



(51) International Patent Classification:

G06F 9/54 (2006.01) H04L 9/08 (2006.01)  
H04L 29/06 (2006.01) H04L 9/32 (2006.01)

(21) International Application Number:

PCT/US2019/037124

(22) International Filing Date:

13 June 2019 (13.06.2019)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/684,190 13 June 2018 (13.06.2018) US  
16/156,570 10 October 2018 (10.10.2018) US  
62/860,715 12 June 2019 (12.06.2019) US

(72) Inventors; and

(71) Applicants: **BAKALIS, Konstantinos** [US/US];  
1117-72nd Street, Apt. 3, Brooklyn, New York 11228 (US).  
**BAKALIS, Alexandros** [US/US]; 1117-72nd Street, Apt.  
#1F, Brooklyn, New York 11228 (US).

(74) Agent: **CANNON, Seth**; WEBER ROSSELLI & CANNON LLP, 7 Skyline Dr., Suite 350, Hawthorne, New York 10532 (US).

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

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, 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,

(54) Title: SYSTEMS AND METHODS FOR SECURE ACCESS TO PROPERTY OR INFORMATION USING BLOCKCHAIN

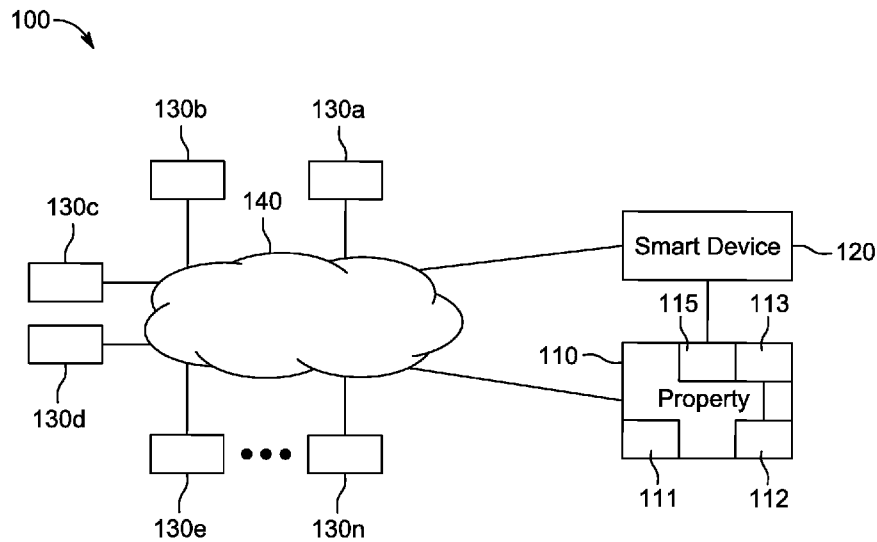


FIG. 1

(57) Abstract: Blockchain-based systems and methods are used to control access to property or information. One system includes a mobile device or other smart device, a key fob or other e-key object, and a server. The mobile device generates an encrypted code and transmits it to the key fob. The key fob transmits the encrypted code to the property and the server updates a log in a hyperledger. The property includes a computing device that validates the encrypted code and grants a key fob user access to the property. Another system includes a server that validates an access key and a first station that transmits an access key to a second station via the server and a satellite. The second station transmits data to the first station via the server and the satellite. The server saves a transmission log in a hyperledger and transmits the access key in response to a request.



MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM,  
TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW,  
KM, ML, MR, NE, SN, TD, TG).

**Published:**

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

## **SYSTEMS AND METHODS FOR SECURE ACCESS TO PROPERTY OR INFORMATION USING BLOCKCHAIN**

### **FIELD**

[0001] This disclosure relates to systems and methods for securing communication data using blockchain. In particular, this disclosure relates to systems and methods for validating communication data prior to communicating the data and for saving communication data in a hyperledger using blockchain.

### **BACKGROUND**

[0002] Data communications are critical in most industries. For decades, many industries utilize satellites for their data communications. Those industries have invested large sums of money and human resources to secure or protect the data communications, especially when those data communications are utilized to control access to various types of property. Many businesses and people have developed and are continuing to develop security systems to protect various types of property from unlawful or unapproved access to or entry into the property, and to protect associated communication data from being hacked or hijacked. Physical locks have been developed in accordance with the needs, requirements, and characteristics of the property. Physical keys corresponding to the physical locks have been likewise developed. However, physical keys are exposed to various risks including the risk of being copied and/or stolen. Further, hacking techniques have been developed to counter advancements in security technology.

### **SUMMARY**

[0003] This disclosure relates to improvements in securing data communications by using blockchain technology so that unauthorized data communication or access can be easily found and tracked, and security against unauthorized and potential breaches can be provided to business or personal properties or in data communication with high reliability.

[0004] A system of one or more computers can be configured to perform particular operations or actions by virtue of having software, firmware, hardware, or a combination of them installed on the system that in operation causes or cause the system to perform the actions. One or more computer programs can be configured to perform particular operations

or actions by virtue of including instructions that, when executed by data processing apparatus, cause the apparatus to perform the actions.

**[0005]** One general aspect includes an e-key system. The e-key system includes: a mobile device configured to generate an encrypted code; a key fob configured to receive the encrypted code from the mobile device and to transmit the encrypted code to a computing device incorporated into property via wireless communication; and a server configured to update an access log of the key fob in a hyperledger. The computing device includes a validation module configured to validate the received encrypted code, and the computing device grants a user of the key fob access to the property when the validation module validates the encrypted code. Other aspects of this aspect include corresponding computer systems, apparatus, and computer programs recorded on one or more computer storage devices, each configured to perform the actions of the e-key system.

**[0006]** Implementations may include one or more of the following features. The wireless communication may be Bluetooth communication. The mobile device and the key fob may be paired according to a Bluetooth protocol. The computing device of the property and the key fob may be paired according to a Bluetooth protocol. The validation module may be constantly powered by the property. The validation module may be an electronic circuit. The code may be encrypted with a public key by the mobile device. The validation module may decrypt the encrypted code with a private key, which is saved in the validation module, corresponding to the public key. The validation module may transmit the access log to the server when a network connection is established with the server. The property may be an aircraft, watercraft, hovering vehicle, land vehicle, or building.

**[0007]** Another general aspect includes a method for granting access to property to a user of a key fob. The method includes: transmitting, by a mobile device, an encrypted code to a key fob; transmitting, by the key fob, the encrypted code to a validation module running on a computing device of property via wireless communication; determining, by the validation module, whether the encrypted code is valid; granting access to the property when it is determined that the encrypted code is valid; and denying access to the property when it is determined that the encrypted code is invalid. Other aspects of this aspect include corresponding computer systems, apparatus, and computer programs recorded on one or more computer storage devices, each configured to perform the actions of the method.

[0008] Implementations may include one or more of the following features. The wireless communication may be Bluetooth. The property and the key fob may be paired according to a Bluetooth protocol. The mobile device and the key fob may be paired according to a Bluetooth protocol. The encrypted code may be encrypted by a public key. The method may further include decrypting, by the validation module, the encrypted code with a private key, which is saved in the validation module, corresponding to the public key. The method may further include transmitting, by the validation module, an access log to a server when a network connection is established with the server. The property may be an aircraft, watercraft, hovering vehicle, land vehicle, or building. Implementations of the described techniques may include hardware, a method or process, or computer software stored on a computer-accessible medium.

[0009] Another general aspect includes a satellite data communication system, which includes: a server configured to validate an access key; a first station configured to transmit an access key via the server and a satellite; and a second station configured to receive the access key and to transmit data to the first station via the server and the satellite when the server validates the access key. The server saves a transmission log in a hyperledger using blockchain and the server transmits the access key in response to a request by the first station. Other aspects of this aspect include corresponding computer systems, apparatus, and computer programs recorded on one or more computer storage devices, each configured to perform the actions of the satellite data communication system.

[0010] Implementations may include one or more of the following features. The transmission log may include a validation history of the access key. The satellite may not be permitted to transmit the data from the second station to the first station when the access key is not validated. The satellite may be permitted to transmit the data from the second station to the first station when the access key is validated. The second station may transmit the access key to the server when the second station receives the access key from the first station. The access key may be encrypted by the server. The server may be further configured to encrypt data before transmitting data to the satellite. The server may be further configured to save the encrypted data in a hyperledger. The second station may be further configured to encrypt the data using a public key. The first station may be further configured to decrypt the data using a private key corresponding to the public key. The first and second stations may be military stations. The access key may be valid for data transmission for a period set by the server or

the second station. The server may block the data transmission between the first and second stations after the set period elapses. Implementations of the described techniques may include hardware, a method or process, or computer software stored on a computer-accessible medium.

**[0011]** Another general aspect includes a method for securing satellite data communications between first and second stations. The method includes: transmitting, by the first station, a request for an access key to a server via a satellite; receiving, by the first station, an access key from the server; transmitting, by the first station, the access key to the second station; transmitting, by the second station, the access key to the server; validating, by the server, the access key transmitted by the second station; permitting satellite data communications between the first station and the second station when the server determines that the access key is valid; and storing, by the server, a transmission log in a hyperledger. Other aspects of this aspect include corresponding computer systems, apparatus, and computer programs recorded on one or more computer storage devices, each configured to perform the actions of the method.

**[0012]** Implementations may include one or more of the following features. The transmission log may include a validation history of the access key. The transmission of the data may be blocked when the server determines that the access key is not valid. The access key may be encrypted by the server. The satellite data communications may be encrypted before being transmitted to the first station. The encrypted satellite data communications may be saved in the hyperledger. The method may further include encrypting, by the second station, the satellite data communications with a public key before transmitting the satellite data communications to the first station. The method may further include decrypting, by the first station, the encrypted satellite data communications with a private key corresponding to the public key. The access key may be valid for satellite data communications for a period set by the server or the second station. The satellite data communications may be blocked between the first and second stations after the set period. Implementations of the described techniques may include hardware, a method or process, or computer software stored on a computer-accessible medium.

**[0013]** Further details and aspects of exemplary aspects of this disclosure are described in more detail below with reference to the appended figures.

**BRIEF DESCRIPTION OF THE DRAWINGS**

[0014] A better understanding of the features and advantages of the disclosed technology will be obtained by reference to the following detailed description that sets forth illustrative aspects, in which the principles of the technology are utilized. It will be appreciated that for simplicity and clarity of the illustration, elements shown in the figures referenced below are not necessarily drawn to scale.

[0015] FIG. 1 is a diagram illustrating an e-key system for providing secure and safe ways to protect against unlawful or unapproved access to property in accordance with aspects of this disclosure;

[0016] FIG. 2 is a block diagram of a server of FIG. 1, which uses blockchain technology, in accordance with aspects of this disclosure;

[0017] FIG. 3 is a block diagram illustrating a procedure for generating a block using blockchain technology in accordance with aspects of this disclosure;

[0018] FIG. 4 is a block diagram illustrating a user interface displayed on a smart device of FIG. 1 in accordance with aspects of this disclosure;

[0019] FIG. 5 is a sequence diagram illustrating a communication procedure for the e-key system in accordance with aspects of this disclosure;

[0020] FIG. 6 is a block diagram illustrating a user interface for granting/denying access to property in accordance with aspects of this disclosure;

[0021] FIG. 7 is a sequence diagram illustrating a communication procedure for granting/denying access to a third party in accordance with aspects of this disclosure;

[0022] FIG. 8 is a flow diagram illustrating a method for controlling access to property using a mobile application in accordance with aspects of this disclosure;

[0023] FIG. 9 is a flow diagram illustrating a method for controlling access to a marine or air vehicle in accordance with aspects of this disclosure;

[0024] FIG. 10 is a flow chart illustrating a method for controlling access to a property in accordance with aspects of this disclosure;

[0025] FIG. 11 is a block diagram of a computing device used in the e-key system of FIG. 1 in accordance with aspects of this disclosure;

[0026] FIG. 12A is a block diagram of an example key fob system in accordance with aspects of this disclosure;

- [0027] FIG. 12B is a sequence diagram illustrating a communication procedure for a key fob system in accordance with aspects of this disclosure;
- [0028] FIG. 13 is a flow diagram illustrating the operation of the example key fob system of FIG. 12A in accordance with aspects of this disclosure;
- [0029] FIG. 14 is a block diagram illustrating a satellite communication system using blockchain in accordance with aspects of this disclosure;
- [0030] FIG. 15 is a flow chart illustrating a method for permitting/denying data communication via a satellite in accordance with aspects of this disclosure;
- [0031] FIG. 16 is a block diagram illustrating a military communication system using satellite and blockchain in accordance with aspects of this disclosure;
- [0032] FIG. 17 is a block diagram illustrating an application of the satellite communication system of FIG. 14 to the aviation industry in accordance with aspects of this disclosure;
- [0033] FIG. 18 is a block diagram illustrating an application of the satellite communication system of FIG. 14 to the marine industry in accordance with aspects of this disclosure;
- [0034] FIG. 19 is a block diagram illustrating an e-key architecture in accordance with aspects of this disclosure;
- [0035] FIG. 20 is a block diagram illustrating a general system architecture in accordance with aspects of this disclosure;
- [0036] FIG. 21 is a block diagram illustrating a smart chip system in accordance with aspects of this disclosure;
- [0037] FIG. 22 is a schematic diagram illustrating the physical layers of a smart card including the smart chip of the smart chip system of FIG. 21;
- [0038] FIG. 23 is a schematic diagram illustrating the contacts of the smart chip of FIG. 23;
- [0039] FIG. 24 is a block diagram illustrating examples of the components of the smart chip of FIG. 21;
- [0040] FIG. 25 is a block diagram illustrating an example of a microcontroller of the smart chip of FIG. 22;
- [0041] FIG. 26 is a block diagram illustrating the smart chip reader of the smart chip system of FIG. 21;

[0042] FIGS. 27A and 27B show a flowchart illustrating a method of reading or writing to an application zone in accordance with aspects of this disclosure;

[0043] FIGS. 28A and 28B show a flowchart illustrating a method of reading or writing to configuration memory in accordance with aspects of this disclosure;

[0044] FIG. 29 is block diagram of a smart chip system incorporated into an e-key system architecture in accordance with aspects of this disclosure;

[0045] FIG. 30A is a block diagram illustrating examples of input components of a biometric system incorporated into the e-key architecture in accordance with an aspect of this disclosure;

[0046] FIG. 30B is a block diagram illustrating examples of input components of a telecommunication system incorporated into the e-key architecture in accordance with an aspect of this disclosure;

[0047] FIG. 30C is a block diagram illustrating examples of input components of a supply chain management system incorporated into the e-key architecture in accordance with an aspect of this disclosure;

[0048] FIG. 30D is a block diagram illustrating examples of input components of a satellite system incorporated into the e-key architecture in accordance with an aspect of this disclosure;

[0049] FIG. 30E is a block diagram illustrating examples of input components of an automobile system incorporated into the e-key architecture in accordance with an aspect of this disclosure;

[0050] FIG. 30F is a block diagram illustrating examples of input components of an aviation system incorporated into the e-key architecture in accordance with an aspect of this disclosure;

[0051] FIG. 30G is a block diagram illustrating examples of input components of a healthcare system incorporated into the e-key architecture in accordance with an aspect of this disclosure;

[0052] FIG. 30H is a block diagram illustrating examples of input components of a fintech system incorporated into the e-key architecture in accordance with an aspect of this disclosure;

[0053] FIG. 30I is a block diagram illustrating examples of input components of a pharmaceutical industry system incorporated into the e-key architecture in accordance with an aspect of this disclosure;

[0054] FIG. 31 is a block diagram illustrating an IoT gateway module incorporated into the e-key architecture in accordance with aspects of this disclosure;

[0055] FIG. 32 is a block diagram illustrating firmware and software interfaces which may be implemented on the SBC of FIG. 29 in accordance with aspects of this disclosure;

[0056] FIG. 33 is a block diagram illustrating an example of the hardware of the single board computer (SBC) of FIG. 29 in accordance with aspects of this disclosure; and

[0057] FIG. 34 is a block diagram illustrating application modules that may be run on the SBC of FIG. 29 in accordance with aspects of this disclosure.

### **DETAILED DESCRIPTION**

[0058] This disclosure relates to systems and methods for securing data communications and property using blockchain. In particular, this disclosure relates to systems and methods for securing satellite and wireless communication between a mobile device and personal or business property using blockchain technology so as to increase the security level and protect the communication data and properties against unintended, unlawful, or unapproved attempts to access or tamper with the communication data or property. Aspects and aspects of securing data communications and property are described with respect to the figures set forth below.

[0059] Wireless key systems have been developed particularly in automobile industries. Wireless key systems generally use radio signals in communicating with vehicles. A holder of the wireless key presses a button on the wireless key, which then communicates with a communication device of the property by transmitting a radio signal to the communication device. When the transmitted radio signal matches the code saved in the property, the holder can gain access to the property. However, the radio signal can be intercepted or hijacked by a simple electronic gadget located near the wireless key. A person intercepting the radio signal can use the intercepted radio signal to gain access to the property without the owner knowing. Thus, communications using the wireless key systems are exposed to risks of being hacked or intercepted.

[0060] Furthermore, since the radio signal saved in the wireless key is generally predetermined or unchanged, a hacker can copy or mimic the radio signal to gain access to

the property. Thus, the predetermined radio signal is also exposed to risks of being hacked or copied.

[0061] Still further, in general, data which is transmitted wirelessly or via wire is exposed to risks of hacking so that important personal, financial, tactical, medical, military, or secret data may be lost, modified, or used for unintended purposes. Therefore, there is a need to develop more secure and safer ways to protect communication data and track and secure entry or access to business or personal property.

### **E-KEY SYSTEM**

[0062] The e-key system may be applied, for example, to a business building, a residential house, a land vehicle, an air vehicle, or a marine vehicle. In aspects, the e-key system may be applied to any property including business or personal buildings, properties, safes, vehicles, machines, or any other things that need protection against unintended, unlawful, unauthorized, or unapproved access or use of those things (e.g., unlawful entry to a person's home or unauthorized starting of an car engine). Further, the e-key system saves access or use history in a hyperledger using blockchain technology and can track every access to or use of properties so that security is preserved and guaranteed.

[0063] Further, wireless e-keys communicate with computing devices incorporated into or associated with the properties using end-to-end encryption (E2EE) to prevent hijacking of communication data, which includes the codes described herein, during transmission, thereby increasing reliability and security. While this disclosure may refer to wireless e-key systems, aspects of this disclosure may also apply to e-key systems that use technology other than wireless technology. For example, an electrical contact of a reader device may make electrical contact with an electrical contact of a smart chip and read information from or write information to the smart chip via the physical electrical connection.

[0064] FIG. 1 shows an e-key system 100 according to aspects of this disclosure. The e-key system 100 includes property 110, which includes an entry 115, a communications device 111, and a computing device 112 in communication with the communications device 111, a smart device 120, and servers 130a-130n, which are connected to each other through a network 140. The e-key system 100 wirelessly and selectively grants access to an entry 115 of the property 110 or wirelessly and selectively grants use of the property. In other aspects, the e-key system 100 may selectively grant access to an entry 115 of the property 110 (or

access to information) via a wired connection or contact (e.g., through electrical contact) or may selectively grant use of the property via a wired connection or contact.

**[0065]** The entry 115 may be a door or a lock. And the use may be starting an engine or powering an electric motor associated with the property 110 (e.g., a truck or an electric car). In particular, if the property 110 is a land vehicle, the entry 115 may be a door to get into the land vehicle; if the property 110 is an air or marine vehicle, the entry 115 may be a door of the air or marine vehicle or the use may be starting an engine so that a user can control or operate the air or marine vehicle; or if the property 110 is a building or a safe, the entry 115 may be one or more doors. The above list of the property 110 and the associated entry 115 or use are not intended to be limited thereto but to provide examples. Other types of properties, entries, and uses are readily understood by those having ordinary skill in the art without departing from the spirit of this disclosure.

**[0066]** The smart device 120 works or functions as a wireless key in the e-key system 100. The smart device 120 receives a code from one or more of the servers 130a–130n. The code may be randomly generated by the servers 130a–130n and works as an authorized key to the entry 115 or use of the property 110. The smart device 120 may wirelessly receive the code through the network 140, which may be the Internet, a local area network, a wide area network, an ad hoc network, or any other network capable of wireless transmission of data including the code.

**[0067]** The smart device 120 may be a cellphone, personal digital assistant (PDA), tablet, phablet, computer, portable computer, smart watch, or any other compatible computing device, which is capable of communicating with the one or more servers 130a–130n by using a first wireless communication method and with the computing device (not shown) of the property 110 by using a second wireless communication method different from the first wireless communication method. The computing device of the property 110 may be the same computing device used to control or operate the property 110.

**[0068]** When the smart device 120 receives the code through the network 140, the servers 130a–130n also transmit the code to the computing device of the property 110 through the network 140 via a wireless communication method, and, in turn, the computing device of the property 110 saves the code in memory, which may be incorporated into the computing device or the property 110. In this way, the computing device of the property 110 may check whether the code transmitted from the smart device 120 is the same as the code saved in the

memory of the property 110 or the computing device. The wireless communication method used between a communication device (e.g., a wireless transceiver) coupled to the one or more servers 130a–130n and the communication devices or electronics coupled to the smart device 120 or the computing device of the property 110 may be Wi-Fi, 2G–5G GSM, TDMA, CDMA, long term evolution (LTE), or any other communication method used for long-distance communication.

**[0069]** During the transmission of the code, the smart device 120 uses another wireless communication method different from the wireless communication method used between the one or more servers 130a–130n and the smart device 120 or the computing device of the property 110. The other wireless communication method may be near field communication (NFC), Bluetooth, Bluetooth low energy (BLE), ZigBee, infrared (IR), or radio frequency identification (RFID). This list merely provides examples of the wireless communication method used between the smart device 120 and the communication device of the property 110. However, other kinds of wireless communication methods may be employed without departing the spirit of this disclosure.

**[0070]** The range of communication between the smart device 120 and communication device of the property 110 may be shorter than the range of communication between the smart device 120 and one or more of the servers 130a–130n or between the communication device of the property 110 and one or more of the servers 130a–130n. In aspects, the communication device of the property 110 may be directly connected to one or more of the servers 130a–130n, in a case where the property 110 is immovable, such as a residential or commercial building, a laboratory, a house, etc.

**[0071]** The smart device 120 may employ E2EE to transmit the code to the communication device of the property 110. By using E2EE, no eavesdroppers can intercept and decipher the code. Only the true sender (i.e., the user of the smart device 120) and the true recipient (i.e., the owner or user of the property 110) can encrypt and decipher the code, respectively, thereby ensuring that the communications between the smart device 120 and the communication device of the property 110 is secure.

**[0072]** The servers 130a–130n may have the same, similar, or different structures with respect to each other. Regardless of the structure of the servers 130a–130n, each of the servers 130a–130n may save the same hyperledger using blockchain technology. When the computing device of the property 110 creates a transaction reflecting an event in which

access to the property is granted or denied, the servers 130a–130n save the transaction into the hyperledger as a block, for example, as described in application number 16/156,570, the entire contents of which are incorporated by reference herein. Thus, when one or more transactions are transmitted to the servers 130a–130n, a block is created and saved in servers 130a–130n. Details about creating a block are described with respect to FIG. 3 below.

**[0073]** FIG. 2 is a block diagram of a server 200, which may be used as each of the servers 130a–130n of FIG. 1 and which uses blockchain in accordance with aspects of this disclosure. The server 200 may include a code generator 210, a network interface 220, an access history generator 230, a block generator 240, and a hyperledger 250. The code generator 210 may generate a random code upon reception of a request from the smart device 120. The size of the random code may be 16 bytes, 64 bytes, 128 bytes, 256 bytes, or longer based on a desired level of security for the property 110 or based on security requirements prescribed for the property 110. For example, if the property 110 is an automobile or a bicycle, the size of the random code may be 16 bytes. In a case where the property 110 is an aircraft, the size of the random code may be 32 bytes or longer. If the property 110 is a piece of military equipment (e.g., a fighter jet or a submarine), the size may be 128 bytes, 256 bytes, or any other size suitable for military equipment. The code may be a numeric sequence, an alphanumeric sequence, a hexadecimal sequence, or any other sequence known to those skilled in the art of cryptography.

**[0074]** The random code generated by the code generator 210 is transmitted to the smart device 120 and the computing device of the property 110 via the network interface 220. The random code is then used by the computing device of the property 110 to check whether the random code transmitted from the smart device 120 matches the code saved in memory of the property 110.

**[0075]** The random code may be valid or effective for a predetermined period. After the predetermined period has elapsed, the random code may be invalidated and access to the property 110 using the random code cannot be granted. To gain access, the smart device 120 would need to receive another random code from the server 200.

**[0076]** In an aspect, a random code may be needed to maintain access every predetermined period. In this situation, the smart device 120 receives a random code every predetermined period, transmits the random code to the computing device of the property 110, and thereby continuously receives access to the property 110. In a case where the

network 140 has an issue (e.g., a communication error or disconnection) and, as a result, wireless communication does not work between the server 200 and the smart device 120 or the computing device of the property 110, the counting of the predetermined period may stop during the period when the wireless communication does not work, and then the counting may resume after the issue is resolved.

**[0077]** In a case where the communication device 111 of the property 110 has a network issue and is unable to communicate with the server 200, but the smart device 120 is able to communicate with the server 200, the server 200 may search the hyperledger 250 to find out which code was the last code transmitted to the communication device 111 of the property 110 and then may transmit the last code to the smart device 120. In this way, the smart device 120 is able to gain access to the property 110 in such situations.

**[0078]** When the code transmitted from the smart device 120 matches the code saved in the memory 113 of the property 110, the smart device 120 is granted access to the property 110. And when the code transmitted from the smart device 120 matches the code saved in the memory 113, the smart device 120 is denied access to the property 110. In either case, the property 110 transmits, to the server 200, a result of granting or denying access to the property 110. The access history generator 230 then generates a log of the result. The log may include the addresses or identifications of the smart device 120 and the property 110 and may further include the result of granting or denying access to the property 110. The log may include a timestamp when the grant or denial of access occurs. In aspects, the log may include the network status between the smart device 120 and the communications device 111 of the property 110, between the smart device 120 and the server 200, and between the communications device 111 of the property 110 and the server 200.

**[0079]** After the access history log is generated or updated by the access history generator 230, the block generator 240 generates a block to be saved in the hyperledger 250. Each block is related to its previous block. For example, FIG. 3 illustrates how blocks are related to each other and, in particular, how each block is related to a preceding block. When current log data 310 (Data\_n2) comes to the block generator 240, the block generator 240 retrieves a hash code from the previous block 320.

**[0080]** A hash code is generated by a hash function. When the hash function receives data as input, the hash function outputs a series of alphanumeric values. One advantage

of using a hash function is that the hash function outputs a series of alphanumeric values very different from the series of alphanumeric values generated from first data when second data, which is slightly different from the first data, is input to the hash function. For example, when the first data is “Sarah” and the second data is “Sara”, the hash code generated from the first data is very different from the hash code generated from the second data. Thus, the usage of hash codes further increases the security level and prevents hacking.

**[0081]** The block generator 240 uses the current data, Data\_n2, of the current block 310 and the previous hash code, Hash\_n1, of the previous block 320, as input to a hash function executed by the block generator 240. The hash function then generates a current hash code, Hash\_n2, and the block generator 240 generates the current block 330 by combining the current hash code, Hash\_n2, and the current data, Data\_n2. Since the current block 330 is partly generated from the hash code of the previous block 320, every block is related to a previous block. Thus, modifying the data content of one block is not feasible unless hash codes of all blocks starting from the original block are modified.

**[0082]** Referring back to FIG. 2, when the current block 330 is generated, it is saved in the hyperledger 250. Since the blockchain technology is used in the hyperledger 250, the hyperledger 250 is secure against potential modifications of data saved in the hyperledger 250. Further, the same hyperledger 250 is saved in all the servers 130a–130n. It is impractical to modify any data saved in the hyperledger 250 unless hash codes of all blocks saved in all the servers 130a–130n are modified at almost the same time. The reason behind the almost same time modification is that the servers 130a–130n validate each block before saving each block in the hyperledger 250. If attempts are made to modify data in the hyperledger 250 saved to a small group of servers 130a–130n, such attempts cannot pass the validation process and the hyperledger 250 cannot be modified.

**[0083]** FIG. 4 illustrates a user interface 400 displayed on the smart device 120 of FIG. 1 in accordance with aspects of this disclosure. When a user of the smart device 120 installs an application for wirelessly accessing a computing device of a property and executes the application, the user interface 400 may be displayed. The application may be a mobile application retrievable from the Google Store for android-based smart

devices or from the App Store for iOS-based smart devices. The application may also be found in third-party application stores.

[0084] The user interface 400 may include three buttons: an OPEN button 410, a CLOSE button 420, and a GUEST ACCESS button 430. Since the GUEST ACCESS button 430 may not be available for every user, it may not be visible to every user. The OPEN button 410 may be used to send a request for opening a door to a property to the server 200, the CLOSE button 420 may be used to send a request for closing the door to the server 200, and the GUEST ACCESS button 430 may be used to send a request for giving permission to other users to use the property to the server 200.

[0085] When the OPEN button 410 or the CLOSE button 420 is clicked by the user, a process illustrated in FIG. 5 is initiated. At block 510, the smart device sends a request to the server to open or close the door of the property. Upon receipt of the request, the server generates and transmits a code (e.g., a random code) to the smart device at block 520 and to the computing device of the property at block 525. Transmission of the generated code to the smart device and the property may be simultaneously or serially performed.

[0086] The length of the code may depend from a level of security embodied in the property and/or security requirements prescribed in the property.

[0087] When the smart device receives the random code, the smart device encrypts the random code and transmits the encrypted code to the property at block 530 by using E2EE. The property deciphers the encrypted code and checks whether the deciphered code matches the code received from the server. When they match, the property grants the user of the smart device to open/close the property at block 540. If not, the property denies the user to open/close the property at block 540. The result of the grant/deny is also sent to the server at block 545.

[0088] When the GUEST ACCESS button 430 is clicked, another user interface 600 may be displayed on the smart device as illustrated in FIG. 6. The user interface 600 includes a list of third parties, which were previously added to the list by the user of the smart device. The user of the smart device can add or remove third parties into the list. In aspects, the user of the smart device may be an owner, a manager, or an operator of the property and may have authority or power to give people access to the property.

[0089] The list includes names of peoples and status buttons informing grant/deny right next to the names. For example, as illustrated in FIG. 6, the user of the smart device has not grant or deny Alan Smith and Sophia Zeller, meaning that Alan Smith and Sophia Zeller are not allowed to open or close the door of the property. In contrast, Ava Christine and harper Leon are granted, meaning that Ava Christine and harper Leon are allowed to open or close the door of the property. Further, Charlotte Dean is specifically indicated that opening or closing the door is not allowed.

[0090] The status button may be toggled among “GRANT/DENY,” “GRANT,” and “DENY.” Thus, the user of the smart device may be able to identify the status of a third party by clicking the status button. In aspects, there are radio buttons for “GRANT/DENY,” “GRANT,” and “DENY,” and the user of the smart device may choose one of the radio buttons. Selection of status among “GRANT/DENY,” “GRANT,” and “DENY” may be done in any other ways, which can be appreciated by a person having skill in the art.

[0091] After completing selection of third parties and corresponding statuses, the user of the smart device clicks “DONE” button. Then, the smart device sends a request for third parties to the server. The request may include information of the third parties, which may include a name, physical address, email address, Internet protocol (IP) address, phone number, or media access control (MAC) address of the smart device of a third party. This list of information of the third parties are not meant to be exhaustive but to provide examples. Further specifics indicative of the third parties may be included in the information.

[0092] The smart device may send a message to the selected third parties, which guides the third parties to install the corresponding mobile application and shows how to open and close the door of the property. In aspects, the server may send the message to the selected third parties based on the information included in the request.

[0093] When a selected third party receives the message and installs the mobile application, the user interface 400 of FIG. 4 may be displayed without the GUEST ACCESS button 430, meaning that only the OPEN button 410 and the CLOSE button 420 may be displayed on the smart device of the selected third party. In other words, the selected third parties do not have an authority or power to grant or deny other third parties to open or close the door to the property.

[0094] In a case when the user of the smart device selects a third party and identifies the corresponding status as DENY, the smart device may send a message to a smart device of the selected third party, notifying that the selected third party is no longer able to open or close the door to the property. In aspects, the mobile application may be disabled so that the selected third party may not be able to execute the mobile application, or when executed, the mobile application may show a user interface indicating that no available options can be selected.

[0095] FIG. 7 illustrates processes for granting or denying a selected third party to open or close the door to the property in accordance with aspects of this disclosure. At block 710, the user selects a group of third parties to grant an entry to the property and clicks the DONE button, the smart device of the user sends a request to the server. In return, the server generates a random code and transmits to the smart device of the user and the smart device of the selected third party at blocks 720 and 725, respectively. Further, the server transmits the random code to the property at block 730. Transmission of the random code may be simultaneously or serially performed. In aspects, the server may transmit a message rather than the random code to the smart device of the user, indicating that the random code is transmitted to the smart device of the selected third party and the property at block 720.

[0096] At block 740, the smart device of the selected third party encrypts the random code and transmits the encrypted code to the property, which then deciphers the encrypted code. When the deciphered code matches the code saved in the property, the property grants an entry to the smart device of the selected third party at block 750 and sends the result of the grant to the server at block 755. Likewise, when the deciphered code does not match the code, a denial of the entry is sent to the server by the smart device of the selected third party at block 755.

[0097] In aspects the server or the property may send a message indicating the grant or denial of the entry to the selected third party so that the user can confirm his selection of the third party and the status.

[0098] FIG. 8 is a graphical flow diagram illustrating the granting or denying of entry to property (e.g., an automobile) using a mobile application running on a smart device 802 in accordance with aspects of the disclosure. At block 805, the user of the smart device initiates a mobile application by, for example, selecting an icon representing the mobile application

displayed on the display of the smart device. The mobile application displays selectable buttons including OPEN, CLOSE, and ACCESS buttons. In the case where the OPEN button is selected by the user at block 810, the mobile application sends an OPEN request to the server 852 and receives a random code from the server 852 at block 815. The mobile application encrypts the random code, sends the encrypted random code to the property (e.g., automobile 822) at block 820, and the user is granted entry to the property by a computing device residing in and controlling access to the property. Then, the door to the property is opened to the user of the smart device by the computing device at block 825, which concludes the process that is performed upon the selection of the OPEN button.

**[0099]** When the user selects the CLOSE button in the mobile application at block 830, the mobile application sends a CLOSE request to the server 852 and receives a random code from the server 852 at block 835. The mobile application encrypts the random code, sends the encrypted random code to the property (e.g., automobile 822) at block 840, and is granted permission to close the door to the property by the computing device. Then, the door to the property is closed by the computing device at block 845, e.g., by controlling the automatic locks of the automobile 822 to close, which concludes the process that is performed upon the selection of the CLOSE button displayed by the mobile application.

**[0100]** When the user selects the ACCESS button at block 850, the mobile application transmits information including status information regarding selected third parties to the server 852 at block 855. When the status information regarding the selected third parties is set to DENY, the server 852 prevents the selected third parties from opening or closing the door to the property at block 885.

**[0101]** In the case where the status information regarding the selected third parties is set to GRANT, the server 852 may cause smart devices of the selected third parties to initiate a mobile application by transmitting a message to the mobile application, which displays buttons including an OPEN button and a CLOSE button. The mobile application may display an ACCESS button in a deactivated state so that the selected third parties cannot select the ACCESS button.

**[0102]** At block 865, if a selected third party clicks the OPEN button, the mobile application of the smart device of the selected third party encrypts and transmits a code received from the server 852 to the property (e.g., the automobile 822) at block 870. The computing device residing in the property then deciphers the encrypted code and

determines whether the deciphered code (e.g., a random code) matches the code saved in memory of the computing device or in memory disposed in or integrated into the property. If the deciphered code matches the code saved in the memory, the computing device causes the property to open its door to the selected third party at block 825 (e.g., the computing device of the property transmits a signal to one or more of the automatic locks of the property to cause the one or more automatic locks to transition from a locked state to an unlocked state.

**[0103]** The code (e.g., the random code) may be transmitted from the server 852 after the selected third party clicks the OPEN button or at a time when the server 852 transmits the message to the mobile application running on the selected third party's smart device. In aspects, the message sent from the server 852 to the mobile application may include the random code.

**[0104]** If the selected third party clicks the CLOSE button at block 875, the mobile application of the smart device of the selected third party encrypts and transmits the code received from the server 852 to the computing device of the property at block 880. The computing device of the property then deciphers the encrypted code using a suitable decryption algorithm and determines whether the deciphered code matches the code saved in the memory of the property. If the deciphered code matches the code save in the memory of the property, the computing device of the property causes the door of the property to close at block 845.

**[0105]** Wirelessly granting or denying entry may be applied to marine vessels and air vessels. FIG. 9 is a graphical flow diagram illustrating the granting or denying of entry to a watercraft or an aircraft using a mobile application in accordance with aspects of the disclosure. In the aspects of FIG. 9, the smart device may be a portable electronic device 910, such as a universal serial bus (USB) drive 910, which includes a display. The USB drive 910 is just one of many examples of electronic devices which include a display and are capable of communicating with the server 922 and the computing devices residing on or integrated into the watercraft or aircraft 932.

**[0106]** The watercraft or aircraft are generally more complicated and larger in size than land vehicles, thus requiring a greater level of security than land vehicles. When a request is made for entry into the marine or air vehicle, the server 922 sends at block 920 a random code, which may be longer than the random code used for land vehicles. The

random code may be valid for a predetermined period. For example, the predetermined period may be 5, 10, 15, 20, 25, 30, or more minutes depending on the requirements of the watercraft or aircraft.

**[0107]** The received random code may be displayed on the display of the portable electronic device 910. The user of the portable electronic device 910 may type the displayed random code into the watercraft or aircraft or insert the portable electronic device 910 into a port of the watercraft or aircraft at block 930. By typing in the random code or inserting the portable electronic device 910, the watercraft or aircraft receives the random code. In aspects, the portable electronic device 910 may encrypt the random code and transmit at block 930 the encrypt code to the watercraft or aircraft by using a short-range communication method, such as NFC, BLE, ZigBee, IR, or RFID. In this short-range communication method, E2EE is used to ensure protection.

**[0108]** The watercraft or aircraft also receives the random code from the server at block 925. The watercraft or aircraft decipheres the received encrypted code and determines whether the deciphered code matches the code saved therein. When it is determined that the deciphered code does not match at block 960, the portable electronic device 910 is caused to display “INVALID” on its screen so that the watercraft or aircraft cannot be controlled by the user at block 970.

**[0109]** In case when it is determined that the deciphered code matches the stored code at block 940, the engine of the watercraft or aircraft can be powered and activated by the user of the portable electronic device 910 so that the user is able to control the watercraft or aircraft at block 950.

**[0110]** The server may store all activities including controls of the watercraft or aircraft and results of random code inputs whether or not valid or invalid in a hyperledger using blockchain technology. Further, the user may access the hyperledger through a valid random code and check all activities related to the portable electronic device 910 and the watercraft or aircraft.

**[0111]** FIG. 10 is a flowchart illustrating a method for granting or denying entry to property in accordance with aspects of the disclosure. The method 1000 may be stored in memory as computer-executable instructions. The method 1000 may perform all or a portion of the blocks when a computer or processor execute the stored computer-executable

instructions. In this way, the method 1000 may be implemented by any electronic gadget, which includes a processor and a storage medium.

**[0112]** The method 1000 starts by displaying selectable buttons or options on a screen of a smart device at block 1010. The buttons may include OPEN, CLOSE, GUEST ACCESS. OPEN and CLOSE are for opening and closing a door of a property, and GUEST ACCESS is for allowing one or more third parties an entry to the property.

**[0113]** At block 1015, it is determined which option is selected. In case OPEN or CLOSE is determined to be selected, the smart device transmits to a server a request for opening or closing the entry to the property at block 1020. In aspects, the entry may mean starting an engine for moving vehicles such as cars, hovercraft, aircraft, and watercraft. In another aspect, the entry may mean actual entry to immovable buildings via a door thereof, such as financial institutions, business establishments, individual residents, and safes.

**[0114]** The request may include information of the smart device, such as a name, physical address, email address, IP address, phone number, or MAC address of the smart device. Further, the selected button may be included in the request.

**[0115]** The smart device may employ a wireless communication method through Internet, which may be Wi-Fi, 2G–5G GSM, TDMA, CDMA, LTE, Bluetooth, or any other communication method used for a long-distance communication.

**[0116]** The server randomly generates a code upon reception of the request and at block 1030, transmits the random code to the smart device via the same communication method. The size or length of the random code may be shorter or longer than or equal to 32 bytes depending on the required level of security of the property. The server may transmit the random code to the property at block 1030.

**[0117]** The smart device encrypted the random code and transmits the encrypted code to the property at block 1035. The communication method employed between the smart device and the property may be a short-range communication method, such as NFC, BLE, ZigBee, IR, or RFID. The short-range communication method is different from the communication method used between the smart device and the server. In this short-range communication method, the E2EE is used to ensure protective measures. E2EE may cause the smart device to encrypt the random code in transmission. The E2EE may utilize a public

key system so that only the smart device and the property can decipher the encrypted code and no other eavesdroppers can decipher the encrypted code.

**[0118]** At block 1035, it is determined whether or not the deciphered code matches the code stored in the property. In case when it is determined that the deciphered code matches the stored code, the request is granted at block 1040 so that the user of the smart device can enter into or exit from the property. If not, the request is denied at block 1045 so that the user is not allowed to enter into or exit from the property. This concludes the selection of OPEN or CLOSE in the options displayed on the smart device.

**[0119]** Now, referring back to block 1015, when it is determined that GUEST ACCESS is selected, the user of the smart device is able to allow one or more third parties to open or close the entry to the property. The smart device displays a list of third parties, which have been previously added to the list at block 1050. When the user selects one or more third parties, a corresponding status, which may be GRANT or DENY, may be selected for each of the one or more third parties. The status of DENY may be selected to prevent a third party from opening or closing the door of the property.

**[0120]** The smart device receives selection of one or more third parties by the user of the smart device at block 1055. At block 1060, the smart device transmits a request with the selection of one or more third parties. The selection may include the selected status for each of the selected third parties. In aspects, the selection may further include information of the smart device of each selected third party, such as a name, physical address, email address, IP address, phone number, or MAC address.

**[0121]** At block 1065, the server generates a random code and transmits the random code to each of the selected third parties based on the information of the selected third parties. The server sends the random code to the property, which then saves the random code for a later checkup. At block 1070, the smart device of each selected third party receives the random code.

**[0122]** The smart device of each selected third party encrypts the random code and transmits the encrypted code to the property at block 1075. The encryption may be performed while transmitting the random code by using the E2EE.

**[0123]** The property deciphers the encrypted code and at block 1080, the property determines whether or not the deciphered code matches the code saved in the property.

When it is determined that the deciphered code does not match the stored code, the request is denied at block 1045.

[0124] When it is determined that the deciphered code match the stored code, the request for opening or closing an entry to the property is granted to the selected third party at block 1040.

[0125] In aspects, after blocks 1040, 1045, and 1085, the result of grant or deny of the request is recorded as an access history log. The server may generate a block using blockchain technology and save the block in a hyperledger, which is then saved in a plurality of servers. The hyperledger may be accessed and searched by the smart device receiving the random code from the server.

[0126] FIG. 11 shows a block diagram of a computing device 1100 representing the smart device 120 or the servers 130a–130n of FIG. 1 in accordance with aspects of this disclosure. The computing device 1100 may be the portable electronic device 910 of FIG. 9. The computing device 1100 includes processor 1110, memory 1120, display 1130, input device 1140, and/or network interface 1150. The memory 1120 may store one or more applications and data.

[0127] The memory 1120 may include any non-transitory computer-readable storage media for storing data and/or software that is executable by the processor 1110 and which controls the operation of the computing device 1100. In aspects, the memory 1120 may include one or more solid-state storage devices such as flash memory chips. Alternatively, or in addition to the one or more solid-state storage devices, the memory 1120 may include one or more mass storage devices connected to the processor 1110 through a mass storage controller (not shown) and a communications bus (not shown). Although the description of computer-readable media contained herein refers to a solid-state storage, it should be appreciated by those skilled in the art that computer-readable storage media can be any available media that can be accessed by the processor 1110. That is, computer readable storage media may include non-transitory, volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data. For example, computer-readable storage media may include random access memory (RAM), read-only memory (ROM), erasable programmable ROM (EPROM), electrically erasable programmable ROM (EEPROM),

flash memory or other solid state memory technology, CD-ROM, DVD, Blu-Ray or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which may be used to store the desired information and which may be accessed by the computing device 1100.

**[0128]** The memory 1120 further includes an operating system (OS) configured to control basic input/output functions and the OS file system. Those skilled in the art will recognize that suitable operating systems include, by way of non-limiting examples, FreeBSD, OpenBSD, NetBSD®, Linux, Apple® Mac OS X Server®, Oracle® Solaris®, Windows Server®, and Novell® NetWare®. Those skilled in the art will also recognize that suitable personal computer operating systems or mobile operating systems include, by way of non-limiting examples, Microsoft® Windows®, Apple® Mac OS X®, UNIX®, and UNIX-like operating systems such as GNU/Linux®. In some aspects, the OS is provided by cloud computing. Those skilled in the art will also recognize that suitable mobile smart phone operating systems include, by way of non-limiting examples, Nokia® Symbian® OS, Apple® iOS®, Research In Motion® BlackBerry OS®, Google® Android®, Microsoft® Windows Phone® OS, Microsoft® Windows Mobile® OS, Linux®, and Palm® WebOS®.

**[0129]** The display 1130 may include a cathode ray tube (CRT), liquid crystal display (LCD), light-emitting diode (LED), or any other forms of display. The display may be touch-sensitive and can be used as an input device.

**[0130]** The network interface 1150 may be configured to connect to a network such as a local area network (LAN) consisting of a wired network and/or a wireless network, a wide area network (WAN), a wireless mobile network, a Bluetooth network, and/or the Internet. The input device 1140 may be any device by means of which a user may interact with the computing device 1100, such as, for example, a mouse, keyboard, foot pedal, touch screen, and/or voice interface.

### **KEY FOB SYSTEM**

**[0131]** FIG. 12A shows a key fob system 1200 and FIG. 12B shows an information flow diagram in the key fob system 1200 according to aspects of present disclosure. The key fob system 1200 includes a property 1210, a smart device 1220, a key fob 1230, and a server 1240, of which all are connected through a network. The key fob system 1200 wirelessly grants access to the property 1210 upon reception of an encrypted code from the key fob 1230. The entry 1215 may be a door, an engine, or a lock. For example, when the property

1210 is a land vehicle, the entry 1215 may be a door to get into the land vehicle or an engine to be started; when the property 1210 is an aircraft or watercraft, the entry 1215 may be a door of the aircraft or watercraft or an engine to be started so that a user can control the aircraft or watercraft; or when the property 1210 is a building or a safe, the entry 1215 may be a door or lock to the building or the safe, respectively. The above lists of the property 1210 and the entry 1215 are not intended to be limited thereto but to provide examples.

**[0132]** The smart device 1220 works as a control device in the key fob system 1200. The smart device 1220 may be a smartphone, tablet, phablet, computer, mobile device, or other suitable device which is capable of communicating with the key fob 1230 and the server 1240 via different communication methods. For example, the smart device 1220 may wirelessly or wiredly communicate with the server 1240 through the Internet and with the key fob 1230 through a Bluetooth connection.

**[0133]** The key fob 1230 may include one or more buttons, which can be used to access the property 1210. In an aspect, the key fob 1230 may include one button for sending an encrypted code to the property 1210 for receiving access to the property 1210. In another aspect, the key fob 1230 may include several buttons, each of which may be assigned to a function of the property 1210. For example, one button is to start an engine in a case where the property 1210 is a land vehicle, watercraft, aircraft, or a military vehicle.

**[0134]** The key fob 1230 may be preprogrammed to be paired with the property 1210 so that the key fob 1230 may communicate with the property 1210 via Bluetooth communication. Thus, no Internet connection may be necessary in the communication between the property 1210 and the key fob 1230. Further, the smart device 1220 may be configured to be paired with the key fob 1230 via Bluetooth communication. Thus, no Internet connection may be needed in the communication between the key fob 1230 and the smart device 1220.

**[0135]** The server 1240 may store the access history of the key fob 1230 in a hyperledger stored in a computing device of the property 1210. Thus, the server 1240 may utilize blockchain technology to generate blocks for storing the access history and save the block in the hyperledger stored in the servers 130a–n of FIG. 1 described above.

**[0136]** Returning to FIG. 12B, the smart device 1220 generates an encrypted code, which works as an authorized key to the entry 1215 of the property 1210. The smart device 1220 may transmit the generated encrypted code to the key fob 1230 at block 1250. In an aspect,

the smart device 1220 and the key fob 1230 may communicate with each other according to the Bluetooth protocol. In aspects, the smart device 1220 and the key fob 1230 may be paired via any suitable communication protocol. Accordingly, any nearby electronic gadgets cannot hack or hijack the encrypted code transmitted from the smart device 1220 to the key fob 1230. The communication between the smart device 1220 and the key fob 1230 may use any suitable communication protocol that ensures that communications are only between the paired devices.

**[0137]** At block 1260, the key fob 1230 may transmit the received encrypted code to the property 1210. The key fob 1230 may utilize Bluetooth in transmitting the encrypted code. Since Bluetooth communication is allowed between the paired devices, the encrypted code is not likely to be exposed to hacking or hijacking. In aspects, the smart device 1220 may transmit the encrypted code to the key fob constantly, sporadically, at regular intervals, or upon request by the key fob 1230. In some aspects, the key fob 1230 may change from being paired with the smart device 1220 to being paired with the computing device of the property 1210 in a case where both the smart device 1220 and the property 1210 are within communication range.

**[0138]** In an aspect, the smart device 1220 may encrypt the code by using end-to-end encryption (E2EE). For example, public and private keys may be used in the encryption. As such, the smart device 1220 may encrypt the code with the public key and the computing device of the property 1210 may decrypt the code received from the key fob 1230 with the private key. In this way, the E2EE may add security to the Bluetooth communication.

**[0139]** After decryption, the computing device of the property 1210 may determine whether the received code from the key fob 1230 is valid. Based on this determination, the property 1210 may grant or deny the key fob 1230 access to the property 1210 at block 1270. Further, the property 1210 transmits information regarding whether the key fob 1230 is granted or denied access to the property 1210, as an access log, to the smart device 1220 at block 1280. Furthermore, the property 1210 may transmit any activities in the access log to the smart device 1220 via the key fob 1230 at block 1280.

**[0140]** In an aspect, the property 1210 may include a validation module which performs the determination of validity of the encrypted code. The validation module may be trained by artificial intelligence (AI) or machine learning described herein.

[0141] When the smart device 1220 establishes a connection with the server 1240, the smart device 1220 may relay the access log to the server 1240 at block 1290. In turn, the server 1240 generates a block to save the access log therein and saves the block in a hyperledger.

[0142] In an aspect, the smart device 1220 may transmit the encrypted code to the property 1210 directly at block 1250. In response, the property 1210 may grant or deny the smart device 1220 access to the property 1210.

[0143] The key fob system 1200 may allow a third party to access the property 1210. FIG. 13 illustrate an aspect of the key fob system 1200, in which the smart device 1220 may be used to share access to the vehicle 1210'. As illustrated in FIG. 13, the vehicle 1210' is a vehicle but can be other types of property, such as an aircraft, watercraft, land vehicle, military vessel, building, room, etc. When the user of the smart device 1220 installs a mobile application thereon, the user has to register to the smart device 1220 at block 1305. When the registration succeeds, the smart device 1220 informs the user of the registration at block 1310.

[0144] The user opens the mobile application and provides login information to the mobile application at block 1315. If the login information is correct, the user may be able to login to the mobile application at block 1320. One or more menus are then displayed on the screen of the smart device 1220. Based on the user's selection, different functions may be performed at block 1325.

[0145] The list of menus may include start/stop engine, open/close door, open/close trunk, car key, and settings. This list of menus may be expanded or reduced according to the functionalities of the property 1210. When start engine is selected, the smart device 1220 may transmit the command to the property 1210, and when the command is valid, the vehicle 1210' starts its engine. When stop engine is selected and the command received from the smart device 1220 is valid, the vehicle 1210' stops or turns off the engine.

[0146] In a similar way, when open/close door or open/close trunk is selected and the command received from the smart device 1220 is valid, the corresponding car door or trunk door may be opened or closed. Successes and failures of these activities are saved in the smart device 1220, which may be transmitted to the server 1240 when a network connection is established between the smart device 1220 and the server 1240 at block 1330.

[0147] In an aspect, the command may include two parts: a code and a selected function. The two parts are encrypted by the smart device 1220 and the encrypted parts are transmitted to the vehicle 1210'. The vehicle 1210' then decrypts the encrypted parts and checks whether the decrypted code is valid. When the code is validated, the vehicle 1210' performs the selected function.

[0148] In another aspect, the encryption and decryption may be performed with a private and public key. Further, the communication between the smart device 1220 and the vehicle 1210' may be secured in the paired devices, e.g., devices paired via a Bluetooth connection.

[0149] In an aspect, the smart device 1220 may transmit the command to the vehicle 1210' via the key fob 1230. That is, the smart device 1220 may transmit the encrypted code to the key fob 1230. When the user of the key fob 1230 presses one of the buttons of the key fob 1230, the key fob 1230 may add the function, corresponding to the selected button, to the encrypted code and transmit the encrypted code to the vehicle 1210'. For example, the key fob 1230 may have several buttons including a first one for a car door, a second one for a car engine, a third one for a trunk door, etc. The button for the car door may be used to open or close the car door. For example, each press of the button may toggle between opening or closing the car door. In a further aspect, the key fob 1230 may include two buttons for the car door. For example, one button may be for opening the car door and the other button may be for closing the car door.

[0150] When the user selects a button of the key fob 1230, the key fob 1230 may combine the encrypted code received from the smart device 1220 and the function associated with the selected button. In an aspect, the key fob 1230 may decrypt the encrypted code received from the smart device 1220, combine and encrypt the decrypted code with the selected function to generate a command, and transmit the command to the vehicle 1210'. Upon reception of the command, the vehicle 1210' may decrypt the command, separate the code and the selected function from the decrypted command, and validate the code. When the code is not valid, the selected function is not performed by the vehicle 1210'.

[0151] In another aspect, the key fob 1230 may combine the encrypted code, which has not gone through decryption, with the selected function to generate a command, and transmit the command to the vehicle 1210'. In this case, the vehicle 1210' may decrypt only the encrypted code and validate the code.

[0152] When the user selects the access button on the screen of the smart device 1220 at block 1335, the user is able to grant or deny a third party access to the vehicle 1210'. The mobile application may provide a list of third parties to the user who can select one or more third parties from the list to grant or deny access to the vehicle 1210'.

[0153] At block 1340, information regarding the user's selection of third parties and the user's grant or denial of access is sent to the server 1240 so that the user's selection is saved in the hyperledger of the server 1240. In an aspect, the smart device 1220 may also send the encrypted code or the public key used in encrypting the code to the server 1240.

[0154] When the user grants a third party access to the vehicle 1210', the third party may be able to select one of the car key buttons displayed on a smart device of the third party at block 1345. The smart device of the third party may communicate with the server 1240 so as to receive the encrypted code or the public key for encryption from the server 1240. In an aspect, when the user of the smart device 1220 selects the grant button, the smart device 1220 may transmit the encrypted code to the smart device of the selected third party. In another aspect, the smart device of the selected third party may receive a guest access code, which can be validated by the vehicle 1210'.

[0155] In still another aspect, the selected third party may receive an updated access key whenever there are new attempts to access the vehicle 1210' by the selected third party. The access ability of the selected third party may be maintained until the user of the smart device 1220 denies the access of the selected third party or until the access key for the selected third party expires. In this regard, the access key may be valid for a period that the user of the smart device 1220 has set.

[0156] The key fob 1230 has been pre-programmed only to access the vehicle 1210'. Thus, a key fob of the selected third party may not be used to access the vehicle 1210' and the selected third party can only use his/her smart device to access the vehicle 1210'.

[0157] At block 1350, the smart device of the selected third party transmits the command including the encrypted code or the guest access code to the vehicle 1210' and the selected third party can access the vehicle 1210' after the command is validated. The smart device of the selected third party may send the access log to the server 1240 at block 1355 so that the access log may be saved in the hyperledger.

[0158] When the user of the smart device 1220 selects the settings button at block 1360, the user is able to change, update, or modify the current settings, personal information,

information about the vehicle 1210' which is paired with the smart device 1220 in Bluetooth communication, information about the key fob 1230 which is paired with the smart device 1220 in Bluetooth communication, or other settings. Such changes, modifications, or updates may be transmitted to the server 1240 at block 1355 so that they may be saved in the hyperledger.

**[0159]** When an unauthorized access is identified, the server 1240 may provide an alert to the user of the smart device 1220 via text message, email, telephone call, social networking service (SNS) message, audio alert, flash light, etc. so that the user can take a proper course of action.

**[0160]** In another aspect, due to usage of the Bluetooth communication, the smart device 1220, the vehicle 1210', and the key fob 1230 can communicate with each other even in the absence of internet connection. The smart device 1220 may need an internet connection to save the access log or history in the hyperledger of the server 1240.

## **SATELLITE COMMUNICATION SYSTEMS**

**[0161]** Provided in FIG. 14 is a satellite communication system 1400, which employs blockchain technology to secure communication data and FIG. 15 shows a flowchart illustrating a satellite communication method 1500 for the satellite communication system 1400 of FIG. 14 in accordance with aspects of this disclosure. The satellite communication system 1400 may allow data communication via a satellite after validating an access key. Thus, when the access key is not validated, data communication via the satellite is denied and the data communication is not initiated.

**[0162]** The satellite communication system 1400 may include a first station 1410, a second station 1420, and servers 1430. The first station 1410 may communicate with the second station 1420 via the servers 1430 and the satellite 1440. Validation results of the access key are saved in a server of the servers 1430. Further, the server of the servers 1430 may encrypt the data in the communication and save the encrypted data in a hyperledger 1435. The first station 1410 and the second station 1420 may include any computer, smart device, tablet, phablet, mobile device, server, or data center, which are capable of communicating with each other.

**[0163]** When data is needed by the second station 1420, e.g., to access an item of property, the second station 1420 transmits a request for an access code to the one or more servers 1430 via the satellite 1440 at block 1510. Specifically, the second station 1420

transmits the request for an access code by uploading the request through the uplink to the satellite 1440, which then relays the request to the one or more servers 1430. In an aspect, the first station 1410 may request that the one or more servers 1430 transmit an access key to the first station 1410 at block 1510, in a case when the first station 1410 transmits data to the second station 1420 or instruct the second station 1420 to do some tasks.

**[0164]** Upon receipt of the request, the one or more servers 1430 generate and transmit an access key to the second station 1420 at block 1520. The second station 1420 and one of the servers 1430 may communicate with each other by using an end-to-end encryption (E2EE) method with private and public keys. This ensures that the communication between the second station 1420 and one or more of the servers 1430 is secure. To transmit data, which includes the request or the access key, satellite dishes may be used between the second station 1420 and the satellite 1440 and between the one or more servers 1430 and the satellite 1440. Other communication systems and apparatuses may additionally or alternatively be utilized as is readily appreciated by a person having ordinary skill in the art.

**[0165]** At block 1530, the second station 1420 transmits the received access key to the satellite 1440 and, in turn, the satellite 1440 sends the access key to the first station 1410 at block 1540. After receiving the access key, the first station 1410 sends the received access key to the one or more servers 1430 for validation at block 1550.

**[0166]** At block 1560, the one or more servers 1430 determine whether or not the received access key from the first station 1410 matches the access key which the one or more servers 1430 sent to the second station 1420. When it is determined that the access keys match each other, the one or more servers 1430 validate the received access key and inform the first station 1410 of the validation. The first station 1410 then permits communication with the second station 1420. At block 1570, the first station 1410 and the second station 1420 start communicating with each other via the satellite 1440.

**[0167]** In a case when the one or more servers 1430 determine that the received access key does not match the access key which the one or more servers 1430 transmitted to the second station 1420 at block 1560, the one or more servers 1430 inform the first station 1410 that the received access key is not valid. At block 1580, the first station 1410 denies any communication with the second station 1420 with the received access key. When the second station 1420 is a valid station for communication, the second station 1420 may make another

attempt to receive another access code from the one or more servers 1430. Then, the satellite communication method 1500 repeats blocks 1510–1580.

**[0168]** In the cases of validation and non-validation, the one or more servers 1430 create a block for such determinations and save the block in the hyperledger.

**[0169]** In an aspect, when the same station continuously transmits an erroneous access key, the one or more servers 1430 may make a black list, which includes that station. The one or more servers 1430 may monitor the stations and may assign a status level to the stations based on the behavior of the stations. For example, when the one or more servers 1430 determine that the behavior of a station exceeds a threshold value, the one or more servers 1430 may change the status of the station from good to bad and/or may prevent the station from accessing the one or more servers 1430 until the station can provide further verification information. For example, when a station (e.g., the first station) unsuccessfully attempts to obtain one or more access keys a predetermined number of consecutive or total times (e.g., three times or five times), the one or more servers 1430 recognize the station as a bad station and informs the station of its updated status.

**[0170]** In another aspect, the access key may be valid for a predetermined period so that the second station 1420 may communicate with the first station 1410 during the predetermined period but cannot do so after the predetermined period elapses. This may prevent a station operator having ill purposes (e.g., a hacker or a hijacker) from attempting to obtain valid access in the future.

**[0171]** In a further aspect, the one or more servers 1430 may encrypt data, which is communicated between the first station 1410 and the second station 1420 via the satellite 1440. Particularly, when the first station 1410 transmits data to the second station 1420, the data may be transmitted to the one or more servers 1430 first. The one or more servers 1430 then encrypt the data and save the encrypted data in the hyperledger. Further, the one or more servers 1430 sends the encrypted data to the second station 1420 via the satellite 1440. The data saved in blocks of the hyperledger may be reference data, which is used to check whether the data has been hacked, modified, tempered, or removed.

**[0172]** In an aspect, each block may include a set amount of data, such as 1 mega bites (MB), 10 MB, 100 MB, or any size suitable for communication purposes between the first station 1410 and the second station 1420. To save storage space, the one or more servers 1430 may not create a block periodically. Rather, the one or more servers 1430 may create a

block whenever communication between the first station 1410 and the second station 1420 occurs. Further, the one or more servers 1430 may not store the data communicated via the satellite 1440 to save storage space. Instead, the first station 1410 may encrypt the data before transmitting to the second station 1420 via the satellite 1440 by using an E2EE with public and private keys.

**[0173]** In another aspect, the satellite 1440 may prohibit any communication initiated from the second station 1420, when the one or more servers 1430 inform of non-validation of the access code. In particular, the satellite 1440 may deny any data communication transmitted by the second station 1420 via the uplink. Thus, any further attempts by the second station 1420 cannot reach the first station 1410. At the same time, the satellite 1440 may allow any data communication transmitted by the first station 1410 to the second station 1420. That is any data communication through downlink to the second station 1420 is not denied. In this way, even if the access key from the second station 1420 is not validated, the first station 1410 may be able to send data to the second station 1420 via the satellite 1440. Nevertheless, the first station 1410 cannot receive any data from the second station 1420 when the access key is not validated.

**[0174]** FIG. 16 illustrates a military system 1600 using the satellite communication system 1400 of FIG. 14 in accordance with aspects of this disclosure. In this aspect, the first station 1410 of FIG. 14 is a military headquarters station 1610, the second station 1420 of FIG. 14 is a battlefield station 1620. In the same way as the first station 1410 communicates with the second station 1420, the battlefield station 1620 communicates with the military headquarters station 1610 by using an access key obtained from and validated by the server 1630. Thus, detailed descriptions regarding communication and validation between the military headquarters station 1610 and the battlefield station 1620 can be found in the above descriptions of FIG. 14.

**[0175]** When the battlefield station 1620 receives data or instructions from the military headquarters station 1610 via the satellite 1640, the battlefield station 1620 may relay the data or instructions to military aircraft, military marine ships, or military vehicles. As the data or instructions from the military headquarters station 1610 are encrypted, the battlefield station 1620 may decrypt the received data or instructions and perform any task according to the data or instructions.

[0176] By using the blockchain or hyperledger, all access history or communicated data can be securely saved and retrieved later as reference data for confirmation and validation. Due to the high security level inherently required in military communications, communication between the battlefield station 1620 and the military headquarters station 1610 may be encrypted using E2EE, which may be different from the encryption protocol used between the military headquarters station 1610 and the battlefield station 1620.

[0177] Any data returning to the battlefield station 1620 from the military vehicles may be transmitted to the military headquarters station 1610 and in turn saved in the hyperledger of the server 1630.

[0178] Returning to FIG. 14, the first station 1410 and the second station 1420 may be any servers of a business entity or private entity. For example, the first station 1410 and the second station 1420 may be research servers for sociology, biology, electrical engineering, material science, cosmology, astronomy, or any other areas of science or knowledge-based study. Alternatively, the first station 1410 and the second station 1420 may be any servers used for business purposes, such as financial transactions, stock trading, email communications, etc. In an aspect, the first station 1410 and the second station 1420 may be any personal computers, mobile devices, tablets, or any personal computing device used for communication via the satellite 1440.

[0179] Access key may be utilized in any communications between two parties, stations, computers, servers, any combination thereof with blockchain technology so that data communication is secured, and hacking and hijacking may be prevented with substantially high assurance and certainty. Application of the combination of the access key and blockchain may be used in other areas as readily appreciated by a person having ordinary skill in the art.

#### **APPLICATIONS OF THE SATELLITE COMMUNICATION SYSTEM**

[0180] Provided in FIG. 17 is an application of the satellite communication system 1400 of FIG. 14 to the aviation industry in accordance with aspects of this disclosure. An airport 1710, an airplane 1720, and a server 1730 correspond to the first station 1410, the second station 1420, and the one or more servers 1430 of FIG. 14, respectively. The airplane 1720 receives an access key from the server 1730 at block 1752 when the airplane 1720 is scheduled to fly to a destination. The access key may be valid for the entire flight.

[0181] The airplane 1720 takes off from the airport 1710 at block 1754 and encounters a potential hijacker at block 1756. In this case, the airplane 1720 may send an alert signal to the satellite 1740 using E2EE at block 1758 and may change the flight mode from manual to automatic at block 1760. Due to the imminent danger, the airplane 1720 may automatically lock the doors at block 1760. Since the alert signal is encrypted, the potential hijacker may not be able to listen to the encrypted alert signal.

[0182] The satellite 1740 may transmit the alert signal to the airport 1710 and the airport 1710 may also transmit the alert signal to the server 1730 so as to save the alert signal in the hyperledger.

[0183] At block 1762, the airplane 1720 may search for nearby airports (e.g., airports 1750a and 1750b) for emergency landing and may send a request for landing to the airports 1750a and 1750b at block 1764. If the airport 1750a rejects the request but the airport 1750b accepts the request, the airport 1750b transmits the acceptance to the airplane 1720 at block 1766.

[0184] The airplane 1720 is guided to land at block 1768 and lands on an airstrip of the airport 1750b at block 1770. Due to the emergency lock on the doors, passengers in the airplane 1720 cannot escape from the airplane 1720 at this time.

[0185] When landing is complete, the airplane 1720 may inform the airport 1710 of the landing via the satellite 1740 at blocks 1772–1776. The airport 1710 may generate and encrypt an unlock code and send it to the airplane 1720 via the satellite 1740 at blocks 1780–1784.

[0186] Upon reception of the encrypted unlock code, the airplane 1720 may decrypt the encrypted unlock code and unlock the doors of the airplane 1720. In this application, the server 1730 may save all history including the potential hijacking event, emergency signal, unlock code, and any related events in the hyperledger. Thus, by using the hyperledger based on the blockchain, data security in the aviation industry is increased while hijacking risks are decreased.

[0187] As another application of the satellite communication system 1400 of FIG. 14, provided in FIG. 18 is a marine system in accordance with aspects of this disclosure. A ground station 1810, a water vehicle 1820, and a server 1830 correspond to the first station 1410, the second station 1420, and the one or more servers 1430 of FIG. 14, respectively.

The water vehicle 1820 receives an access key from the server 1830 before leaving a port. The access key may be valid for the entire trip taken by the water vehicle 1820.

**[0188]** The water vehicle 1820 departs from the ground station 1810 and may encounter a potential pirate. In this case, the water vehicle 1820 may send an alert signal to the satellite 1840 using E2EE at block 1872. Since the alert signal is encrypted, the potential pirate boat may not be able to listen to the encrypted alert signal.

**[0189]** The satellite 1840 may transmit the alert signal to the ground station 1810 via a satellite dish at blocks 1876 and 1878. The ground station 1810 may transmit the alert signal to a naval security agency 1860 at block 1880, which instructs a security vehicle 1870 to save the water vehicle 1820. The ground station 1810 may also send the alert signal to the server 1830 so as to save the alert signal in the hyperledger. Further, the ground station 1810 may transmit all history including the potential piracy, emergency signal, and any related events in the hyperledger. Thus, by using the hyperledger based on the blockchain, data used in the marine industry is secured while piracy risks are minimized.

**[0190]** FIG. 19 is a block diagram illustrating a wireless e-key architecture in accordance with aspects of this disclosure. The wireless e-key architecture includes various building blocks of an e-key system and how they are connected to each other to collect, store, and process data.

**[0191]** The IoT architecture may contain one or more of the following components: things 1901 equipped with sensors to gather data and/or actuators to perform commands received from the cloud 1904 (e.g., key fobs, smart home devices, baby monitors, currency, passports, drivers licenses, and other IoT-enabled devices); gateways 1902 for filtering, preprocessing, and moving data to the cloud 1904, and for receiving control data 1918 including commands from the cloud 1904; cloud gateways 1906 to ensure that data is properly transferred between field gateways and central servers; streaming data processors 1910 to distribute the data coming from sensors among relevant e-key system components; a data lake 1912 for storing data including defined values, undefined values, and/or sensor data 1911; a big data warehouse for collecting valuable data; control applications 1916 to send control data 1918 including commands to actuators; machine learning 1920 to generate the models 1922 which are then used by the control applications 1916; user web applications 1932 and user mobile applications 1934 to

enable users to monitor and control their connected things 1901; and data analytics 1940 for manual data processing.

[0192] Things 1901, e.g., e-key objects, may be equipped with sensors that gather data, which is transferred over a secured network (including, for example, a Bluetooth or Wi-Fi connection) and actuators that allow things to act (for example, to switch an engine on or off, to open or close a door, or to open or close a trunk).

[0193] The e-key system may utilize gateways 1902. Data may be transmitted between things 1901 (e.g., Internet of Things (IoT)-enabled devices) and the cloud 1904 through the gateways 1902. A gateway 1902 provides connectivity between things 1901 and the cloud part of the IoT solution, enables secured data preprocessing and filtering before moving data to the cloud 1904 (e.g., to reduce the volume of data for detailed processing and storing), and transmits control data 1918, e.g., commands, from the cloud 1904 to the things 1901. Things 1901 then, for example, execute the commands using their actuators.

[0194] The e-key system may utilize cloud gateways 1906, e.g., API-Cloud Gateways, which facilitate data compression and secure data transmission between field gateways 1902 and IoT servers of the cloud 1904. The cloud gateways 1906 also ensure compatibility with various protocols and communicate with field gateways 1902 using different protocols depending on what protocol is supported by the gateways 1902, 1906.

[0195] The e-key system may incorporate a streaming data processor 1910, which ensures effective transmission of input data or sensor data 1911 to the data lake 1912 and control applications 1916 and ensures that no data is lost or corrupted.

[0196] The e-key system may also utilize a data lake 1912. The data lake 1912 may be used for storing the data generated by connected devices in its natural format. Big data may come in “batches” or in “streams”. When the data is needed for meaningful insights or analytics (e.g., in response to a query), the data may be extracted from a data lake 1912 and loaded into a central system 1915, e.g., a big data warehouse.

[0197] The e-key system may also utilize machine learning (ML) 1920 and the models 1922 that ML 1920 generates. With machine learning 1920, there is an opportunity to create more precise and more efficient models for the control applications 1916. The models 1922 may be regularly updated (for example, once a week or once a month) based on the historical data accumulated in the central system 1915, e.g., a big

data warehouse. When the applicability and efficiency of new models are tested and approved by data analysts and/or data analytics processes 1940, new models are used by control applications 1916.

**[0198]** The e-key system may include control applications 1916 that send commands and alerts to actuators. For example, the e-key system can receive an automatic command to an actuator to open or close a door depending on a user's interaction with a user application taken from the user.

**[0199]** The e-key system may include control or central applications, which can be either rule-based applications or machine-learning-based applications. The rule-based applications may include control applications working according to the rules stated by specialists. The machine-learning-based applications may include control applications that use models 1922 which are regularly updated (e.g., once a week or once a month depending on the specifics of the e-key system) with the historical data stored in a big data warehouse.

**[0200]** Although control applications ensure better automation of an e-key system, the e-key system may provide an option for users to influence the behavior of such applications.

**[0201]** User applications, e.g., web applications 1932 or mobile applications 1934, may be software components of the e-key system which enables the connection of users to the e-key system and gives the option to centrally monitor and control smart things while they are connected to a network of similar things, e.g., cars. With web applications 1932 or mobile applications 1934, users can monitor the state of their things 1901, send commands to control applications 1916, set the options of automatic behavior (e.g., automatic notifications and actions when certain sensor data 1911 comes from the sensors).

**[0202]** The e-key system may include device management utilities to ensure sufficient and proper functioning of e-key devices. There may be some processes and/or procedures required to manage the performance of connected devices (e.g., to facilitate the interaction between devices and ensure secure data transmission).

**[0203]** The e-key system may include device identification routines to establish the identity of the device to be sure that the device is a genuine device with trusted software transmitting reliable data.

[0204] The e-key system may include configuration settings and/or controls to tune things, e.g., IoT-enabled devices, according to the purposes of the e-key system. Some parameters may need to be written once a device is installed (e.g., unique device ID or Blockchain Token). Other settings may need updates (e.g., the time between sending messages with data).

[0205] The e-key system may include monitoring and diagnostic features to ensure smooth and secure performance of everything, e.g., IoT-enabled devices, in a network and reduce the risk of breakdowns.

[0206] The e-key system may be configured to perform software updates and maintenance to add functionality, fix bugs, and/or address security vulnerabilities.

[0207] The e-key system may include user management features, which, along with the device management features, provide control over the users having access to the e-key system. The user management features may involve identifying users, their roles, access levels, and/or ownership in a system. The user management features may include options such as adding and removing users, managing user settings, controlling access of various users to certain information, providing the permissions to perform certain operations within a system, and/or controlling and recording user activities and more.

[0208] The e-key system may include security monitoring features. Connected things produce large volumes of data, which need to be securely transmitted and protected against attacks from cyber criminals. Another aspect is that the things connected to the Internet can be entry points for villains. What is more, cyber-criminals may get the access to the “brain” of the e-key system and take control of it. To prevent such attacks, the various components of the e-key system may log and analyze the commands sent by control or central applications to things, monitor the actions of users, and store this data in the cloud. These features may allow the e-key system to address security breaches at the earliest stages and take measures to reduce their influence on the e-key system. For example, the e-key system may block certain commands coming from control applications. Also, the e-key system may identify patterns of suspicious behavior, store these samples, and compare them with the logs generated by the e-key system to prevent potential penetrations and minimize their impact on the e-key system.

[0209] The e-key architecture may also include device and user management components to provide stable and secure functioning of things and control user access.

[0210] The e-key architecture may provide consistency (e.g., giving enough attention to every element of the e-key architecture and making them work together), flexibility (e.g., providing opportunity to add new functions and new logic), and integrability with enterprise systems.

[0211] The e-key system may include a 64-bit high performance expandable single board computer (SBC) which can be integrated with any IoT gateway (e.g., gateways 1902). A group of system nodes may have the computing power of a supercomputer. The smart chips may be incorporated into cards or any other objects suitable for applications such as access control, authentication, and security key storage. For example, as shown in FIG. 22, a smart card may incorporate a smart chip of the smart chip system of FIG. 21.

[0212] The smart chips may include embedded memory. This disclosure provides details regarding a smart chip reader system and its usage with objects incorporating smart chips.

[0213] The system builds on the current trends of automation and data exchange in manufacturing and industrial technologies. The system includes technologies such as 5G and 6G, and, in future, the system can integrate 7G Data Connectivity, Cyber Security Systems, Cyber-Physical Systems, the IoT, Cloud Computing, Blockchain, and AI Cognitive Computing. Within modular structured smart factories, cyber-physical systems monitor physical processes, create a virtual copy of the physical world, and make decentralized decisions. Over the IoT, cyber-physical systems communicate and cooperate with each other and with humans in real time both internally and across organizational services offered and used by participants of the chain.

[0214] The system contains two major hardware sections. The first hardware section, which may be referred to as the smart chip, acts as a passive authentication and application input client. The first hardware section is activated when it is in contact with or in the range of the reading terminal. The second hardware section, which may be referred to as the smart chip reader, receives input data, processes it, and returns a response.

[0215] The system may include a contactless smart chip module that generates the necessary control signals on the smart chip side. The commands or data for accessing the smart chip module are fed in either by a microcontroller interface or the configuration logic. The IP interface block diagram is shown in FIG. 21.

[0216] The smart chip module illustrated in FIG. 24 includes two memory areas: the application zone and the configuration memory area. The application zone may be divided

[0217] into sixteen zones of 1,024 bits each. Access to the application zones is allowed after security requirements have been met. The security requirements may be defined by the system software during the device personalization process and are stored in the configuration memory.

[0218] The configuration memory may include 2,048 bits of EEPROM memory used for storing passwords, blockchain tokens, token keys, client programming and codes, and for defining security levels to be used for each application zone. Access rights to the configuration memory are defined in the control logic and may not be altered by the system settings.

[0219] A key may be used for accessing the supplied smart chip module’s configuration memory. For example, the key may be 0xB6A405. This key may be required for unlocking the configuration memory for any modification made to the configuration memory. A user may access the user zones and the configuration memory area in the manner described herein.

[0220] The operating procedure may include the following operations:

1. Connection and activation of the contacts by the interface device.
2. Reset the card.
3. Answer to reset by the card
4. Information exchange between the card and the interface device.
5. Deactivation of the contacts by the interface device.

Address ADDR[2:0]	Type	Width	Reset Value	Name
0x2	Write	8	0	Transmit Data Register
0x3	Read	8	0	Receive Data Register
0x4	Read	8	0	Status Register
0x5	Write	8	0	Reset Register

**Table 1**

[0221] As shown in Table 1 above, there may be four internal registers in the interface device: Reset Register, Transmit Data Register, Receive Data Register, and Status Register. The Reset Register resets the smart card. The Transmit Register transmits data to the CARD\_IO bus. The Receive Register receives data from the CARD\_IO bus. The

Status Register indicates the status of data transaction. There may also be two counters: Counter and Bit Counter. The counter is used to sample each bit at the rate  $f$  (e.g., 1 MHz) at the middle of the bit. Bit Counter is used to count 11 bits of a packet. Examples of offset addresses for the smart card reader registers are shown in Table 1 above.

**[0222]** As illustrated in FIG. 23, the smart chip includes a power supply contact (VCC), a reset signal contact (RST), which is used to reset the smart card's communications, a clock contact (CLK), which provides the smart card with a clock signal, a ground contact (GND), which provides a ground or reference voltage, a programming voltage contact (VPP), through which a programming voltage is provided, an input/output contact (I/O), which provides a half-duplex serial input and output, and C4 and C8 contacts, which are the two remaining contacts used for USB interfaces and other functions.

**[0223]** FIG. 24 illustrates a smart chip circuit block diagram. The smart chip circuit includes a processor, which, for example, may be an processor, random access memory (RAM), which may be configured to store a token code, a read-only memory (ROM), which may be configured to store a hashcode, e.g., a 128 bit hashcode, an electronically erasable programmable read-only memory (EEPROM), which may be configured to store the blockchain token and the client application, a power supply contact through which the smart chip is powered, and an input/output (I/O) contact, which is configured for serial input and output communications. FIG. 25 shows an example of a microcontroller of the smart chip.

**[0224]** FIG. 26 is a block diagram illustrating the smart chip reader of the smart chip system of FIG. 21. The smart chip reader includes an address decoder, a status register, a reset register, a transmit data register, a transmit shift register, a receive data register, a receive shift register, a counter, and a bit counter. The counter provides a count signal to the status register, the receive shift register, and the bit counter. The bit counter provides a bit count signal to the transmit shift register. The transmit data register is coupled to data input line, e.g., an eight-bit data input line. The transmit shift register and the receive shift register are connected to a card input/output line.

**[0225]** The Transmit Data Register contains data that may be transmitted to the smart chip. Examples of the Transmit Data Register bit settings are shown in Table 2 below.

7	6	5	4	3	2	1	0
TX DATA[7]	TX DATA[6]	TX DATA[5]	TX DATA[4]	TX DATA[3]	TX DATA[2]	TX DATA[1]	TX DATA[0]

Table 2 - Transmit Data Register

[0226] The receive data register may contain the data that is received from the smart chip. Examples of the receive data register bit settings are shown in Table 3 below.

7	6	5	4	3	2	1	0
RX DATA[7]	RX DATA[6]	RX DATA[5]	RX DATA[4]	RX DATA[3]	RX DATA[2]	RX DATA[1]	RX DATA[0]

Table 3 - Receive Data Register

[0227] Examples of the Status Register bit settings are shown in Table 4 below.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	TxRx_DONE

Table 4 - Status Register

[0228] By default, the content of the Status Register may be 0x00. When a byte of data is transferred out of the transmit register and shifted out serially through the card I/O pin to the smart chip, the content of the status register may change to 0x01. This bit may be monitored by the host to confirm that the data has been completely shifted out to the smart chip. Once the complete byte is shifted out of the transmit register, the status register bit is set to 0x01. The content may be cleared to 0x00 before the next data transfer. Similarly, for the read operation, the status register bit may be monitored to check whether any data is available in the receive register. Once the data is read from the smart chip and a byte of information is available in the receive register, the status register bit becomes 1 and may be cleared by the host before receiving the next byte.

[0229] Examples of the reset register bit settings are shown in Table 5 below.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	CARD_RST

Table 5 - Reset Register

By default, the reset register content may be 0x00. For any transaction to be performed, the initial step may be to reset the chip. For normal operation, reset register bit may be set to 1.

**[0230]** FIGS. 27A and 27B show a flowchart illustrating a method of reading or writing to an application zone of the smart chip in accordance with aspects of this disclosure. After initiating the system reset process at block 2701, the Smart Card or Object reset (Card\_rst) is applied to Smart Chip by writing 0x01 in the reset register (e.g., the reset register illustrated in Table 2 above) at block 2702. At block 2704, eight bytes of Smart Chip Answer To Reset (ATR) is obtained from the receive data register (e.g., the receive data register illustrated in Table 3 above). At block 2706, five bytes of the command for “Set Application Zone” is sent to the transmit data register (e.g., the transmit data register illustrated in Table 2 above). At block 2708, a Smart Chip Acknowledgement byte (same as INS byte of command sent in previous block) is obtained from the Receive Data Register. At block 2710, Smart Chip Status bytes Status Word 1 and Status Word 2 are obtained from the receive data register (e.g., the receive data register illustrated in Table 3 above).

**[0231]** Then, at block 2712, it is determined whether to read from the “Read Application Zone” of the smart chip or write to the “Write Application Zone” of the smart chip. This determination may be based on the task that is being performed. For example, when authenticating the identity of a person the smart chip would be read via blocks 2714–2723 or when updating the hyperledger with status information the status information would be written to the smart chip via blocks 2724–2733.

**[0232]** If reading from the “Read Application Zone” is determined to be performed at block 2712, five bytes of the command for “Read Application Zone” is sent to the transmit data register at block 2714. At block 2716, a Smart Card Acknowledgement byte (same as INS byte of command sent in previous block) is obtained from the receive data register. At block 2718, N bytes of data to be written in application zone is sent to the transmit data register. At block 2720, smart chip status bytes SW1 and SW2 are obtained from the receive data register. At block 2722, the smart card reset setting (Card\_rst) is removed from the smart chip by writing 0x00 in the reset register (e.g., the reset register illustrated in Table 5 above). At block 2723, the read application zone cycle is ended.

**[0233]** If writing to the “Write Application Zone” is determined to be performed at block 2712, five bytes of the command for “Write Application Zone” is sent to the transmit data

register at block 2724. At block 2726, a smart card acknowledgement byte (same as INS byte of command sent in previous block) is obtained from the receive data register. At block 2728, N bytes of data to be written in application zone is sent to the transmit data register. At block 2730, smart chip status bytes SW1 and SW2 are obtained from the receive data register. At block 2732, the reset smart card setting (Card\_rst) is removed from the smart chip by writing 0x00 in the reset register (e.g., the reset register illustrated in Table 5 above). At block 2733, End of Write Application Zone Cycle.

**[0234]** FIGS. 28A and 28B show a flowchart illustrating a method of reading or writing to configuration memory in accordance with aspects of this disclosure. After initiating the system reset process at block 2801, the Smart Card or Object reset (Card\_rst) is applied to Smart Chip by writing 0x01 in the reset register (e.g., the reset register illustrated in Table 2 above) at block 2802. At block 2804, eight bytes of Smart Chip Answer To Reset (ATR) is obtained from the receive data register (e.g., the receive data register illustrated in Table 3 above). At block 2806, five bytes of the command for “Unlock Configuration Zone” is sent to the transmit data register (e.g., the transmit data register illustrated in Table 2 above). At block 2808, a Smart Chip Acknowledgement byte (same as INS byte of command sent in previous block) is obtained from the Receive Data Register. At block 2810, three bytes of security code for unlocking the configuration zone is sent to the transmit data register (e.g., the transmit data register illustrated in Table 2 above).

**[0235]** Then, at block 2812, it is determined whether to read from the “Read Application Zone” of the smart chip or write to the “Write Application Zone” of the smart chip. This determination may be based on the task that is being performed. For example, when authenticating the identity of a person the smart chip would be read via blocks 2814–2823 or when updating the hyperledger with status information the status information would be written to the smart chip via blocks 2824–2833.

**[0236]** If reading from the “Read Configuration Zone” is determined to be performed at block 2812, five bytes of the command for “Read Config Mem” (i.e., read configuration memory or zone) is sent to the transmit data register at block 2814. At block 2816, a smart card acknowledgement byte (same as INS byte of command sent in previous block) is obtained from the receive data register. At block 2818, N bytes of data is obtained from configuration memory’s receive data register. At block 2820, smart chip status bytes SW1 and SW2 are obtained from the receive data register. At block 2822, the smart card reset

setting (Card\_rst) is removed from the smart chip by writing 0x00 in the reset register (e.g., the reset register illustrated in Table 5 above). At block 2823, the read from configuration memory cycle is ended.

[0237] If writing to the “Write Configuration Zone” is determined to be performed at block 2812, five bytes of the command for “Write Config Mem” (i.e., write configuration memory or zone) is sent to the transmit data register at block 2824. At block 2826, a smart card acknowledgement byte (same as INS byte of command sent in previous block) is obtained from the receive data register. At block 2828, N bytes of data to be written in the configuration memory is sent to the transmit data register. At block 2830, smart chip status bytes SW1 and SW2 are obtained from the receive data register. At block 2832, the reset smart card setting (Card\_rst) is removed from the smart chip by writing 0x00 in the reset register (e.g., the reset register illustrated in Table 5 above). At block 2833, the write to configuration memory cycle is ended.

[0238] FIG. 29 is block diagram of a smart chip system 2900 incorporated into a wireless e-key system architecture in accordance with aspects of this disclosure. The smart chip system 2900 includes a hyper terminal 2902, a single board computer (SBC) 2904, the smart chip reader 2102, and the smart chip 2104, which may reside on a card or any other suitable object. The SBC 1904 may be not only a computer, but also an affordable 64-bit high performance expandable single board computer which can be integrated with any IoT gateway. A group of SBC nodes may have the computing power of a supercomputer. The SBC 2904 may come with a preloaded Linux operating system, blockchain, and AI infrastructure which can be integrated with any IoT gateway, e.g., biometrics, camera, sensors, etc.

[0239] As illustrated in FIG. 33, the SBC 2904 may include one or more of the following features: CPU, GPU, RAM memory, FLASH memory (e.g., NAND FLASH memory), eMMC memory, ability to work at any extreme industrial temperature and weather, SATA connector with SATA power jack, HDMI port, USB Low-Full-High-Speed hosts with power control and current limiter, USB-OTG with power control and current limiter, VGA output, native Ethernet, battery connector with battery-charging capabilities, audio headphones output, microphone input on connector, UEXT connectors, LCD connector compatible with LCD modules of different sizes, GPIO connectors, MicroSD card connector, SD/MMC card connector, DEBUG-UART connector for console debug, status LED, battery charge status

LED, Power LED, EEPROM for MAC address storage, etc., BUTTONS with ANDROID functionality and RESET button, mount holes, input power supply, Wi-Fi 802.11n, 5G & 6G GSM Board, expandable IoT Board, and heat sink.

**[0240]** FIGS. 30A–30I illustrate examples of the IoT gateway modules 1902 of FIG. 19. The SBC 2904 can be deployed in thousands of systems across a wide range of industries and applications using expandable IoT gateway modules 1902.

**[0241]** FIG. 30A is a block diagram illustrating examples of input components of a biometric system incorporated into the e-key architecture in accordance with an aspect of this disclosure. The input components may include a fingerprint sensor, fingerprint reading sensor, an iris reading sensor, a facial recognition system, a smart chip reading sensor, which may be configured to read a smart chip in a passport, a smart chip in a credit or debit card, and/or a smart chip in a driver's license.

**[0242]** FIG. 30B is a block diagram illustrating examples of input components of a telecommunication system incorporated into the wireless e-key architecture in accordance with an aspect of this disclosure. The input components may include a GSM or CDMA SIM card. The input components may be enabled for 5G and 6G technologies. Every event may be recorded in the blockchain. The telecommunication system may feature reliable and secure communication, cheap and fast communication, and security features making it difficult to clone or duplicate mobile phones or smartphones.

**[0243]** FIG. 30C is a block diagram illustrating examples of input methods of a supply chain management system incorporated into the wireless e-key architecture in accordance with an aspect of this disclosure. The input methods include unloading the supply chain item, measuring the supply chain item dimensions, finding a location to place the supply chain item, recording the event in a blockchain, and storing the supply chain item. After storing the supply chain item, the input method may further include finding and retrieving the supply chain item, managing inventory of supply chain items, and identifying supply chain items using a barcode, a QR code, an RFID object, or a three-color code.

**[0244]** FIG. 30D is a block diagram illustrating examples of an input method of a satellite system incorporated into the e-key architecture in accordance with an aspect of this disclosure. The SBC 2904 may be installed on a satellite. According to the input method, every event or data transfer is recorded in a blockchain. The input method eliminates man-in-the-middle attacks and features peer-to-peer communications. During communications,

tampered with data is resent. According to the input method, the ground station acts as the hyperledger. The input method provides reliable and secure data transfer as well as fast and cheap data transfer.

[0245] FIG. 30E is a block diagram illustrating examples of input components of an automobile system incorporated into the wireless e-key architecture in accordance with an aspect of this disclosure. The input method may involve actuators and/or control devices configured to perform one or more of opening/closing a car door, opening/closing a car trunk, starting/stopping a car engine, turning on/off a car's air conditioning, and generating an alert or alarm to check car fuel. The input method may include locating a car, checking tire air pressure, and accessing logs.

[0246] FIG. 30F is a block diagram illustrating examples of input methods of an aviation system incorporated into the wireless e-key architecture in accordance with an aspect of this disclosure. The input methods may include monitoring whether the cockpit is breached and performing actions in response to a breach event. The input method may include causing an airplane to go into autopilot mode in response to an event. The event may also automatically cause one or more of the following actions to occur: send an alert to officials and/or the police department, gather biological details (e.g., health information, blood type, etc.) of passengers, and initiate flight in-order system.

[0247] FIG. 30G is a block diagram illustrating examples of input methods of a healthcare system incorporated into the wireless e-key architecture in accordance with an aspect of this disclosure. The input methods may involve healthcare sensors and may include gathering biological details of patients, scheduling appointments, continuously screening patients, and suggesting medical procedures. In response to detecting events or conditions during the continuous screens, one or more of the following actions, among others, may be performed: automatically transmitting alerts to hospital officials, automatically recording all patient health data in a blockchain, and automatically transmitting medication alerts to patients.

[0248] FIG. 30H is a block diagram illustrating examples of input method of a fintech system incorporated into the wireless e-key architecture in accordance with an aspect of this disclosure. The input method may include inputting the receiver address, entering a private key, adding the data to a new block of a blockchain, validating, by all nodes, the transaction,

and obtaining, by the receiver, the transaction. The input method may also include sending smart contracts. This input method allows for peer-to-peer and secure transactions.

[0249] FIG. 30I is a block diagram illustrating examples of input methods of a pharmaceutical industry system incorporated into the e-key architecture in accordance with an aspect of this disclosure. The input method may include inputting a drug barcode, entering a private key, adding drug data to a new block of a blockchain, validating, by all nodes, the transaction, and obtaining, by the receiver, the drugs. The input method may also include one or more of tracking drug cargo and sales, tracking drugs by batches, and finding substitute medicines, among others.

[0250] FIG. 31 is a block diagram illustrating an IoT gateway module incorporated into the e-key architecture in accordance with aspects of this disclosure. The IoT gateway may be implemented by software and/or hardware that causes sensors to obtain sensor data, reading the sensor data, processing the sensor data, and sending the sensor data to the single board computer (SBC) 2904 of FIG. 29.

[0251] FIG. 32 is a block diagram illustrating a single board computer (SBC) interface module incorporated into the wireless e-key architecture in accordance with aspects of this disclosure. As shown in FIG. 32, the system includes three software interfaces: (1) a firmware interface 3210, (2) a system software interface 3220, and (3) an application interface. The firmware interface 3210 may be permanent and may load up the configuration files and initial data to bootup the hardware. The firmware interface 3210 may be divided into different levels of firmware. The low-level firmware 3212 may be stored in ROM, OTP/PROM, and/or PLA structures. In some aspects, the low-level firmware 3212 is stored in read-only memory (ROM) and cannot be changed or updated. The low-level firmware 3212 may include hardware or driver software.

[0252] The high-level firmware 3214 may be stored in flash memory and may include system software and system software updates. The subsystem firmware 3216 may include fixed microcode embedded in flash chips, CPUs, and/or LCD units. The subsystem firmware 3216 may be considered part of the hardware device and/or the high-level firmware 3214.

[0253] The system software interface 3220 may boot up the core operating system 3221 of the board and may also preload all required system files 3222, system services 3223, system preferences 3224, including libraries of functions, hardware drivers 3225

including drivers for the board and drivers for other IoT hardware, security software 3226, other configuration files 3227, and/or firewalls including an artificial intelligence firewall. The system software may also include assemblers, compilers, file management tools, system utilities, and/or debuggers.

**[0254]** The application interface may come with preloaded infrastructure and functionalities of the blockchain, AI interface, and IoT. The application interface may support C and Python-based programs and applications. The AI interface may act as a smart firewall, which monitors the passive and active inputs and outputs, or the events triggered by the users or the system, and the AI interface makes wise decisions. If the system identifies any potential threat to the system itself or to the physical control panels being hacked or tampered, the system may enter a safe mode, alert appropriate people (e.g., officials) and go into auto mode.

**[0255]** The system incorporates blockchain functions allowing users to add immutable data into the hyperledger and ensuring integrity and provenance of the data including audio streams, video streams, text messages, images, and documents being exchanged as smart contracts from one peer to another peer. The system verifies and authenticates the data.

**[0256]** Assuming, for example, that a man-in-the-middle-attack is performed between two peers, after receiving tampered with or corrupted data, the receiver peer decrypts the data and sends back an acknowledgement token to the sender peer. If the acknowledgement token is found to be invalid, the sender peer resumes sending the valid data.

**[0257]** FIG. 34 is a block diagram illustrating the application modules 3400 that may be run on the SBC 2904 of FIG. 29 in accordance with aspects of this disclosure. The hosting platform 3402 contains several API endpoints and storage functions which are used to give requests and get responses from online services. Various hosting services may be used. For example, AWS EC2 m5.12xlarge instance may be used as the application server and AWS S3 instance may be used for file storage purposes. Amazon AWS EC2 m5.12xlarge instance provides a wide selection of cloud computing instance types optimized to fit different use cases. Instance types comprise varying combinations of 48 vCPU, 192GB memory, EBS storage, 7,000 Dedicated EBS Bandwidth (Mbps) and 10 networking Network Performance (Gbps) capacity and give the flexibility to choose the appropriate mix of resources for

applications. Each instance type includes one or more instance sizes, allowing for the scaling of resources to the requirements of the target workload.

**[0258]** Amazon AWS S3 instance provides easy-to-use management features to organize data and configure finely tuned access controls to meet specific business, organizational, and compliance requirements. Amazon AWS S3 is designed for 99.999999999% (11 9s) of durability, and stores data for millions of applications for companies all around the world.

**[0259]** The server interface 3404 may come with several preloaded servers such as Gunicorn, Redis messaging queuing server, and Nginx web server. The Gunicorn server is broadly compatible with a number of web frameworks, simply implemented, light on server resources, and fairly fast. The Redis Messaging Queuing Server is an in-memory database project implementing a distributed, in-memory key-value store with optional durability. Redis supports different kinds of abstract data structures, such as strings, lists, maps, sets, sorted sets, hyperloglogs, bitmaps, and spatial indexes. The Nginx Web Server is a high-performance HTTP server and reverse proxy, as well as an IMAP/POP3 proxy server. NGINX has high performance, stability, a rich feature set, a simple configuration, and low resource consumption.

**[0260]** The application modules 3400 may include a client application interface 3406, which may use Django 2.1 and Django REST framework. The Django Web Framework is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. The Django REST Framework is a powerful and flexible toolkit for building Web APIs. The Django REST Framework provides web-browsable APIs, authentication policies including packages for OAuth1a and OAuth2, and serialization that supports both ORM and non-ORM data sources.

**[0261]** The bigdata interface or platform 3408 contains various powerful bigdata tools. The bigdata tools may include Apache Hadoop, which is a collection of software utilities that facilitate using a network of many computers to solve problems involving massive amounts of data and computation. The bigdata tools may also include Apache Pig, which is a platform for analyzing large data sets that consist of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating those programs. Pig may run on Apache Hadoop YARN and may make use of MapReduce and the Hadoop Distributed File System (HDFS). The language for the platform is called Pig Latin, which abstracts from the Java MapReduce idiom into a form similar to SQL.

[0262] The bigdata tools may also include Apache Hive, which is a data warehouse software project built on top of Apache Hadoop for providing data query and analysis. Hive gives a SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop. The bigdata tools may also include MongoDB, which is a database for modern apps, and MongoDB Atlas, which is a global cloud database on AWS, Azure, and GCP. These databases easily organize, use, and enrich data in real time, anywhere. MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schema.

[0263] The bigdata tools may also include Apache Storm, which is a free and open source distributed real-time computation system. Storm makes it easy to reliably process unbounded streams of data, doing for real-time processing what Hadoop does for batch processing. The bigdata tools may also include Apache Spark, which is a distributed general-purpose cluster-computing framework. Spark provides an interface for programming entire clusters with implicit data parallelism and fault tolerance.

[0264] The application modules 3400 may include a blockchain platform or interface 3410, which is a distributed, decentralized, public ledger. The blockchain interface 3410 may include software to facilitate the following process: inputting, by the sender, receiving address; adding, by the sender, the data; inputting, by the sender, his private key to validate the data; including, by the miner, data into the next block; confirming, by the nodes, that the data is valid; and receiving, by the receiver, the data within a short time, e.g., a few seconds. The blockchain may include a hyperledger and a blockchain explorer.

[0265] The hyperledger may incorporate distributed ledger technology, which may be a database that is shared amongst computers spread around the globe, creating a decentralized environment rather than a centralized one. There are three types of hyperledgers that may be used: blockchain, directed acyclic graph, and hash graph.

[0266] The blockchain explorer may be a simple, powerful, easy-to-use, well-maintained utility to browse activity on the underlying blockchain network. The blockchain explorer may not perform any event actions or interacts with AI and IoT modules. The blockchain explorer may be a web dashboard which enables system users to view all blockchain activities.

[0267] The artificial intelligence modules 3412 may include various artificial intelligence (AI) libraries and techniques to build and drive smart and machine intelligence. The AI libraries may include NumPy, which is a library for the Python programming language, and

which adds support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The AI libraries may also include SciPy, which is a Python library used for scientific computing and technical computing. The AI libraries may also include Scikit-learn, which is a machine learning library for the Python programming language. Scikit-learn features various classification, regression, and clustering algorithms including support vector machines, random forests, gradient boosting, kmeans, and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

**[0268]** The artificial intelligence modules 3412 may also include TensorFlow, which is an end-to-end machine learning. TensorFlow has a comprehensive, flexible ecosystem of tools, libraries, and community resources that allows users to push the state-of-the-art in ML and developers to easily build and deploy ML-powered applications. The AI libraries may also include Keras, which is a Python Deep Learning library. Keras is a high-level neural network API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. The AI libraries may also include Theano, which is a Python library that allows a user to define, optimize, and efficiently evaluate mathematical expressions involving multi-dimensional arrays. Theano is built on top of NumPy and features tight integration with NumPy and a similar interface to NumPy's `numpy.ndarrays` may also be used internally in Theano-compiled functions.

**[0269]** The AI libraries may also include Pandas, which is a Python data analysis library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. The artificial intelligence modules 3412 may also include Matplotlib, which tries to make easy things easy and hard things possible. A user can generate plots, histograms, power spectra, bar charts, error charts, scatterplots, etc., with just a few lines of code.

**[0270]** It should be understood that the foregoing description is only illustrative of this disclosure. Various alternatives and modifications can be devised by those skilled in the art without departing from the disclosure. Accordingly, this disclosure is intended to embrace all such alternatives, modifications, and variances. The aspects described with reference to the attached drawing figures are presented only to demonstrate certain examples of the disclosure. Other elements, steps, methods, and techniques that are insubstantially different

from those described above and/or in the appended claims are also intended to be within the scope of the disclosure.

**CLAIMS**

1. An e-key system comprising:

a mobile device configured to generate an encrypted code;

a key fob configured to receive the encrypted code from the mobile device and to transmit the encrypted code to a computing device incorporated into property via wireless communication; and

a server configured to update an access log of the key fob in a hyperledger,

wherein the computing device includes a validation module configured to validate the received encrypted code, and

wherein the computing device grants a user of the key fob access to the property when the validation module validates the encrypted code.

2. The e-key system according to claim 1, wherein the wireless communication is Bluetooth communication.

3. The e-key system according to claim 1, wherein the mobile device and the key fob are paired according to a Bluetooth protocol.

4. The e-key system according to claim 1, wherein the computing device of the property and the key fob are paired according to a Bluetooth protocol.

5. The e-key system according to claim 1, wherein the validation module is constantly powered by the property.

6. The e-key system according to claim 5, wherein the validation module is an electronic circuit.

7. The e-key system according to claim 1, wherein the code is encrypted with a public key by the mobile device.

8. The e-key system according to claim 7, wherein the validation module decrypts the encrypted code with a private key, which is saved in the validation module, corresponding to the public key.

9. The e-key system according to claim 1, wherein the validation module transmits the access log to the server when a network connection is established with the server.
10. The e-key system according to claim 1, wherein the property is an aircraft, watercraft, hovering vehicle, land vehicle, or building.
11. A method for granting access to property to a user of a key fob, the method comprising:
  - transmitting, by a mobile device, an encrypted code to a key fob;
  - transmitting, by the key fob, the encrypted code to a validation module running on a computing device of property via wireless communication;
  - determining, by the validation module, whether the encrypted code is valid;
  - granting access to the property when it is determined that the encrypted code is valid;and
  - denying access to the property when it is determined that the encrypted code is invalid.
12. The method according to claim 11, wherein the wireless communication is Bluetooth.
13. The method according to claim 11, wherein the property and the key fob are paired according to a Bluetooth protocol.
14. The method according to claim 11, wherein the mobile device and the key fob are paired according to a Bluetooth protocol.
15. The method according to claim 11, wherein the encrypted code is encrypted by a public key.
16. The method according to claim 15, further comprising decrypting, by the validation module, the encrypted code with a private key, which is saved in the validation module, corresponding to the public key.
17. The method according to claim 11, further comprising transmitting, by the validation module, an access log to a server when a network connection is established with the server.

18. The method according to claim 11, wherein the property is an aircraft, watercraft, hovering vehicle, land vehicle, or building.
19. A satellite data communication system comprising:  
a server configured to validate an access key;  
a first station configured to transmit an access key via the server and a satellite; and  
a second station configured to receive the access key and to transmit data to the first station via the server and the satellite when the server validates the access key,  
wherein the server saves a transmission log in a hyperledger using blockchain, and  
wherein the server transmits the access key in response to a request by the first station.
20. The satellite data communication system according to claim 19, wherein the transmission log includes a validation history of the access key.
21. The satellite data communication system according to claim 19, wherein the satellite is not permitted to transmit the data from the second station to the first station when the access key is not validated.
22. The satellite data communication system according to claim 19, wherein the satellite is permitted to transmit the data from the second station to the first station when the access key is validated.
23. The satellite data communication system according to claim 19, wherein the second station transmits the access key to the server when the second station receives the access key from the first station.
24. The satellite data communication system according to claim 19, wherein the access key is encrypted by the server.
25. The satellite data communication system according to claim 19, wherein the server is further configured to encrypt data before transmitting data to the satellite.

26. The satellite data communication system according to claim 25, wherein the server is further configured to save the encrypted data in a hyperledger.
27. The satellite data communication system according to claim 19, wherein the second station is further configured to encrypt the data using a public key.
28. The satellite data communication system according to claim 27, wherein the first station is further configured to decrypt the data using a private key corresponding to the public key.
29. The satellite data communication system according to claim 19, wherein the first and second stations are military stations.
30. The satellite data communication system according to claim 19, wherein the access key is valid for data transmission for a period set by the server or the second station.
31. The satellite data communication system according to claim 30, wherein the server blocks the data transmission between the first and second stations after the set period elapses.
32. A method for securing satellite data communications between first and second stations, the method comprising:
- transmitting, by the first station, a request for an access key to a server via a satellite;
  - receiving, by the first station, an access key from the server;
  - transmitting, by the first station, the access key to the second station;
  - transmitting, by the second station, the access key to the server;
  - validating, by the server, the access key transmitted by the second station;
  - permitting satellite data communications between the first station and the second station when the server determines that the access key is valid; and
  - storing, by the server, a transmission log in a hyperledger.
33. The method according to claim 32, wherein the transmission log includes a validation history of the access key.

34. The method according to claim 32, wherein transmission of the data is blocked when the server determines that the access key is not valid.

35. The method according to claim 32, wherein the access key is encrypted by the server.

36. The method according to claim 32, wherein the satellite data communications are encrypted before being transmitted to the first station.

37. The method according to claim 36, wherein the encrypted satellite data communications are saved in the hyperledger.

38. The method according to claim 32, further comprising:

encrypting, by the second station, the satellite data communications with a public key before transmitting the satellite data communications to the first station.

39. The method according to claim 38, further comprising:

decrypting, by the first station, the encrypted satellite data communications with a private key corresponding to the public key.

40. The method according to claim 32, wherein the access key is valid for satellite data communications for a period set by the server or the second station.

41. The method according to claim 40, wherein the satellite data communications are blocked between the first and second stations after the set period.

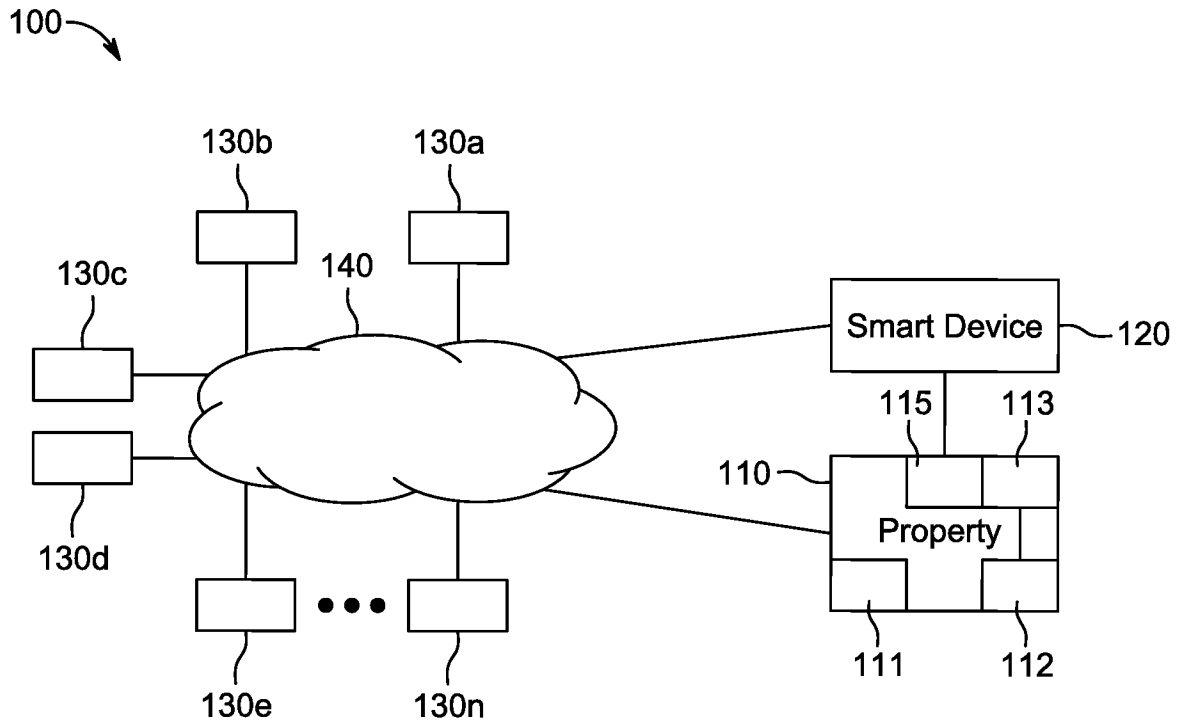


FIG. 1

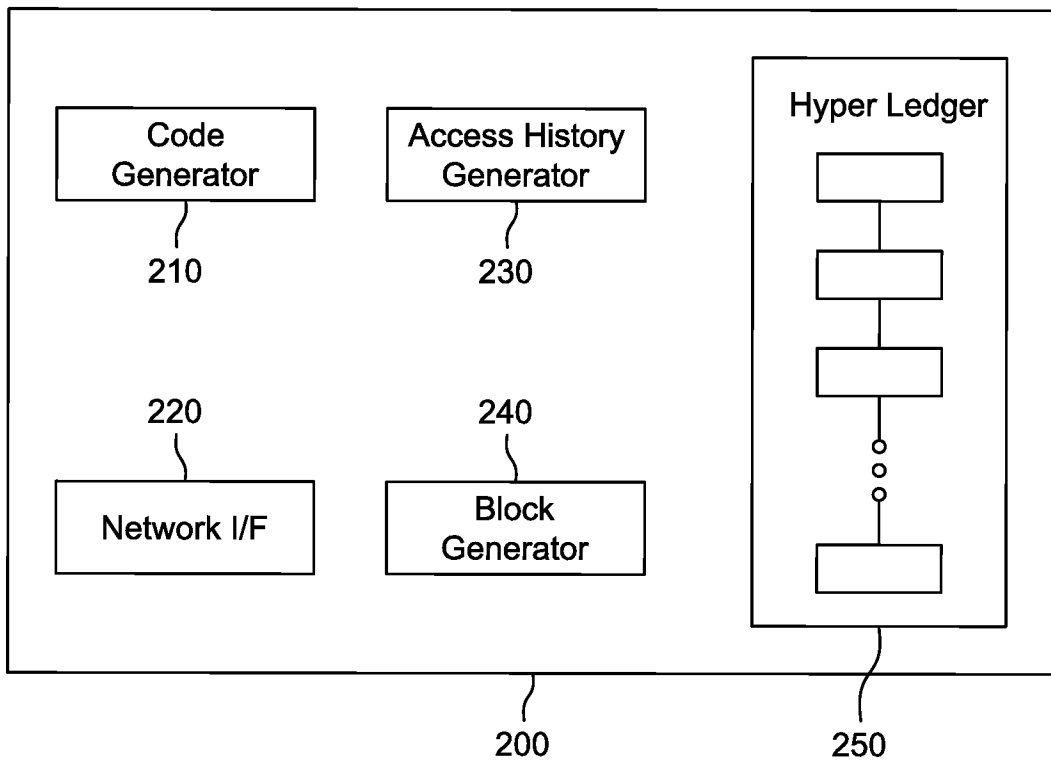


FIG. 2

2/36

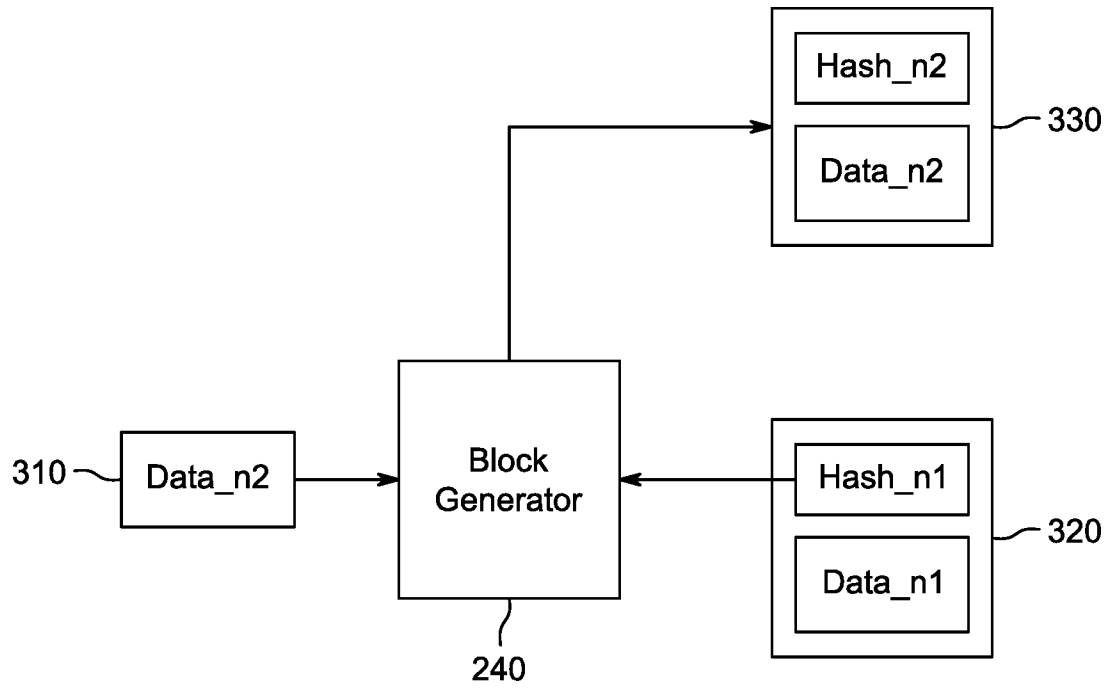


FIG. 3

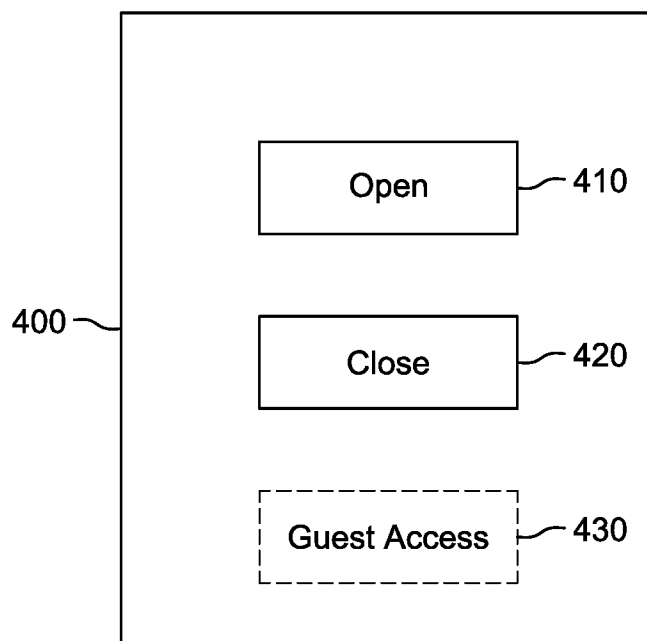


FIG. 4

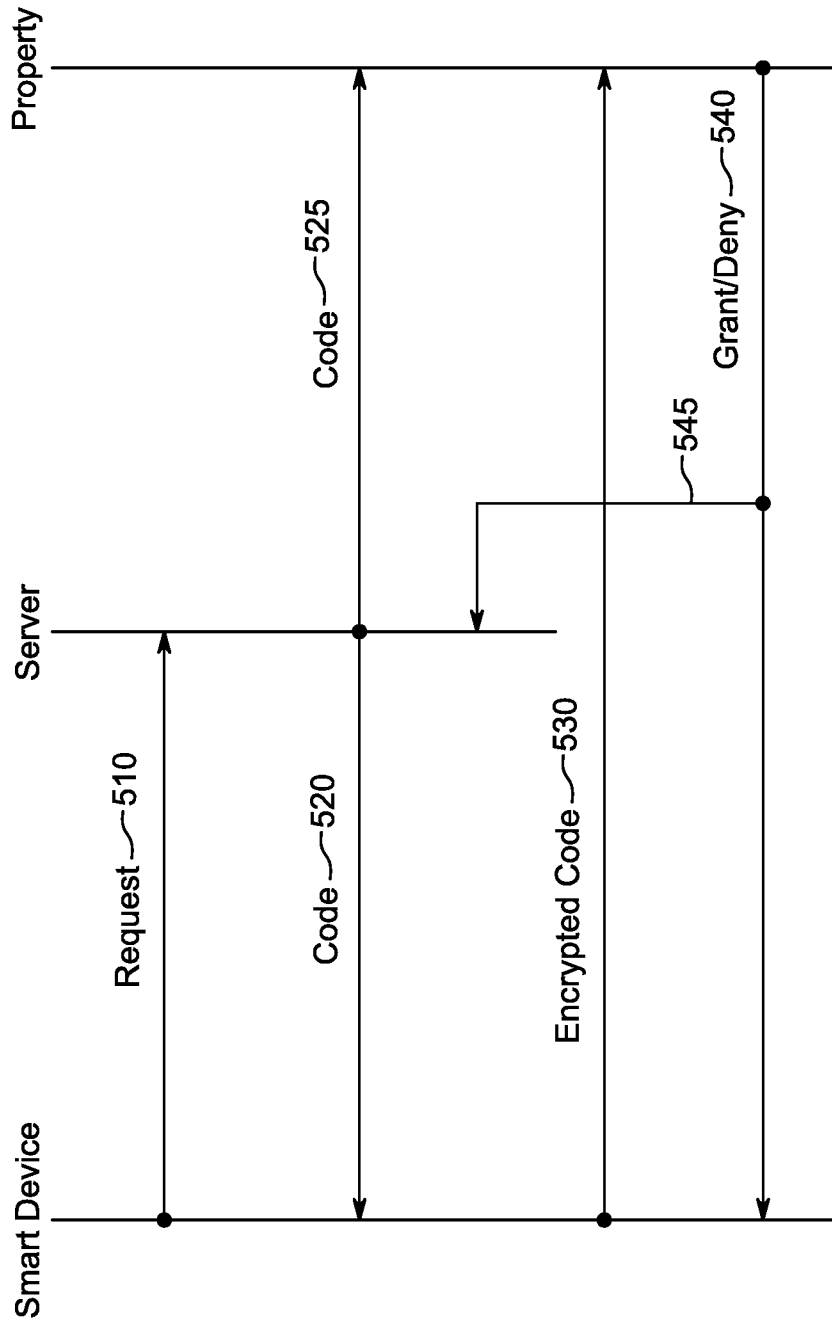


FIG. 5

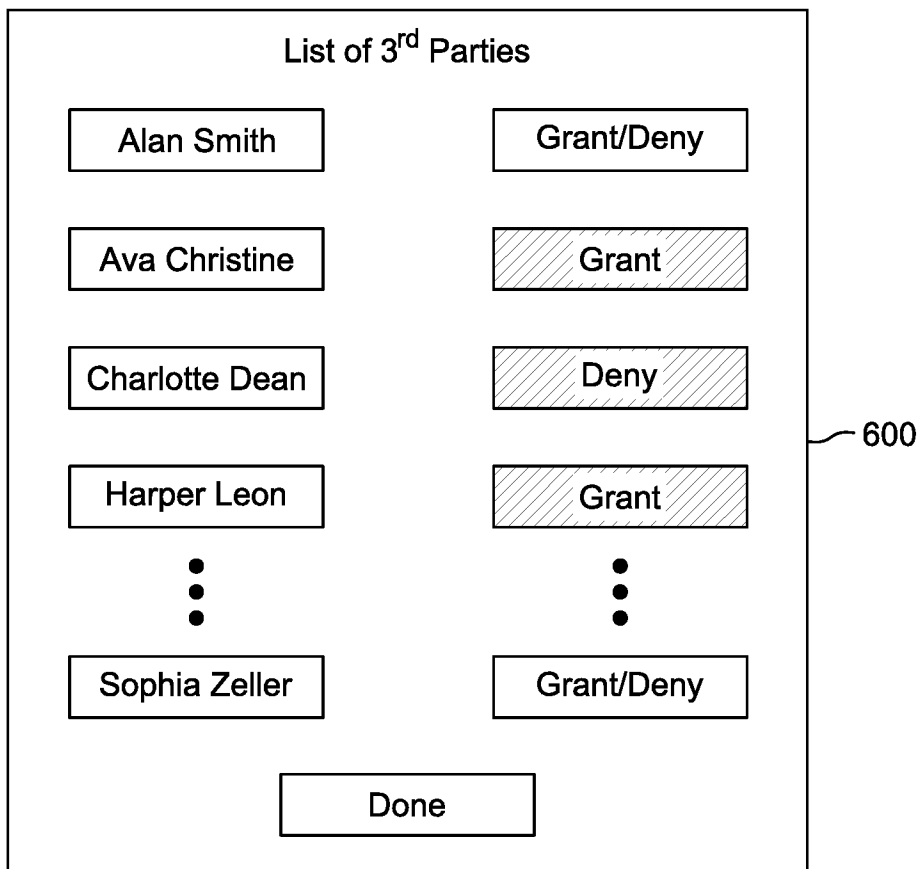


FIG. 6

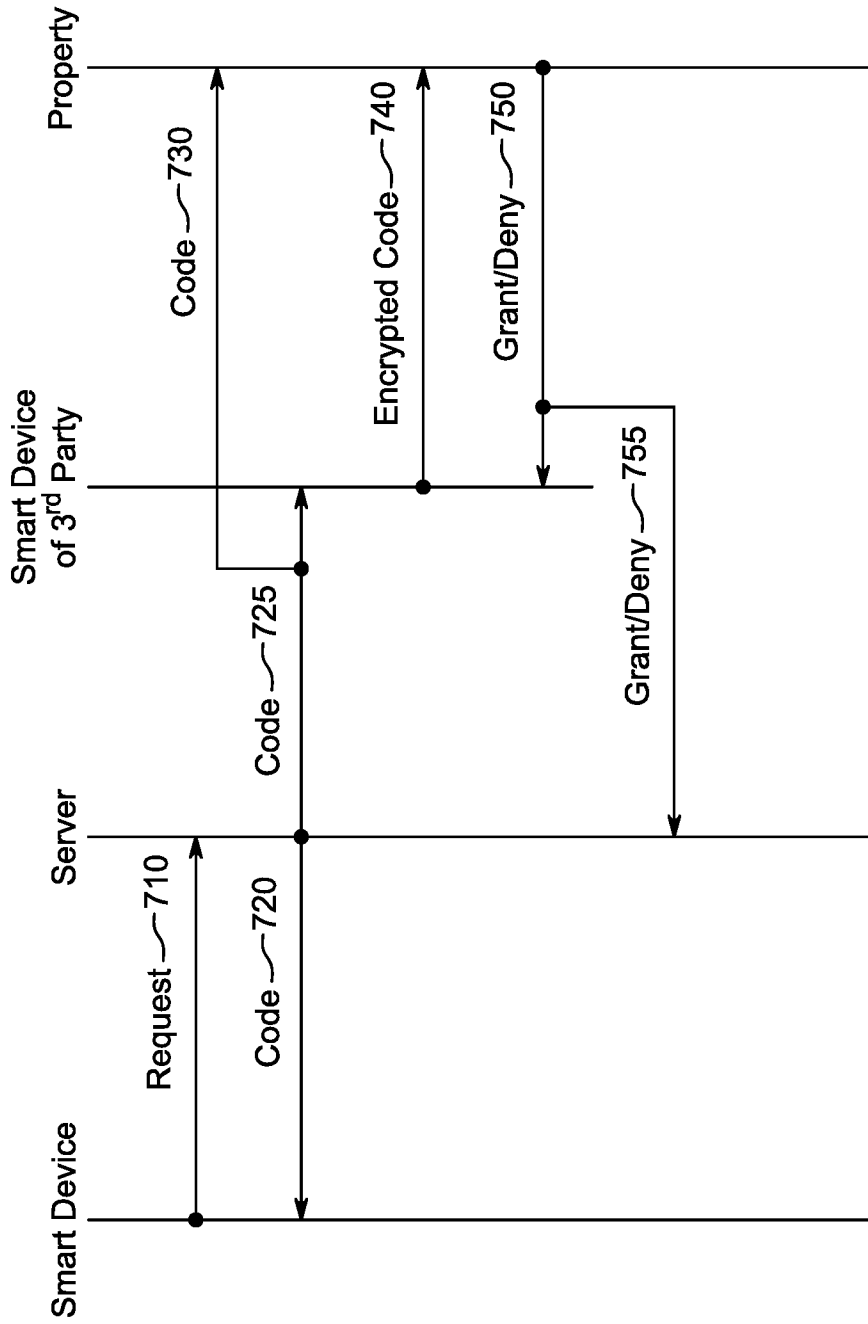


FIG. 7



7/36

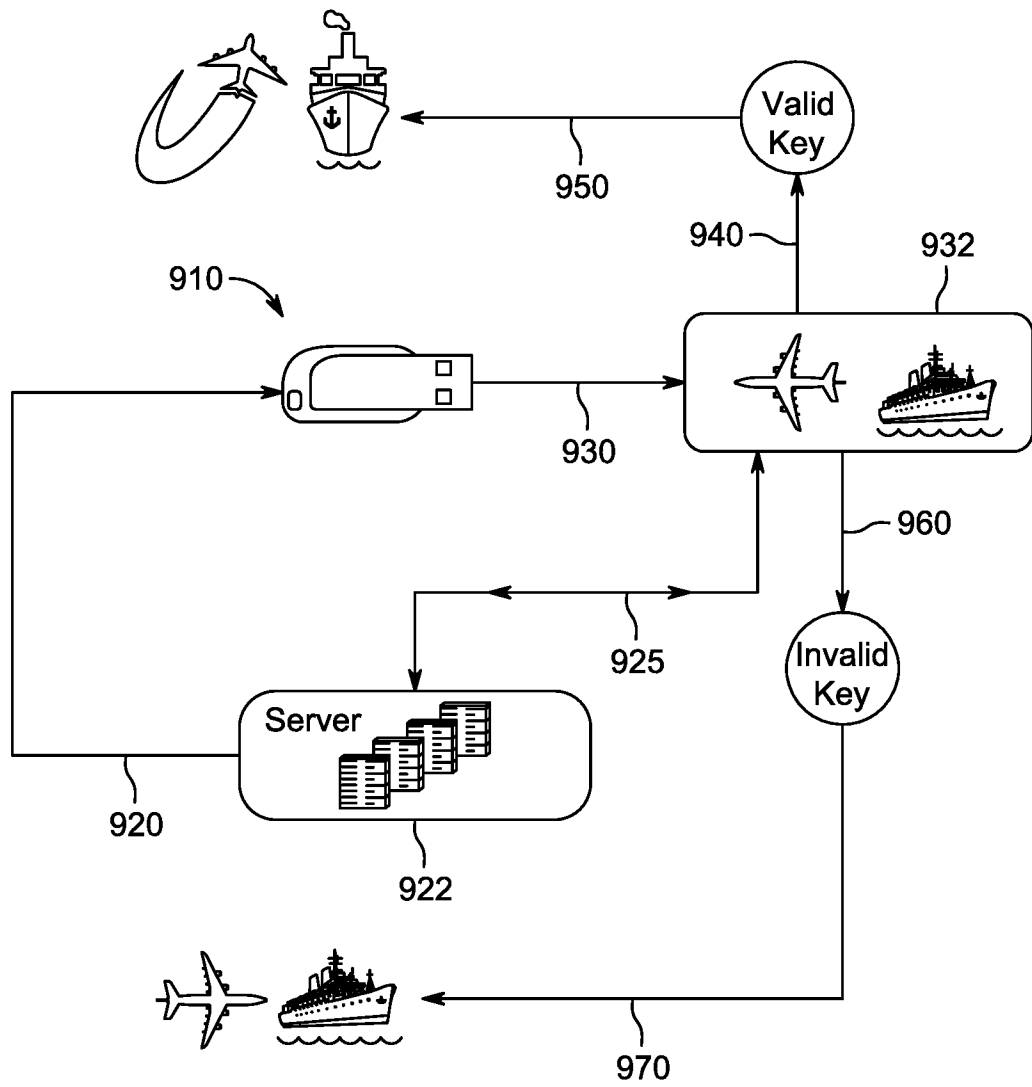


FIG. 9

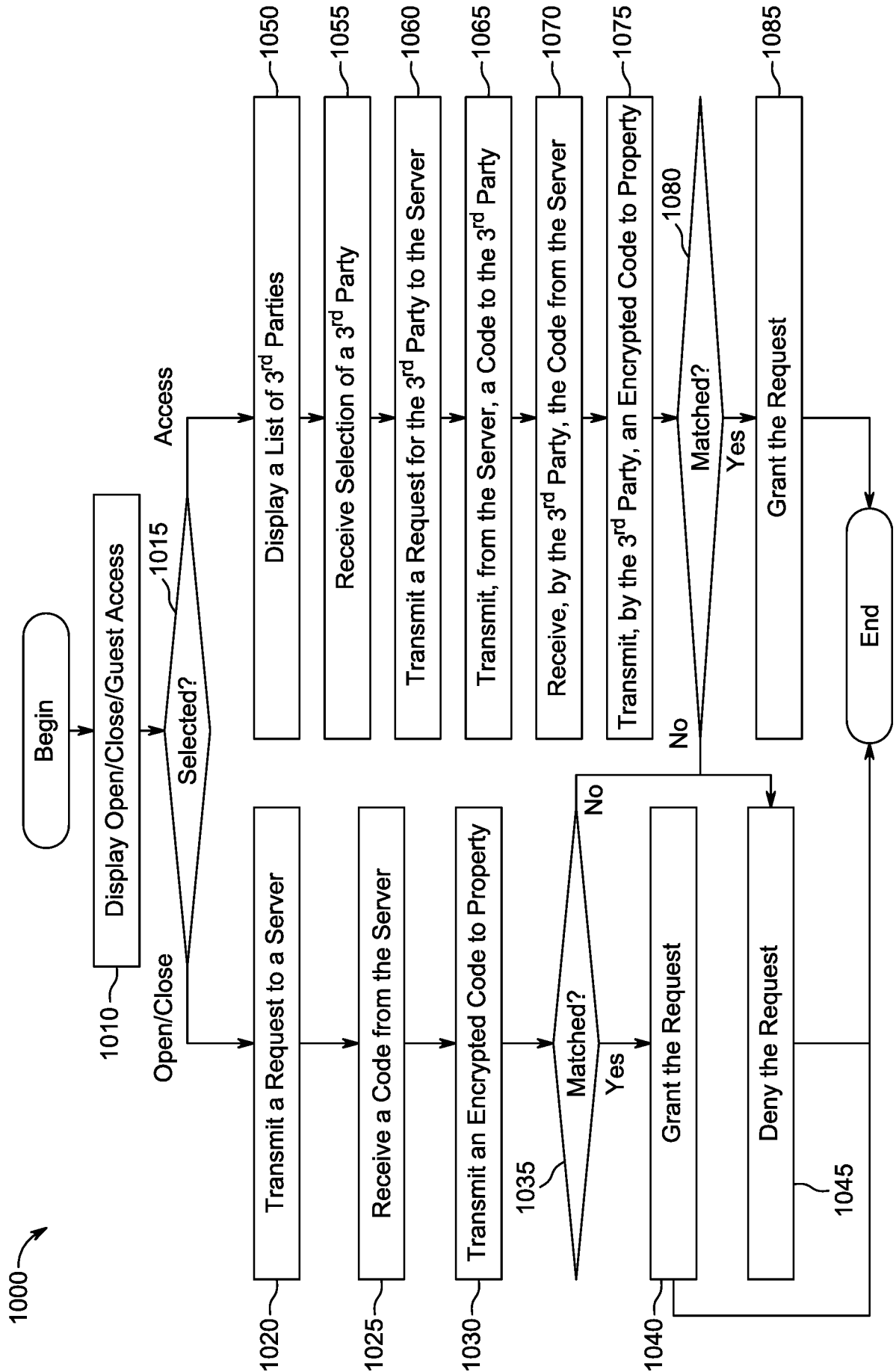


FIG. 10

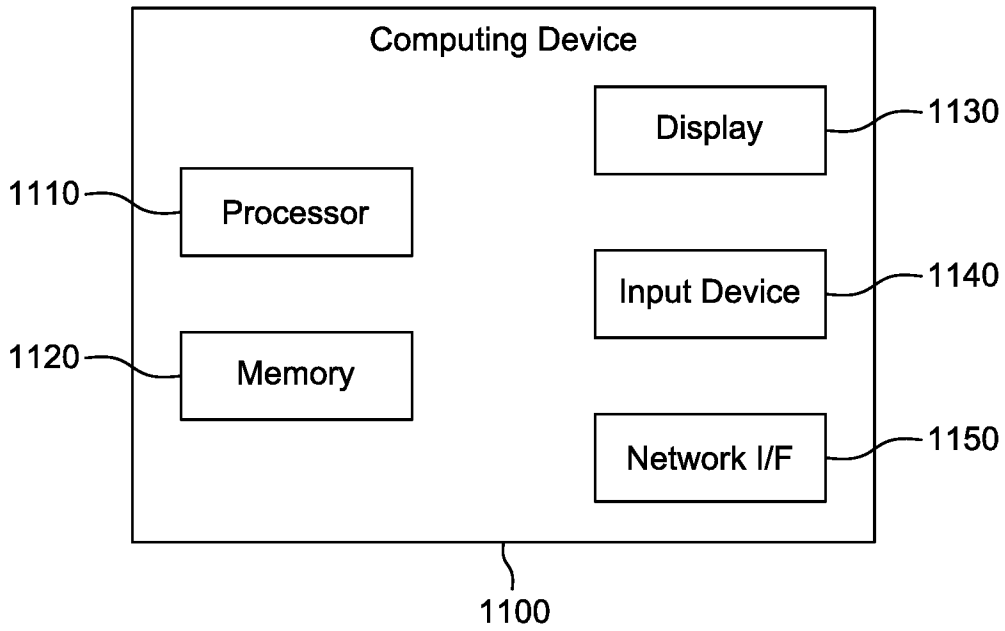


FIG. 11

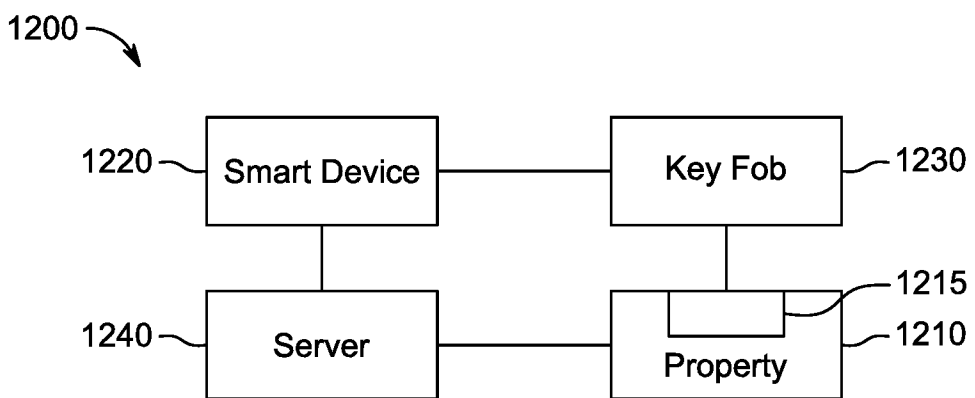


FIG. 12A

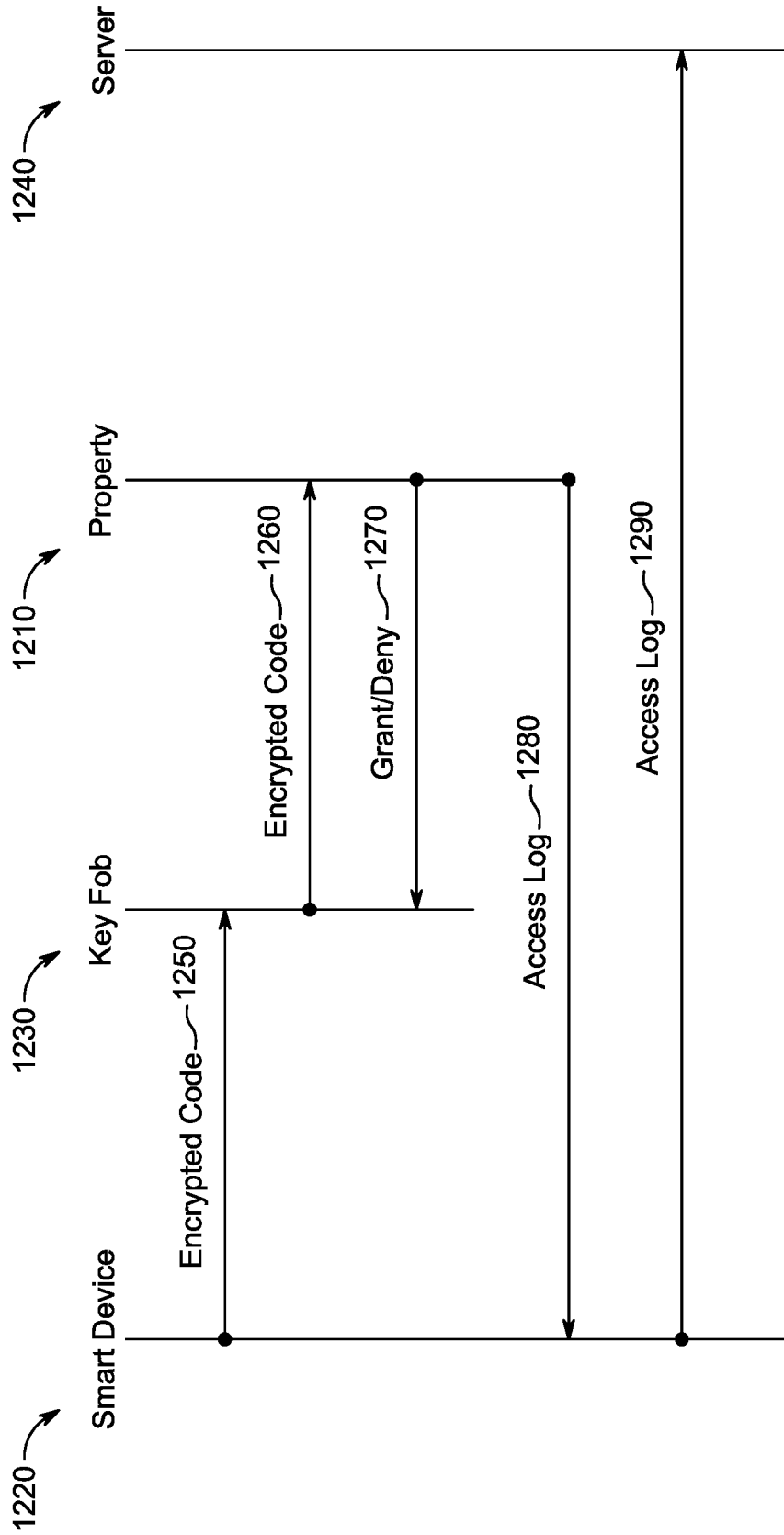


FIG. 12B

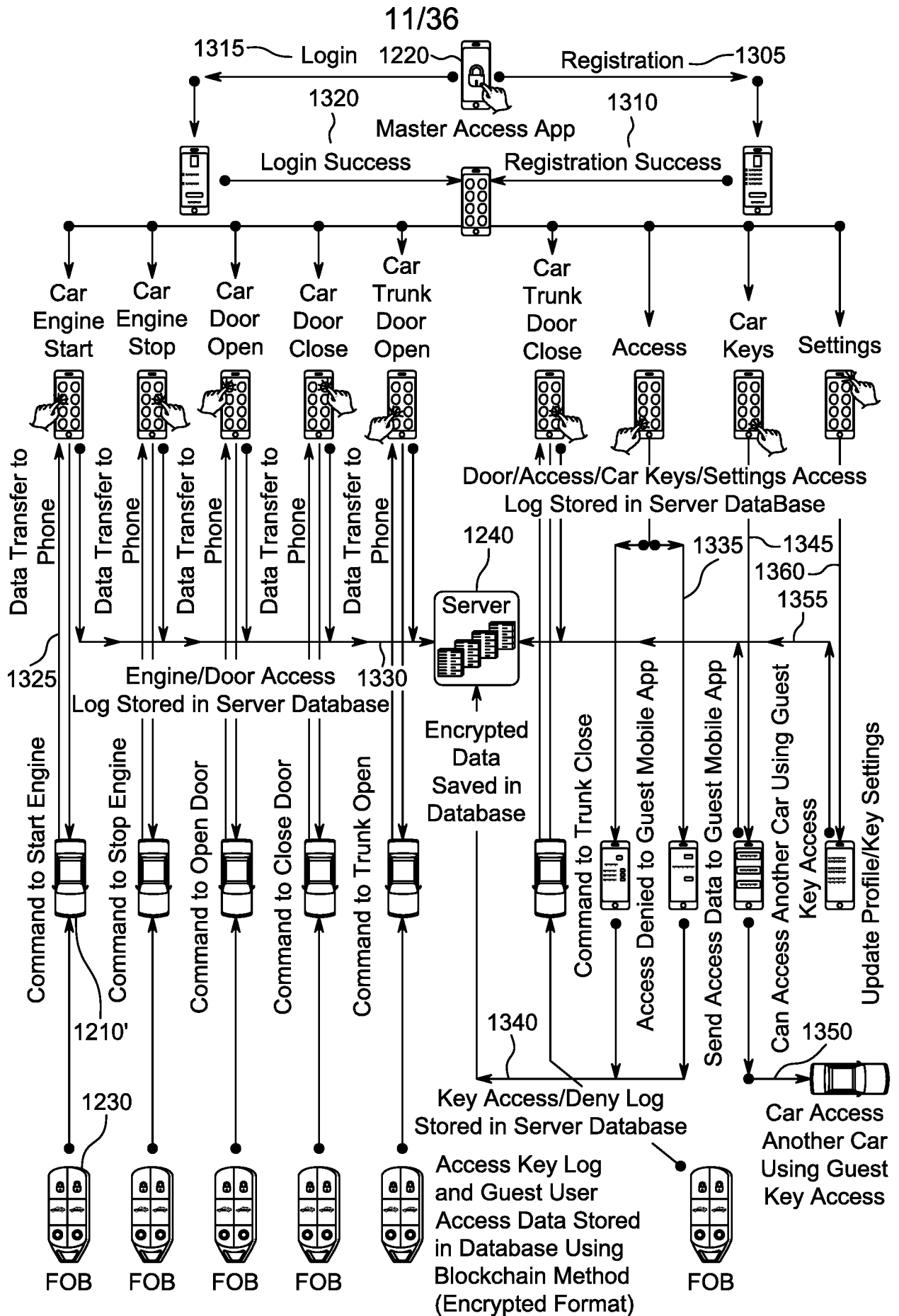


FIG. 13

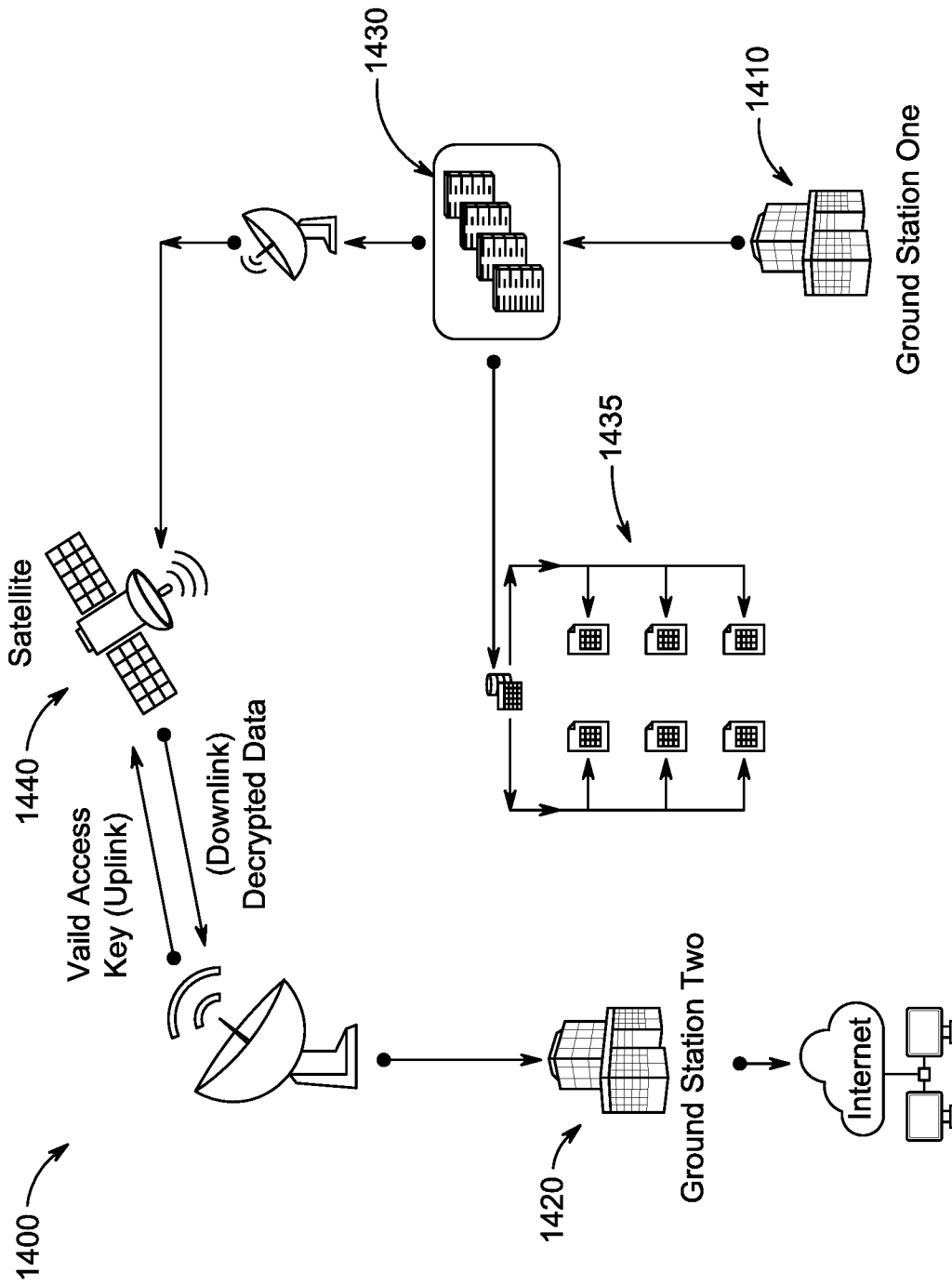


FIG. 14

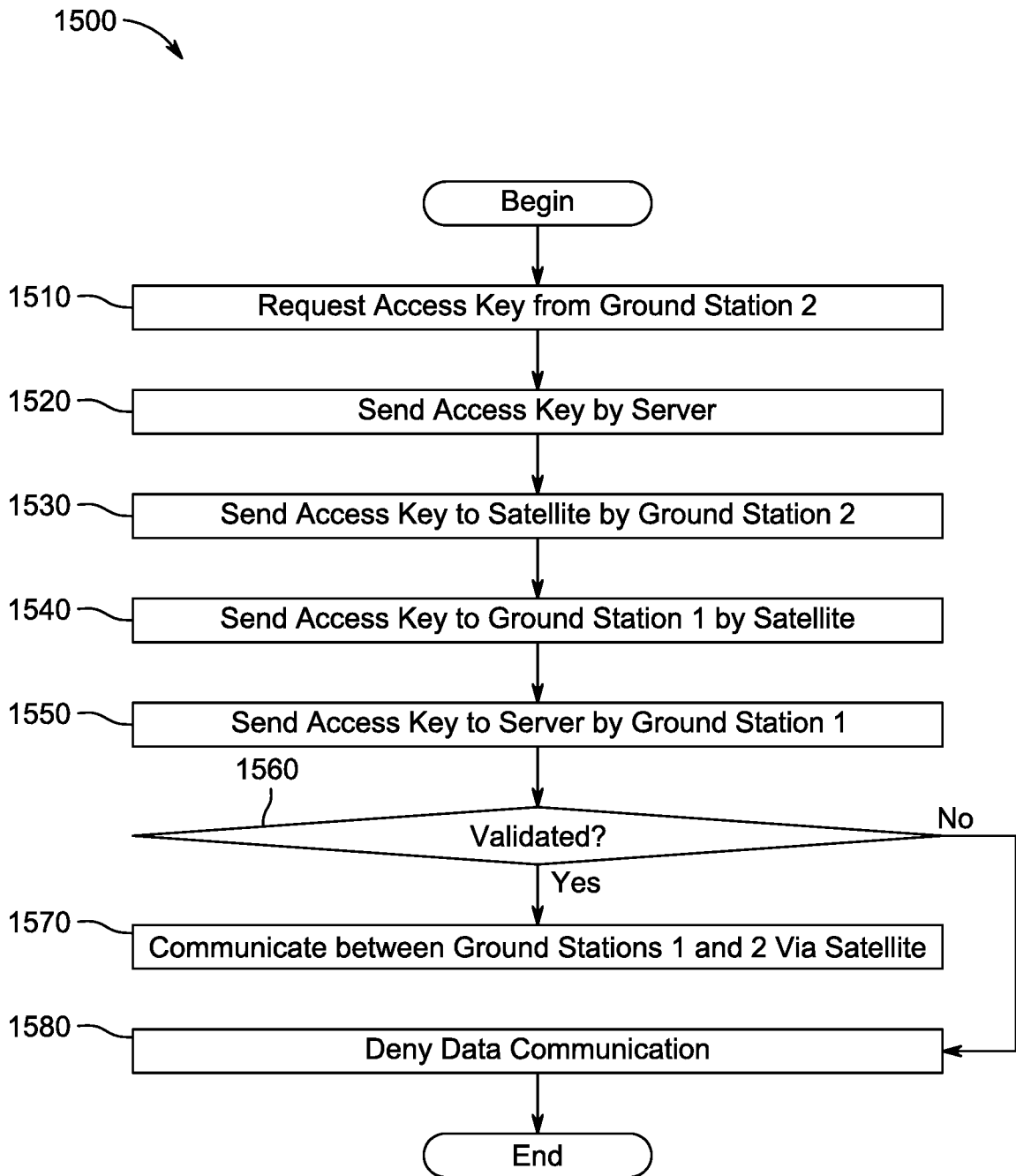
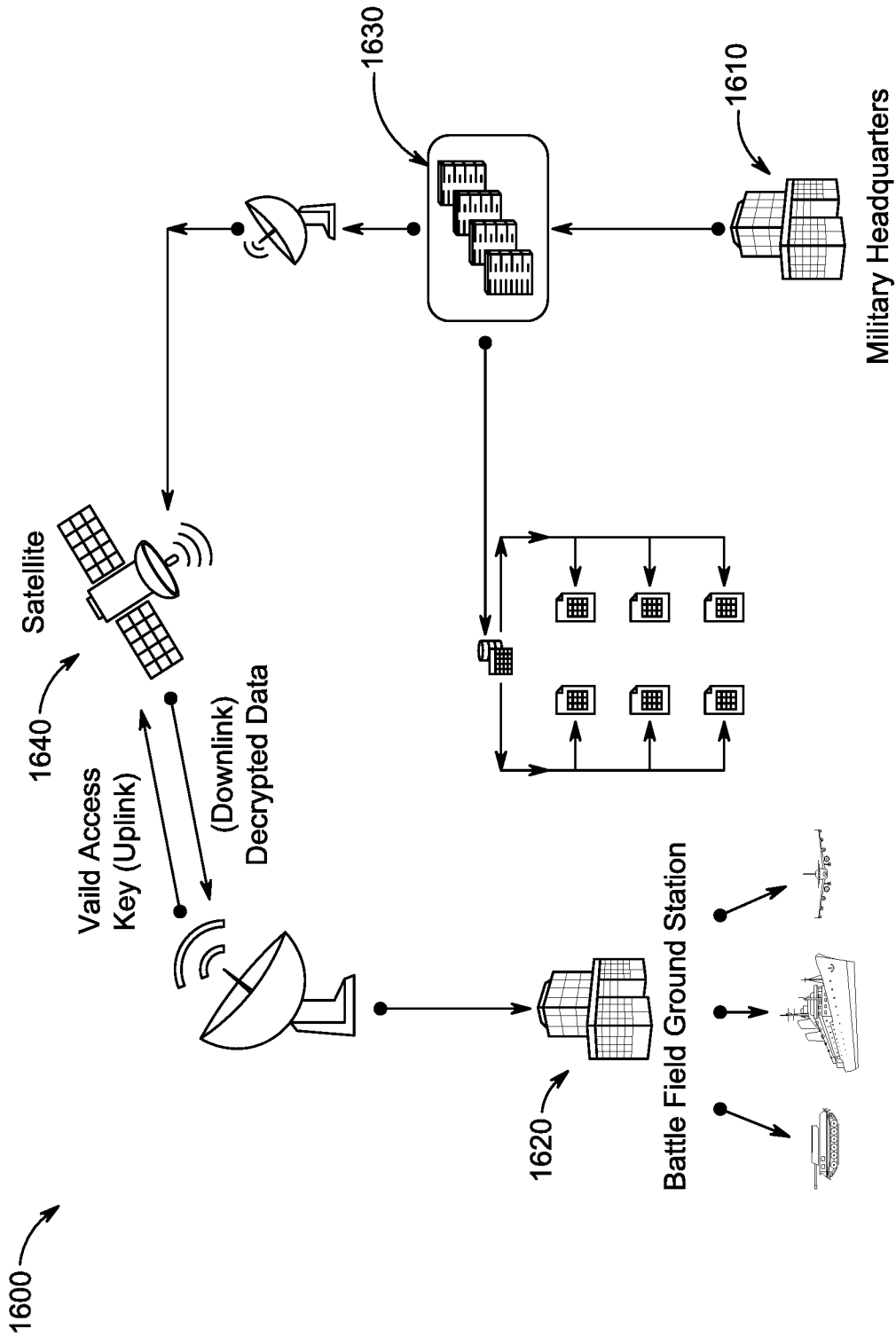


FIG. 15



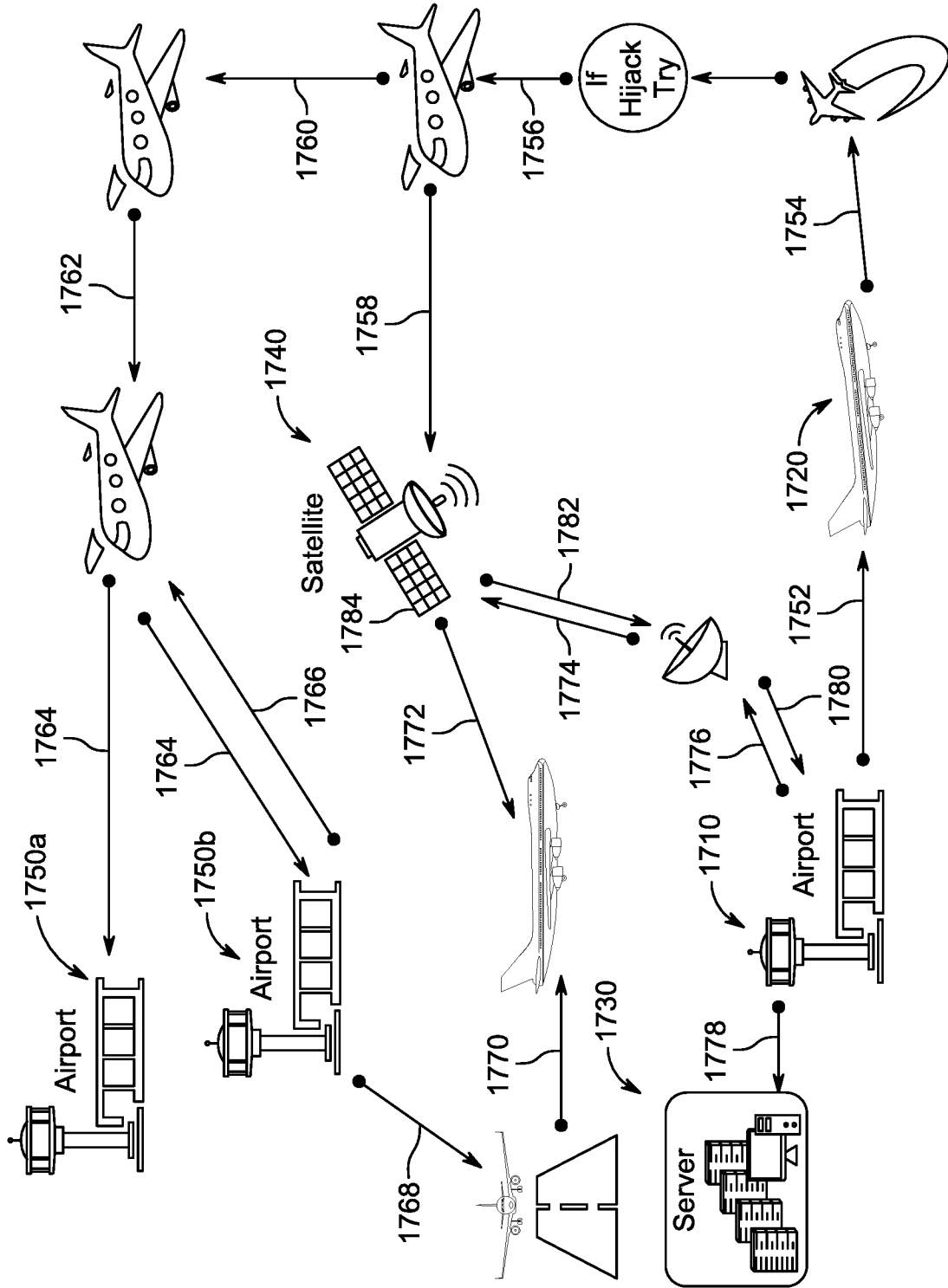


FIG. 17

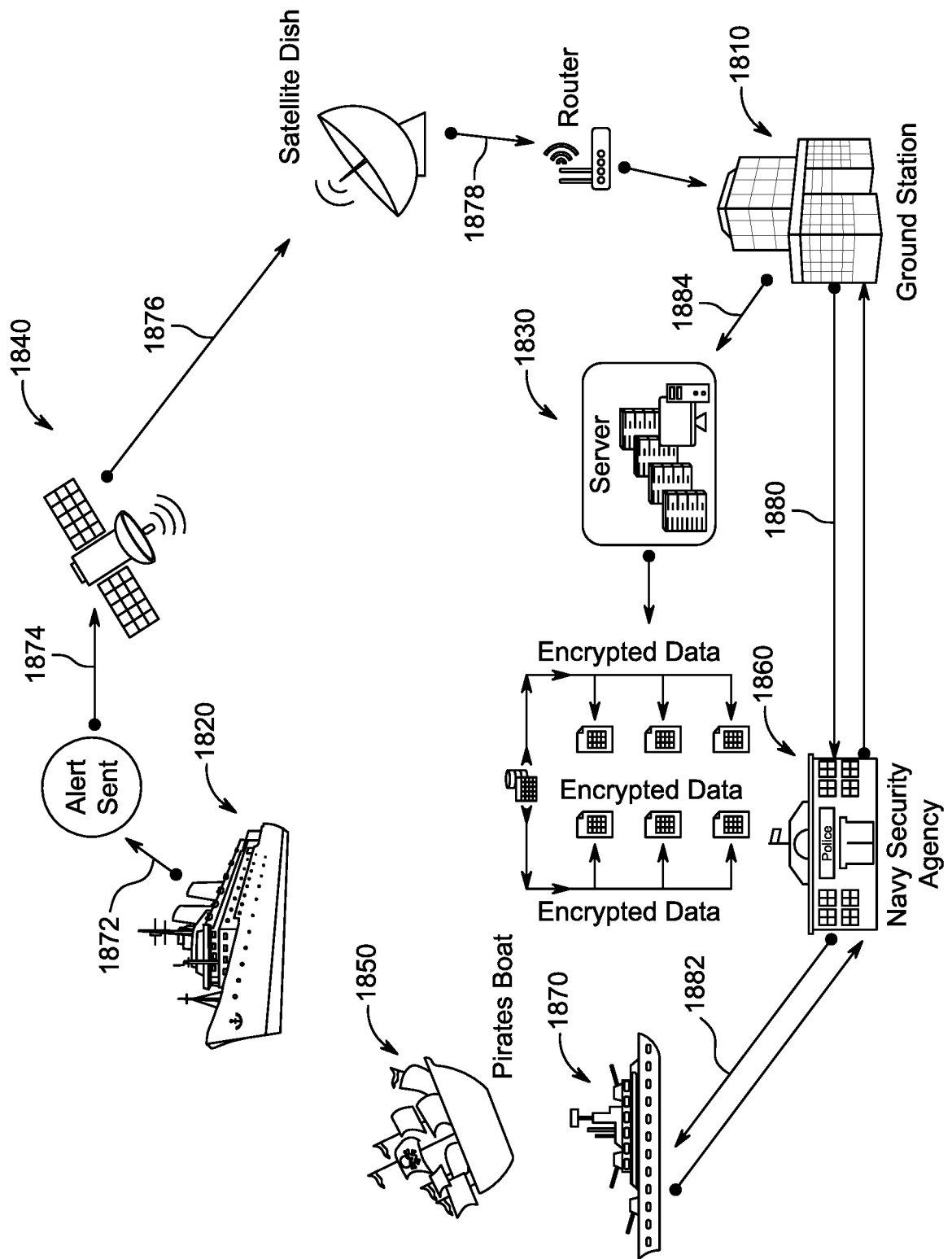


FIG. 18

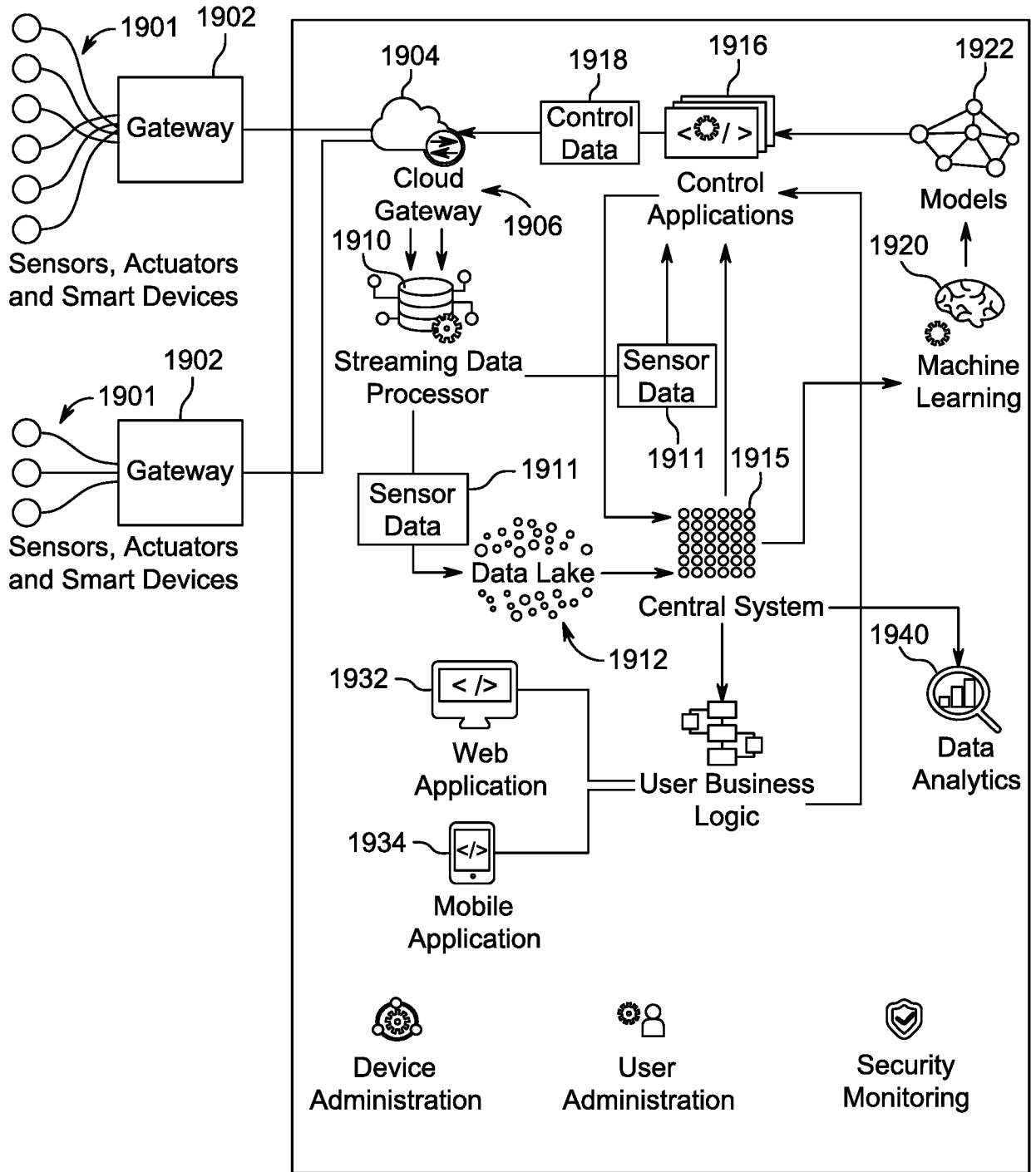


FIG. 19

18/36

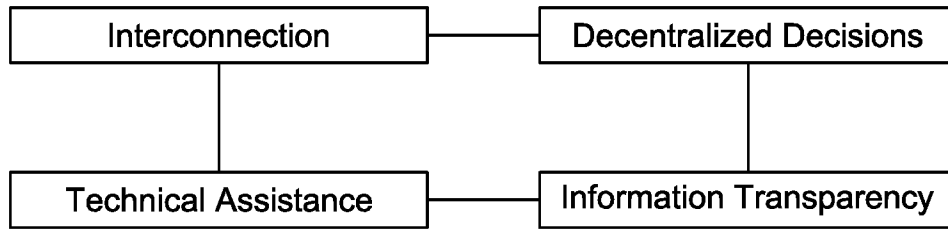


FIG. 20

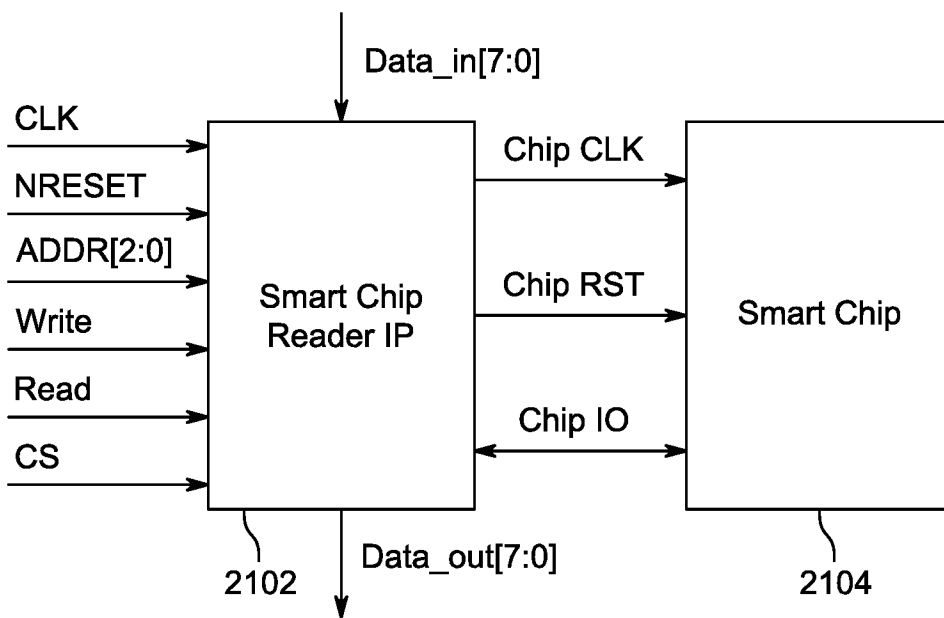


FIG. 21

19/36

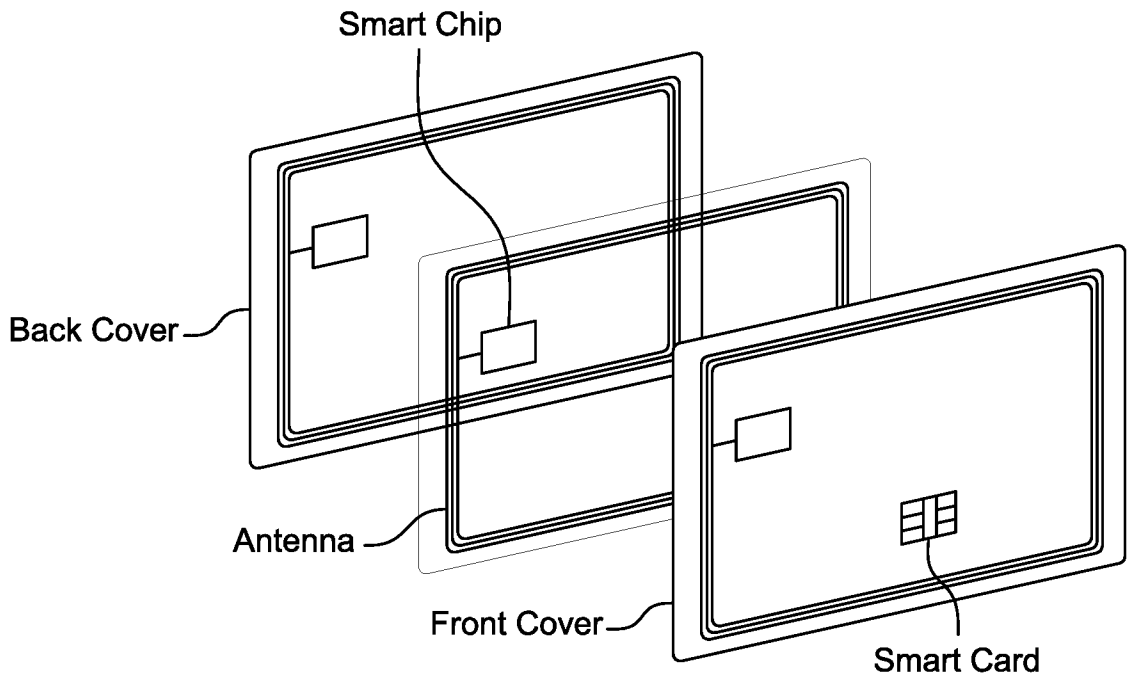


FIG. 22

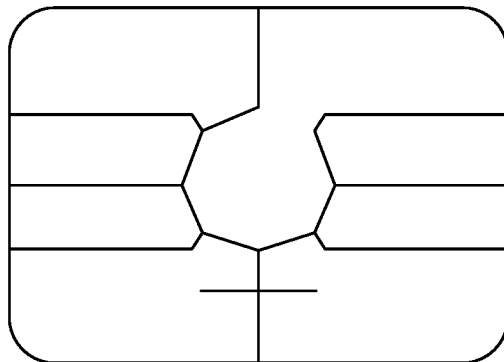


FIG. 23

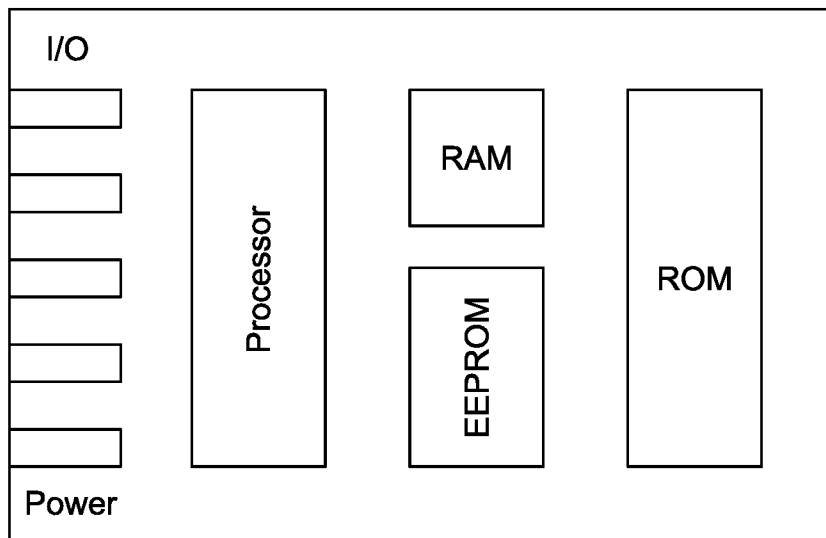


FIG. 24

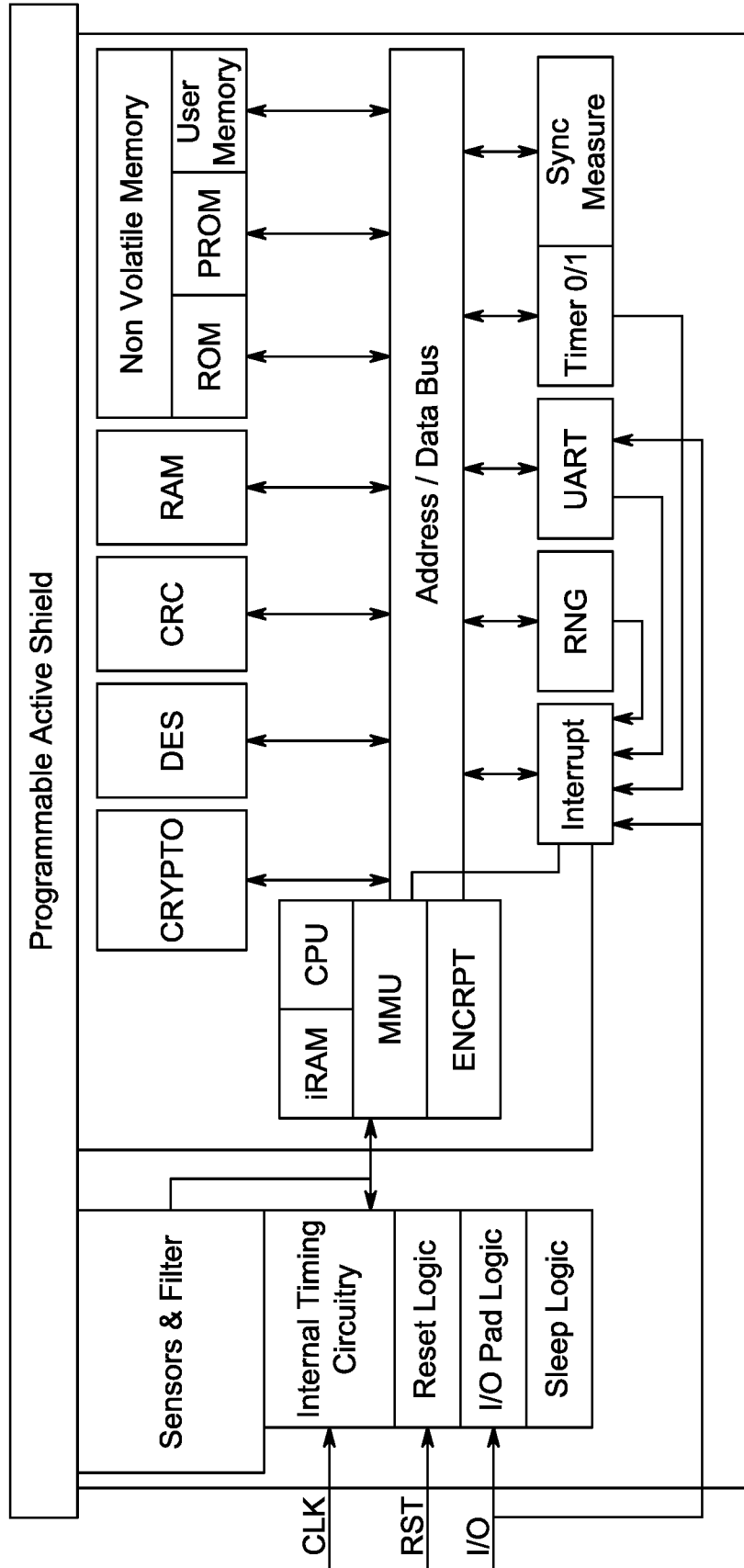


FIG. 25

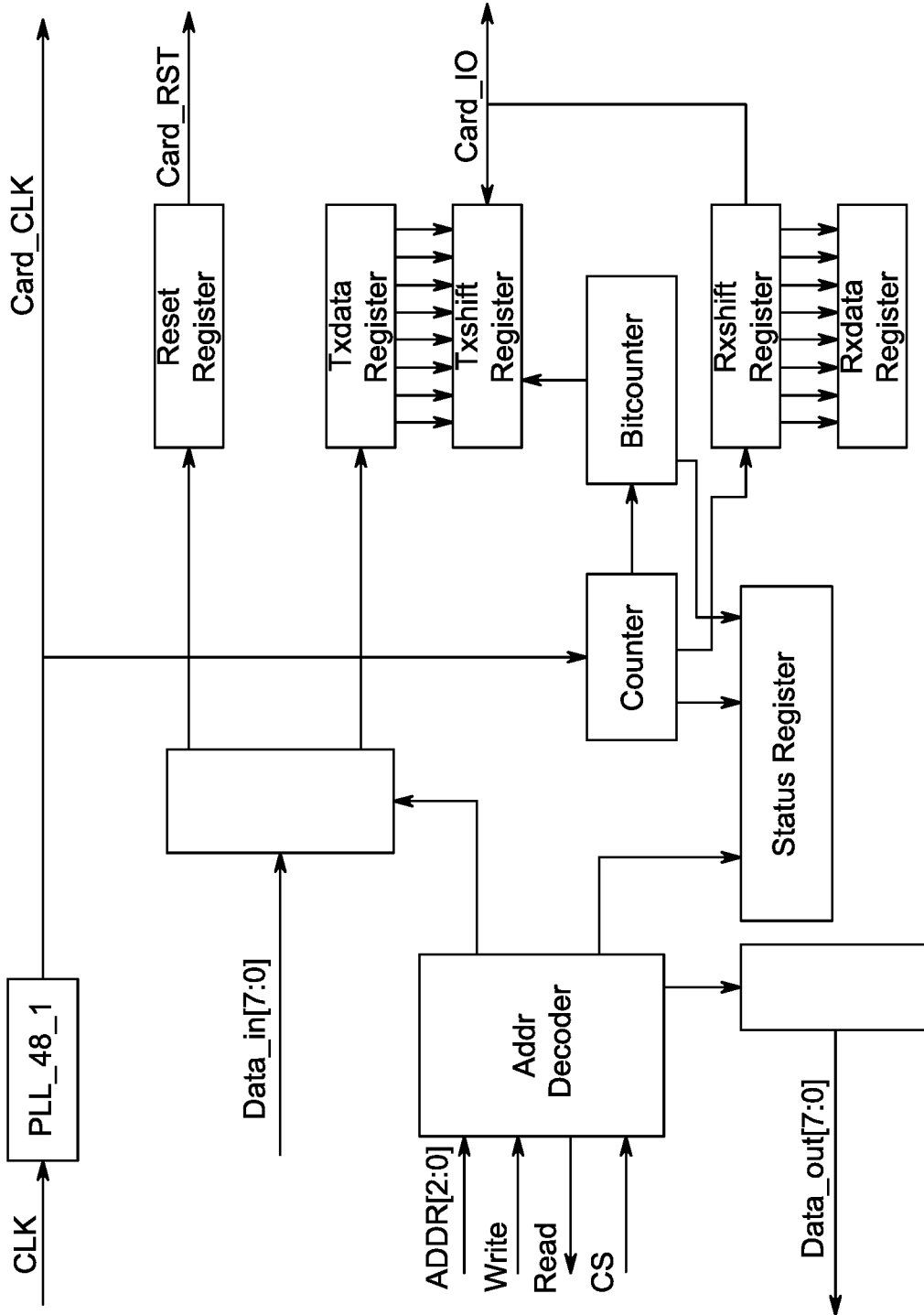


FIG. 26

23/36

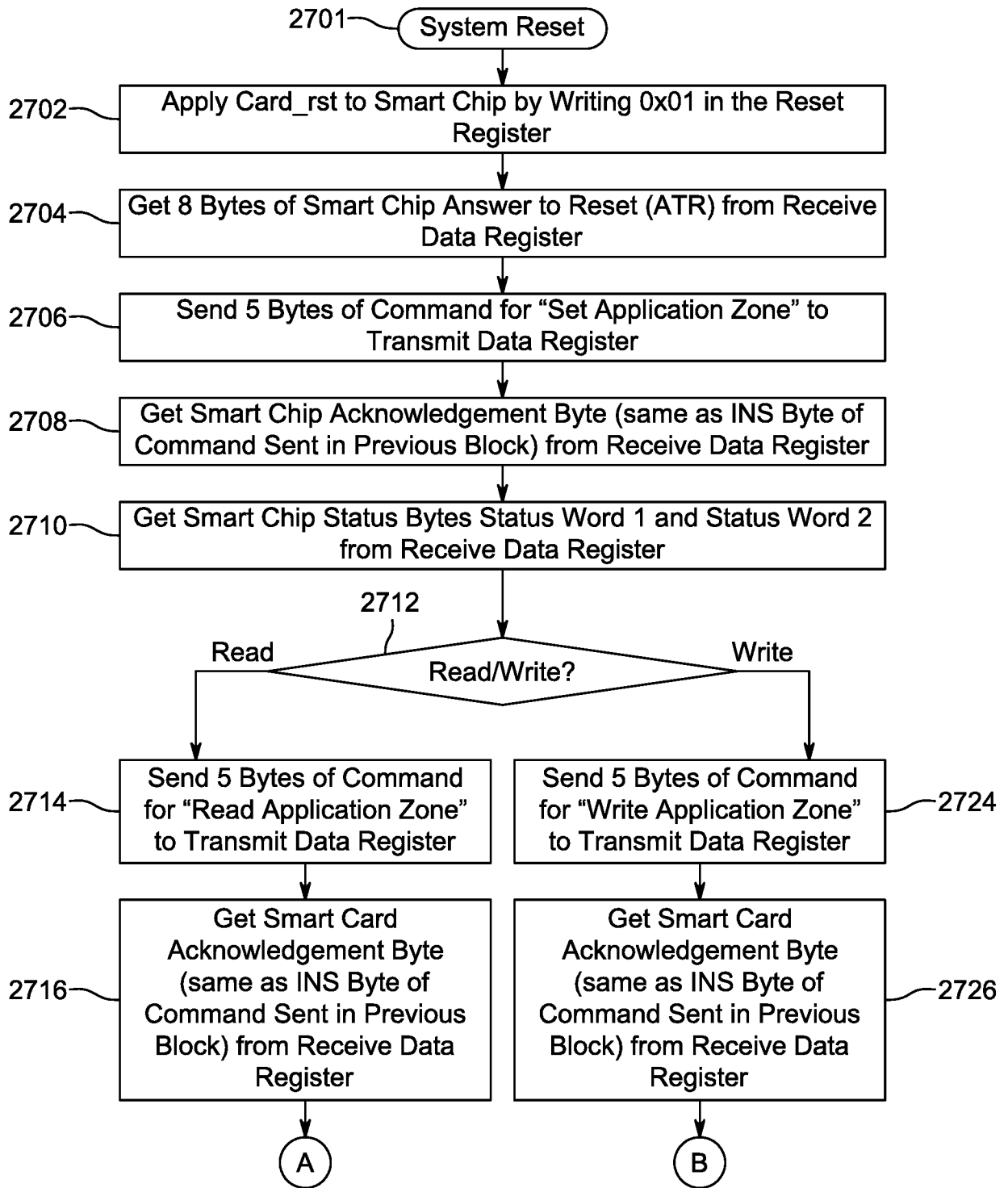


FIG. 27A

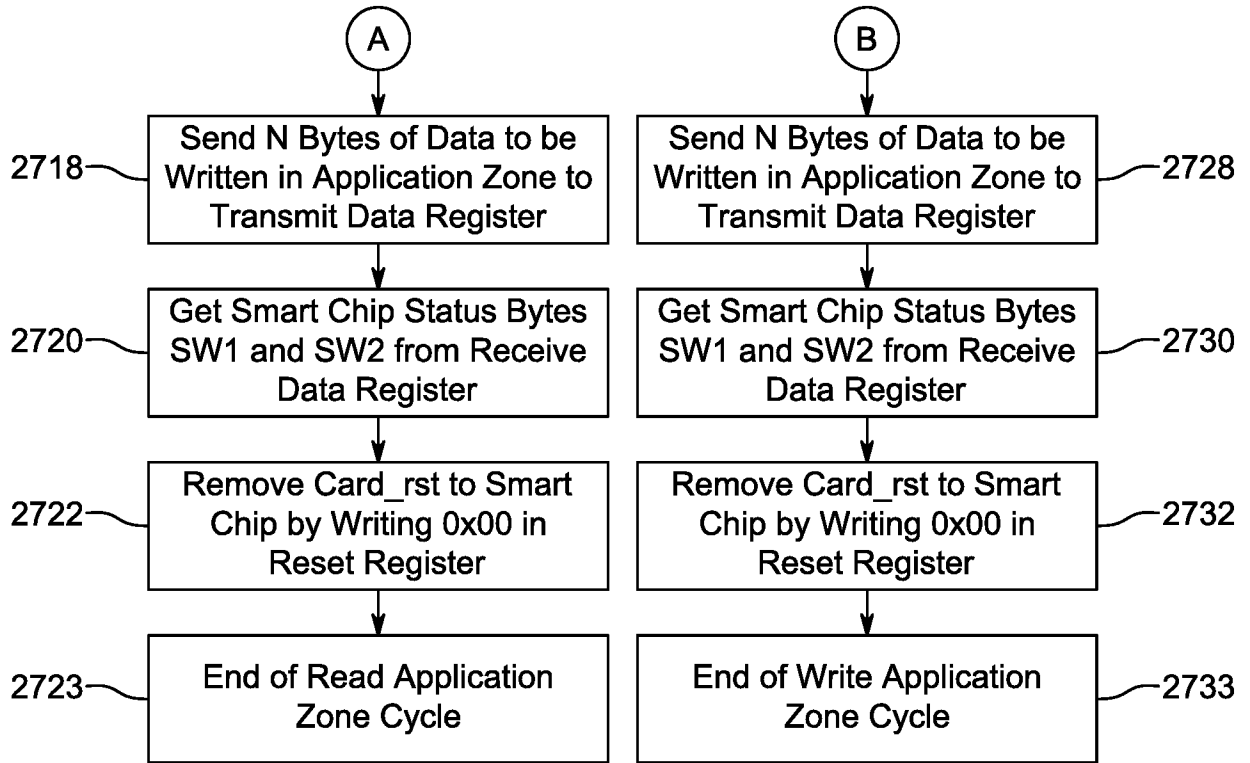


FIG. 27B

25/36

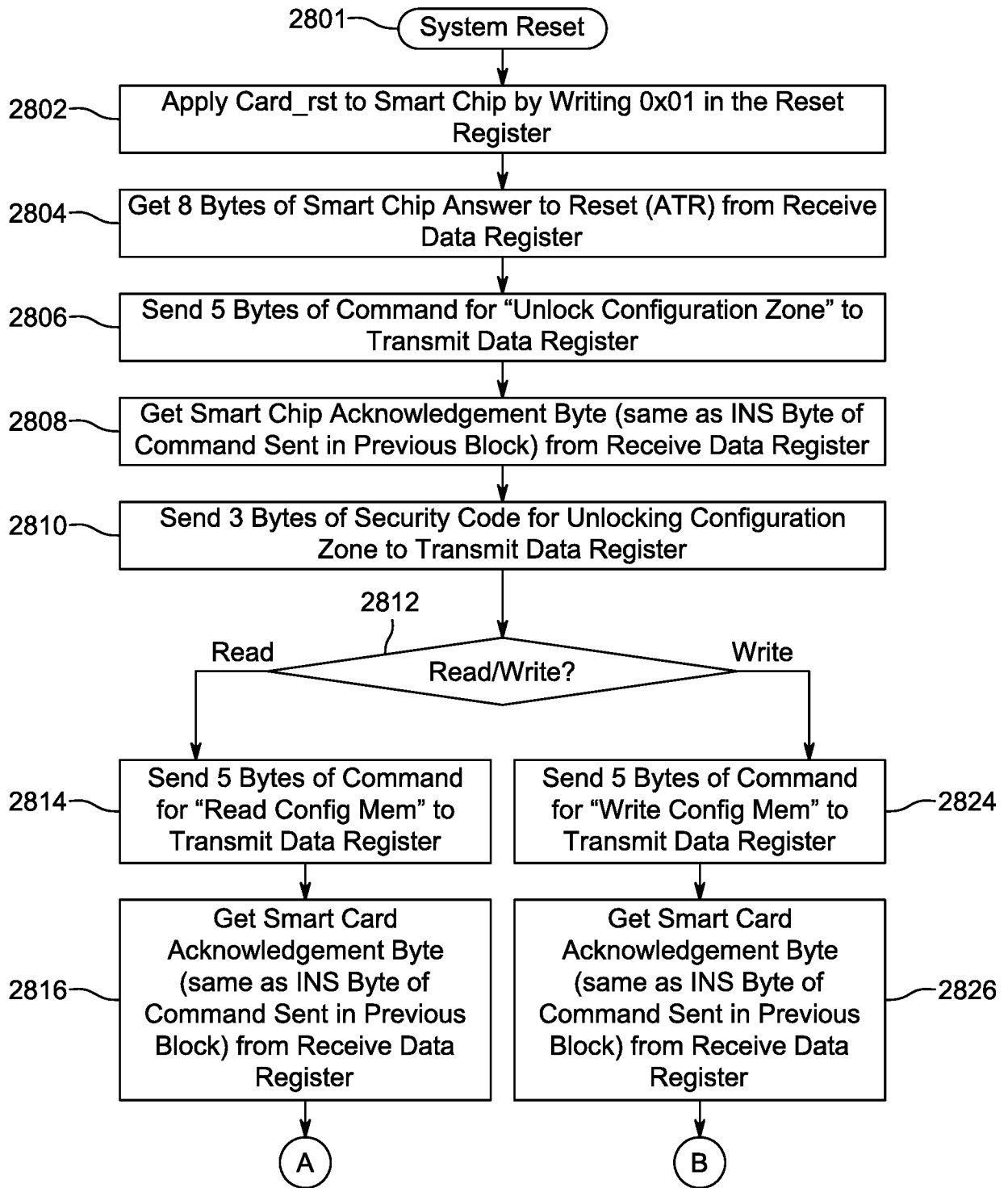


FIG. 28A

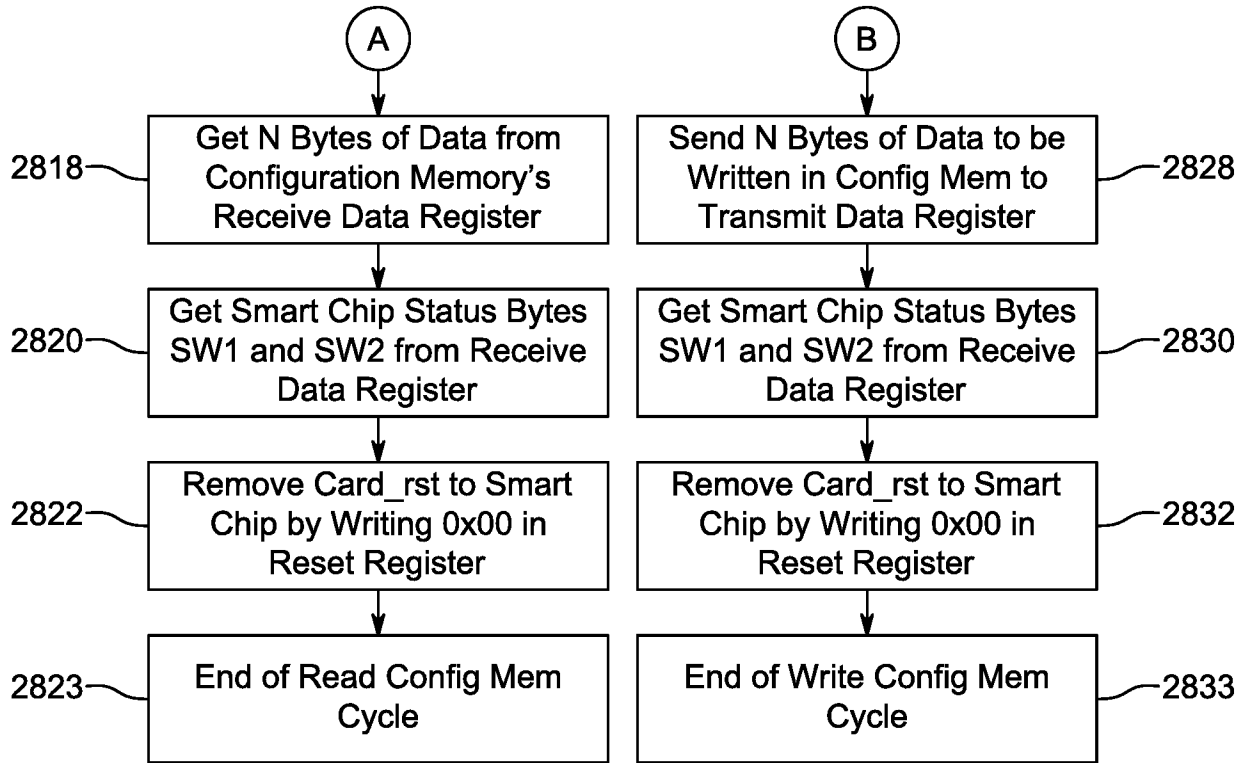


FIG. 28B

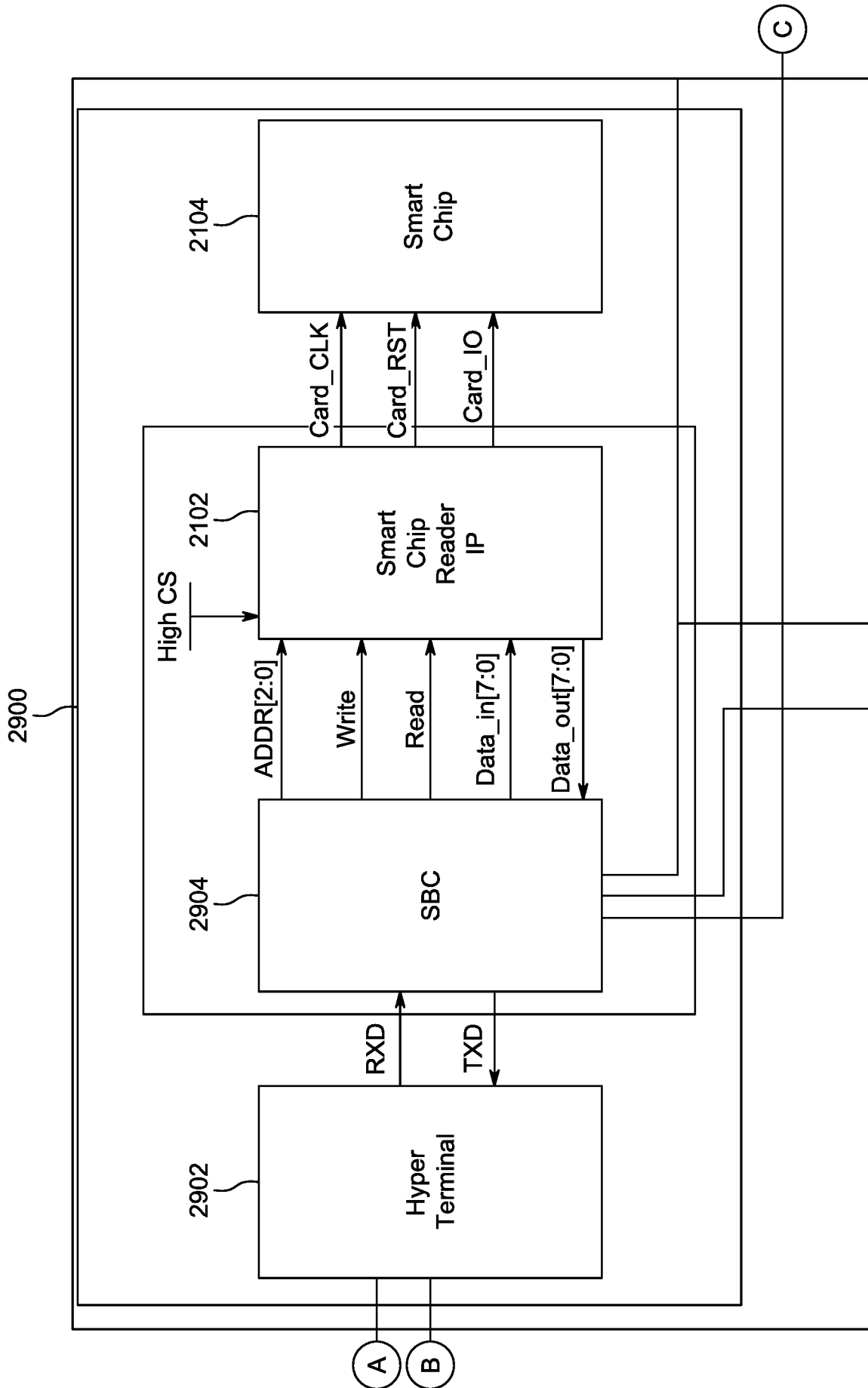


FIG. 29

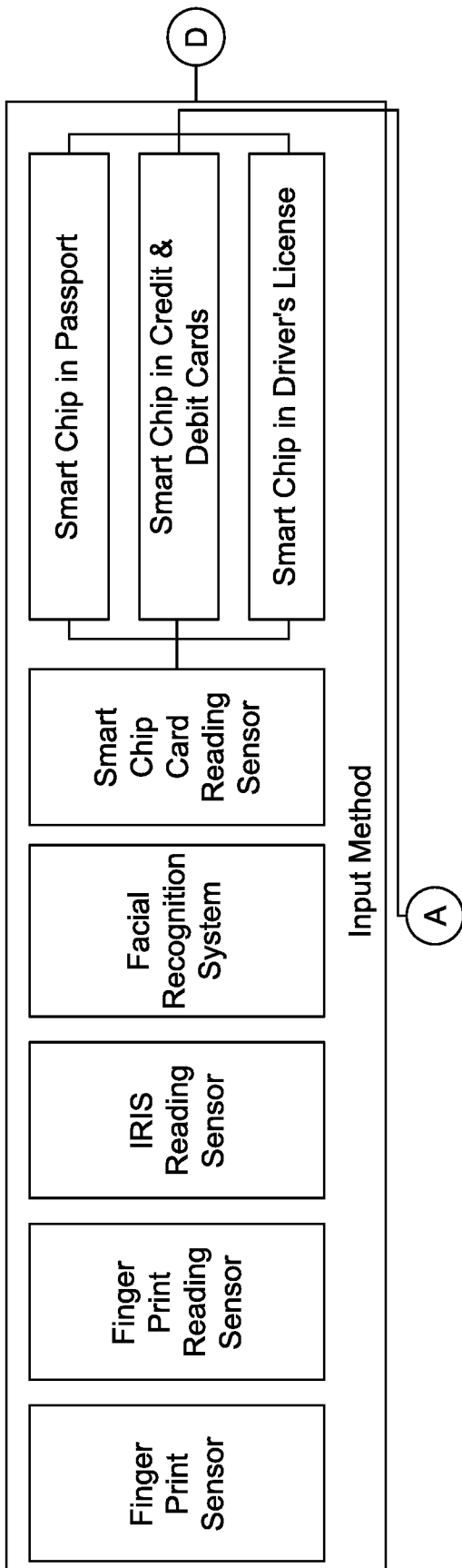


FIG. 30A

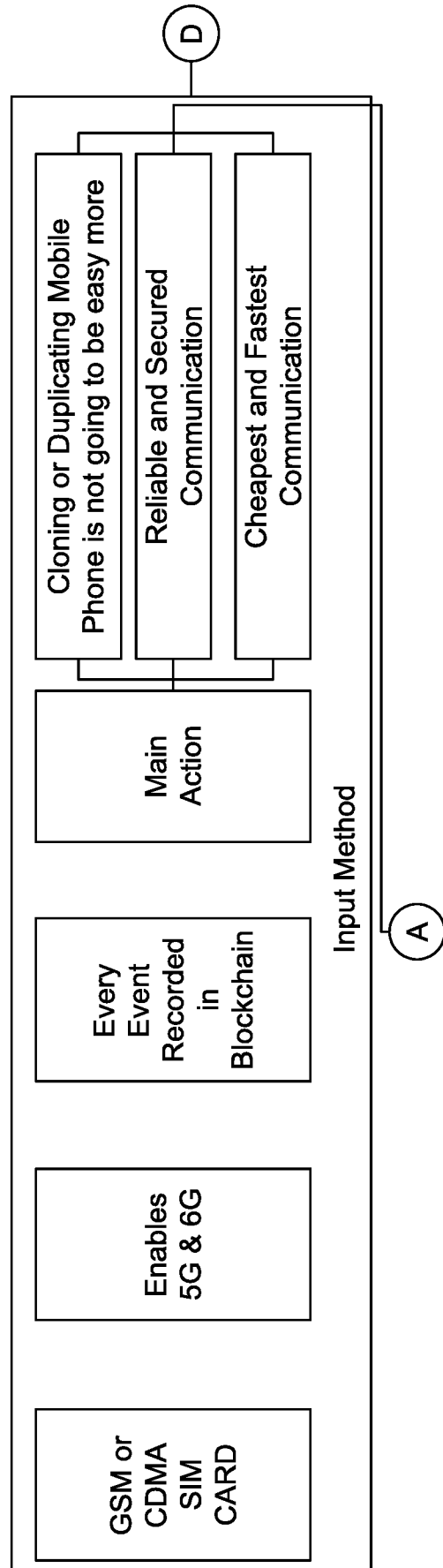


FIG. 30B

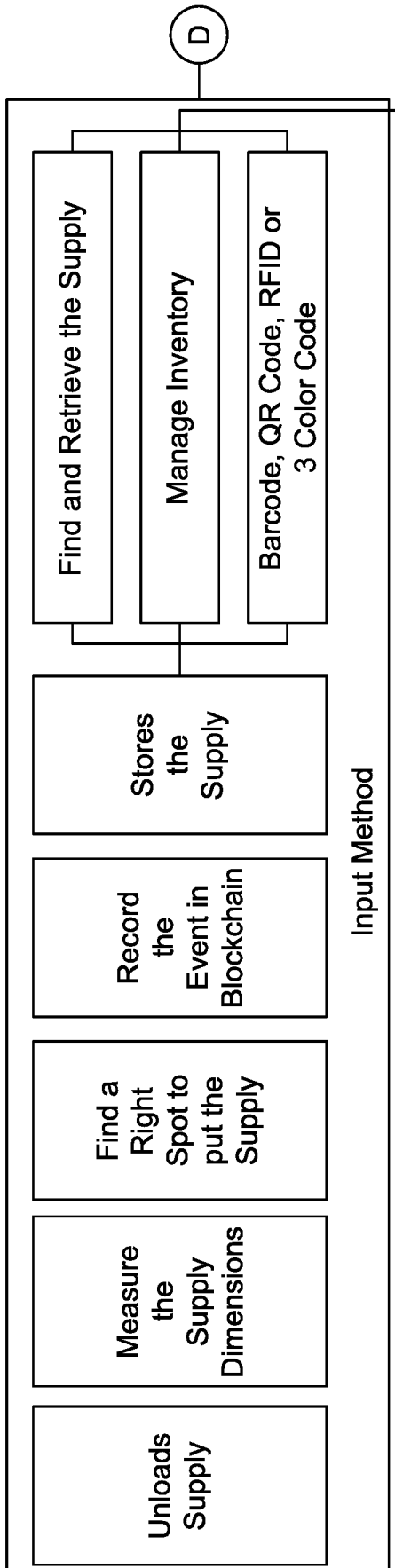


FIG. 30C

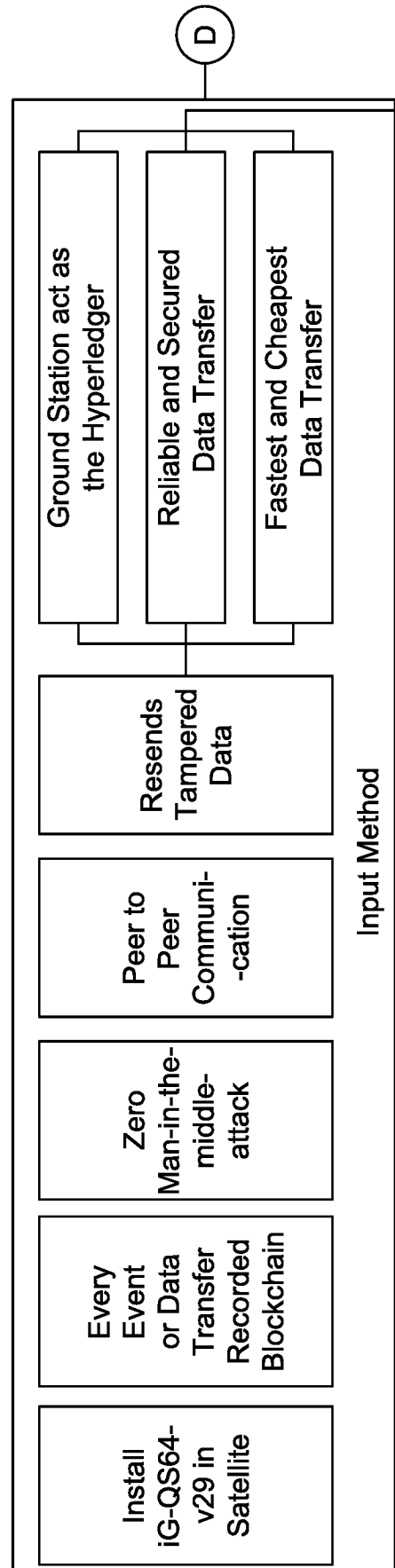


FIG. 30D

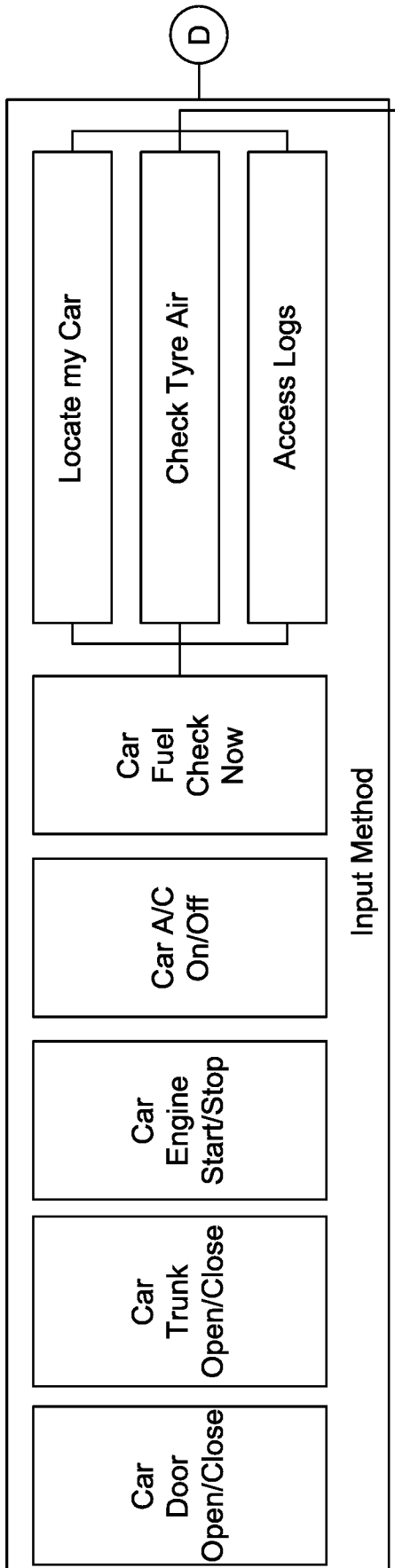


FIG. 30E

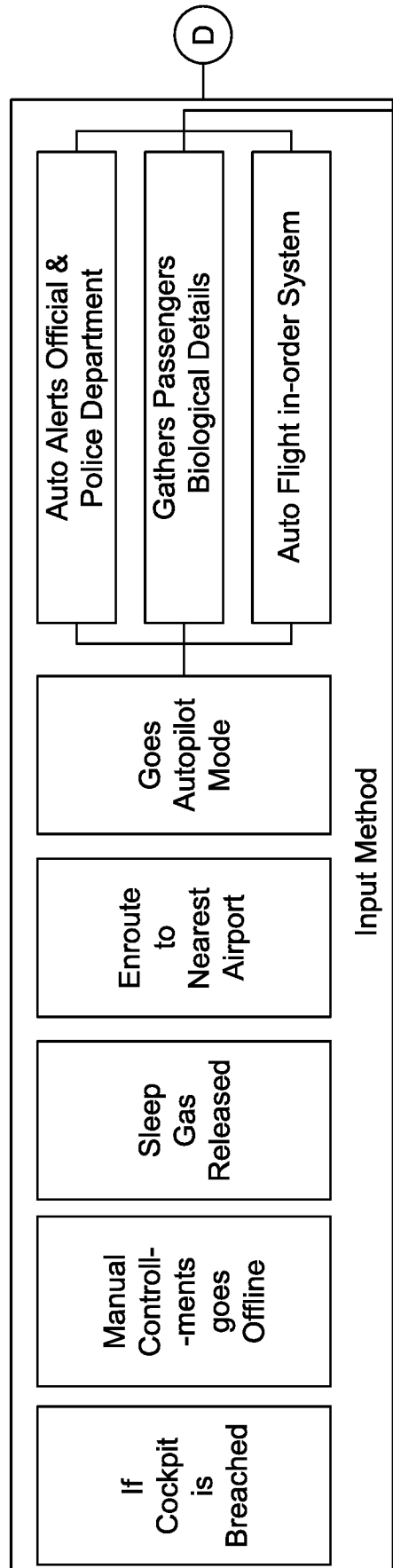


FIG. 30F

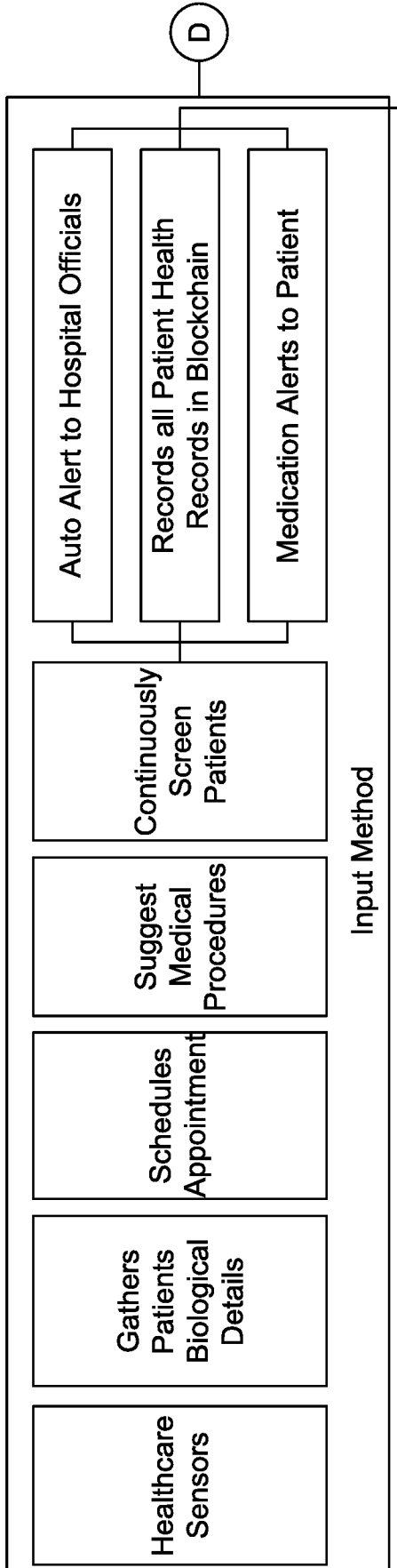


FIG. 30G

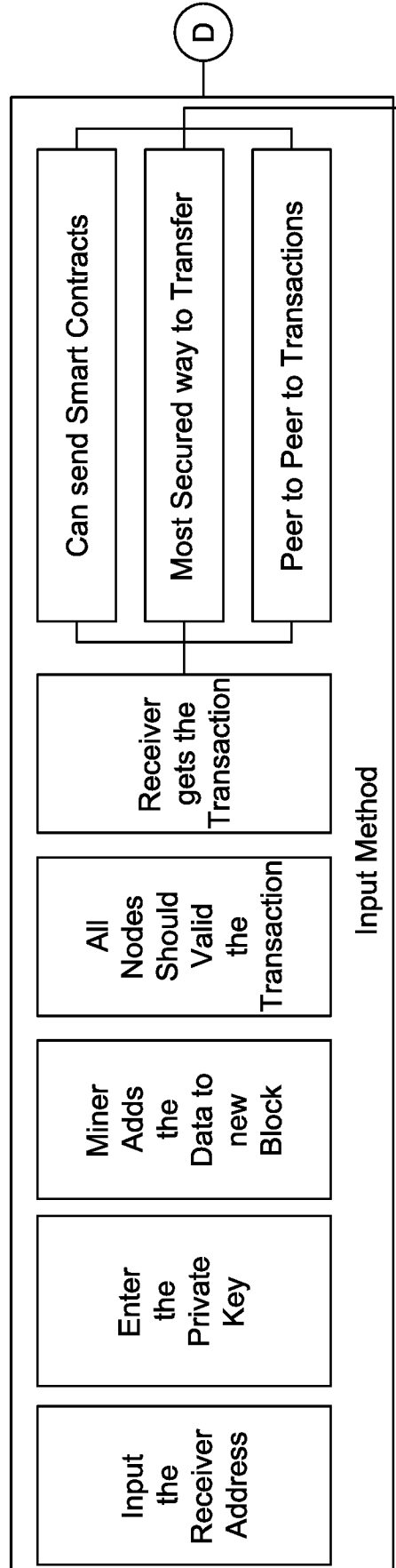


FIG. 30H

32/36

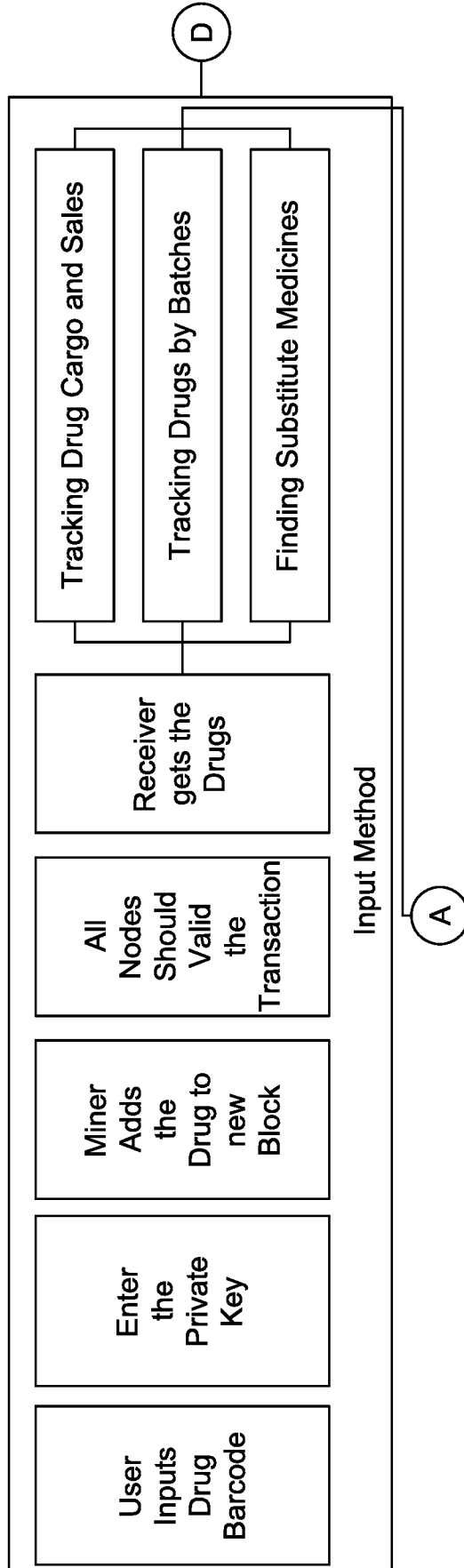


FIG. 30I

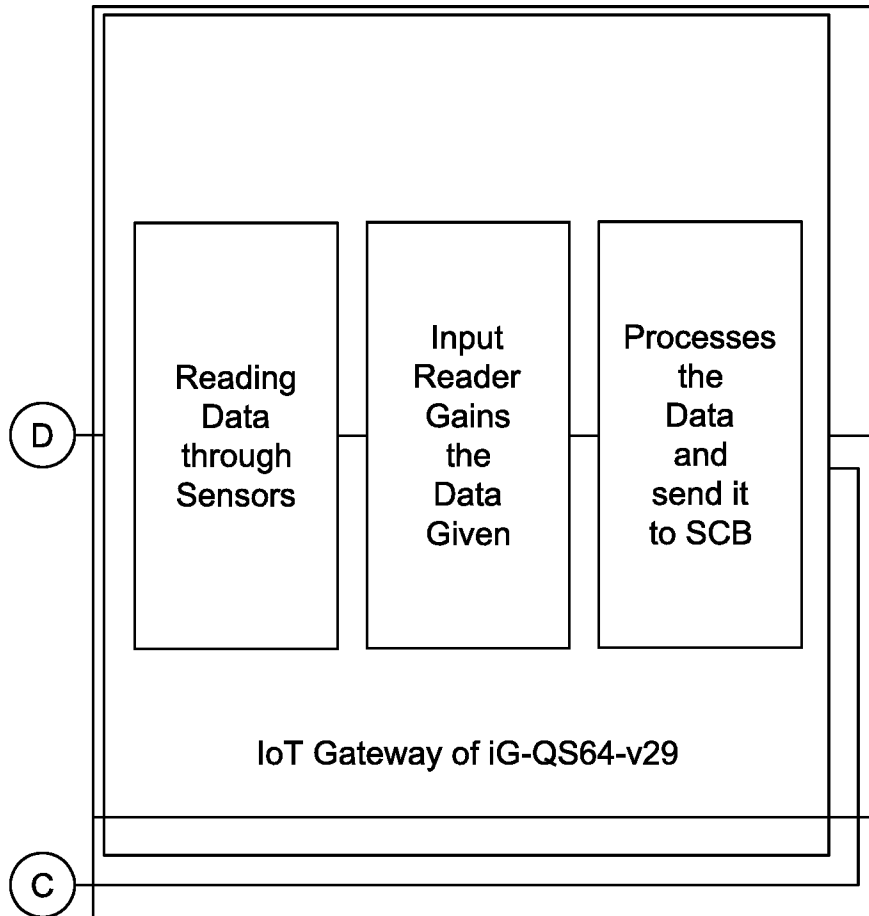


FIG. 31

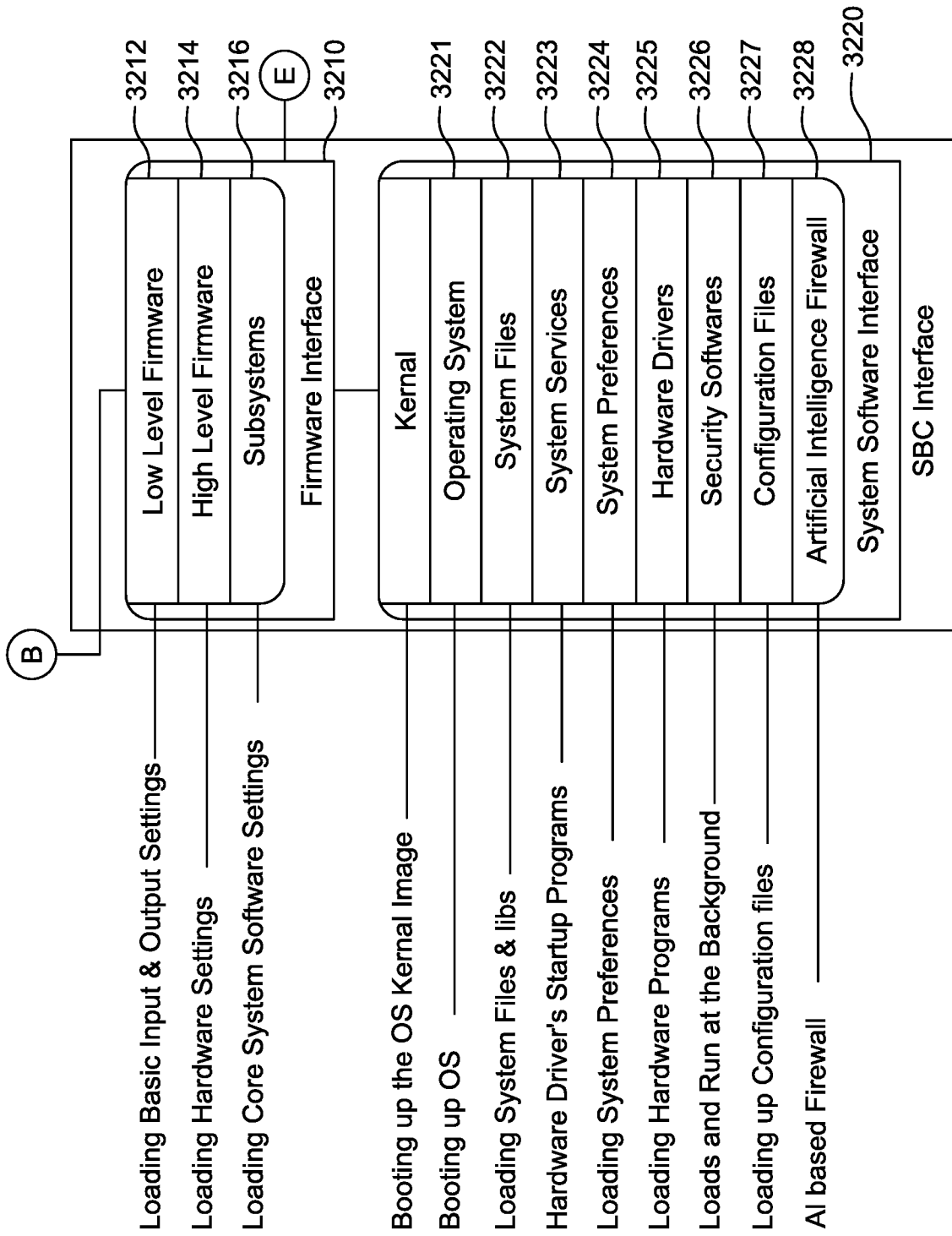


FIG. 32

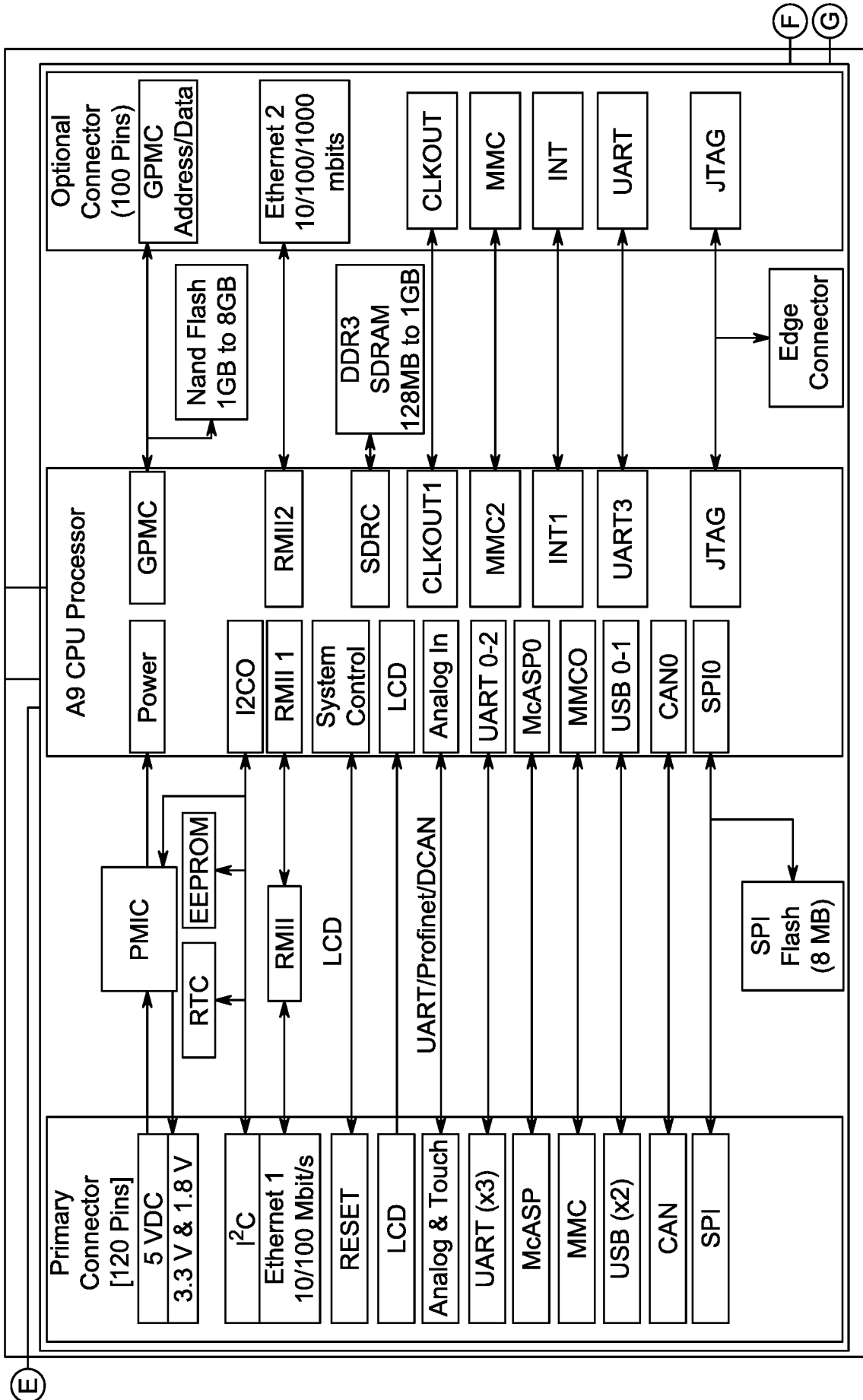


FIG. 33

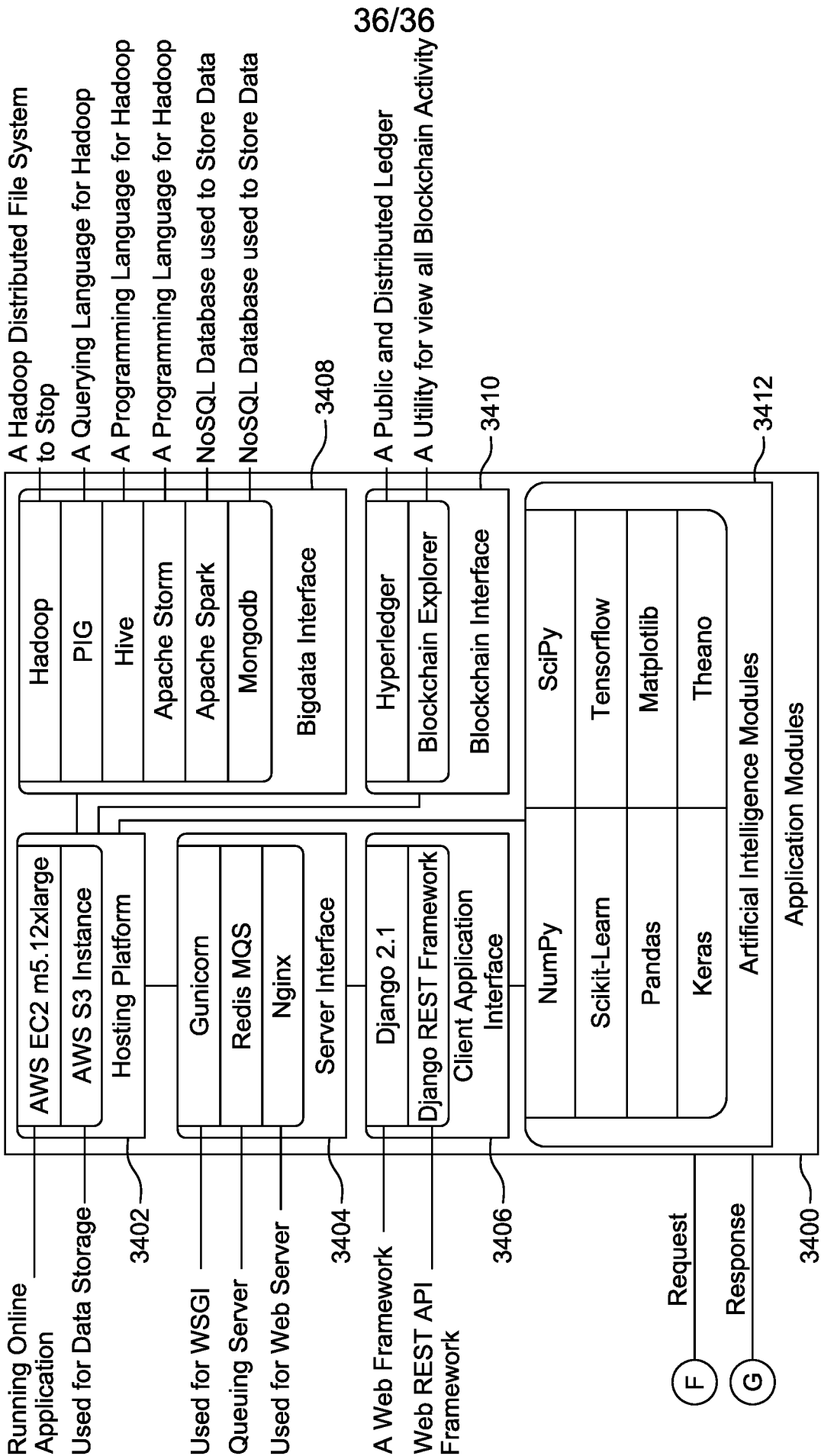


FIG. 34

INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US 19/37124

A. CLASSIFICATION OF SUBJECT MATTER  
IPC(8) - G06F 9/54; H04L 29/06; H04L 9/08; H04L 9/32 (2019.01)  
CPC - G06F 21/00; G06F 21/34; G06F 21/445; G06Q 20/065; G06Q 20/3278; G06Q 20/3678; G06Q 20/3821; G06Q 20/38215; G06Q 20/3829; G06Q 20/4097; H04L 63/0428; H04L 63/0492; H04L 63/0807; H04L 63/083; H04L 63/0853; H04L 63/0869; H04L 63/105; H04L 63/18; H04L 9/3213; H04L 9/3226; H04W 12/06; H04W 4/80

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)

See Search History Document

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

See Search History Document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

See Search History Document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X -- Y	US 2016/0261411 A1 (HOVERKEY LTD.) 08 September 2016 (08.09.2016), entire document, especially abstract and para [0002], [0022]-[0034].	1-9, 11-17 ----- 10, 18
Y	US 2017/0257358 A1 (ShoCard, Inc.) 07 September 2017 (07.09.2017), entire document, especially abstract and para [0093].	10, 18

Further documents are listed in the continuation of Box C.  See patent family annex.

\* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"E" earlier application or patent but published on or after the international filing date	"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
"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)	"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
"O" document referring to an oral disclosure, use, exhibition or other means	"&" document member of the same patent family
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search  
30 September 2019 (30.09.2019)

Date of mailing of the international search report  
**18 OCT 2019**

Name and mailing address of the ISA/US  
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents  
P.O. Box 1450, Alexandria, Virginia 22313-1450  
Facsimile No. 571-273-8300

Authorized officer:  
Lee W. Young

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

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 19/37124

**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:

(See 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.:  
1-18

**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.

In continuation of Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet):

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.

Group I: Claims 1-18, directed to an e-key system.

Group II: Claims 19-41, directed to a satellite data communication system.

The groups of inventions listed above 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 feature of the Group I invention is a key fob configured to receive the encrypted code from a mobile device and to transmit the encrypted code to a computing device incorporated into property via wireless communication; validating the encrypted code, and granting/denying access to the property based on the validating, not required by Group II.

The special technical feature of the Group II invention is transmitting/receiving an access key between at least two stations via a server and a satellite, wherein the server validates the access key, and wherein the server saves a transmission log in a hyperledger, not required by Group I.

Groups I-II share the technical features of, in various combination, transmitting a code/access key, utilizing a server, a log, and validating the code/access key.

However, these shared technical features fail to represent a contribution over the prior art of US 2016/0261411 A1 to Hoverkey Ltd. (hereinafter 'Hoverkey'), which discloses transmitting a code/access key (para [0031]-[0034], [0043] - "transmitting the encrypted authorization to a separate portable security token; on the token, decrypting the encrypted authorization and generating at least partially therefrom an unlock response"; "securely transmitting the unlock response to the mobile device"), utilizing a server (para [0498] - "a server 1230 might transmit a requested code"), a log (para [0187], [0463] - "Logs Activity logs for recent auditable events"; "block chain can be used not just for implementing cryptocurrencies but also for creating secure, distributed transaction ledgers"; "such distributed transaction ledger systems include SCP from Stellar, Hyperledger and Ethereum"), and validating the code/access key (para [0031]-[0034], [0045] - "unlocking the resource if the required unlock response and the separate authentication are both valid").

Thus, the inventions listed as Groups I-II lack unity of invention because they do not share a same or corresponding special technical feature providing a contribution over the prior art.