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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 April 2015 (02.04.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/047807 A1

(51) International Patent Classification:
G06F 21/50 (2013.01)

(21) International Application Number:
PCT/US2014/055983

(22) International Filing Date:
17 September 2014 (17.09.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/884,719 30 September 2013 (30.09.2013) US

(71) Applicant: **JVL VENTURES, LLC** [US/US]; 230 Park Avenue, 27th Floor, New York, NY 10169-0005 (US).

(72) Inventor: **WATSON, Curtis, W.**; 10734 Fishtrap Road, Aubrey, TX 73227 (US).

(74) Agents: **BERSCHADSKY, Jonathan** et al.; Fitzpatrick, Cella, Harper & Scinto, 1290 Avenue of the Americas, New York, NY 10104-3800 (US).

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

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

[Continued on next page]

(54) Title: SYSTEMS, METHODS, AND COMPUTER PROGRAM PRODUCTS FOR SECURELY MANAGING DATA ON A SECURE ELEMENT

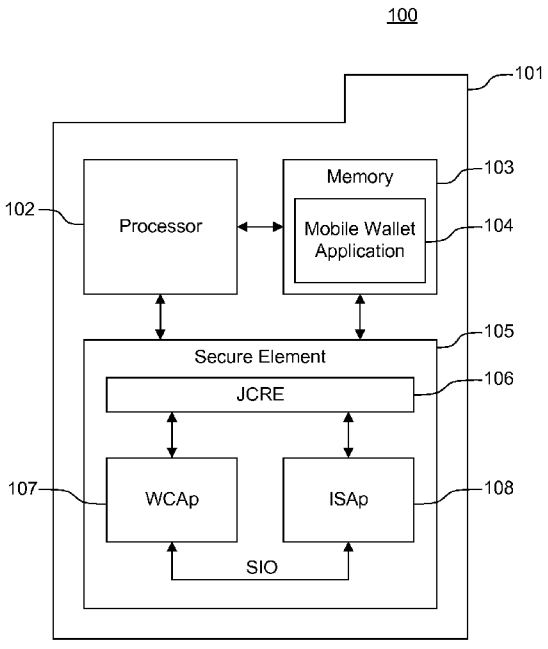

FIG. 1

(57) Abstract: Systems, methods, and computer program products are provided for managing applets. A first request to personalize the first applet is received over a communications network. A second request including a command requesting at least a portion of the second applet data is communicated to the second applet. At least a portion of the second applet data is communicated to the first applet. One or more values of the first applet data are replaced with one or more values of at least the portion of the second applet data.

WO 2015/047807 A1

Published:

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

TITLE

SYSTEMS, METHODS, AND COMPUTER PROGRAM PRODUCTS FOR SECURELY MANAGING DATA ON A SECURE ELEMENT

BACKGROUND

Field

[0001] The present invention relates generally to systems, methods, and computer program products for securely managing data on a secure element.

Related Art

[0002] Applications stored and functioning on mobile devices are increasingly being used to conduct secure communications which require the transmission of highly critical data. Such applications include mobile wallet applications, which may be used to perform contactless transactions. Contactless transactions may be

financial (*e.g.*, payments, commerce) or non-financial (*e.g.*, venue admissions, transit ticketing). These secure communications, including contactless transactions, typically involve the exchange of critical data between mobile devices and other systems such as reader terminals using, for example, near field communication (NFC) technology.

[0003] Mobile devices include, or have stored on the mobile device memory, applications used to initiate contactless transactions, as well as those applications' corresponding non-critical data. On the other hand, the applications' critical data (*e.g.*, personal data, security keys, passcodes, identifiers) is stored in a secure element (SE) associated with the mobile device. Secure elements are highly tamper resistant components which securely store data in accordance with specific security requirements. Because of their specialized security mechanisms, secure element storage is more costly than typical memory (*e.g.*, mobile device memory) and thus, storage on secure elements is often exclusively limited to critical data.

[0004] Critical data is managed by corresponding applets on the secure element which control, for example, how the data is stored, when the data can be distributed, and which devices, applets and applications can access (*e.g.*, read, write) the data. The applets which manage critical data on secure elements may need, or choose to be, altered or deleted, for example, to update out-of-date or unsupported applet versions or to repair corrupted applet versions. Such alteration or deletion of applets that manage critical data may cause those applets' corresponding critical data to be deleted or be left unmanaged on the secure element during periods in which those managing applets are not yet installed, updated or activated. Deletion of critical data may result in the need for that critical data to be requested and acquired from its source, or worse, that critical data may be lost.

[0005] Given the foregoing, it would be beneficial to store critical data on secure elements in a manner which allows for managing applets to be altered (*e.g.*, updated, deleted) without resulting in data loss or minimization of the security of the critical data.

[0006] One technical challenge involves securely storing critical data during time periods when managing applets are not fully active (*e.g.*, pending update). Another technical challenge involves managing applets receiving the most up-to-date critical data when those managing applets become fully active (*e.g.*, post-update).

BRIEF DESCRIPTION

[0007] The example embodiments presented herein meet the above-identified needs by providing systems, methods, and computer program products for securely managing data on a secure element.

[0008] In one example embodiment, a system for managing applets comprises at least one memory operable to store a first applet including first applet data and a second applet including second applet data. The system also includes a processor coupled to the at least one memory. A first request to personalize the first applet is received, over a communications network. A second request including a command requesting at least a portion of the second applet data is communicated to the second applet. At least a portion of the second applet data is communicated to the first applet. One or more values of the first applet data are replaced with one or more values of at least the portion of the second applet data.

[0009] In another example embodiment, a method for managing applets, the method includes: receiving, over a communications network, a first request to personalize a first applet; communicating a second request to a second applet, the second request including a command requesting at least a portion of second applet data; communicating at least the portion of the second applet data to the first applet; and replacing one or more values of first applet data with one or more values of at least the portion of the second applet data.

[0010] In another example embodiment, a non-transitory computer-readable medium has stored thereon sequences of instructions that, when executed by a computer processor, cause the processor to: receive, over a communications network, a first request to personalize the first applet; communicate a second request to a second applet, the second request including a command requesting at least a portion of second applet data; communicate at least the portion of the

second applet data to the first applet; and replace one or more values of first applet data with one or more values of at least the portion of the second applet data.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] The features and advantages of the example embodiments presented herein will become more apparent from the detailed description set forth below when taken in conjunction with the following drawings.

[0012] FIG. 1 is a diagram of a system for securely managing data according to an exemplary embodiment.

[0013] FIG. 2 is a sequence diagram for establishing a shareable interface object (SIO) between applets and providing authentication of the SIO-requesting applet according to an exemplary embodiment.

[0014] FIGs. 3a and 3b are sequence diagrams for replacing a WCAp on a secure element according to an exemplary embodiment.

[0015] FIG. 4 is a diagram of an example system useful for implementing the present invention.

DETAILED DESCRIPTION

I. Overview

[0016] The example embodiments presented herein are directed to systems, methods and computer program products for securely managing data on a secure element, which are described herein in terms of applets and applications for conducting contactless mobile transactions (*e.g.*, commerce and payment) in a mobile wallet environment. This description is not intended to limit the application of the example embodiments presented herein. In fact, after reading the following description, it will be apparent to one skilled in the relevant art(s) how to implement the following example embodiments for any type of applets and/or applications on mobile devices, within or outside of a mobile wallet environment.

[0017] In exemplary embodiments presented herein, a wallet companion applet (WCAp) is an applet stored on a secure element and acts as a representative of a

mobile wallet application. The mobile wallet application may use the WCAp for, among other things, securely storing and managing data (*e.g.*, critical data) in the secure element on its behalf. A secure assistant applet (ISAp) is an applet stored on the secure element and acts as a secure storage location for data of (or corresponding to) other applets including, for example, the WCAp.

[0018] The WCAp is replaced with a new WCAp (or WCAp instance). A WCAp establishes a shareable interface object (SIO) with an ISAp, over which data and communications may be exchanged. The WCAp transmits to the ISAp, over the SIO, data to be stored or backed up on behalf of the WCAp. The WCAp is deleted from the secure element, in accordance with a request received from a trusted service manager (TSM). A new WCAp package is loaded on the secure element and a new WCAp instance is created using the newly loaded WCAp package. The new WCAp (or WCAp instance) is personalized using non-critical data and/or parameters received from the TSM. The WCAp transmits a request to the ISAp to receive critical parameters stored on behalf of the WCAp that was previously deleted and/or replaced. The ISAp transmits critical parameters to the WCAp, which are, in turn, stored in or by the WCAp on the secure element. The WCAp, ISAp and SIO are explained in further detail below with reference to at least FIGS. 1-3.

II. System

[0019] FIG. 1 is a diagram of a system 100 for securely managing data, in accordance with example embodiments presented herein. As shown in FIG. 1, the system 100 includes a mobile device 101, which includes a processor 102, memory 103 and a secure element (SE) 105. The mobile device 101 may be, for example, a cellular phone, tablet or the like. Although not illustrated in FIG. 1, the mobile device 101 may include a contactless frontend (CLF), a baseband modem, and a user interface such as a display screen. A baseband modem is a digital modem that is used for wireless communications. A CLF is circuitry which handles the analog aspects of contactless or NFC and the communication protocol layers of contactless transmission link.

[0020] The mobile device 101 also includes a mobile wallet application 104, which may be stored on the memory 103 of the mobile device. The mobile

wallet application 104 includes instructions which, when executed by the processor 102 of the mobile device 101, cause the mobile device 101 to act as an instrument for processing contactless transactions and the like. The mobile wallet application 104 also includes (*e.g.*, uses, operates on, is associated with) non-critical data which may be stored on the memory 103 of the mobile device 101. Non-critical data may include information used by the mobile wallet application 104 during its functionality, including images, system information, preferences, and the like. Each application (*e.g.*, mobile wallet application 104) or entity/provider managing each application have corresponding standards that define which types of data are non-critical (as opposed to critical). The mobile wallet application 104 may also be associated with critical data, which may include codes (*e.g.*, passcodes), credentials, security keys, identifiers, and the like. Critical data is typically stored in a secure element, such as secure element 105.

[0021] Secure element 105 may be implemented as a Universal Integrated Circuit Card (UICC), embedded SE card, secure micro digital (microSD) card, and the like. Secure element 105 may also be implemented as a virtual secure element, which can be maintained outside of the mobile device 101 on a memory accessible by the mobile device 101, including but not limited to, for example, a remote server or computer, in a cloud-based architecture, and the like. A secure element (*e.g.*, secure element 105) is generally considered secure because it is a self-contained system, including dedicated memory, and is protected by hardware and software hardening techniques that are verified by independent testing.

[0022] The secure element 105 includes a Java Card Runtime Environment (JCRE) 106, which is a secure element card execution environment that allows applets stored therein to run, function and/or communicate, for example, by offering for use classes for input/output (I/O), messaging and cryptography. Such applets may include, for example, a wallet companion applet (WCAp) 107 and a secure assistant applet (ISAp) 108, as shown in FIG. 1.

[0023] The WCAp 107 is an applet stored on the secure element 105 and acts as a representative of the mobile wallet application 104. The mobile wallet application 104 may use the WCAp 107 for, among other things, securely storing

and managing data (*e.g.*, critical data) in the secure element 104 on its behalf. ISAp 108 is an applet stored on the secure element 105 and acts as a secure storage location for data of (or corresponding to) other applets including, for example, WCAp 107.

[0024] In one example embodiment, the WCAp 107 maintains and/or stores a list of data (*e.g.*, data objects, data elements) used, or which may be used, by the mobile wallet application 104. Table 1 below lists examples of data stored by the WCAp 107 and corresponding maximum data size in bytes for each data element according to an exemplary embodiment.

Table 1

Data Element	Max Size in Bytes
ICCID	> 0 and <= 16
IMEI / MEID / Device ID	> 0 and <= 64
Wallet ID	> 0 and <= 64
Wallet Passcode	8
Wallet Server Key	16
Wallet Server KVC	3
Enhanced WS Key	24
Enhanced WS KVC	3
Enhanced WS HMAC Key	32
Enhanced WS HMAC KVC	3
SIO Authentication Secret	32
Widget Authentication Blob	<= 1K
Wallet Unique Code	16

[0025] Integrated Circuit Card Identifier (ICCID) is a unique serial number or identifier (ID) corresponding to a subscriber identity module (SIM) or other secure element.

[0026] International Mobile Equipment Identifier (IMEI) refers to a unique number or identifier corresponding to a mobile device (*e.g.*, mobile phone). A Mobile Equipment Identifier (MEID) or other device ID are similar unique

numbers or identifiers corresponding to other types of mobile devices, such as those functioning on code division multiple access (CDMA) networks.

[0027] Wallet ID refers to a unique number or identifier corresponding to a wallet client (*e.g.*, mobile wallet application).

[0028] Wallet Passcode refers to a unique passcode or password used to authenticate a wallet client user. The Wallet Passcode may be a 4-character code in UNICODE.

[0029] Wallet Server Key refers to an authentication key for providing authentication between a wallet client (and/or associated applets (*e.g.*, wallet companion applet (WCAp))) and a wallet server. The Wallet Server Key may be generated, for example, in accordance with a triple data encryption algorithm (TDEA) symmetric key-block cypher or the like.

[0030] Wallet Server Key Verification Code (KVC) refers to a value for verifying or authenticating a Wallet Server Key.

[0031] Enhanced Wallet Server (WS) Key refers to a key for providing authentication between a wallet client (and/or associated applets) and a wallet server. The Enhanced WS Key may be generated, for example, in accordance with a triple data encryption algorithm (TDEA) symmetric key-block cypher or the like. Enhanced WS KVC refers to a value for verifying or authenticating an Enhanced WS Key.

[0032] Enhanced WS HMAC Key and Enhanced WK HMAC KVC refer respectively to a key and key verification code or value developed in accordance with hash message authentication code (HMAC) cryptographic functions.

[0033] SIO Authentication Secret refers to a value or code used to authenticate an applet or system requesting the establishment of a Shareable Interface Object (SIO) (*e.g.*, JavaCard SIO). Establishment of an SIO is described in more detail below with reference to FIG. 2.

[0034] Widget Authentication Blob refers to a set of stored data used to authenticate widgets (*e.g.*, application, component of an interface) requesting access to applets on secure elements. For example, a Widget Authentication Blob may be used to store a widget ID, widget signature and widget version corresponding to one or more widgets. When a widget requests access to an

applet on a secure element, the widgets data is compared to the data stored in the Widget Authentication Blob and access may be granted to the applet based on that comparison.

[0035] Wallet Client Unique Code refers to a value used by a wallet client for authentication during a transaction.

[0036] In one example embodiment, the ISAp 108 maintains and/or stores data for or on behalf of the WCAp 107. Such data maintained by the ISAp 108 typically includes data which is not stored other than in the WCAp 107. That is, the ISAp 108 acts as the sole backup or disaster recovery storage for the WCAp 107 within the secure element 105, thereby making that data accessible to the WCAp 107 in the event that the WCAp 107 is deleted, updated and/or modified and that data is needed and/or used to make the WCAp 107 (or another instance of a WCAp) functional.

[0037] Table 2 below lists examples of data stored by the WCAp 107 and an indication of whether that data is stored in the WCAp 107 only or in the WCAp 107 and the ISAp 108.

Table 2

Data Element	Storage Location
ICCID	WCAp
IMEI / MEID / Device ID	WCAp
Wallet ID	WCAp
Wallet Passcode	WCAp and ISAp
Wallet Server Key	WCAp and ISAp
Wallet Server KVC	WCAp
Enhanced WS Key	WCAp and ISAp
Enhanced WS KVC	WCAp
Enhanced WS HMAC Key	WCAp and ISAp
Enhanced WS HMAC KVC	WCAp
SIO Authentication Secret	WCAp
Widget Authentication Blob	WCAp

Wallet Unique Code	WCAp and ISAp

[0038] Applets in the secure element 105, such as WCAp 107 and ISAp 108, may communicate with or among each other to exchange information. For example, WCAp 107 may communicate with ISAp 108 to obtain and/or retrieve data that is needed for the WCAp 107 to be personalized. In one example embodiment, applets may communicate using an SIO or the like.

A. SIO Establishment and Applet Authentication

[0039] FIG. 2 is a sequence diagram 200 for establishing an SIO between applets and providing authentication of the SIO-requesting applet. In particular, in FIG. 2, an SIO is established between a WCAp 201 (*e.g.*, FIG. 1, WCAp 107) and an ISAp 202 (*e.g.*, FIG. 1, ISAp 108), and the requesting applet WCAp 201 is authenticated. Although not illustrated in FIG. 2, the WCAp 201 and ISAp 202 may communicate via the runtime environment (*e.g.*, FIG. 1, JCRE 106) on which the applets are deployed. It should be understood that this process may be used to establish an SIO between and authenticate any two applets.

[0040] At step 250, the WCAp 201 transmits a "request SIO" command to the ISAp 202. The "request SIO" message may include an application identifier (AID) corresponding to the transmitting and/or requesting applet, *i.e.*, WCAp 201. At step 252, the ISAp 202 checks whether the received AID corresponds to an expected and/or authorized applet and, if so, transmits a "send SIO" message to the WCAp 201, at step 254. The send SIO message may include information associated with the established SIO.

[0041] In turn, at step 256, the WCAp 201 transmits a "get challenge" command to the ISAp 202, requesting that a challenge be returned to the WCAp 201. The ISAp 202, in response to receiving the get challenge command, generates a challenge at step 258. A challenge may be a random value such as an 8-byte random number. At step 260, the ISAp 202 transmits the generated challenge to the WCAp 201.

[0042] At step 262, the WCAp 201 uses the received challenge to generate an authentication message. The authentication message may be made up of a

combination of all or a portion of data available to or known by the WCAp 201 and the ISAp 202, including the challenge generated at step 258, shared authentication keys, and/or the AID of the WCAp 201. The authentication message may be generated using symmetric key algorithms such as data encryption standard (DES).

[0043] In turn, the WCAp 201 transmits to the ISAp 202, at step 264, the generated authentication message. The ISAp 202, at step 266, checks the received authentication message by (1) generating a comparison authentication message using the same data and algorithm expected to have been used by the WCAp 201 at step 262, (2) comparing the comparison authentication message to the authentication message received from the WCAp 201, and (3) determining whether the comparison authentication message and the authentication message received from the WCAp 201 match.

[0044] At step 268, the ISAp 202 transmits an authentication response to the WCAp 201. For example, if the two authentication messages are determined to be a match at step 266, the ISAp 202 transmits an authentication response indicating that access by the WCAp 201 to the ISAp 202 is granted. Otherwise, the ISAp 202 may transmit an authentication response indicating access is not granted and/or a reason for why access is not granted.

[0045] Data stored by the WCAp 107 in FIG. 1 may be transmitted or provided to the WCAp 107 via commands during a personalization phase. For example, those commands may be application protocol data unit (APDU) commands such as a "store data" command issued by a trusted service manager (TSM) and used to store data in the WCAp 107.

[0046] Once the WCAp 107 is personalized (*e.g.*, loaded with data), the WCAp 107 populates the ISAp 108 with data typically stored (*e.g.*, expected) by the ISAp 108, as discussed above with reference to Table 2. That is, the WCAp 107, when personalized, may transmit data to the ISAp 108, which is operable to store or back up data on behalf of the WCAp 107.

[0047] To determine whether the ISAp 108 is populated and/or needs to be populated by the WCAp 107, the WCAp 107 may call a method (or function) such as:

public abstract void ISAaToWCApReport (byte Code)

[0048] That method (*e.g.*, ISAaToWCApReport) allows the ISAp 108 to report to the WCAp 107 by returning a code (*e.g.*, byte Code) indicating, for example, whether (1) the ISAp 108 is being installed with no data (*i.e.*, ISAp 108 is empty), (2) the ISAp 108 is being deleted, or (3) data needs to be uploaded, transmitted or provided to the ISAp 108 (*i.e.*, at least some expected data is missing from the ISAp 108).

III. Process

[0049] FIGs. 3a and 3b are sequence diagrams 300a and 300b, respectively, of processes for replacing a WCAp (*e.g.*, FIG. 1, WCAp 107) stored on a secure element (*e.g.*, FIG. 1; secure element 105) of a mobile device (*e.g.*, FIG. 1; mobile device 101), according to an exemplary embodiment. It should be understood that the above process may be used to replace other types of applets (or applet data) on secure elements of any form factor, including secure elements within or outside of a mobile device. It should also be understood that replacement of data can be performed via transfers between devices in a number of ways. For example, such data transfer can be achieved over-the-air or by sideloading (*e.g.*, via USB, Bluetooth, etc.). Sideloading generally refers to the transfer of data, via an upload or download, between two devices (*e.g.*, mobile device, secure element)

[0050] As discussed above with reference to FIG. 1, a WCAp (*e.g.*, WCAp 301) includes and/or stores data along the lines of that shown in Table 1. Typically, ISAp 302 (*e.g.*, FIG. 1, ISAp 108) stores data on behalf of the WCAp 301 along the lines of that shown in Table 2. At any time during the lifecycle of the WCAp 301, the WCAp 301 may obtain a report from ISAp 302 to determine if ISAp 302 is missing any data. For example, the WCAp 301 may obtain a report from the ISAp 302 prior to the deletion and/or replacement of the WCAp 301. This can be accomplished by the WCAp 301, for example, by calling the ISAaToWCApReport function described above. If that function returns a code indicating that at least some data expected to be stored on the ISAp 302 is

missing, the WCAp 301 may update and/or replenish the ISAp 302 so that it contains proper critical data (as deemed necessary by the WCAp 301).

[0051] FIG. 3a is a sequence diagram 300a for updating (or replenishing) an ISAp (*e.g.*, ISAp 302) in accordance with an exemplary embodiment. The WCAp 301 may establish an SIO with the ISAp 302 as described above with reference to FIG. 2. In particular, at step 350, the WCAp 301 transmits a "request SIO" command to the ISAp 302 indicating that the WCAp 301 would like to communicate over an SIO. The "request SIO" command may include an AID corresponding to the WCAp 301. The ISAp 302 validates the "request SIO" command, for example, by determining whether the AID and/or any data received in that command corresponds to a trusted or known applet, based on information stored by the ISAp 302 such as a list or table of trusted applets and/or corresponding AIDs.

[0052] If the ISAp 302 validates the "request SIO" command received at step 350, in returns, at step 352, an SIO (including associated information) over which the WCAp 301 and the ISAp 302 may communicate. The request SIO command may include the AID of the requesting applet (*e.g.*, WCAp 301). The ISAp 302 may validate the request SIO command by determining whether the AID included in the command matches an authorized or expected AID.

[0053] At step 354, the WCAp 301 requests a challenge by transmitting a "get challenge" command to the ISAp 302. The ISAp 302, in turn, returns a challenge to the WCAp 301 at step 356. In turn, WCAp 301 transmits an authentication message to the ISAp 302. The ISAp 302 analyzes the authentication message sent at step 356 and, if authentication is successful (*e.g.*, authentication message matches expected value), the ISAp 302 transmits, at step 360, an authentication response to the WCAp 301 indicating that authentication passed and access to the ISAp 302 is granted.

[0054] At step 362, the WCAp 301 transmits a "put data" command (or the like) to the ISAp 302, including information which WCAp 301 would like updated and/or replenished on the ISAp 302. For example, the information in the put data command may include a passcode, WC unique code, or any other information typically stored on the ISAp 302. In turn, at step 364, the ISAp 302 transmits a

response to the WCAp 301 indicating whether or not the information transmitted in the put data command was successfully added to and/or stored by the ISAp 302, as requested by the WCAp 301.

[0055] FIG. 3b is a sequence diagram 300b for replacing a WCAp (*e.g.*, WCAp 301) stored on a secure element (*e.g.*, secure element 303) in accordance with an exemplary embodiment. When replacing the WCAp 301, applets (*e.g.*, payment applets) associated with the WCAp 301 and which may be stored on the secure element 303, are placed into a locked state. This may be done at any point during the replacement of the WCAp 301 but is typically performed prior to initiating a WCAp replacement process, to ensure that any associated applets are not misused during the time that the WCAp 301 is not functional (*e.g.*, while it is being replaced with a new WCAp or WCAp instance). For example, the state of each applet may be changed from active to locked to prevent their use. It should be understood that applets associated with the WCAp 301 (or associated with any applet being replaced) may be locked using any processes executed by any applets or applications having such privileges.

[0056] At step 380, a trusted service manager (TSM) 305 (or any system having applet management privileges for the WCAp 301) transmits a delete command to the secure element 303, to delete the WCAp 301. The delete command may include an AID corresponding to the applet to be deleted (*e.g.*, WCAp 301). In turn, at step 382, the secure element 303 deletes the WCAp 301. Although not illustrated, the secure element 303 may transmit a notification to the TSM 305 indicating whether or not the WCAp 301 was successfully deleted. In one exemplary embodiment, the TSM 305 may transmit communications to the secure element 303 via a central security domain (not illustrated) on the secure element 303.

[0057] In turn, at step 384, the TSM 305 transmits a load command to the secure element 303. The load command includes instructions to load a WCAp package on the secure element 303. The load command may include the WCAp package to be loaded on the secure element 303. At step 386, the package is loaded on the secure element 303. Although not illustrated, the secure element 303 may

transmit a notification to the TSM 305 indicating whether or not the WCAp package was successfully loaded on the secure element 303.

[0058] At step 388, the WCAp package loaded at step 384 is instantiated on the secure element 303 to create WCAp 304 on the secure element 303. Typically, instantiation includes creating an applet instance from a loaded package and, if necessary, extraditing the created applet instance to a storage area on a secure element (*e.g.*, a corresponding security domain). At step 390, the package loaded at step 386 is used to create a new WCAp instance (*i.e.*, WCAp 304), and that instance may be extradited to a security domain on the secure element 303. Although not illustrated, the secure element 303 may transmit a notification to the TSM 305 indicating whether or not a new WCAp instance was successfully created (and, if necessary, extradited) on the secure element 303.

[0059] Once the WCAp 304 has been created, the TSM 305 transmits, at step 392, a personalization command to the secure element 303 to personalize the WCAp 304. The personalization command may include data to be stored on or by the WCAp 304. Such data may include non-critical parameters stored by the WCAp 304, as outlined in Tables 1 and 2. Non-critical parameters are those solely stored by a WCAp and not backed up by an ISAp. In particular, the personalization command transmitted at step 392 may include, for example, ICCID, IMEI, and wallet ID. At step 394, the secure element 303 uses the data (*e.g.*, non-critical parameters) received in the personalization command to personalize the WCAp 304, for example, by calling a StoreData command. Specifically, the data received in the personalization command is stored in, by, or in association with the WCAp 304.

[0060] In turn, at step 396, the WCAp 304 in the secure element 303 transmits a get data command or the like to an associated ISAp (*e.g.*, ISAp 302), to retrieve critical parameters stored by the ISAp 302. Examples of critical parameters stored by an ISAp (*e.g.*, ISAp 302) are described above with reference to Table 2. In one exemplary embodiment, the WCAp 304 may establish an SIO with and/or be authenticated by the ISAp 302 prior to the exchange of critical parameters and/or other data. Establishing an SIO and/or authenticating an ISAp (*e.g.*, ISAp 302) is/are described above in more detail with reference to FIG. 2. The get data

command transmitted at step 396 may include an indication of the types of data and/or parameters requested by the WCAp 304.

[0061] If the ISAp 302 determines that it has stored thereon some or all of data requested by the WCAp 304 in the get data command, the ISAp 302 retrieves and transmits, at step 398, some or all of the requested data (*e.g.*, critical parameters) to the WCAp 304. Alternatively, and although not illustrated in FIG. 3, the ISAp 302 may transmit a notification to the WCAp 304 indicating, for example, whether or not (1) the ISAp 302 includes or has stored thereon the data requested by the WCAp 304, or (2) processing of the get data command transmitted at step 396 was successful.

[0062] In an exemplary embodiment, applets that were associated with the WCAp 301 prior to it being replaced with WCAp 304, may be unlocked and placed in a usable or active state, if they were or had been placed in a locked state. In particular, applets, if any, that were locked to prevent their functionality during the replacement of WCAp 301 may be unlocked to allow for their operability to be resumed. It should be understood that unlocking applets may be achieved in any manner as desired by an applet owner or provider.

IV. Computer Readable Medium Implementation

[0063] The example embodiments described above such as, for example, the systems and procedures depicted in or discussed in connection with FIGs. 1-3 or any part or function thereof, may be implemented by using hardware, software or a combination of the two. The implementation may be in one or more computers or other processing systems. While manipulations performed by these example embodiments may have been referred to in terms commonly associated with mental operations performed by a human operator, no human operator is needed to perform any of the operations described herein. In other words, the operations may be completely implemented with machine operations. Useful machines for performing the operation of the example embodiments presented herein include general purpose digital computers or similar devices.

[0064] FIG. 4 is a block diagram of a general and/or special purpose computer 400, in accordance with some of the example embodiments of the invention. The

computer 400 may be, for example, a user device, a user computer, a client computer and/or a server computer, among other things.

[0065] The computer 400 may include without limitation a processor device 410, a main memory 425, and an interconnect bus 405. The processor device 410 may include without limitation a single microprocessor, or may include a plurality of microprocessors for configuring the computer 400 as a multi-processor system. The main memory 425 stores, among other things, instructions and/or data for execution by the processor device 410. The main memory 425 may include banks of dynamic random access memory (DRAM), as well as cache memory.

[0066] The computer 400 may further include a mass storage device 430, peripheral device(s) 440, portable storage medium device(s) 450, input control device(s) 480, a graphics subsystem 460, and/or an output display 470. For explanatory purposes, all components in the computer 400 are shown in FIG. 4 as being coupled via the bus 405. However, the computer 400 is not so limited. Devices of the computer 400 may be coupled via one or more data transport means. For example, the processor device 410 and/or the main memory 425 may be coupled via a local microprocessor bus. The mass storage device, 430, peripheral device(s) 440, portable storage medium device(s) 450, and/or graphics subsystem 460 may be coupled via one or more input/output (I/O) buses. The mass storage device 430 may be a nonvolatile storage device for storing data and/or instructions for use by the processor device 410. The mass storage device 430 may be implemented, for example, with a magnetic disk drive or an optical disk drive. In a software embodiment, the mass storage device 430 is configured for loading contents of the mass storage device 430 into the main memory 425.

[0067] The portable storage medium device 450 operates in conjunction with a nonvolatile portable storage medium, such as, for example, a compact disc read only memory (CD-ROM), to input and output data and code to and from the computer 400. In some embodiments, the software for storing an internal identifier in metadata may be stored on a portable storage medium, and may be inputted into the computer 400 via the portable storage medium device 450. The peripheral device(s) 440 may include any type of computer support device, such as, for example, an input/output (I/O) interface configured to add additional

functionality to the computer 400. For example, the peripheral device(s) 440 may include a network interface card for interfacing the computer 400 with a network 420.

[0068] The input control device(s) 480 provide a portion of the user interface for a user of the computer 400. The input control device(s) 480 may include a keypad and/or a cursor control device. The keypad may be configured for inputting alphanumeric characters and/or other key information. The cursor control device may include, for example, a mouse, a trackball, a stylus, and/or cursor direction keys. In order to display textual and graphical information, the computer 400 may include the graphics subsystem 460 and the output display 470. The output display 470 may include a cathode ray tube (CRT) display and/or a liquid crystal display (LCD). The graphics subsystem 460 receives textual and graphical information, and processes the information for output to the output display 470.

[0069] Each component of the computer 400 may represent a broad category of a computer component of a general and/or special purpose computer. Components of the computer 400 are not limited to the specific implementations provided here.

[0070] Portions of the example embodiments of the invention may be conveniently implemented by using a conventional general purpose computer, a specialized digital computer and/or a microprocessor programmed according to the teachings of the present disclosure, as is apparent to those skilled in the computer art. Appropriate software coding may readily be prepared by skilled programmers based on the teachings of the present disclosure.

[0071] Some embodiments may also be implemented by the preparation of application-specific integrated circuits, field programmable gate arrays, or by interconnecting an appropriate network of conventional component circuits.

[0072] Some embodiments include a computer program product. The computer program product may be a storage medium or media having instructions stored thereon or therein which can be used to control, or cause, a computer to perform any of the procedures of the example embodiments of the invention. The storage medium may include without limitation a floppy disk, a mini disk, an optical

disc, a Blu-ray Disc, a DVD, a CD-ROM, a micro-drive, a magneto-optical disk, a ROM, a RAM, an EPROM, an EEPROM, a DRAM, a VRAM, a flash memory, a flash card, a magnetic card, an optical card, nanosystems, a molecular memory integrated circuit, a RAID, remote data storage/archive/warehousing, and/or any other type of device suitable for storing instructions and/or data.

[0073] Stored on any one of the computer readable medium or media, some implementations include software for controlling both the hardware of the general and/or special computer or microprocessor, and for enabling the computer or microprocessor to interact with a human user or other mechanism utilizing the results of the example embodiments of the invention. Such software may include without limitation device drivers, operating systems, and user applications. Ultimately, such computer readable media further include software for performing example aspects of the invention, as described above.

[0074] Included in the programming and/or software of the general and/or special purpose computer or microprocessor are software modules for implementing the procedures described above.

[0075] While various example embodiments of the invention have been described above, it should be understood that they have been presented by way of example, and not limitation. It is apparent to persons skilled in the relevant art(s) that various changes in form and detail can be made therein. Thus, the invention should not be limited by any of the above described example embodiments, but should be defined only in accordance with the following claims and their equivalents.

[0076] In addition, it should be understood that the figures are presented for example purposes only. The architecture of the example embodiments presented herein is sufficiently flexible and configurable, such that it may be utilized and navigated in ways other than that shown in the accompanying figures. Further, the purpose of the Abstract is to enable the U.S. Patent and Trademark Office and the public generally, and especially the scientists, engineers and practitioners in the art who are not familiar with patent or legal terms or phraseology, to determine quickly from a cursory inspection the nature and essence of the technical disclosure of the application. The Abstract is not intended to be

limiting as to the scope of the example embodiments presented herein in any way. It is also to be understood that the procedures recited in the claims need not be performed in the order presented.

WHAT IS CLAIMED IS:

1. A system for managing applets comprising:

at least one memory operable to store a first applet including first applet data and a second applet including second applet data; and

a processor coupled to the at least one memory, the processor being operable to:

receive, over a communications network, a first request to personalize the first applet;

communicate a second request to the second applet, the second request including a command requesting at least a portion of the second applet data;

communicate at least the portion of the second applet data to the first applet; and

replace one or more values of the first applet data with one or more values of at least the portion of the second applet data.

2. The system of Claim 1, wherein the first applet data and the second applet data each comprise cryptographic parameters including at least one of: (1) a passcode, and (2) a mobile wallet client unique code.

3. The system of Claim 2, wherein the first applet is a set of instructions stored on the at least one memory, which when executed by the processor, cause the processor to manage the first applet data.

4. The system of Claim 2, wherein the second applet is a set of instructions stored on the at least one memory, which when executed by the processor, cause the processor to manage the second applet data.

5. The system of Claim 1, wherein the system is included in a secure element associated with a mobile device.

6. The system of Claim 1, wherein the processor is further operable to:

communicate a notification to the first applet including information indicating that the second applet data is incomplete;

determine incomplete data of the second applet data;

communicate, from the first applet to the second applet, third applet data corresponding to the incomplete data of the second applet data; and

replace values of the second applet data with values of the third applet data.

7. The system of Claim 1, wherein the first applet data and the second applet data are exclusively stored and managed by the first applet and the second applet, respectively.

8. A method for managing applets, the method comprising steps of:

receiving, over a communications network, a first request to personalize a first applet;

communicating a second request to a second applet, the second request including a command requesting at least a portion of second applet data;

communicating at least the portion of the second applet data to the first applet; and

replacing one or more values of first applet data with one or more values of at least the portion of the second applet data.

9. The method of Claim 8, wherein the first applet data and the second applet data each comprise cryptographic parameters including at least one of: (1) a passcode, and (2) a mobile wallet client unique code.

10. The method of Claim 9, wherein the first applet is a set of instructions stored on the at least one memory, which when executed by a processor, cause the processor to manage the first applet data.

11. The method of Claim 9, wherein the second applet is a set of instructions stored on the at least one memory, which when executed by a processor, cause the processor to manage the second applet data.

12. The method of Claim 8, wherein the first applet and the second applet are stored in a secure element associated with a mobile device.

13. The method of Claim 8, further comprising steps of:

communicating a notification to the first applet including information indicating that the second applet data is incomplete;

determining incomplete data of the second applet data;

communicating, from the first applet to the second applet, third applet data corresponding to the incomplete data of the second applet data; and

replacing values of the second applet data with values of the third applet data.

14. The method of Claim 8, wherein the first applet data and the second applet data are exclusively stored and managed by the first applet and the second applet, respectively.

15. A non-transitory computer-readable medium having stored thereon sequences of instructions that, when executed by a computer processor, cause the processor to:

receive, over a communications network, a first request to personalize the first applet;

communicate a second request to a second applet, the second request including a command requesting at least a portion of second applet data;

communicate at least the portion of the second applet data to the first applet; and

replace one or more values of first applet data with one or more values of the at least a portion of the second applet data.

16. The non-transitory computer-readable medium of Claim 15, wherein the first applet data and the second applet data each comprise cryptographic parameters including at least one of: (1) a passcode, and (2) a mobile wallet client unique code.

17. The non-transitory computer-readable medium of Claim 16, wherein the first applet is a set of instructions stored on a memory, which when executed by the computer processor, cause the computer processor to manage the first applet data.

18. The non-transitory computer-readable medium of Claim 16, wherein the second applet is a set of instructions stored on a memory, which when executed by the computer processor, cause the computer processor to manage the second applet data.

19. The non-transitory computer-readable medium of Claim 15, wherein the first applet and the second applet are included in a secure element associated with a mobile device.

20. The non-transitory computer-readable medium of Claim 15, wherein the sequences of instructions, when executed by the computer processor, further cause the processor to:

communicate a notification to the first applet including information indicating that the second applet data is incomplete;

determine incomplete data of the second applet data;

communicate, from the first applet to the second applet, third applet data corresponding to the incomplete data of the second applet data; and

replace values of the second applet data with values of the third applet data.

21. The non-transitory computer-readable medium of Claim 15, wherein the first applet data and the second applet data are exclusively stored and managed by the first applet and the second applet, respectively.

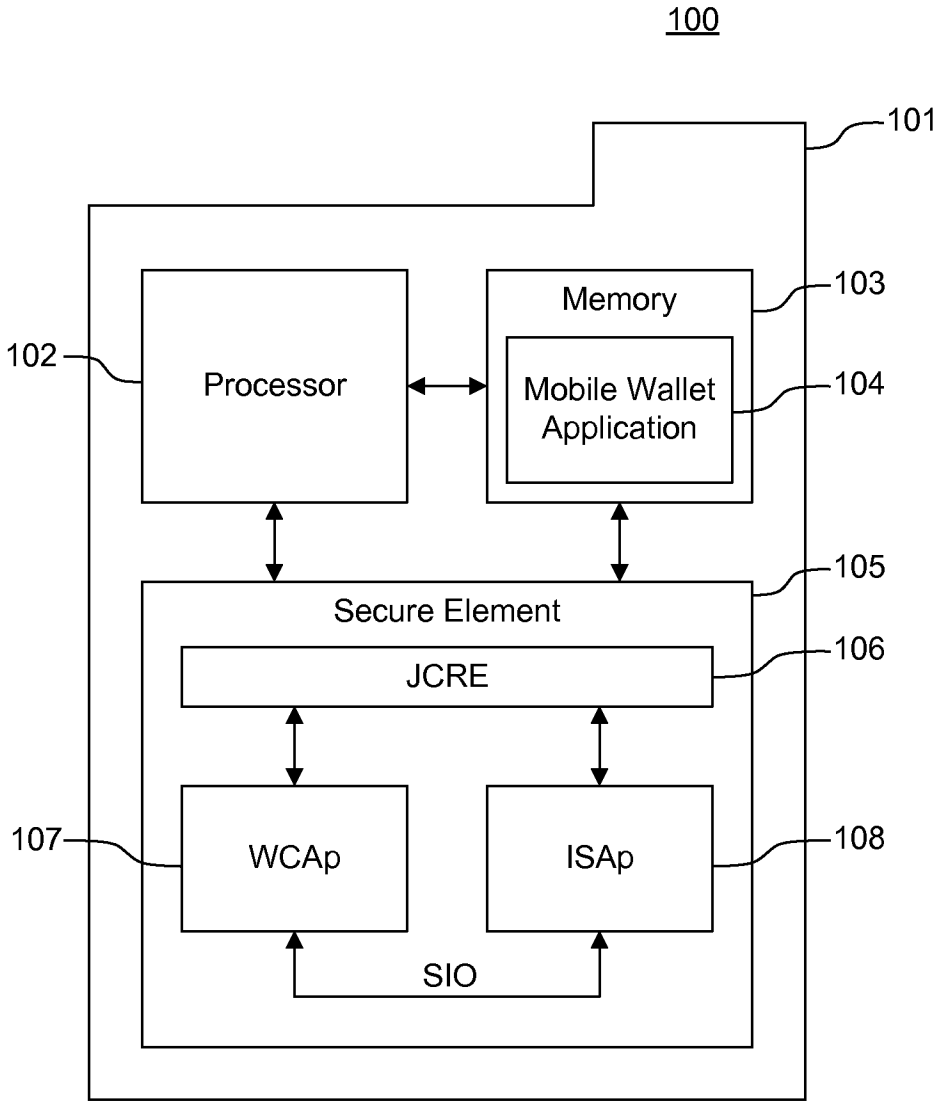
100
101
Memory
103
102
Processor
Mobile Wallet Application
104
105
Secure Element
106
JCRE
107
WCAp
ISAp
108
SIO

FIG. 1

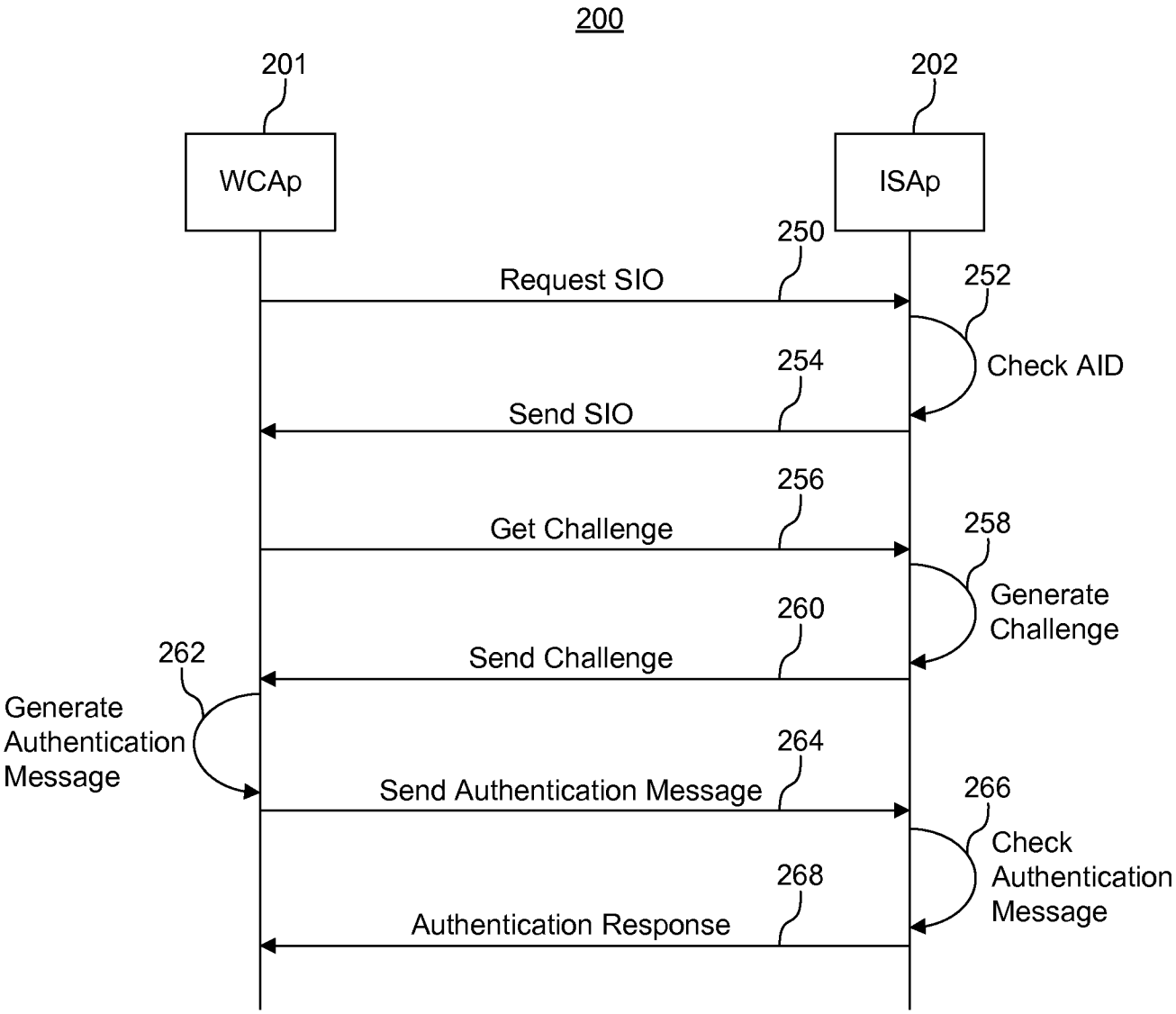
200
201
202
WCAp
ISAp
250
Request SIO
252
Check AID
254
Send SIO
256
Get Challenge
258
Generate Challenge
260
Send Challenge
262
Generate Authentication Message
264
Send Authentication Message
266
Check Authentication Message
268
Authentication Response

FIG. 2

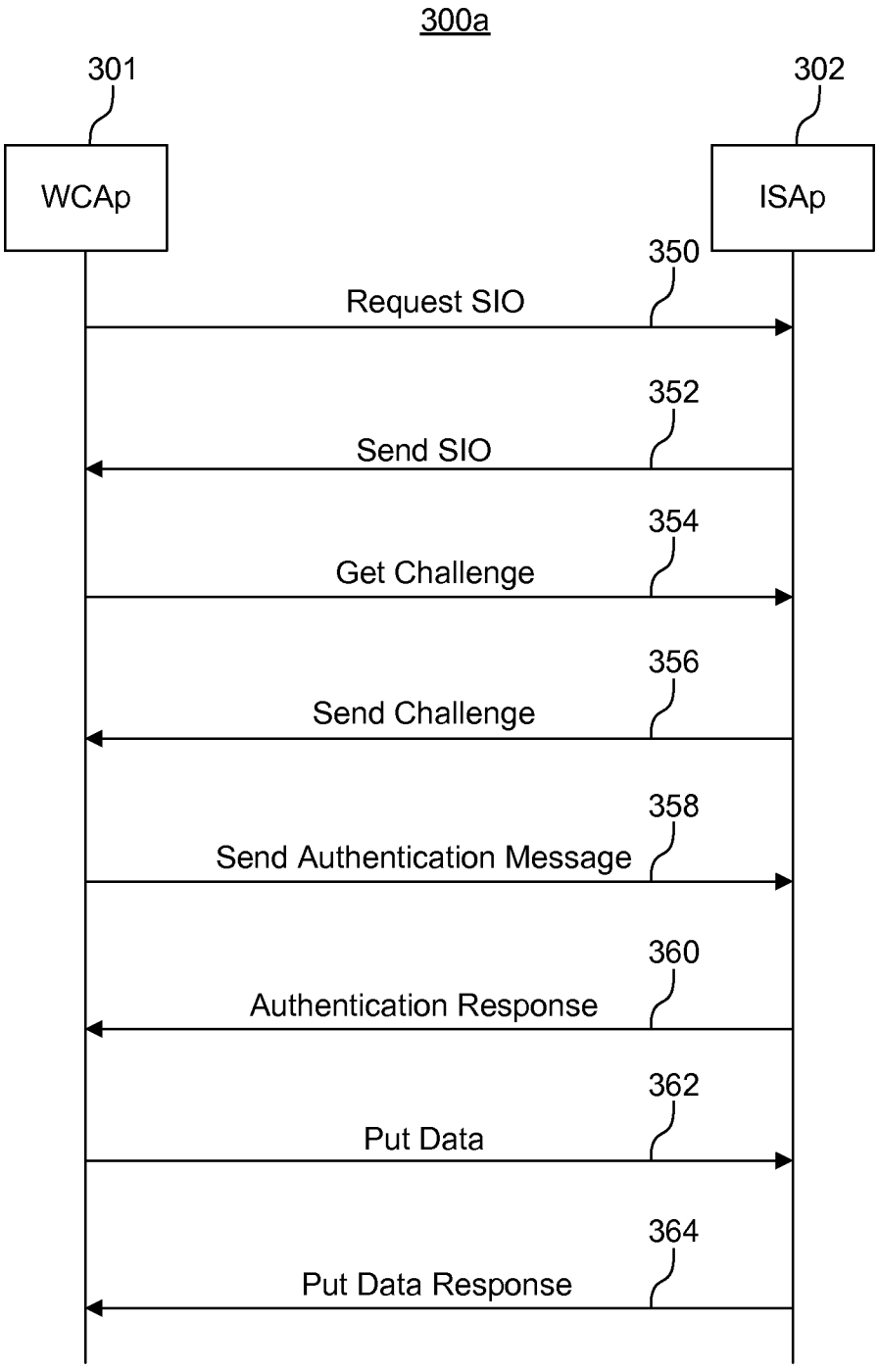
300a
301
302
WCAp
ISAp
350
Request SIO
352
Send SIO
354
Get Challenge
356
Send Challenge
358
Send Authentication Message
360
Authentication Response
362
Put Data
364
Put Data Response

FIG. 3A

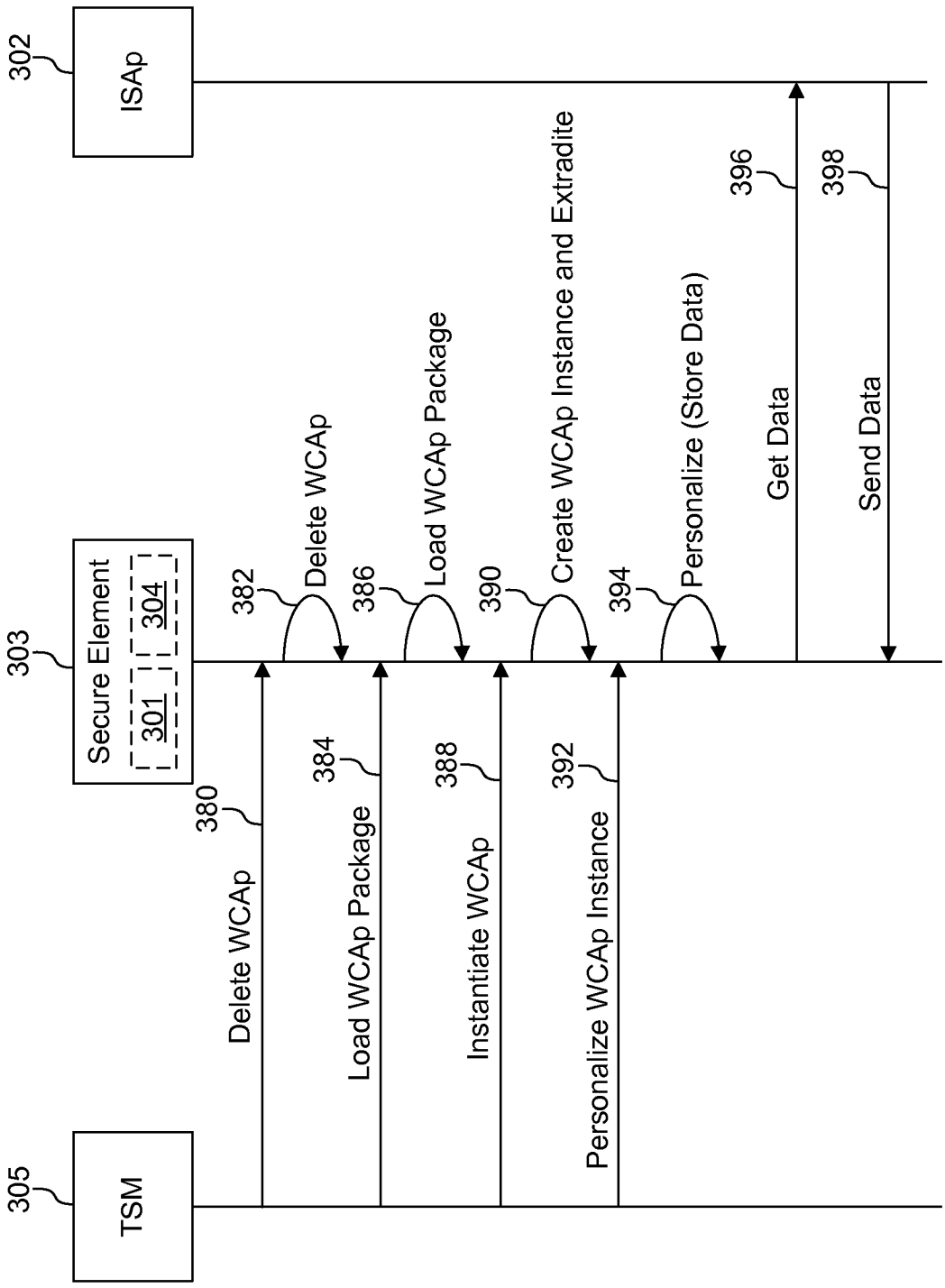
300b
302
ISAp
303
Secure Element
301
304
305
TSM
380
Delete WCAp
382
Delete WCAp
384
Load WCAp Package
386
Load WCAp Package
388
Instantiate WCAp
390
Create WCAp Instance and Extradite
392
Personalize WCAp Instance
394
Personalize (Store Data)
396
Get Data
398
Send Data

FIG. 3B

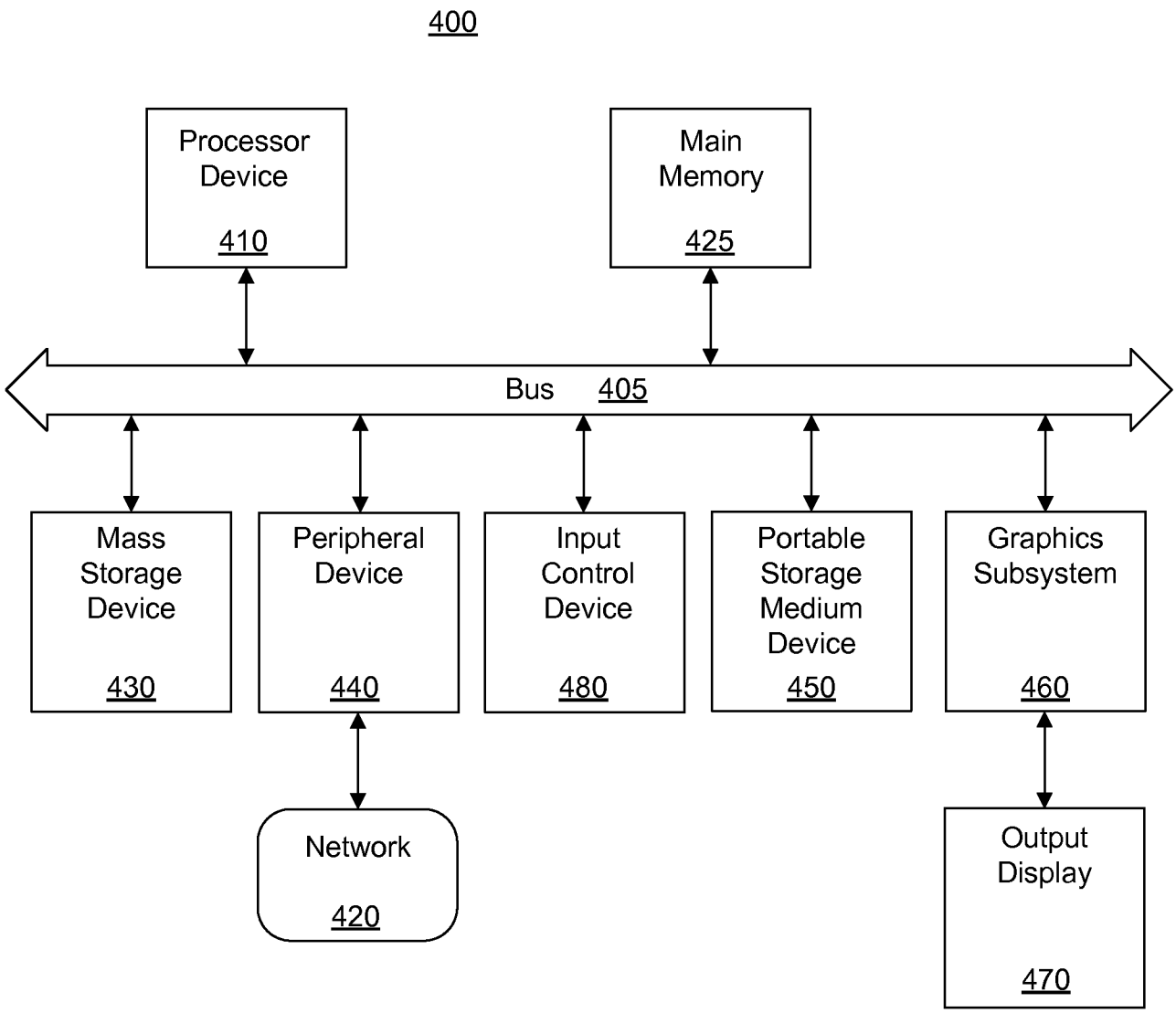
400
Processor Device 410
Main Memory 425
Bus 405
Mass Storage Device 430
Peripheral Device 440
Input Control Device 480
Portable Storage Medium Device 450
Graphics Subsystem 460
Network 420
Output Display 470

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/055983

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/50(2013.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 21/50; H04L 9/08; G06F 15/16; G06F 11/14; H04L 29/06; G06F 12/14; H04L 9/00; G06F 21/22; G06F 11/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: applet, memory, data, store, personalize

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 2003842 A1 (RESEARCH IN MOTION LIMITED) 17 December 2008 See paragraphs [0009], [0038], [0057], [0091], [0094]; claims 21, 29-30; and figures 3, 6.	1,8,15
Y		2-7,9-14,16-21
Y	US 2013-0160134 A1 (VINCENZO KAZIMIERZ MARCOVECCHIO et al.) 20 June 2013 See paragraphs [0020], [0032], [0042], [0076], [0083]; and figure 5.	2-5,7,9-12,14,16-21
Y	US 2010-0318812 A1 (RAHUL V. AURADKAR et al.) 16 December 2010 See paragraphs [0089], [0107]; and figure 6.	6,13,20
A	WO 2006-007329 A2 (MOTOROLA, INC.) 19 January 2006 See claim 1, claim 8; and figure 2.	1-21
A	US 2009-0006640 A1 (MICHAEL LAMBERTUS) 01 January 2009 See paragraphs [0036]-[0039], [0043]-[0047]; claim 1; and figures 5, 6.	1-21

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
15 December 2014 (15.12.2014)	**15 December 2014 (15.12.2014)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea
Facsimile No. +82-42-472-7140

Authorized officer
AHN, Jeong Hwan
Telephone No. +82-42-481-8440

Form PCT/ISA/210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/US2014/055983

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 2003842 A1	17/12/2008	CA 2634576 A1	15/12/2008
		CA 2634576 C	29/07/2014
		CN 101400060 A	01/04/2009
		CN 101400060 B	04/07/2012
		EP 2003842 B1	04/05/2011
US 2013-0160134 A1	20/06/2013	CA 2788051 A1	15/06/2013
		EP 2605202 A1	19/06/2013
US 2010-0318812 A1	16/12/2010	US 8321688 B2	27/11/2012
WO 2006-007329 A2	19/01/2006	CN 101006428 C	25/07/2007
		EP 1769355 A2	04/04/2007
		EP 1769355 A4	01/12/2010
		JP 2008-504592 A	14/02/2008
		US 2005-0283662 A1	22/12/2005
		WO 2006-007329 A3	26/05/2006
US 2009-0006640 A1	01/01/2009	US 2012-0260099 A1	11/10/2012
		US 8209540 B2	26/06/2012
		US 8671279 B2	11/03/2014