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

(19) **World Intellectual Property Organization**
International Bureau

(43) **International Publication Date**
3 January 2002 (03.01.2002)

PCT

(10) **International Publication Number**
WO 02/01368 A2

(51) **International Patent Classification**7: **G06F 12/14**

(21) **International Application Number:** PCT/US01/18756

(22) **International Filing Date:** 7 June 2001 (07.06.2001)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
09/604,377 27 June 2000 (27.06.2000) US

(71) **Applicant** *(for all designated States except US)*: **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, CA 95052 (US).

(72) **Inventors; and**
(75) **Inventors/Applicants** *(for US only)*: **HASBUN, Robert** [US/US]; 2460 Mortara Circle, Placerville, CA 95667 (US). **VOGT, James** [US/US]; 4002 Tea Rose Court, El Dorado Hills, CA 95762 (US). **BRIZEK, John** [US/US]; 2029 Williamstown Road, Franklinville, NJ 08322 (US).

(74) **Agents: MALLIE, Michael, J.**; Blakely Sokoloff Taylor & Zafman LLP, 7th Floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 et al. (US).

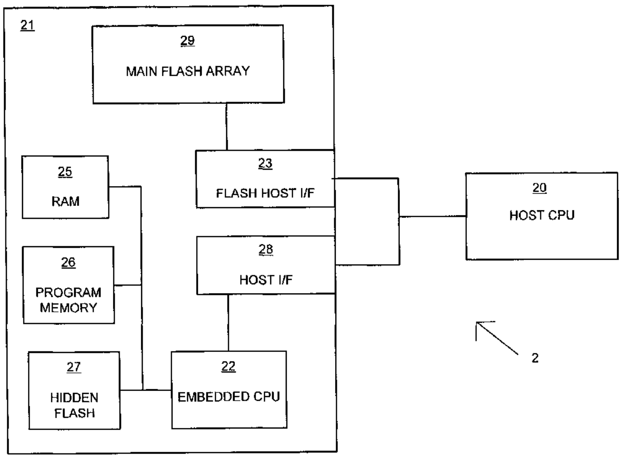
(81) **Designated States** *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) **Title:** EMBEDDED SECURITY DEVICE WITHIN A NONVOLATILE MEMORY DEVICE



(57) **Abstract:** An improved security device to control access to restricted resources on an authorized basis. A security engine, such as a processor with associated security functions, is coupled between a first modifiable non-volatile memory, such as flash memory, and a first external interface, all on the same integrated circuit. The first memory contains secure data, and is controlled solely by the security engine, which also controls the first external interface and thereby prevents read or write access to the first memory by any external device. The integrated circuit also contains a second modifiable non-volatile memory, such as flash memory, that is coupled to a second external interface for read and write access by an external device. The second memory contains non-secure data, and is controlled through the second external interface by an external device. By isolating secure processing and storage from unsecure storage on the same integrated circuit, the security functions/data are protected from dedicated attack that could intercept or control transmissions between the two, while the benefits of placing all the functions on a single integrated circuit are achieved.

EMBEDDED SECURITY DEVICE WITHIN A NONVOLATILE MEMORY DEVICE

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention pertains generally to security systems. In particular, it pertains to embedded security systems for controlling the usage of portable devices.

2. Description of the Related Art

Improvements in circuit miniaturization, battery power, and communications technology have led to widespread use of portable devices that access the resources of much larger distributed systems. An example is the use of cellular telephones, which allow subscribers to access the resources of national and global telephone systems with a device they can carry on their person. Some degree of security is built into this system by embedding a unique identification number (ID) into each cell phone, and registering both the user and the unique ID at the time the user subscribes to the service. However, a serious weakness in this approach is the fact that the cell phones are so small they can easily be lost or stolen, and anyone who has possession of the cell phone has access to the resources being paid for by the subscriber. For the users of any type of portable device that accesses restricted services, this is an obvious security problem. The same is true of any system in which physical possession of the portable device permits access to the supposedly secure system.

A conventional way to address this problem is through the use of the subscriber interface module (SIM), which is one version of a device sometimes referred to as a smart card. A SIM embeds various types of security data and processing capability in a credit-card sized artifact that communicates user-specific data to the host device before the host device will access the desired resources. This approach places at least a portion of the security processing in the artifact (the card), and typically uses a user-specific password or PIN number to verify that the person using that particular card is the person authorized to do so. Since access depends on possessing the SIM, password, and host device, this method is presumably more secure. The chance of an unauthorized party obtaining all three is less than the chance of their obtaining only the host device. This extra degree of safety assumes that the SIM is programmed to work only with that particular host device, such as a specific cell phone. If not, then possession of the SIM and password is sufficient for unauthorized use.

Fig. 1 shows a conventional system 1 using a SIM. Host system 11, which can be a cell phone, includes a host processor 12 coupled to various types of memory, which might include Read Only Memory (ROM) for program storage, random access memory (RAM) for working space, and flash memory for nonvolatile storage that is subject to infrequent change. Host system 11 also includes a user interface 14 such as a keyboard, which permits the user to input a password or personal identification number (PIN). SIM 10 is typically a plastic card, approximately the size of a credit card, containing limited processing ability in the form of its own CPU, RAM, and flash memory for maintaining the user's identification information and other related data. When SIM 10 is inserted into an interface port in host 11, interface pins (not shown) on the SIM contact mating pins in the host, which allows communication between the two devices. Power is also typically provided from the host to the SIM card through this interface.

Once connected in this manner, host CPU 12 can interrogate SIM 10 for identifying information, while the user can input his or her password through keyboard 14. If the password matches the password associated with that card, the host CPU can enable the specific services associated with that user.

Although this artifact-and-password approach provides a reasonable degree of protection when the host device is lost or taken in a random act of theft, it provides very little protection from a dedicated attack. The password and other secure data are passed between the SIM and host during operation. This data can be intercepted by placing a monitoring device into the interface, or by modifying the unsecured host, and the information obtained thereby can be used for unauthorized access through the host. Modifying a host in this manner can potentially compromise every SIM used with that host. Alternatively, if the SIM is stolen, it can be extensively analyzed to derive its secure information by plugging it into a host simulator, which would interrogate it as would a real host device. The information obtained can then permit unauthorized use and/or duplication of that particular SIM.

Encryption is sometimes used to further protect data being transferred between the SIM and host. However, dedicated security attacks are frequently devoted to determining encryption keys and decrypting the supposedly secure data.

The artifact-and-password approach is also susceptible to destructive attacks, designed to interfere with the operation of the host. One such approach is to deliberately give the system more than its maximum allowed number of sequential invalid passwords, which can cause the SIM to lock up and be unusable thereafter, unless a special password is used to override the lockup.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a system of the prior art.

Fig. 2 shows one embodiment of the invention.

Fig. 3 shows a more detailed view of the embodiment of Fig. 2.

Figs. 4A, 4B, and 4C show flow charts of various methods of the invention.

DETAILED DESCRIPTION OF THE INVENTION

Since the use of an unsecured host unnecessarily exposes the security data that is processed within that host, the present invention places both the data and the processing within a single integrated circuit so that the security functions and secure data are in a single, non-penetrable area.

Fig. 2 shows one embodiment of a system of the invention. Device 2 incorporates a host CPU 20 to control the operation of the device. Host CPU 20 can be an unsecure processor, such as the CPU in a cell phone that controls overall cell phone operations. Although a cell phone is used as an example of device 2, many other types of devices can also incorporate the invention, such as desk-top computer systems. Secure circuit 21 can be a single integrated circuit that provides a self-contained security environment within device 2, that cannot be accessed externally without its permission. Circuit 21 includes its own embedded CPU 22, so called because it is embedded within secure circuit 21. CPU 22 also controls a host interface 28 to host CPU 20. Embedded CPU 22 operates with memories 25, 26 and 27. Program memory 26 can be programmable read-only memory (PROM) or other non-volatile memory that contains the instructions for operating CPU 22. RAM 25 can be used as working space while the CPU is in operation, but would normally not be used to store permanent data, since RAM 25 will lose it contents if device 2's battery become discharged or disconnected. Hidden flash memory 27 can be used for security data that will change periodically, but must survive a power loss. Hidden flash memory 27 is where the secure user-specific data can be stored, such as user ID, password, and a list of services that the designated user is authorized to use. Although RAM 25, program memory 26 and flash memory 27 are shown as three separate types of memory, two or more of them can be consolidated into a single memory type. For example, flash memory can be used in place of RAM 25 and/or program memory 26. Although this disclosure uniformly describes the use of flash memory, other types of writeable non-volatile memory may also be used without departing from the scope of the invention.

Main flash array 29 provides a separate writeable non-volatile memory that can be used for non-secured data, and is accessible by host CPU 20 through flash host interface 23. Although host interface 28 and flash host interface 23 are shown as sharing a common bus, they can also be implemented with completely separate connections.

Secure circuit 21 provides a secure boundary surrounding all secure functions because its operation and contents are not accessible from outside circuit 21, except under specific, limited conditions which it controls. However, to be useful, user information must somehow be initially written into circuit 21. To provide an initial starting point for entering user information, in one embodiment relevant user information can be initially stored in flash memory 27 under controlled conditions, before device 2 has been placed into operation. For example, this initial setup can establish the user password and functionality for a system administrator, who would then be the only one that could subsequently enter new user data. Alterriately, the first user to input information could automatically be established as the system administrator, who would have to enter or authorize any subsequent users. Methods of entering initial user information in a security system are well known in the art.

When a potential user tries to use the system, the password or other identifying information can be input to host CPU 20, which then passes the access request and relevant data to secure circuit 21 through host interface 28. Once embedded CPU 22 determines if the user is authorized, secure circuit 21 gives a verified/ not verified indication (and possibly an indication of user-authorized services) to host 20 through interface 28, but does not output any secure information. The password and any other user identification information cannot be read from secure circuit 21 through any port.

This has significant advantages over the prior art system. For example, in the system of Fig. 2, the secure data contained in secure circuit 21 cannot be exposed because none of the buses, memory, or processing that are associated with secure data can be accessed externally to circuit 21. Among its other functions, circuit 21 is essentially a write-only storage device for security information. After the initial data is written into circuit 21 under controlled conditions, circuit 21 does not permit any of the security data to be read out by external devices, and does not permit further entry of security data except under the control of circuit 21. This makes device 2 virtually impervious to security attacks. Not only is the secure data protected, but proper checks on input data can prevent destructive data from being entered into circuit 21.

Fig. 3 shows a more detailed view of security circuit 21. Embedded CPU 22 interfaces with flash memory 27, program memory 26, RAM 25, random number generator (RNG) 38, multiplier/accumulator 39, algorithm accelerator 37, watchdog timer 36, and monotonic counter 24 over a common internal bus that is not accessible to external devices. The first three devices on this internal bus are the same as those shown in Fig. 2; the remainder are used to perform security-related functions and are described in more detail below.

Base clock 31 provides a clock source for circuit 21. One embodiment provides a 70 megahertz (MHz) clock to CPU 22. Clock divide circuit 33 can divide the base clock down to a slower rate, to be used as a source clock for watchdog timer 36 and other functions, such as alarm logic 34. Clock detector 32 can determine if base clock 31 is active and within predetermined frequency limits, while undervoltage/overvoltage (UV/OV) detector 35 can monitor the voltage levels in circuit 21. Alarm logic 34 can receive various types of alarm signals from other parts of circuit 21 and provide a consolidated alarm indication to CPU 22 and to other circuits.

The functions of circuit 21 are described in more detail below:

Processor

CPU 22 can process commands and perform flash memory management. In one embodiment, CPU 22 processes standard SIM commands so that existing legacy software can be used in the system. CPU 22 may also perform some of the cryptographic related processing, such as a hashing algorithm or a crypto algorithm. The CPU can have enough performance to execute these algorithms in real time without impacting performance. CPU 22 can also incorporate a Memory Management Unit (MMU). The MMU is a highly desirable component in security designs. It can enforce separation of code from data, and can separate the data for one processing context from that of another processing context. This separation can be used to assure that no private data inadvertently becomes mixed with non-private data.

Host Interface

Host interface 28 can provide an interface to host CPU 20 of Fig. 2. This interface can be of various types, such as parallel or serial, high or low speed, etc. To preserve compatibility with existing host devices, host interface 28 can duplicate the interface currently used in existing host systems.

In one embodiment, transfers between host CPU 20 and embedded CPU 22 can be performed one byte (or other unit of data) at a time with appropriate handshaking signals. In another embodiment, a first-in first-out buffer (FIFO) can be used in interface 28 to buffer multiple bytes, thus allowing either or both CPUs to operate more efficiently in a burst mode.

Host interface 28 can also include other signals, such as one or more pins to transfer alarm information from alarm logic 34, and to receive an external clock signal into circuit 21. The operation of host interface 28 can be under the control of embedded CPU 22, which may be able to enable or disable all or part of host interface 28 to control the flow of data and other signals being transferred to or from host CPU 20.

Program Memory

Program memory 26 contains the instructions for performing the functions that CPU 22 performs. To protect the security of the system, program memory 26 should not be alterable while in the system. It can be permanent memory such as PROM, or semi-permanent such as EPROM or flash memory.

Flash Memory

Flash memory 27 is used to store data that may change from time to time, but must survive a power loss. Flash memory is well suited for this purpose in portable devices, since it operates at voltages that are commonly available in portable devices. Flash memory can only be erased in blocks, so sufficient amounts of flash memory are used to assure that when data is changed, the entire block containing the change can be copied into a blank section, while the old block is then erased to provide a copy block for the next change.

Although uniformly described as flash memory in this disclosure, other types of non-volatile memory that are programmable in-circuit can also be used and are included within the scope of the invention.

Main flash array 29 can be used for non-secure information, and can be accessible by host CPU 20 through flash host interface 23. Although main flash array 29 and its interface 23 are functionally separated from the remainder of circuit 21, placing it on the same integrated circuit as hidden flash 27 can make efficient use of integrated circuit real estate, as well as reduce overall chip count and improve manufacturing efficiencies. Interface 23 may be the same type of interface as host interface 28, and may even connect to a common bus as shown in Fig. 2. Interfaces 23 and 28 may also be of different types, and/or may have no common connections in the system. In one embodiment, main flash memory is functionally completely separate from the security functions in circuit 21. In another embodiment, processor 22 can enable all or part or flash memory 29 after authenticating a user, and disable all or part of flash memory 29 under other conditions.

RAM Memory

Random access memory 25 is used as workspace memory while the system is operating. Since the contents of RAM memory are lost when power is removed from the RAM circuits, the data placed in RAM should not include anything that must not be lost, or that cannot be recovered upon resumption of power.

Random Number Generator

Many types of encryption require the generation of truly random numbers. A hardware generator such as RNG 38 can provide greatly superior performance over software RNG's. Hardware RNGs are known in the art. Some standards require the randomness of the RNG results to be tested in-circuit. This can require approximately 2500 bits of RAM (or alternatively, flash) memory be devoted to the testing function.

Multiplier/Accumulator

To perform encryption functions, multiplier/accumulator 39 (M/A) can support fast exponentiation and modulo reduction, and can be optimized for those functions. It need not be used for general purpose arithmetic operations, which can be performed in CPU 22. Design of the M/A function is closely related to the design of the embedded CPU. If CPU 22 is a digital signal processor (DSP), then the M/A of the DSP can be used and a separate M/A 39 on the bus may not be necessary.

Algorithm Accelerator

Algorithm accelerator 37 can be specific to the type of cryptographic algorithm being used. This dedicated hardware requires much less processing time to perform the algorithm than will a CPU. Algorithm accelerator 37 is separate in function and implementation from M/A 39. The M/A can be used to accelerate multiplication and exponentiation operations that are used in asymmetrical algorithms such as the public key encryption methodology. The algorithm accelerator speeds up symmetrical algorithms that are frequently employed to provide message privacy. Both the need for, and the specific design of, M/A 39 and accelerator 37 can depend on the particular cryptographic algorithm(s) to be employed in the circuit.

Undervoltage/Overvoltage Detection

Undervoltage/Overvoltage (UV/OV) detector 35 can protect the system from a class of cryptographic attacks based on varying the voltage inputs. These attacks drive the supply voltage outside the specified operating range for the device in an attempt to force the subject under attack to mis-operate so that plain text or keys are exposed. UV/OV 35 can detect these out-of-range voltage conditions and alert CPU 22, which can take action to stop operating before the secret information can be exposed. This also protects the system against an uncontrolled crash in the event the power supplies degrade or fail. In one embodiment, comparators are used to monitor the input voltage against reference voltages. The reference voltages are set using precision resistors as a voltage divider to bias an op amp.

Clock

Base clock 31 can provide a clock source for circuit 21. In one embodiment, base clock 31 is an internal clock operating at 70 MHz. It can be fed directly to CPU 22 as a CPU clock. It can also be divided down to lower frequencies by clock divide circuit 33 to operate such things as watchdog timer 36 and alarm logic 34. The use of an internal clock rather than an external clock prevents a dedicated attacker from manipulating the circuit by controlling the clock.

Clock Detector

Clock detector 32 can monitor the frequency of the clock signal. If the clock frequency is outside a preset range, an alarm can be generated so that the CPU can take appropriate action to shut down or otherwise protect private information. This detector is useful primarily when an external clock source is used.

Watchdog Timer

Watchdog timer 36 can monitor program execution and data transfers. The program can be designed to pre-load the timer with predetermined values, either at periodic intervals or at the start of a particular routine. If the program operates as expected, the timer will always be reloaded or stopped before time expires. If the timer expires, it indicates that an unexpected change has occurred in program execution and an alarm can be generated. Watchdog timer 36 can also be used to monitor events that depend on external operations, such as data transfers between circuit 21 and another device. Because watchdog timers normally measure time in milliseconds rather than microseconds, base clock 31 can be reduced to a lower frequency clock to provide a more useful time base for the watchdog timer.

Alarm Logic

An alarm system is critical to any security design because it protects against failures or malicious attacks that threaten the operation of the device by alerting the system to take additional protective measures. Alarm logic 34 provides a consolidation point for the various alarms that can be generated, and sends appropriate signals to CPU 22 so that it can take action to prevent loss of private information or other data. As shown in Fig. 3, alarm signals can also be sent to host interface 28, and from there to the host system, and can be provided directly to external devices.

In addition to the alarms described in the previous paragraphs, alarm logic 34 can also process the following alarms:

1) Bad key alarm - This monitors cryptographic keys and generates an alarm when a bad key is encountered. The specific identification of bad keys is unique for each algorithm.

2) Manual key entry alarm - This monitors the veracity of keys that are manually loaded. Manually loaded keys should have an error detection code, such as a parity code, or should use duplicate entries in order to verify the accuracy of the entered keys.

3) Randomizer alarm - This tests the output of RNG 38 and verifies that the output is statistically random. Various known tests can be used to perform this verification, both at power up and at various points during operation.

4) Software/firmware alarm - On power up, the program can be tested to verify that it has not been corrupted. This can be done by an Error Detection Code (EDC) or by a digital signature applied to the program contents.

5) Self Tests - Various system self tests can be performed on power up, after a reset, or when commanded by the host. Self tests can include an instruction set test, a flash memory test, a RAM test, and known-answer test with M/A 39.

Monotonic Counter

Monotonic counter 24 is shown connected to the internal bus, but can also be implemented with other connections, or can be implemented in software or firmware. A monotonic counter is a counter that can only increment (or only decrement) and never repeats a number, implying that it must never be allowed to reset or cycle back to its starting count. Monotonic counter 24 can be used to provide a unique identification number for every communication to/from circuit 21. This prevents a communication from being recorded and later played back to simulate a legitimate communication. Since the counter value used with the recorded communication would no longer match the current counter value, this type of security attack can be detected as soon as the recorded communication is transmitted to circuit 21. Additional security can be achieved by having the counter increment in a non-linear fashion, so that the current counter value cannot be guessed simply by counting the number of communications that have taken place since the recorded transmission.

Although the security contents of circuit 21 are generally inaccessible and unmodifiable from external to the circuit, in one embodiment the program of embedded CPU 22 can be modified or replaced by downloading a new program into secure circuit 21. The downloaded program can be authenticated by embedded CPU 22 before being accepted and used, to prevent an illicit program from being inserted to compromise the security of the system. The downloading can take place through host interface 28, or can take place through a separate security interface (not shown).

Figs. 4A-4C show flow charts of various method embodiments of the invention. Fig. 4A shows a method 400 of the invention. At step 401, secure data is written into a flash memory that is externally secure, i.e., it is protected from unauthorized access by devices external to the secure flash memory. At step 402, the user ID of a user needing access to the secure data is read. At step 403, the user ID is compared with the secure data to determine if the user has access rights to the data. If he does, a verify signal is sent at step 404. If he does not, a non-verify signal is sent at step 405.

Fig. 4B shows a method 410 of the invention. At step 411, non-secure data is written by an external device into a non-secure flash memory in the otherwise secure integrated circuit. At step 412, the non-secure data is read from the non-secure flash memory by the device. This method, when combined with the method of Fig. 4A, shows how the same device can include both secure and non-secure flash memory and data.

Fig. 4C shows a method 420 of the invention. At step 421, a program is transferred into the integrated circuit (IC). At step 422, the program is authenticated by the processor in the IC, and at step 423 the authenticated program is executed by the processor. The validation step permits the code in the secure system to be updated, while still protecting the secure functions from external tampering.

Secure circuit 21 can be designed around legacy components by following conventional security standards and borrowing from conventional software programs. The invention can support SIM commands, protocol, and/or electrical interfaces defined in the well-known standards ISO 7816-3 and -4, and GSM 11.11, as well as subsequent versions of those standards. This can allow secure circuit 21 to operate with existing host systems with little or no modification of the host's software interface.

The invention can also emulate the electrically erasable memories used in conventional systems.

The invention can be implemented in circuitry, as a method, or as a combination of the two. The invention can also be implemented as instructions stored on a machine-readable medium, which can be read and executed by at least one processor to perform the functions described herein. A machine-readable medium includes any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computer). For example, a machine-readable medium can include read only memory (ROM); random access memory (RAM); magnetic disk storage media; optical storage media; flash memory devices; electrical, optical, acoustical or other form of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.), and others.

The foregoing description is intended to be illustrative and not limiting. Variations will occur to those of skill in the art. Those variations are intended to be included in the invention, which is limited only by the spirit and scope of the appended claims.

We claim:

1. An apparatus, comprising:

an integrated circuit including:

a processor;

a first interface coupled to the processor to communicate between the processor and a device external to the integrated circuit;

a first non-volatile memory coupled to the processor and decoupled from the first interface, the first non-volatile memory to contain secure identification data;

a second non-volatile memory decoupled from the first memory and the first interface, the second non-volatile memory to contain non-secure data; and

a second interface coupled to the second memory for communicating between the second memory and the device;

wherein contents of the first memory cannot be read from external to the apparatus and cannot be written from external to the apparatus.

2. The apparatus of claim 1, wherein contents of the second memory cannot be read by the processor and cannot be written by the processor.

3. The apparatus of claim 1, wherein the first memory is a flash memory.

4. The apparatus of claim 1, wherein the second memory is a flash memory.

5. The apparatus of claim 1, wherein the first interface is compatible with a subscriber interface module interface.

6. The apparatus of claim 1, wherein the integrated circuit further comprises a third interface for transferring a program into the integrated circuit for authentication by the processor and for subsequent execution by the processor.

7. A system, comprising:

a device to control user access to resources;

an integrated circuit including:

a processor;

a first interface coupled to the processor and the device to communicate between the processor and the device;

a first non-volatile memory coupled to the processor and decoupled from the first interface, the first non-volatile memory to contain secure data;

a second non-volatile memory decoupled from the first interface and from the first non-volatile memory, the second non-volatile memory to contain non-secure data; and

a second interface coupled to the second memory and to the device to communicate between the second memory and the device;

wherein contents of the first memory cannot be read from external to the integrated circuit and cannot be written from external to the integrated circuit.

8. The system of claim 7, wherein contents of the second memory cannot be read by the processor and cannot be written by the processor.

9. The system of claim 7, wherein the first memory is a flash memory.

10. The system of claim 7, wherein the second memory is a flash memory.

11. The system of claim 7, wherein the first interface is compatible with a subscriber interface module interface.

12. The system of claim 7, wherein the integrated circuit further comprises a third interface for transferring a program into the integrated circuit for authentication by the processor and for subsequent execution by the processor.

13. A method comprising:

providing an integrated circuit having:

a processor and a first non-volatile memory whose contents are readable and writeable by the processor and whose contents are unreadable and unwriteable from external to the integrated circuit;

a second non-volatile memory whose contents are readable and writeable from external to the integrated circuit and whose contents are unreadable and unwriteable by the processor;

storing secure data in the first memory;

inputting user identification data to the processor from a device external to the integrated circuit;

verifying whether the user identification data corresponds to the secure data stored in the first memory;

sending a verification signal from the integrated circuit to the device if the user identification data corresponds to the secure data stored in the first memory; and

sending a non-verification signal from the integrated circuit to the device if the user identification data does not correspond to the secure data stored in the first memory.

14. The method of claim 13, further comprising writing non-secure data from the device to the second memory.

15. The method of claim 13, further comprising reading non-secure data from the second memory by the device.

16. The method of claim 13, wherein the first and second memories are flash memory.

17. The method of claim 13, further comprising:

transferring a program into the integrated circuit;

authenticating the program by the processor; and

executing the program by the processor.

18. A machine-readable medium having stored thereon instructions, which when executed by at least one first processor cause said at least one first processor to perform:

storing secure data in a first memory in an integrated circuit, contents of the first memory being readable and writeable by a CPU in the integrated circuit and being unreadable and unwriteable from external to the integrated circuit;

inputting user identification data to the CPU from a device external to the integrated circuit;

verifying whether the user identification data corresponds to secure data stored in the first memory;

sending a verification signal from the integrated circuit to the device if the user identification data corresponds to the secure data;

sending a non-verification signal from the integrated circuit to the device if the user identification data does not correspond to the secure data.

19. The medium of claim 18, wherein the first memory is a flash memory.

20. The medium of claim 18, wherein said instructions further cause said at least one processor to perform:

transferring a program into the integrated circuit;

authenticating the program by the CPU; and

executing the program by the CPU.

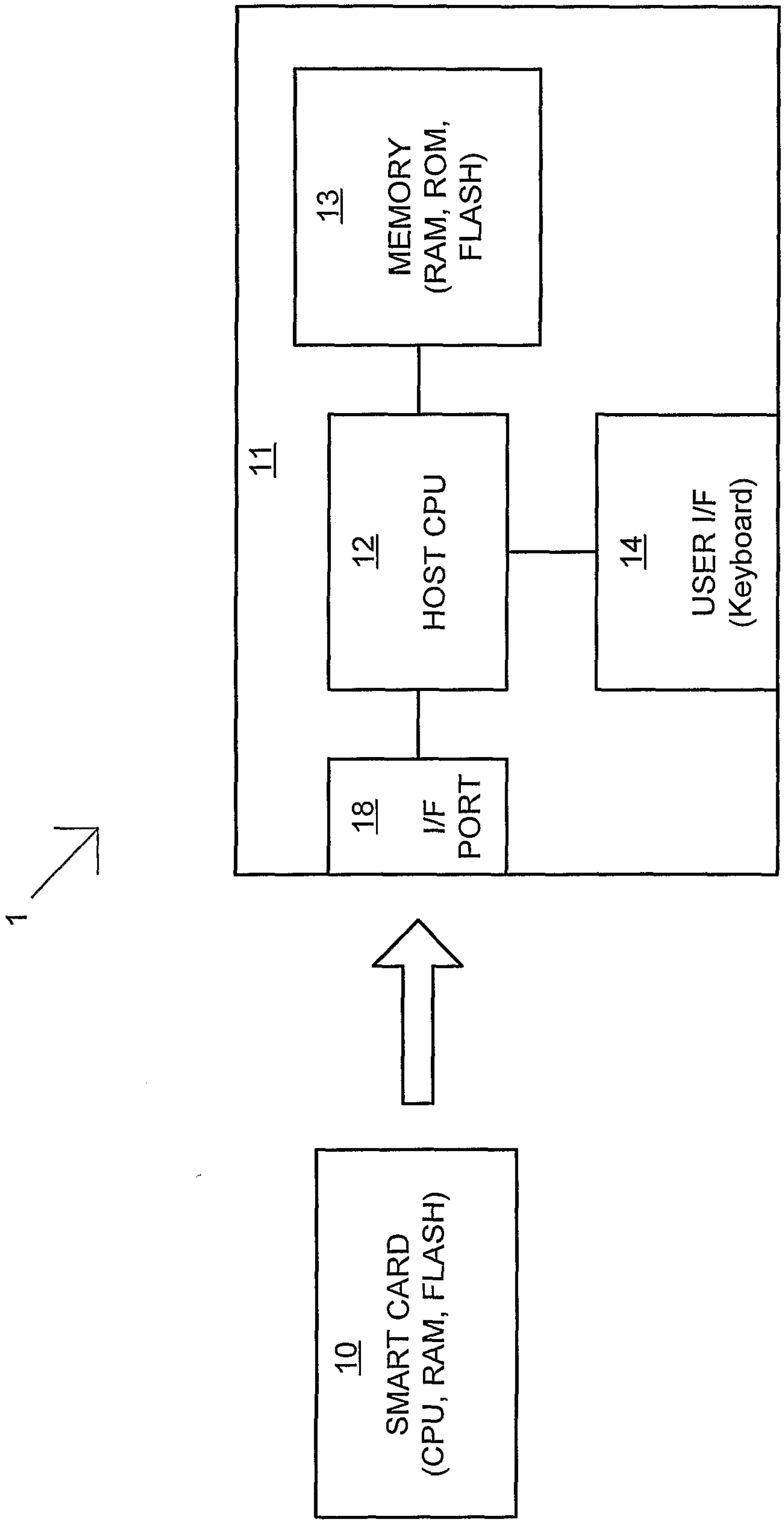


Fig. 1 Prior Art

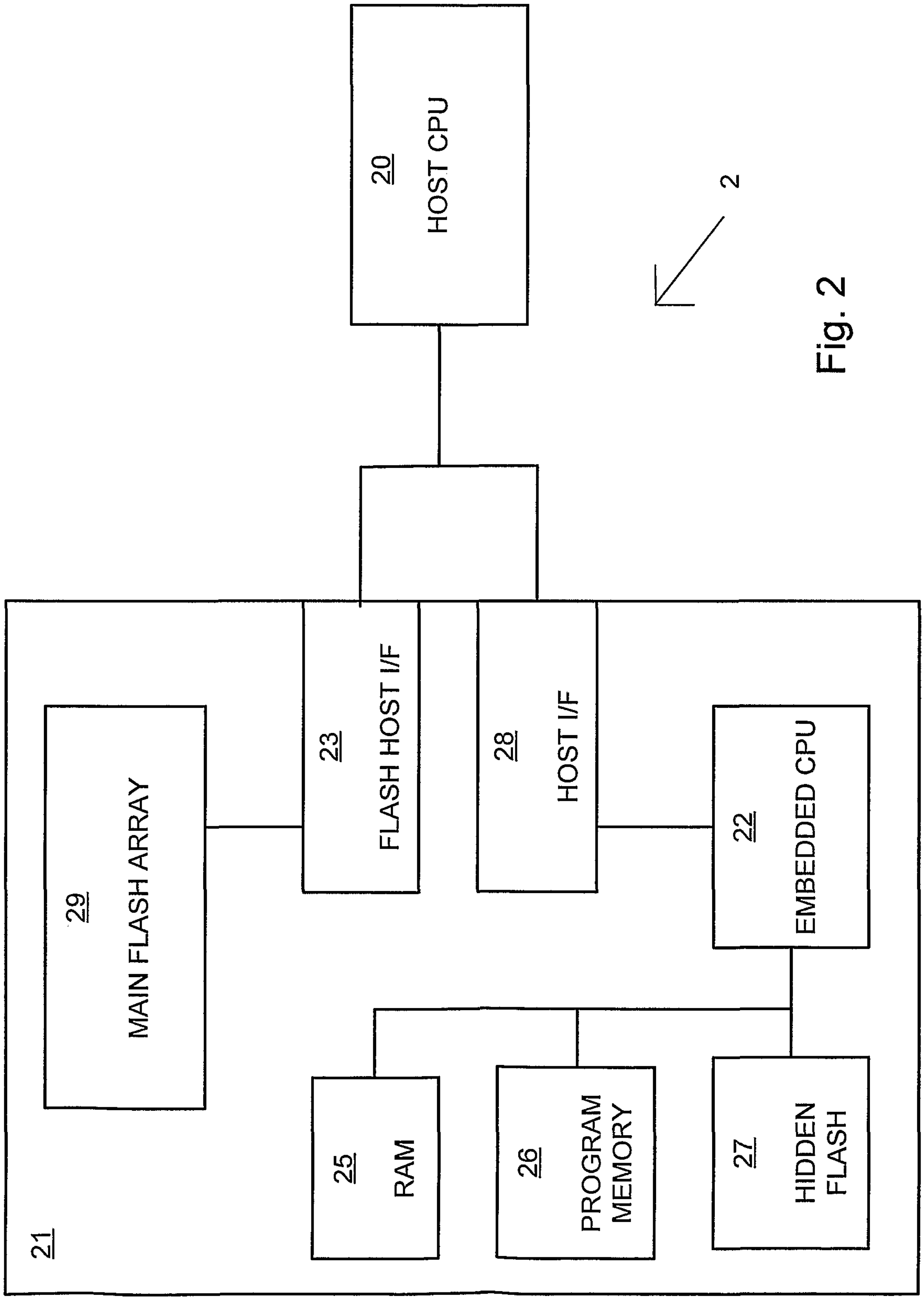

20
HOST CPU
2
21
23
FLASH HOST I/F
28
HOST I/F
29
MAIN FLASH ARRAY
22
EMBEDDED CPU
25
RAM
26
PROGRAM MEMORY
27
HIDDEN FLASH

Fig. 2

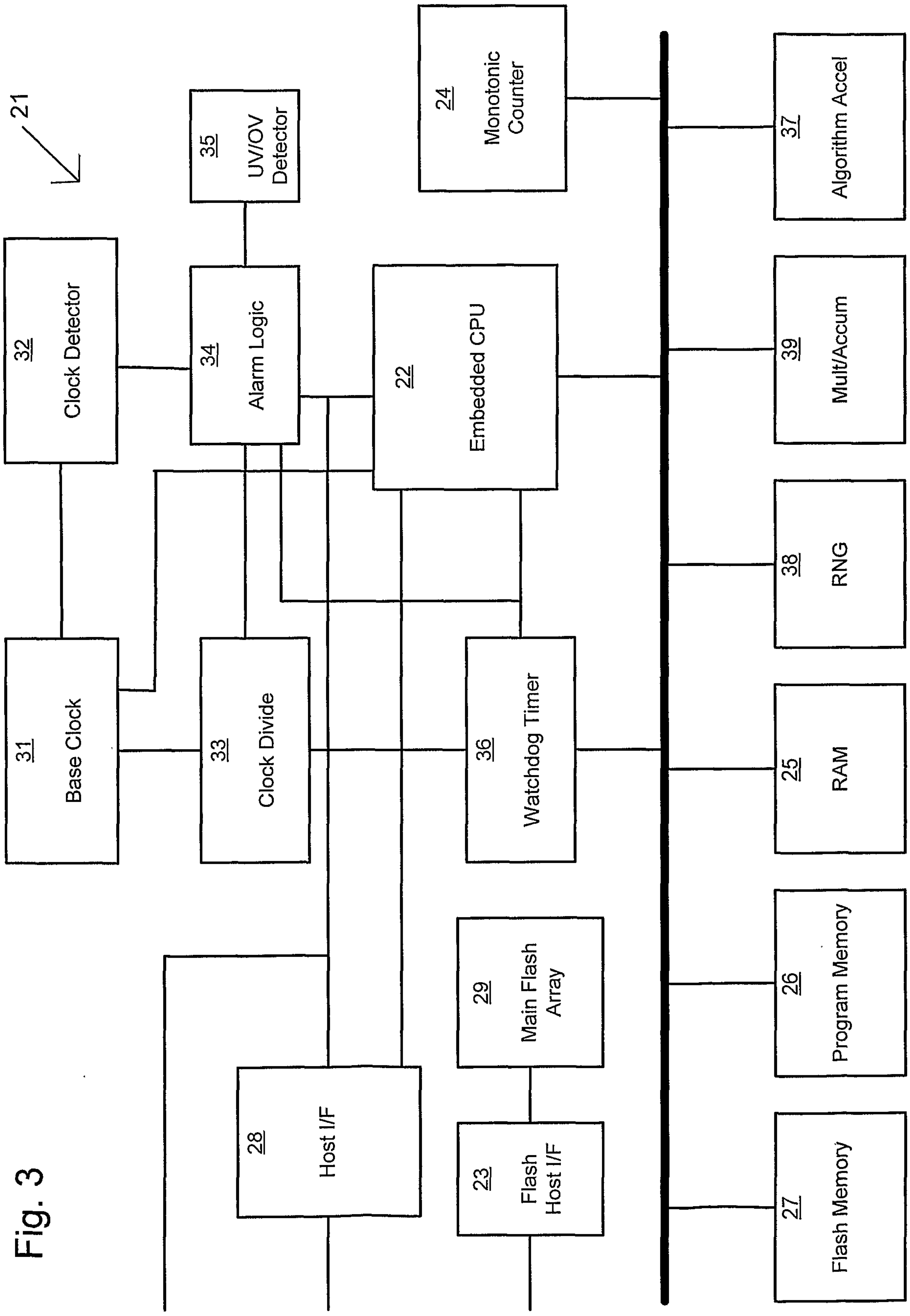


Fig. 3

21
31 Base Clock
32 Clock Detector
33 Clock Divide
34 Alarm Logic
35 UV/OV Detector
36 Watchdog Timer
22 Embedded CPU
24 Monotonic Counter
29 Main Flash Array
28 Host I/F
23 Flash Host I/F
37 Algorithm Accel
39 Mult/Accum
38 RNG
25 RAM
26 Program Memory
27 Flash Memory

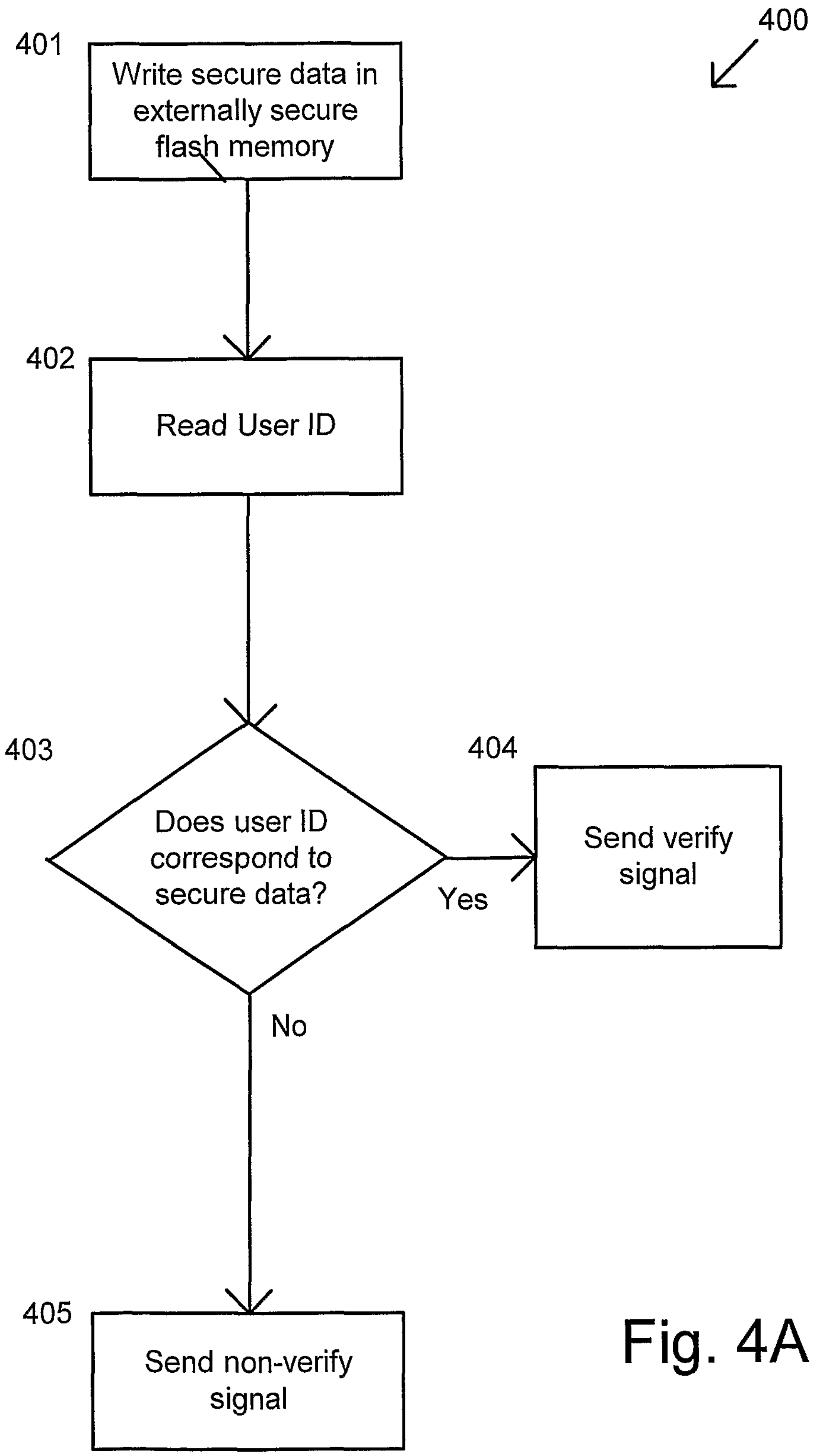

400
401
Write secure data in externally secure flash memory
402
Read User ID
403
Does user ID correspond to secure data?
404
Yes
Send verify signal
No
405
Send non-verify signal

Fig. 4A

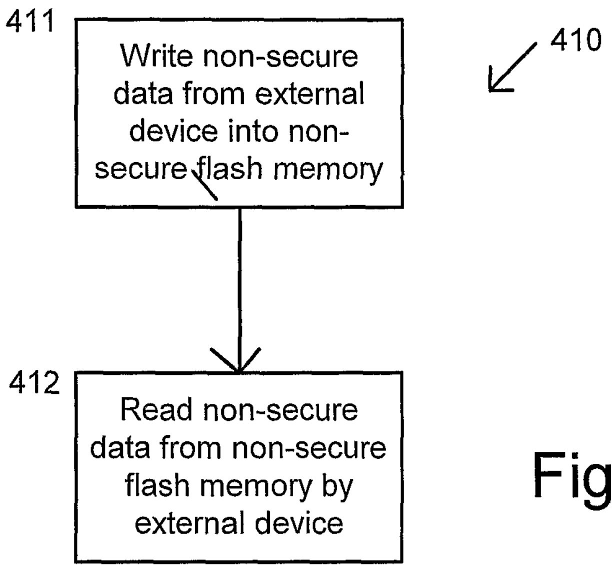

411
Write non-secure data from external device into non-secure flash memory
410
412
Read non-secure data from non-secure flash memory by external device

Fig. 4B

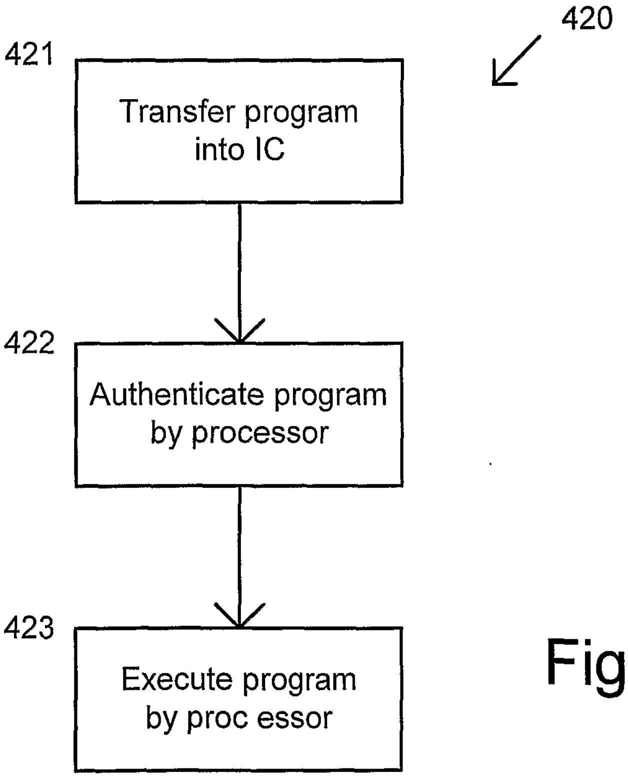

420
421
Transfer program into IC
422
Authenticate program by processor
423
Execute program by proc essor

Fig. 4C