

(21) Application No:	1219514.5	(51) INT CL:	<i>G06F 21/57</i> (2013.01) <i>G06F 12/14</i> (2006.01)
(22) Date of Filing:	30.10.2012		<i>G06F 21/00</i> (2013.01) <i>G06F 21/71</i> (2013.01)
			<i>G06F 21/78</i> (2013.01)
(71) Applicant(s):	Barclays Bank PLC (Incorporated in the United Kingdom) 29TH Floor, One Churchill Place, LONDON, E14 5HP, United Kingdom	(56) Documents Cited:	EP 1785902 A1 US 20090276617 A1 US 20060174109 A1 US 20030037237 A1
(72) Inventor(s):	Paul Bilke Steven Bradley Andrew Crichton George French Arthur Leung Michael Naggar Ashutosh Sureka	(58) Field of Search:	INT CL <b>G06F</b> Other: <b>ONLINE: WPI &amp; EPODOC</b>
(74) Agent and/or Address for Service:	R G C Jenkins & Co 26 Caxton Street, London, SW1H 0RJ, United Kingdom		

(54) Title of the Invention: **Device and method for secure memory access**

Abstract Title: **Providing secure access to encryption key data in a protected memory of a portable device**

(57) Encryption key data 61 is stored in a protected, non-volatile memory (NVM) 57 to which access is controlled. The encryption key data is only accessible for a predetermined time after the device 3 is powered on, preferably a predefined number of clock cycles after power-on. A boot loader program 41 is initiated after the portable device is powered on and the encryption key data is loaded from the protected NVM. Access to the protected NVM is disabled after a predetermined time has elapsed from power-on and the encryption key data is loaded before initiation of any external communication interface 21, 23. The encryption key data may be loaded from a protected NVM of a protected storage chip 51 and be loaded to volatile memory 33 on the device which may be provided on a processor 27 of the device. The protected storage chip may also comprise a microcontroller 53 for controlling access to the protected NVM, e.g. by enabling and disabling access to memory addresses.

FIG. 2

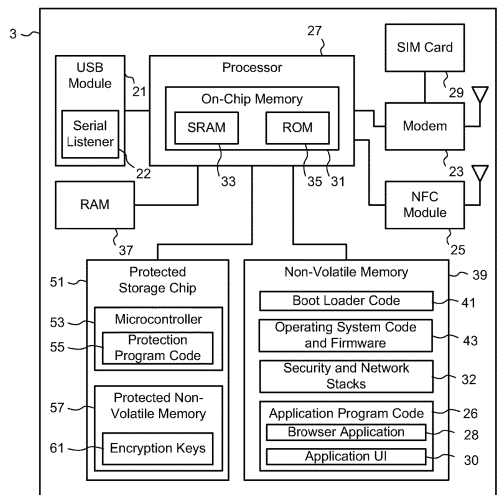
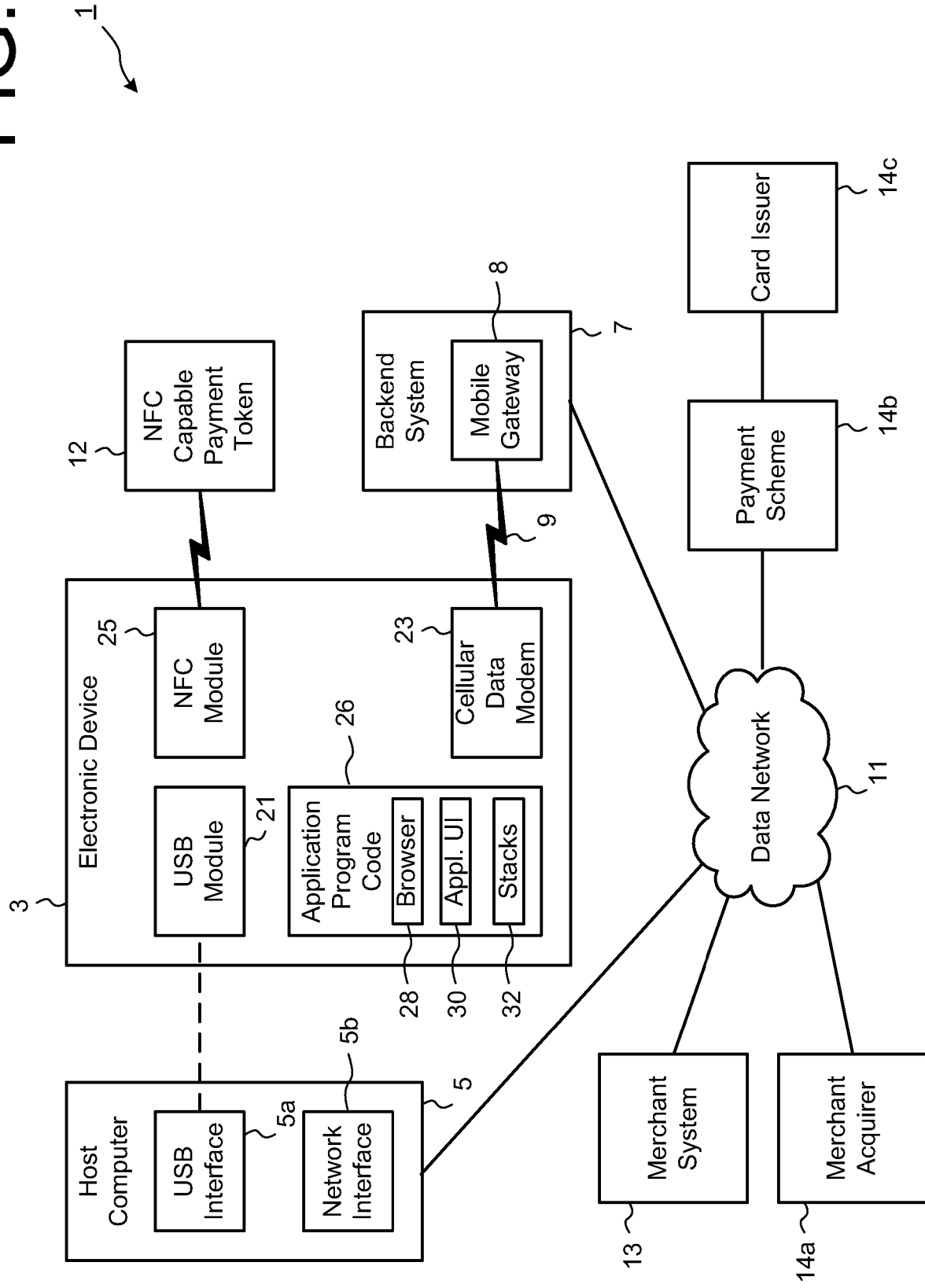


FIG. 1



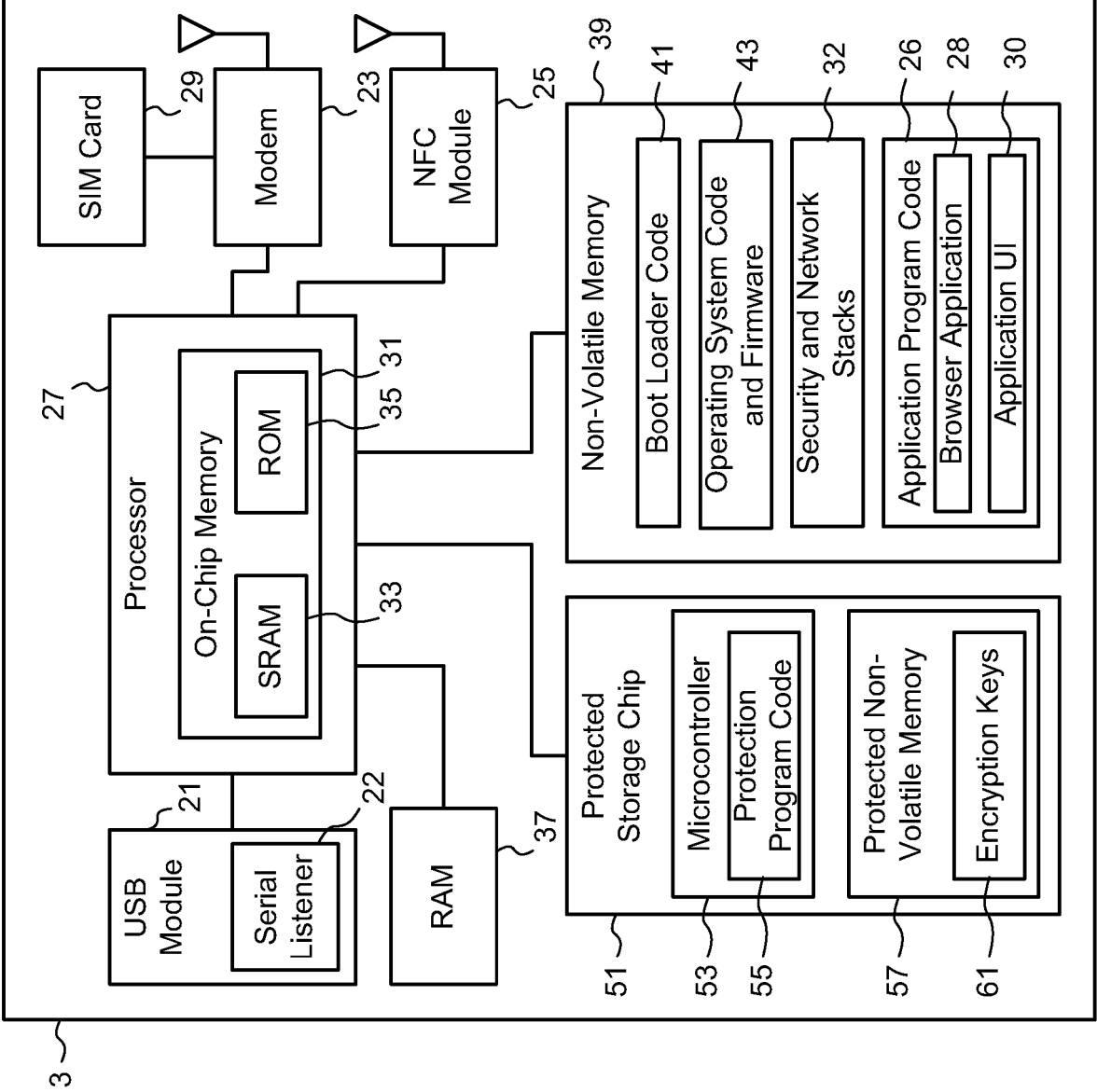


FIG. 2

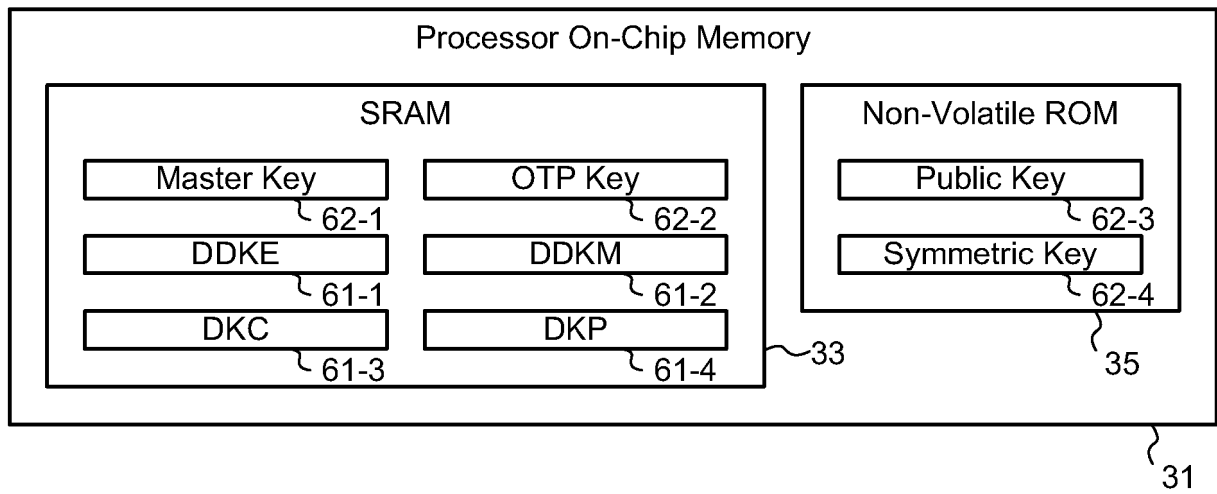


FIG. 3

4/5

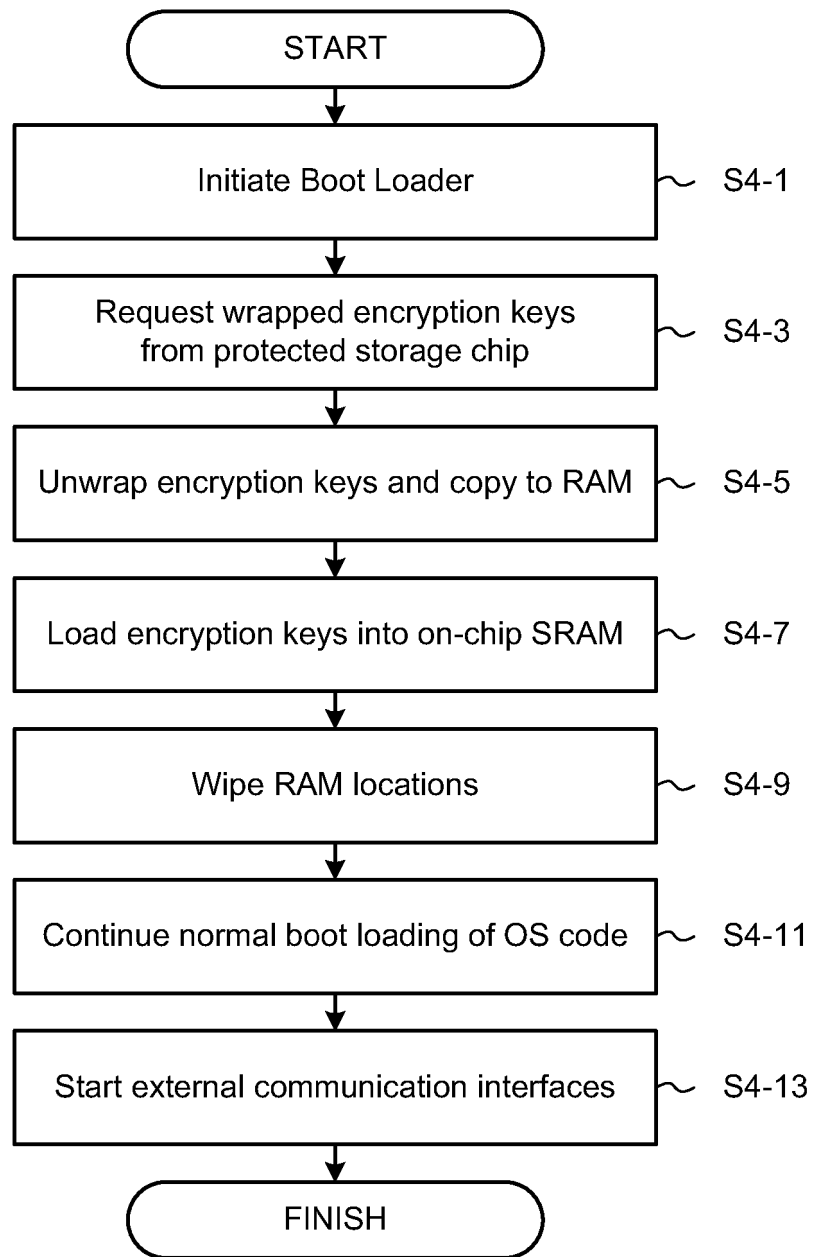


FIG. 4

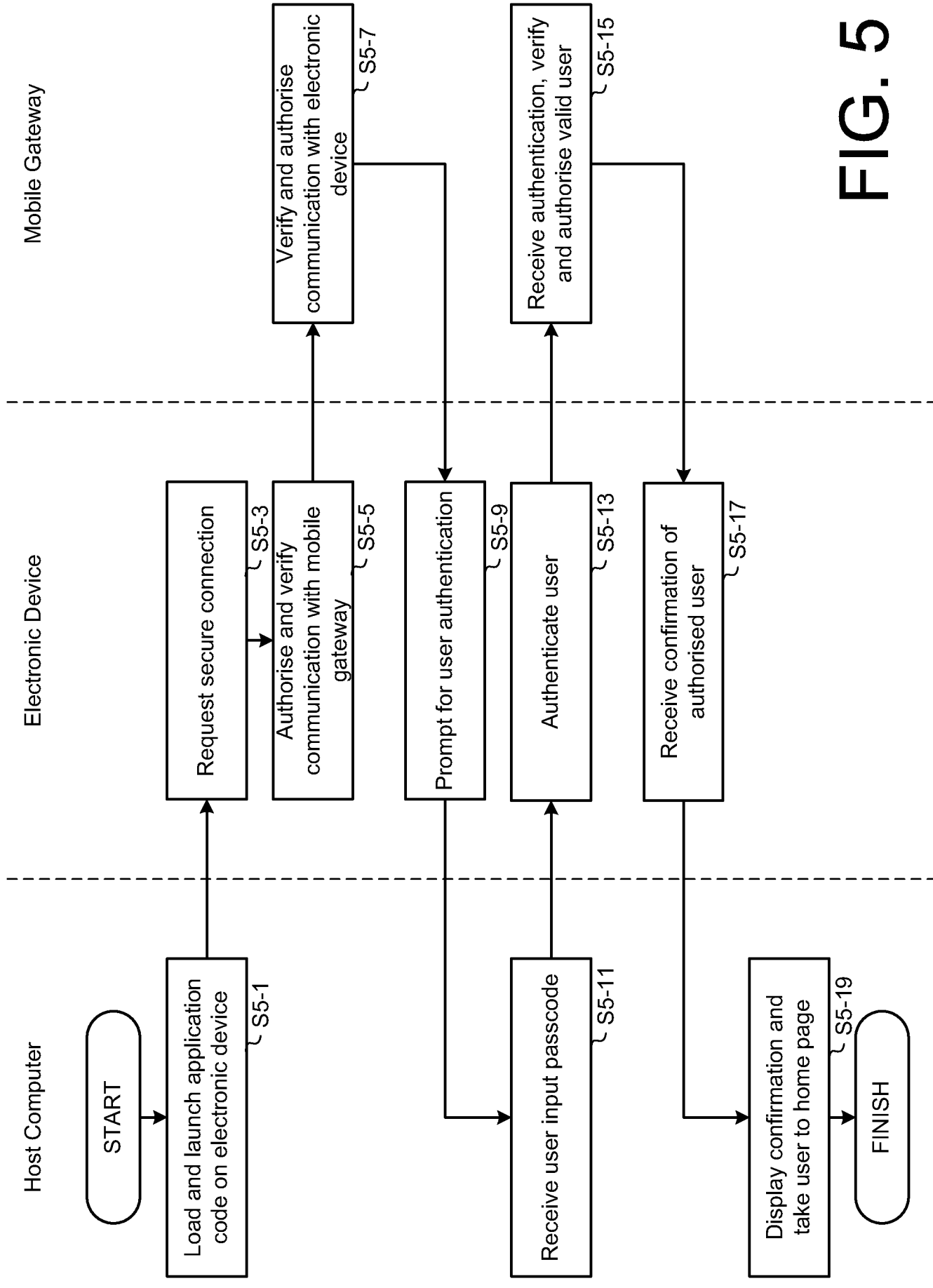


FIG. 5

## **Device and Method for Secure Memory Access**

### **Field of the Invention**

5     **[0001]** This invention relates to secure data storage, access and communication, and more particularly to a system, device and method for providing enhanced secure access to protected non-volatile memory.

### **Background of the Invention**

10     **[0002]** Secure computing environments that store and run software applications from a portable electronic Universal Serial Bus (USB) flash memory device plugged into a host computer are generally known, such as IronKey, Imation, Option CloudKey and Kobil mIDentity, in which secure authentication to the associated service and encryption is provided by the USB device itself. Typically in such environments, the USB devices are provisioned with one or more encryption keys that are used to authenticate, encrypt and decrypt program data. The encryption keys are stored within a memory of the USB device, such as a non-volatile memory. Unauthorised access to the encryption key data in the memory poses a crucial security risk.

15     **[0003]** As such secure computing environments become more prevalent, there is a need for improved systems and techniques to provide enhanced protection and security of software application data and encryption key data that are stored in the protected memory of these devices.

### **Statements of the Invention**

20     **[0004]** Aspects of the present invention are set out in the accompanying claims.

25     **[0005]** According to one aspect of the present invention, there is provided a method for loading stored encryption key data from a protected non-volatile memory of a portable device, comprising: initiating a boot loader program after the portable device is powered on, loading the encryption key data from the protected non-volatile memory of the portable device, disabling access to the protected non-volatile memory after a predetermined time after the portable device is powered on, and loading a

boot operating system of the portable device to initiate at least one external communication interface of the portable device.

5 [0006] In this way, the encryption key data is loaded from the protected non-volatile memory of a portable device before the at least one external communication interface is initiated. Preferably, the predetermined time is a predefined number of clock cycles once the electronic portable device is powered on.

10 [0007] According to another aspect of the present invention, there is provided a method for providing secured access to encryption key data stored in a protected memory of a portable device, comprising storing encryption key data in a protected non-volatile memory and enabling access to the stored encryption key data within a predetermined time after the portable device is powered on.

[0008] In a further aspect of the present invention there is provided a USB device, a data communication system, and associated computer programs arranged to carry out at least one of the above methods.

## 15 **Brief Description of the Drawings**

[0009] There now follows, by way of example only, a detailed description of embodiments of the present invention, with references to the figures identified below.

[0010] Figure 1 is a block diagram showing the main components of a secure computing environment.

20 [0011] Figure 2 is a block diagram showing the main components of an electronic device in the secure computing environment of Figure 1 according to an embodiment of the invention.

[0012] Figure 3 is a block diagram showing an example of encryption key data stored in the on-chip memory of the electronic device shown in Figure 2.

25 [0013] Figure 4 is a flow diagram illustrating the main processing steps performed by the electronic device of Figure 2 according to an embodiment.

[0014] Figure 5 is a flow diagram illustrating the main processing steps performed by components of the computing environment of Figure 1 for an example of a user authorisation process.



## Detailed Description of Embodiments of the Invention

### Secure Computing Environment

[0015] Portable USB flash memory devices that store and run software applications completely within the device itself are a way of providing flexible and mobile, yet  
5 highly secured and controlled access to online services in a secure computing environment, without using the network connection of the host computer to which the USB device is connected. In an online banking environment for example, the USB device enables secure access to a user's financial account data and account services provided by an online banking backend system, via custom browser software securely  
10 stored on the device. The software is automatically loaded and executed by the USB device when connected to a host computer, to render an application user interface (UI) for display to the user in the host computer environment.

[0016] Referring to Figure 1, a secure computing environment 1 is made up of a number of components: the electronic USB device 3, the host computer 5 and the  
15 backend system 7. The electronic device 3 is a secure and self-contained device with a USB serial communication module 21 for connecting the device to a USB interface 5a of the host computer 5. The electronic device 3 also includes an on-board cellular data modem 23 for secure network access to services provided by a backend system 7, via a direct and authenticated connection to a mobile gateway 8 of the backend  
20 system 7 over a cellular network 9. The mobile gateway 8 may be a computer server providing APIs to customer banking functionalities such as looking up account balance, making payments, making transfers, etc. The cellular network 9 may be any suitable cellular data communication network such as GPRS, EDGE, 3G, LTE, 4G, for example.

[0017] The USB serial communication module 21 provides a link between the custom  
25 browser application 28 and security and network stacks 32 on the electronic device 3, in order to translate and transmit HTTP/HTTPS requests from the browser application 28 running on the electronic device 3 via the host computer 5 over the serial USB interfaces 5a,21 and to return the responses back to the browser. Optionally, this USB serial communication module 21 can also include a set of interfaces that allow the  
30 custom browser application 28 access to custom functions on the electronic device 3.

**[0018]** The host computer 5, which can be a personal computer, portable laptop, tablet PC, or the like, typically communicates data over a data network 11 via a communication interface 5b. The host computer 5 may also include components included in commonly known computing devices, such as a processor, a display, user input devices and controllers, etc., which are not shown for clarity. The data network 11 may be any suitable data communication network such as a wireless network, a local- or wide-area network including a corporate intranet or the Internet, using for example the TCP/IP protocol. Such communication protocols are of a type that are known *per se* in data networks and need not be described further.

**[0019]** Optionally, the USB device 3 can include circuitry and logic to enable contactless payment transactions, such as a Near Field Communication (NFC) module 25 that communicates with an NFC capable payment token 12, such as an NFC payment card or NFC capable mobile device. Components of the host computer 5 can also be in communication with a merchant system 13, which could be for example a merchant's Point of Sale (POS) back-end system or an online merchant's website server system, as well as merchant acquirer 14a, payment scheme 14b and card issuer 14c components over the data network 11, which are typically provided for authorizing and settling payment transactions with the merchant system 13, and need not be described further.

**[0020]** In the normal user operation, the user plugs the electronic device 3 into the host computer 5 to automatically load and launch application program code 26 stored on the electronic device 3. In this exemplary embodiment, the application program code 26 includes an application UI 30, that can be built in HTML5 for example, and the custom browser application 28 that is used to render the application UI 30 to the user on the host computer 5. Preferably, the browser application 28 is customized to restrict use for only this device's application UI 30. The browser application 28 is coupled to the USB serial communication module 21 to make HTTP requests and receive responses via the electronic device 3 rather than directly using the host computer's network interface 5b.

## Electronic Device Architecture

**[0021]** Referring to Figure 2, an electronic device 3 according to an embodiment of the invention includes the USB interface 21 and modem 23, as discussed above, that are coupled to a processor 27. The electronic device 3 also includes a Subscriber Identity Module (SIM) 29 coupled to the modem 23 and optionally, an NFC module 25 and associated antenna. The processor 27 may be any type of processor, including but not limited to a general-purpose digital signal processor or a special purpose processor, such as the i.MX28 Application Processor family by Freescale. The processor 27 itself includes on-chip memory 31, in the form of Static Random Access Memory (SRAM) 33 and Read Only Memory (ROM) 35. The processor 27 is also coupled for access to volatile Random Access Memory (RAM) 37 and non-volatile memory 39 of the electronic device 3, for example via a data bus (not illustrated for clarity).

**[0022]** The non-volatile memory 39 stores boot loader code 41, operating system (OS) code and firmware 43, code for the security and network stacks 32, and code for application programs 26, including the custom browser application 28 and the application UI 30. The processor 27 runs the boot loader code 41 upon power up of the electronic device 3, to load the OS code 45, the security and network stacks 32 and the application program code 26 into RAM 37 for subsequent execution by the processor 27. The network and security stacks 32 include a cryptographic library that provides encryption and decryption functionality for data communicated to and from the electronic device 3.

**[0023]** Electronic device 3 is configured to route data traffic via the host computer 5, or via the onboard cellular data modem 23. The security stack consists of all the components necessary to ensure secure access to the electronic device 3, including device authorization, user authentication and network traffic encryption. The USB serial communication module 21 integrates with the security stack to apply the necessary encryption and headers to the requests it receives from the browser application 28. The network stack consists of all the components necessary to make HTTP and HTTPS requests over the cellular network 9 and the data network 11. The USB serial communication module 21 also integrates with the network stack to submit

the requests it receives from the browser application 28. Optionally, the electronic device 3 is configured with logic to perform routing of requests based on predetermined factors, such as signal strength, bandwidth speed, network data charges, etc. For example, if the cellular data signal is found to be weak or unavailable, the network stack may route the request via the network interface 5b of the host computer 5.

**[0024]** Preferably, the non-volatile memory 39 consists of one or more flash memory components, although other forms of non-volatile memory may be suitable. Optionally, the non-volatile memory 39 can be divided into logical storage partitions, for example a main partition storing current firmware and application program code, a backup partition storing a working copy of backup firmware and application program code, and an updates partition storing new version of firmware and application program code installers. One or more further spare partitions may be provided for future applications.

**[0025]** The electronic device 3 further includes a protected storage module 51, also coupled to the processor 27. In this embodiment, the protected storage module 51 is a storage chip with a dedicated microcontroller (or microprocessor) 53 for executing protection program code 55 that controls access to encryption key data 57 stored in protected non-volatile memory 59 on the storage chip 51. The protection program code 55 controls access to the protected non-volatile memory 57 by making the stored encryption key data 61 available only during a pre-defined time window. Preferably, the protected non-volatile memory 57 is only addressable for a pre-defined number of clock cycles once the electronic device 3 is powered on.

**[0026]** The loading of the encryption key data 61 is carried out as one of the initial steps in a boot loading (or bootstrapping) process and prior to initiating and accepting any external communications to the electronic device 3. During operation of the boot loading process, the encrypted boot loader code 41 is loaded by the processor 27 from the non-volatile memory 39 to RAM 37, and executed to load the encryption key data 61 from the protected storage chip 51 during the pre-defined time window for addressing the protected non-volatile memory 57. The loaded encryption keys 61 are

then available for subsequent use by the processor, for example when executing the OS code 43 and the application program code 45 to authenticate a user of the electronic device 3 and to handle service requests to and from the backend system 7. Thereafter, the processor 27 in the boot loader mode executes the remaining instructions to continue normal loading of the boot OS code and initialisation of the external communication interfaces, such as the USB interface 21 and the modem 23.

**[0027]** In an exemplary embodiment, the encryption key data 61 includes a Device Derivation Key for Encryption (DDKE) 61-1, a Device Derivation Key for Message Authentication Coding - MAC'ing (DDKM) 61-2, a Device Key for Credential Protection (DKC) 61-3 and a Device Key for Policy/Command authentication (DKP) 61-4. On device bootup, these encryption keys 61 are loaded into predefined memory slots of the SRAM 33 on the processor 27, as schematically illustrated in Figure 3. Additional encryption keys 62 may also be loaded into the on-chip memory 31, such as one or more master derivation keys 62-1 and an OTP hardware key 62-2 also loadable into the on-chip SRAM 33, and a public key 62-3 and symmetric key 62-4 loadable into the on-chip ROM 35.

**[0028]** The encryption keys 61 and 62 can be used for message encryption, message authentication (MAC), credential encryption and policy/command authentication. For example, the two DDKs (DDKE 61-1 and DDKM 61-2) can be used by the device 3 to create session based keys for each transaction between the device 3 and the backend system 7, which allows the device 3 to communicate in a secure and authenticated way with backend systems 7. The session keys can be generated by each device 3 using the appropriate Device Derivation Key, generated per message to and from the device 3 and the backend system 7. The static Device Keys for credential protection 61-3 and Policy/Command authentication 61-4 can be used to validate commands sent to the device 3 from the backend system 7 and the protection of the user entered credentials for transmission to the backend system 7, respectively. The static User Credential Protection key 61-3 allows user credentials to be protected for transmission to and processing on the backend system 7. The static Policy/Command Key 61-4 allows the device to validate instructions from the backend system 7 which will alter

the operation of the device 3. The master derivation keys 62-1 can be used during the creation of the electronic device 3 allowing a unique key set to be installed into each instance of an electronic device 3.

**[0029]** By generating a set of device specific keys 61 for each device 3, this allows a fine level of control to be exercised by the backend systems 7 as well as the ability to authenticate a specific device 3. Preferably, the encryption keys 61 can be wrapped for secured transport between the backend system 7 and the electronic device 3 during device provisioning. The wrapping can be based on, for example:

- a Device Transport Key (DTK) determined from a unique serial number and an OTP hardware key that is burnt into the electronic device at manufacture and protected such that it cannot be read out programmatically,
- a Device Transport Message Authentication Coding (MAC) Key (DTMK) used to MAC the wrapped encryption keys 61 for transport, and
- a Device Unique Key (DUK) generated by the processor combining the OTP key and various unique fields from a fuser bay, etc.

Each encryption key 61 can be wrapped under the DTK, concatenated and MAC'd. The overall message block can be type-length-value (TLV) encoded and then protected during transport via Transport Layer Security (TLS). On boot, the wrapped encryption keys 61 can be unwrapped and loaded into respective slots in the on-chip memory 31.

**[0030]** Optionally, the electronic device 3 can be further adapted to include circuitry and logic to provide a defence against subversion of hardware attacks, such as voltage tampering, etc.

**[0031]** By controlling access to the protected memory in this way, persistent storage of the encryption keys is facilitated without comprising security of the secure computing environment. The keys 61 are wrapped for storage within the new protected storage chip 51 and the microcontroller 53 functions to securely load the encryption keys 61 into the SRAM 33 of the on-chip memory 31 for use by the processor 27 of the electronic device 3. Another advantage is that the design does not affect existing cryptographic processes of the keys for encryption and the MAC'ing

processes can safely assume that the encryption keys 61 are loaded in place prior to execution of application program code 26 by the processor 27.

### **Boot Load Process**

5     **[0032]** An embodiment of a process of boot loading the electronic device 3 will now be described with reference to Figure 4, to illustrate the technical advantage of the secure data access embodiment described above. Booting (or booting up) is the initial set of operations that the computer system of the electronic device 3 performs when electrical power is switched on. The booting process begins when the electronic device 3 that has been turned off is re-energized, and ends when the device is ready to  
10     perform its normal operations.

15     **[0033]** Referring to Figure 4, the boot loading process begins at step S4-1 where the processor 27 initiates execution of the boot loader code 41 stored in the non-volatile memory 39. Optionally, the boot loader code 41 is encrypted to provide an additional layer of security and the processor 27 can decrypt the encrypted boot loader code 41 using for example the master key 62-1, before execution of the binary code. The boot loader code 41 may include instructions for the processor 27 to conduct one or more initial self-tests to confirm that components of the electronic device 3 are in working order. At step S4-3, the processor 27 proceeds to request loading of the encryption keys 61 from the protected storage chip 51. Preferably, the encryption keys 61 are  
20     wrapped using device keys as discussed above. In an embodiment, the boot loader code 41 includes instructions for addressing the protected non-volatile memory 57 via the microcontroller 53 to retrieve the encryption keys 61. As discussed above, the microcontroller 53 of the protected storage chip 51 is configured with protection program code 55 to allow or enable the protected non-volatile memory 57 to be  
25     addressable for a pre-defined number of clock cycles after the system has powered on, such that the wrapped encryption keys 61 are only available during this period of time.

30     **[0034]** At step S4-5, the processor 27 unwraps the encryption keys 61, if necessary before loading a copy of the unwrapped encryption keys 61 to RAM 37. At step S4-7, the processor 27 loads the unwrapped encryption keys 61 from RAM 37 into the SRAM 33 of the processor on-chip memory 31. The processor 27 then wipes the RAM 37

locations of the unwrapped encryption keys 61, at step S4-9. It will be appreciated that as an alternative, the system architecture may be configured to enable direct loading of the unwrapped encryption keys 61 from the protected storage chip 51 to the on-chip memory 31. However, this is less preferable and the on-chip memory 31 may not facilitate direct loading of the encryption keys 61 for additional security.

[0035] Once the unwrapped encryption keys 61 have been loaded into SRAM 33, the processor 27 continues the normal boot loading process at step S4-11 and loads the OS code 43 from the non-volatile memory 39 into RAM 37 for execution. During the normal boot loading operations, the processor 27 starts the external communication interfaces, illustrated at step S4-13, such as the USB interface 21 and the modem 23. Thereafter, the processor 27 may transition to a user operator mode and automatically load the application program code 45 from the non-volatile memory 39 to RAM 37 for execution during normal user operation, for example to carry out a process of device and user authentication.

## **Device and User Authentication Process**

[0036] An exemplary embodiment of the process of device and user authentication using the electronic device 3 will now be described with reference to Figure 5. At step S5-1, the user plugs the electronic device 3 into the host computer 5 to automatically load and launch application program code 26 stored on the electronic device 3. In the exemplary embodiment, the custom browser application 28 is launched and used to render and display the application UI 30 to the user in the host computer 5 environment. The user can interact with the UI 30 being displayed in the browser 28, for example by clicking a link or a button to select one or more functions or services that requires communication with the mobile gateway 8 of the backend system 7. In response, the browser application 28 sends a data request to a serial communication handler (not illustrated) on the host computer 5, responsible for interfacing with the USB serial communication module 21 of the electronic device 3. The serial communication handler sends the data requests to a serial listener 22 of the USB module 21, for example via a USB serial driver installed on the host computer 5.



**[0037]** At step S5-3, the processor 27 of the electronic device 3 requests a secure connection to the mobile gateway 8 of the backend system 7 over the cellular network 9, before user requests can be securely communicated with the backend system 7. Accordingly, at step S5-5, authentication and authorisation of the electronic device 3 is processed, by authorising and verifying communication with the mobile gateway 8. The requests are encrypted by the security stack 28 using the encryption keys 61 loaded into the SRAM 33 during the secure boot loading process described above.

**[0038]** In an exemplary implementation, the serial listener 22 of the USB module 21 sends the data request to the cryptography library of the network and security stack 32, to encrypt the data request using the encryption keys 61. The serial listener 22 submits the request to the network stack 32, which first checks if good cellular signal strength is available via the cellular data modem 23. If a strong cellular signal is detected, the data request is sent to the mobile gateway 8 over the cellular data network 9. Otherwise, the request can be submitted using the host computer 5 network interface 5b over the data network 11. It will be appreciated that this is one example of a possible data routing process by the electronic device 3, and in other examples, the routing decision can instead or additionally be based on other predetermined cellular network-related factors, such as bandwidth speed, network data charges, etc. If the request is sent over the cellular network 9, the data request is converted into an encrypted HTTPS request using an Open SSL Library and passed to the cellular data modem 23, which transmits the request to the mobile gateway 8. On the other hand, if the request is sent using the host computer 5 network interface 5b, the serial listener 22 sends the request to the host computer 5 via the USB serial interface 5a. The HTTPS request is sent by the host computer 5 to the mobile gateway 8 over the data network 11 (e.g. the Internet) via the network interface 5b.

**[0039]** At step S5-7, the mobile gateway 8 authorises and verifies communication with the electronic device 3, in a corresponding manner. The electronic device 3 processes authentication of the user after the electronic device 3 has been authenticated. At step S5-9, the electronic device 3 prompts the user for authentication. User authentication can take one or more of any known forms, for example by prompting

the user to input a pre-registered passcode via the application UI 30 and browser application 28, or via additional communication interfaces (not shown) that are made available on the device, such as a thumbprint scanner, dials or buttons to select passcode digits, etc. At step S5-11, the host computer 5 receives user input of a passcode via the application UI 30. The user input passcode is verified by the electronic device 3 against a stored pre-registered passcode in order to authenticate the user at step S5-13. At step S5-15, authentication of the user is securely communicated to the mobile gateway 8, which verifies that the user is valid for example by comparing received details with stored records for the user.

[0040] At step S5-17, the electronic device 3 receives confirmation from the mobile gateway 8 that the user is authorised. In response to confirmation that both the electronic device 3 and the user are authenticated and authorised, the browser application 28 and application UI 30 display confirmation to the user and proceed with normal user operation at step S5-19, for example by displaying to the user a secure web home page for the services provided by a backend system 7.

### **Alternative Embodiments**

[0041] It will be understood that embodiments of the present invention are described herein by way of example only, and that various changes and modifications may be made without departing from the scope of the invention.

[0042] For example, in the embodiments described above, the portable electronic device is a USB device. It will be appreciated that the data communication interface may be any form of standard or proprietary computing interface between the portable device and a host computing device or platform, such as IEEE 1394 (Firewire), SCSI, Thunderbolt, Lightning, etc, although in some cases, a separate power source would be required.

[0043] In the embodiments described above, the electronic device is powered by the host computer via the USB interfaces when connected. Optionally, the electronic device can include a battery and associated power charging circuitry, for powering the components of the device and enabling persistent storage of data in volatile memory if necessary.

**[0044]** In the embodiments described above, the cellular network 9 and the data network 11 are illustrated as separate networks. It will be appreciated that the data network itself can include communication links or paths over a cellular communication network such as GPRS, EDGE, 3G, 4G, LTE, for example, or a combination of such communication paths.

**[0045]** The encryption keys and passcodes described above may take any respective form, and may be composed of numeric or alphabetic symbols, non-alphanumeric symbols, or a combination of such symbols.

**[0046]** Various software implementations are described in terms of the exemplary electronic device. After reading this description, it will become apparent to a person skilled in the art how to implement the invention using other computer systems and/or computer architectures.

**[0047]** The computer programs (also called computer control logic) discussed in the embodiments above, when executed, enable the computer system of the electronic device to implement embodiments of the present invention as discussed herein. Accordingly, such computer programs represent controllers of the computer system. Where the embodiment is implemented using software, the software may be stored in a computer program product and loaded into the computer system using a removable storage drive, hard disk drive, or communication interface, to provide some examples. The terms "computer program medium" and "computer usable medium" are used generally to refer to media such as removable storage drive, a hard disk installed in hard disk drive, and signals. These computer program products are means for providing software to computer system of the electronic device. However, these terms may also include signals (such as electrical, optical or electromagnetic signals) that embody the computer program disclosed herein.

**[0048]** Alternative embodiments may be implemented as control logic in hardware, firmware, or software or any combination thereof. Further alternative embodiments may be envisaged, for example with variations and modifications to the particular sequence of steps described in the embodiments above, which nevertheless fall within the scope of the following claims.

**CLAIMS:**

1. A computer-implemented method of loading stored encryption key data from a protected non-volatile memory of an electronic portable device, comprising:
  - a. initiating a boot loader program after the portable device is powered on;
  - b. loading the encryption key data from the protected non-volatile memory of the portable device;
  - c. disabling access to the protected non-volatile memory after a predetermined time has elapsed once the portable device is powered on; and
  - d. initiating at least one external communication interface of the portable device;whereby the encryption key data is loaded from the protected non-volatile memory of a portable device before the at least one external communication interface is initiated.
2. The method of claim 1, wherein the predetermined time is a predefined number of clock cycles once the electronic portable device is powered on.
3. The method of claim 1 or 2, wherein the encryption key data is loaded from a protected non-volatile memory of a protected storage chip.
4. The method of claim 3, wherein the encryption key data is loaded to volatile memory on the electronic portable device.
5. The method of claim 4, wherein the volatile memory is provided on a processor of the electronic portable device.
6. The method of any preceding claim, wherein the at least one external communication interface includes one or more of a Universal Serial Bus (USB) interface, a modem and a Near Field Communication (NFC) interface.

7. The method of any preceding claim, wherein the electronic portable device is powered through a USB interface when connected to a host computer.
8. The method of any preceding claim, wherein the protected storage chip further comprises a microcontroller for disabling access to the protected non-volatile memory.
9. The method of any preceding claim, wherein the microcontroller controls access to the protected non-volatile memory by enabling and disabling access to memory addresses of the protected non-volatile memory.
10. The method of any preceding claim, further comprising loading and running application program code stored in the electronic portable device when the device is plugged into a host computer.
11. The method of claim 10, wherein the application program code uses the loaded encryption key data to encrypt data for communication over the external communication interface.
12. The method of any preceding claim, wherein the encryption keys stored in the protected non-volatile memory are wrapped using device specific keys, and unwrapped after loading from the protected non-volatile memory.
13. A computer-implemented method of providing secured access to encryption key data stored in a protected memory of a portable device, comprising:
  - a. storing encryption key data in a protected non-volatile memory; and
  - b. controlling access to the protected non-volatile memory whereby the stored encryption key data is only accessible within a predetermined time after the portable device is powered on.
14. A memory device comprising means for performing the method of any one of the preceding claims.

15. A storage medium comprising machine readable instructions stored thereon for causing a computer system to perform a method in accordance with any one of claims 1 to 14.
16. A method substantially as herein described with reference to the accompanying  
5 drawings.
17. A system substantially as herein described with reference to the accompanying drawings.

**CLAIMS:**

1. A computer-implemented method of loading stored encryption key data from a protected non-volatile memory of an electronic portable device, comprising:
  - a. initiating a boot loader program after the portable device is powered on;
  - b. loading the encryption key data from the protected non-volatile memory of the portable device;
  - c. disabling access to the protected non-volatile memory after a predetermined time has elapsed once the portable device is powered on; and
  - d. initiating at least one external communication interface of the portable device;whereby the encryption key data is loaded from the protected non-volatile memory of the portable device before the at least one external communication interface is initiated.
2. The method of claim 1, wherein the predetermined time is a predefined number of clock cycles once the electronic portable device is powered on.
3. The method of claim 1 or 2, wherein the encryption key data is loaded from the protected non-volatile memory of a protected storage chip.
4. The method of claim 3, wherein the protected storage chip further comprises a microcontroller for disabling access to the protected non-volatile memory.
5. The method of claim 4, wherein the microcontroller controls access to the protected non-volatile memory by enabling and disabling access to memory addresses of the protected non-volatile memory.
6. The method of any one of claims 3 to 5, wherein the encryption key data is loaded to volatile memory on the electronic portable device.

7. The method of claim 6, wherein the volatile memory is provided on a processor of the electronic portable device.
8. The method of any preceding claim, wherein the at least one external communication interface includes one or more of a Universal Serial Bus (USB) interface, a modem and a Near Field Communication (NFC) interface.
9. The method of any preceding claim, wherein the electronic portable device is powered through a USB interface when connected to a host computer.
10. The method of any preceding claim, further comprising loading and running application program code stored in the electronic portable device when the device is plugged into a host computer.
11. The method of claim 10, wherein the application program code uses the loaded encryption key data to encrypt data for communication over the external communication interface.
12. The method of any preceding claim, wherein the encryption keys stored in the protected non-volatile memory are wrapped using device specific keys, and unwrapped after loading from the protected non-volatile memory.
13. A electronic portable device comprising a protected non-volatile memory storing encryption key data, the device being configured to:
  - a. initiate a boot loader program after the portable device is powered on;
  - b. load the encryption key data from the protected non-volatile memory of the portable device;
  - c. disable access to the protected non-volatile memory after a predetermined time has elapsed once the portable device is powered on; and
  - d. initiate at least one external communication interface of the portable device;

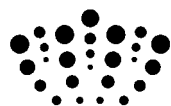


whereby the encryption key data is loaded from the protected non-volatile memory of the portable device before the at least one external communication interface is initiated.

5 14. A storage medium comprising machine readable instructions stored thereon for causing a computer system to perform a method in accordance with any one of claims 1 to 12.

15. A method substantially as herein described with reference to the accompanying drawings.

10 16. A system substantially as herein described with reference to the accompanying drawings.



**Application No:** GB1219514.5

**Examiner:** Mr Jonathan Golding

**Claims searched:** 1 - 17

**Date of search:** 11 April 2013

## Patents Act 1977: Search Report under Section 17

### Documents considered to be relevant:

Category	Relevant to claims	Identity of document and passage or figure of particular relevance
X	13-15	US 2006/0174109 A1 (FLYNN) Particularly figure 14 (integers 338, 339), paragraphs 9, 11, 13, 71, 72, 79, 80 and 81.
X	13-15	US 2009/0276617 A1 (GRELL ET AL) Particularly figures 1d, 1f, 1g, paragraphs 12, 14, 27, 289, 40, 43, 44, 45, claim 5 and claim 17.
X	13-15	EP 1785902 A1 (EMMA MIXED SIGNAL) Particularly figure 2, paragraphs 26, 30, 31 and claim 6.
X	13-15	US 2003/0037237 A1 (ABGRALL ET AL) Particularly figure 1 and paragraphs 56 - 59.

### Categories:

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

### Field of Search:

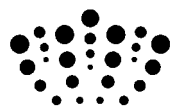
Search of GB, EP, WO & US patent documents classified in the following areas of the UKC<sup>X</sup> :

Worldwide search of patent documents classified in the following areas of the IPC

G06F

The following online and other databases have been used in the preparation of this search report

WPI & EPODOC



**International Classification:**

<b>Subclass</b>	<b>Subgroup</b>	<b>Valid From</b>
G06F	0021/57	01/01/2013
G06F	0012/14	01/01/2006
G06F	0021/00	01/01/2013
G06F	0021/71	01/01/2013
G06F	0021/78	01/01/2013