



(19) **United States**

(12) **Patent Application Publication**  
**Wall et al.**

(10) **Pub. No.: US 2015/0242851 A1**

(43) **Pub. Date: Aug. 27, 2015**

(54) **SECURE RESET OF PERSONAL AND SERVICE PROVIDER INFORMATION ON MOBILE DEVICES**

*H04W 12/04* (2006.01)  
*H04L 9/08* (2006.01)  
*H04L 9/32* (2006.01)

(52) **U.S. Cl.**  
CPC ..... *G06Q 20/3829* (2013.01); *H04L 9/0894* (2013.01); *H04L 9/3263* (2013.01); *H04W 12/04* (2013.01); *G06Q 20/3825* (2013.01); *G06Q 20/3278* (2013.01); *G06Q 20/3226* (2013.01)

(71) Applicant: **GOOGLE INC.**, Mountain View, CA (US)

(72) Inventors: **Jonathan Wall**, San Francisco, CA (US); **Rob von Behren**, Berkeley, CA (US)

(21) Appl. No.: **14/636,902**

(57) **ABSTRACT**

(22) Filed: **Mar. 3, 2015**

Systems and methods are described herein for supporting end users of a mobile device, such as a mobile phone, to reset a secure element associated with the communication device. The reset process may include clearing the secure element, associated memories, and storage devices of any user specific or personalized information associated with the user. The reset process may also include removing or resetting keys or other identifiers within the secure element that associate the mobile device with a particular secure service provider. According to various embodiments, a computer-implemented method for resetting a secure element within a network device may include receiving an encrypted reset request message at the secure element, decrypting the encrypted reset request message using a communication key, verifying authorization for the reset request message, and atomically clearing parameters associated with the secure element.

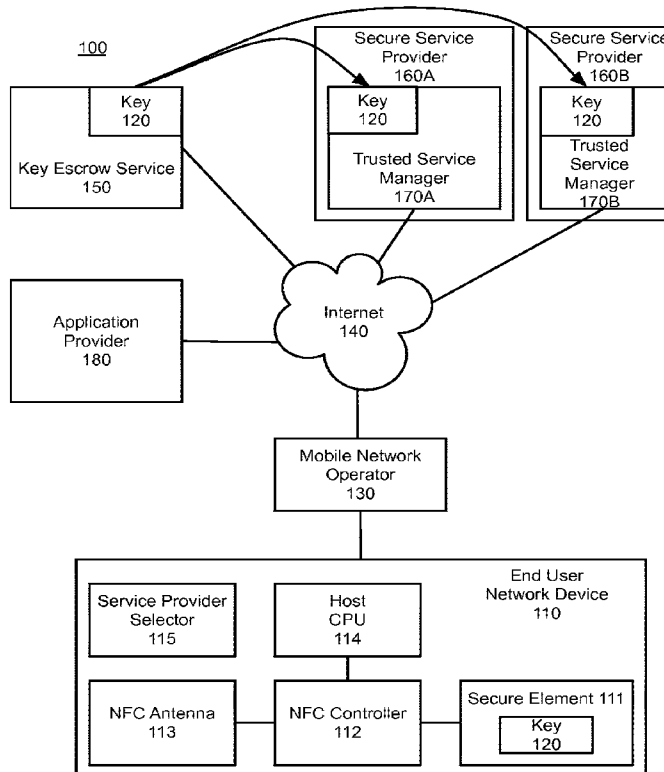
**Related U.S. Application Data**

(63) Continuation of application No. 13/868,041, filed on Apr. 22, 2013, now Pat. No. 8,971,533, which is a continuation of application No. 13/547,029, filed on Jul. 11, 2012, now Pat. No. 8,429,409.

(60) Provisional application No. 61/621,081, filed on Apr. 6, 2012.

**Publication Classification**

(51) **Int. Cl.**  
*G06Q 20/38* (2006.01)  
*G06Q 20/32* (2006.01)



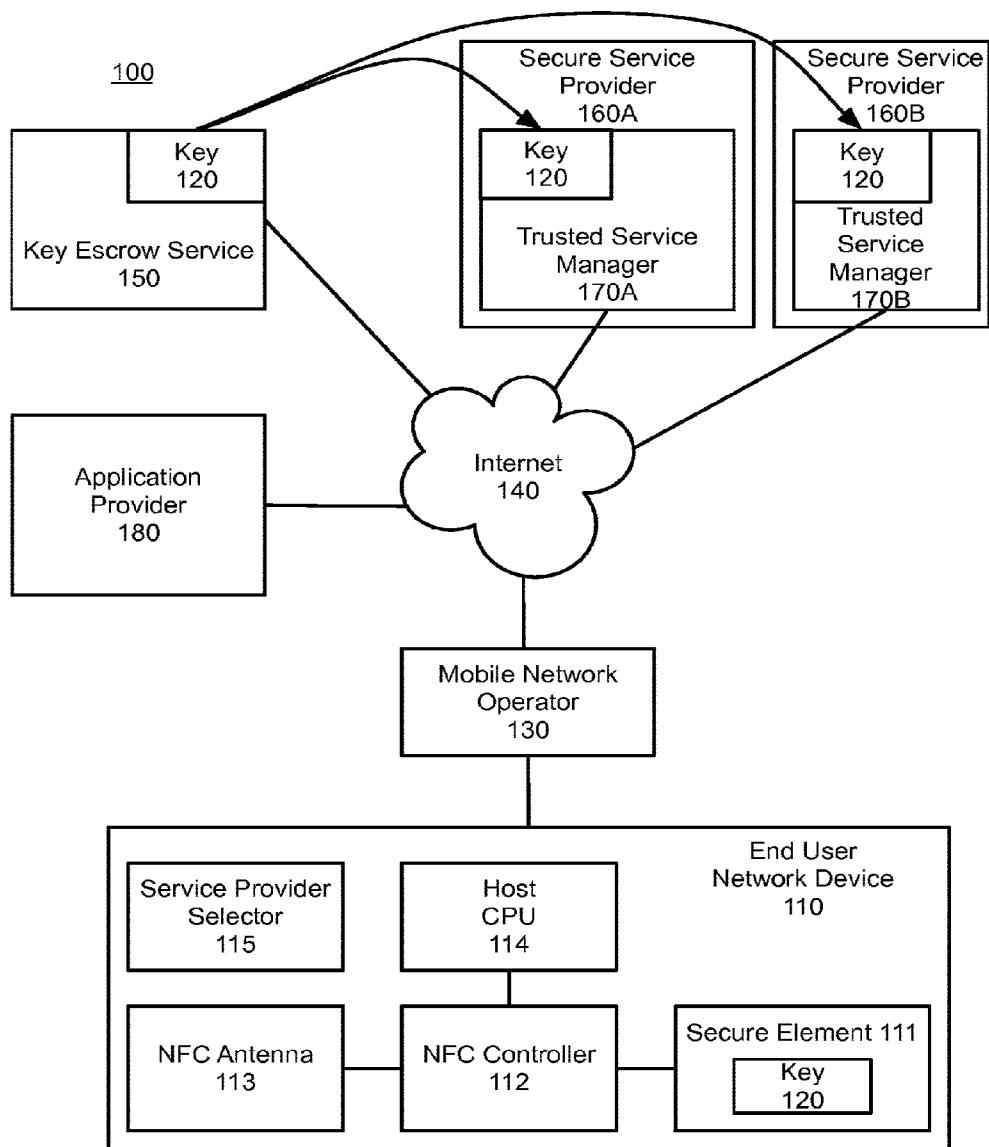


Fig. 1

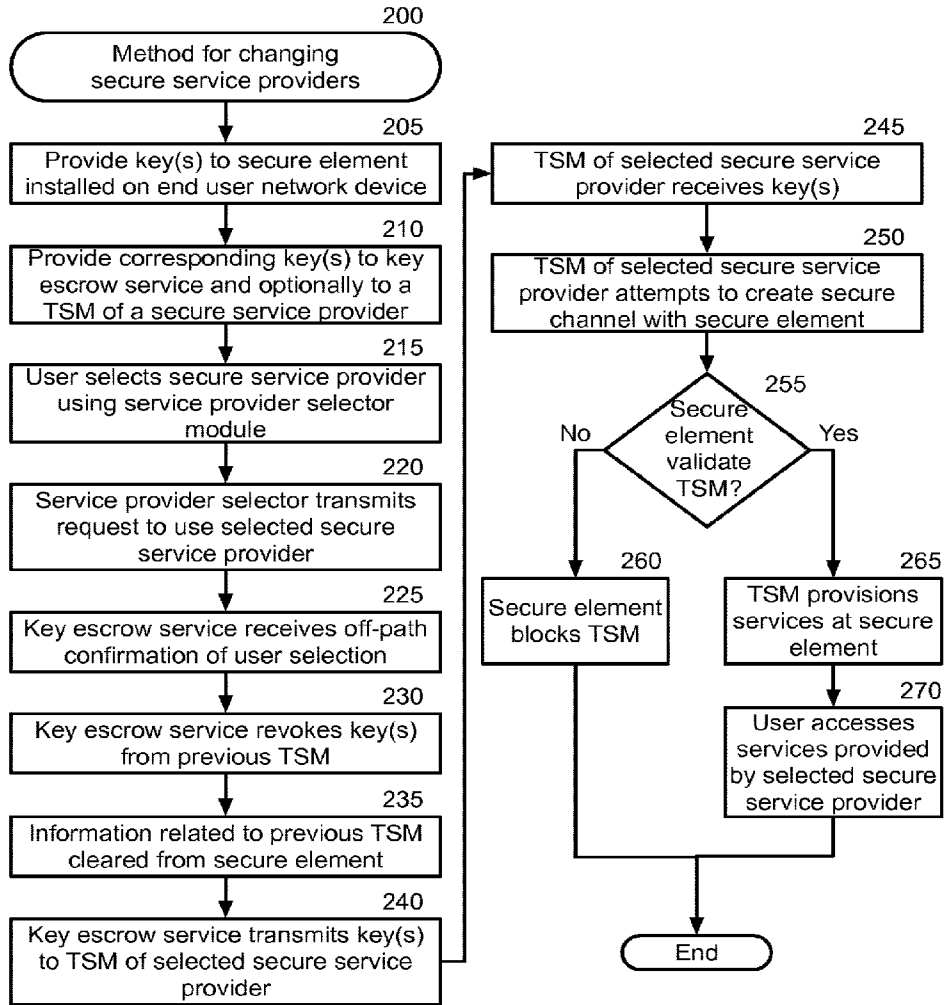


Fig. 2

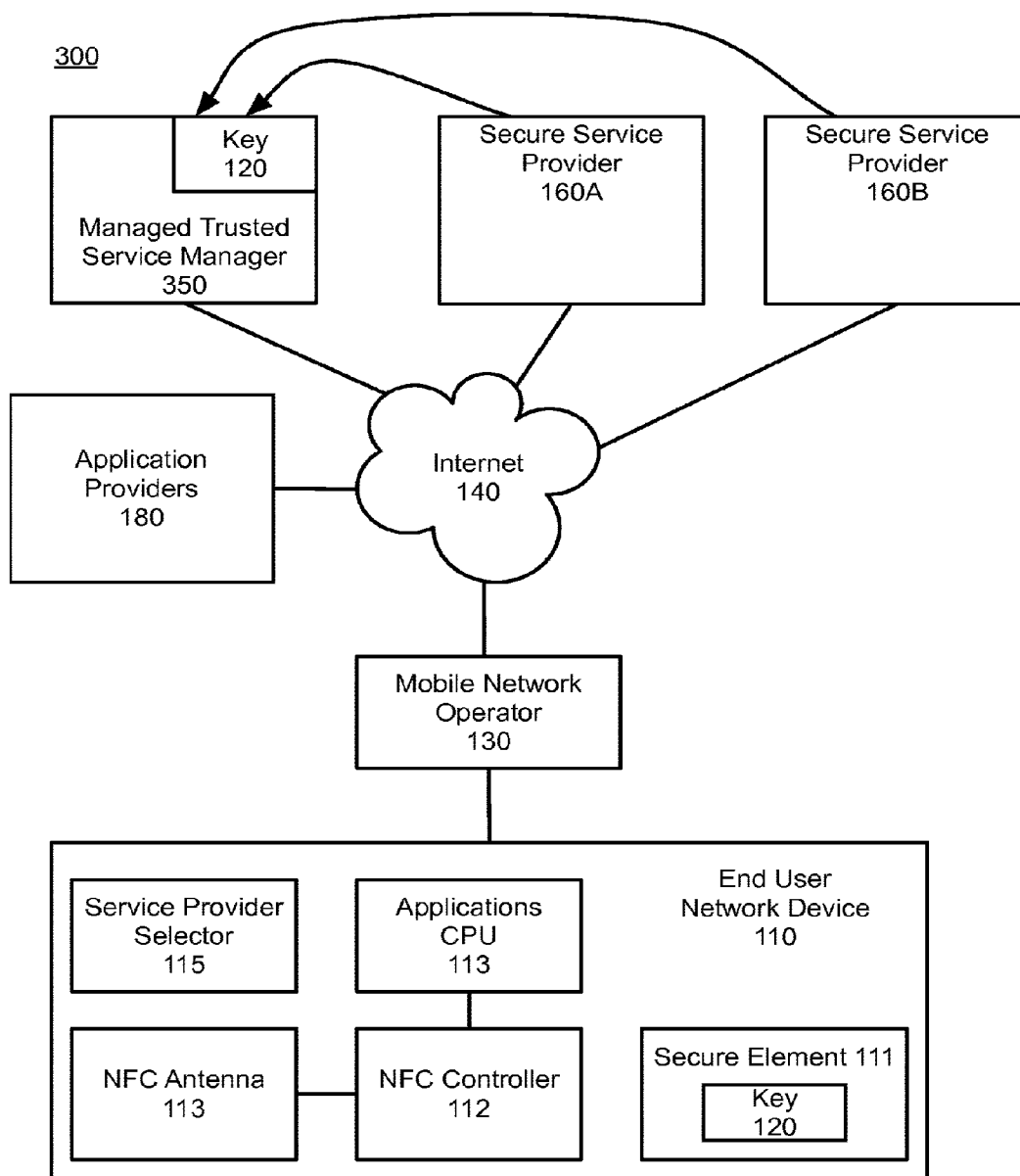


Fig. 3

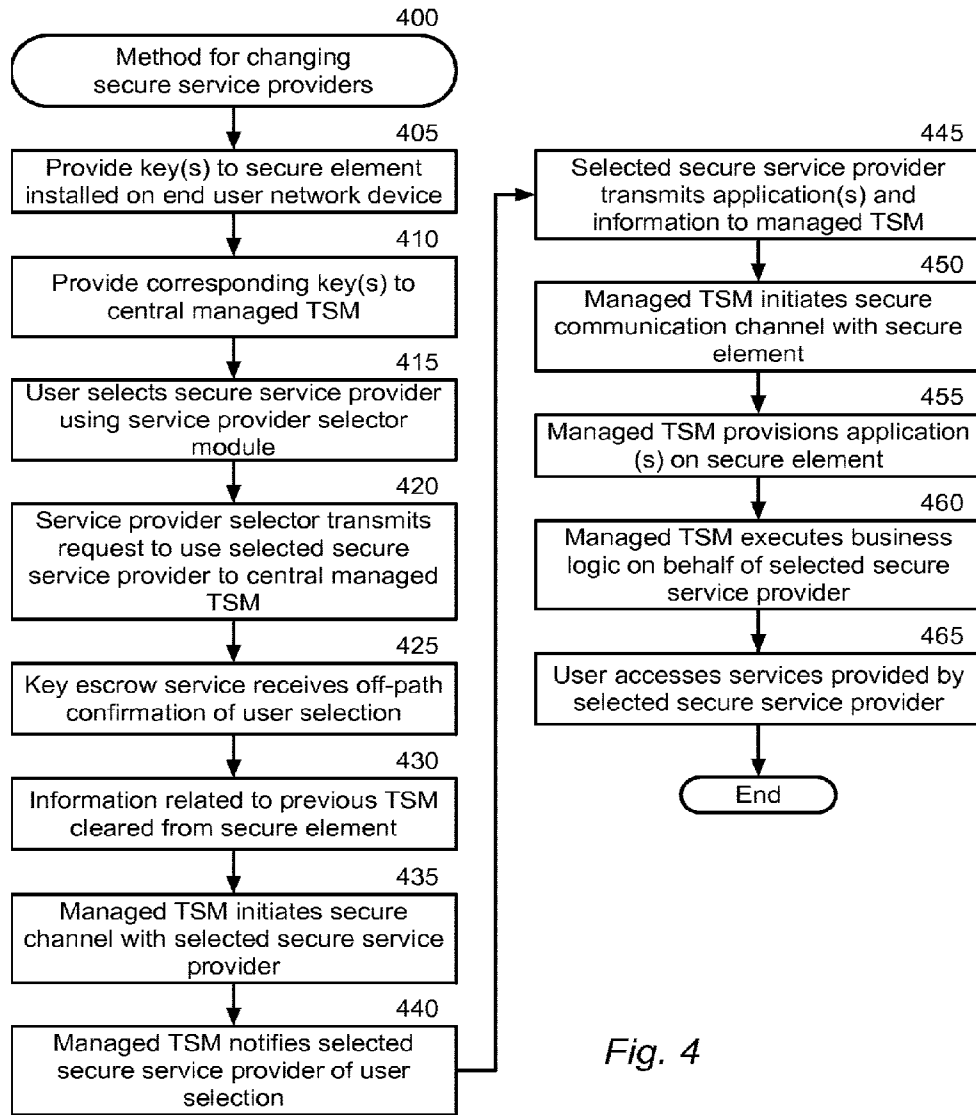


Fig. 4

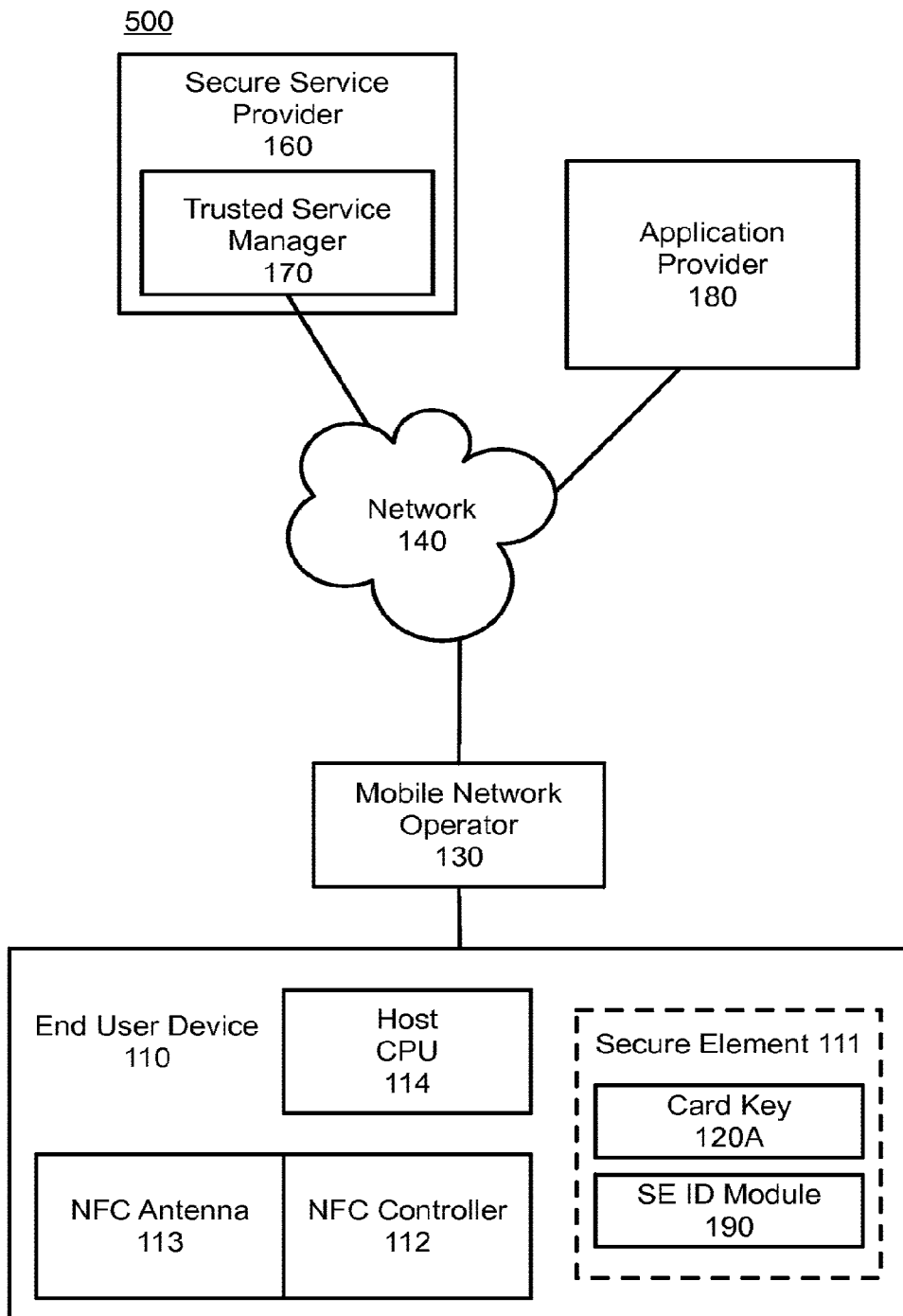


Fig. 5

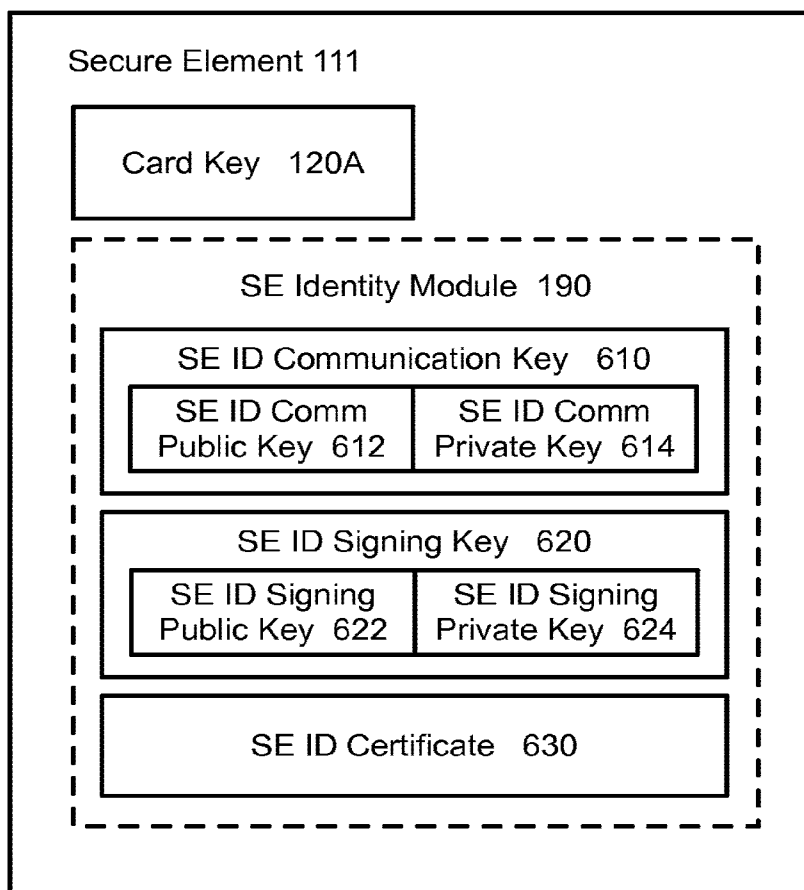


Fig. 6

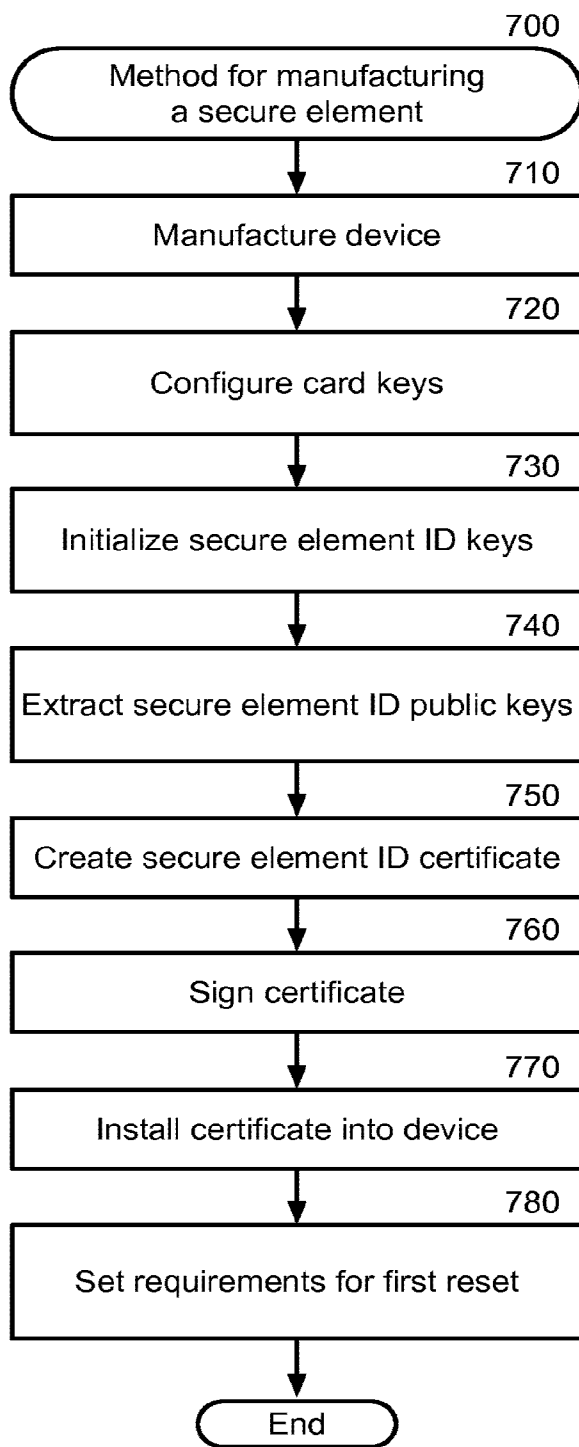


Fig. 7



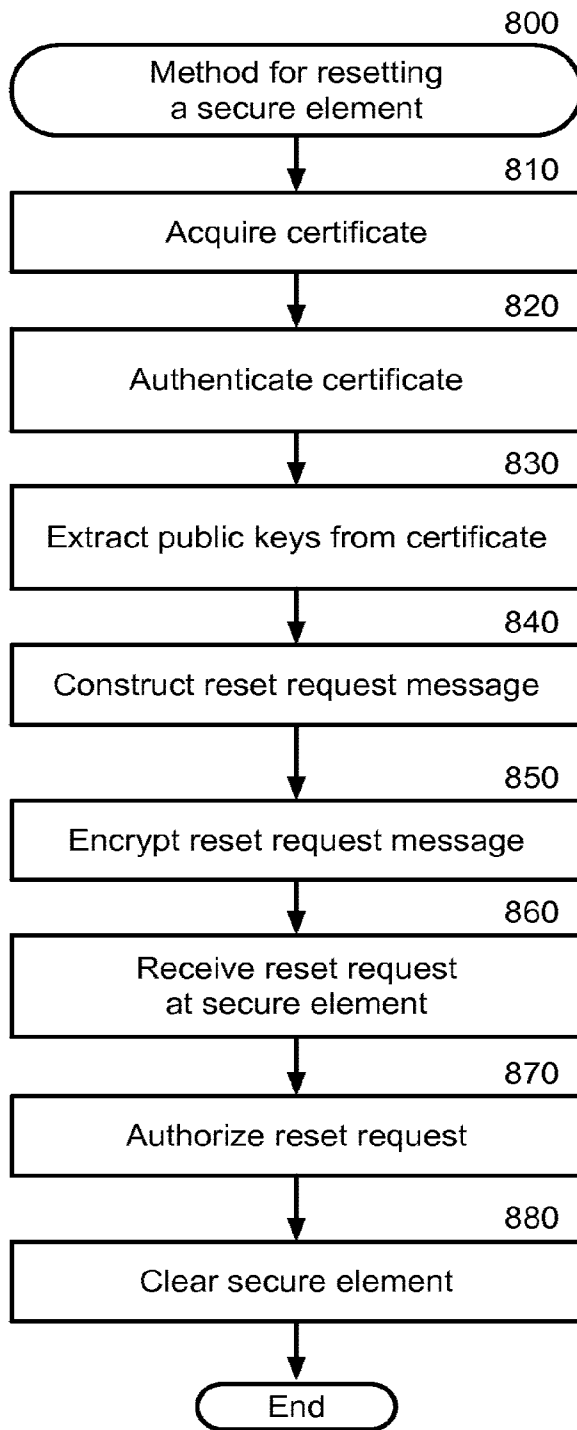


Fig. 8

**SECURE RESET OF PERSONAL AND SERVICE PROVIDER INFORMATION ON MOBILE DEVICES**

**TECHNICAL FIELD**

[0001] The present disclosure relates to systems and methods for enabling mobile device users to reset or clear the contents of secure elements associated with mobile communication devices.

**BACKGROUND**

[0002] The current Near Field Communication (“NFC”) eco-system relies on a piece of hardware commonly referred to as a “secure element” installed on communication devices to provide a secure operating environment for financial transactions, transit ticketing, physical security access, and other functions. A secure element generally includes its own operating environment with a tamper-proof microprocessor, memory, and operating system. A Trusted Service Manager (TSM), among other things, installs, provisions, and personalizes the secure element. The secure element has one or more keys that are typically installed at manufacture time. A corresponding key is shared by the TSM so that the TSM can establish a cryptographically secure channel to the secure element for installation, provisioning, and personalization of the secure element while the device having the secure element is in the possession of an end user. In this way, the secure element can remain secure even if the host CPU in the device has been compromised.

[0003] The problem with current NFC systems is that there is a tight coupling between the secure element and the TSM. For current deployments, only one TSM has access to the keys of a particular secure element. Therefore, the end user can choose to provision secure element features that are supplied by the one TSM only. The manufacturer of the device typically chooses this TSM. For example, a smart phone manufacturer may select the TSM for smart phones under guidance from a Mobile Network Operator (“MNO”), such as SPRINT or VERIZON, that purchases the smart phone rather than the end user. Thus, the TSM features available to the end user may not be in the end user’s interest. As an example, the MNO may have a business relationship with one payment provider, such as MASTERCARD or BANK of AMERICA, only. That TSM may allow the secure element to be provisioned with payment instructions from the one payment provider only. Thus, the end user would not be able to access services from other payment providers, such as VISA.

[0004] In addition to not being able to change TSM pairings to keys in the secure element, the end user is not able to clear their private data and keys from the secure element. This may be desirable when selling, transferring, returning, or exchanging devices. The end user may wish to remove their information for privacy and security as well as preparing the device to allow the new user to select their own secure services to run on the device.

**SUMMARY**

[0005] In certain exemplary embodiments described herein, methods and systems can support an end user of a mobile device, such as a mobile phone, to reset a secure element associated with the communication device. The reset process may include clearing the secure element, associated memories, and storage devices of any user specific or person-

alized information associated with the user. The reset process may also include removing or resetting keys or other identifiers within the secure element that associate the mobile device with a particular secure service provider.

[0006] According to various embodiments, a computer-implemented method for resetting a secure element within a network device may include receiving an encrypted reset request message at the secure element, decrypting the encrypted reset request message using a communication key, verifying authorization for the reset request message, and atomically clearing parameters associated with the secure element.

[0007] These and other aspects, objects, features, and advantages of the exemplary embodiments will become apparent to those having ordinary skill in the art upon consideration of the following detailed description of illustrated exemplary embodiments, which include the best mode of carrying out the invention as presently perceived.

**BRIEF DESCRIPTION OF THE DRAWINGS**

[0008] FIG. 1 depicts a Near Field Communication (“NFC”) system, in accordance with certain exemplary embodiments.

[0009] FIG. 2 is a block flow diagram depicting a method for changing secure service providers in the NFC system of FIG. 1, in accordance with certain exemplary embodiments.

[0010] FIG. 3 depicts another NFC system, in accordance with certain exemplary embodiments.

[0011] FIG. 4 is a block flow diagram depicting a method for changing secure service providers in the NFC system of FIG. 3, in accordance with certain exemplary embodiments.

[0012] FIG. 5 depicts an operating environment for a secure end user device, in accordance with certain exemplary embodiments.

[0013] FIG. 6 depicts a secure element from an end user device, in accordance with certain exemplary embodiments.

[0014] FIG. 7 is a block flow diagram depicting a method for manufacturing a secure element, in accordance with certain exemplary embodiments.

[0015] FIG. 8 is a block flow diagram depicting a method for resetting a secure element, in accordance with certain exemplary embodiments.

**DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS**

**Overview**

[0016] The methods and systems described herein enable an end user of a mobile device, such as a mobile phone, to reset a secure element associated with the communication device. The reset process may include clearing the secure element, associated memories, and storage devices of any user specific or personalized information associated with the user. The reset process may also include removing or resetting keys or other identifiers within the secure element that associate the mobile device with a particular secure service provider. In various embodiments, a reset mechanism may be provided to support securely wiping or resetting existing data from a secure device. The mechanism may incorporate a secure identification technique to verify appropriate authority to reset the device. Such a process may be used to leave the device in a cleared state or to install new keys, new identities, or new provider associations.

**[0017]** Technology to securely reset or clear a secure element associated with a communication device as described herein can support an end user of a communication device, such as a mobile phone, to select or change a secure service provider to use with a secure element stored on the communication device. In one embodiment, a system includes a key escrow service that manages cryptographic keys for one or more users and one or more secure service providers. Typically, the secure element and one or more cryptographic keys for the secure element are installed on each user communication device at the time that the communication devices are manufactured. These keys or corresponding keys are provided to the key escrow service. Each user device also includes a service provider selector (“SPS”) module or software application that enables the users to select from available secure service providers. The SPS transmits, via a secure channel, information identifying the selected service provider to the key escrow service in response to a user selection. The key escrow service provides the key for the user’s secure element to a Trusted Service Manager (“TSM”) of the selected secure service provider. The key escrow service also revokes the key for the user’s secure element from the TSM of the user’s previous secure service provider. In addition, the SPS can prevent unauthorized secure service providers, such as the previous secure service provider, from accessing the secure element.

**[0018]** In another embodiment, a central TSM performs business logic and application provisioning on behalf of other secure service providers. Rather than distributing the cryptographic keys to selected secure service providers, the central TSM acts as a proxy between the selected secure service provider and the secure element installed on the communication device.

**[0019]** The exemplary systems and methods described herein overcome the deficiencies of conventional NFC systems to allow users to reset or wipe secure identifiers and associations. This ability can support reassignment of secure devices as well as simplified access to services of more than one secure service provider. Rather than being limited to the functionality and services provided by the one secure service provider, the user can dissociate from a current provider and select from multiple secure service providers. For example, if a secure service provider does not provide services that the user desires, such as making payments via a particular brand of credit card, the user can select a secure service provider that does provide these services.

**[0020]** One or more aspects of the exemplary embodiments may include a computer program that embodies the functions described and illustrated herein, wherein the computer program is implemented in a computer system that comprises instructions stored in a machine-readable medium and a processor that executes the instructions. However, it should be apparent that there could be many different ways of implementing the exemplary embodiments in computer programming, and the exemplary embodiments should not be construed as limited to any one set of computer program instructions. Further, a skilled programmer would be able to write such a computer program to implement an embodiment based on the appended flow charts and associated description in the application text. Therefore, disclosure of a particular set of program code instructions is not considered necessary for an adequate understanding of how to make and use the exemplary embodiments. Moreover, any reference to an act being performed by a computer should not be construed as being

performed by a single computer as the act may be performed by more than one computer. The functionality of the exemplary embodiments will be explained in more detail in the following description, read in conjunction with the figures illustrating the program flow.

**[0021]** Turning now to the drawings, in which like numerals indicate like (but not necessarily identical) elements throughout the figures, exemplary embodiments are described in detail.

#### System Architecture

**[0022]** FIG. 1 depicts a Near Field Communication (“NFC”) system 100, in accordance with certain exemplary embodiments. As depicted in FIG. 1, the system 100 includes one or more end user network devices 110, one or more application providers 180, a key escrow service 150, a mobile network operator (“MNO”) 130, and multiple secure service providers 160. Each of the application providers 180, key escrow service 150, and secure service providers 160 include a network device configured to communicate via the Internet 140. For example, each of the application providers 180, key escrow service 150, and secure service providers 160 may include a server, desktop computer, laptop computer, tablet computer, smartphone, handheld computer, personal digital assistant (“PDA”), or any other wired or wireless, processor-driven device. In one embodiment, the key escrow service 150 includes (or is communicably coupled to) a first network communication module for receiving requests to change (or select) from available secure service providers 160 and a second network communication module for transmitting cryptographic keys 120 to secure service providers 160. The first and second network communication modules may be the same or different network communication modules.

**[0023]** The end user network devices 110 may be mobile phones, smart phones, PDAs, netbook computers, laptop computers, tablet computers, or any other wired or wireless, processor-driven device. As shown in FIG. 1, the end user network devices 110 access the Internet 140 via the MNO 130. Exemplary MNOs include VERIZON, SPRINT, and AT&T. The MNOs provide Internet access to the end user network devices 110 via a mobile network (not shown), such as a 3G or 4G mobile communication network. Of course, the end user network devices 110 can access the Internet 140 via other mechanisms, such as Wi-Fi in connection with an Internet provider.

**[0024]** The end user network devices 110 each include a secure element 111 having one or more cryptographic keys 120, an NFC controller 112, an NFC antenna 113, a host CPU 114, and an SPS 115. The NFC controller 112 and the NFC antenna 113 enable the end user network device 110 to communicate with other NFC-enabled devices (not shown). For example, the end user network devices 110 can communicate with NFC-enabled merchant point of sale (“POS”) devices, ticketing devices, security devices, and other end user network devices 110.

**[0025]** The host CPU 114 executes applications stored on the end user network device 110. For example, the host CPU 114 may execute applications that interact with the NFC controller 112, such as NFC payment applications that enable the user operating the end user network device 110 to complete purchases via an NFC-enabled POS or a transit or event ticketing application that enables the user to enter a transit facility or event via an NFC-enabled ticketing POS. Other applications, including identification, authentication, secu-

ity, and coupon clipping and redemption applications, also may be stored on the end user network device **110** for execution by the host CPU **114** in connection with the NFC controller **112** and the NFC antenna **113**.

[0026] Each of the applications may be provided by a respective application provider **180**. For example, a credit card company may provide a credit card payment application; a transit or other ticketing company may provide a ticket purchasing and redemption application; a manufacturer, retailer, or other entity that sells products or services may provide a coupon application; and an authentication company may provide a user authentication application.

[0027] NFC applications are typically stored in the secure element **111** of the end user network device **110** for security purposes. The secure element **111** provides a secure operating environment for the NFC (or other) applications. The secure element **111** typically includes its own operating environment with tamper-proof microprocessor, operating system, and memory for storing information, such as payment credentials. The secure element **111** may exist within a fixed chip of the end user network device **110**, a Subscriber Identification Module (“SIM”) card, a Universal Integrated Circuit Card (“UICC”), a removable smart chip, or in a memory card, such as a microSD card. The secure element **111** also may include a memory controller for managing Read Only Memory (“ROM”), Ready Access Memory (“RAM”), and EEPROM flash memory of the card or chip in which the secure element **111** is installed.

[0028] In general, the secure service providers **160** serve as intermediaries that assist application providers **180** and other service providers in securely distributing and managing applications and services, such as NFC contactless applications services. A TSM **170** of the secure service provider **160** typically hosts the applications and installs and provisions the applications onto the secure element **111**. As shown in FIG. 1, each TSM **170** can receive, store, and utilize the keys **120** for users’ secure elements **111**. By having the keys **120**, the TSM **170** can access the secure elements **111** via a secure encrypted communication channel to install, provision, and customize applications within the secure elements **111**. Exemplary secure service providers **160** include GEMALTO and FIRST DATA.

[0029] In certain exemplary embodiments, the secure service providers **160** bypass the host CPU **114** and the NFC controller **112** when communicating with the secure element **111**. For example, in certain UICC/SIM secure elements, the secure service providers **160** communicate with the secure element **111** via a radio CPU (not shown) installed on the end user network device **110**. Thus, the involvement of the NFC controller **112** and the host CPU **114** may be optional during the provisioning of applications on the secure element **111** in certain exemplary embodiments. In certain exemplary embodiments, the host CPU **114** and the radio CPU interact with one another to coordinate access controls to the secure element **111**.

[0030] The key escrow service **150** maintains the keys **120** for the secure elements **111**. The key escrow service **150** also distributes the keys to the TSMs **170**, for example in response to a user selection. For instance, if a user elects to switch from a first secure service provider **160A** to a second secure service provider **160B**, the key escrow service **150** revokes the keys **120** from the first TSM **170A** and provides the keys **120** to the second TSM **170B**. The second TSM **170** can then access the secure element **111** of the user’s network device **110**.

[0031] The SPS **115** is implemented in software and/or hardware and enables the user of the end user network device **110** to select or change secure service providers **160** via the key escrow service **150**. The SPS **115** provides a user interface that allows the user to make a selection of a secure service provider **160**. In response to a user selection, the SPS **115** transmits information regarding the selected secure service provider **160** to the key escrow service **150**. The key escrow service **150** also can confirm the selection via one or more off-path mechanisms. The SPS **115**, key escrow service **150**, and other components of the exemplary system **100** are described in more detail hereinafter with reference to the method depicted in FIG. 2.

[0032] FIG. 3 depicts another NFC system **300**, in accordance with certain alternative exemplary embodiments. The exemplary system **300** includes many of the same components as the system **100**, including one or more end user network devices **110**, one or more application providers **180**, an MNO **130**, and multiple secure service providers **160**. However, rather than a key escrow service **150**, the system **300** includes a central managed TSM **350**. The managed TSM **350** includes a network device configured to communicate with the Internet **140**, such as a server, desktop computer, laptop computer, tablet computer, smartphone, handheld computer, PDA, or other wired or wireless, processor-driven device. Similar to the key escrow service **150**, the managed TSM **350** maintains the keys **120** for the secure elements **111** and enables the users operating the end user network devices **110** to select from multiple secure service providers **160**. Rather than distributing the keys **120** to the selected TSMs **170**, the managed TSM **350** can interact with the secure elements **111** on behalf of the selected secure service provider **160**. That is, the managed TSM **350** can install, provision, and interact with applications installed on the secure elements **111**. Or, the managed TSM **170** can establish (and terminate) a secure communication channel between the selected TSM **170** and the secure element **111** such that the selected TSM **170** can interact with the secure element **111**. This secure communication channel may be encrypted with a different key that is not associated with the secure element **111**, and may be specific to each secure service provider **160**. The managed TSM **350** also can perform business logic on behalf of the secure service providers **160**. The managed TSM **350** and other components of FIG. 3 are described in more detail hereinafter with reference to the method depicted in FIG. 4.

[0033] FIG. 5 depicts an operating environment **500** for a secure end user device **110**, in accordance with certain exemplary embodiments. As depicted in FIG. 5, the operating environment **500** includes one or more end user devices **110**, one or more application providers **180**, a mobile network operator (“MNO”) **130**, and one or more secure service providers **160**. Each of the application providers **180** and secure service providers **160** include a network device configured to communicate via the network **140**. For example, each of the application providers **180** and secure service providers **160** may include a server, desktop computer, laptop computer, tablet computer, smartphone, handheld computer, personal digital assistant (“PDA”), or any other wired or wireless, processor-driven device.

[0034] The end user devices **110** may be mobile phones, smart phones, PDAs, netbook computers, laptop computers, tablet computers, or any other wired or wireless, processor-driven devices. End user devices **110** can access the network **140** via the MNO **130**. Exemplary MNOs **130** include VERI-

ZON, SPRINT, and AT&T. These MNOs **130** may provide access to the network **140** for the end user devices **110**. The network **140** may be a wired, wireless, optical, radio frequency, mobile, cellular, any other data or voice network, or any combination thereof. For example, the network **140** may be an internet or the Internet. The MNO **130** may provide connectivity for part of the network **140** as a 3G or 4G mobile communication network. Of course, the end user network devices **110** can access the Internet **140** via other mechanisms, such as Wi-Fi or Wi-Max or in connection with an Internet provider.

[0035] The end user devices **110** each include a secure element **111** having a secure element (“SE”) identity (“ID”) module **190** and one or more card keys **120A**. The end user devices **110** may also each include an NFC controller **112**, an NFC antenna **113**, and a host CPU **114**. The NFC controller **112** and the NFC antenna **113** enable the end user device **110** to communicate with other NFC-enabled devices (not shown). For example, the end user devices **110** can communicate with NFC-enabled merchant point of sale (“POS”) devices, ticketing devices, security devices, and other end user devices **110**.

[0036] The card keys **120A** may include any keys used to secure applications or processes of the secure element **111**. According to various embodiments, the card keys **120A** may include card manager keys, fare keys, financial transaction keys, banking keys, or so forth. The card keys **120A** may be associated with, shared with, or split with the secure service provider **160**, trusted service manager **170**, application provider **180**, or any other entity or system securely interfacing with the end user device **110** and its secure element **111**. The card keys **120A** may operate in conjunction with, or be protected by, other keys associated with the SE ID module **190** as discussed in further detail below.

[0037] The host CPU **114** can execute applications stored on the end user network device **110**. For example, the host CPU **114** may execute applications that interact with the NFC controller **112**, such as NFC payment applications that enable the user operating the end user device **110** to complete purchases via an NFC-enabled POS or a transit or event ticketing application that enables the user to enter a transit facility or event via an NFC-enabled ticketing POS. Other applications, including identification, authentication, security, and coupon clipping and redemption applications, also may be stored on the end user device **110** for execution by the host CPU **114** in connection with the NFC controller **112** and the NFC antenna **113**.

[0038] Applications associated with an end user device **110** may also be associated with an application provider **180**. For example, a credit card company may provide a credit card payment application; a transit or other ticketing company may provide a ticket purchasing and redemption application; a manufacturer, retailer, or other entity that sells products or services may provide a coupon application; and an authentication company may provide a user authentication application.

[0039] NFC applications are typically stored in the secure element **111** of the end user device **110** for security purposes. The secure element **111** provides a secure operating environment for the NFC (or other) applications. The secure element **111** typically includes its own operating environment with tamper-proof microprocessor, operating system, and memory for storing information, such as payment credentials. The secure element **111** may exist within a fixed chip of the end

user device **110**, a Subscriber Identification Module (“SIM”) card, a Universal Integrated Circuit Card (“UICC”), a removable smart chip, or in a memory card, such as a microSD card. The secure element **111** also may include a memory controller for managing Read Only Memory (“ROM”), Ready Access Memory (“RAM”), and EEPROM flash memory of the card or chip in which the secure element **111** is installed.

[0040] In general, the secure service providers **160** serve as intermediaries that assist application providers **180** and other service providers in securely distributing and managing applications and services, such as NFC contactless applications services. A TSM **170** of the secure service provider **160** typically hosts the applications and installs and provisions the applications onto the secure element **111**. Each TSM **170** can receive, store, and utilize the card keys **120A** for associated secure elements **111**. The TSM **170** can access the secure element **111** using the card keys **120A** for that secure element. Exemplary secure service providers **160** include GEMALTO and FIRST DATA.

[0041] In certain exemplary embodiments, the secure service providers **160** bypass the host CPU **114** and the NFC controller **112** when communicating with the secure element **111**. For example, in certain UICC/SIM secure elements, the secure service providers **160** communicate with the secure element **111** via a radio CPU (not shown) installed on the end user device **110**. Thus, the involvement of the NFC controller **112** and the host CPU **114** may be optional during the provisioning of applications on the secure element **111** in certain exemplary embodiments. In certain exemplary embodiments, the host CPU **114** and the radio CPU interact with one another to coordinate access controls to the secure element **111**.

[0042] The user of the end user device **110** may wish to delete or reset security features associated with the secure element **111** such as card keys **120A**. For example, the user may wish to dissociate the end user device **110** from the secure service provider **160** so that they can securely transfer the end user device **110** to another user. Similarly, the user may wish to dissociate the end user device **110** from one secure service provider **160** to allow associating with another provider. For example, if the user wishes to change the financial institution associated with NFC payments or fund transfers made with the end user device **110**. Talk about reset/deletion of keys. The SE ID Module **190** can support a secure deletion or reset process for the secure element **111** as discussed in further detail below.

[0043] FIG. 6 depicts a secure element **111** from an end user device **110**, in accordance with certain exemplary embodiments. As depicted in FIG. 6, the secure element **111** can include a card key **120A** and a secure element identity module **190**. The secure element identity module **190** can include a secure element identity communication key **610**, a secure element identity signing key **620**, and a secure element identity certificate **630**. The secure element identity module **190** may be used to verify that an application or service is interacting with a particular secure element **111** and also secure messages sent to and from the secure element **111**. For example, element identity communication key **610** may be used to encrypt communications with the secure element **111** so that those communications may only be transacted by the intended secure element **111**. Similarly, the secure element identity signing key **620** may be used verify which secure element **111** is being communicated with. These mechanisms may be used, as discussed below to securely reset the secure element **111** such as clearing or resetting card keys **120A**. The

secure element identity certificate **630** may be used to certify exchange and transfer of the various keys such as the secure element identity communication key **610** and the secure element identity signing key **620**.

[0044] The secure element identity communication key **610** and the secure element identity signing key **620** may operate in a key split or public/private key pair configurations. For example, the secure element identity communication key **610** may comprise a public key **612** and a private key **614**. Similarly, the secure element identity signing key **620** may comprise a public key **622** and a private key **624**.

[0045] The secure element identity module **190** may provide secure identification functionality. Such functionality may include features provided by, or in association with, the secure element **111** in order to support secure element **111** reset procedures as discussed in further detail below. One such feature involves supporting secure communications and secure authentication with the secure element identity module **190** from a secure application, application provider **180**, or a secure service provider **160**. Another feature involves supporting a reset operation in association with the secure element identity module **190**. The reset operation may atomically clear the secure element **111**. This may include clearing or resetting one or more card keys **120A** from the secure element **111**. The atomic nature of the reset operation may provide a reset that is all or nothing in nature and cannot be operationally subdivided or partially carried out. Another feature may include a mechanism for verifying that the reset operation originates from a trusted source. For example, the verification may ensure that the reset is actually being done on behalf of the owner of the end user device **110** or some other approved individual, provider, or application. These features of the secure element identity module **190** may also be associated with an operating system, a smart-card operating system, or an applet such as a card manager applet.

#### System Process

[0046] FIG. 2 is a block flow diagram depicting a method **200** for changing secure service providers in the NFC system **100** of FIG. 1. The method **200** is described with reference to the components illustrated in FIG. 1.

[0047] In block **205**, one or more secure cryptographic keys **120** are provided for a secure element **111**. In certain exemplary embodiments, the secure element **111** and its keys **120** are installed on an end user network device **110** at manufacture time. In certain exemplary embodiments, the secure element **111** and its keys **120** are installed on a removable card or chip, such as a SIM card or microSD card, that is later installed on the end user network device **110**.

[0048] In block **210**, the keys **120** for the secure element **111** or corresponding keys are provided to the key escrow service **150**. These keys **120** enable the key escrow service **150** (or another entity that receives the keys **120**) to create a secure communication channel with, and gain access to, the secure element **111**. Optionally, the keys **120** also are provided to a TSM **170** of a secure service provider **160**. Conventionally, the secure service provider **160** and the TSM **170** for the secure element **111** are selected by the manufacturer of the end user network device **110**, typically under guidance from the MNO **130** that purchases the end user network device **110**. In this case, the keys **120** may be provided to that TSM **170**. Alternatively, the keys **120** are provided to the key escrow service **150** only. In this case, the user operating the end user network device **110** (or another entity, such as the

MNO **130**) can make an initial selection of secure service providers **160** using the SPS **115**.

[0049] In block **215**, the user selects a secure service provider **160** and thus, a TSM **170**, using the SPS **115**. For example, the user may access the SPS **115** using the end user network device **110**. The SPS **115** may present a user interface that lists available secure service providers **160** and optionally the services supported by the secure service providers **160**. For example, the SPS **115** may display financial institutions for which contactless transactions are supported by each secure service provider **160**. In another example, the SPS **115** may display applications provisioned and supported by each available secure service provider **160**. In yet another example, the SPS **115** may provide a search function that enables users to search secure service providers **160** based on their features and services. When the user finds an appropriate secure service provider **160**, the user can select that secure service provider **160** using the SPS **115**.

[0050] In block **220**, the SPS **115** transmits a request to use the selected service provider **160** to the key escrow service **150** in response to the user selection. The request typically includes information identifying the selected secure service provider **160**. In response to receiving the request, the key escrow service **150** processes the request.

[0051] In block **225**, the key escrow service **150** performs an off-path confirmation procedure to confirm that the user initiated the request to use the selected secure service provider **160**. This block **225** is optional and provides an additional level of security for the SPS **115**/key escrow service **150** system for example to prevent another person from accessing this feature in the event that the end user network device **110** is lost or stolen.

[0052] In one embodiment, the off-path confirmation procedure includes the key escrow service **150** communicating to the user that the request was made via a different communication channel than through the end user network device **110**. For example, the key escrow service **150** may transmit an SMS text message to a mobile phone of the user that indicates that the request was made. Or, key escrow service **150** may make a telephone call to the user with a message that the request was made. The text message or voice message may instruct the user to call a certain telephone number if the user did not make the request. The key escrow service **150** also may require that the user confirm the request. For example, the text message may instruct the user to respond to the text message, access a web site of the key escrow service **150**, or call the key escrow service **150** to confirm the request. Also, a code may be provided in the message to the user and the user may be required to enter the code via phone or via the web site to confirm the request.

[0053] In block **230**, if another TSM **170** possessed the keys **120** for the secure element **111**, the key escrow service **150** revokes the keys **120** from that previous TSM **170**. In one embodiment, the key escrow service **150** sends a message, for example an SMS text message, to the previous TSM **170** requesting that the TSM discard the keys **120**. The secure service providers **160** may be obligated under contract to discard the keys **120** in response to such a request.

[0054] In another embodiment, the key escrow service **150** revokes the keys **120** from the previous TSM **170** by instructing the secure element **111** to block the previous TSM **170**. The secure element **111** can include program code that identifies TSMs **170** attempting to access the secure element **111** and a list of allowed and/or blocked TSMs **170**. When a TSM

**170** attempts to access the secure element **111**, the secure element **111** can compare information identifying that TSM **170** to the list(s) to determine whether to grant access. The key escrow service **150** also can send a request to the previous TSM **170** requesting that the previous TSM discard the keys **120**. Of course, the blocked TSM **170** can be unblocked in the event that the user reselects the secure service provider **160** for that TSM **160**. For example, the key escrow service **150** may send a message to the secure element **111** requesting that the secure element **110** unblock the TSM **170**.

[**0055**] In yet another embodiment, the key escrow service **150** revokes the keys **120** from the previous TSM **170** via the use of a master key and TSM specific keys. A TSM specific key may be provided to the secure element **111** for each available TSM or for a selected TSM **170**. The TSM specific keys also are distributed to the respective TSMs **170**. The TSM specific keys may be preloaded onto the secure element **111** at manufacture time, installed at a later date by the key escrow service **150**, or installed by the key escrow service **150** in response to the user selecting a TSM **170**. The secure element **111** can control which of the TSM specific keys are active and which TSM specific keys are inactive. For example, if a user requests to switch from secure service provider **160A** to secure service provider **160B**, the SPS **115** communicates this request (and information identifying the selected TSM **170B**) to a key management applet or module (not shown) of the secure element **111**. The key management applet activates the TSM specific key for the TSM **170B** and deactivates the TSM specific key for the TSM **170A** in response to the request. At this point, the secure element **111** allows access to the TSM **170B** while blocking access from the TSM **170A**.

[**0056**] In block **235**, information stored on the secure element **111** related to the previous TSM **170** and/or previous secure service provider **160** is removed from the secure element **111**. For example, payment card credentials associated with the previous TSM **170** may be stored on the secure element **111** while that TSM **170** is being used in conjunction with the secure element **111**. These credentials are removed from the secure element **111** prior to enabling another TSM **170** access to the secure element **111**. In addition, any applications installed on the secure element **111** for the previous TSM **170** are uninstalled. In certain exemplary embodiments, the key escrow service **150** sends a command to an applet or module of the secure element **111**, such as a card manager applet, to remove the information related to the previous TSM **170**.

[**0057**] In block **240**, the key escrow service **150** transmits the keys **120** to the TSM **170** of the selected secure service provider **160**. This transmission is typically made via a secure communication channel. For example, the key escrow service **150** may send the keys **120** to the selected TSM **170** via an encrypted communication channel. In block **245**, the selected TSM **170** receives the keys **120**.

[**0058**] In certain exemplary embodiments, the key escrow service **150** delays transmitting the keys **120** to the TSM **170** of the selected secure service provider **160** until receiving confirmation that the information and applications related to the previous TSM **170** are removed from the secure element **111**. In some embodiments, the key escrow service **150** may not transmit the keys **120** to the TSM **170** of the selected secure service provider **160** without receiving off-path confirmation from the user that the user requested to use the selected secure service provider **160**.

[**0059**] In block **250**, the TSM **170** of the selected secure service provider **160** attempts to create a secure communication channel with the secure element **111** using the received keys **120**. In one embodiment, the TSM **170** sends an encrypted message to the secure element **111** requesting access to the secure element **111**. The TSM **170** encrypts the message by performing a cryptographic algorithm on the message using the received keys **120**.

[**0060**] In block **255**, the secure element **111** determines whether to grant access to the TSM **170**. In one embodiment, the processor of the secure element **111** performs a cryptographic algorithm on the received message using the keys **120** stored on the secure element **111** to determine whether to grant access to the TSM **170**.

[**0061**] In certain exemplary embodiments, the SPS **115** makes an initial determination as to whether to grant access to a TSM **170** prior to the secure element **111** validating the TSM **170**. For example, when the end user network device **110** receives a request for access to the secure element **111**, the SPS **115** may evaluate the request to determine whether the TSM **170** that issued the request is the TSM **170** that the user selected prior to the request being passed to the secure element **111**. If the SPS **115** determines that the TSM **170** that issued the request is the selected TSM **170**, then the secure element **111** may validate the request in accordance with the acts of block **255**.

[**0062**] If the secure element **111** grants access to the TSM **170**, the method **200** follows the “Yes” branch to block **265**. Otherwise, if the secure element **111** determines that the TSM **170** should be blocked, the method **200** follows the “No” branch to block **260**.

[**0063**] In block **260**, the secure element **111** blocks the TSM **170** from accessing the secure element **111**. The secure element **111** also may send a message to the TSM **170** to notify the TSM **170** that the TSM **170** was not granted access.

[**0064**] In block **265** the TSM **170** provisions services at the secure element **111**. The TSM **170** may transmit to the secure element **111** one or more applications and credentials for use with those applications. The applications may be selected by the user. For example, the user may request an application from an application provider **180**. In response, the application provider **180** requests the TSM **170** to install the application onto the secure element **111** of the user. The application provider **180** also may provide information regarding the user or account information of the user to the TSM **170** for storing at the secure element **111**. For example, a credit card company may provide a payment application and information regarding a payment account of the user to the TSM **170** for installing/storing on the secure element **111**. In certain exemplary embodiments, the user may request the application from the key escrow service **150** or the secure service provider **160**.

[**0065**] In block **270**, the user accesses services provided by the selected secure service provider **160** in connection with one or more application providers **180**. For example, if the application provider **180** is a credit card company, the user may complete purchases using the end user network device **110** at an NFC-enabled POS. The NFC controller **112** may interact securely with the secure element **111** to obtain payment credentials from the secure element **111** and provide those credentials to the NFC-enabled POS via the NFC antenna **113**.

[0066] After block 270, the method 200 ends. Of course, the user can continue to access services provided by the selected secure service provider 160 or switch to another secure service provider 160.

[0067] FIG. 4 is a block flow diagram depicting a method 400 for changing secure service providers in the NFC system 300 of FIG. 3, in accordance with certain exemplary embodiments. The method 400 is described with reference to the components illustrated in FIG. 3.

[0068] In block 405, one or more secure cryptographic keys 120 are provided for a secure element 111. In certain exemplary embodiments, the secure element 111 and its keys 120 are installed on an end user network device 110 at manufacture time. In certain exemplary embodiments, the secure element 111 and its keys 120 are installed on a removable card or chip, such as a SIM card or microSD card, that is later installed on the end user network device 110.

[0069] In block 410, the keys 120 for the secure element 111 or corresponding keys are provided to the managed TSM 350. These keys 120 enable the managed TSM 350 (or another entity that receives the keys 120) to create a secure communication channel with and gain access to the secure element 111.

[0070] In block 415, the user selects a secure service provider 160 using the SPS 115. This block 415 can be the same as or similar to block 215 illustrated in FIG. 2 and described above. In block 420, the SPS 115 transmits a request to use the selected service provider 160 to the managed TSM 350 in response to the user selection. The request typically includes information identifying the selected secure service provider 160. In response to receiving the request, the managed TSM 350 processes the request.

[0071] In block 425, the managed TSM 350 performs an off-path confirmation procedure to confirm that the user initiated the request to use the selected secure service provider 160. This block is optional and is substantially similar to block 225 of FIG. 2 described above. However, the managed TSM 350 performs the off-path confirmation in block 425 rather than the key escrow service 150.

[0072] In block 430, information stored on the secure element 111 related to the previous TSM 170 and/or previous secure service provider 160 is removed from the secure element 111. For example, payment card credentials associated with the previous TSM 170 may be stored on the secure element 111 while that TSM 170 is being used in conjunction with the secure element 111. These credentials are removed from the secure element 111 prior to enabling another TSM 170 access to the secure element 111. In addition, any applications installed on the secure element 111 for the previous TSM 170 are uninstalled. In certain exemplary embodiments, the managed TSM 350 sends a command to an applet or module of the secure element 111, such as a card manager applet, to remove the information related to the previous TSM 170.

[0073] In block 435, the managed TSM 350 creates a secure communication channel with the secure service provider 160 that the user selected. This secure communication channel may be encrypted, for example using one or more cryptographic keys different than the keys 120. Other encryption techniques may be used as would be appreciated by one of ordinary skill in the art having the benefit of the present disclosure.

[0074] In block 440, the managed TSM 350 notifies the selected secure service provider 160 that the user has request

to access the services of that secure service provider 160. The managed TSM 350 also may request one or more applications from the secure service provider 160 on behalf of the user. Or, the user may request the one or more applications from the application provider 180 and the application provider 180, in turn, transmits a request to the secure service provider 160 to provide the one or more applications to the user's secure element 111. In block 445, the selected secure service provider 160 transmits the requested application(s) and any other appropriate information to the managed TSM 350. For example, this other appropriate information may include credential for accessing the secure service, such as payment card credentials.

[0075] In block 450, the managed TSM 350 creates a secure communication channel with the secure element 111 using the one or more keys 120. In block 455, the managed TSM 350 provisions services at the secure element 111. The managed TSM 350 may transmit to the secure element 111 one or more applications and credentials for use with those applications. The managed TSM 350 also may provide information regarding the user or an account of the user to the secure element 111. For example, a credit card company may provide a payment application and information regarding a payment account of the user to the managed TSM 350 for installing/storing on the secure element 111.

[0076] In block 460, which is optional, the managed TSM 350 executes business logic for the selected secure service provider 160 and serves as a proxy or intermediary between the selected secure service provider 160. Examples of business logic performed by the managed TSM 350 includes validating whether a user has a payment card with a partnered financial institution, validating credit card credentials provided by a user so that the credit card can be provisioned to the secure element 111, validating whether the selected secure service provider 160 provides a requested service for the given end user network device 150 on the MNO 130 that the end user network device 150 communicates with, and receiving a provisioning request from the user and translating the provisioning instructions for the secure element 111.

[0077] In block 465, the user accesses services provided by the selected secure service provider 160 in connection with one or more application providers 180. For example, if the application provider 180 is a credit card company, the user may redeem transit tickets using the end user network device 110 at an NFC-enabled POS. The NFC controller 112 may interact securely with the secure element 111 to obtain transit ticket credentials from the secure element 111 and provide those credentials to the NFC-enabled POS via the NFC antenna 113.

[0078] After block 465, the method 400 ends. Of course, the user can continue to access services provided by the selected secure service provider 160 or switch to another secure service provider 160.

[0079] FIG. 7 is a block flow diagram depicting a method 700 for manufacturing a secure element 111 in accordance with certain exemplary embodiments. The method 700 is described with reference to the components illustrated in FIGS. 5 and 6.

[0080] In block 710, hardware for the end user device 110 may be manufactured. Manufacturing the end user device 110 may include manufacturing the secure element 111. Alternatively, a previously manufactured secure element 111 may be incorporated into or associated with the end user device 110.



[0081] In block 720, initial card keys 120A for the secure element 111 discussed in hardware manufacturing block 610 may be configured. The card keys 120A may be configured with initial keys ready for use or keys that are merely place holders. One or more of the card keys 120A may remain open or uninitialized for future installation.

[0082] In block 730, keys associated with the secure element identity module 190 may be initialized. These keys may include the secure element identity communication key 610 and the secure element identity signing key 620. Initializing the secure element identity communication key 610 may comprise creating a public key 612 and a private key 614. Similarly, initializing the secure element identity signing key 620 may comprise creating a public key 622 and a private key 624. These keys may be used by, or in association with, the secure element identity module 190 for signing to show the authenticity of messages and for encrypted and decrypting to protect messages.

[0083] In block 740, public keys may be extracted from the secure element identity keys created in block 730. For example, the public key 612 may be extracted from the secure element identity communication key 610. Similarly, the public key 622 may be extracted from the secure element identity signing key 620. These public keys may be used by other systems to sign or encrypt messages that may then be received or verified at the secure element identity module 190 using the associated private keys.

[0084] In block 750, a secure element identity certificate 630 may be created for the secure element 111 discussed in block 310. The secure element identity certificate 630 may include the public keys extracted in block 740. The secure element identity certificate 630 may be used to authenticate the secure element identity module 190. The secure element identity certificate 630 may also be used to establish secure communications with the secure element identity module 190. These features may be supported through use of the public keys placed within the secure element identity certificate 630.

[0085] In block 760, the secure element identity certificate 630 may be signed with a key such as the manufacturer's private key. Prior to use of the secure element identity certificate 630, this signature may be verified to ensure that the secure element identity certificate 630 is authentic and correct.

[0086] In block 770, the secure element identity certificate 630 may be installed into the end user device 110. The secure element identity certificate 630 may be installed within, or in association with, the secure element identity module 190 within the secure element 111 of the end user device 110.

[0087] In block 780, initial requirements for the first reset operation may be set within the secure element identity module 190. These requirements may include the type of authorization for use in the first reset operation as discussed below with respect to block 870 of method 800. For example, the first reset operation may require having been signed by a specified key available within the secure element identity certificate 630 or some other key.

[0088] After block 780, the method 700 ends. Of course, additional secure elements may continue to be manufactured according to the method 700.

[0089] FIG. 8 is a block flow diagram depicting a method 800 for resetting a secure element 111 in accordance with

certain exemplary embodiments. The method 800 is described with reference to the components illustrated in FIGS. 5 and 6.

[0090] In block 810, the secure element identity certificate 630 may be acquired from the secure element 111. The secure element identity certificate 630 may be requested from the secure element 111 and the secure element 111 may respond with the secure element identity certificate 630 that is stored within the secure element 111.

[0091] In block 820, the secure element identity certificate 630 acquired in block 810 may be authenticated. A signature applied to the secure element identity certificate 630 may be verified to authenticate the secure element identity certificate 630. The signature may be based on a key associated with the manufacturer of the secure element 111 or the end user device 110.

[0092] In block 830, public keys may be extracted from the secure element identity certificate 630 that was authenticated in block 820. A public key for communicating secure messages with the secure element 111 may be the public key 612 originally extracted from the secure element identity communication key 610. Similarly, the public key for signing may be the public key 622 originally extracted from the secure element identity signing key 620. These public keys may be used to sign or encrypt messages that may then be received or verified at the secure element identity module 190 using the associated private keys.

[0093] In block 840, a reset request message may be constructed. This message is the reset request that may be sent to the secure element 111 to request that it reset or clear its keys such as card keys 120A. The reset request message may include new key to be installed such as new card manager keys, new fare keys, and so forth. The reset request message may also include authentication information to specify how the reset request is being authenticated as being from an approved individual such as the owner of the end user device 110. The reset request message may also include configuration information. The configuration information may include specifications as to how the next reset will be authenticated or other configuration parameters such as default key patterns for the cleared keys or so forth.

[0094] In block 850, the reset request message constructed in block 840 may be encrypted for secure communication. The reset request message may be encrypted using the secure element identity communication key 610. According to such encryption, the secure element identity module 190 may securely receive the reset request message and verify that the message was not sent maliciously, erroneously, or intended for another secure element 111.

[0095] In block 860, the reset request message that was constructed in block 840 and encrypted in block 850 may be issued to the secure element 111 of the end user device where the message may be received, verified, and acted upon by the secure element identity module 190.

[0096] In block 870, the reset request message received at the secure element 111 in block 860 may be authorized to allow the reset operation to continue. If the secure element 111 allows unrestricted access to the reset operation, an attacker could potentially use this feature to maliciously destroy information on another user's secure element 111. Hence, it is desirable to have some an authorization or confirmation mechanism to protect against unapproved access to the reset operation. Of the various options for providing this authorization protection, a few examples are discussed here.

[0097] According to a first example for authenticating reset operations, a well-known or trusted authority may cryptographically sign the reset request message. In this scenario, the secure element identity module 190 may be provided with a public key associated with the well-known authority. This key may be used to verify a signature placed on any reset request message prior to executing the reset request. Accordingly, the entity that owns the corresponding private key is entrusted to verify that the reset request is only issued by legitimate parties.

[0098] According to a second example for authenticating reset operations, submitting an authorization code over the contactless interface may be required within the secure element 111 to authorize reset request messages. This protects the user from remote attackers that try to reset their secure element 111, because the reset request message will only execute when the user intentionally enters an authorization code into a reader and sends an associated authorization command to the secure element 111 wirelessly.

[0099] According to a third example for authenticating reset operations, reset request messages may require a particular pre-determined physical communication with the end user device 110 or the secure element 111. For example, if there were a trusted operating system on the end user device 110, the trusted operating system could inform the secure element 111 that the reset request had been authorized by trusted user input on the end user device 110. Alternatively, if there were a dedicated security button or input on the end user device 110, the secure element 111 could require that this button or input be pressed or activated before allowing a reset operation to occur. This requirement would be a physical verification that the owner of the end user device 110 or some other authorized individual was intending to reset the secure element 111 and the rest would be less likely to be spoofed from an unauthorized source.

[0100] In block 880, the secure element 111 may be cleared or reset. The reset operation may be atomic to ensure that the reset occurs entirely or not at all. The reset operation may reset, delete, or wipe the contents of the secure element 111. The reset operation may include deleting all existing applications, applets, keys, and data. New card keys 120A may be installed as part of the reset operation. This may include removing all applets and execution instances. All security domains may also be removed along with all card manager keys, fare keys, and any other keys or certificates. The reset operation may clear all persistent heap and other memory. The operation may also reset all communications control related data such as secure channel sequence counters, or frequency sets. The reset operation may install new card keys 120A such as card manager keys and may also in saves the configuration information to control the authorization requirements for the next reset.

[0101] According to certain embodiments, the keys and certificates associated with the secure element identity module 190 may be exempt from the clearing of the reset operation since these keys and certificates are tied to the specific secure element 111 partly to support reset and initialization.

[0102] After block 880, the method 800 ends. Of course, the secure element 111 may continue to receive reset request messages to enable future secure element 111 reset operations within the end user device 110.

#### General

[0103] The exemplary methods and blocks described in the embodiments presented previously are illustrative, and, in alternative embodiments, certain blocks can be performed in a different order, in parallel with one another, omitted entirely, and/or combined between different exemplary methods, and/or certain additional blocks can be performed, without departing from the scope and spirit of the invention. Accordingly, such alternative embodiments are included in the invention described herein.

[0104] The invention can be used with computer hardware and software that performs the methods and processing functions described above. As will be appreciated by those having ordinary skill in the art, the systems, methods, and procedures described herein can be embodied in a programmable computer, computer executable software, or digital circuitry. The software can be stored on computer readable media. For example, computer readable media can include a floppy disk, RAM, ROM, hard disk, removable media, flash memory, memory stick, optical media, magneto-optical media, CD-ROM, etc. Digital circuitry can include integrated circuits, gate arrays, building block logic, field programmable gate arrays ("FPGA"), etc.

[0105] Although specific embodiments of the invention have been described above in detail, the description is merely for purposes of illustration. Various modifications of, and equivalent blocks corresponding to, the disclosed aspects of the exemplary embodiments, in addition to those described above, can be made by those having ordinary skill in the art without departing from the spirit and scope of the invention defined in the following claims, the scope of which is to be accorded the broadest interpretation so as to encompass such modifications and equivalent structures.

1. A computer-implemented method to reset secure memories within computing devices configured to conduct financial transactions, comprising:

receiving an encrypted reset request message for a secure memory of a computing device configured to conduct financial transactions, the encrypted reset request message being associated with a request to remove financial account information from the secure memory;

providing a communication key within the secure memory; decrypting the encrypted reset request message within the secure memory using the communication key; verifying authorization for the reset request message; and clearing parameters associated with the financial account information from the secure memory base instructions provided in the verified reset request message.

2. (canceled)

3. The computer-implemented method of claim 1, further comprising receiving a certificate request and providing a certificate associated with the communication key in response to receiving the certificate request.

4. The computer-implemented method of claim 1, wherein clearing parameters associated with the secure memory comprises resetting one or more card keys associated the account information within the secure memory.

5. The computer-implemented method of claim 1, wherein verifying authorization for the reset request message comprises verifying a signature of the reset request message from a trusted authority.

6. The computer-implemented method of claim 1, wherein verifying authorization for the reset request message comprises receiving an authorization code over a wireless interface.

7. The computer-implemented method of claim 1, wherein the communication key comprises a private public key pair.

8. The computer-implemented method of claim 1, further comprising providing a signing key within the secure memory.

9. The computer-implemented method of claim 1, wherein the secure memory is associated with a Near Field Communication interface.

10. The computer-implemented method of claim 1, wherein clearing parameters associated with the secure memory comprises an atomic operation.

11. A computer program product, comprising:

a non-transitory computer-readable medium having computer-readable program instructions embodied therein that when executed by a computer cause the computer to reset secure memories within computing devices configured to conduct financial transactions, the computer-readable program instructions comprising:

computer-readable program instructions to receive an encrypted reset request message for a secure element of a computing device configured to engage in financial transactions, the encrypted reset request message being associated with a request to remove financial account information from the secure element;

computer-readable program instructions to store supporting a communication key within a secure certificate associated with the secure element of the computing device;

computer-readable program instructions to decrypt the encrypted reset request message within the secure element using the communication key;

computer-readable program instruction to verify authorization for the reset request message; and

computer-readable program instructions to clear parameters associated with the financial account information from the secure element.

12. (canceled)

13. (canceled)

14. The computer program product of claim 11, wherein clearing parameters associated with the secure element comprises resetting card keys associated with the secure element.

15. The computer program product of claim 11, wherein the secure element is associated with a Near Field Communication interface.

16. The computer program product of claim 11, wherein verifying authorization for the reset request message comprises verifying a signature of the reset request message from a trusted authority.

17. The computer program product of claim 11, wherein verifying authorization for the reset request message comprises receiving an authorization code over a wireless interface.

18. The computer program product of claim 11, wherein verifying authorization for the reset request message comprises receiving an authorization code over a Near Field Communications interface.

19. The computer program product of claim 11, wherein clearing parameters associated with the secure element comprises an atomic operation.

20. The computer program product of claim 11, wherein the secure certificate comprises a public key for signing a message associated with the secure element.

21-26. (canceled)

27. The computer-implemented method of claim 1, wherein the secure memory is comprised in a secure element.

28. A system to reset secure memories within computing devices, comprising:

a storage device;

a processor communicatively coupled to the storage device, wherein the processor executes application code instructions that are stored in the storage device to cause the system to:

receive an encrypted reset request message for a secure memory of a computing device configured to engage in financial transactions, the encrypted reset request message being associated with a request to remove financial account information from the secure memory;

store a communication key within a secure certificate associated with the secure memory of the computing device;

decrypt the encrypted reset request message within the secure memory using the communication key;

verify authorization for the reset request message; and clear parameters associated with the financial account information from the secure memory.

29. The system of claim 28, wherein the secure memory is associated with a Near Field Communication interface.

30. The system of claim 28, wherein clearing parameters associated with the secure memory comprises resetting card keys associated with the secure memory.

31. The system of claim 28, wherein the secure memory is comprised in a secure element.

\* \* \* \* \*