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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
20 November 2008 (20.11.2008)

PCT

(10) International Publication Number
WO 2008/140977 A1

(51) International Patent Classification:
G06F 21/00 (2006.01) *G06F 21/22* (2006.01)

(21) International Application Number:
PCT/US2008/062513

(22) International Filing Date: 2 May 2008 (02.05.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/747,416 11 May 2007 (11.05.2007) US

(71) Applicant *(for all designated States except US)*: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: **KHILNANI, Reshma**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **IVERSON, Kristofer N.**; One Microsoft Way, Redmond, Washington 98052-6399 (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, BR, BW, BY, BZ, CA, CH, 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, IS, JP, KE, KG, KM, 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, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

[Continued on next page]

(54) **Title:** TRUSTED OPERATING ENVIRONMENT FOR MALWARE DETECTION

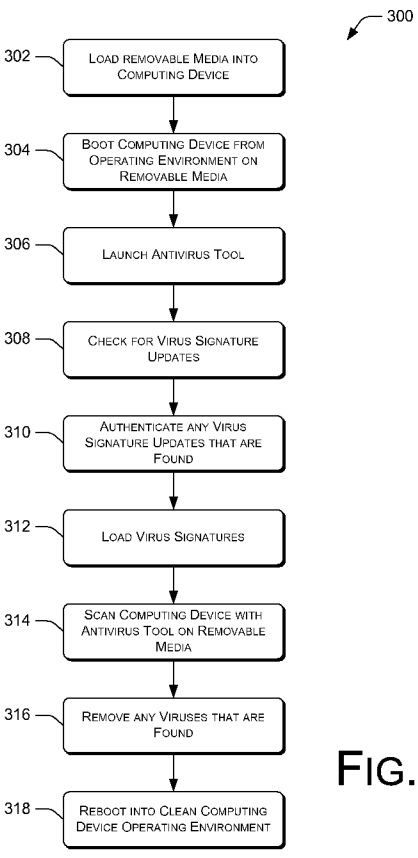

FIG. 3

(57) **Abstract:** Techniques and apparatuses for scanning a computing device for malware are described. In one implementation, a trusted operating environment, which includes a trusted operating system and a trusted antivirus tool, is embodied on a removable data storage medium. A computing device is then booted from the removable data storage medium using the trusted operating system. The trusted antivirus tool searches the computing device for malware definition updates (e.g., virus signature updates) and uses the trusted operating system to scan the computing device for malware. In another implementation, a computing device is booting from a trusted operating system on a removable device and a trusted antivirus tool on the removable device scans the computing device for malware. The removable device can update its own internal components (e.g., virus signatures and antivirus tool) by searching the computing device or a remote resource for updates and authenticating any updates that are located.

WO 2008/140977 A1

Published:

— *with international search report*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

TRUSTED OPERATING ENVIRONMENT FOR MALWARE DETECTION

BACKGROUND

[0001] Computer security is a serious concern in today's technology-driven culture. A breach of a computer's security can occur when the computer is infected with viruses and other forms of malicious software (malware). Such infections can occur, for example, when files (e.g., email) infected with malware are downloaded and opened, or infections may occur when malware accesses a computer over a network without any direct user intervention. In any case, the prevalence of these security threats has resulted in a wide variety of security-related tools that are available for computers. Examples of these tools include antivirus programs, adware scanners, firewalls, and the like. Despite the availability of these tools, computers continue to be infected with malware.

[0002] One reason for the persistence of malware infection is the ability of some malware to hide from security tools. Malware can hide from many security tools through the use of a rootkit, which generally stated, is a set of software tools intended to conceal running processes, files or system data from a computer's operating system. Rootkits can hook themselves very low in a computer's system (e.g., at the kernel level) and intercept the principal system services that the computer's operating system and other applications utilize while running on the computer. In one example, an antivirus tool that resides on a computer scans the computer's hard disk for viruses. As part of the scan process, the computer's operating system makes one or more function calls, such as an "open file" call for a certain file. However, malware that is resident on the computer may use a rootkit to intercept the "open file" function call and return a "file missing" error or return the wrong file. Thus, the antivirus tool is unable to access the requested file and check it for virus infection. If the requested file is infected with a virus, the infection will persist undetected.

SUMMARY

[0003] Techniques and apparatuses for scanning a computing device for malware are described. In one implementation, a trusted operating environment, which includes a trusted operating system and a trusted antivirus tool, is embodied on a

removable data storage medium. A computing device is then booted from the removable data storage medium using the trusted operating system. The trusted antivirus tool searches the computing device for malware definition updates (e.g., virus signature updates) and interacts with the trusted operating system to scan the computing device for malware. Another implementation uses a removable device, such as a universal serial bus (USB) drive with a microcontroller, to store the trusted operating environment. A computing device is booted using the trusted operating system and the trusted antivirus tool scans the computing device for malware. The removable device can update its own internal components (e.g., virus signatures and antivirus tool) by searching the computing device or a remote resource for updates and authenticating any updates that are located.

[0004] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] The detailed description is described with reference to the accompanying figures. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The use of the same reference numbers in different figures indicates similar or identical items.

[0006] Fig. 1 illustrates an exemplary architecture and removable data storage medium for implementing techniques to scan a computing device for malware.

[0007] Fig. 2 illustrates an exemplary architecture and removable device for implementing techniques to scan a computing device for malware.

[0008] Fig. 3 is a flow diagram of an exemplary process for searching for malware updates and scanning a computing device for malware.

[0009] Fig. 4 is a flow diagram of an exemplary process for scanning a computing device for malware and updating components of a removable device.

[0010] Fig. 5 is a flow diagram of an exemplary process for authenticating removable device component updates.

[0011] Fig. 6 is a flow diagram of an exemplary process for secure file storage.

DETAILED DESCRIPTION

[0012] The devices and techniques described herein provide a trusted operating environment through which a computing device may be scanned for viruses and other forms of malicious software (malware), and disinfected of any such entities. The terms "virus" and "malware" are used interchangeably herein, and both refer generally to collections of computer code that are designed to infiltrate and/or damage a computer system without the owner's informed consent. Other examples of malware include trojan horses, worms, spyware and the like. The trusted operating environment is created by a trusted authority, such as a software or hardware manufacturer, and is then embodied on a device or a computer-readable medium that can be interfaced with a computing device. Unauthorized access to the trusted operating environment is prevented through the use of read-only media, authentication protocols, and microcontrollers that permit only trusted data to have access to the trusted operating environment. As used herein, the term authentication refers to any suitable method or protocol that may be implemented to verify the identity of an entity from which a communication or data file originates and to ensure that the communication or data file has not been tampered with or impermissibly altered by an unauthorized entity.

[0013] In one example, the trusted operating environment includes a trusted operating system and a trusted antivirus tool. A computing device can be booted using the trusted operating system and thus circumvent the problem of rootkits and other malware that may be hiding on the computing device. The trusted antivirus tool can then scan the computing device and be assured that its interaction with the trusted operating system will accurately reflect the state of the computing device.

Exemplary Architecture

[0014] Fig. 1 shows an architecture 100 that can implement the described processes and techniques. As part of architecture 100 is computing device 102. Although illustrated as a desktop PC, computing device 102 may be implemented as any of a variety of conventional computing devices including, for example, a server, a notebook or portable computer, a workstation, a mainframe computer, a

mobile communication device, a PDA, an entertainment device, a set-top box, an Internet appliance, a game console, and so forth.

[0015] Computing device 102 can include, but is not limited to, one or more processors 104, a memory 106, Input/Output (I/O) devices 108 (e.g., keyboard and mouse), and a system bus (not illustrated) that operatively couples various components including processor(s) 104 to the memory 106. The memory of computing device 102 includes computer-readable media in the form of volatile memory, such as Random Access Memory (RAM) and/or non-volatile memory, such as Read Only Memory (ROM) or flash RAM. Memory 106 typically includes data and/or program modules, such as operating system 110 and virus update package 112. As part of virus update package 112 are digital signature 114 and virus signature updates 116, which will be discussed in more detail below.

[0016] To implement a virus scanning process, a user loads removable data storage medium 118 into computing device 102 such that the computing device can read the removable medium. Removable data storage medium 118 is a data storage medium that can be interfaced with a computing device (e.g., by inserting it in a disk drive) and removed without having to disassemble the computing device. Although removable data storage medium 118 is illustrated here as a compact disk (CD) or a digital versatile disk (DVD), any suitable computer-readable removable data storage media may be employed, including other optical storage media, flash memory, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other removable storage medium which can be used to store data and which can be accessed by a computer. Further, removable data storage medium 118 may be a read-only medium such that the data and files contained thereon cannot be contaminated by malware from an infected source (e.g., computing device 102) that attempts to infect the storage medium.

[0017] The user then boots computing device 102 from removable data storage medium 118. Removable data storage medium 118 includes removable medium memory 120, which stores data and/or program modules and components for implementing the described processes. The terms "module" or "component" as used herein generally represent software, firmware, or a combination of software

and firmware. As part of the boot process, computing device 102 loads trusted operating system (OS) 122 from removable medium memory 120. Trusted antivirus tool 124 is then launched from removable medium memory 120, either automatically or by express action by the user.

[0018] Trusted antivirus tool 124 then loads virus signatures 126 that are stored on removable medium memory 120. Virus signatures 126 are a set of virus "fingerprints" that are used to identify viruses. An exemplary virus signature is a binary pattern of all or part of a virus's computer code. Trusted antivirus tool 124 also searches computing device 102 and identifies any virus signatures that are not currently stored on removable medium memory 120. In one example, trusted antivirus tool 124 locates virus update package 112, which contains virus signature updates 116, on memory 106. Virus signature updates 116 include one or more virus signatures, some of which may be different (e.g., they may identify more recently catalogued viruses) than any stored on removable data storage medium 118. Before utilizing virus signature updates 116 in a virus scan, trusted antivirus tool 124 verifies the authenticity of virus update package 112 using authentication tool 128.

[0019] To verify and/or authenticate virus update package 112, authentication tool 128 utilizes root certificate 130 from removable medium memory 120 to authenticate digital signature 114. Generally stated, digital signature 114 is a coded message or other piece of data that is compared with root certificate 130 to determine if virus update package 112 originates from a trusted authority and has not been tampered with. In one example, digital signature 114 is part of a digital certificate, such as an X.509 certificate, that is part of virus update package 112. If digital signature 114 is determined to be authentic, trusted antivirus tool 124 designates the virus signature updates as authenticated and loads the virus signature updates. This authentication method is presented for purposes of example only, and any suitable cryptographic, verification and/or authentication protocol may be utilized to verify that the virus signature updates have originated from a trusted authority.

[0020] Trusted antivirus tool 124 then runs on computing device 102 and scans computing device 102 for viruses using virus signatures 126 and any authenticated virus signature updates from virus update package 112. By interacting with trusted operating system 122 in the virus scan, a user can be assured that operating system calls made during the virus scan process will correctly return the requested files and give an accurate description of computing device 102.

[0021] Further to architecture 100, computing device 102 may use network(s) 132 to access remote resource 134. Network(s) 132 may include, but is not limited to, a Local Area Network (LAN), a Wide Area Network (WAN), and a Metropolitan Area Network (MAN). Remote resource 134 may be a web server, a server farm, a mainframe computer, a data center, or any other resource capable of storing and/or transmitting data. Thus, trusted antivirus tool 124 can utilize computing device 102 and network(s) 132 to access remote resource 134 and identify and download one or more virus signature updates that reside on the remote resource. Any virus signature updates that are identified on remote resource 134 would be authenticated, as discussed above.

[0022] Fig. 2 shows an exemplary architecture 200 that can implement the described processes and techniques and variations thereof. Architecture 200 includes computing device 102, introduced in Fig. 1. Also shown is removable device 202. Although removable device 202 is shown here as a universal serial bus (USB) device, any suitable removable and/or portable device may be utilized, such as a PC card, smartcard, Firewire device, and the like. In operation, removable device 202 is interfaced with computing device 102 and the computing device is booted from the removable device. As part of the boot process, computing device 102 accesses memory 204 on removable device 202 and loads trusted operating system 206. Memory 204 typically includes data, program modules and components for implementing the described processes that are immediately accessible to and/or presently operated on by microcontroller 216 and/or computing device 102.

[0023] Trusted antivirus tool 208 is then launched from memory 204 and the trusted antivirus tool loads virus signatures 210 from memory 204. Trusted

antivirus tool 208 can optionally locate and authenticate virus signature updates on computing device 102 using processes similar to those discussed above with respect to Fig. 1. Trusted antivirus tool 208 then proceeds to scan computing device 102 for viruses using the loaded virus signatures. By interacting with trusted operating system 206 that is running on computing device 102, trusted antivirus tool 208 can perform a thorough virus scan of computing device 102, including any storage devices (e.g., hard drives), the basic input/output system (BIOS), hardware, firmware, and the like. If any viruses are located, the viruses are removed from computing device 102. Computing device 102 can then be rebooted into a clean operating environment using its own internal operating system.

[0024] Removable device 202 has the ability to securely update its own internal components using update agent 214 and microcontroller 216. Microcontroller 216 is an integrated circuit or microprocessor that includes components necessary to control certain procedures and actions of removable device 202. Though not illustrated in Fig. 2, components of microcontroller 216 include one or more processors, one or more forms of memory (e.g., read-only memory and/or random access memory), input/output ports, timers, and the like.

[0025] Any components of removable device 202 may be updated, including virus signatures 210 and trusted antivirus tool 208. As part of the update process, microcontroller 216 can utilize update agent 214 to search computing device 102 for one or more removable device component updates. Microcontroller 216 can also utilize update agent 214 to access remote resource 134 via computing device 102 and network(s) 132 to obtain the desired component updates. If any component updates are located, authentication tool 212 can use any suitable authentication and/or cryptographic protocol to verify that the component updates originate from a trusted authority. If a component update is determined to be authentic (i.e., it passes the authentication process) by authentication tool 212, the component update is written to memory 204. Otherwise, microcontroller 216 prevents any unauthenticated and unauthorized data (e.g., component updates that have not passed the authentication process) from being written onto memory 216. Leveraging microcontroller 216 as a "gatekeeper" permits removable device 202 to

be dynamically updated while maintaining the trusted aspect of its internal components.

Exemplary Processes

[0026] Fig. 3 illustrates an exemplary process 300 for performing a virus scan. The process 300 is illustrated as a collection of blocks in a logical flow graph, which represents a sequence of operations that can be implemented in hardware, software, or a combination thereof. In the context of software, the blocks represent computer instructions that, when executed by one or more processors, perform the recited operations. While the processes below are discussed in terms of separate acts, this is not intended to be limiting, and the discussed acts may be combined in some examples. For discussion purposes, the process 300 is described with reference to architecture 100 shown in Fig. 1.

[0027] At 302, a removable data storage medium (such as removable data storage medium 118) is loaded into a computing device. At 304, the computing device is booted from a trusted operating environment on the removable data storage medium. As part of this boot process, a trusted operating system is loaded from the removable data storage medium onto the computing device. At 306, a trusted antivirus tool is launched from the removable data storage medium. Act 306 may be accomplished by running the trusted antivirus tool from the removable data storage medium, or loading the trusted antivirus tool onto the computing device and running the tool from computing device. At 308, the trusted antivirus tool searches the computing device and locates any virus signature updates. The virus signature updates may be part of a virus update package that includes the virus signature updates and a digital signature that can be used to authenticate the updates. The digital signature may be part of an authentication certificate, such as an X.509 certificate, that is stored as part of the virus update package or elsewhere on the computing device. The trusted antivirus tool may also utilize the computing device and a network to search a remote resource for virus signature updates. If any virus signature updates are located and/or identified, the updates are authenticated at 310. In one example, act 310 includes processing the digital signature to determine if the virus signature updates originate from a trusted authority.

[0028] At 312, the trusted antivirus tool loads any virus signatures that are stored on the removable data storage medium, along with any authenticated virus signature updates. At 314, the trusted antivirus tool interacts with the trusted operating system loaded on the computing device to scan the computing device for any viruses that match the loaded virus signatures. The trusted antivirus tool can perform a comprehensive scan of the computing device, including any storage devices (e.g., the hard drive), the BIOS, hardware, firmware, and the like. At 316, any viruses that are located are removed from the computing device. At 318, the computing device is rebooted into a clean internal operating environment. The internal operating environment includes an operating system that has been scanned and cleaned of any viruses or other malware using the techniques discussed herein.

[0029] Fig. 4 illustrates an exemplary process 400 for performing a virus scan. The process 400 is illustrated as a collection of blocks in a logical flow graph, which represents a sequence of operations that can be implemented in hardware, software, or a combination thereof. In the context of software, the blocks represent computer instructions that, when executed by one or more processors, perform the recited operations. For discussion purposes, the process 400 is described with reference to architecture 200 shown in Fig. 2.

[0030] At 402, a removable device (e.g., removable device 202) is interfaced with a computing device that a user wishes to scan for viruses. In one example, act 402 is accomplished by plugging the removable device into the appropriate port (e.g., a USB port) on the computing device. At 404, the computing device is booted from the removable device. As part of the boot process, a trusted operating system is loaded from the removable device onto the computing device. At 406, a trusted antivirus tool is launched from the removable device. Act 406 may be accomplished by running the trusted antivirus tool from the removable device, or loading the trusted antivirus tool onto the computing device and running the tool from computing device. As part of act 406, the trusted antivirus tool loads one or more virus signatures that are stored on the removable device. The removable device may also locate one or more virus signature updates that are stored on the computing device and/or a remote resource, authenticate the virus signature

updates, and load any authenticated virus signature updates. At 408, the trusted antivirus tool interacts with the trusted operating system loaded on the computing device to scan the computing device for any viruses. The antivirus tool identifies any viruses based on the virus signatures stored on the removable device, plus any authenticated virus signature updates from the computing device. Any viruses that are located are removed at 410.

[0031] At 412, the computing device is rebooted into a clean internal environment that includes an operating system that has been scanned and cleaned of viruses. The internal operating system is an operating system that resides on the computing device (i.e., it is not the operating system that was loaded from the removable device). At 414, the removable device then checks for updates to its internal components (e.g., virus signatures and the antivirus tool itself). The removable device may search the computing device for component updates, or may optionally access a remote resource to search for component updates. If any component updates are located, the component updates are authenticated at 416 using any suitable authentication and/or cryptographic process. If any component updates pass the authentication process, the component updates are installed on the removable device at 418. Although not shown here, the process may optionally return to act 404 and rescan the computing device for viruses using any virus signature updates (or other component updates) that were installed on the removable device at 418.

[0032] Thus, as illustrated, process 400 allows a virus scan to be performed on a computing device using a trusted operating system loaded from a removable device. Once the computing device has been scanned and disinfected of any viruses, the removable device can then search for any updates to its own internal components, authenticate any updates that are located, and load the authenticated updates. In some implementations, this process is achieved by operating the removable device in a read-only mode during most regular file system operations (e.g., during a virus scan), thus preventing viruses and other malware from infecting the removable device. During update operations, component update packages are transferred to the removable device and authenticated. A

microcontroller on the removable device controls the authentication process and checks the component update packages for signatures and/or certificates. If the microcontroller determines that the signature and/or certificate for a particular update package is valid, the microcontroller will allow the update package to be written to the removable device. Otherwise, the removable device will remain in a read-only state to maintain the integrity of its trusted operating environment.

[0033] Fig. 5 illustrates one example of a process for authenticating removable device component updates, as illustrated above in act 416 of Fig. 4. At 500, a digital signature that is associated with any removable device component updates is located and retrieved. As discussed above, this digital signature may be part of an overall authentication certificate. At 502, the digital signature is processed according to any suitable authentication and/or cryptographic protocol. In one example, the digital signature is verified with a root certificate that is stored on the removable device. At 504, it is determined if the digital signature has passed the authentication process. If the digital signature is determined to originate from a trusted authority (i.e., is designated as authentic), then the removable device component updates associated with the digital signature are written to the removable device at 506. If the digital signature fails the authentication process, then at 508, the removable device component updates are not written to the removable device. In one example, act 508 may include deleting from the computing device any removable device component updates that have failed the authentication process.

Secure Storage

[0034] Fig. 6 illustrates and exemplary process 600 for implementing a secure storage device. The process 600 is illustrated as a collection of blocks in a logical flow graph, which represents a sequence of operations that can be implemented in hardware, software, or a combination thereof. In the context of software, the blocks represent computer instructions that, when executed by one or more processors, perform the recited operations. For discussion purposes, the process 600 is described with reference to removable device 202, as shown in Fig. 2.

[0035] At 602, a removable device is interfaced with a computing device. In one example, act 602 is accomplished by plugging the removable device into the appropriate port (e.g., a USB port) on the computing device. At 604, at least one data file that a user wishes to copy and/or transfer to the removable device is located on the computing device. In one example, multiple files may be located, and thus the illustrated process would be performed for each file. The data file may be stored locally on the computing device or may be located on a remote resource that is accessible to the computing device. At 606, the data file is scanned for viruses with an antivirus tool that resides on the removable device. At 608, it is determined if the data file is infected with a virus or other malware. If any viruses/malware are detected in the data file, the data files are disinfected of the viruses using the antivirus tool at 610 (i.e., the viruses are removed from the files). If no viruses or other malware are detected in the file, the file is copied to the removable device at 616. At 612, it is determined if the infected file was successfully disinfected. If the file was not successfully disinfected, at 614 the file is prevented from being copied to the removable device. Optionally, the file may be rescanned with the antivirus tool in a further attempt to disinfect the file. If the file was successfully disinfected, then at 616, the disinfected data file is copied to the removable device. The removable device may then transfer the copied file to other devices without fear of spreading any viruses that may have infected to file. This is particularly useful in public terminal scenarios (e.g., libraries) where many different users have access to a particular device. In such situations, users may unwittingly access and download contaminated files onto the public terminal. Using the secure storage process, a user can ensure that any files downloaded from such a terminal are free from virus contamination.

Conclusion

[0036] Although the invention has been described in language specific to structural features and/or methodological steps, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or steps described. Rather, the specific features and steps are disclosed as preferred forms of implementing the claimed invention.

CLAIMS

1. A method comprising:

as part of a computing device boot process, loading a trusted operating system onto the computing device from a removable device (404), the removable device including the trusted operating system and an antivirus tool (202);

launching the antivirus tool from the removable device (406);

scanning the computing device with the antivirus tool (408);

searching for one or more removable device component updates (414);

if any removable device component updates are located, writing one or more of the removable device component updates to the removable device (416, 418).

2. A method as recited in claim 1, wherein launching the antivirus tool from the removable device comprises loading one or more virus signatures from the removable device.

3. A method as recited in claim 1, wherein launching the antivirus tool from the removable device comprises:

searching for one or more virus signature updates on the computing device;

if any virus signature updates are located, authenticating the virus signature updates; and

utilizing any authenticated virus signature updates to scan the computing device.

4. A method as recited in claim 1, wherein scanning the computing device with the antivirus tool comprises removing any viruses that are detected and rebooting the computing device using an operating system that is internal to the computing device.

5. A method as recited in claim 1, wherein the removable device component updates are virus signature updates.

6. A method as recited in claim 1, wherein the removable device component updates are antivirus tool updates.

7. A method as recited in claim 1, wherein the searching for one or more removable device component updates is performed at least in part by an update agent on the removable device.

8. A method as recited in claim 1, wherein writing one or more of the removable device component updates to the removable device comprises:

authenticating the removable device component updates; and

writing only the authenticated removable device component updates to the removable device.

9. A method as recited in claim 8, wherein the authentication is implemented by an authentication tool on the removable device.

10. A portable device comprising:

a computer-readable memory (204), the computer-readable memory comprising:

a trusted operating system component to boot an external computing device (206);

an antivirus tool component to scan the external computing device (208);

an update agent component to update one or more components of the portable device (214); and

a processor for controlling access to the components of the removable device (216).

11. A portable device as recited in claim 10, wherein the computer-readable memory further comprises one or more virus signatures for use by the antivirus tool.

12. A portable device as recited in claim 10, wherein the update agent is configured to search for portable device component updates.

13. A portable device as recited in claim 12, wherein the update agent is configured to search for the portable device component updates on at least one of the external computing device and a resource remote to the external computing device.

14. A portable device as recited in claim 12, wherein the component updates comprise one or more virus signature updates.

15. A portable device as recited in claim 12, wherein the component updates comprise one or more antivirus tool updates.

16. A portable device as recited in claim 12, wherein the computer-readable memory further comprises an authentication tool for authenticating any portable device component updates located by the update agent.

17. A method comprising:

interfacing a removable device with a computing device (602);

locating one or more data files on the computing device to be stored on the removable device (604);

scanning the data files with an antivirus tool that resides on the removable device (606); and

writing the data files to the removable device (616).

18. A method as recited in claim 17, wherein the scanning further comprises removing any viruses that are detected in the files.

19. A method as recited in claim 17, wherein the removable device comprises a computer-readable memory and a processor.

20. A method as recited in claim 17, wherein the removable device is a universal serial bus device.

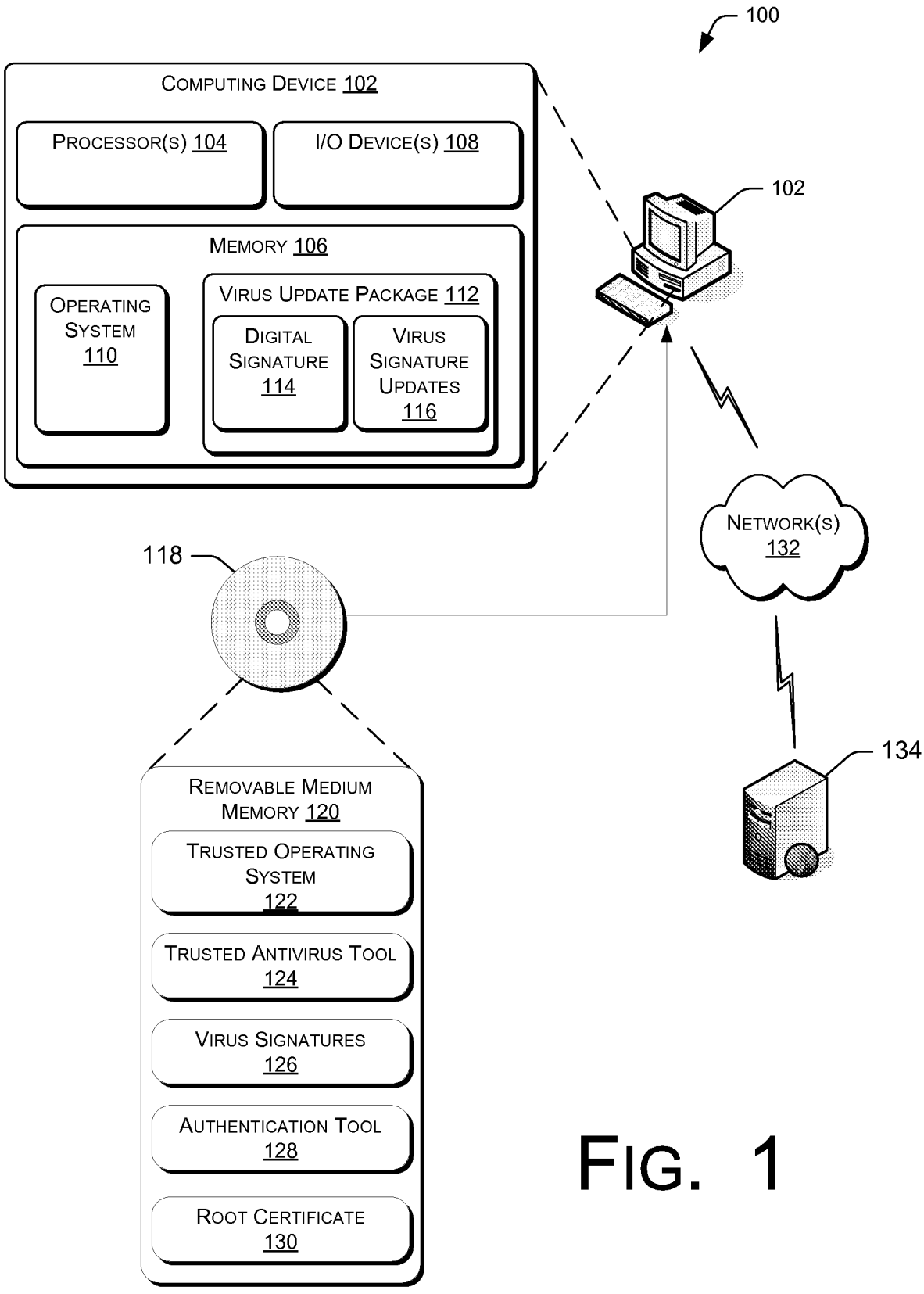
100
COMPUTING DEVICE 102
PROCESSOR(S) 104
I/O DEVICE(S) 108
MEMORY 106
OPERATING SYSTEM 110
VIRUS UPDATE PACKAGE 112
DIGITAL SIGNATURE 114
VIRUS SIGNATURE UPDATES 116
102
NETWORK(S) 132
118
134
REMOVABLE MEDIUM MEMORY 120
TRUSTED OPERATING SYSTEM 122
TRUSTED ANTIVIRUS TOOL 124
VIRUS SIGNATURES 126
AUTHENTICATION TOOL 128
ROOT CERTIFICATE 130

FIG. 1

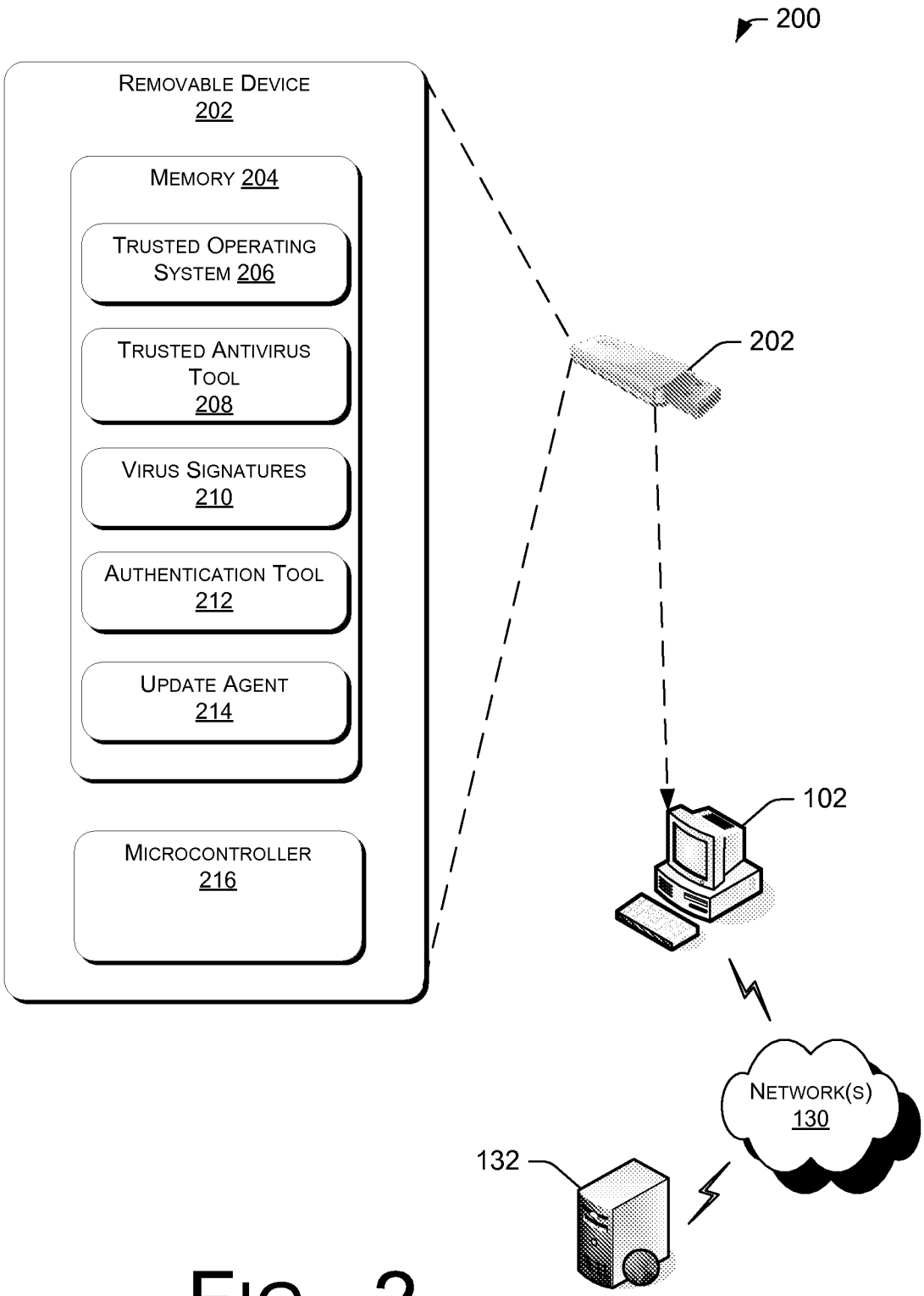
200
REMOVABLE DEVICE
202
MEMORY 204
TRUSTED OPERATING SYSTEM 206
TRUSTED ANTIVIRUS TOOL
208
VIRUS SIGNATURES
210
AUTHENTICATION TOOL
212
UPDATE AGENT
214
MICROCONTROLLER
216
202
102
NETWORK(S)
130
132

FIG. 2

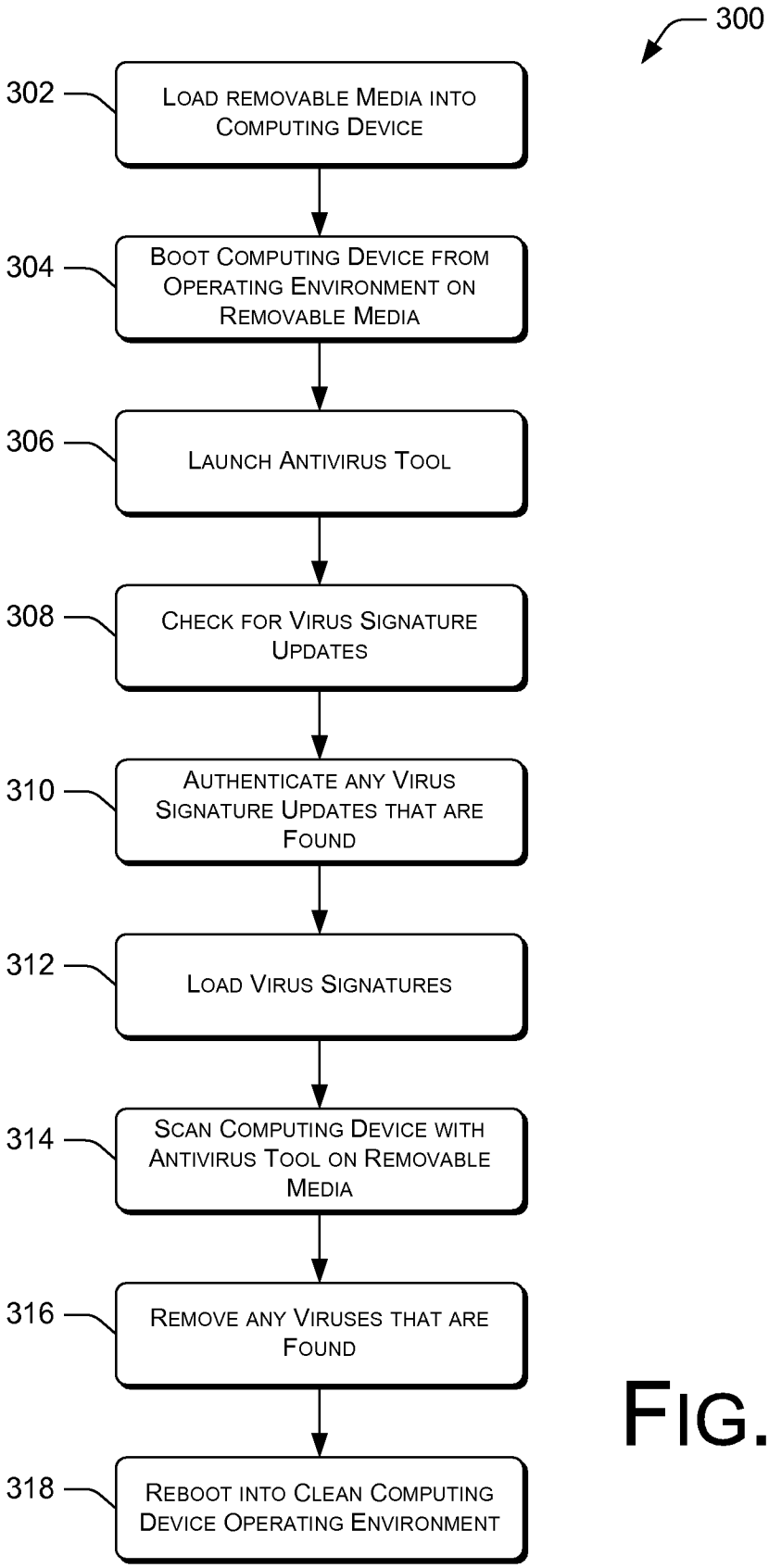
300
302
LOAD REMOVABLE MEDIA INTO COMPUTING DEVICE
304
BOOT COMPUTING DEVICE FROM OPERATING ENVIRONMENT ON REMOVABLE MEDIA
306
LAUNCH ANTIVIRUS TOOL
308
CHECK FOR VIRUS SIGNATURE UPDATES
310
AUTHENTICATE ANY VIRUS SIGNATURE UPDATES THAT ARE FOUND
312
LOAD VIRUS SIGNATURES
314
SCAN COMPUTING DEVICE WITH ANTIVIRUS TOOL ON REMOVABLE MEDIA
316
REMOVE ANY VIRUSES THAT ARE FOUND
318
REBOOT INTO CLEAN COMPUTING DEVICE OPERATING ENVIRONMENT

FIG. 3

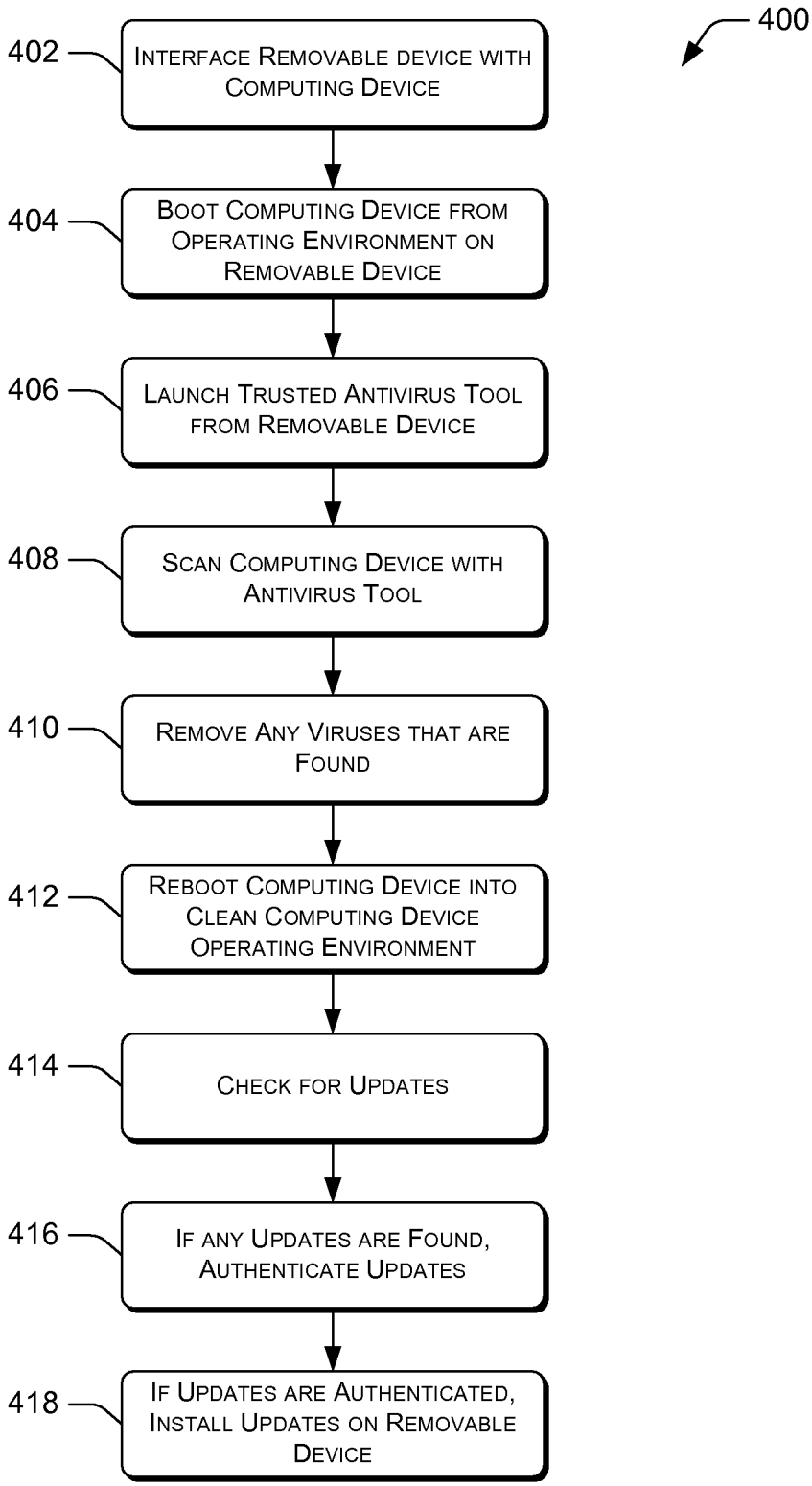
400
402
Interface Removable device with Computing Device
404
Boot Computing Device from Operating Environment on Removable Device
406
Launch Trusted Antivirus Tool from Removable Device
408
Scan Computing Device with Antivirus Tool
410
Remove Any Viruses that are Found
412
Reboot Computing Device into Clean Computing Device Operating Environment
414
Check for Updates
416
If any Updates are Found, Authenticate Updates
418
If Updates are Authenticated, Install Updates on Removable Device

FIG. 4

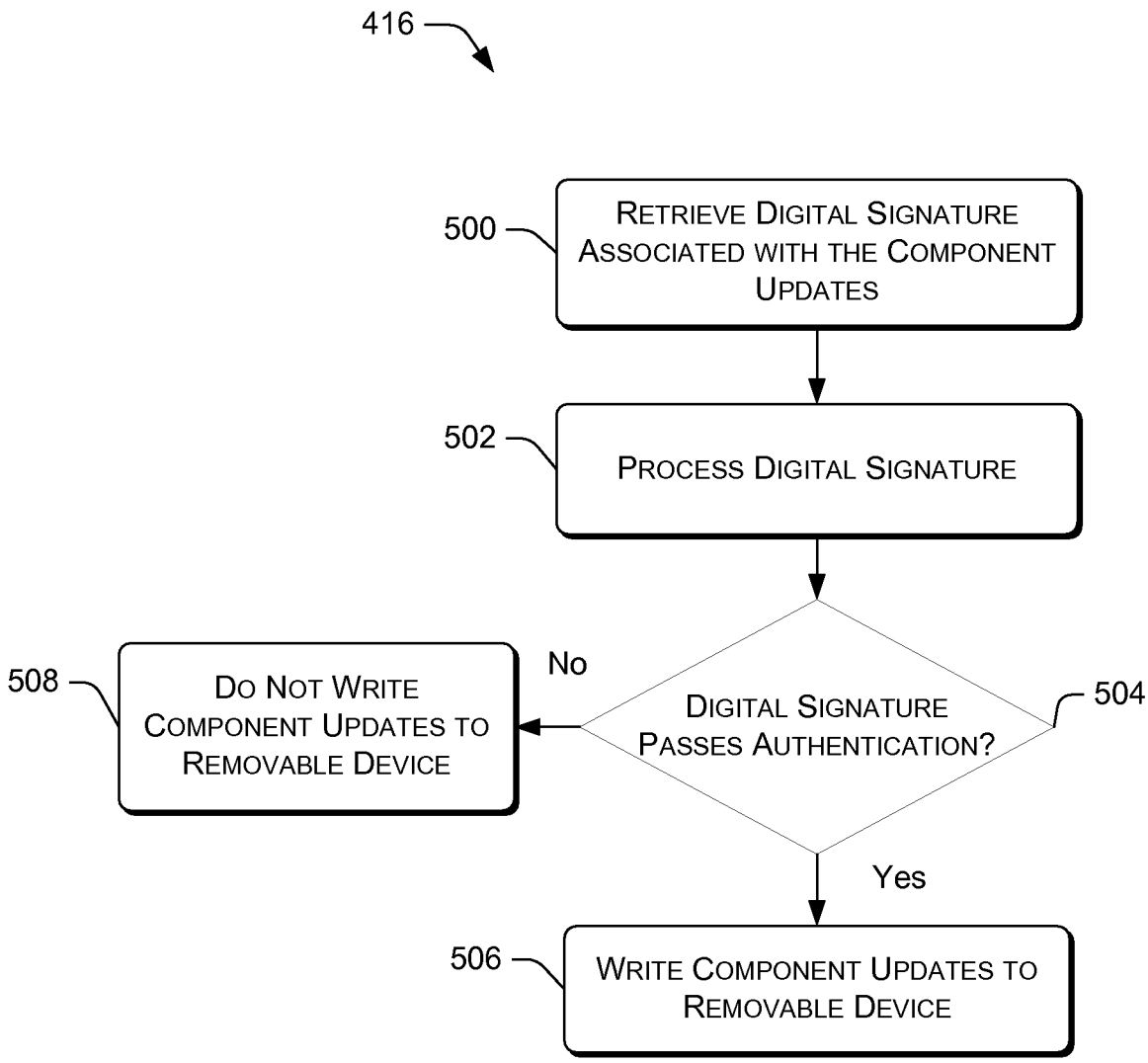
416
500
RETRIEVE DIGITAL SIGNATURE ASSOCIATED WITH THE COMPONENT UPDATES
502
PROCESS DIGITAL SIGNATURE
DIGITAL SIGNATURE PASSES AUTHENTICATION?
504
No
508
DO NOT WRITE COMPONENT UPDATES TO REMOVABLE DEVICE
Yes
506
WRITE COMPONENT UPDATES TO REMOVABLE DEVICE

FIG. 5

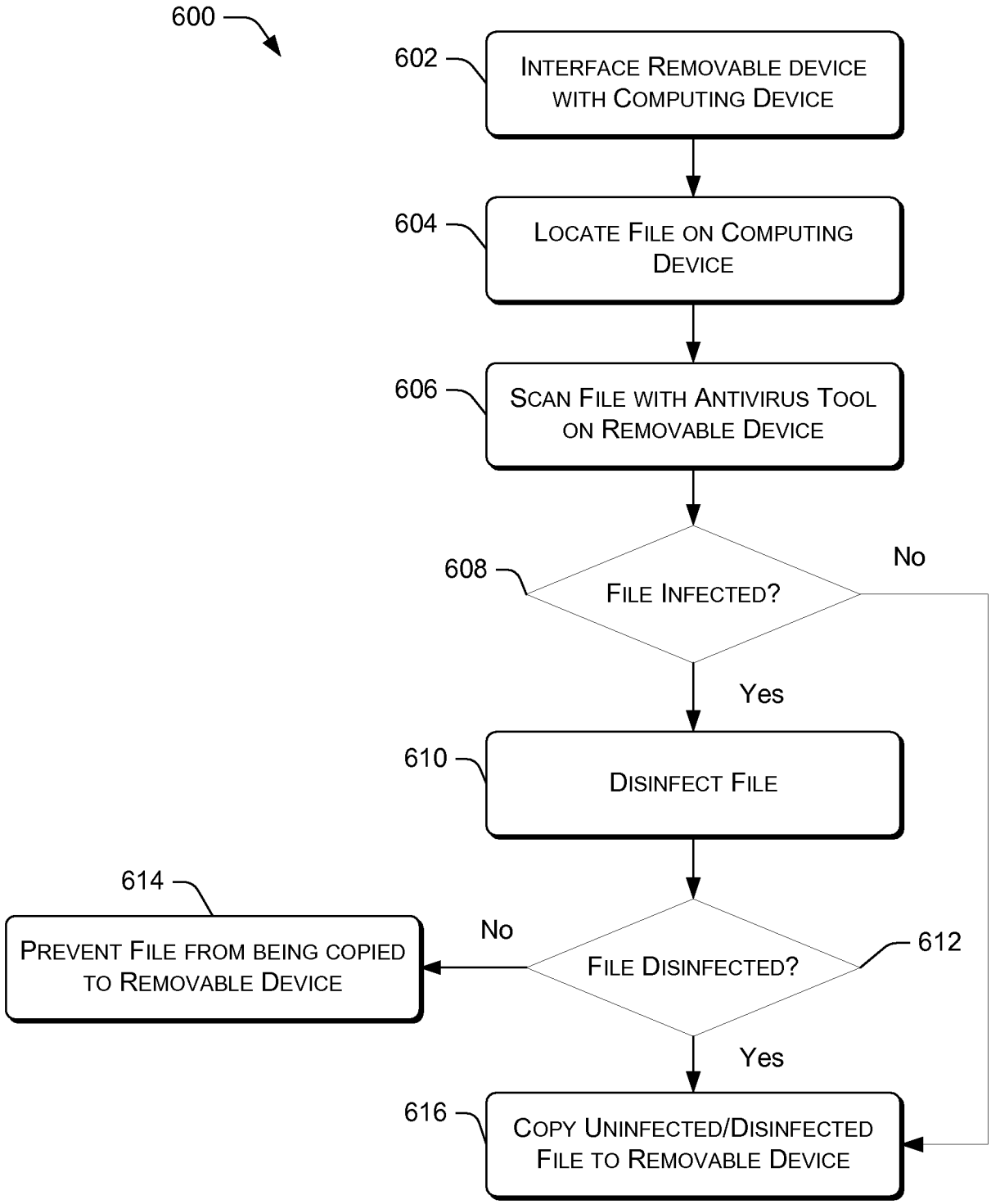
600
602
INTERFACE REMOVABLE DEVICE WITH COMPUTING DEVICE
604
LOCATE FILE ON COMPUTING DEVICE
606
SCAN FILE WITH ANTIVIRUS TOOL ON REMOVABLE DEVICE
608
FILE INFECTED?
No
Yes
610
DISINFECT FILE
614
PREVENT FILE FROM BEING COPIED TO REMOVABLE DEVICE
No
FILE DISINFECTED?
612
Yes
616
COPY UNINFECTED/DISINFECTED FILE TO REMOVABLE DEVICE

FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2008/062513

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/00(2006.01)i, G06F 21/22(2006.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 8 : G06F

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

Utility models and applications for Utility Models since 1975

Japanese Utility Models and application for Utility Models since 1975

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

eKIPASS(KIPO internal) "malware, boot*, operat*, update"

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y X	US 2005/0193188 A1 (EVAN S. HUANG) 01 September 2005 See the abstract, pages 2-6, and figures 9, 10, 13	1-16 17-20
Y	US 2007/0094654 A1 (MIHAI COSTEA) 26 April 2007 See the abstract, pages 3-5, and figures 2, 3	1-20
P,Y	US 2008/0052507 A1 (CHOW et al.) 28 February 2008 See the abstract and figures 3, 6, 8	1-20
P,Y	SY Dai et al. "MAPMon: A Host-Based Malware Detection tool", IEEE International Symposium on 13th Pacific Rim Dependable Computing, pp.349-356, 17 December 2007 See the abstract and chapters 2, 3	1-20

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

☒ See patent family annex.

* Special categories of cited documents:

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

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

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of 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
20 OCTOBER 2008 (20.10.2008)	**20 OCTOBER 2008 (20.10.2008)**

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 139 Seonsa-ro, Seo-gu, Daejeon 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

KYUNG, Youn Jeong

Telephone No. 82-42-481-8536

Form PCT/ISA/210 (second sheet) (July 2008)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2008/062513

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2005/0193188 A1	01.09.2005	WO 2005/091745 A2	06.10.2005
		WO 2005/091745 A3	06.10.2005
US 2007/0094654 A1	26.04.2007	None.	
US 2008/0052507 A1	28.02.2008	CN2859750 Y	17.01.2007
		DE10001672 A1	26.04.2001
		JP2001-118046 A	27.04.2001
		US 6854984 B1	15.02.2005
		US 6874044 B1	29.03.2005
		US 7069369 B2	27.06.2006
		US 7257714 B1	14.08.2007
		US 2006-0161725 A1	20.07.2006
		US 2007-0130436 A1	07.06.2007
		US 2007-0300028 A1	27.12.2007
		US 2007-0300029 A1	27.12.2007
		US 2007-0300030 A1	27.12.2007
		US 2008-0005580 A1	03.01.2008
		US 2008-0005581 A1	03.01.2008
		US 2008-0005582 A1	03.01.2008
		US 2008-0005583 A1	03.01.2008
		US 2008-0005584 A1	03.01.2008
		US 2008-0005585 A1	03.01.2008
		US 2008-0006927 A1	10.01.2008
		US 2008-0010465 A1	10.01.2008
		US 2008-0020641 A1	24.01.2008
		US 2008-0046635 A1	24.04.2008
		US 2008-0046633 A1	21.02.2008
		US 2008-0046634 A1	21.02.2008
		US 2008-0052439 A1	28.02.2008
		US 2008-0052452 A1	28.02.2008
		US 2008-0059680 A1	06.03.2008
		US 2008-0093720 A1	24.04.2008
		US 2008-0094807 A1	24.04.2008
		US 2008-0082813 A1	03.04.2008