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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 June 2014 (12.06.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/089011 A1

(51) **International Patent Classification**:
G06F 9/445 (2006.01) *G06F 21/56* (2013.01)

(21) **International Application Number**:
PCT/US2013/072756

(22) **International Filing Date**:
3 December 2013 (03.12.2013)

(25) **Filing Language**: English

(26) **Publication Language**: English

(30) **Priority Data**:
13/692,421 3 December 2012 (03.12.2012) US

(71) **Applicant**: **IMATION CORP.** [US/US]; One Imation Place, Oakdale, MN 55128 (US).

(72) **Inventor**: **HAMID, Laurence**; 561 Brookridge Crescent, Ottawa, Ontario, K4A 1Z3 (CA).

(74) **Agents**: **LONGLEY, Nathaniel, P.** et al.; 50 South Sixth Street, Suite 1500, Minneapolis, MN 55402-1498 (US).

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

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

Published:

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

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

WO 2014/089011 A1

(54) **Title**: PREBOOT ENVIRONMENT WITH SYSTEM SECURITY CHECK

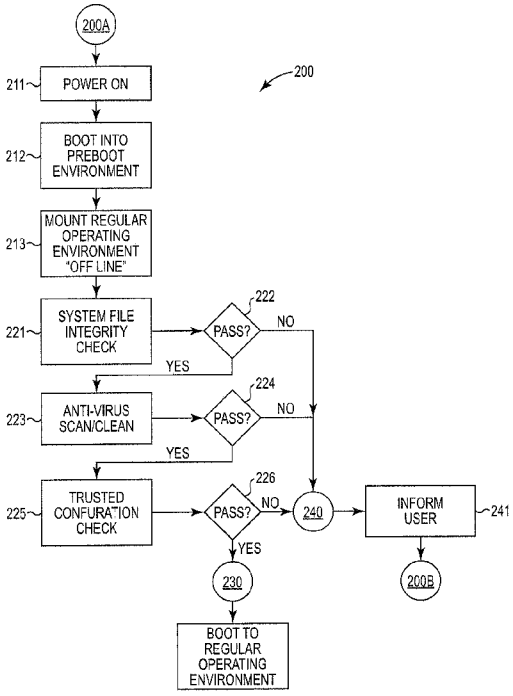


Fig. 2

(57) **Abstract**: Booting an operating system that includes a secure preboot environment that performs integrity checks against security threats. A computer system boots to a preboot environment, which performs integrity checks and other anti-malware operations. Once the preboot environment finishes, the system reboots into a regular environment. The preboot environment can reside on a secure portion of a flash memory, with a computer system booting therefrom; or the preboot environment can reside securely in the computer system. The preboot environment includes integrity checks for a regular environment, and anti-malware programming. Once the preboot environment is done, the computer system reboots into a regular environment, such as from the flash memory or on the computer system. The integrity checks confirm that files in the regular environment are unchanged or uninfected. The integrity checks include determining the accuracy of a trusted system configuration on the computer system, such as using a TPM.

PREBOOT ENVIRONMENT WITH SYSTEM SECURITY CHECK

BACKGROUND

FIELD OF THE DISCLOSURE. This application generally relates to a preboot environment with a system security check, and related matters.

BACKGROUND OF THE DISCLOSURE. "Flash memory" is a commonly used term for non-volatile data storage that can be read and written multiple times. Flash memory is often used in devices including memory that can communicate with a computer system using a universal serial bus (USB) protocol, sometimes called "flash drives", "flash memory devices", or "USB flash drives". Flash drives can often maintain relatively large amounts of data, often sufficiently large that operating system software and a suite of application program software can be maintained in the flash memory. Other devices, such as a rewritable compact disc (CD-RW or CD), or such as a rewritable digital video disc (DVD-RW, DVD+RW, or DVD-RAM), can also sometimes maintain relatively large amounts of data.

One consequence of being able to store such relatively large amounts of data is that flash drives might be used to maintain an entire operating environment, including both operating system software and application software, as well as documents or other data in use or saved by a user. An operating environment that can be carried about, often on a key ring or in a pocket, is sometimes called a "portable operating environment". For example, one such environment is the "Windows to Go" ™ environment promoted by Microsoft Corporation of Redmond, Washington. Advantages of portable operating environments include (A) that a user can maintain an entire set of settings for personalized use on any available computer system, and (B) that a user can conduct their computer processing on any available computer system, without leaving personal or otherwise sensitive data for later review by unauthorized persons.

Personal computer systems are often subject to attacks from "malware", that is, software that is designed to damage the computer system, compromise its security, or otherwise use its resources as they were not intended by the user. For some examples, malware includes computer viruses, "rootkits", and other threats. Malware has become increasingly adept at hiding and at preventing its removal by tools designed to remove it, including loading itself into critical and privileged system components and hiding from the usual methods that anti-virus products use to detect viruses and other malware. Moreover, at least some malware has become adept at installing itself through new and

different techniques, including network connections, removable storage devices, and otherwise.

For example, some rootkits have been known to infect BIOS firmware, which might not be detectable by ordinary anti-virus techniques. Malware has been known to attack infected systems, such as by downloading malicious payloads (including other and further malware), stealing private information (including user identification and passwords), hijacking computer systems (such as for use in generating spam), and allowing malicious persons to remotely terminate operation of the computer system.

While portable operating environments have several advantages, it sometimes occurs that they might be subject to malware that has infected a computer system on which the user desires to run the portable operating environment. In such cases, the portable operating environment might be at risk for itself being infected by the computer system on which the user desires to run the portable operating environment, particularly when the user of the portable operating environment cannot be assured of the security of that computer system. For example, while executing the portable operating environment, the computer system might also be executing malware that can infect the portable operating environment.

One possibility is to include, in the portable operating environment, at least some software that scans for, and possibly removes, malware. The portable operating environment might attempt to clean itself of malware, or possibly might attempt to clean the host computer system of malware as well. While this might achieve the general purpose of searching for and possibly removing malware, it has the drawback that it can be very difficult to detect and remove malware in a then-executing system.

A further possibility is to boot the computer system into a “safe mode”, in which software that scans for malware, and possibly removes it, can freely operate. While this might achieve the general purpose of searching for and possibly removing malware, it might be subject to several drawbacks. (A) Even a “safe mode” might not be free of malware infections. (B) The user might not be familiar with booting the computer system into a distinct environment. (C) While conducting these operations, the user would have unfettered access to privileged portions of the computer system, with the possibility that the user might accidentally or intentionally damage the computer system, and with the possibility that the user would be able to read and copy sensitive information on the computer system (such as files belonging to other users).

Each of these examples, as well as other possible considerations, can cause difficulty in a portable operating environment, particularly when reliability and security of that portable operating environment are important to the user. For example, there is the possibility that the portable operating environment might become infected with malware, or the host computing system might become infected with malware. Each of these possibilities might have a detrimental effect on the value of the computer system and on use of a portable operating environment therewith.

BRIEF SUMMARY OF THE DISCLOSURE

This application provides techniques for booting an operating system which includes a secure preboot environment that performs integrity checks against possible malware or other threats, and which prevents unauthorized access to privileged software and other data on the computer system. In one embodiment, a computer system first boots to a secure preboot environment, which performs integrity checks and possibly performs other anti-malware operations. Once the preboot environment has performed the integrity checks, the computer system reboots into a regular operating environment.

In one embodiment, the integrity checks can include confirming that one or more system files in the regular operating environment are unchanged or uninfected. The integrity checks can also include detecting malware and can, in some embodiments, include removing that malware, or those operations can be separately performed by an anti-malware program which is executed by the preboot environment. In one embodiment, the integrity checks can include determining the accuracy of a trusted system configuration on the computer system, such as if the computer system includes a Trusted Platform Module (TPM) or other security subsystem.

For example, other anti-malware operations can include: (A) checking system software for being up-to-date, for including all security updates, and for not including any viruses or other malware, (B) checking one or more networks, to which the computer system has access, for malicious or other untrustworthy activity, (C) checking security certificates on the computer system for being out-of-date or otherwise untrustworthiness, and (D) checking the master boot record (MBR) against a known good MBR.

In one embodiment, the secure preboot environment can be invoked at times other than just when the computer system is initially started. For example, the secure preboot environment can be invoked in response to (A) erroneous, suspicious, or other untrustworthy activity, (B) in response to user request, (C) in response to system shutdown,

or (D) in response to a system crash. In such cases, the system would shutdown and reboot into the secure preboot environment.

In one embodiment, the secure preboot environment can determine which integrity checks or other anti-malware operations to conduct, with the effect that not all such checks or other operations are necessarily conducted at every opportunity. For example, the secure preboot environment can determine which checks or other operations to perform in response to (A) a schedule, such as daily, weekly, or monthly, (B) a set of rules or heuristics, such as whether there have been too many suspicious activities or system crashes within a selected time duration, (C) a set of user-selected parameters, such as whether the user desires to have a "more strict" or "less strict" security policy, or (D) a trusted server, such as a source of anti-virus signatures.

For another example, the secure preboot environment can transition into an environment other than the regular operating environment. For example, at a time for system shutdown, the secure preboot environment can ask the user if the system should be rebooted, or if integrity checks and other anti-malware operations should be followed by a complete system shutdown (or restart), rather than rebooting into the regular operating environment.

In one embodiment, the system uses a flash memory device, which is coupled to a computer system. The computer system boots from the flash memory device, first from a preboot environment located at least in part in a secure section on the flash memory device, such as a read-only portion of the flash memory device. The preboot environment includes a program that performs integrity checks for a regular operating environment, as well as possibly an anti-malware program. Once the integrity checks have been performed and the anti-malware program has performed its operations, the computer system can reboot from the flash memory device into a regular operating environment, such as either (A) a portable operating environment, or (B) a regular operating environment on the computer system.

In alternative embodiments, the computer system boots directly from a read-only preboot environment, which performs similar functions as the preboot environment described with respect to the flash memory device. The read-only preboot environment similarly performs integrity checks, similarly possibly detects anti-malware checks and, in some embodiments, similarly removes that malware. The read-only preboot environment similarly can communicate with a TPM or other security subsystem, such as by using the computer system. The read-only preboot environment can similarly reboot to a

regular operating environment, such as a portable operating environment, or a regular operating environment on the computer system.

For example, the system can use a storage device that conforms to the Opal Storage Specification, to store relatively large amounts of data in a trusted manner. Opal drives generally include self-encrypting disk drives that are installed in the computer system and are not generally portable. As described herein, an Opal drive can include a environment, such as a portion that has been configured as read-only. For example, a pre-boot environment as described herein can maintain information on an Opal drive, can maintain information partially on a first Opal drive and on a second Opal drive, or can maintain information partially on an Opal drive and partially on a flash memory device. In this context, "maintaining information partially" on an Opal drive and partially otherwise can include the possibility that the information should be combined with a function, such as a cryptographic function, of data in separate locations.

After reading this application, those skilled in the art would recognize that techniques shown in this application are applicable to fields and information other than those fields specifically described herein, and to other than flash memories. In the context of the invention, there is no particular requirement for any such limitation. For example, the invention is also applicable to other forms of data storage in which the computer system can boot from that data storage.

Moreover, after reading this application, those skilled in the art would recognize that techniques shown in this application are applicable to methods and systems other than those involving booting from read-only sections of portable devices. In the context of the invention, there is no particular requirement for any such limitation. For example, other contexts can include other forms of secured data, such as data that is otherwise hardware-protected or software-protected.

After reading this application, which shows and describes illustrative embodiments of the disclosure, other and further embodiments of the present disclosure will become apparent to those skilled in the art. As will be realized, the various embodiments of the present disclosure are capable of modifications in various obvious aspects, all without departing from the scope and spirit of the present disclosure. The drawings and detailed description are illustrative in nature and not restrictive in any way.

BRIEF DESCRIPTION OF THE FIGURES

FIG. 1 shows a conceptual drawing of an apparatus including a flash memory device.

FIG. 2 shows a conceptual drawing of a method of operation.

DETAILED DESCRIPTION

EXAMPLE SYSTEM ELEMENTS

FIG. 1 shows a conceptual drawing of an apparatus including a flash memory device.

In one embodiment, a system 100 can include a flash memory device 110 and a computer system 120, and optionally a TPM 140. In one embodiment, a user 150, such one or more persons, can operate the system 100.

FLASH MEMORY DEVICE. The flash memory device 110 can include one or more memories 111, one or more processing elements 112, and at least one flash device plug 113. For example, the flash memory device 110 can be disposed to be compatible with a universal serial bus (USB) standard, or a variant thereof, and can be packaged in a relatively small package disposed to be coupled to a USB port 121 (described with reference to the computer system 120), or a variant thereof.

While this application primarily describes a system in which the flash memory device 110 is coupled to the computer system 120 using a USB standard or a variant thereof, in the context of the invention, there is no particular requirement for any such limitation. For a first example, the flash memory device 110 can be coupled to the computer system 120 using another way to couple devices to computer systems, such as a Thunderbolt ™ drive available from Apple, Inc. of Cupertino, California. For a second example, the flash memory device 110 can be coupled to the computer system 120 using a network connection, a wireless connection, or another type of connection available for communication between computing devices.

In one embodiment, the one or more memories 111 can include a read-only zone 131, for which the flash memory device 110 allows only read operations and prohibits write operations that might be requested by the processor 112 or the computing system 120. This has the effect that the processor 112, the computing system 120, or other users of the flash memory device 110, cannot readily alter the content of the read-only zone 131.

In one embodiment, the read-only zone 131 can include a preboot environment, including a set of instructions interpretable by a computing device, and data usable by a computing device (such as in combination or conjunction with that set of instructions), and directing that computing device to perform operations as described herein. For example, the read-only zone 131 can include any non-modifiable memory or storage ele-

ment, such an Opal storage device, ROM, or other device that is read-only. In one embodiment, the preboot environment can be maintained on more than one such read-only device, such as (A) a flash memory device plus an Opal storage device, (B) more than one flash memory device, (C) more than one Opal storage device, (D) some combination or conjunction thereof, or otherwise.

The preboot environment can direct the computing device to perform (A) integrity checking operations, such as confirming that a regular operating environment is accurate and is free from malware, (B) anti-malware operations, such as scanning a regular operating environment for malware, and (C) other security operations, such as determining that a computing system coupled to the flash memory device 110 is trusted or is authorized to operate with the flash memory device 110.

In one embodiment, the computing device that operates with the read-only zone 131 can include the processor 112, the computing system 120, some combination or conjunction thereof, or otherwise. For example, the processor 112 can make multiple security determinations. These can include at least the following. (A) The computing system 120 is authorized to operate with the set of instructions and the data in the read-only zone 131. (B) The computing system 120 can boot from the read-only zone 131 into a preboot environment. (C) The computing system 120 can communicate with a Trusted Platform Module, to assure a degree of security for the preboot environment and for the computing system 120.

In one embodiment, the one or more memories 111 can include a secure read/write zone 132, for which the flash memory device 110 allows both read operations and write operations that might be requested by the processor 112 or the computing system 120. This has the effect that the processor 112, the computing system 120, or other users of the flash memory device 110, can only read from the secure read/write zone 132 with sufficient access authorization. However, the flash memory device 110 does not allow viewing data at the request of programs in the regular operating environment. For example, the secure read/write zone 132 could be maintained hidden from a directory structure of files maintained by the flash memory device 110, without special access by the computer system 120 (such as “administrator” privileges). This has the effect that the secure read/write zone 132 can maintain information for use by the preboot environment, without that information being readily available to application programs in the regular operating environment.

In one embodiment, the secure read/write zone 132 can include read/write data locations for use by the preboot environment.

For a first example, the preboot environment can compute a secure hash value, as described below, with respect to one or more files in the regular operating environment, and can store one or more results of those computations in the secure read/write zone 132. In alternative embodiments, the preboot environment can compute the secure hash value and store one or more results of those computations in an ordinary read/write portion of the one or more memories 111, as the preboot environment has control of the processor 112 or the computing system 120 during those computations.

For a second example, the preboot environment can maintain a set of virus signatures, or other anti-malware database, in the secure read/write zone 132. This has the effect that the preboot environment can update those virus signatures, or other anti-malware database, in response to changes from a trusted source, without any requirement for a hardware change to the flash memory device 110.

Although this application primarily describes embodiments in which the TPM 140 is coupled to the computer system 120 and not to the flash memory device 110, in the context of the invention, there is no particular requirement for any such limitation. For example, the flash memory device 110 or other removable storage could be coupled to the Trusted Platform Module 140, either directly or using the computing system 120, and can communicate with the Trusted Platform Module 140. This would have the effect that the flash memory device 110 (or other removable storage) could use the TPM 140 to authenticate the preboot environment, determine the authenticity and authorization of the computer system 120, perform other security checks, or otherwise.

COMPUTER SYSTEM. The computer system 120 can include a USB port 121, a processor 122, and memory or storage 123. For example, the USB port 121 can be coupled to the processor 122, which can be coupled to the memory or storage 123.

In one embodiment, the memory or storage 123 can include a read-only zone 133, for which the computer system 120 allows only read operations and prohibits write operations that might be requested by the processor 122. This has the effect that the processor 122, or other users of the computer system 120, cannot readily alter the content of the read-only zone 133. For a first example, the read-only zone 133 can be protected against writing by hardware elements, such as if the read-only zone 133 includes ROM or other hardware for which there is no writing circuit. For a second example, the read-only zone 133 can be protected against writing by software elements, such as if a relatively low-

level or otherwise secure portion of the computer system 120 prevents writing to the read-only zone 133, or such as if the device itself implements a security technique that prevents unwanted write operations (such as an Opal storage element with a band configured to be read-only).

In one embodiment, the read-only zone 133 can be similar to the read-only zone 131 of the flash memory device 110. The read-only zone 133 can include a preboot environment for the computer system 120, including a set of instructions interpretable by a computing device, and data usable by a computing device (such as in combination or conjunction with that set of instructions), and directing that computing device to perform operations as described herein. Similar to the flash memory device 110, the preboot environment can direct the computing device to perform integrity checking operations (such as confirming that a regular operating environment is accurate and is free from malware), anti-malware operations (such as scanning a regular operating environment for malware), and other security operations (such as authenticating a TPM, as described herein).

In one embodiment, the computing system 120 can boot directly to the preboot environment in the read-only zone 133.

In one embodiment, the computing system 120 can communicate with a TPM 140 to assure that the preboot environment is authorized and secure. The preboot environment communicates with the TPM 140, which can exchange security information (not shown) with the computing system 120 to determine whether the preboot environment is authorized and secure. If the preboot environment is authorized and secure, the TPM 140 allows the preboot environment to proceed.

Similar to the flash memory device 110, in one embodiment, the memory or storage 123 can include a secure read/write zone 134, for which the computing system 120 allows both read operations and write operations that might be requested by a computing device. Similar to described above, the computing device can include the computing system's processor 122, the flash memory device's processor 112, some combination or conjunction thereof, or otherwise. Similar to described above, this has the effect that the computing system's processor 122, the flash memory device's processor 112, or other users of the read/write zone 134, can only read from the secure read/write zone 134 with sufficient access authorization. Similar to described above, the computing system 120 does not allow data in the read/write zone 134 to be viewed at the request of programs in the regular operating environment.

Similar to described above, in one embodiment, the secure read/write zone 134 can include read/write data locations for use by the preboot environment, such as computed information (similar to described above), such as virus signatures (similar to described above), or otherwise.

TRUSTED PLATFORM MODULE. The TPM 140 can be coupled to the computer system 120 and can include processing capability and security information sufficient to determine the authenticity and authorization of the flash memory device 110, the computer system 120, the preboot environment, the regular operating environment, or otherwise.

In one embodiment, the TPM 140 can be coupled to the computer system 120. The computer system 120 can communicate with the TPM 140 and can exchange security information with the TPM 140. For example, the security information can include cryptographic keys, such as asymmetric keys in a public-key cryptosystem, or symmetric keys in a symmetric cryptosystem. The security information can include, in addition or instead, other information such as shared secrets, trusted certificates, or otherwise.

In one embodiment, the computer system 120 operates under direction of the preboot environment to determine that the computer system 120 is authorized to operate with the flash memory device 110. For example, the flash memory device's processor 112 can execute instructions from the preboot environment to direct the computer system 120 to communicate with the TPM 140, and can determine if the responses from the TPM 140 indicate that the computer system 120 can be trusted.

Although this application primarily describes embodiments in which the TPM 140 operates in combination or conjunction with the preboot environment, in the context of the invention, there is no particular requirement for any such limitation. For example, the regular computing environment can, in addition or instead of the preboot environment, also use the TPM 140 to authenticate and determine a trust value for the computer system 120.

Moreover, although this application primarily describes embodiments in which the TPM 140 operates to determine whether the computer system 120 is trusted, in the context of the invention, there is no particular requirement for any such limitation, nor is there any particular requirement for any embodiment to include a TPM. For example, the TPM 140, when one is used, can operate to determine whether one or more of the following is trusted: (A) the flash memory device 110 or other removable storage, (B) the collective combination of the flash memory device 110 or other removable storage and

the computer system 120, or (C) other devices coupled to or accessible by the flash memory device 110 or the computer system 120.

METHOD OF OPERATION

FIG. 2 shows a conceptual drawing of a method of operation.

A method 200 includes a set of flow labels and method steps, as further described herein. Although this application describes one or more flow labels and method steps in sequence, in the context of the invention, there is no particular requirement for any such limitation. For example, the method 200 could be performed in parallel, in a pipelined manner, by more than one computing device or other electronic hardware, by more than one process or thread in the one or more computing devices or other electronic hardware, by one or more special purpose electronic devices, or otherwise.

PRE-BOOT ENVIRONMENT. A flow label 200A indicates a beginning of the method 200. In a first embodiment, the method 200 operates on a computer system 120. In a second embodiment, a flash memory device 110 is ready to be coupled to the computer system 120.

At a step 211, the system is powered-on. In a first embodiment, the computer system 120 is powered-on by itself. In a second embodiment, the flash memory device 110 is coupled to the computer system 120 (such as having its USB plug 113 inserted into a corresponding USB port 121 of the computer system 120), with the effect that the flash memory device 110 is powered on using power from the computer system 120.

At a step 212, the system boots into a preboot environment. In a first embodiment, the computer system 120 boots into its own preboot environment by itself, accessing a ROM, a read-only portion of RAM, or another trusted memory. In a second embodiment, the computer system 120 boots from a read-only zone 131 of the flash memory device 110 coupled to the computer system 120 (alternatively, from a read-only zone 133 of the computer system 120 itself). In both such embodiments, the computer system 120 boots into a preboot environment, distinct from its regular operating environment.

At a step 213, the preboot environment mounts the regular operating environment as data, with the effect that the regular operating environment is not yet executed by the computer system 120. In one embodiment, the computer system 120 (such as the processor 122 on the computer system 120) executes instructions from the preboot environment.

SYSTEM CHECKS. At a step 221, the preboot environment performs an integrity check for a set of system files. In one embodiment, the preboot environment maintains a list of system files to maintain secure, computes a secure hash for each of those system files, and compares that secure hash for each system file with a secure hash value it maintains for that corresponding system file. In one embodiment, the preboot environment maintains a set of metadata for each such system file, such as a modified date, and compares that metadata for each system file with a secure metadata value it maintains for that corresponding system file.

At a step 222, the preboot environment determines whether the integrity check passed. In one embodiment, the preboot environment can determine whether the integrity check passed in response to whether all system files are determined to be secure. For example, the preboot environment can determine whether all system files have the correct hash value and the correct metadata. If the integrity check passes, the method 200 proceeds with the next step. If the integrity check does not pass, the method 200 proceeds with a flow label 240.

At a step 223, the preboot environment performs an anti-virus scan, and possibly other anti-malware operations. In one embodiment, the preboot environment obtains a list of files to scan from the system files which have passed the integrity check (such as a root directory, with the effect that the preboot environment has a complete list of systems files to scan), and performs an anti-virus scan on those files to scan.

In one embodiment, the preboot environment maintains a set of virus signatures for the anti-virus scan. In one embodiment, the preboot environment updates those virus signatures using a secure communication path with a trusted source of virus signatures. For example, the preboot environment can maintain a digital certificate of the trusted source, and communicate with the trusted source using an encrypted communication protocol, such as secure sockets layer (SSL) or a variant thereof.

In one embodiment, the preboot environment can perform other anti-malware operations. For example, the preboot environment can scan for rootkits, firmware modifications, and other possible malware threats.

At a step 224, the preboot environment determines whether the anti-virus scan passed (and optionally also determines if other anti-malware operations passed). In one embodiment, the preboot environment can determine that the integrity check passed if all scanned files were shown to be absent of viruses or other malware. If the anti-virus scan

passes, the method 200 proceeds with the next step. If the anti-virus scan does not pass, the method 200 proceeds with a flow label 240.

At a step 225, if a TPM or other security subsystem is available, the preboot environment can perform a trusted configuration check in conjunction or cooperation with the TPM. In one embodiment, the preboot environment can maintain a cryptographic key (such as either a symmetric key, an asymmetric key used in a public-key cryptosystem, a shared secret, or some other secure authentication system) with which it can determine whether the computer system 120 can be trusted. See, e.g., Wikipedia article "Trusted Platform Module" (accessed the date this application is filed), hereby incorporated by reference as if fully set forth herein, and including all articles referenced therein.

At a step 226, the preboot environment determines whether the TPM passes its associated security tests. If the TPM passes its associated security tests, the method 200 proceeds with the flow label 230. If the TPM did not pass, the method 200 proceeds with the flow label 240.

REGULAR ENVIRONMENT. A flow label 230 indicates that the system is ready to boot into the regular operating environment.

At a step 231, the computer system 120 boots into the regular operating environment. As noted above, the regular operating environment can include a regular operating environment maintained on the computer system 120, such as in the memory or storage 123, or can include a portable operating environment maintained on the flash memory device 110, some combination or conjunction thereof, or otherwise.

POTENTIAL THREAT. A flow label 240 indicates that the system has detected a potential threat. While the method 200 is performed so that reaching this flow label indicates that a threat is likely, in the context of the invention, there is no particular requirement for any such limitation. For example, the method 200 can reach this flow label if the probability of a threat, or the seriousness of a threat, is sufficient to take action as indicated.

At a step 241, the method 200 informs the user 150 of the potential threat. In one embodiment, the method 200 can display information about the potential threat on an output device (not shown) coupled to the computing system 120, or using a readout on the flash memory device 110, or otherwise.

After informing the user 150 of the potential threat, the method 200 proceeds to the flow label 200B (where the method 200 terminates).

TERMINATION. A flow label 200B indicates that the method 200 terminates. In one embodiment, the method 200 is repeated each time the computer system 120 is powered on, or each time the flash memory device 110 is coupled to the computer system 120, or each time the user 150 attempts to boot the regular operating environment, or on the occurrence of any similar event.

ALTERNATIVE EMBODIMENTS. The present disclosure and many of its attendant advantages will be understood by the foregoing description, and it will be apparent that various changes may be made in the form, construction, and arrangement of the components without departing from the disclosed subject matter or without sacrificing all of its material advantages. The form described is merely explanatory, and it is the intention of the following claims to encompass and include such changes.

Certain aspects of the embodiments described in the present disclosure may be provided as a computer program product, or software, that may include, for example, a computer-readable storage medium or a non-transitory machine-readable medium having stored thereon instructions, which may be used to program a computer system (or other electronic devices) to perform a process according to the present disclosure. A non-transitory machine-readable medium includes any mechanism for storing information in a form (e.g., software, processing application) readable by a machine (e.g., a computer). The non-transitory machine-readable medium may take the form of, but is not limited to, a magnetic storage medium (e.g., floppy diskette, video cassette, and so on); optical storage medium (e.g., CD-ROM); magneto-optical storage medium; read only memory (ROM); random access memory (RAM); erasable programmable memory (e.g., EPROM and EEPROM); flash memory; and so on.

While the present disclosure has been described with reference to various embodiments, it will be understood that these embodiments are illustrative and that the scope of the disclosure is not limited to them. Many variations, modifications, additions, and improvements are possible. More generally, embodiments in accordance with the present disclosure have been described in the context of particular embodiments. Functionality may be separated or combined in procedures differently in various embodiments of the disclosure or described with different terminology. These and other variations, modifications, additions, and improvements may fall within the scope of the disclosure as defined in the claims that follow.

CLAIMS

1. A method, including steps of

booting an computer system into a secure preboot environment, said preboot environment including instructions directing a processor to perform integrity checks against malware, said preboot environment preventing unauthorized access to said instructions;

upon completing said integrity checks, rebooting said computer system into a regular operating environment, said regular operating environment including operating system software for controlling said computer system and for allowing execution of application software.

2. A method as in claim {1}, including steps of

communicating with a trusted platform module;

determining a trust value for said regular operating environment in response to a result of said steps of communicating with a trusted platform module.

3. A method as in claim {1}, including steps of

communicating with a trusted platform module;

determining an accuracy of a trusted system configuration on the computer system in response to a result of said steps of communicating with a trusted platform module.

4. A method as in claim {1}, including steps of

coupling a flash memory to said computer system;

wherein said steps of booting into said secure preboot environment include booting at least in part from a secure portion of said flash memory.

5. A method as in claim {1}, including steps of

coupling a flash memory to said computer system;

wherein said steps of rebooting said computer system into a regular operating environment including rebooting at least in part from a memory coupled to said computer system.

6. A method as in claim {1}, including steps of

coupling a flash memory to said computer system;

wherein said steps of rebooting said computer system into a regular operating environment include rebooting at least in part from a read-write portion of said flash memory.

7. A method as in claim {1}, wherein

said integrity checks include one or more of:

confirming that one or more files in said regular operating environment are unchanged or uninfected,

scanning one or more files in said regular operating environment for malware or viruses,

removing malware or viruses from one or more files in said regular operating environment.

8. A method as in claim {1}, wherein

said regular operating environment includes one or more of:

a portable operating environment,

a regular operating environment on the computer system.

9. A method as in claim {1}, wherein

said secure preboot environment is maintained at least in part on a removable storage device.

10. A method as in claim {1}, wherein

said secure preboot environment is maintained at least in part on a secure read-only memory.

11. Apparatus including

a computer system including a processor, said processor automatically booting into a secure preboot environment upon startup and operating under control thereof, said preboot environment including instructions directing a processor to perform integrity checks against malware, said preboot environment preventing unauthorized access to said instructions;

said preboot environment including instructions directing said processor to reboot said computer system into a regular operating environment, said regular operating environment including operating system software for controlling said computer system and for allowing execution of application software.

12. Apparatus as in claim {11}, including

instructions directing said processor to communicate with a trusted platform module;

instructions directing said processor to determine a trust value for said regular operating environment in response to performing said instructions to communicate with said trusted platform module.

13. Apparatus as in claim {11}, including

instructions directing said processor to communicate with a trusted platform module;

instructions directing said processor to determine an accuracy of a trusted system configuration on the computer system in response to performing said instructions to communicate with said trusted platform module.

14. Apparatus as in claim {11}, including

a flash memory coupled to said computer system;

wherein said secure preboot environment is maintained at least in part in a secure portion of said flash memory.

15. Apparatus as in claim {11}, including

a flash memory coupled to said computer system;

wherein said secure preboot environment is maintained at least in part in a secure storage device coupled to said computer system.

16. Apparatus as in claim {11}, including

a flash memory coupled to said computer system;

wherein said regular operating environment is maintained at least in part in a read-write portion of said flash memory.

17. Apparatus as in claim {11}, wherein

said instructions directing a processor to perform integrity checks include one or more of:

instructions directing said processor to confirm that one or more files in said regular operating environment are unchanged or uninfected,

instructions directing said processor to scan one or more files in said regular operating environment for malware or viruses,

instructions directing said processor to remove malware or viruses from one or more files in said regular operating environment.

18. Apparatus as in claim {11}, wherein

said regular operating environment includes one or more of:

a portable operating environment,

a regular operating environment on the computer system.

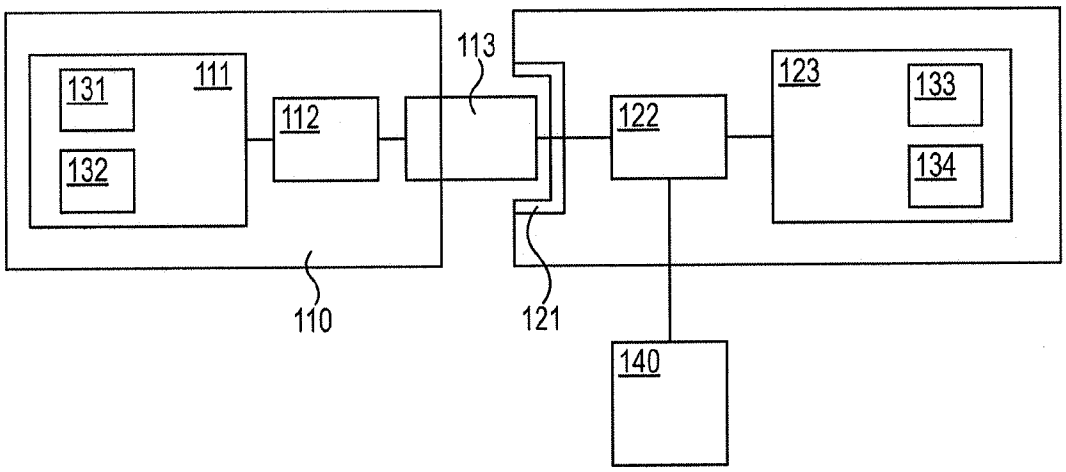


Fig. 1

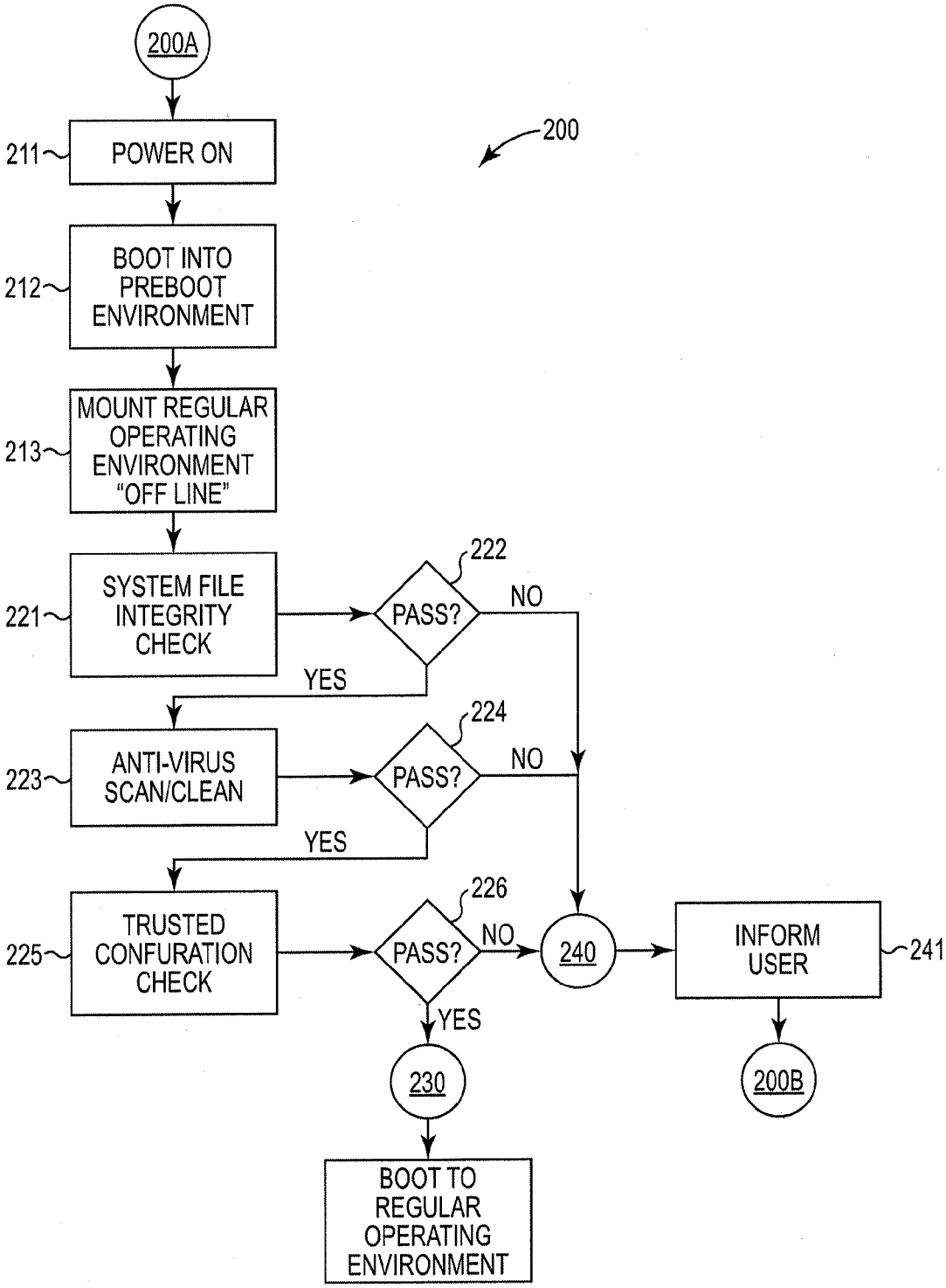

200A
211
POWER ON
200
212
BOOT INTO PREBOOT ENVIRONMENT
213
MOUNT REGULAR OPERATING ENVIRONMENT "OFF LINE"
221
SYSTEM FILE INTEGRITY CHECK
222
PASS?
NO
YES
223
ANTI-VIRUS SCAN/CLEAN
224
PASS?
NO
YES
225
TRUSTED CONFURATION CHECK
226
PASS?
NO
240
INFORM USER
241
YES
230
200B
BOOT TO REGULAR OPERATING ENVIRONMENT

Fig. 2

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 2013/072756

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/445 *(2006.01)*
G06F 21/56 *(2013.01)*

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 9/00-9/445, 11/00-11/36, H04L 1/00-1/22, 9/00, G06F 21/00-21/56

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

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

PatSearch (RUPTO internal), USPTO, PAJ, Esp@cenet, Information Retrieval System of FIPS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	BENNET YEE. Using Secure Coprocessors. School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213 [online], May 1994 [retrieved on 2014-04-22]	1-18
A	US 6061788 A (SIEMENS INF & COMM NETWORKS) 09.05.2000	1-18
A	US 6314532 B1 (LUCENT TECHNOLOGIES INC) 06.11.2001	1-18
A	US 6625754 B1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 23.09.2003	1-18
A	US 6381694 B1 (APPLE COMPUTER INC) 30.04.2002	1-18
A	US 7174457 B1 (MICROSOFT CORPORATION) 06.02.2007	1-18

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

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
24 April 2014 (24.04.2014)	29 April 2014 (29.04.2014)
Name and mailing address of the ISA/ FIPS Russia, 123995, Moscow, G-59, GSP-5, Berezhkovskaya nab., 30-1 Facsimile No. +7 (499) 243-33-37	Authorized officer A. Tokarev Telephone No. (499) 240-25-91