑421, 以基于指定的用户偏好和所存储的控制代码422来控制电子设备430-432。

[0061] 所示示例中的IoT装置102被用于控制照明431。具体而言, IoT装置102中的传感器405可以是光电传感器或光电检测器, 其被配置为检测由灯具431(或其他照明设备)产生的光的当前亮度。用户可以经由用户装置135向IoT中心110指定所需的照明水平(包括开或关的指示)。作为响应, 控制逻辑412将向IR/Rf发射器402发送命令以控制灯431的当前亮度水平(例如, 如果当前亮度太低则增加照明, 或者如果当前亮度太高则减少照明; 或者只是打开灯或关闭灯)。

[0062] 所示示例中的IoT装置103被配置为控制视听设备432(例如, 电视机、A/V接收器、有线/卫星接收器、AppleTV™等等)。IoT装置103中的传感器406可以是用于检测当前环境音量水平的音频传感器(例如, 麦克风和相关联的逻辑)和/或基于电视机产生的光来检测电视机的开启关闭情况的光电传感器(例如, 通过测量特定光谱内的光)。作为另外一种选择, 传感器406可包括连接到视听设备的温度传感器, 以基于所检测的温度来检测音频设备的开启关闭情况。再次, 响应于经由用户装置135的用户输入, 控制逻辑412可以经由IoT装置103的IR发射器403向视听设备发送命令。

[0063] 应该指出的是, 以上仅是本发明的一个实施方案的例示性示例。本发明的基本原理不限于由IoT装置控制的任何特定类型的传感器或设备。

[0064] 在IoT装置101-103经由蓝牙LE连接而耦接到IoT中心110的实施方案中, 传感器数据和命令通过蓝牙LE信道发送。然而, 本发明的基本原理不限于蓝牙LE或任何其他通信标准。

[0065] 在一个实施方案中, 控制每个电子设备所需的控制代码被存储在IoT中心110上的数据库413中和/或IoT服务120上的数据库422中。如图4B所示, 控制代码可以从用于IoT服务120上所维护的不同设备的控制代码422的主数据库被提供给IoT中心110。最终用户可以指定要通过在用户装置135上执行的应用程序和浏览器来控制的电子(或其他)设备的类型, 并且作为响应, IoT中心上的远程控制代码学习模块491可以检索来自IoT服务120上的远程控制代码数据库492的所需IR/Rf代码(例如, 用唯一的ID来识别每个电子设备)。

[0066] 另外, 在一个实施方案中, IoT中心110配备有IR/Rf接口490, 以允许远程控制代码学习模块491直接从与电子设备一起提供的原始远程控制器495“学习”新的远程控制代码。例如, 如果远程控制数据库中不包括用于与空调430一起提供的原始远程控制器的控制代码, 则用户可以经由用户装置135上的应用程序/浏览器与IoT中心110进行交互, 以教导IoT

中心110由原始远程控制器产生的各种控制代码(例如,增加温度、降低温度等)。一旦获知远程控制代码,它们就可以存储在IoT中心110上的控制代码数据库413中,并且/或者被发送回IoT服务120,以便被包括在中央远程控制代码数据库492中(并且随后被具有相同空调单元430的其他用户使用)。

[0067] 在一个实施方案中,每个IoT装置101-103具有非常小的外形,并且可以使用双面胶带、小钉子、磁性附件等附接在它们各自的电子设备430-432上或附近。为了控制某件设备(诸如空调430),需将IoT装置101放置得足够远以使得传感器404能够准确地测量家中的环境温度(例如,若将IoT装置直接放置在空调上,则当空调运行时温度测量值会太低,而当加热器运行时温度测量值会太高)。相比之下,用于控制照明的IoT装置102可以放置在传感器405的照明器具431上或其附近,以检测当前照明水平。

[0068] 除了提供所述总体控制功能之外,IoT中心110和/或IoT服务120的一个实施方案向最终用户发送与每个电子设备的当前状态相关的通知。该通知可以是文本消息和/或特定于应用程序的通知,其继而可以被显示在用户的移动装置135的显示器上。例如,如果用户的空调已经打开了一段较长时间但是温度没有发生变化,则IoT中心110和/或IoT服务120可以向用户发送空调未正常工作的通知。如果用户不在家(这可以通过运动传感器来检测,或基于当前检测到的用户位置来检测),并且传感器406指示视听设备430开启或者传感器405指示灯开启,则可向用户发送通知,询问用户是否想要关闭视听设备432和/或灯431。针对任何设备类型都可以发送相同类型的通知。

[0069] 一旦用户接收到通知,他/她就可以通过用户装置135上的应用程序或浏览器来远程地控制电子设备430-432。在一个实施方案中,用户装置135是触摸屏装置,并且应用程序和浏览器显示出带有用于控制设备430-432的用户可选按钮的远程控制器图像。用户收到通知后,可以打开图形远程控制器,关闭或调整各种不同的设备。如果通过IoT服务120连接,则用户的选择可以从IoT服务120转发到IoT中心110,然后将通过控制逻辑412来控制设备。作为另外一种选择,用户输入可以从用户装置135直接发送到IoT中心110。

[0070] 在一个实施方案中,用户可以对IoT中心110上的控制逻辑412进行编程,以对电子设备430-432执行各种自动控制功能。除了如上所述保持所需的温度、亮度水平和音量水平之外,如果检测到某些条件,则控制逻辑412可以自动关闭电子设备。例如,如果控制逻辑412检测到用户不在家并且空调未工作,则可以自动关闭空调。类似地,如果用户不在家,并且传感器406指示视听设备430开启或者传感器405指示灯开启,则控制逻辑412可以经由IR/RF发射器403和402分别自动发送命令,以关闭视听设备和灯。

[0071] 图5示出了配备有用于监视电子设备530-531的传感器503-504的IoT装置104-105的另外的实施方案。具体地讲,本实施方案的IoT装置104包括温度传感器503,其可以放置在炉灶530上或炉灶附近,以检测炉灶何时仍然开着。在一个实施方案中,IoT装置104将由温度传感器503测量的当前温度发送到IoT中心110和/或IoT服务120。如果检测到炉灶打开超过阈值时间段(例如,基于所测量的温度),则控制逻辑512可以向最终用户的装置135发送通知,告知用户炉530处于开启状态。另外,在一个实施方案中,IoT装置104可以包括控制模块501,以响应于接收到来自用户的指令而关闭炉灶或自动地(如果控制逻辑512被用户编程为这样做)关闭炉灶。在一个实施方案中,控制逻辑501包括切断炉灶530的电力或供气的开关。然而,在其他实施方案中,控制逻辑501可被集成在炉灶内。

[0072] 图5还示出了具有运动传感器504的IoT装置105,以用于检测某些类型的电子设备(诸如,洗衣机和/或烘干机)的运动。可用到的另一种传感器是用于检测环境音量水平的音频传感器(例如,麦克风及逻辑)。如同上述其他实施方案一样,如果满足某些指定的条件(例如,如果检测到运动进行了一段较长时间,指示洗衣机/烘干机未关闭),则该实施方案可以向最终用户发送通知。虽然未在图5中示出,IoT装置105还可以配备有控制模块,以自动地和/或响应于用户输入来关闭洗衣机/烘干机531(例如,通过关断电/气)。

[0073] 在一个实施方案中,具有控制逻辑和开关的第一IoT装置可以被配置为关闭用户家中的所有电力,而具有控制逻辑和开关的第二IoT装置可以被配置为关闭用户家中的所有供气。然后可以将具有传感器的IoT装置设置在用户家中的供电设备或供气设备之上或附近。如果用户被通知存在特定的设备(例如,炉灶530)未关断,则用户随后可以发送关断家中所有电力或供气的命令,以防止造成损坏。作为另外一种选择,IoT中心110和/或IoT服务120中的控制逻辑512可以被配置为在这种情况下自动关断电力或供气。

[0074] 在一个实施方案中,IoT中心110和IoT服务120以周期性的间隔进行通信。如果IoT服务120检测到与IoT中心110的连接已丢失(例如,在指定的持续时间内未能从IoT中心接收到请求或响应),则其将该信息发送给最终用户的设备135(例如通过发送文本消息或特定于应用程序的通知)。

[0075] 用于通过中间装置传送数据的设备和方法

[0076] 如上所述,因为用于互连IoT装置的无线技术(诸如蓝牙LE)通常是短程技术,所以如果用于实施IoT的中心处于IoT装置的范围之外,则IoT装置将无法将数据传输到IoT中心(反之亦然)。

[0077] 为了克服这项缺陷,本发明的一个实施方案提供了一种用于IoT装置的机构,其处于IoT中心的无线范围之外,以便当一个或多个移动装置在范围内时周期性地与所述一个或多个移动装置连接。一旦建立连接,IoT装置就可以将需要提供给IoT中心的任何数据传输到移动装置,移动装置然后将数据转发到IoT中心。

[0078] 如图6所示,一个实施方案包括IoT中心110、超出IoT中心110的范围的IoT装置601,和移动装置611。超出范围的IoT装置601可以包括能够收集和传送数据的任何形式的IoT装置。例如,IoT装置601可以包括配置在冰箱内的数据收集装置,以监视冰箱中可食用的食品、食用这些食品的用户以及当前温度。当然,本发明的基本原理不限于任何特定类型的IoT装置。本文所述的技术可以使用任何类型的IoT装置来实施,包括那些用于收集和传输用于以下家电的数据的装置:智能仪表、炉灶、洗衣机、烘干机、照明系统、HVAC系统和视听设备(仅举几例)。

[0079] 此外,在操作中的移动装置、图6所示的IoT装置611可以是能够传送数据和存储数据的任何形式的移动装置。例如,在一个实施方案中,移动装置611是其上安装有应用程序的智能手机,以有利于本文所述的技术。在另一个实施方案中,移动装置611包括可穿戴设备,诸如贴在项链或手镯上的通信令牌、智能手表或健身装置。可穿戴令牌对于老年用户或未拥有智能手机装置的其他用户可能特别有用。

[0080] 在操作中,超出范围的IoT装置601可以周期性地或连续地检查与移动装置611的连通性。在建立起连接时(例如,作为用户在冰箱附近移动的结果),IoT装置601上所收集的任何数据605被自动发送到移动装置611上的临时数据储存库615。在一个实施方案中,IoT

装置601和移动装置611使用低功率无线标准(诸如BTLE)来建立本地无线通信信道。在这样的情况下,最初可以使用已知的配对技术将移动装置611与IoT装置601配对。

[0081] 一旦已将数据传输到临时数据储存库,移动装置611就将在与IoT中心110建立起通信时(例如,当用户在IoT中心110的范围内走动时)传输数据。IoT中心然后可以将数据存储在中央数据储存库413中,并且/或者通过因特网将数据发送到一个或多个服务和/或其他用户装置。在一个实施方案中,移动装置611可以使用不同类型的通信信道来将数据提供给IoT中心110(可能是更高功率的通信信道,诸如WiFi)。

[0082] 超出范围的IoT装置601、移动装置611和IoT中心都可以用程序代码和/或逻辑来配置,以实施本文所述的技术。如图7所示,例如,IoT装置601可以被配置有中间连接逻辑和/或应用程序,移动装置611可以被配置有中间连接逻辑/应用程序,并且IoT中心110可以被配置有中间连接逻辑/应用程序721,以执行本文所述的操作。每个装置上的中间连接逻辑/应用程序可以用硬件、软件或其任何组合来实现。在一个实施方案中,IoT装置601的中间连接逻辑/应用程序701搜索并建立与移动装置上的中间连接逻辑/应用程序711(其可以被实施为装置应用程序)的连接,以将数据传输到临时数据储存库615。移动装置611上的中间连接逻辑/应用程序701然后将数据转发到IoT中心上的中间连接逻辑/应用程序,后者将数据存储在中央数据储存库413中。

[0083] 如图7所示,每个装置上的中间连接逻辑/应用程序701、711、721可以基于当前应用程序来配置。例如,对于冰箱而言,连接逻辑/应用程序701可能仅需要周期性地发送几个数据包。对于其他应用程序(例如,温度传感器),连接逻辑/应用程序701可能需要传送更频繁的更新。

[0084] 在一个实施方案中,IoT装置601(而非移动装置611)可以被配置为建立与位于IoT中心110的范围内的一个或多个中间IoT装置的无线连接。在该实施方案中,IoT中心范围之外的任何IoT装置601可以通过使用其他IoT装置形成“链”而链接到该中心。

[0085] 另外,尽管为了简单起见,图6至图7仅示出了单个移动装置611,然而在一个实施方案中,不同用户的多个此类移动装置可以被配置为与IoT装置601进行通信。而且,可以对多个其他IoT装置实施相同的技术,从而在整个家中形成中间装置数据收集系统。

[0086] 此外,在一个实施方案中,本文所述的技术可以用于收集各种不同类型的相关数据。例如,在一个实施方案中,每当移动装置611与IoT装置601连接时,所收集的数据605就可包括用户的身份。这样一来,IoT系统可以被用来跟踪家中不同用户的行为。例如,如果在冰箱内使用该系统,则所收集的数据605可包括经过冰箱的每个用户的身份、打开冰箱的每个用户的身份,以及每个用户所食用的特定食物。可以从其他类型的IoT装置收集不同类型的数据。通过使用该数据,系统能够确定例如哪个用户洗了衣服、哪个用户在给定的日子看了电视、每个用户入眠和醒来的时间等等。所有这些来源于人群的数据然后可以在IoT中心的数据储存库413内得到编译和/或被转发到外部服务或用户。

[0087] 本文所述的技术的另一有益应用程序为用于监视可能需要帮助的老年用户。对于该应用程序,移动装置611可以是由老年用户佩戴的非常小的令牌,以收集用户家中不同房间里的信息。例如,每当用户打开冰箱时,该数据将与所收集的数据605会一起得以包括,并且经由令牌传输到IoT中心110。然后,IoT中心可以将数据提供给一个或多个外部用户(例如,照顾老年用户的儿童或其他个人)。如果在指定的时间段内(例如12小时)没有收集到数

据,那么这意味着老年用户没有在家中移动和/或没有打开冰箱。然后,IoT中心110或连接到IoT中心的外部服务可以向这些其他个人发送警报通知,通知他们应该查看老年用户的情况。另外,所收集的数据605可以包括其他相关信息,诸如用户正在食用的食物、以及是否需要去一趟杂货店、老年用户是否正在看电视以及老年用户看电视的频率、老年人用户洗衣服的频率等。

[0088] 在另一个具体实施中,如果洗衣机、冰箱、HVAC系统等电子设备存在问题,则所收集的数据可以包括对需要更换的部件的指示。在这种情况下,可能会向技术人员发送通知,请求解决问题。技术人员然后带着所需的更换部件去往这户人家。

[0089] 图8中示出根据本发明的一个实施方案的方法。该方法可在上述架构的环境内实施,但不限于任何具体架构。

[0090] 在801处,在超出IoT中心的范围的IoT装置处周期性地收集数据(例如,打开冰箱门、食用的食品等)。在802处,IoT装置周期性或连续地检查与移动装置的连通性(例如,使用标准本地无线技术来建立连接,诸如BTLE标准所指定的连接)。如果建立了与移动装置的连接,则在802处确定该连接,然后在803处将所收集的数据传输到移动装置。在804处,移动装置将数据传输到IoT中心、外部服务和/或用户。如上所述,如果移动装置已经连接(例如,经由WiFi链路),则可以立即传输数据。

[0091] 除了收集来自IoT装置的数据之外,在一个实施方案中,本文所述的技术可以用于更新或以其他方式向IoT装置提供数据。一个示例示于图9A中,其示出了具有需要安装在IoT装置601(或一组这样的IoT装置)上的程序代码更新901的IoT中心110。程序代码更新可以包括系统更新、补丁、配置数据以及IoT装置按用户需要进行操作所需的任何其他数据。在一个实施方案中,用户可以经由移动装置或计算机为IoT装置601指定配置选项,然后将这些配置选项存储在IoT中心110上,并使用本文所述的技术提供给IoT装置。具体而言,在一个实施方案中,IoT中心110上的中间连接逻辑/应用程序721与移动装置611上的中间连接逻辑/应用程序711进行通信,以将程序代码更新存储在临时存储器615内。当移动装置611进入IoT装置601的范围时,移动装置611上的中间连接逻辑/应用程序711与IoT装置601上的中间连接逻辑/应用程序701相连接,以将程序代码更新提供给装置。在一个实施方案中,IoT装置601然后可以进入自动更新过程,以安装新的程序代码更新和/或数据。

[0092] 更新IoT装置的方法示于图9B中。该方法可在上述系统架构的环境内实施,但不限于任何具体系统架构。

[0093] 在900处,在IoT中心和/或外部服务(例如,通过互联网联接到移动装置)上提供新的程序代码或数据更新。在901处,移动装置代表IoT装置接收并存储程序代码或数据更新。在902处,IoT装置和/或移动装置周期性地进行检查,以确定是否已经建立连接。如果在903处确定建立了连接,则在904处,将更新传输到IoT装置并进行安装。

[0094] 提高安全性的实施方案

[0095] 在一个实施方案中,每个IoT装置101的低功率微控制器200和IoT中心110的低功率逻辑/微控制器301包括安全密钥存储器,以用于存储由下述实施方案使用的加密密钥(参见,例如图10至图15和相关文字)。作为另外一种选择,密钥可以被保护在如下所述的用户识别模块(SIM)中。

[0096] 图10示出了一种高级架构,使用公共密钥基础结构(PKI)技术和/或对称密钥交

换/加密技术来加密IoT服务120、IoT中心110和IoT装置101-102之间的通信。

[0097] 现将首先描述使用公共密钥/私有密钥对的实施方案,随后描述使用对称密钥交换/加密技术的实施方案。具体而言,在使用PKI的实施方案中,唯一的公共密钥/私有密钥对与每个IoT装置101-102、每个IoT中心110和IoT服务120相关联。在一个实施方案中,当建立新的IoT中心110时,其公共密钥被提供给IoT服务120,并且当建立新的IoT装置101时,其公共密钥被提供给IoT中心110和IoT服务120。下文描述了用于在装置之间安全地交换公共密钥的各种技术。在一个实施方案中,所有公共密钥由所有接收装置已知的主密钥签名(即,证书的一种形式),使得任何接收装置都可以通过验证签名来验证公共密钥的有效性。因此,这些证书将被交换,而不仅仅是交换原始公共密钥。

[0098] 如图所示,在一个实施方案中,每个IoT装置101、102分别包括安全密钥存储器1001、1003,以用于安全存储每个设备的私有密钥。然后安全逻辑1002、1304利用安全存储的私有密钥来执行本文描述的加密/解密操作。类似地,IoT中心110包括安全存储器1011,以用于存储IoT中心私有密钥以及IoT装置101-102和IoT服务120的公共密钥;和安全逻辑1012,以用于使用密钥执行加密/解密操作。最后,IoT服务120可以包括安全存储1021,以用于安全存储其自身的私有密钥、各种IoT装置和IoT中心的公共密钥;和安全逻辑1013,以用于使用密钥加密/解密与IoT中心和设备的通信。在一个实施方案中,当IoT中心110接收到来自IoT装置的公共密钥证书时,IoT中心可以验证该证书(例如,通过如上所述使用主密钥验证签名),然后从其中提取公共密钥,并将公共密钥存储于其安全密钥存储器1011中。

[0099] 举例来说,在一个实施方案中,当IoT服务120需要向IoT装置101传输命令或数据时(例如,解锁门的命令、读取传感器的请求、IoT装置待处理/显示的数据等),安全逻辑1013使用IoT装置101的公共密钥对该数据/命令进行加密以生成加密的IoT装置数据包。在一个实施方案中,然后该安全逻辑使用IoT中心110的公共密钥对IoT装置数据包进行加密以生成IoT中心数据包,并且将IoT中心数据包传输到IoT中心110。在一个实施方案中,服务120用上述的私有密钥或主密钥签署该加密消息,使得装置101可以验证它是否正从可信来源接收未更改的消息。装置101然后可以使用对应于私有密钥和/或主密钥的公共密钥来验证该签名。如上所述,可使用对称密钥交换/加密技术来代替公共/私有密钥加密。在这些实施方案中,可以为每个装置提供相同的对称密钥的副本以用于对签名进行加密和验证,而不是私密地存储密钥以及向其他装置提供对应的公共密钥。对称密钥算法的一个示例是高级加密标准(AES),但是本发明的基本原理不限于任何特定类型的对称密钥。

[0100] 使用对称密钥实现方式,每个装置101进入安全密钥交换协议以与IoT中心110交换对称密钥。安全密钥预置协议(诸如动态对称密钥预置协议(DSKPP))可用于通过安全通信信道交换密匙(例如,见意见征求稿(RFC) 6063)。然而,本发明的基本原理不限于任何特定密钥预置协议。

[0101] 一旦对称密钥已被交换,它们可被每个装置101和IoT中心110用来对通信进行加密。类似地,IoT中心110和IoT服务120可以执行安全对称密钥交换,然后使用交换的对称密钥对通信进行加密。在一个实施方案中,在装置101和中心110之间以及中心110和IoT服务120之间周期性地交换新的对称密钥。在一个实施方案中,新的对称密钥随着装置101、中心110和服务120之间的每个新的通信会话而交换(例如,为每个通信会话生成并安全地交换新的密钥)。在一个实施方案中,如果IoT中心中的安全模块1012是可信的,则服务120可以

与中心安全模块1312协商会话密钥,然后安全模块1012将与每个装置120协商会话密钥。然后在被重新加密以传输到装置101之前,来自服务120的消息将在中心安全模块1012中被解密和验证。

[0102] 在一个实施方案中,为了防止中心安全模块1012上的泄密,可以在安装时在装置101和服务120之间协商一次性(永久)安装密钥。当向装置101发送消息时,服务120可以首先用该装置安装密钥来加密/MAC,然后使用该中心的会话密钥加密/MAC。然后中心110将验证并提取被加密的装置模糊点并将其发送到装置。

[0103] 在本发明的一个实施方案中,采用计数器机制来防止重放攻击。例如,可以为从装置101到中心110(或反之亦然)的每个连续通信分配持续增加的计数器值。中心110和装置101都将跟踪该值并且验证该值在装置之间的每个连续通信中是否正确。在中心110和服务120之间可以实现相同的技术。以这种方式使用计数器将使得更难以欺骗每个装置之间的通信(因为计数器值将是不正确的)。但是,即使没有该计数器机制,服务和装置之间的共享安装密钥也将防止对所有装置的网络(中心)广泛的攻击。

[0104] 在一个实施方案中,当使用公共/私有密钥加密时,IoT中心110使用其私有密钥对IoT中心数据包进行解密并且生成加密的IoT装置数据包,然后将其传输到相关联的IoT装置101。IoT装置101然后使用其私有密钥对IoT装置数据包进行解密以生成源自IoT服务120的命令/数据。然后它可以处理该数据和/或执行该命令。使用对称加密,每个装置将使用共享的对称密钥进行加密和解密。在任一种情况下,每个传输装置也可以使用其私有密钥签署消息,使得接收装置可以验证其真实性。

[0105] 可以使用不同组的密钥对从IoT装置101到IoT中心110和IoT服务120的通信进行加密。例如,在一个实施方案中,使用公共/私有密钥布置,IoT装置101上的安全逻辑1002使用IoT中心110的公共密钥对发送到IoT中心110的数据包进行加密。然后IoT中心110上的安全逻辑1012可以使用IoT中心的私有密钥对数据包进行解密。类似地,IoT装置101上的安全逻辑1002和/或IoT中心110上的安全逻辑1012可以使用IoT服务120的公共密钥对发送到IoT服务120的数据包进行加密(然后其可由IoT服务120上的安全逻辑1013使用服务的私有密钥进行解密)。使用对称密钥,装置101和中心110可以共享对称密钥,而中心和服务120可以共享不同的对称密钥。

[0106] 虽然上文在以上说明中陈述了某些具体细节,但应该注意,本发明的基本原理可以使用各种不同的加密技术来实现。例如,虽然上述一些实施方案使用非对称公共/私有密钥对,但另选的实施方案可以使用在各种IoT装置101-102、IoT中心110和IoT服务120之间安全交换的对称密钥。此外,在一些实施方案中,数据/命令本身不被加密,而是使用密钥在该数据/命令(或其他数据结构)上生成签名。然后接受方可以使用其密钥来查验签名。

[0107] 如图11所示,在一个实施方案中,使用可编程用户识别模块(SIM)1101来实现每个IoT装置101上的安全密钥存储。在该实施方案中,最初可通过安置在IoT装置101上的SIM接口1100内的未编程的SIM卡1101将IoT装置101提供给最终用户。为了用一组一个或多个加密密钥对SIM进行编程,用户将可编程SIM卡1101从SIM接口500取出并将其插入到IoT中心110上的SIM编程接口1102中。然后IoT中心上的编程逻辑1125对SIM卡1101进行安全地编程以将IoT装置101与IoT中心110和IoT服务120进行注册/配对。在一个实施方案中,可以由编程逻辑1125随机地生成公共/私有密钥对,然后可以将该密钥对中的公共密钥存储在IoT中

心的安全存储装置411中,而将该密钥对中的私有密钥存储在可编程SIM 1101内。另外,编程逻辑525可以将IoT中心110、IoT服务120和/或任何其他IoT装置101的公共密钥存储在SIM卡1401上(待由IoT装置101上的安全逻辑1302用来对传出数据进行加密)。一旦SIM 1101被编程,使用SIM作为安全标识符(例如,使用用于使用SIM注册装置的现有技术)新的IoT装置101可以预置有IoT服务120。在预置之后,IoT中心110和IoT服务120均将安全地存储在在与IoT装置101的通信进行加密时要使用的IoT装置的公共密钥的副本。

[0108] 上文结合图11所述的技术为向最终用户提供新的IoT装置提供了巨大的灵活性。SIM卡可以由最终用户通过IoT中心110直接编程并且编程的结果可被安全地传送到IoT服务120,而不需要用户在销售/购买时将每个SIM直接注册到特定的服务提供商(如当前所做的那样)。因此,新的IoT装置101可以由在线或本地零售商出售给最终用户,并且随后被安全地预置有IoT服务120。

[0109] 尽管上文在SIM(用户识别模块)的特定上下文中描述了注册和加密技术,但是本发明的基本原理不限于“SIM”装置。相反,可以使用任何类型的用于存储一组加密密钥的安全存储装置来实现本发明的基本原理。而且,尽管上述实施方案包括可移除的SIM装置,但是在一个实施方案中,SIM装置不可移除,但是IoT装置本身可以插入到IoT中心110的编程接口1102内。

[0110] 在一个实施方案中,在分发到最终用户之前将SIM预编程到IoT装置101中,而不是要求用户对SIM(或其他装置)进行编程。在该实施方案中,当用户设置IoT装置101时,可以使用本文所述的各种技术来安全地交换IoT中心110/IoT服务120和新IoT装置101之间的加密密钥。

[0111] 例如,如图12A所示,每个IoT装置101或SIM 401可以与唯一识别IoT装置101和/或SIM 1001的条形码或QR码1501封装在一起。在一个实施方案中,条形码或QR码1201包括用于IoT装置101或SIM 1001的公共密钥的编码表示。作为另外一种选择,条形码或QR码1201可以由IoT中心110和/或IoT服务120用来识别或生成公共密钥(例如,用作指向已存储在安全存储装置中的公共密钥)。条形码或QR码601可印刷在单独的卡上(如图12A所示)或者可直接印刷在IoT装置本身上。无论条形码印刷在何处,在一个实施方案中,IoT中心110配备有条形码读取器206,以用于读取条形码并将结果数据提供给IoT中心110上的安全逻辑1012和/或IoT服务120上的安全逻辑1013。然后IoT中心110上的安全逻辑1012可以将IoT装置的公共密钥存储在其安全密钥存储装置1011内,并且IoT服务120上的安全逻辑1013可以将公共密钥存储在其安全存储装置1021内(以用于随后的加密通信)。

[0112] 在一个实施方案中,包含在条形码或QR码1201中的数据也可以通过安装有IoT服务提供商设计的IoT应用程序或基于浏览器的小程序的用户装置135(例如,诸如iPhone或Android装置)来捕捉。一旦被捕捉,条形码数据可以通过安全连接(例如,诸如安全套接字层(SSL)连接)被安全地传送到IoT服务120。条形码数据也可以通过安全的本地连接(例如,通过本地WiFi或蓝牙LE连接)从客户端装置135提供给IoT中心110。

[0113] IoT装置101上的安全逻辑1002和IoT中心110上的安全逻辑1012可以使用硬件、软件、固件或其任何组合来实现。例如,在一个实施方案中,安全逻辑1002、1012被实现在用于在IoT装置101和IoT中心110之间建立本地通信信道130的芯片内(例如,蓝牙LE芯片,如果本地信道130是蓝牙LE)。无论安全逻辑1002、1012的具体位置如何,在一个实施方案中,安

全逻辑1002、1012被设计为建立用于执行某些类型的程序代码的安全执行环境。例如,这可以通过使用TrustZone技术(在某些ARM处理器上可用)和/或可信执行技术(由Intel设计)来实现。当然,本发明的基本原理不限于任何特定类型的安全执行技术。

[0114] 在一个实施方案中,条形码或QR码1501可用于将每个IoT装置101与IoT中心110配对。例如,可以将嵌入在条形码或QR码1501内的配对代码提供给IoT中心110,以将IoT中心与对应的IoT装置配对,而不是使用当前用于配对蓝牙LE装置的标准无线配对过程。

[0115] 图12B示出了其中IoT中心110上的条形码读取器206捕捉与IoT装置101相关联的条形码/QR码1201的一个实施方案。如上所述,条形码/QR码1201可以直接印刷在IoT装置101上,或者可以印刷在设置有IoT装置101的单独的卡上。在任一种情况下,条形码读取器206从条形码/QR码1201中读取配对代码,并将配对代码提供给本地通信模块1280。在一个实施方案中,本地通信模块1280是蓝牙LE芯片和相关联的软件,但是本发明的基本原理不限于任何特定的协议标准。一旦接收到配对代码,则将其存储在包含配对数据1285的安全存储装置中,并且IoT装置101和IoT中心110被自动配对。每当IoT中心以这种方式与新的IoT装置配对时,用于该配对的配对数据就被存储在安全存储装置685内。在一个实施方案中,一旦IoT中心110的本地通信模块1280接收到配对代码,则其可以使用该代码作为密钥对IoT装置101的本地无线信道上的通信进行加密。

[0116] 类似地,在IoT装置101方面,本地通信模块1590将配对数据存储在指示与IoT中心配对的本地安全存储装置1595内。配对数据1295可以包括在条形码/QR码1201中识别的预编程的配对代码。配对数据1295还可以包括建立安全本地通信信道所需的从IoT中心110上的本地通信模块1280接收的配对数据(例如,用于对与IoT中心110的通信进行加密的附加密钥)。

[0117] 因为配对代码不通过无线传输,因此条形码/QR码1201可用于以比当前无线配对协议更安全的方式执行本地配对。另外,在一个实施方案中,可以使用用于配对的相同的条形码/QR码1201来识别加密密钥以建立从IoT装置101到IoT中心110以及从IoT中心110到IoT服务120的安全连接。

[0118] 图13示出了根据本发明的一个实施方案的用于编程SIM卡的方法。该方法可在上述的系统架构内实施,但不限于任何特定系统架构。

[0119] 在1301处,用户接收具有空白SIM卡的新IoT装置,并且在1602处,用户将空白SIM卡插入到IoT中心中。在1303处,用户用一组一个或多个加密密钥对空白SIM卡进行编程。例如,如上所述,在一个实施方案中,IoT中心可以随机生成公共/私有密钥对,并且将私有密钥存储在SIM卡上,并将公共密钥存储在其本地安全存储装置中。另外,在1304处,至少公共密钥被传输到IoT服务,使得其可以用于识别IoT装置并且建立与IoT装置的加密通信。如上所述,在一个实施方案中,在图13所示的方法中可使用除了“SIM”卡之外的可编程装置来执行与SIM卡相同的功能。

[0120] 图14示出了用于将新的IoT装置集成到网络中的方法。该方法可在上述的系统架构内实施,但不限于任何特定系统架构。

[0121] 在1401处,用户接收已预分配了加密密钥的新的IoT装置。在1402处,密钥被安全地提供给IoT中心。如上所述,在一个实施方案中,这涉及读取与IoT装置相关联的条形码以识别分配给该装置的公共/私有密钥对的公共密钥。条形码可以由IoT中心直接读取,也可

以通过移动装置通过应用程序或浏览器来捕捉。在另选的实施方案中,可以在IoT装置和IoT中心之间建立安全通信信道诸如蓝牙LE信道、近场通信(NFC)信道或安全WiFi信道以交换密钥。无论密钥是如何传输的,一旦被接收到,则将其存储在IoT中心装置的安全密钥库中。如上所述,在IoT中心上可以使用各种安全执行技术来存储和保护密钥,诸如安全区域、可信执行技术(TXT)和/或Trustzone。另外,在803处,密钥被安全地传输到IoT服务,该IoT服务将密钥存储在其自己的安全密钥库中。然后其可以使用该密钥对与IoT装置的通信进行加密。再次,可以使用证书/签署的密钥来实现交换。在中心110内,防止修改/添加/移除所存储的密钥是特别重要的。

[0122] 图15示出了用于使用公共/私有密钥将命令/数据安全地传送到IoT装置的方法。该方法可在上述的系统架构内实施,但不限于任何特定系统架构。

[0123] 在1501处,IoT服务使用IoT装置公共密钥对数据/命令进行加密以创建IoT装置数据包。然后使用IoT中心的公共密钥对该IoT装置数据包进行加密以创建IoT中心数据包(例如,在IoT装置数据包周围创建IoT中心包装)。在1502处,IoT服务将IoT中心数据包传输到IoT中心。在1503处,IoT中心使用IoT中心的私有密钥对IoT中心数据包进行解密以生成IoT装置数据包。然后在1504处,其将IoT装置数据包传输到IoT装置,在1505处,使用IoT装置私有密钥对IoT装置数据包进行解密以生成数据/命令。在1506处,IoT装置处理数据/命令。

[0124] 在使用对称密钥的实施方案中,可以在每个装置之间(例如,每个装置和中心之间以及中心和服务之间)协商对称密钥交换。一旦完成密钥交换,每个传输装置在将数据传输到接收装置之前使用对称密钥对每次传输进行加密和/或签署。

[0125] 用于在物联网(IoT)系统中建立安全通信信道的设备和方法

[0126] 在本发明的一个实施方案中,无论用于支持通信信道的中间装置如何(例如,诸如用户的移动装置611和/或IoT中心110),都要在IoT服务120和每个IoT装置101之间执行数据的加密和解密。图16A示出了通过IoT中心110进行通信的一个实施方案,而图16B示出了不需要IoT中心的另一个实施方案。

[0127] 首先转到图16A,IoT服务120包括管理一组“服务会话密钥”1650的加密引擎1660,并且每个IoT装置101包括管理用于对IoT装置101和IoT服务120之间的通信进行加密/解密的一组“装置会话密钥”1651的加密引擎1661。当执行本文所述的安全/加密技术时,加密引擎可以依赖于不同的硬件模块,该硬件模块包括用于(除别的以外)生成会话公共/私有密钥对并且防止对该密钥对中的私有会话密钥进行访问的硬件安全模块1630-1631,以及用于使用得出的密码生成密钥流的密钥流生成模块1640-1641。在一个实施方案中,服务会话密钥1650和装置会话密钥1651包括相关的公共/私有密钥对。例如,在一个实施方案中,IoT装置101上的装置会话密钥1651包括IoT服务120的公共密钥和IoT装置101的私有密钥。如下文详细讨论的,在一个实施方案中,为了建立安全通信会话,每个加密引擎1660和1661分别使用公共/私有会话密钥对1650和1651来生成相同的密码,然后该密码被SKGM 1640-1641用来生成密钥流以对IoT服务120与IoT装置101之间的通信进行加密和解密。下文提供了根据本发明的一个实施方案的与密码的生成和使用相关联的附加细节。

[0128] 在图16A中,一旦使用密钥1650-1651生成了密码,则客户端将总是通过IoT服务120向IoT装置101发送消息,如清除事务1611所示。本文所用的“清除”意指表示底层消息没有使用本文所述的加密技术进行加密。然而,如图所示,在一个实施方案中,在客户端装置

611和IoT服务120之间建立安全套接字层(SSL)信道或其他安全信道(例如,互联网协议安全(IPSEC)信道)以保护通信。然后IoT服务120上的加密引擎1660使用所生成的密码对消息进行加密,并且在1602处将加密的消息传输到IoT中心110。在一个实施方案中,使用密码和计数器值来生成用于对每个消息数据包进行加密的密钥流,而不是使用密码直接对消息进行加密。下文结合图17描述了该实施方案的细节。

[0129] 如图所示,可以在IoT服务120和IoT中心110之间建立SSL连接或其他安全信道。IoT中心110(其在一个实施方案中不具有对消息进行解密的能力)在1603处(例如,通过蓝牙低功耗(BTLE)通信信道)将加密的消息传输到IoT装置。然后IoT装置101上的加密引擎1661可以使用密码对消息进行解密并且处理消息内容。在使用密码来生成密钥流的实施方案中,加密引擎1661可以使用密码和计数器值来生成密钥流,然后使用密钥流对消息数据包进行解密。

[0130] 消息本身可以包括IoT服务120和IoT装置101之间的任何形式的通信。例如,消息可以包括指示IoT装置101执行特定功能诸如进行测量并将结果报告回客户端装置611的命令数据包,或者可以包括用于配置IoT装置101的操作的配置数据。

[0131] 如果需要响应,那么在1604处,IoT装置101上的加密引擎1661使用密码或得出的密钥流对响应进行加密并且将所加密的响应传输到IoT中心110,该IoT中心在1605处将响应转发到IoT服务120。然后在1606处(例如,通过SSL或其他安全通信信道),IoT服务120上的加密引擎1660使用密码或得出的密钥流对响应进行解密并且将解密的响应传输到客户端装置611。

[0132] 图16B示出了不需要IoT中心的实施方案。相反地,在该实施方案中,IoT装置101和IoT服务120之间的通信通过客户端装置611发生(例如,如上文结合图6至图9B所述)。在该实施方案中,为了向IoT装置101传输消息,客户端装置611在1611处将消息的未加密版本传输到IoT服务120。在1612处,加密引擎1660使用密码或得出的密钥流对消息进行加密,并且将加密的消息传输回客户端装置611。然后客户端装置611在1613处将加密的消息转发到IoT装置101,并且加密引擎1661使用密码或得出的密钥流对消息进行解密。然后IoT装置101可以如本文所述处理消息。如果需要响应,那么在1614处加密引擎1661使用密码对响应进行加密并且将所加密的响应传输到客户端装置611,该客户端装置在1615处将所加密的响应转发到IoT服务120。然后在1616处,加密引擎1660对该响应进行解密并且将解密的响应传输到客户端装置611。

[0133] 图17示出了可最初在IoT服务120和IoT装置101之间执行的密钥交换和密钥流生成。在一个实施方案中,每当IoT服务120和IoT装置101建立新的通信会话时可执行该密钥交换。作为另外一种选择,可以执行密钥交换,并且所交换的会话密钥可以使用指定的时间段(例如,一天、一周等)。虽然为简单起见在图17没有示出中间装置,但通信可以通过IoT中心110和/或客户端装置611发生。

[0134] 在一个实施方案中,IoT服务120的加密引擎1660向HSM 1630(例如,其可以是诸如由Amazon提供的CloudHSM)发送命令以生成会话公共/私有密钥对。随后,HSM 1630可以防止对该密钥对中的私有会话密钥进行访问。类似地,IoT装置101上的加密引擎可以向HSM 1631(例如,得自Atmel Corporation®的Atecc508 HSM)传输命令,HSM 1631生成会话公共/私有密钥对并且防止对该密钥对中的会话私有密钥进行访问。当然,本发明的基本原理不

限于任何特定类型的加密引擎或制造商。

[0135] 在一个实施方案中,在1701处,IoT服务120将使用HSM 1630生成的会话公共密钥传输到IoT装置101。IoT装置使用其HSM 1631来生成其自己的会话公共/私有密钥对,并且在1702处将其密匙对中的公共密钥传输到IoT服务120。在一个实施方案中,加密引擎1660-1661使用椭圆曲线Diffie-Hellman (ECDH) 协议来建立共享密码,该协议是允许双方具有椭圆曲线公共-私有密匙对的匿名密钥协定。在一个实施方案中,使用这些技术,在1703处,IoT服务120的加密引擎1660使用IoT装置会话公共密钥和其自己的会话私有密钥生成密码。类似地,在1704处,IoT装置101的加密引擎1661使用IoT服务120会话公共密钥和其自己的会话私有密钥单独地生成相同的密码。更具体地讲,在一个实施方案中,IoT服务120上的加密引擎1660根据以下公式生成密码:密码=IoT装置会话公共密钥*IoT服务会话私有密钥,其中‘*’表示IoT装置会话公共密钥以点对点的方式乘以IoT服务会话私有密钥。IoT装置101上的加密引擎1661根据以下公式生成密码:密码=IoT服务会话公共密钥*IoT装置会话私有密钥,其中IoT服务会话公共密钥以点对点的方式乘以IoT装置会话私有密钥。最后,如下所述,IoT服务120和IoT装置101均已生成了用于对通信进行加密的相同的密码。在一个实施方案中,加密引擎1660-1661分别依赖于硬件模块诸如KSGM 1640-1641来执行上述用于生成密码的操作。

[0136] 一旦秘密已被确定,其可由加密引擎1660和1661用来直接对数据进行加密和解密。或者,在一个实施方案中,加密引擎1660-1661向KSGM1640-1641发送使用密码生成新的密钥流的命令以对每个数据包进行加密/解密(即,为每个数据包生成新的密钥流数据结构)。具体地讲,密钥流生成模块1640-1641的一个实施方案采用Galois/计数器模式(GCM),其中计数器值针对每个数据包递增,并且与密钥组合使用以生成密钥流。因此,为了将数据包传输到IoT服务120,IoT装置101的加密引擎1661使用密码和当前计数器值来使KSGM 1640-1641生成新的密钥流并且使计数器值递增以生成下一个密钥流。然后,在传输到IoT服务120之前,新生成的密钥流被用于对数据包进行加密。在一个实施方案中,密钥流与数据进行异或运算(XOR)以生成加密的数据包。在一个实施方案中,IoT装置101将具有所加密数据包的计数器值传输到IoT服务120。然后IoT服务上的加密引擎1660与KSGM 1640通信,KSGM 1640使用所接收的计数器值和密码生成密钥流(由于使用的是相同的密码和计数器值,其应该是相同的密钥流)并且使用所生成的密钥流对数据包进行解密。

[0137] 在一个实施方案中,从IoT服务120传输到IoT装置101的数据包以相同的方式被加密。具体地讲,计数器对于每个数据包递增,并且与密码一起使用以生成新的密钥流。然后该密钥流被用来对数据进行加密(例如,执行数据和密钥流的XOR运算),并且所加密的数据包与计数器值一起被传输到IoT装置101。然后IoT装置101上的加密引擎1661与KSGM1641通信,KSGM 1641使用计数器值和密码生成用于对数据包进行解密的相同的密钥流。因此,在该实施方案中,加密引擎1660-1661使用它们自己的计数器值生成用于对数据进行加密的密钥流,并且使用与所加密的数据包一起被接收的计数器值生成用于对数据进行解密的密钥流。

[0138] 在一个实施方案中,每个加密引擎1660-1661跟踪其从另一个加密引擎接收到的最后的计数器值,并且包括用于检测计数器值是否不按顺序接收或者是否多次接收到相同的计数器值的排序逻辑。如果一个计数器值不按顺序接收,或者如果多次收到相同的计数

器值,这可能指示正在尝试重放攻击。作为响应,加密引擎1660-1661可以从通信信道断开和/或可以生成安全警报。

[0139] 图18示出了本发明的一个实施方案中所采用的包括4字节计数器值1800、可变大小的加密数据字段1801和6字节标签1802的示例性加密数据包。在一个实施方案中,标签1802包括校验和值以查验解密的数据(一旦该数据已被解密)。

[0140] 如上所述,在一个实施方案中,可以周期性地和/或响应于每个新的通信会话的发起来生成在IoT服务120和IoT装置101之间交换的会话公共/私有密钥对1650-1651。

[0141] 本发明的一个实施方案实现用于验证IoT服务120和IoT装置101之间的会话的附加技术。具体地讲,在一个实施方案中,使用公共/私有密钥对的分级结构,包括主密钥对、一组工厂密钥对和一组IoT服务密钥对,以及一组IoT装置密钥对。在一个实施方案中,主密钥对包括所有其他密钥对的信任根并且被保持在单个高度安全的位置(例如,在实现本文所述的IoT系统的组织的控制下)。主私有密钥可用于对各种其他密钥对诸如工厂密钥对生成签名(并由此进行验证)。然后可使用主公钥来核验签名。在一个实施方案中,制造IoT装置的每个工厂被分配其自己的工厂密钥对,然后可以用它来验证IoT服务密钥和IoT装置密钥。例如,在一个实施方案中,使用工厂私有密钥来生成对IoT服务公共密钥和IoT装置公共密钥的签名。然后可使用对应的工厂公共密钥来核验这些签名。请注意,这些IoT服务/装置公共密钥与上文结合图16A至图16B所述的“会话”公共密钥/私有密钥不同。上述会话公共密钥/私有密钥是临时的(即,针对服务/装置会话生成),而IoT服务/装置密钥对是永久的(即,在工厂生成)。

[0142] 考虑到主密钥、工厂密钥、服务/装置密钥之间的上述关系,本发明的一个实施方案执行以下操作以在IoT服务120和IoT装置101之间提供附加的验证和安全层:

[0143] A. 在一个实施方案中,IoT服务120初始生成包含以下内容的消息:

[0144] 1. IoT服务的唯一ID:

- [0145] • IoT服务的序列号;
- [0146] • 时间戳;
- [0147] • 用于签署此唯一ID的工厂密钥的ID;
- [0148] • 唯一ID的类别(即,服务);
- [0149] • IoT服务的公共密钥
- [0150] • 对该唯一ID的签名。

[0151] 2. 工厂证书包括:

- [0152] • 时间戳
- [0153] • 用于签署该证书的主密钥的ID
- [0154] • 工厂的公共密钥
- [0155] • 工厂证书的签名

[0156] 3. IoT服务会话公共密钥(如上文结合图16A至图16B所述)

[0157] 4. IoT服务会话公共密钥签名(例如,使用IoT服务的私有密钥签署)

[0158] B. 在一个实施方案中,消息在协商信道上被发送至IoT装置(如下所述)。IoT装置解析消息并:

[0159] 1. 核验工厂证书的签名(仅当其存在于消息有效负载中时)

- [0160] 2.使用由唯一ID标识的密钥来核验唯一ID的签名
- [0161] 3.使用来自唯一ID的IoT服务的公共密钥来核验IoT服务会话公共密钥签名
- [0162] 4.保存IoT服务的公共密钥以及IoT服务的会话公共密钥
- [0163] 5.生成IoT装置会话密钥对
- [0164] C.IoT装置然后生成包含以下内容的消息：
- [0165] 1.IoT装置的唯一ID
- [0166] • IoT装置序列号
- [0167] • 时间戳
- [0168] • 用于签署此唯一ID的工厂密钥的ID
- [0169] • 唯一ID的类别(即,IoT装置)
- [0170] • IoT装置的公共密钥
- [0171] • 唯一ID的签名
- [0172] 2.IoT装置的会话公共密钥
- [0173] 3.使用IoT装置的密钥签署(IoT装置会话公共密钥+IoT服务会话公共密钥)的签名
- [0174] D.该消息被发送回IoT服务。IoT服务解析消息并：
- [0175] 1.使用工厂公共密钥来核验唯一ID的签名
- [0176] 2.使用IoT装置的公共密钥来核验会话公共密钥的签名
- [0177] 3.保存IoT装置的会话公共密钥
- [0178] E.IoT服务然后生成包含用IoT服务的密钥签署(IoT装置会话公共密钥+IoT服务会话公共密钥)的签名的消息。
- [0179] F.IoT装置解析消息并：
- [0180] 1.使用IoT服务的公共密钥来核验会话公共密钥的签名
- [0181] 2.由IoT装置会话私有密钥和IoT服务会话公共密钥生成密钥流
- [0182] 3.IoT装置然后发送“消息传送可用”消息。
- [0183] G.IoT服务然后执行以下操作：
- [0184] 1.由IoT装置会话私有密钥和IoT装置的会话公共密钥生成密钥流
- [0185] 2.在消息传送信道上创建包含以下内容的新消息：
- [0186] • 生成并存储随机的2字节值
- [0187] • 用回旋镖(boomerang)属性Id(如下所述)和随机值设置属性消息
- [0188] H.IoT装置收到该消息并：
- [0189] 1.尝试解密该消息
- [0190] 2.在指定的属性Id上使用相同的值发出更新
- [0191] I.IoT服务识别包含回旋镖属性更新的消息有效负载并：
- [0192] 1.将其配对状态设置为真
- [0193] 2.在协商信道上发送配对完成消息
- [0194] J.IoT装置接收该消息并将其配对状态设置为真
- [0195] 尽管关于“IoT服务”和“IoT装置”描述了上述技术,但是本发明的基本原理可以被实现为在包括用户客户端装置、服务器和互联网服务的任何两个装置之间建立安全通信信

道。

[0196] 上述技术是高度安全的,因为私有密钥永远不会通过无线共享(相比之下,在当前的蓝牙配对技术中密码从一方传输到另一方)。窃听整个对话的攻击者只有公共密钥,而这些公共密钥不足以生成共享密码。这些技术还通过交换经签署的公共密钥来防止中间人攻击。另外,由于在每个设备上都使用GCM和单独的计数器,因此可以防止任何类型的“重放攻击”(在这种情况下,中间人捕捉数据并再次发送)。一些实施方案还通过使用不对称计数器来防止重放攻击。

[0197] 在没有正式配对装置的情况下交换数据和命令的技术

[0198] GATT是通用属性配置文件(Generic Attribute Profile)的首字母缩写,并且GATT定义了两个蓝牙低功耗(BTLE)装置来回传送数据的方式。它利用称为属性协议(ATT)的通用数据协议,该协议用于将服务、特征和相关数据存储在简单的查找表中,对于表中的每个条目使用16位特征ID。请注意,虽然“特征”有时被称为“属性”。

[0199] 在蓝牙装置上,最常用的特征是装置“名称”(具有特征ID 10752(0×2A00))。例如,蓝牙装置可通过使用GATT读取由那些其他蓝牙装置发布的“名称”特征来标识其附近的其他蓝牙装置。因此,蓝牙装置具有交换数据的固有能力和无需正式配对/绑定装置(注意,“配对”和“绑定”有时可以互换使用;本讨论的其余部分将使用术语“配对”)。

[0200] 本发明的一个实施方案利用这种能力与已启用BTLE的IoT装置进行通信,而无需与这些装置正式配对。由于与每个装置配对所需的时间量,并且一次只可建立一个配对连接,所以与每个单独的IoT装置配对将非常低效。

[0201] 图19示出了其中蓝牙(BT)装置1910与IoT装置101的BT通信模块1901建立网络套接字抽象而不正式建立配对的BT连接的一个特定实施方案。BT装置1910可被包括在如图16A所示的IoT中心110和/或客户端装置611中。如图所示,BT通信模块1901维护包含特征ID列表、与这些特征ID相关联的名称以及这些特征ID的值的结构。根据当前的BT标准,每个特征的值可存储在由特征ID标识的20字节缓冲区内。然而,本发明的基本原理不限于任何特定的缓冲区大小。

[0202] 在图19的示例中,“名称”特征是被分配给“IoT装置14”的特定值的BT定义的特征。本发明的一个实施方案指定要用于与BT装置1910协商安全通信信道的第一组附加特征和要用于与BT装置1910的加密通信的第二组附加特征。具体地讲,在例示的示例中由特征ID<65532>标识的“协商写入”特征可用于传输传出协商消息,并且由特征ID<65533>标识的“协商读取”特征可用于接收传入协商消息。“协商消息”可包括由BT装置1910和BT通信模块1901用来建立如本文所述的安全通信信道的消息。举例来说,在图17中,IoT装置101可经由“协商读取”特征<65533>来接收IoT服务会话公共密钥1701。密钥1701可从IoT服务120传输到已启用BTLE的IoT中心110或客户端装置611,然后可使用GATT将密钥1701写入由特征ID<65533>标识的协商读取值缓冲区。然后,IoT装置应用程序逻辑1902可从由特征ID<65533>标识的值缓冲区中读取密钥1701并且如上所述处理它(例如,使用它来生成密码并且使用该密码来生成密钥流等)。

[0203] 如果密钥1701大于20字节(在一些当前具体实施中为最大缓冲区大小),则该密钥可以20字节部分来写入。例如,可通过BT通信模块1903将前20个字节写入特征ID<65533>,并由IoT装置应用程序逻辑1902读取,该IoT装置应用程序逻辑然后可将确认消息写入由特

征ID<65532>标识的协商写入值缓冲区。使用GATT,BT通信模块1903可从特征ID<65532>读取该确认,并且响应地将密钥1701的接下来的20个字节写入由特征ID<65533>标识的协商读取值缓冲区。以这种方式,建立由特征ID<65532>和<65533>定义的网络套接字抽象,以交换用于建立安全通信信道的协商消息。

[0204] 在一个实施方案中,一旦建立安全通信信道,就使用特征ID<65534>(用于从IoT装置101传输所加密的数据包)和特征ID<65533>(用于通过IoT装置接收加密的数据包)建立第二网络套接字抽象。也就是说,当BT通信模块1903具有用于传输的加密数据包(例如,诸如图16A的加密消息1603)时,它使用由特征ID<65533>标识的消息读取值缓冲区开始写入所加密的数据包,每次20个字节。IoT装置应用程序逻辑1902随后将从读取值缓冲区中读取加密数据包,每次20个字节,根据需要经由特征ID<65532>所标识的写入值缓冲区向BT通信模块1903发送确认消息。

[0205] 在一个实施方案中,下文所述的GET、SET和UPDATE的命令用于在两个BT通信模块1901和1903之间交换数据和命令。例如,BT通信模块1903可发送标识特征ID<65533>并且包含SET命令的数据包以写入由特征ID<65533>标识的值字段/缓冲区,该数据包然后可由IoT装置应用程序逻辑1902读取。为了从IoT装置101检索数据,BT通信模块1903可传输被指向由特征ID<65534>标识的值字段/缓冲区的GET命令。响应于GET命令,BT通信模块1901可向包含来自由特征ID<65534>标识的值字段/缓冲区的数据的BT通信模块1903传输UPDATE数据包。另外,响应于IoT装置101上的特定属性的改变,可以自动传输UPDATE数据包。例如,如果IoT装置与照明系统相关联并且用户打开灯,则可以发送UPDATE数据包以反映与照明应用程序相关联的开/关属性的改变。

[0206] 图20示出了根据本发明的一个实施方案的用于GET、SET和UPDATE的示例性数据包格式。在一个实施方案中,这些数据包在协商之后通过消息写入<65534>和消息读取<65533>信道传输。在GET数据包2001中,第一个1字节的字段包括将该数据包标识为GET数据包的值(0×10)。第二个1字节的字段包括请求ID,该请求ID唯一地标识当前的GET命令(即,标识GET命令与之关联的当前交易)。例如,从服务或装置传输的GET命令的每个实例均可被分配不同的请求ID。这可以通过例如递增计数器并使用计数器值作为请求ID来完成。然而,本发明的基本原理不限于设置请求ID的任何特定方式。

[0207] 2字节的属性ID标识数据包所指向的特定于应用程序的属性。例如,如果GET命令正被发送到图19中所示的IoT装置101,则属性ID可用于标识被请求的具体的特定于应用程序的值。回到上面的示例,GET命令可被指向特定于应用程序的属性ID诸如照明系统的电力状态,其包括标识灯是开启还是关闭的值(例如,1=开启,0=关闭)。如果IoT装置101是与门相关联的安全设备,则值字段可标识门的当前状态(例如,1=打开,0=关闭)。响应于GET命令,可发送包含由属性ID标识的当前值的响应。

[0208] 如图20所示的SET数据包2002和UPDATE数据包2003还包括标识数据包类型(即,SET和UPDATE)的第一个1字节字段、包含请求ID的第二个1字节字段和标识应用程序定义属性的2字节属性ID字段。另外,SET数据包包括标识包含在n字节值数据字段中的数据长度的2字节长度值。值数据字段可包括将在IoT装置上执行的命令和/或用于以某种方式配置IoT装置的操作(例如,设置期望参数、关闭IoT装置等)的配置数据。例如,如果IoT装置101控制风扇的速度,则值字段可反映当前的风扇速度。

[0209] UPDATE数据包2003可被传输以提供SET命令的结果的更新。UPDATE数据包2003包括2字节长度值字段,以标识可包括与SET命令的结果有关的数据的n字节值数据字段的长度。另外,1字节的更新状态字段可标识正在更新的变量的当前状态。例如,如果SET命令试图关闭由IoT装置控制的光,则更新状态字段可指示光是否被成功关闭。

[0210] 图21示出了IoT服务120和涉及SET命令和UPDATE命令的IoT装置101之间的交易的示例性序列。未示出中间装置诸如IoT中心和用户的移动装置,以避免模糊本发明的基本原理。在2101处,SET命令2101从IoT服务传输到IoT装置101,并且由BT通信模块1901接收,该SET命令响应地在2102更新由特征ID标识的GATT值缓冲区。在2103处,通过低功率微控制器(MCU) 200(或者通过在低功率MCU上执行的程序代码,诸如图19中所示的IoT装置应用程序逻辑1902)从值缓冲区读取该SET命令。在2104处,MCU 200或程序代码响应于该SET命令而执行操作。例如,SET命令可包括指定新配置参数诸如新温度的属性ID,或者可包括状态值诸如开/关(以使IoT装置进入“开”或者低功率状态)。因此,在2104处在IoT装置中设置新值,并且在2105处返回UPDATE命令,在2106处在GATT值字段中更新实际值。在某些情况下,实际值将等于期望值。在其他情况下,更新的值可能不同(即,因为IoT装置101可能需要时间来更新某些类型的值)。最后,在2107处,UPDATE命令被传输回包含来自GATT值字段的实际值的IoT服务120。

[0211] 图22示出了根据本发明的一个实施方案的用于实现IoT服务和IoT装置之间的安全通信信道的方法。该方法可在上述网络架构的环境内实施,但不限于任何特定架构。

[0212] 在2201处,IoT服务使用椭圆曲线数字签名算法(ECDSA)证书创建用于与IoT中心通信的加密信道。在2202处,IoT服务使用会话密码加密IoT装置数据包中的数据/命令以创建加密的装置数据包。如上所述,会话密码可由IoT装置和IoT服务独立生成。在2203处,IoT服务通过加密信道将加密的装置数据包传输到IoT中心。在2204处,在不解密的情况下,IoT中心将加密的装置数据包传递给IoT装置。在22-5处,IoT装置使用会话密码来解密加密的装置数据包。如上所述,在一个实施方案中,这可通过如下方式来完成:使用密码和计数器值(与加密的装置数据包一起提供)来生成密钥流,然后使用该密钥流来解密数据包。在2206处,IoT装置然后提取并处理包含在装置数据包内的数据和/或命令。

[0213] 因此,使用上述技术,可以在两个已启用BT的装置之间建立双向、安全的网络套接字抽象,而无需使用标准配对技术正式地配对BT装置。虽然上面关于与IoT服务120通信的IoT装置101描述了这些技术,但是本发明的基本原理可以被实现为在任何两个已启用BT的装置之间协商并建立安全通信信道。

[0214] 图23A至图23C示出了根据本发明的一个实施方案的用于配对装置的详细方法。该方法可在上述系统架构的环境内实施,但不限于任何特定系统架构。

[0215] 在2301处,IoT服务创建包含IoT服务的序列号和公共密钥的数据包。在2302处,IoT服务使用工厂私有密钥签署数据包。在2303处,IoT服务通过加密信道将该数据包发送到IoT中心,并且在2304处,IoT中心通过未加密信道将该数据包转发到IoT装置。在2305处,IoT装置核验该数据包的签名,并且在2306处,IoT装置生成包含IoT装置的序列号和公共密钥的数据包。在2307处,IoT装置使用工厂私有密钥签署该数据包,并且在2308处,IoT装置通过未加密信道将该数据包发送到IoT中心。

[0216] 在2309处,IoT中心通过加密信道将该数据包转发到IoT服务,并且在2310处,IoT

服务核验该数据包的签名。在2311处, IoT服务生成会话密钥对, 并且在2312处, IoT服务生成包含会话公共密钥的数据包。IoT服务然后在2313处利用IoT服务私有密钥签署该数据包, 并且在2314处IoT服务通过加密信道将该数据包发送到IoT中心。

[0217] 转到图23B, IoT中心在2315处通过未加密信道将该数据包转发到IoT装置, 并且在2316处, IoT装置核验数据包的签名。在2317处, IoT装置(例如, 使用上述技术)生成会话密钥对, 并且在2318处, 生成包含IoT装置会话公共密钥的IoT装置数据包。在2319处, IoT装置使用IoT装置私有密钥签署IoT装置数据包。在2320处, IoT装置通过未加密信道将该数据包发送到IoT中心, 并且在2321处, IoT中心通过加密信道将该数据包转发到IoT服务。

[0218] 在2322处, IoT服务核验该数据包的签名(例如, 使用IoT装置公共密钥), 并且在2323处, IoT服务使用IoT服务私有密钥和IoT装置公共密钥来生成会话密码(如上文所详细描述)。在2324处, IoT装置使用IoT装置私有密钥和IoT服务公共密钥来生成会话密码(再次如上所述), 并且在2325处, IoT装置生成随机数字并使用会话密码对其进行加密。在2326处, IoT服务通过加密信道将所加密的数据包发送到IoT中心。在2327处, IoT中心通过未加密信道将所加密的数据包转发到IoT装置。在2328处, IoT装置使用会话密码解密该数据包。

[0219] 转到图23C, 在2329处, IoT装置使用会话密码重新加密该数据包, 并且在2330处, IoT装置通过未加密信道将所加密的数据包发送到IoT中心。在2331处, IoT中心通过加密信道将所加密的数据包转发到IoT服务。在2332处, IoT服务使用会话密码解密该数据包。在2333处, IoT服务核验到该随机数字与其发送的随机数字匹配。IoT服务随后在2334处发送指示配对已完成的数据包, 并且在2335处使用会话密码加密所有后续消息。

[0220] 本发明的实施方案可包括上文所述的各种步骤。这些步骤可体现为可用于致使通用处理器或专用处理器执行这些步骤的机器可执行指令。或者, 这些步骤可由包含用于执行这些步骤的硬连线逻辑的特定硬件组件执行, 或由编程的计算机组件和定制硬件组件的任何组合执行。

[0221] 如本文所述, 指令可指代特定硬件配置, 诸如专用集成电路(ASIC), 其被配置为执行某些特定操作或具有体现在非暂态计算机可读介质中的存储器中所存储的预定功能或软件指令。因此, 可以使用在一个或多个电子装置(例如, 终端站、网络元件等)上存储和执行的代码和数据来实现附图中所示的技术。此类电子装置使用计算机机器可读介质存储和传送(在内部和/或与网络上的其他电子装置)代码和数据, 所述计算机机器可读介质诸如是非暂态计算机机器可读存储介质(例如, 磁盘; 光盘; 随机存取存储器; 只读存储器; 闪存存储装置; 相变存储器)和暂态计算机机器可读通信介质(例如, 电学、光学、声学或其他形式的传播信号—诸如载波、红外信号、数字信号等)。此外, 这类电子装置通常包括连接到一个或多个其他组件的一个或多个处理器的集合, 所述其他组件诸如一个或多个存储装置(非暂态机器可读存储介质)、用户输入/输出装置(例如, 键盘、触摸屏和/或显示器)、以及网络连接。所述处理器集合和其他组件的耦接通常通过一个或多个总线和桥接器(也称为总线控制器)进行。存储装置和承载网络通信量的信号分别表示一个或多个机器可读存储介质和机器可读通信介质。因此, 给定电子装置的存储装置通常将用于执行的代码和/或数据存储在该电子装置的一个或多个处理器的集合上。当然, 可以使用软件、固件和/或硬件的不同组合来实现本发明的实施方案的一个或多个部分。

[0222] 在整个具体实施方式中, 出于解释的目的, 陈述了许多特定细节以便透彻理解本

发明。然而，本领域的技术人员将容易明白，可在没有这些特定细节中的一些的情况下实践本发明。在某些情况下，为避免模糊本发明的主题，未详细描述众所周知的结构和功能。因此，应依据所附权利要求书确定本发明的范围和实质。

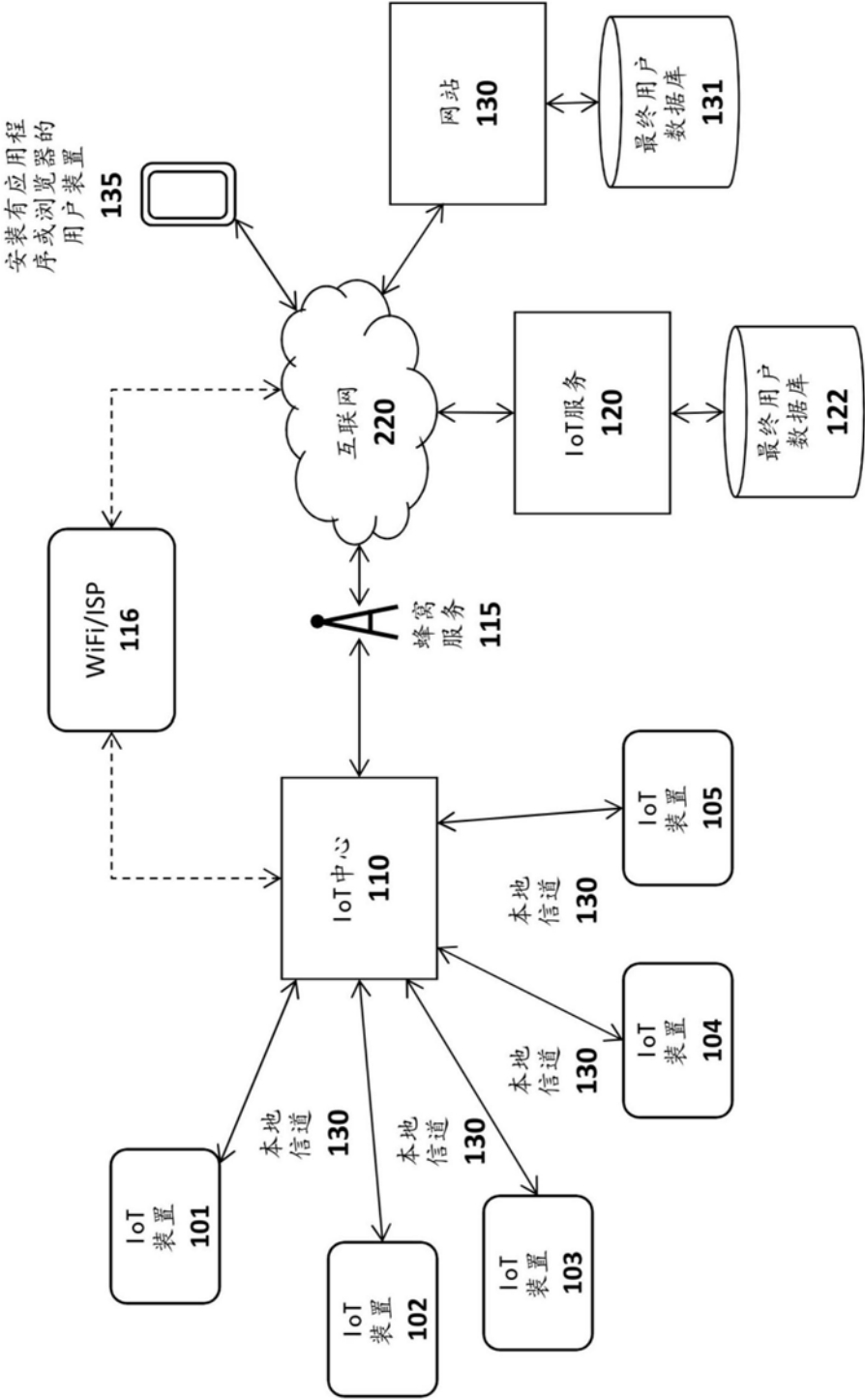


图1A

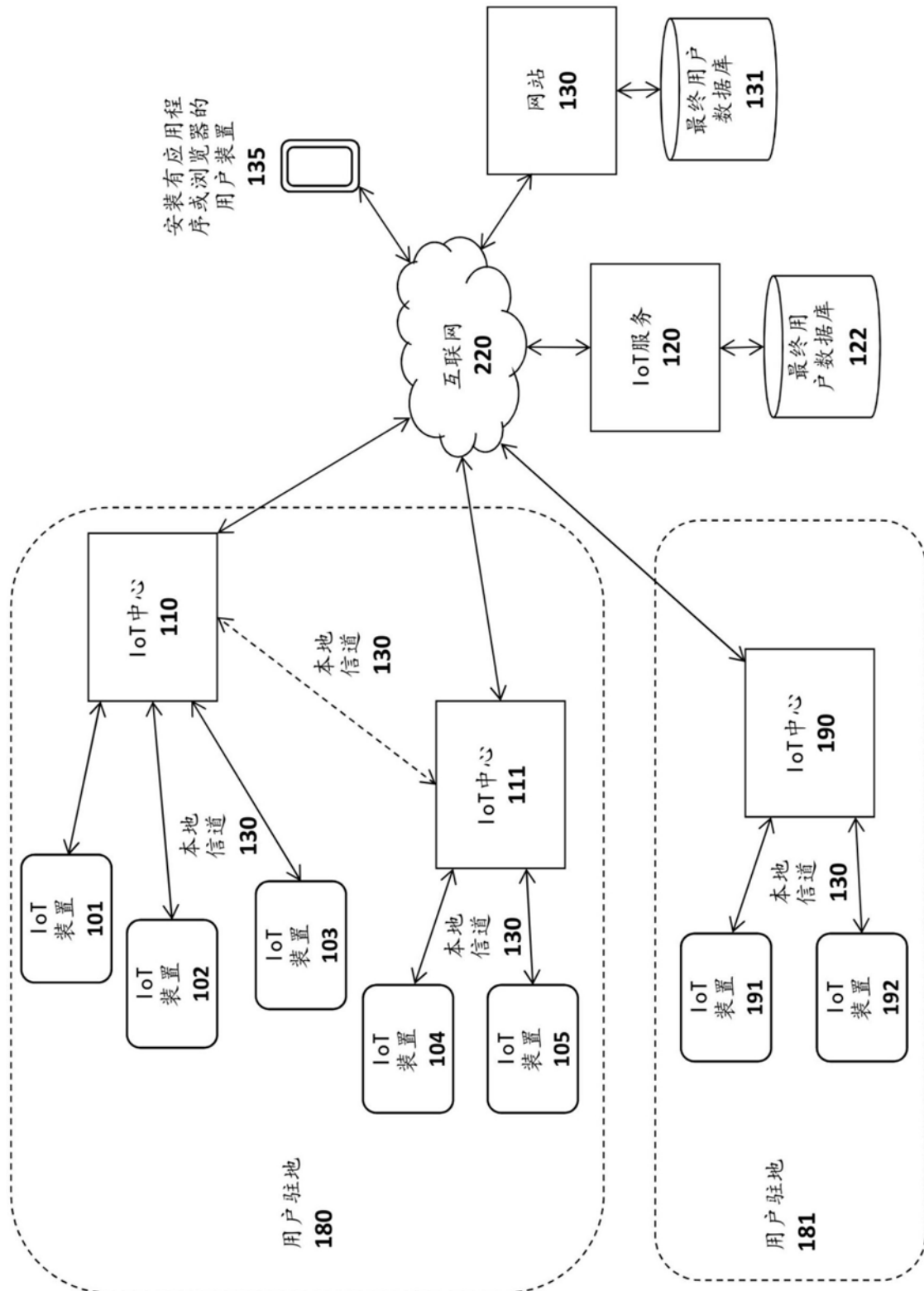


图1B

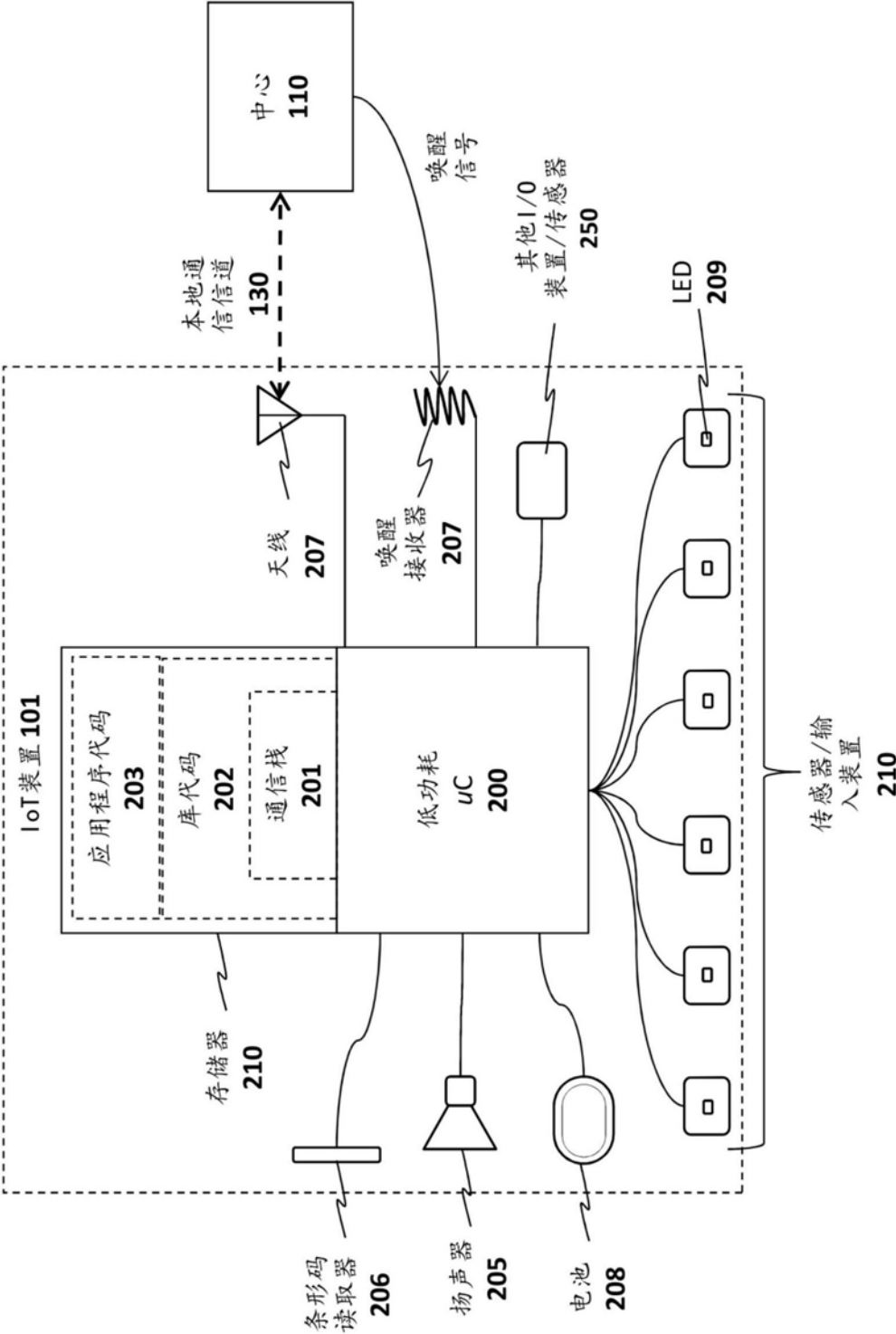


图2

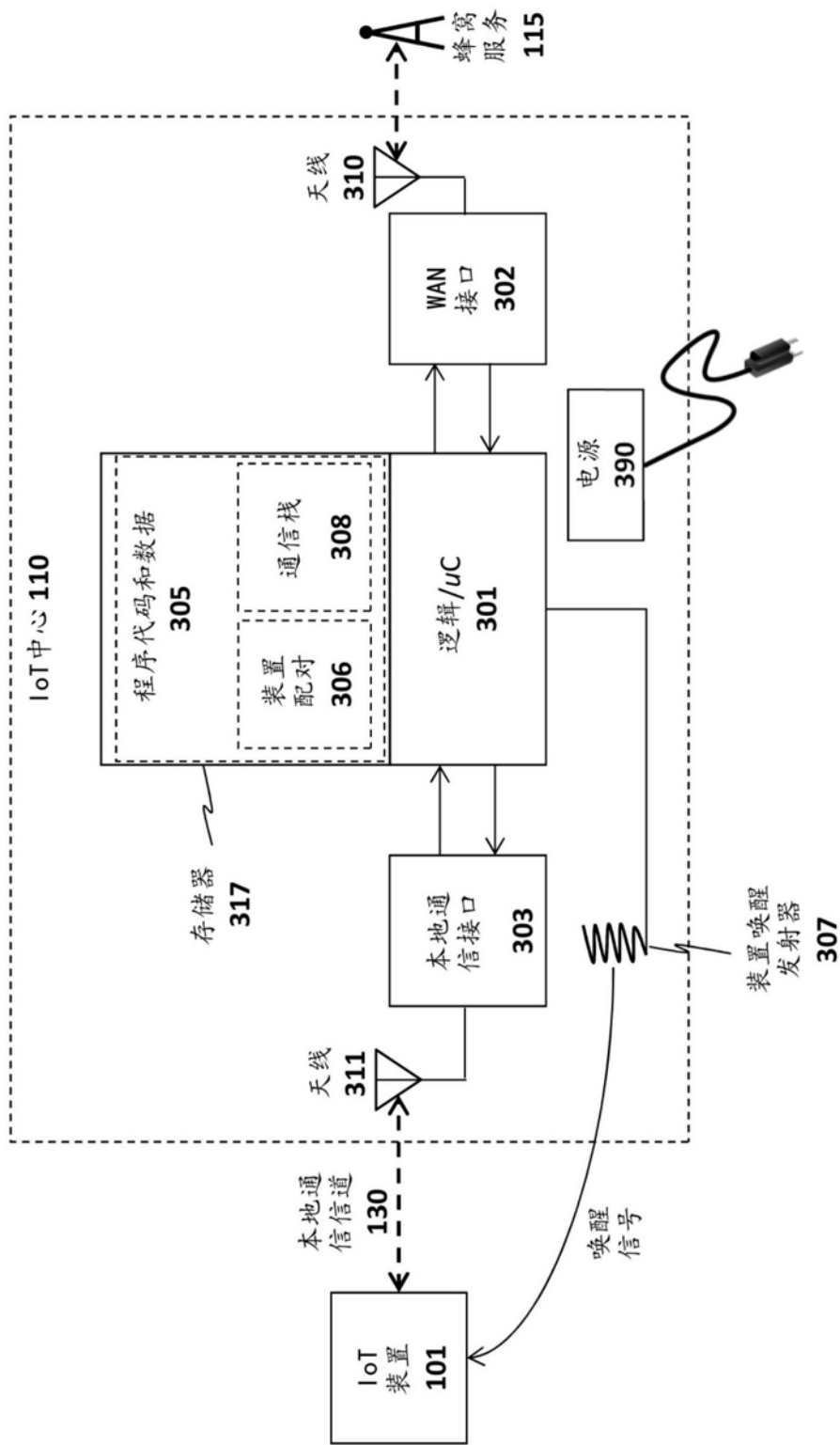


图3

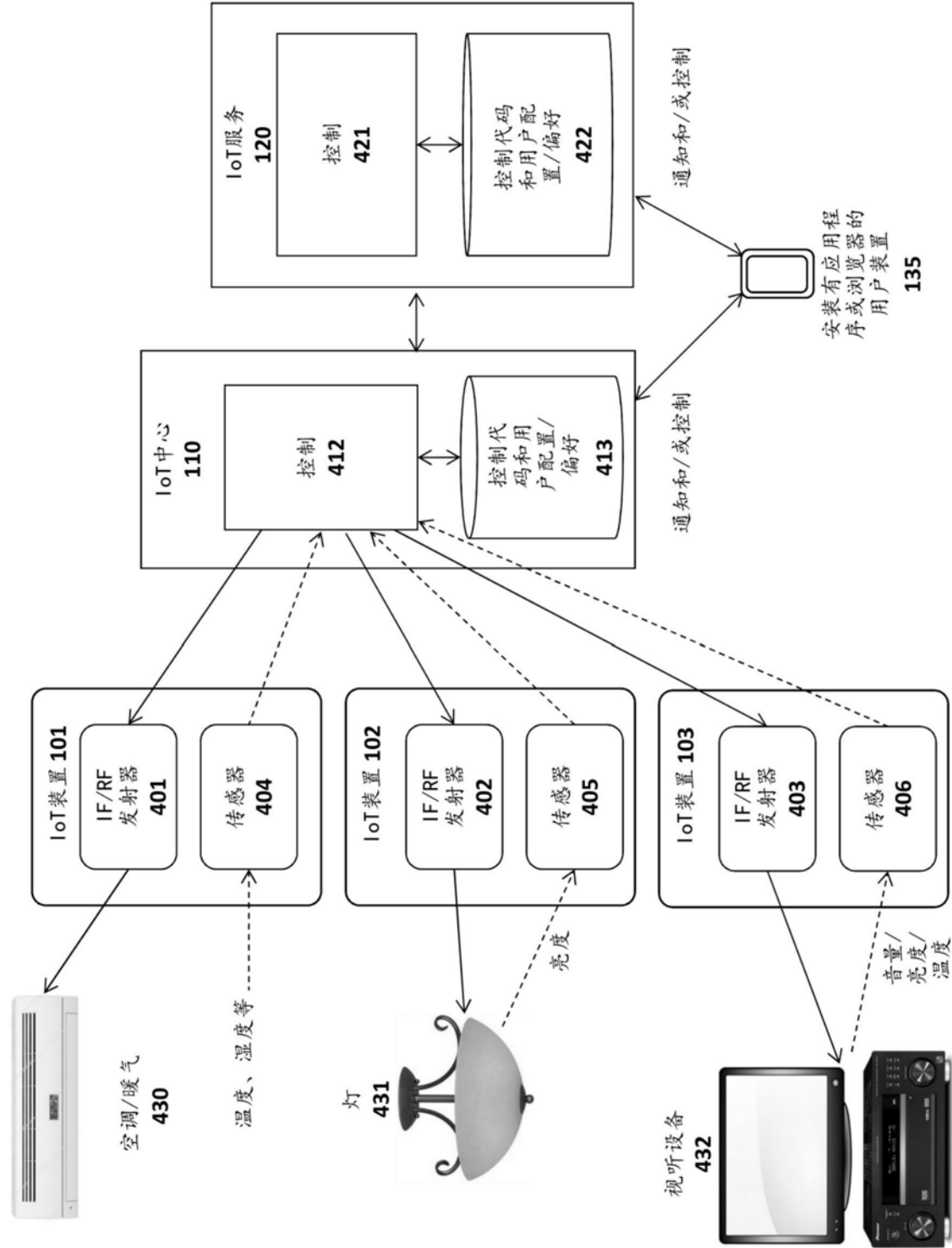


图4A

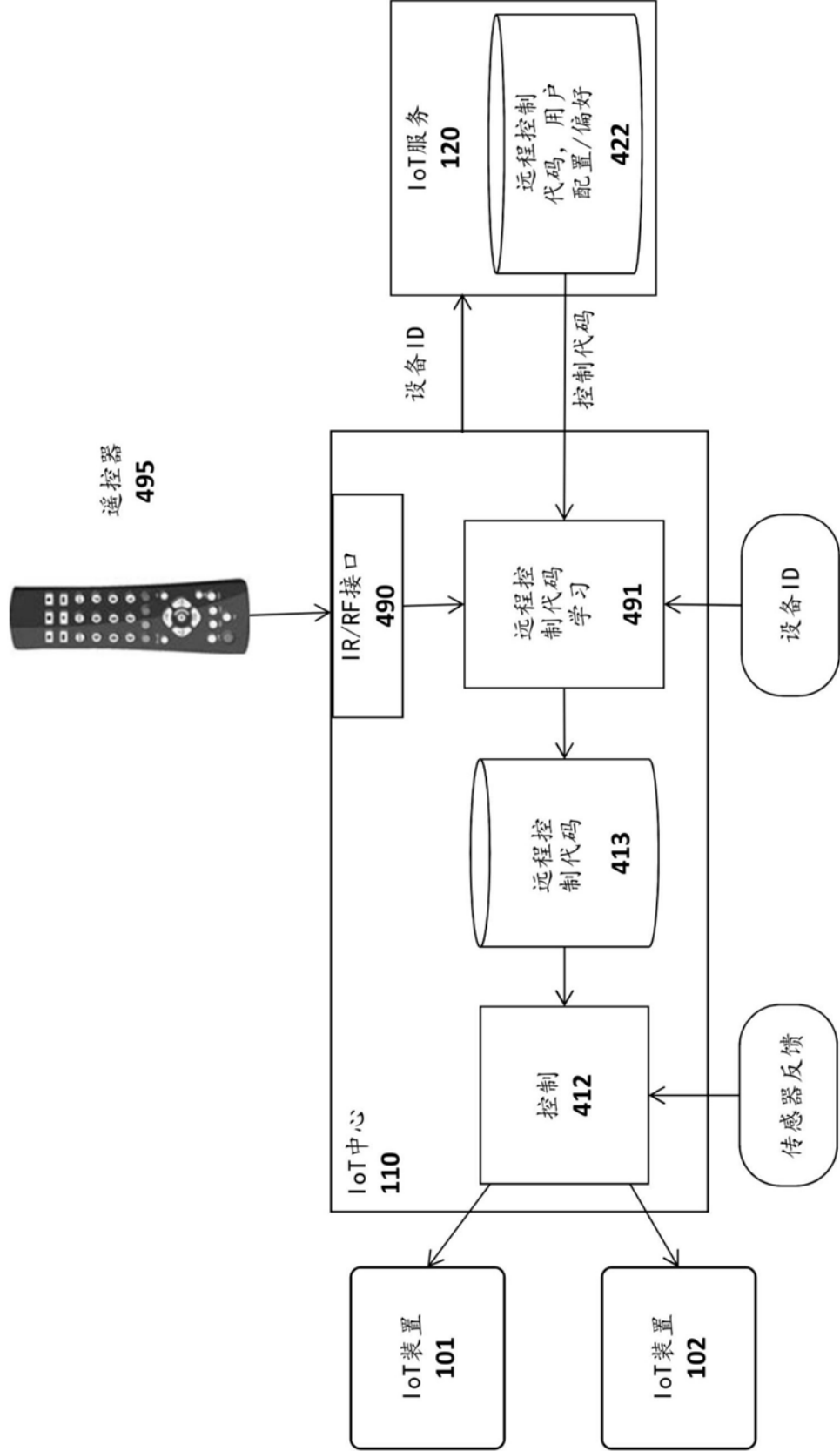


图4B

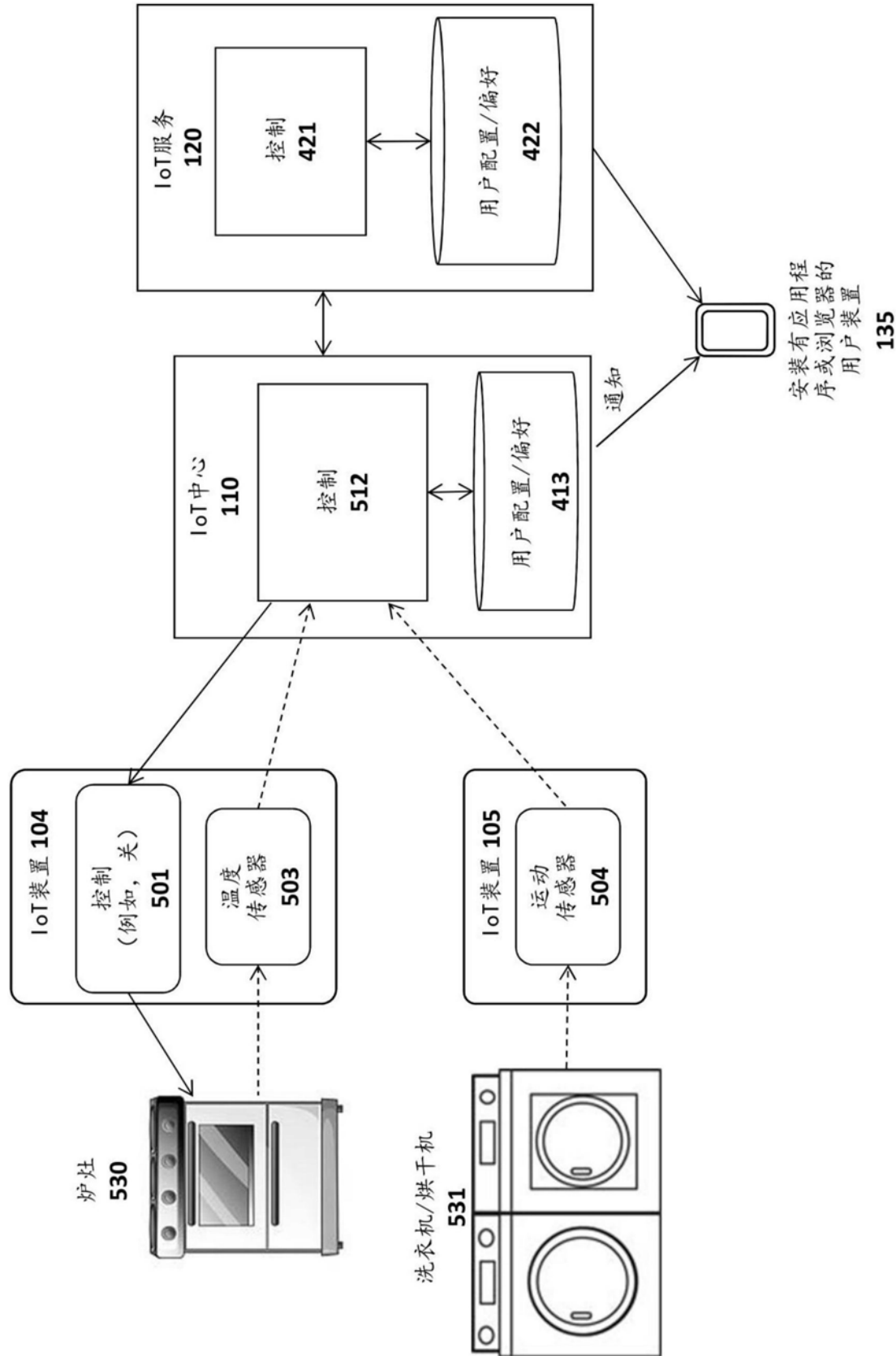


图5

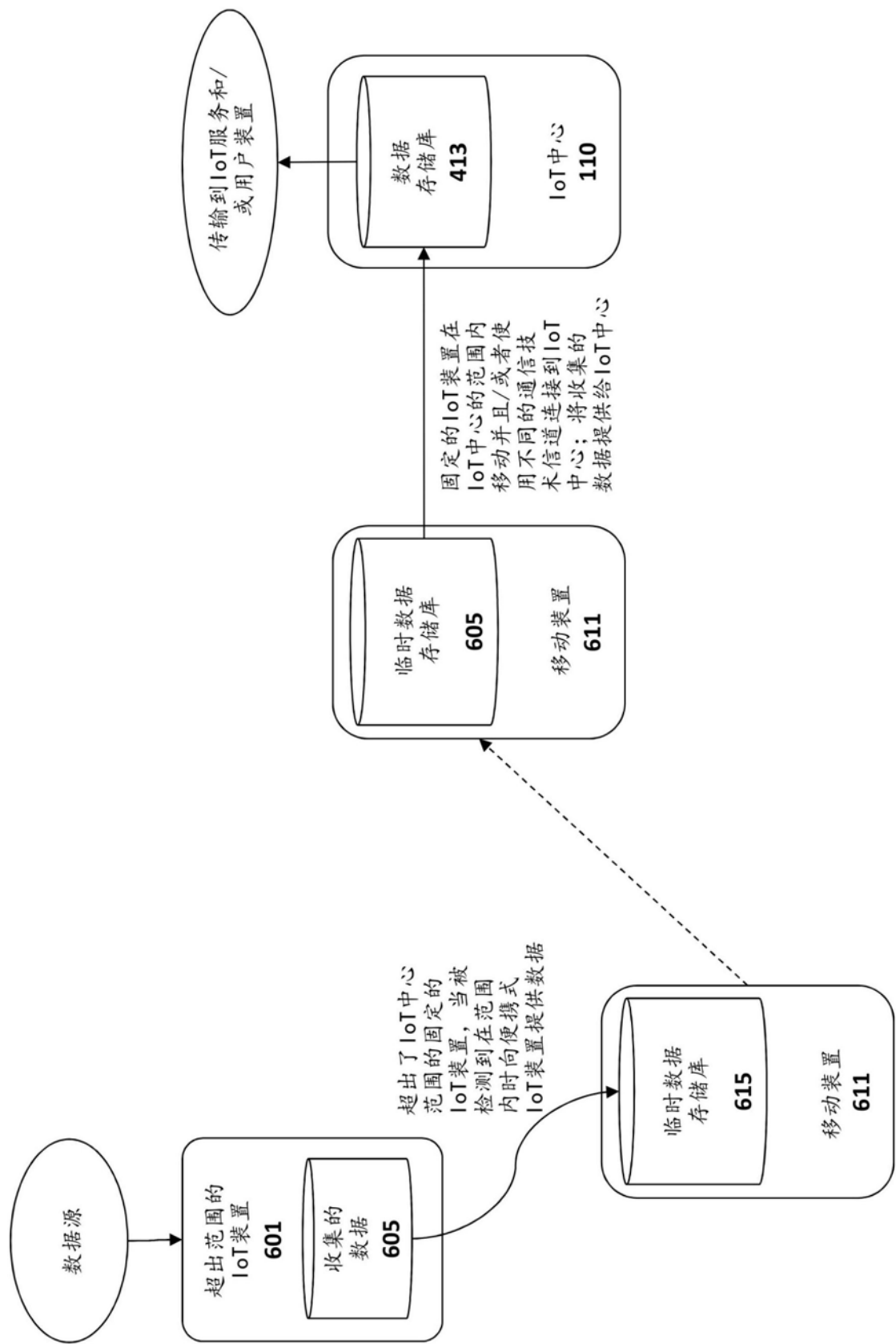


图6

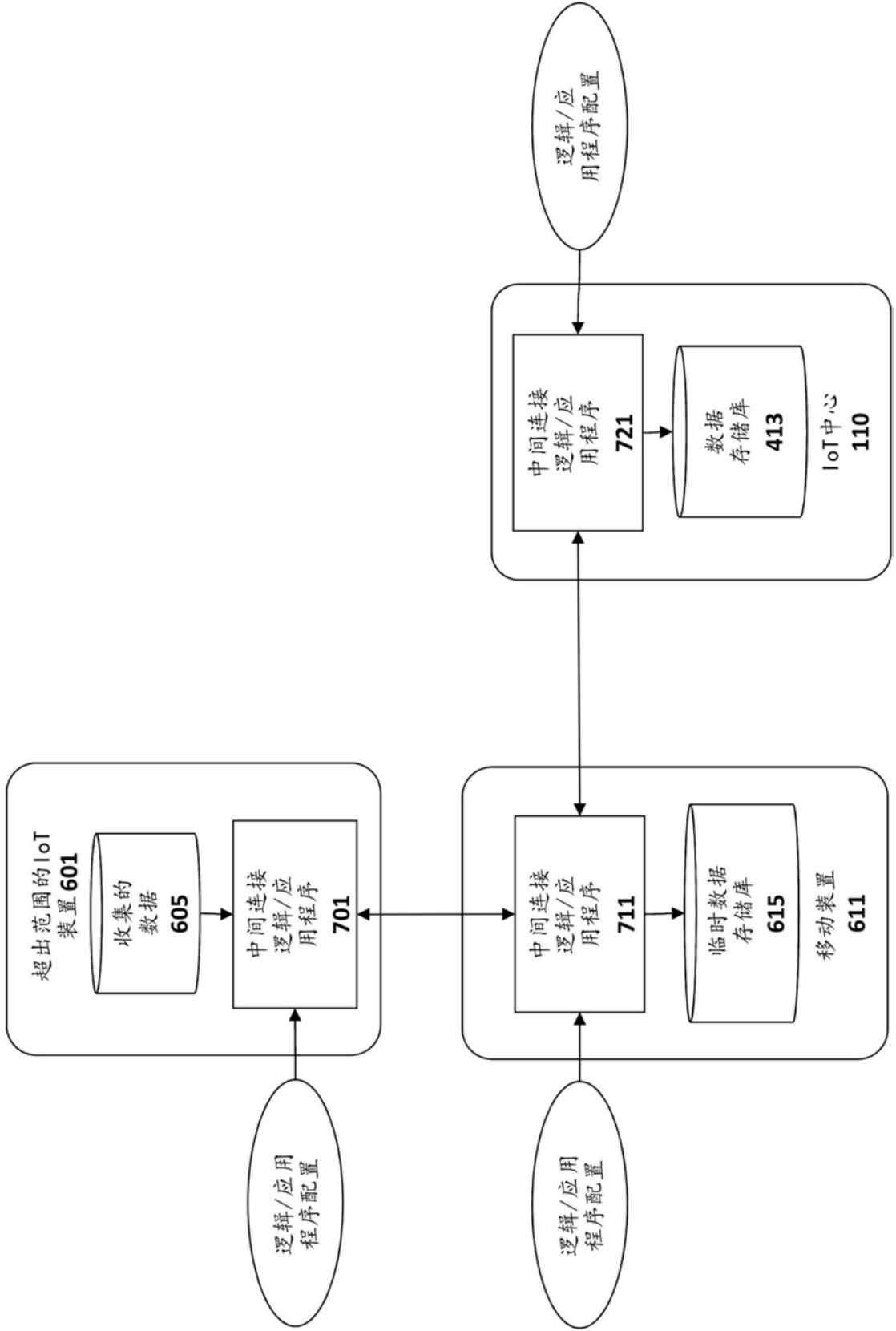


图7

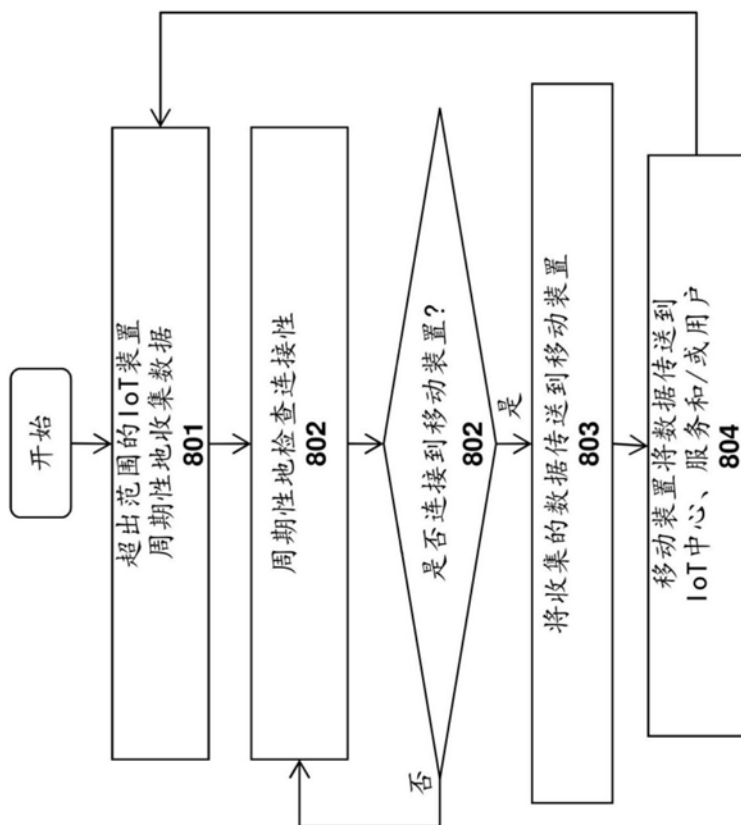


图8

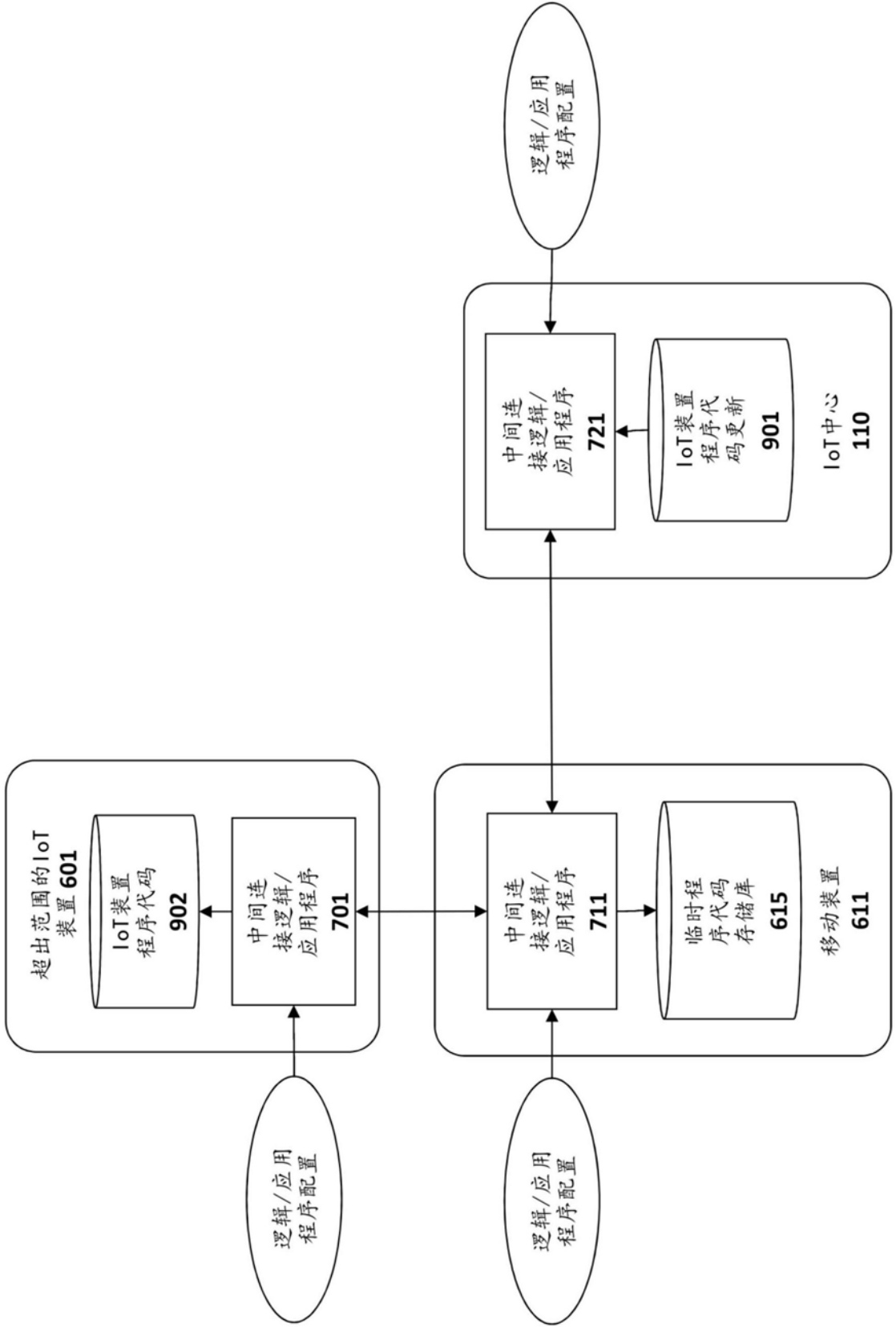


图9A

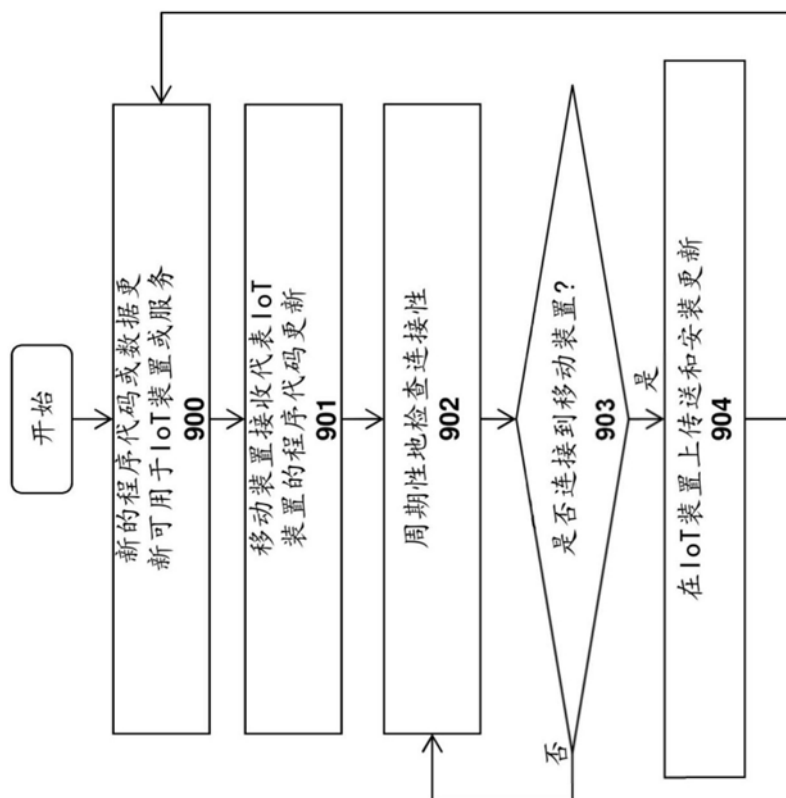


图9B

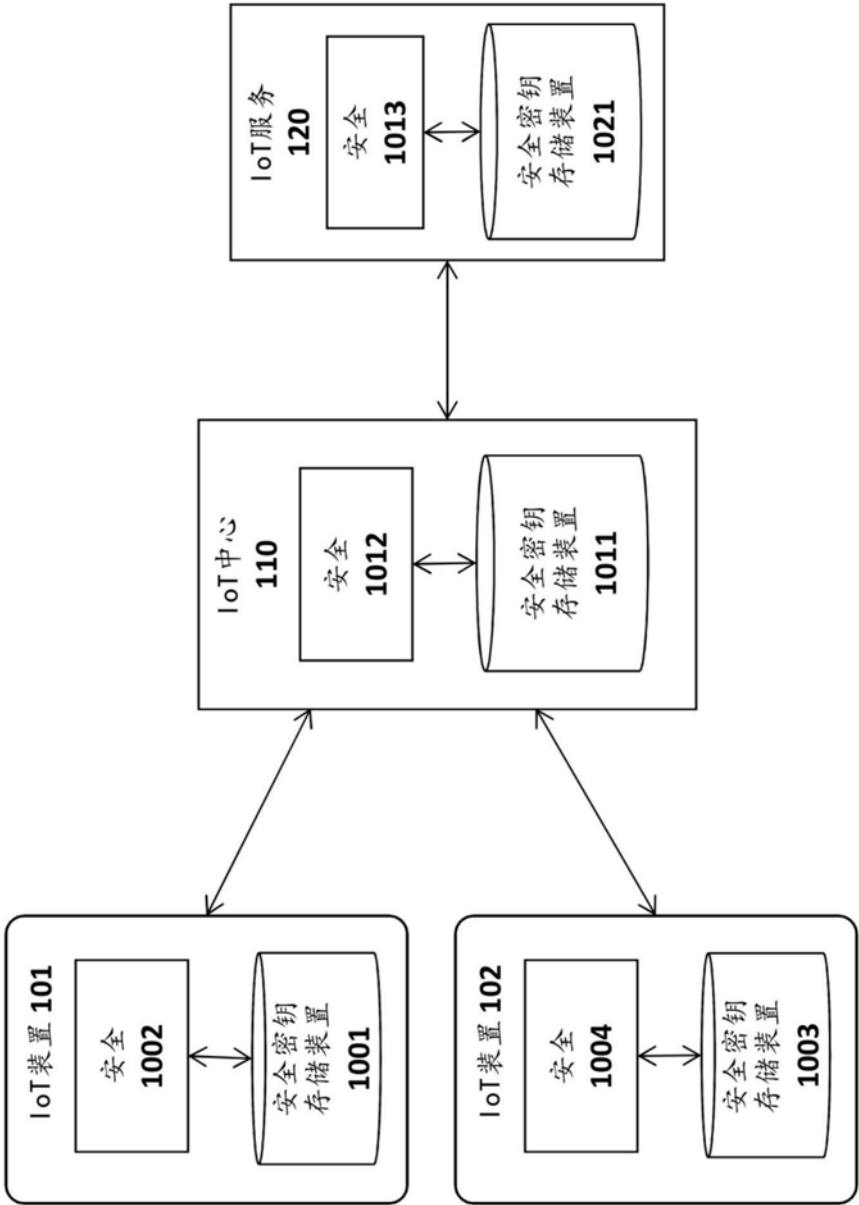


图10

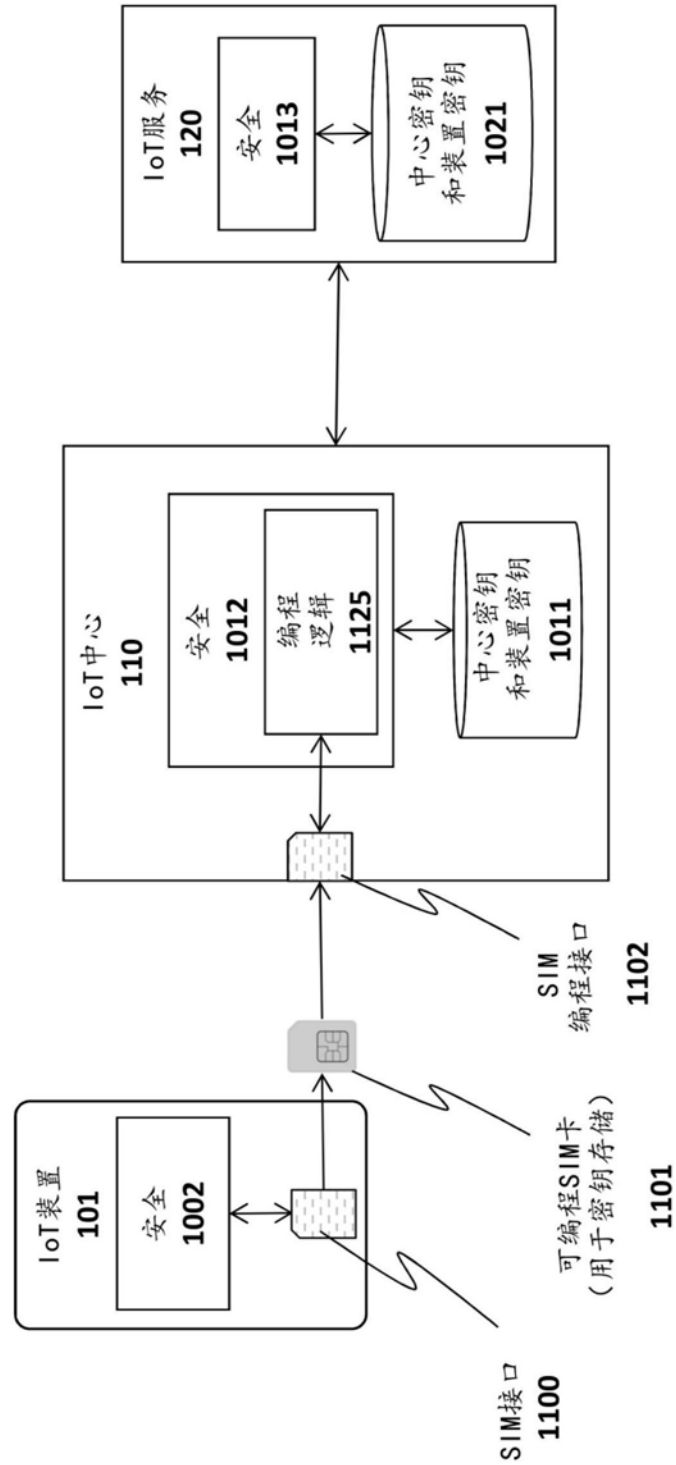


图11

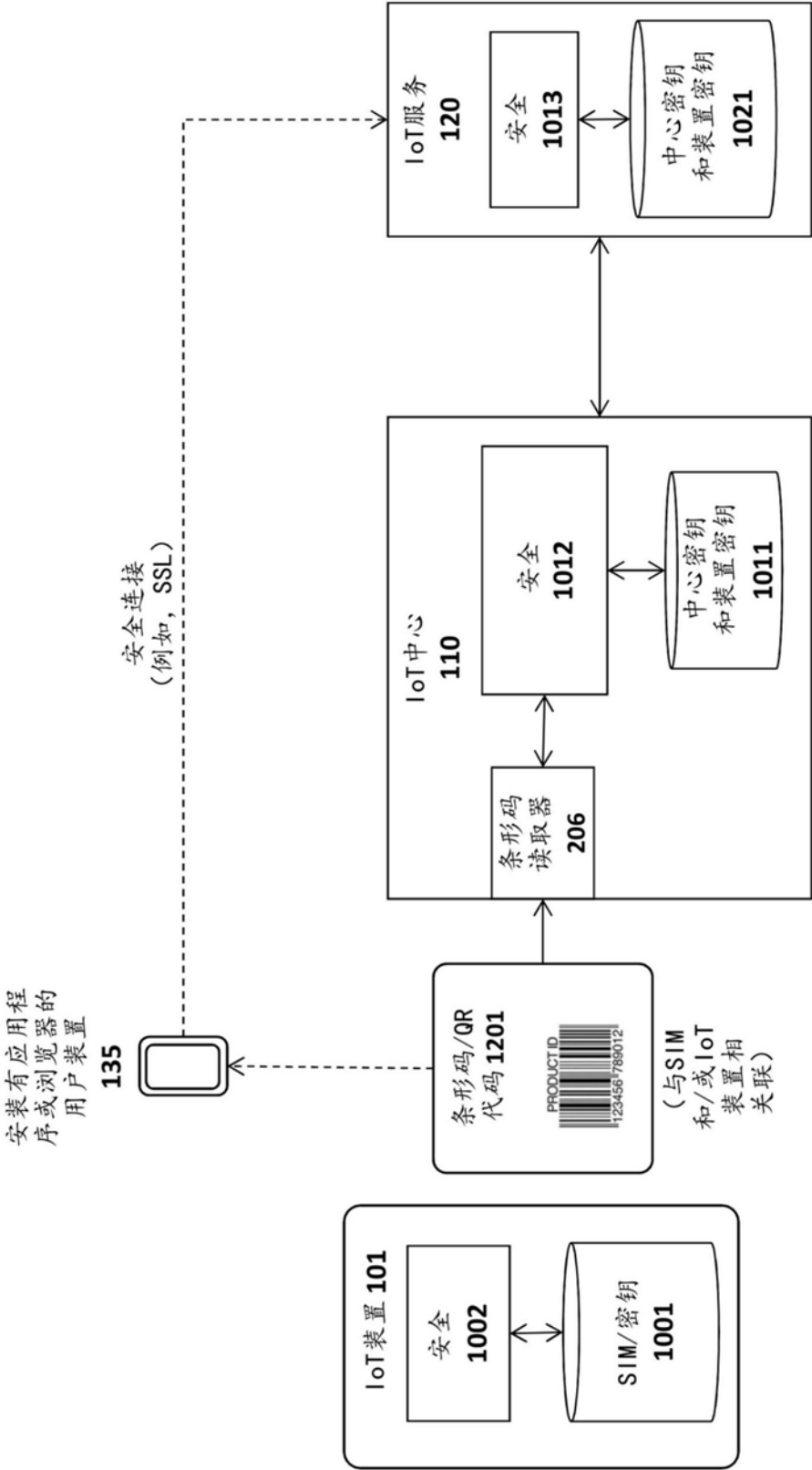


图12A

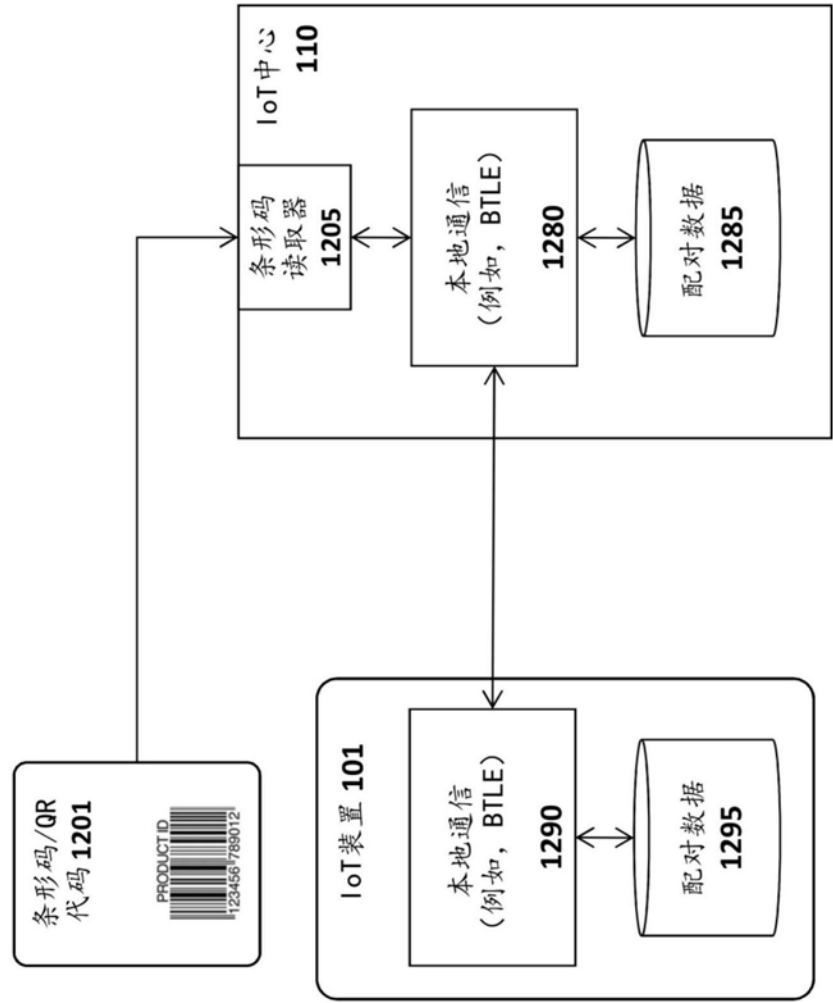


图12B

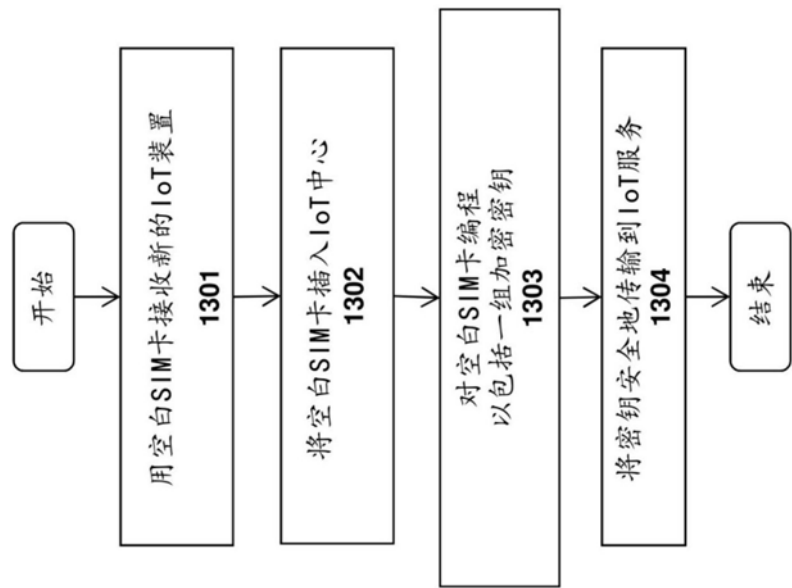


图13

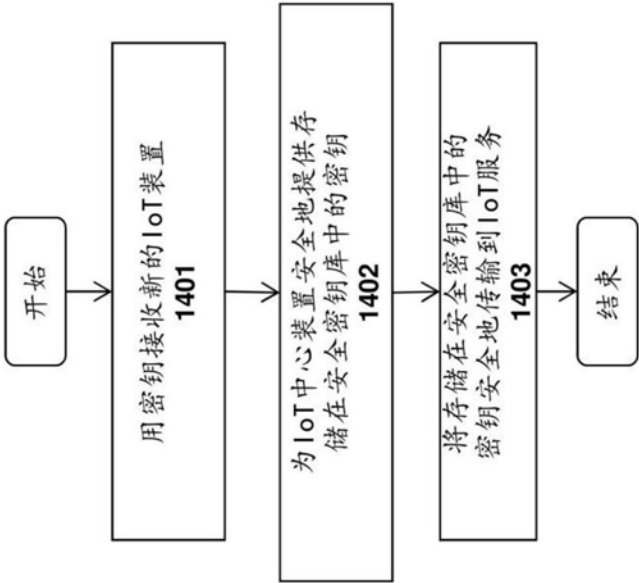


图14

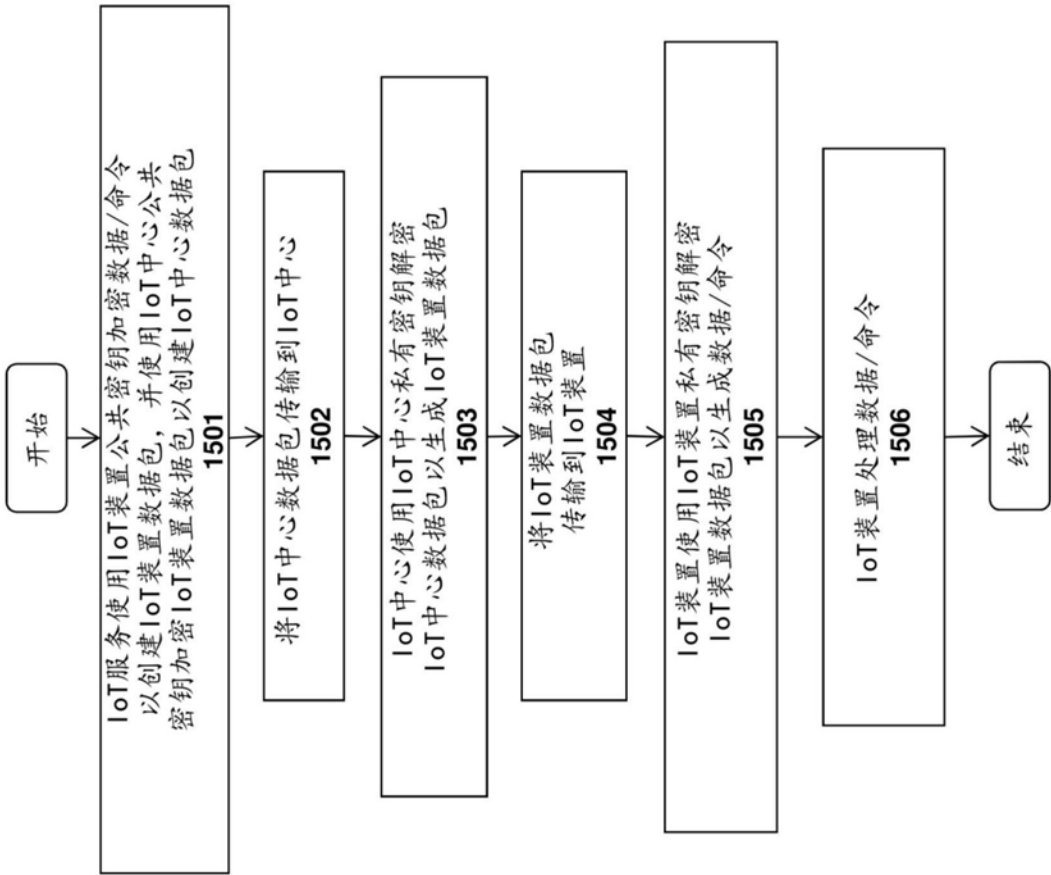


图15

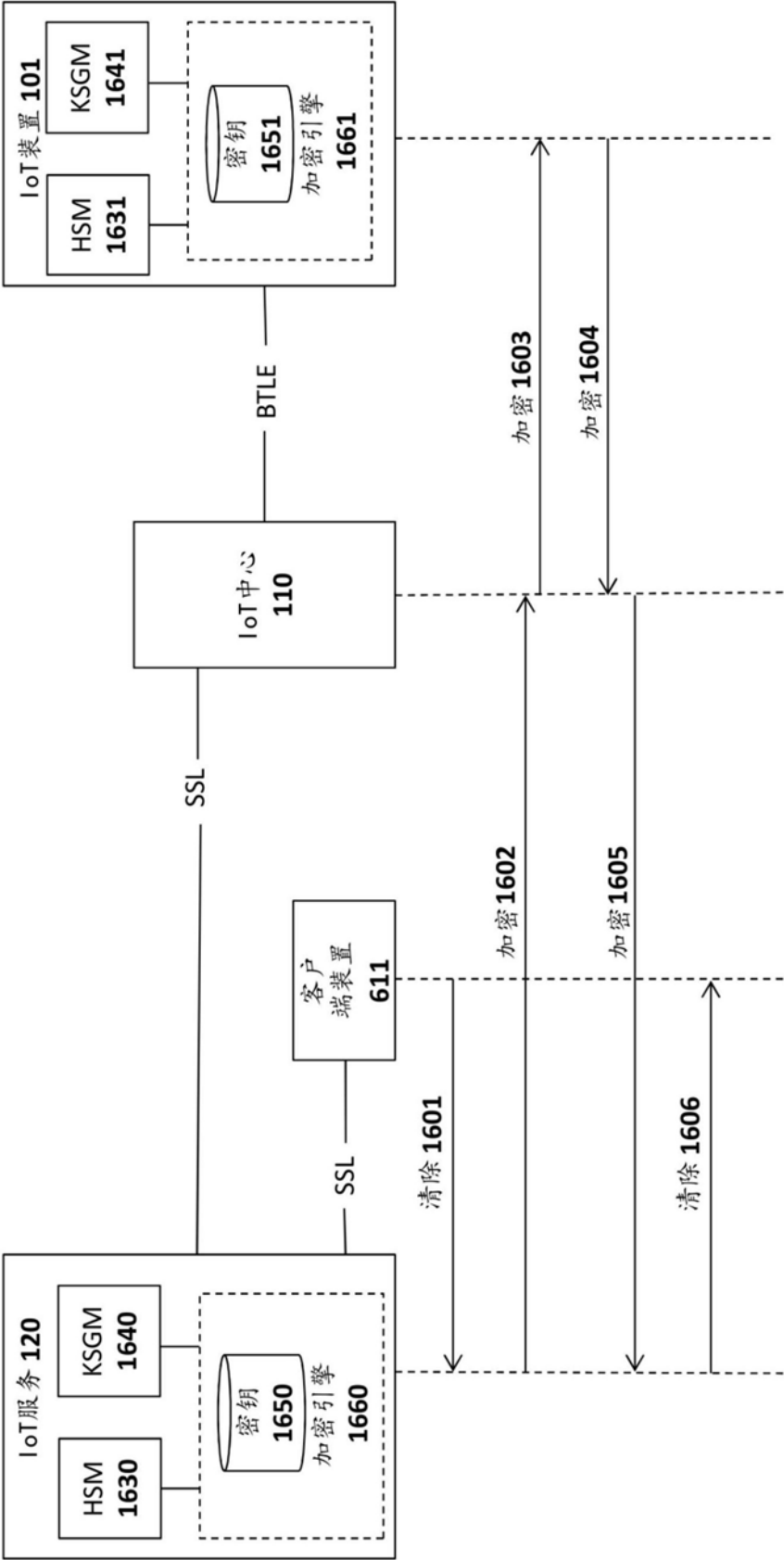


图16A

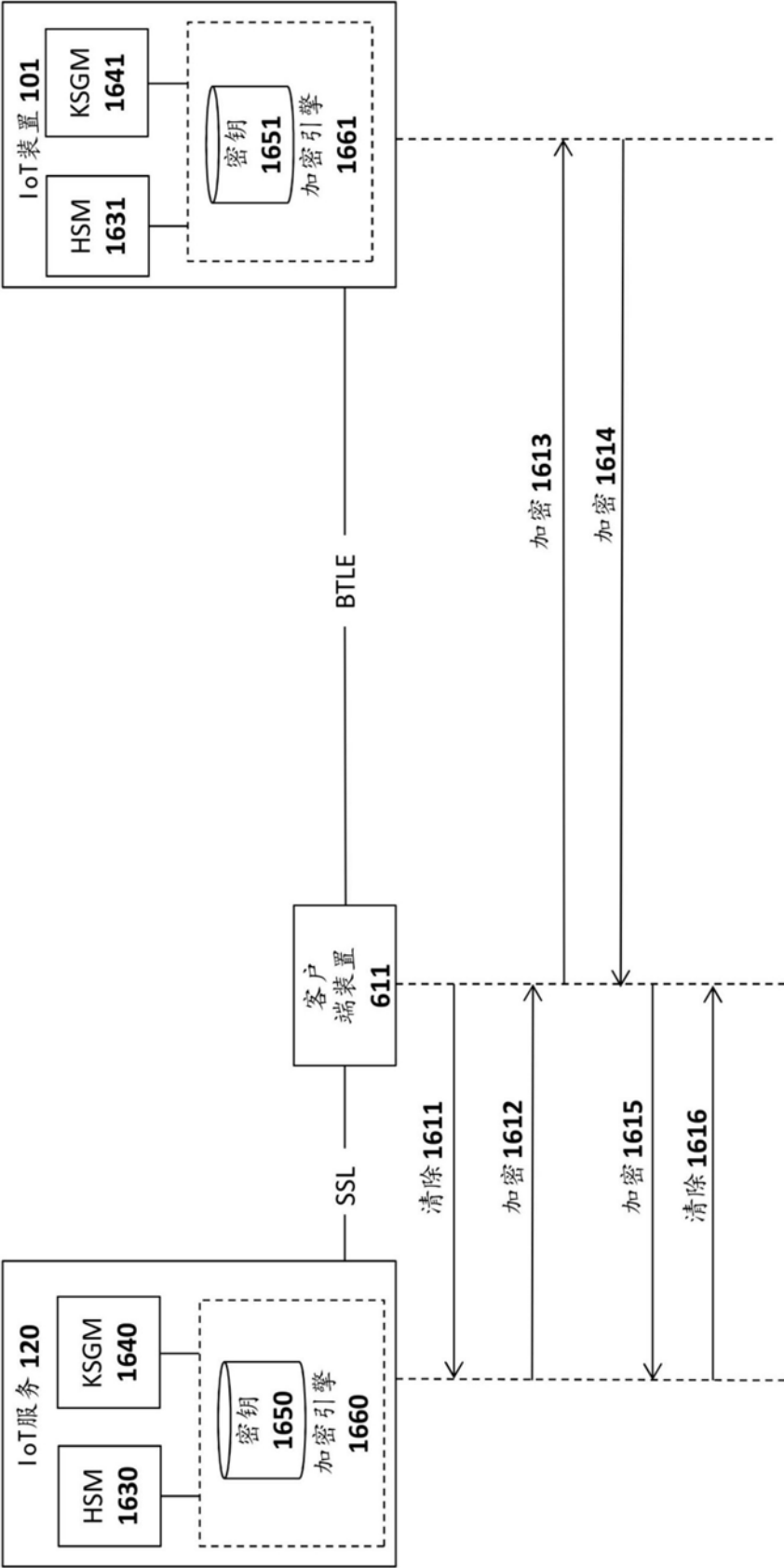


图16B

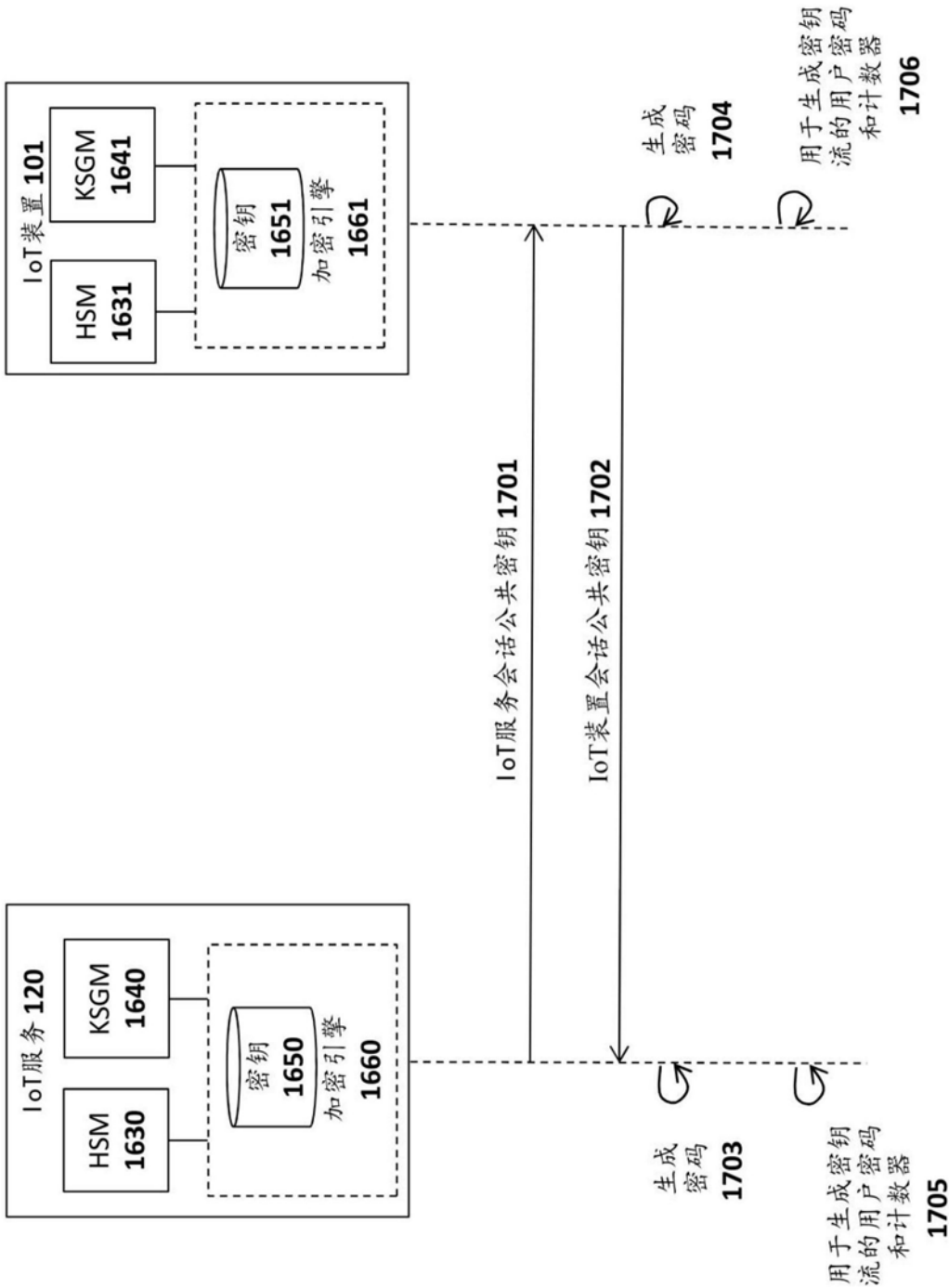


图17

4字节	N字节	6字节
计数器 1800	加密数据 1801	标签 1802

图18

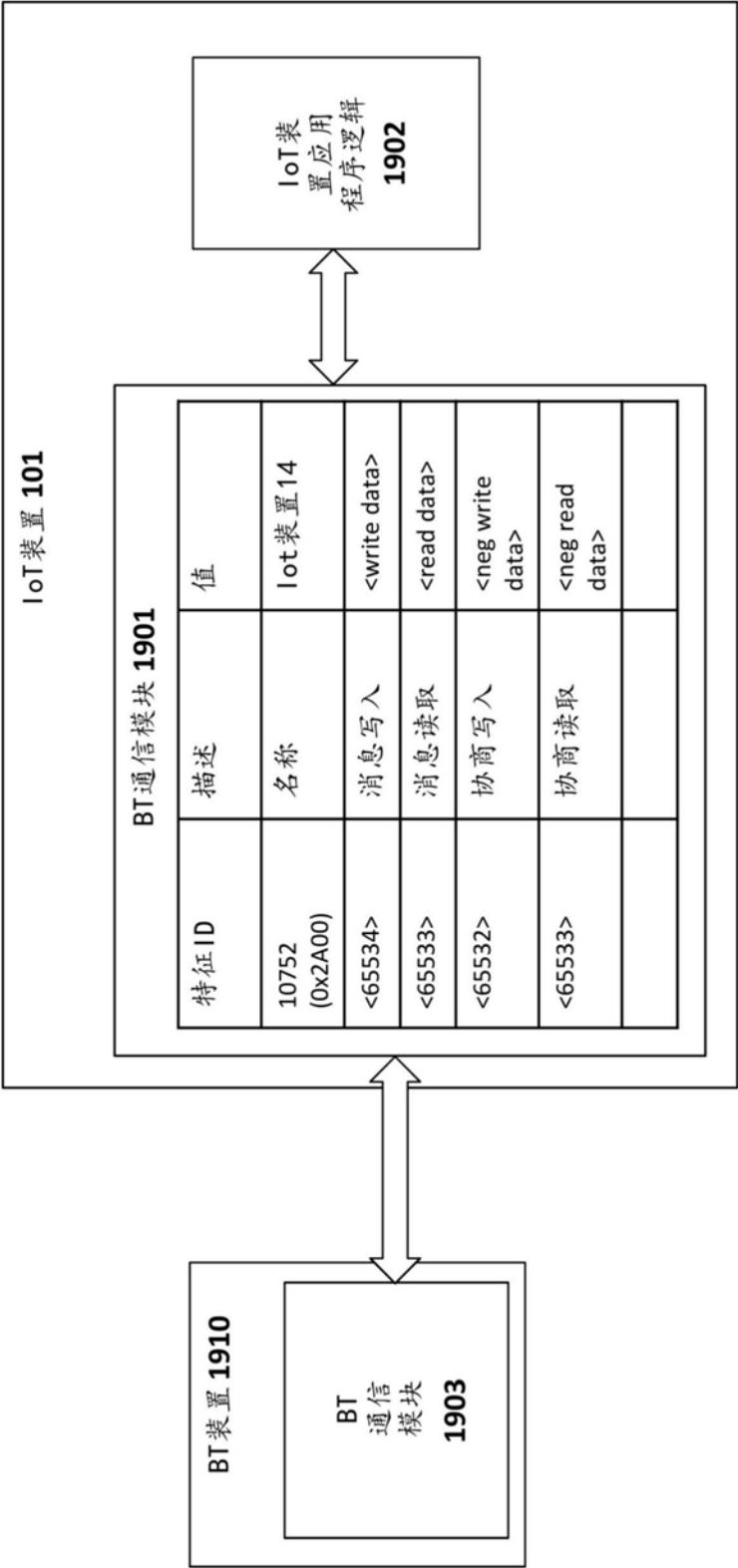


图19

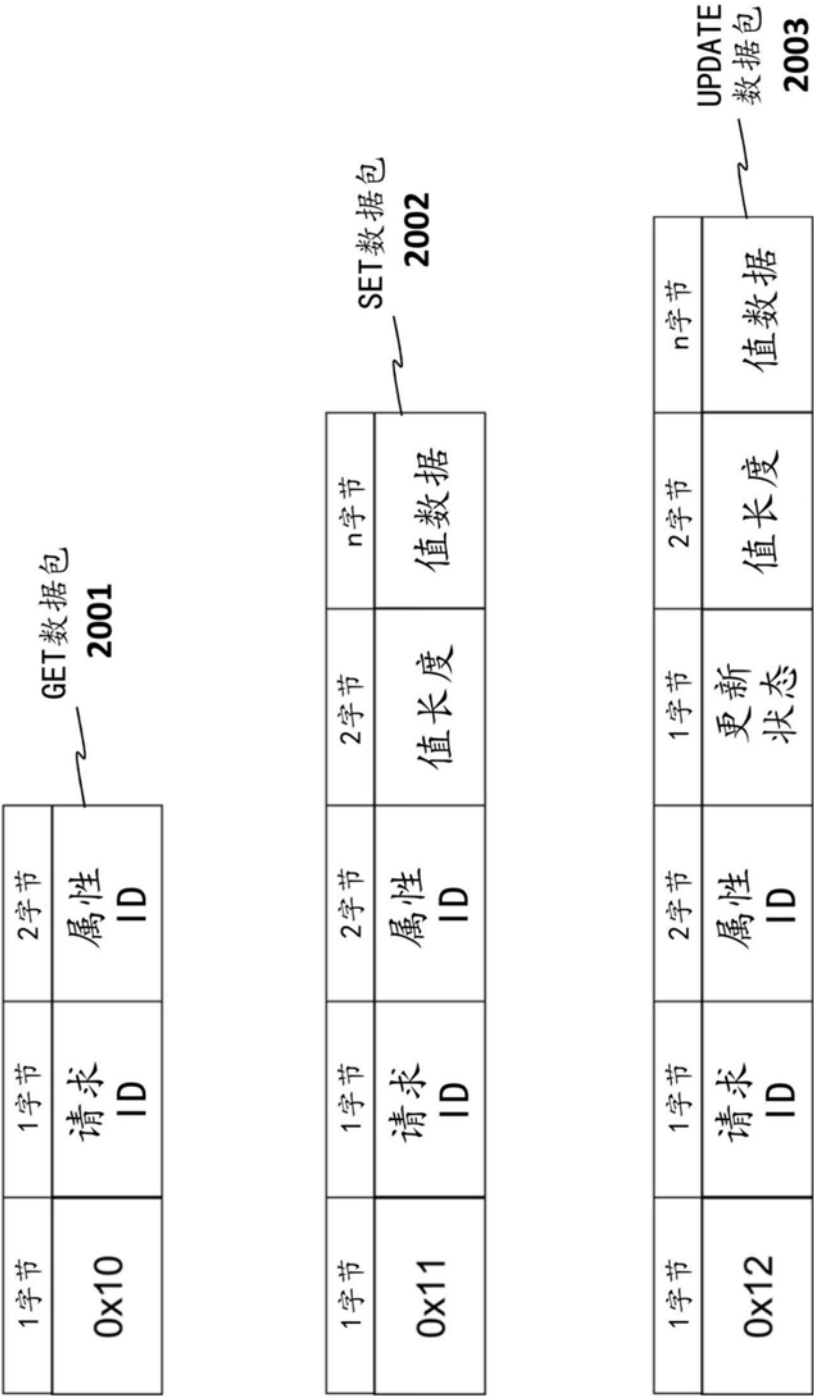


图20

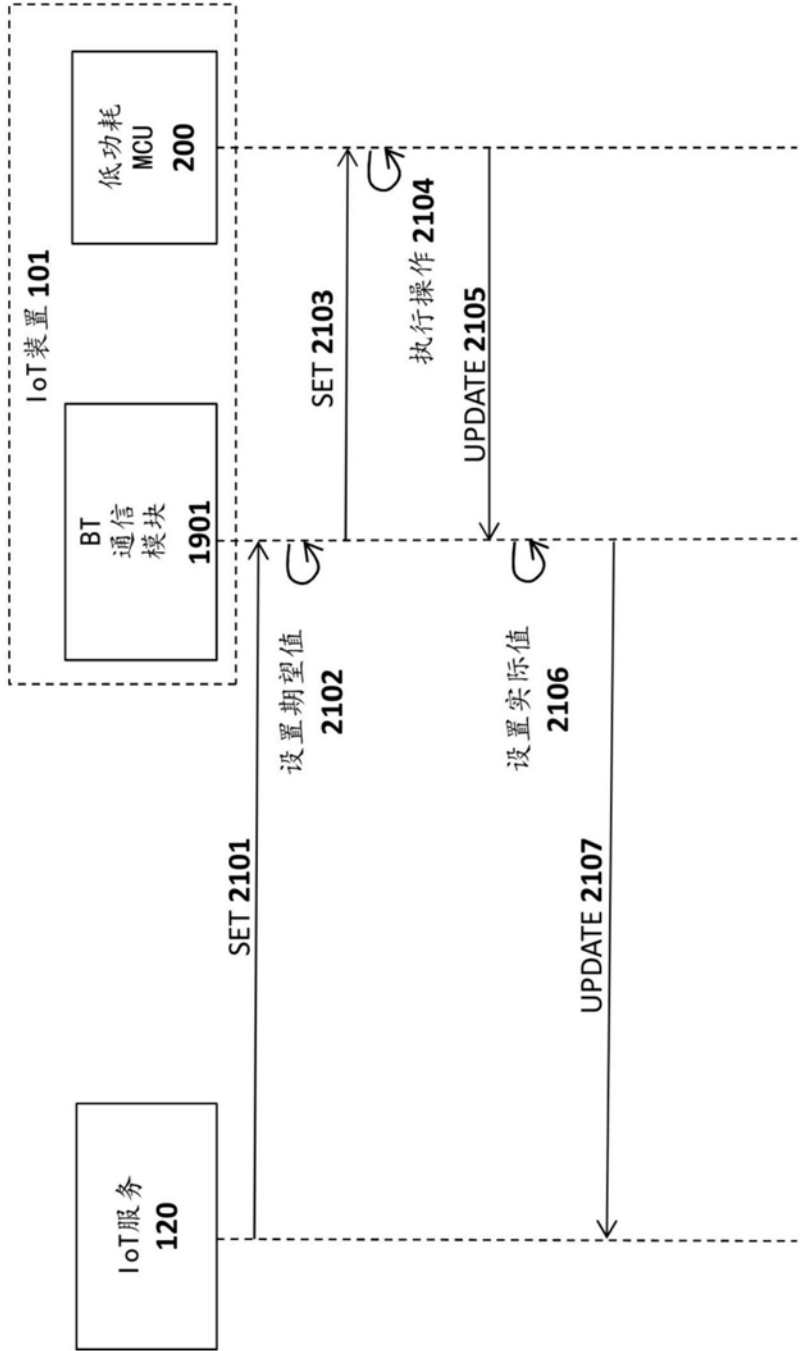


图21

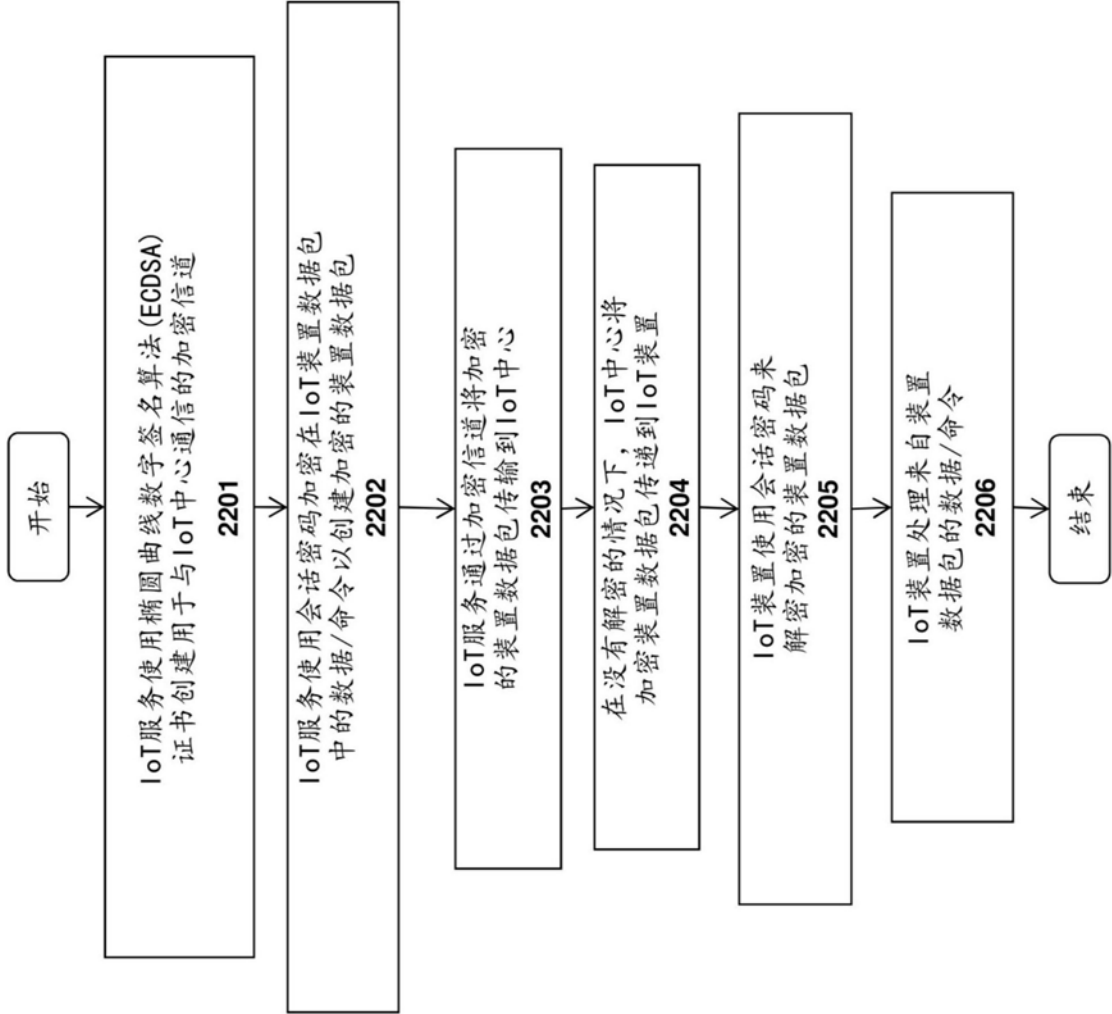


图22

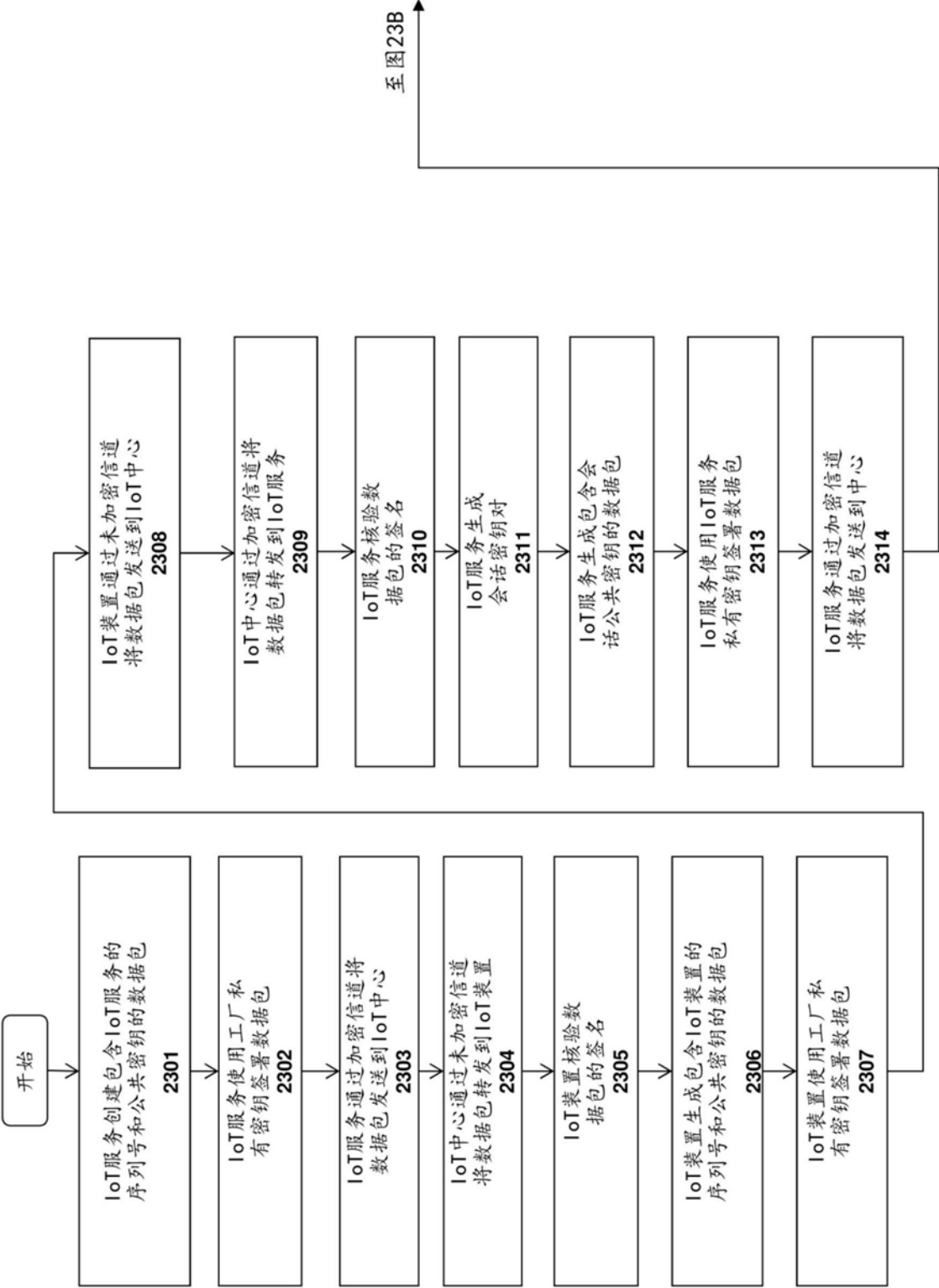


图23A

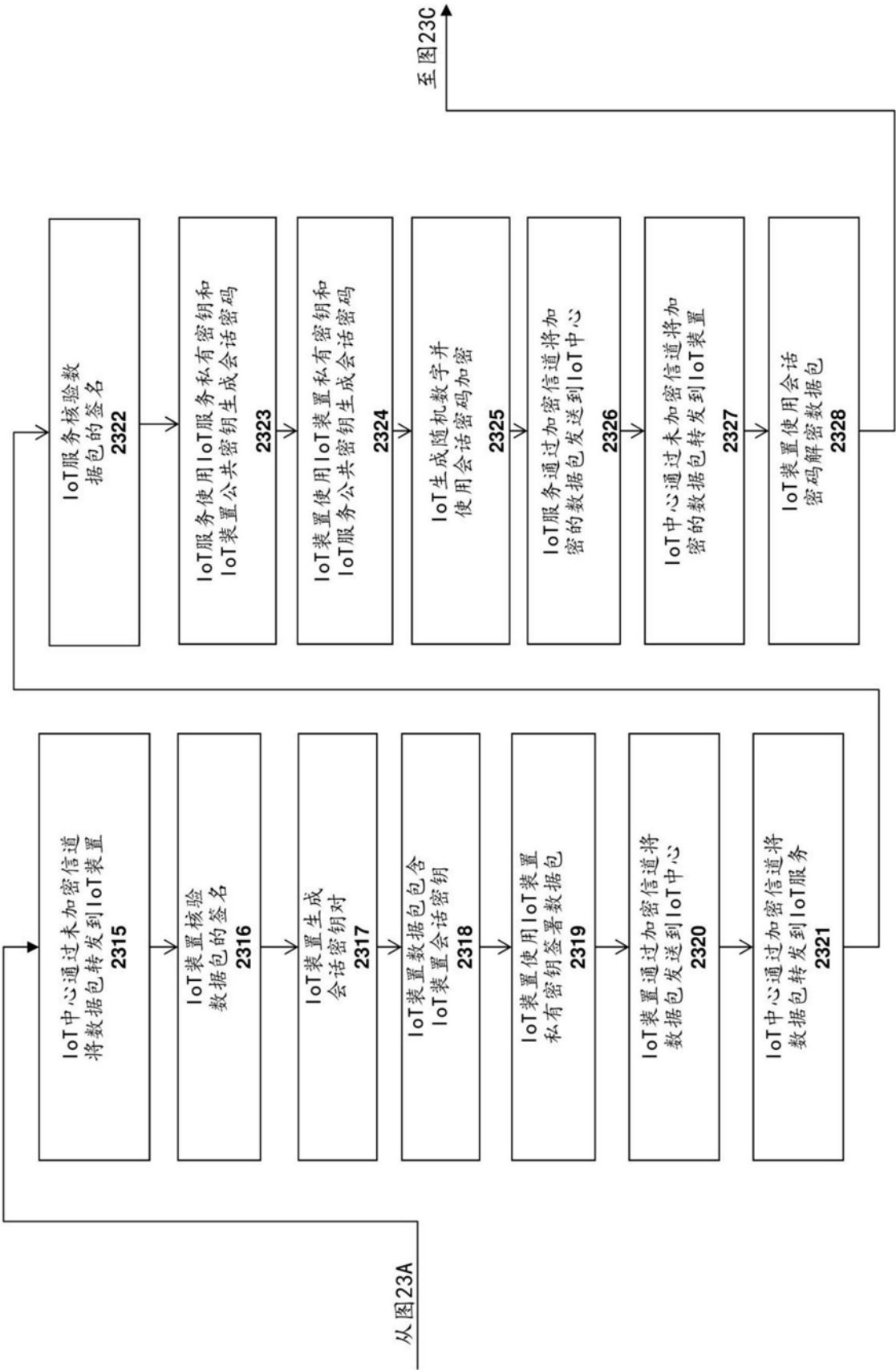


图23B

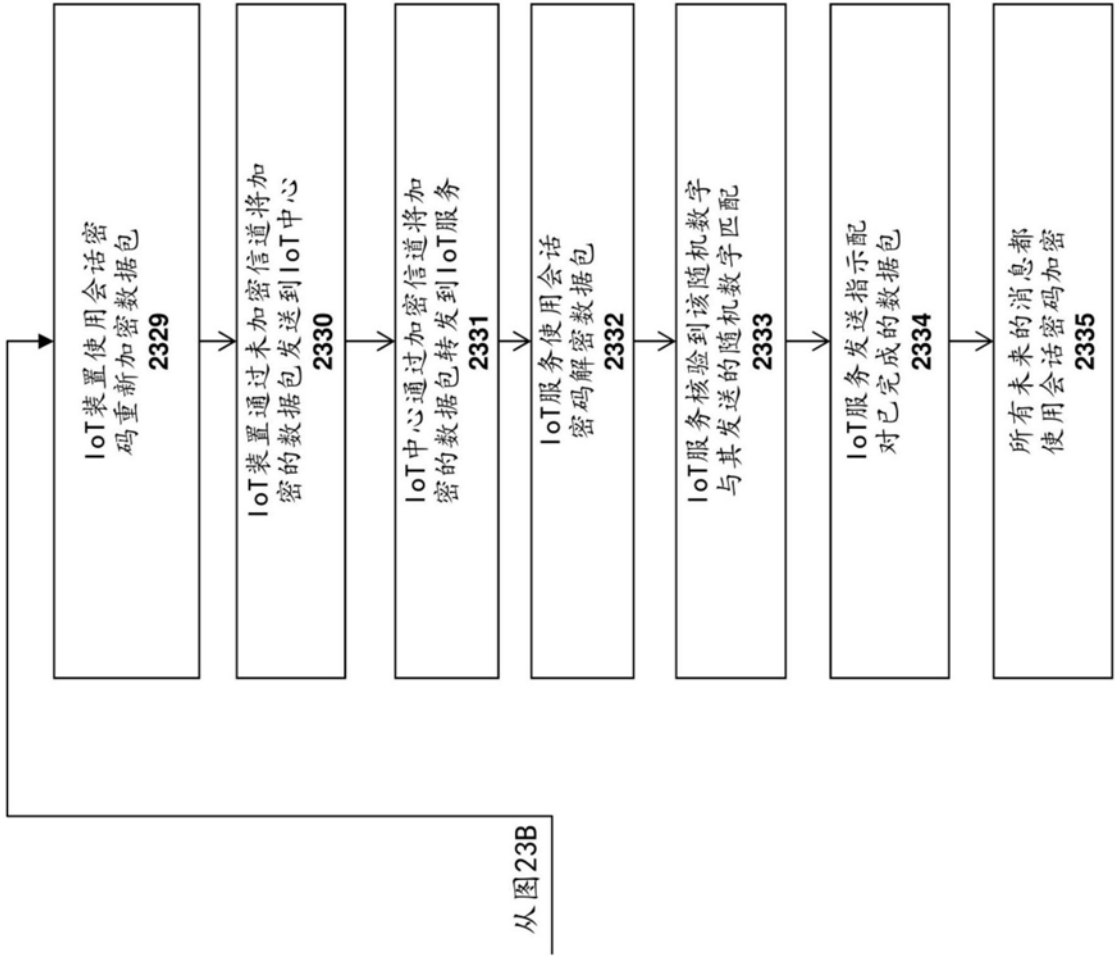


图23C