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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 March 2017 (30.03.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/052888 A1

(51) International Patent Classification:
G06F 21/60 (2013.01) *G06F 21/78* (2013.01)

(21) International Application Number:
PCT/US2016/048104

(22) International Filing Date:
23 August 2016 (23.08.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
14/866,765 25 September 2015 (25.09.2015) US

(71) Applicant: **MCAFEE, INC.** [US/US]; 2821 Mission College Boulevard, Santa Clara, California 95054-1838 (US).

(72) Inventor: **COCHIN, Cedric**; 35 SW 68th Ave, Portland, Oregon 97225 (US).

(74) Agent: **PEMBERTON, John D.**; Patent Capital Group, c/o CPA Global, 900 Second Avenue South, Suite 600, Minneapolis, Minnesota 55402 (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, 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, ST, 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).

[Continued on next page]

(54) Title: ENFORCEMENT OF FILE CHARACTERISTICS

(57) Abstract: Particular embodiments described herein provide for an electronic device that can be configured to determine a file characteristic for a characteristic of a file, determine that the file has been modified to create a new file, determine a new characteristic for the characteristic of the new file, and create a security event if the new file characteristic does not match the file characteristic.

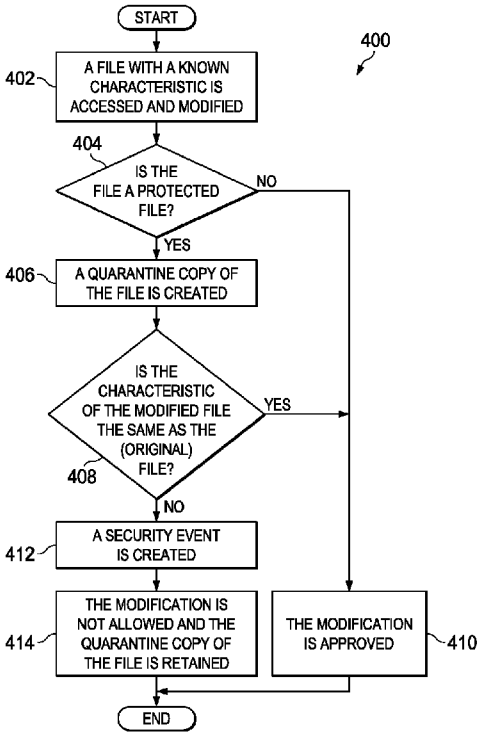

FIG. 4

WO 2017/052888 A1

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))*

Published:

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

ENFORCEMENT OF FILE CHARACTERISTICS

CROSS REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of priority to U.S. Nonprovisional (Utility) Patent Application No. 14/866,765 filed 25 September 2015 entitled "ENFORCEMENT OF FILE CHARACTERISTICS", which is incorporated herein by reference in its entirety.

TECHNICAL FIELD

[0002] This disclosure relates in general to the field of information security, and more particularly, to enforcement of file characteristics.

BACKGROUND

[0003] The field of network security has become increasingly important in today's society. The Internet has enabled interconnection of different computer networks all over the world. In particular, the Internet provides a medium for exchanging data between different users connected to different computer networks via various types of client devices. While the use of the Internet has transformed business and personal communications, it has also been used as a vehicle for malicious operators to gain unauthorized access to computers and computer networks and for intentional or inadvertent disclosure of sensitive information.

[0004] Malicious software ("malware") that infects a host computer may be able to perform any number of malicious actions, such as stealing sensitive information from a business or individual associated with the host computer, propagating to other host computers, and/or assisting with distributed denial of service attacks, sending out spam or malicious emails from the host computer, etc. Hence, significant administrative challenges remain for protecting computers and computer networks from malicious and inadvertent exploitation by malicious software.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] To provide a more complete understanding of the present disclosure and features and advantages thereof, reference is made to the following description, taken in conjunction with the accompanying figures, wherein like reference numerals represent like parts, in which:

[0006] FIGURE 1 is a simplified block diagram of a communication system for enforcement of file characteristics in accordance with an embodiment of the present disclosure;

[0007] FIGURE 2 is a simplified flowchart illustrating potential operations that may be associated with the communication system in accordance with an embodiment;

[0008] FIGURE 3 is a simplified flowchart illustrating potential operations that may be associated with the communication system in accordance with an embodiment;

[0009] FIGURE 4 is a simplified flowchart illustrating potential operations that may be associated with the communication system in accordance with an embodiment;

[0010] FIGURE 5 is a simplified flowchart illustrating potential operations that may be associated with the communication system in accordance with an embodiment;

[0011] FIGURE 6 is a block diagram illustrating an example computing system that is arranged in a point-to-point configuration in accordance with an embodiment;

[0012] FIGURE 7 is a simplified block diagram associated with an example ARM ecosystem system on chip (SOC) of the present disclosure; and

[0013] FIGURE 8 is a block diagram illustrating an example processor core in accordance with an embodiment.

[0014] The FIGURES of the drawings are not necessarily drawn to scale, as their dimensions can be varied considerably without departing from the scope of the present disclosure.

DETAILED DESCRIPTION OF EXAMPLE EMBODIMENTS

EXAMPLE EMBODIMENTS

[0015] FIGURE 1 is a simplified block diagram of a communication system 100 for file type enforcement of file characteristics in accordance with an embodiment of the present disclosure. As illustrated in FIGURE 1, an embodiment of communication system 100 can include electronic device 102, a server 104, and a cloud 106. Electronic device 102 can include a

processor 108a, memory 110a, one or more files 112a-112c, a file type module 114, and a security module 120. Memory can include a file type database 116. Each file 112a-112c can include a file type 118a-118c respectively. Server 104 can include a processor 108b and memory 110b. Memory 110b can include file type database 116. Cloud 106 can include a processor 108c and memory 110c. Memory 110c can include file type database 116. Electronic device 102, server 104, and cloud 106 may be in communication using network 122. In an example, malicious device 124 can attempt to infect electronic device 102 with ransomware 126.

[0016] In an example, communication system 100 can be configured to determine a file characteristic for a file (e.g., file type, attributes, metadata, etc.) and when the file is modified, analyze the file to determine if one or more of the file characteristics had changed. If the file characteristics had changed or a specific file characteristic, or a predefined group of characteristics, then the change may be an indication of malicious activity and a security event (e.g., scanning or otherwise analyzing the system for malware, not allowing the modification of the file, etc.) can be created. In a specific example, communication system 100 can be configured to determine a type for a file, determine that the file has been modified to create a new file, determine a new file type for the new file, and create a security event if the new file type does not match the file type. In an example, the security event can including analyzing or scanning the system for malware using security module 120. Also, communication system 100 can be configured to determine if the file was modified or created by a trusted application and if the file was not modified or created by a trusted application, then file type module 144 can analyze the file to determine the file type. The file type can be stored in a secure area of memory, for example, file type database 116 may be in a secured area of memory 110a. Before the file is created or modified, a backup of the file can be created and if the new file type does not match the (original) file type, then the modification is not allowed.

[0017] Elements of FIGURE 1 may be coupled to one another through one or more interfaces employing any suitable connections (wired or wireless), which provide viable pathways for network (e.g., network 122) communications. Additionally, any one or more of these elements of FIGURE 1 may be combined or removed from the architecture based on particular configuration needs. Communication system 100 may include a configuration capable of transmission control protocol/Internet protocol (TCP/IP) communications for the transmission or reception of packets in a network. Communication system 100 may also operate in

conjunction with a user datagram protocol/IP (UDP/IP) or any other suitable protocol where appropriate and based on particular needs.

[0018] For purposes of illustrating certain example techniques of communication system 100, it is important to understand the communications that may be traversing the network environment. The following foundational information may be viewed as a basis from which the present disclosure may be properly explained.

[0019] Ransomware (e.g. ransomware 126) is a type of malware that restricts access to a computer system that it infects and demands a ransom paid to the creator(s) of the malware in order for the restriction to be removed. Some forms of ransomware encrypt files on the system's hard drive, while some may simply lock the system and display messages intended to coax the user into paying. Ransomware typically propagates as a trojan like a conventional computer worm, entering a system through, for example, a downloaded file or a vulnerability in a network service. The ransomware will then run a payload such as one that will begin to encrypt personal files on the hard drive. More sophisticated ransomware may hybrid-encrypt the victim's documents with a random symmetric key and a fixed public key. The malware author is the only party that knows the needed private decryption key. CryptoLocker is one of the most prevalent ransomware.

[0020] Current security solutions (e.g., antivirus solutions, malware detection systems, etc.) often do not address the problem of files encrypted by ransomware. While some security solutions may detect the ransomware itself, they have no direct mechanism to protect the files, especially documents that the ransomware may encrypt. Some security solutions may attempt to restore the encrypted files as part of their malware repair but this is not possible in the case of an encryption using a private key that is not stored on the infected endpoint.

[0021] Some aspect of ransomware can be addressed with current security solutions upon detection of the malicious file. Upon detection, some security solutions can trigger a repair process and remove any artefact of the ransomware, including the simple lock that prevented the proper usage of the computer. However when files have been encrypted and the private key needed for decryption is not present on the infected device, the security solution cannot restore the files that have been encrypted by the ransomware. As a result, the contents of files such as documents and images, (e.g., Microsoft® Office® files, images, PDFs, etc), are lost.

[0022] Whitelisting and application control solutions, while protecting executables and key operating system components, including configuration files, from malicious modifications, do not offer protection for content on the system. The content is either locked, preventing any modification to be made, or the access to the content is restricted to a list of whitelisted applications. Whitelisting and application control solutions, while partially successful in highly restricted enterprise environments are mostly not applicable on ever changing consumer devices. What is needed is a system and method to identify and protect against ransomware.

[0023] A communication system for file type enforcement, as outlined in FIGURE 1 can resolve these issues (and others). Communication system 100 may be configured to store the file type of a file object as an external attribute and monitor all modifications made to the file to ensure that the modification doesn't result into a change of file type. By monitoring file and their associated file format and malicious and destructive changes to the file may be prevented. More specifically, communication system can be configured to prevent files from being modified by the ransomware in the first place.

[0024] A file format is a standard way that information is encoded for storage in a computer file. The file format specifies how bits are used to encode information in a digital storage medium. A typical approach to identify a file format is to use information regarding the format stored inside the file itself, either information meant for this purpose or binary strings that happen to always be in specific locations in files of some formats. File type module 114 can be configured to recognize files without external attributes and can include a file format recognition library to identify file types. Security module 120 can be configured to apply appropriate security rules.

[0025] Typically, when ransomware encrypts files, it do not maintain the original file format. For example, a proper JPG file, after encryption by the ransomware, will not be a JPG file anymore, but an encrypted blob of data that file format libraries would either identify as encrypted data or fail to recognize. Security module 120 can be configured to monitor file events and when a protected file is accessed, a quarantine copy of the file can be made. If the file is modified, upon completion of the modification, the file format can be identified by file type module 114. If the resulting file format is identical to or the same as the stored file format, the modification is approved. However, if the resulting file format is different than the stored file format, the modification is flagged and reverted to the original file format using the quarantine

copy of the file. The flagged modification can be used to alert a security module that malicious activity may have taken place and the system can be analyzed for malware.

[0026] Turning to the infrastructure of FIGURE 1, communication system 100 in accordance with an example embodiment is shown. Generally, communication system 100 can be implemented in any type or topology of networks. Network 122 represents a series of points or nodes of interconnected communication paths for receiving and transmitting packets of information that propagate through communication system 100. Network 122 offers a communicative interface between nodes, and may be configured as any local area network (LAN), virtual local area network (VLAN), wide area network (WAN), wireless local area network (WLAN), metropolitan area network (MAN), Intranet, Extranet, virtual private network (VPN), and any other appropriate architecture or system that facilitates communications in a network environment, or any suitable combination thereof, including wired and/or wireless communication.

[0027] In communication system 100, network traffic, which is inclusive of packets, frames, signals, data, etc., can be sent and received according to any suitable communication messaging protocols. Suitable communication messaging protocols can include a multi-layered scheme such as Open Systems Interconnection (OSI) model, or any derivations or variants thereof (e.g., Transmission Control Protocol/Internet Protocol (TCP/IP), user datagram protocol/IP (UDP/IP)). Additionally, radio signal communications over a cellular network may also be provided in communication system 100. Suitable interfaces and infrastructure may be provided to enable communication with the cellular network.

[0028] The term "packet" as used herein, refers to a unit of data that can be routed between a source node and a destination node on a packet switched network. A packet includes a source network address and a destination network address. These network addresses can be Internet Protocol (IP) addresses in a TCP/IP messaging protocol. The term "data" as used herein, refers to any type of binary, numeric, voice, video, textual, or script data, or any type of source or object code, or any other suitable information in any appropriate format that may be communicated from one point to another in electronic devices and/or networks. Additionally, messages, requests, responses, and queries are forms of network traffic, and therefore, may comprise packets, frames, signals, data, etc.

[0029] In an example implementation, electronic device 102, server 104, and cloud 106 are network elements, which are meant to encompass network appliances, servers, routers, switches, gateways, bridges, load balancers, processors, modules, or any other suitable device, component, element, or object operable to exchange information in a network environment. Network elements may include any suitable hardware, software, components, modules, or objects that facilitate the operations thereof, as well as suitable interfaces for receiving, transmitting, and/or otherwise communicating data or information in a network environment. This may be inclusive of appropriate algorithms and communication protocols that allow for the effective exchange of data or information.

[0030] In regards to the internal structure associated with communication system 100, each of electronic device 102, server 104, and cloud 106 can include memory elements (e.g., memory 110a-110c) for storing information to be used in the operations outlined herein. Each of electronic device 102, server 104, and cloud 106 may keep information in any suitable memory element (e.g., random access memory (RAM), read-only memory (ROM), erasable programmable ROM (EPROM), electrically erasable programmable ROM (EEPROM), application specific integrated circuit (ASIC), etc.), software, hardware, firmware, or in any other suitable component, device, element, or object where appropriate and based on particular needs. Any of the memory items discussed herein should be construed as being encompassed within the broad term 'memory element.' Moreover, the information being used, tracked, sent, or received in communication system 100 could be provided in any database, register, queue, table, cache, control list, or other storage structure, all of which can be referenced at any suitable timeframe. Any such storage options may also be included within the broad term 'memory element' as used herein.

[0031] In certain example implementations, the functions outlined herein may be implemented by logic encoded in one or more tangible media (e.g., embedded logic provided in an ASIC, digital signal processor (DSP) instructions, software (potentially inclusive of object code and source code) to be executed by a processor, or other similar machine, etc.), which may be inclusive of non-transitory computer-readable media. In some of these instances, memory elements can store data used for the operations described herein. This includes the memory elements being able to store software, logic, code, or processor instructions that are executed to carry out the activities described herein.

[0032] In an example implementation, network elements of communication system 100, such as electronic device 102, server 104, and cloud 106 may include software modules (e.g., file type module 114 and security module 120) to achieve, or to foster, operations as outlined herein. These modules may be suitably combined in any appropriate manner, which may be based on particular configuration and/or provisioning needs. In example embodiments, such operations may be carried out by hardware, implemented externally to these elements, or included in some other network device to achieve the intended functionality. Furthermore, the modules can be implemented as software, hardware, firmware, or any suitable combination thereof. These elements may also include software (or reciprocating software) that can coordinate with other network elements in order to achieve the operations, as outlined herein.

[0033] Additionally, each of electronic device 102, server 104, and cloud 106 may include a processor (e.g., processor 108a-108c) that can execute software or an algorithm to perform activities as discussed herein. A processor can execute any type of instructions associated with the data to achieve the operations detailed herein. In one example, the processors could transform an element or an article (e.g., data) from one state or thing to another state or thing. In another example, the activities outlined herein may be implemented with fixed logic or programmable logic (e.g., software/computer instructions executed by a processor) and the elements identified herein could be some type of a programmable processor, programmable digital logic (e.g., a field programmable gate array (FPGA), an EPROM, an EEPROM) or an ASIC that includes digital logic, software, code, electronic instructions, or any suitable combination thereof. Any of the potential processing elements, modules, and machines described herein should be construed as being encompassed within the broad term 'processor.'

[0034] Electronic device 102 can be a network element and include, for example, desktop computers, laptop computers, mobile devices, personal digital assistants, smartphones, tablets, or other similar devices. Server 104 can be a network element such as a server or virtual server and can be associated with clients, customers, endpoints, or end users wishing to initiate a communication in communication system 100 via some network (e.g., network 122). The term 'server' is inclusive of devices used to serve the requests of clients and/or perform some computational task on behalf of clients within communication system 100. Although file type module 114 and security module 120 are represented in FIGURE 1 as being located in electronic device 102, this is for illustrative purposes only. File type module 114 and security module 120

could be combined or separated in any suitable configuration. Furthermore, file type module 114 and security module 120 could be integrated with or distributed in another network accessible by electronic device 102. Cloud 106 is configured to provide cloud services to electronic device 102. Cloud services may generally be defined as the use of computing resources that are delivered as a service over a network, such as the Internet. Typically, compute, storage, and network resources are offered in a cloud infrastructure, effectively shifting the workload from a local network to the cloud network.

[0035] Turning to FIGURE 2, FIGURE 2 is an example flowchart illustrating possible operations of a flow 200 that may be associated with file type enforcement, in accordance with an embodiment. In an embodiment, one or more operations of flow 200 may be performed by file type module 114 and security module 120. At 202, a file type for a file is determined. For example, file type module 114 may determine file type 118a for file 112a. At 204, the determined file type is stored in a protected area of memory. For example, file type 118a for file 112a may be stored in file type database 116, in electronic device 102, server 104, and/or cloud 108.

[0036] Turning to FIGURE 3, FIGURE 3 is an example flowchart illustrating possible operations of a flow 300 that may be associated with file type enforcement, in accordance with an embodiment. In an embodiment, one or more operations of flow 300 may be performed by file type module 114 and security module 120. At 302 a file is created or stored in memory. At 304, the system determines if the file was created or stored by a trusted operation. For example, security module 120 may determine whether or not the file was created by a trusted application. The trusted operation may be from a trusted program or process. If the file was created by a trusted operation, then the file type is stored, as in 306. If the file was not created or stored by a trusted operation, then a file type module (e.g., file type module 114) determines a file type of the file, as in 308. If the operation is not a trusted operation, then the operation may be malicious and could be attempting to mask or hide the file type. At 306, the file type is stored.

[0037] Turning to FIGURE 4, FIGURE 4 is an example flowchart illustrating possible operations of a flow 400 that may be associated with file type enforcement, in accordance with an embodiment. In an embodiment, one or more operations of flow 400 may be performed by file type module 114 and security module 120. At 402, a file is accessed and modified. At 404, the system determines if the file is a protected file. If the file is not a protected file, then the

modification is approved, as in 410. If the file is a protected file, then a quarantine copy of the file is created, as in 406. At 408, the system determines if the format of type of the modified file is the same as the (original) file. If the format or type of the modified file is the same as the (original) file, then the modification is approved. If the format or type of the modified file is not the same as the (original) file, then a security event is created as in 412. At 414, the modification is not allowed and the quarantine copy of the file is retained. If ransomware attempts to modify a file and changes the file format, then the system can determine that the modified file is not the same file type as the original file and security module can analyze the system for malware. .

[0038] Turning to FIGURE 5, FIGURE 5 is an example flowchart illustrating possible operations of a flow 500 that may be associated with file type enforcement, in accordance with an embodiment. In an embodiment, one or more operations of flow 500 may be performed by file type module 114 and security module 120. At 502 a file type for a file is determined. For example, the extension of the file may be used to determine the file type. At 504, a file type definition for the file is determined. For example, the file may be analyzed by file type module 114 to determine the file type. At 506, the determined file type is compared to a stored file type definition for the file. At 508, the system determines if the file type matches the stored file type. If the file type matches the stored file type, then the file is classified as trusted or benign, as in 510. If the file type does not match the stored file type, then a security event is crated, as in 512.

[0039] Turning to FIGURE 6, FIGURE 6 illustrates a computing system 600 that is arranged in a point-to-point (PtP) configuration according to an embodiment. In particular, FIGURE 6 shows a system where processors, memory, and input/output devices are interconnected by a number of point-to-point interfaces. Generally, one or more of the network elements of communication system 100 may be configured in the same or similar manner as computing system 600.

[0040] As illustrated in FIGURE 6, system 600 may include several processors, of which only two, processors 670 and 680, are shown for clarity. While two processors 670 and 680 are shown, it is to be understood that an embodiment of system 600 may also include only one such processor. Processors 670 and 680 may each include a set of cores (i.e., processor cores 674A and 674B and processor cores 684A and 684B) to execute multiple threads of a program. The cores may be configured to execute instruction code in a manner similar to that discussed above with reference to FIGURES 1-5. Each processor 670, 680 may include at least one shared cache

671, 681. Shared caches 671, 681 may store data (e.g., instructions) that are utilized by one or more components of processors 670, 680, such as processor cores 674 and 684.

[0041] Processors 670 and 680 may also each include integrated memory controller logic (MC) 672 and 682 to communicate with memory elements 632 and 634. Memory elements 632 and/or 634 may store various data used by processors 670 and 680. In alternative embodiments, memory controller logic 672 and 682 may be discreet logic separate from processors 670 and 680.

[0042] Processors 670 and 680 may be any type of processor and may exchange data via a point-to-point (PtP) interface 650 using point-to-point interface circuits 678 and 688, respectively. Processors 670 and 680 may each exchange data with a chipset 690 via individual point-to-point interfaces 652 and 654 using point-to-point interface circuits 676, 686, 694, and 698. Chipset 690 may also exchange data with a high-performance graphics circuit 638 via a high-performance graphics interface 639, using an interface circuit 692, which could be a PtP interface circuit. In alternative embodiments, any or all of the PtP links illustrated in FIGURE 6 could be implemented as a multi-drop bus rather than a PtP link.

[0043] Chipset 690 may be in communication with a bus 620 via an interface circuit 696. Bus 620 may have one or more devices that communicate over it, such as a bus bridge 618 and I/O devices 616. Via a bus 610, bus bridge 618 may be in communication with other devices such as a keyboard/mouse 612 (or other input devices such as a touch screen, trackball, etc.), communication devices 626 (such as modems, network interface devices, or other types of communication devices that may communicate through a computer network 660), audio I/O devices 614, and/or a data storage device 628. Data storage device 628 may store code 630, which may be executed by processors 670 and/or 680. In alternative embodiments, any portions of the bus architectures could be implemented with one or more PtP links.

[0044] The computer system depicted in FIGURE 6 is a schematic illustration of an embodiment of a computing system that may be utilized to implement various embodiments discussed herein. It will be appreciated that various components of the system depicted in FIGURE 6 may be combined in a system-on-a-chip (SoC) architecture or in any other suitable configuration. For example, embodiments disclosed herein can be incorporated into systems including mobile devices such as smart cellular telephones, tablet computers, personal digital

assistants, portable gaming devices, etc. It will be appreciated that these mobile devices may be provided with SoC architectures in at least some embodiments.

[0045] Turning to FIGURE 7, FIGURE 7 is a simplified block diagram associated with an example ARM ecosystem SOC 700 of the present disclosure. At least one example implementation of the present disclosure can include the detection of malicious strings features discussed herein and an ARM component. For example, the example of FIGURE 7 can be associated with any ARM core (e.g., A-7, A-15, etc.). Further, the architecture can be part of any type of tablet, smartphone (inclusive of Android™ phones, iPhones™), iPad™, Google Nexus™, Microsoft Surface™, personal computer, server, video processing components, laptop computer (inclusive of any type of notebook), Ultrabook™ system, any type of touch-enabled input device, etc.

[0046] In this example of FIGURE 7, ARM ecosystem SOC 700 may include multiple cores 706-707, an L2 cache control 708, a bus interface unit 709, an L2 cache 710, a graphics processing unit (GPU) 715, an interconnect 702, a video codec 720, and a liquid crystal display (LCD) I/F 725, which may be associated with mobile industry processor interface (MIPI)/ high-definition multimedia interface (HDMI) links that couple to an LCD.

[0047] ARM ecosystem SOC 700 may also include a subscriber identity module (SIM) I/F 730, a boot read-only memory (ROM) 735, a synchronous dynamic random access memory (SDRAM) controller 740, a flash controller 745, a serial peripheral interface (SPI) master 750, a suitable power control 755, a dynamic RAM (DRAM) 760, and flash 765. In addition, one or more example embodiments include one or more communication capabilities, interfaces, and features such as instances of Bluetooth™ 770, a 3G modem 775, a global positioning system (GPS) 780, and an 802.11 Wi-Fi 785.

[0048] In operation, the example of FIGURE 7 can offer processing capabilities, along with relatively low power consumption to enable computing of various types (e.g., mobile computing, high-end digital home, servers, wireless infrastructure, etc.). In addition, such an architecture can enable any number of software applications (e.g., Android™, Adobe® Flash® Player, Java Platform Standard Edition (Java SE), JavaFX, Linux, Microsoft Windows Embedded, Symbian and Ubuntu, etc.). In at least one example embodiment, the core processor may implement an out-of-order superscalar pipeline with a coupled low-latency level-2 cache.

[0049] Turning to FIGURE 8, FIGURE 8 illustrates a processor core 800 according to an embodiment. Processor core 800 may be the core for any type of processor, such as a micro-processor, an embedded processor, a digital signal processor (DSP), a network processor, or other device to execute code. Although only one processor core 800 is illustrated in Figure 8, a processor may alternatively include more than one of the processor core 800 illustrated in Figure 8. For example, processor core 800 represents one example embodiment of processors cores 674a, 674b, 684a, and 684b shown and described with reference to processors 670 and 680 of FIGURE 6. Processor core 800 may be a single-threaded core or, for at least one embodiment, processor core 800 may be multithreaded in that it may include more than one hardware thread context (or "logical processor") per core.

[0050] FIGURE 8 also illustrates a memory 802 coupled to processor core 800 in accordance with an embodiment. Memory 802 may be any of a wide variety of memories (including various layers of memory hierarchy) as are known or otherwise available to those of skill in the art. Memory 802 may include code 804, which may be one or more instructions, to be executed by processor core 800. Processor core 800 can follow a program sequence of instructions indicated by code 804. Each instruction enters a front-end logic 806 and is processed by one or more decoders 808. The decoder may generate, as its output, a micro operation such as a fixed width micro operation in a predefined format, or may generate other instructions, microinstructions, or control signals that reflect the original code instruction. Front-end logic 806 also includes register renaming logic 810 and scheduling logic 812, which generally allocate resources and queue the operation corresponding to the instruction for execution.

[0051] Processor core 800 can also include execution logic 814 having a set of execution units 816-1 through 816-N. Some embodiments may include a number of execution units dedicated to specific functions or sets of functions. Other embodiments may include only one execution unit or one execution unit that can perform a particular function. Execution logic 814 performs the operations specified by code instructions.

[0052] After completion of execution of the operations specified by the code instructions, back-end logic 818 can retire the instructions of code 804. In one embodiment, processor core 800 allows out of order execution but requires in order retirement of instructions. Retirement logic 820 may take a variety of known forms (e.g., re-order buffers or the like). In this manner, processor core 800 is transformed during execution of code 804, at least in terms

of the output generated by the decoder, hardware registers and tables utilized by register renaming logic 810, and any registers (not shown) modified by execution logic 814.

[0053] Although not illustrated in FIGURE 8, a processor may include other elements on a chip with processor core 800, at least some of which were shown and described herein with reference to FIGURE 6. For example, as shown in FIGURE 6, a processor may include memory control logic along with processor core 800. The processor may include I/O control logic and/or may include I/O control logic integrated with memory control logic.

[0054] Note that with the examples provided herein, interaction may be described in terms of two, three, or more network elements. However, this has been done for purposes of clarity and example only. In certain cases, it may be easier to describe one or more of the functionalities of a given set of flows by only referencing a limited number of network elements. It should be appreciated that communication system 100 and its teachings are readily scalable and can accommodate a large number of components, as well as more complicated/sophisticated arrangements and configurations. Accordingly, the examples provided should not limit the scope or inhibit the broad teachings of communication system 100 as potentially applied to a myriad of other architectures.

[0055] It is also important to note that the operations in the preceding flow diagrams (i.e., FIGURES 3-5) illustrate only some of the possible correlating scenarios and patterns that may be executed by, or within, communication system 100. Some of these operations may be deleted or removed where appropriate, or these operations may be modified or changed considerably without departing from the scope of the present disclosure. In addition, a number of these operations have been described as being executed concurrently with, or in parallel to, one or more additional operations. However, the timing of these operations may be altered considerably. The preceding operational flows have been offered for purposes of example and discussion. Substantial flexibility is provided by communication system 100 in that any suitable arrangements, chronologies, configurations, and timing mechanisms may be provided without departing from the teachings of the present disclosure.

[0056] Although the present disclosure has been described in detail with reference to particular arrangements and configurations, these example configurations and arrangements may be changed significantly without departing from the scope of the present disclosure. Moreover, certain components may be combined, separated, eliminated, or added based on

particular needs and implementations. Additionally, although communication system 100 has been illustrated with reference to particular elements and operations that facilitate the communication process, these elements and operations may be replaced by any suitable architecture, protocols, and/or processes that achieve the intended functionality of communication system 100

[0057] Numerous other changes, substitutions, variations, alterations, and modifications may be ascertained to one skilled in the art and it is intended that the present disclosure encompass all such changes, substitutions, variations, alterations, and modifications as falling within the scope of the appended claims. In order to assist the United States Patent and Trademark Office (USPTO) and, additionally, any readers of any patent issued on this application in interpreting the claims appended hereto, Applicant wishes to note that the Applicant: (a) does not intend any of the appended claims to invoke paragraph six (6) of 35 U.S.C. section 112 as it exists on the date of the filing hereof unless the words "means for" or "step for" are specifically used in the particular claims; and (b) does not intend, by any statement in the specification, to limit this disclosure in any way that is not otherwise reflected in the appended claims.

OTHER NOTES AND EXAMPLES

[0058] Example C1 is at least one machine readable storage medium having one or more instructions that when executed by at least one processor, cause the at least one processor to determine a file characteristic for a characteristic of a file, determine that the file has been modified to create a new file, determine a new characteristic for the characteristic of the new file, and create a security event if the new characteristic does not match the file characteristic.

[0059] In Example C2, the subject matter of Example C1 can optionally include where the file characteristic is a file type associated with the file.

[0060] In Example C3, the subject matter of any one of Examples C1-C2 can optionally include where the one or more instructions that when executed by the at least one processor, further cause the processor to determine the file type using a file type module if the file was not modified by a trusted application.

[0061] In Example C4, the subject matter of any one of Examples C1-C3 can optionally include where the security event includes analyzing a system that includes the file for malware.

[0062] In Example C5, the subject matter of any one of Examples C1-C4 can optionally include where the file characteristic is stored in a protected area of memory.

[0063] In Example C6, the subject matter of any one of Example C1-C5 can optionally include where the one or more instructions that when executed by the at least one processor, further cause the processor to create a copy of the file before the file is been modified to create the new file.

[0064] In Example A1, an electronic device can include a file type module, where the file type module is configured to determine a file characteristic for a characteristic of a file, determine that the file has been modified to create a new file, determine a new characteristic for the characteristic of the new file, and create a security event if the new characteristic does not match the file characteristic.

[0065] In Example, A2, the subject matter of Example A1 can optionally include where the file characteristic is a file type associated with the file.

[0066] In Example A3, the subject matter of any one of Examples A1-A2 can optionally include where the file characteristic is stored in a protected area of memory.

[0067] In Example A4, the subject matter of any one of Examples A1-A3 can optionally include a security module, where the security module is configured to receive the created security event and scan a system that includes the file for malware.

[0068] In Example A5, the subject matter of any one of Examples A1-A4 can optionally include where the security module is further configured to create a copy of the file before the file is been modified to create the new file.

[0069] Example M1 is a method including determining a file characteristic for a characteristic of a file, determining that the file has been modified to create a new file, determining a new characteristic for the characteristic of the new file, and creating a security event if the new characteristic does not match the file characteristic.

[0070] In Example M2, the subject matter of Example M1 can optionally include where the file characteristic is a file type associated with the file.

[0071] In Example M3, the subject matter of any one of the Examples M1-M2 can optionally include determining the new characteristic is performed by a file type module if the file was not modified by a trusted application.

[0072] In Example M4, the subject matter of any one of the Examples M1-M3 can optionally include where the file type is stored in a protected area of memory.

[0073] In Example M5, the subject matter of any one of the Examples M1-M4 can optionally include analyzing a system that includes the file for malware.

[0074] Example S1 is a system for enforcement of file characteristics, the system including a file type module configured for determining a file characteristic for a characteristic of a file, determining that the file has been modified to create a new file, determining a new characteristic for the characteristic of the new file, and creating a security event if the new characteristic does not match the file characteristic.

[0075] In Example S2, the subject matter of Example S1 can optionally include where the file characteristic is a file type associated with the file.

[0076] In Example S3, the subject matter of any one of the Examples S1-S2 can optionally include where the file characteristic is stored in a protected area of memory.

[0077] In Example S4, the subject matter of any one of the Examples S1-S3 can optionally include a security module configured for receiving the created security event and analyzing a system that includes the file for malware.

[0078] In Example S5, the subject matter of any one of the Examples S1-S4 can optionally include where the security module configured for creating a copy of the file before the file is been modified to create a new file.

[0079] Example SS1 is a system for enforcement of file characteristics, the system including means for determining a file characteristic for a characteristic of a file, means for determining that the file has been modified to create a new file, means for determining a new characteristic for the characteristic of the new file, and means for creating a security event if the new characteristic does not match the file characteristic.

[0080] In Example SS2, the subject matter of Example SS1 can optionally include where the file characteristic is a file type associated with the file.

[0081] In Example SS3, the subject matter of any one of the Examples SS1-SS2 can optionally include where the file characteristic is stored in a protected area of memory.

[0082] In Example SS4, the subject matter of any one of the Examples SS1-SS3 can optionally include means for receiving the created security event and analyzing a system that includes the file for malware.

[0083] In Example SS5, the subject matter of any one of the Examples SS1-SS4 can optionally include where means for creating a copy of the file before the file is been modified to create a new file.

[0084] Example X1 is a machine-readable storage medium including machine-readable instructions to implement a method or realize an apparatus as in any one of the Examples A1-A5, or M1-M5. Example Y1 is an apparatus comprising means for performing of any of the Example methods M1-M5. In Example Y2, the subject matter of Example Y1 can optionally include the means for performing the method comprising a processor and a memory. In Example Y3, the subject matter of Example Y2 can optionally include the memory comprising machine-readable instructions.

CLAIMS:

1. At least one computer-readable medium comprising one or more instructions that when executed by at least one processor, cause the at least one processor to:

determine a file characteristic for a characteristic of a file;

determine that the file has been modified to create a new file;

determine a new characteristic for the characteristic of the new file; and

create a security event if the new characteristic does not match the file characteristic.

2. The at least one computer-readable medium of Claim 1, wherein the characteristic is a file type associated with the file.

3. The at least one computer-readable medium of any of Claims 1 and 2, wherein the new characteristic is determined using a file type module if the file was not modified by a trusted application.

4. The at least one computer-readable medium of any of Claims 1-3, wherein the security event includes analyzing a system that includes the file for malware.

5. The at least one computer-readable medium of any of Claims 1-4, wherein the file characteristic is stored in a protected area of memory.

6. The at least one computer-readable medium of any of Claims 1-5, further comprising one or more instructions that when executed by the at least one processor, further cause the processor to:

create a copy of the file before the file has been modified to create the new file.

7. An apparatus comprising:

a file type module configured to:

determine a file characteristic for a characteristic of a file;

determine that the file has been modified to create a new file;

determine a new characteristic for the characteristic of the new file; and

create a security event if the new characteristic does not match the file characteristic.

8. The apparatus of Claim 6, wherein the characteristic is a file type associated with the file.

9. The apparatus of any of Claims 7 and 8, wherein the file characteristic is stored in a protected area of memory.

10. The apparatus of any of Claims 7-9, further comprising:

a security module configured to:

receive the created security event; and

analyze a system that includes the file for malware.

11. The apparatus of any of Claims 7-10, wherein the security module is further configured to:

create a copy of the file before the file is been modified to create the new file.

12. A method comprising:

determining a file characteristic for a characteristic of a file;

determining that the file has been modified to create a new file;

determining a new characteristic for the characteristic of the new file; and

creating a security event if the new characteristic does not match the file characteristic.

13. The method of Claim 12, wherein the characteristic is a file type associated with the file.

14. The method of any of Claims 12 and 13, wherein determining the new characteristic is performed by a file type module if the file was not modified by a trusted application.

15. The method of any of Claims 12-14, wherein the file characteristic is stored in a protected area of memory.

16. The method of any of Claims 12-15, further comprising:

creating a copy of the file before the file is been modified to create a new file.

17. The method of any of Claims 12-16, further comprising:

analyzing a system that includes the file for malware.

18. A system for enforcement of file characteristics, the system comprising:

a file type module configured for:

determining a file characteristic for a characteristic of a file;

determining that the file has been modified to create a new file;

determining a new characteristic for the characteristic of the new file; and

creating a security event if the new characteristic does not match the file characteristic.

19. The system of Claim 18, wherein the file characteristic is a file type associated with the file.

20. The system of any of Claims 18 and 19, wherein the file characteristic is stored in a protected area of memory.

21. The system of any of Claims 18-20, further comprising:

a security module configured for:

receiving the created security event; and

analyzing a system that includes the file for malware.

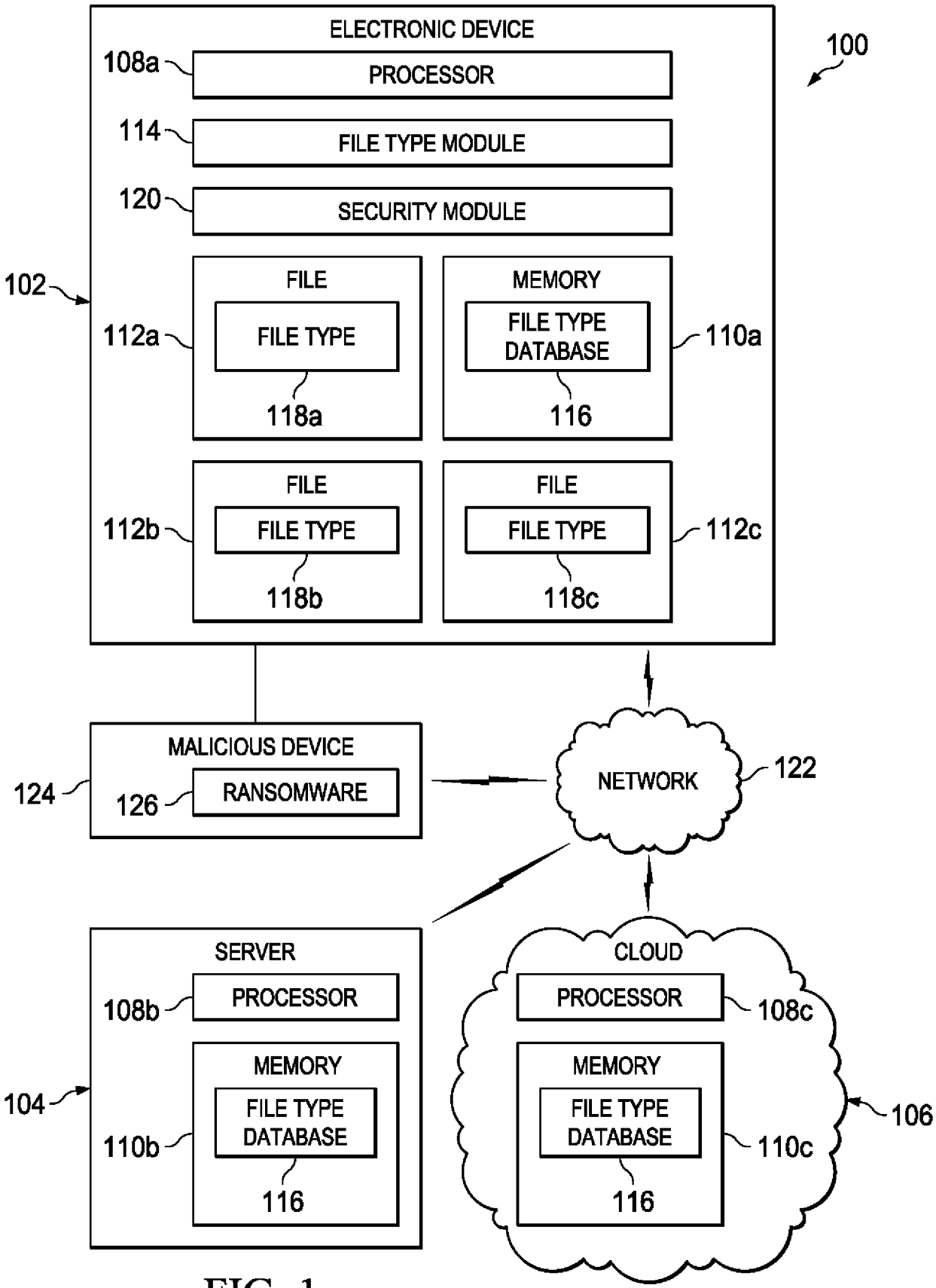
ELECTRONIC DEVICE
100
108a
PROCESSOR
114
FILE TYPE MODULE
120
SECURITY MODULE
102
FILE
112a
FILE TYPE
118a
MEMORY
FILE TYPE DATABASE
116
110a
FILE
112b
FILE TYPE
118b
FILE
FILE TYPE
118c
112c
MALICIOUS DEVICE
124
126
RANSOMWARE
NETWORK
122
SERVER
108b
PROCESSOR
MEMORY
104
FILE TYPE DATABASE
110b
116
CLOUD
PROCESSOR
108c
MEMORY
FILE TYPE DATABASE
110c
116
106

FIG. 1

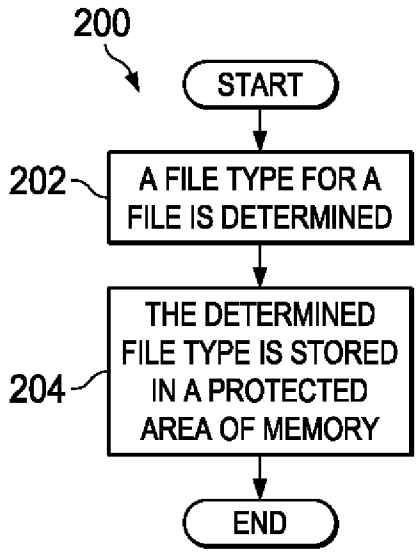
200
START
202
A FILE TYPE FOR A FILE IS DETERMINED
204
THE DETERMINED FILE TYPE IS STORED IN A PROTECTED AREA OF MEMORY
END

FIG. 2

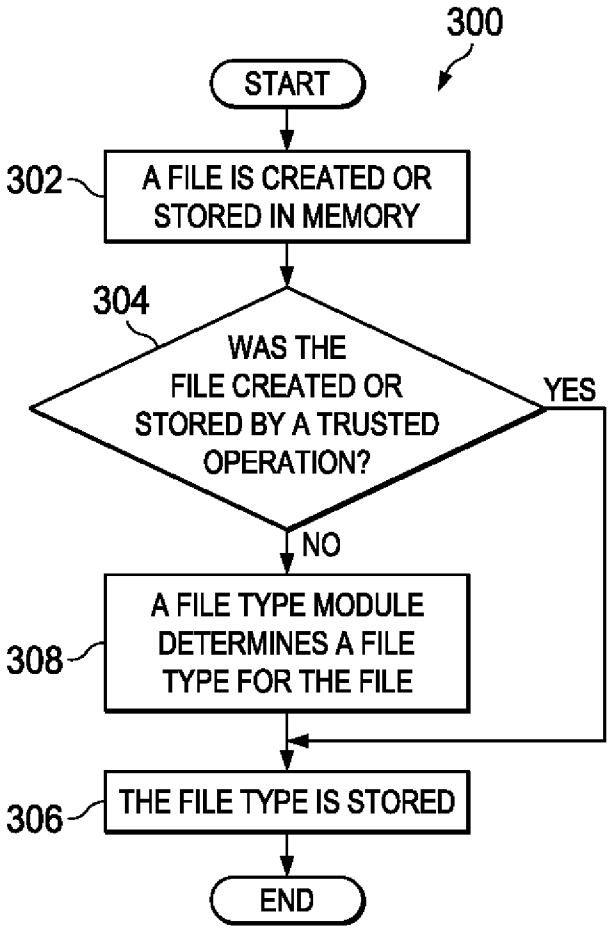
300
START
302
A FILE IS CREATED OR STORED IN MEMORY
304
WAS THE FILE CREATED OR STORED BY A TRUSTED OPERATION?
YES
NO
308
A FILE TYPE MODULE DETERMINES A FILE TYPE FOR THE FILE
306
THE FILE TYPE IS STORED
END

FIG. 3

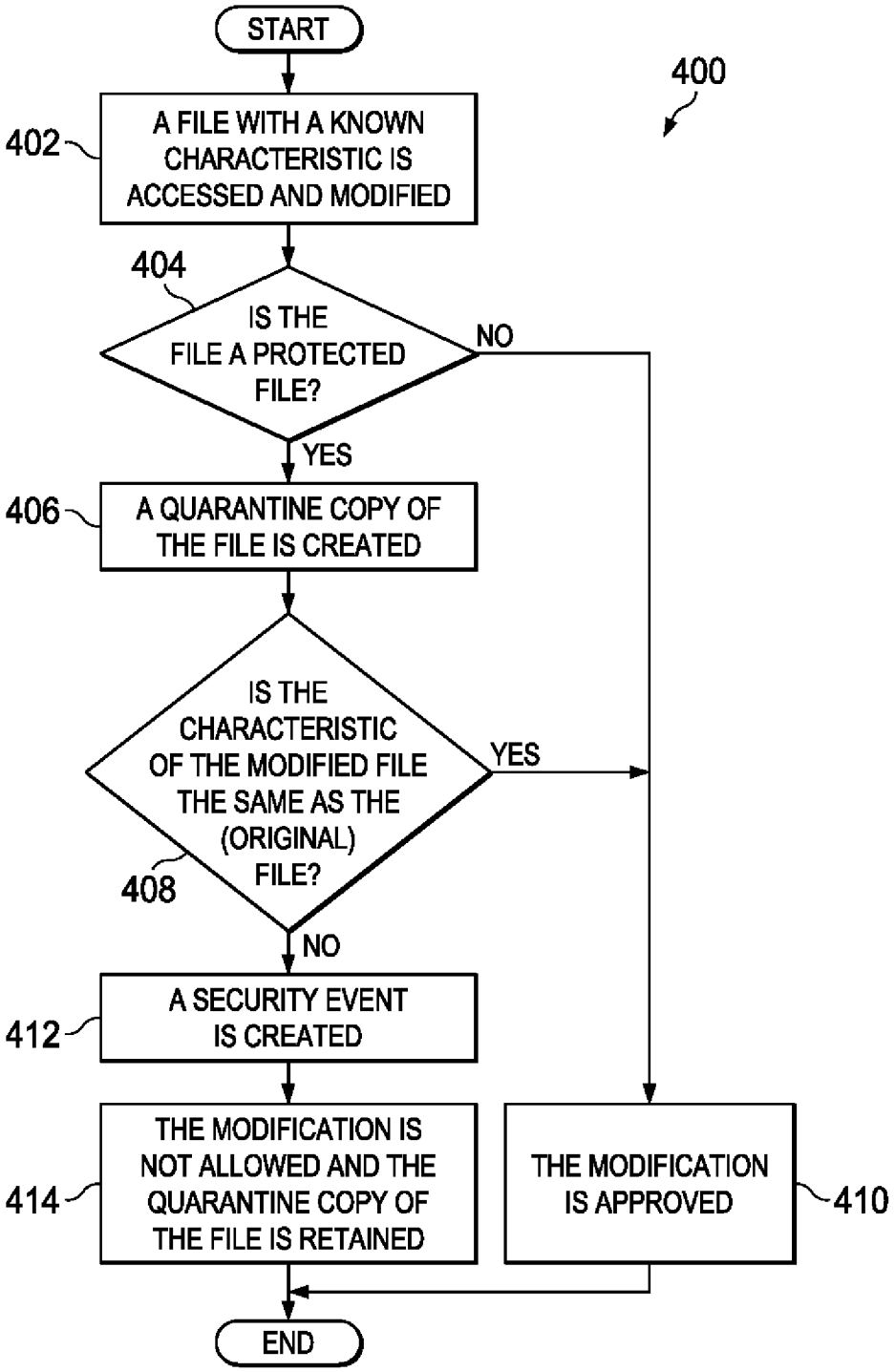
START
400
A FILE WITH A KNOWN CHARACTERISTIC IS ACCESSED AND MODIFIED
402
404
IS THE FILE A PROTECTED FILE?
NO
YES
A QUARANTINE COPY OF THE FILE IS CREATED
406
IS THE CHARACTERISTIC OF THE MODIFIED FILE THE SAME AS THE (ORIGINAL) FILE?
YES
408
NO
A SECURITY EVENT IS CREATED
412
THE MODIFICATION IS NOT ALLOWED AND THE QUARANTINE COPY OF THE FILE IS RETAINED
414
THE MODIFICATION IS APPROVED
410
END

FIG. 4

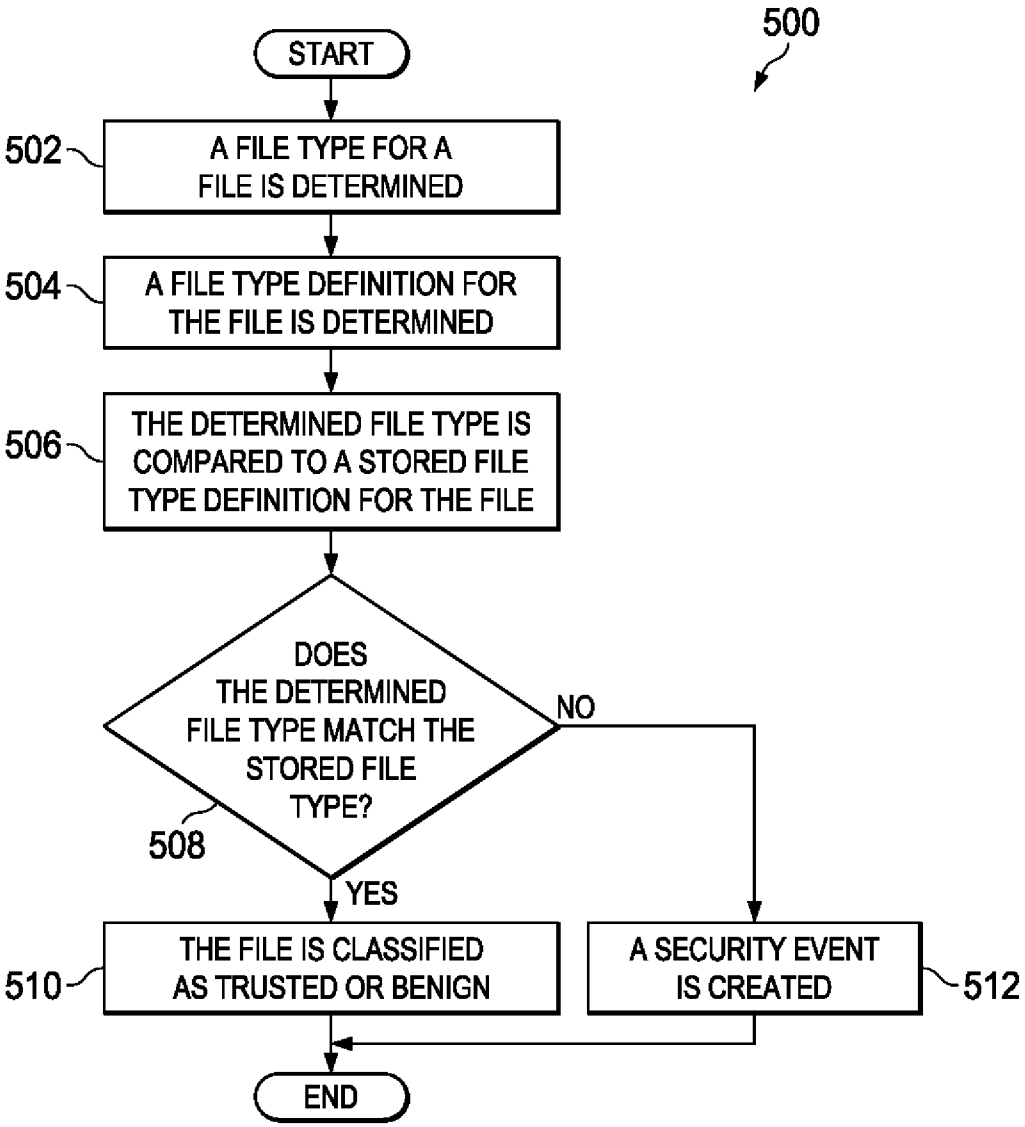
500
START
502
A FILE TYPE FOR A FILE IS DETERMINED
504
A FILE TYPE DEFINITION FOR THE FILE IS DETERMINED
506
THE DETERMINED FILE TYPE IS COMPARED TO A STORED FILE TYPE DEFINITION FOR THE FILE
DOES THE DETERMINED FILE TYPE MATCH THE STORED FILE TYPE?
508
NO
YES
510
THE FILE IS CLASSIFIED AS TRUSTED OR BENIGN
A SECURITY EVENT IS CREATED
512
END

FIG. 5

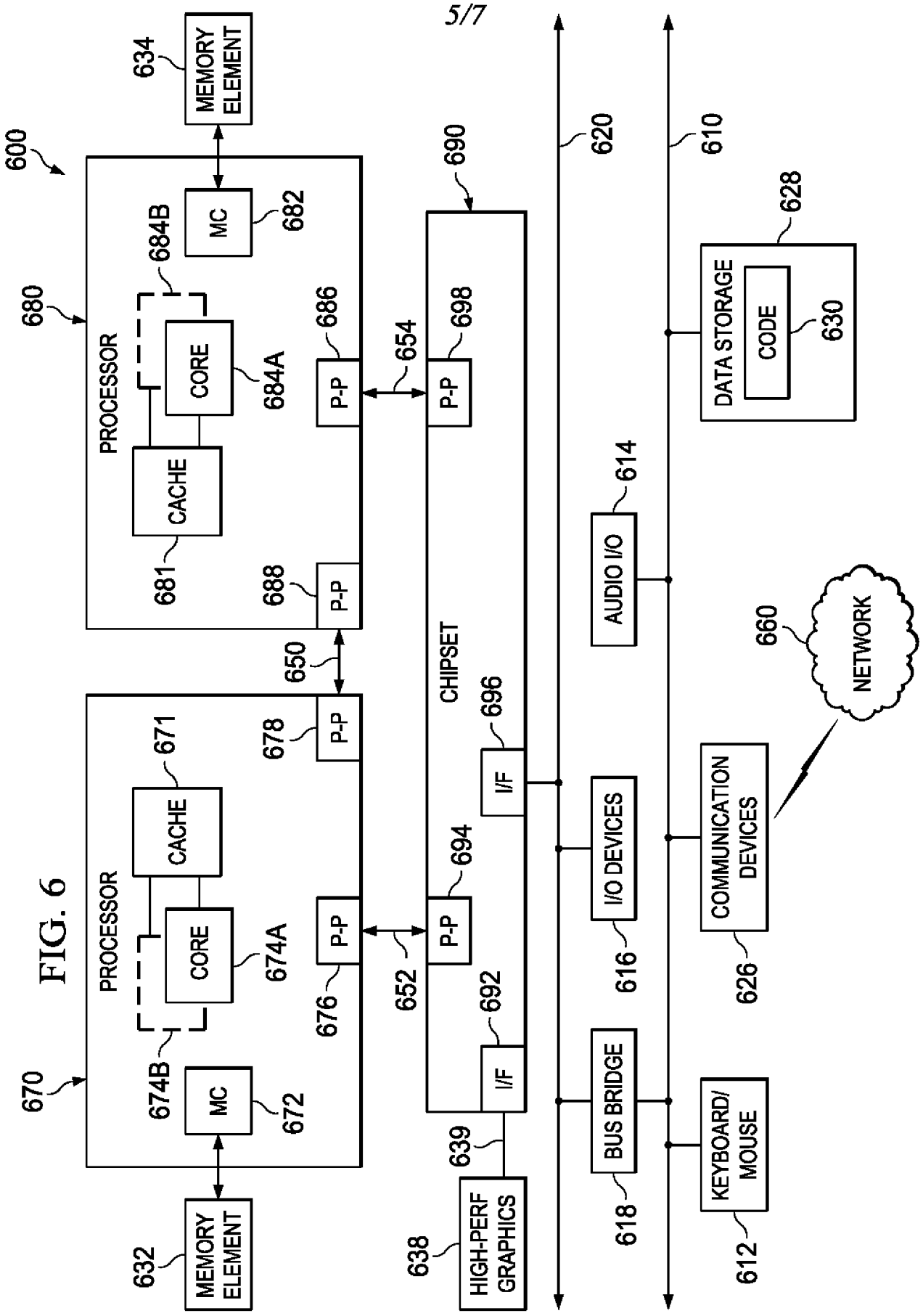
600
634
MEMORY ELEMENT
680
PROCESSOR
684B
MC
682
CORE
684A
CACHE
681
686
P-P
688
P-P
654
650
690
CHIPSET
698
P-P
696
I/F
694
P-P
692
I/F
639
638
HIGH-PERF GRAPHICS
620
610
628
DATA STORAGE
CODE
630
614
AUDIO I/O
660
NETWORK
616
I/O DEVICES
626
COMMUNICATION DEVICES
618
BUS BRIDGE
612
KEYBOARD/ MOUSE
670
PROCESSOR
671
CACHE
674A
CORE
674B
MC
672
678
P-P
676
P-P
652
632
MEMORY ELEMENT

FIG. 6

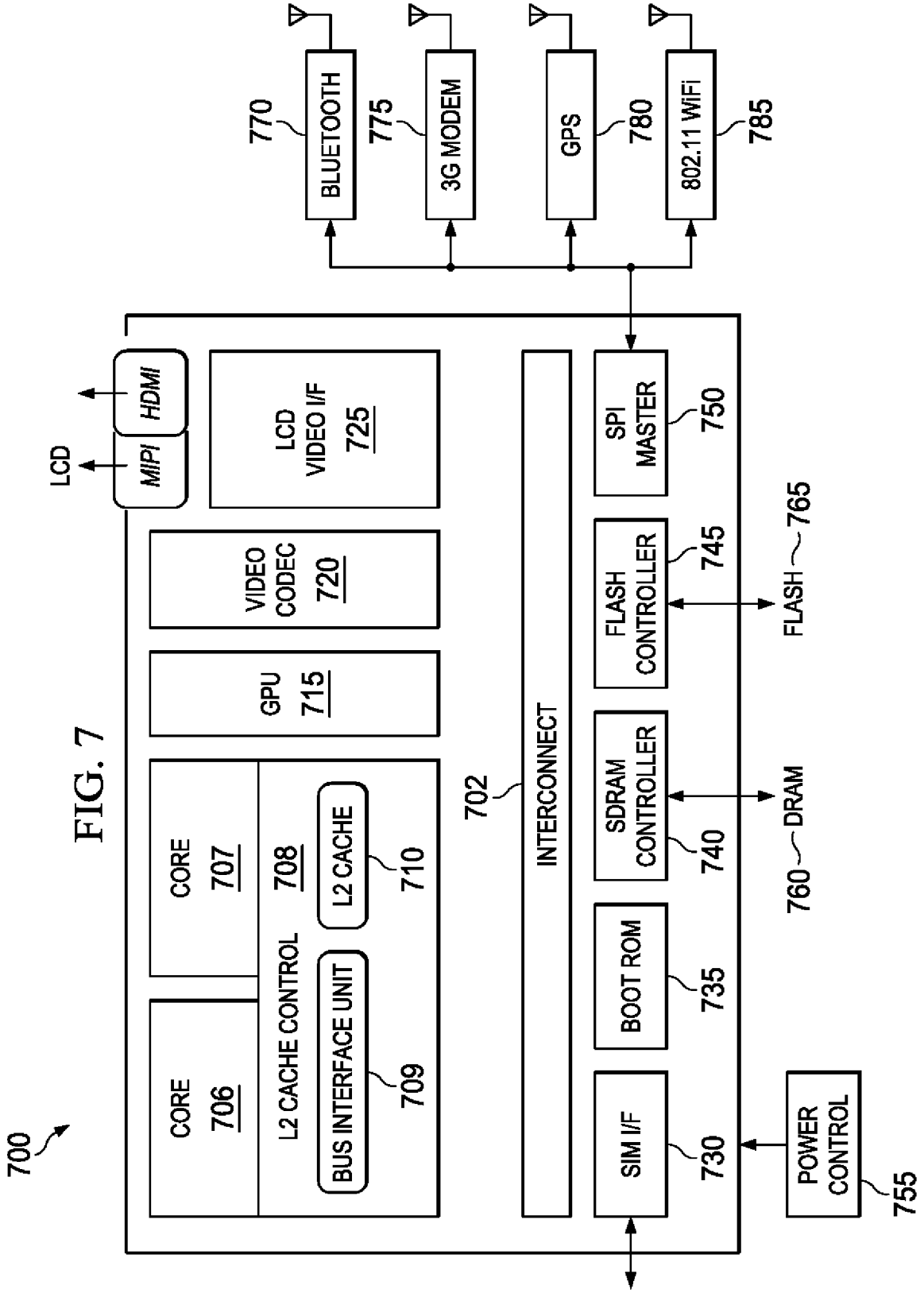
700
CORE 706
CORE 707
L2 CACHE CONTROL 708
BUS INTERFACE UNIT 709
L2 CACHE 710
GPU 715
VIDEO CODEC 720
LCD VIDEO I/F 725
HDMI
MIPI
LCD
INTERCONNECT 702
SIM I/F 730
BOOT ROM 735
SDRAM CONTROLLER 740
FLASH CONTROLLER 745
SPI MASTER 750
POWER CONTROL 755
DRAM 760
FLASH 765
BLUETOOTH 770
3G MODEM 775
GPS 780
802.11 WiFi 785

FIG. 7

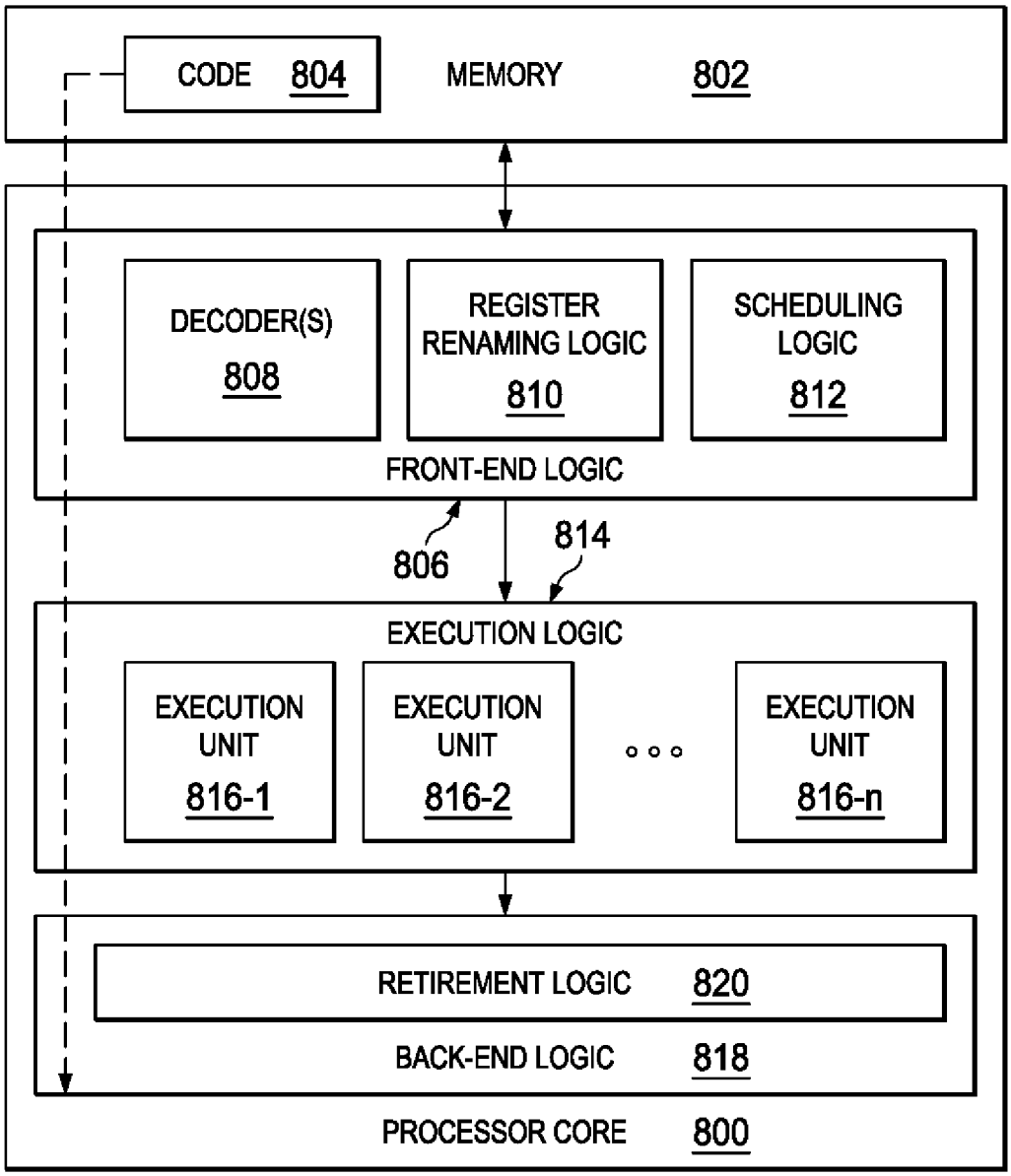
CODE 804
MEMORY 802
DECODER(S) 808
REGISTER RENAMING LOGIC 810
SCHEDULING LOGIC 812
FRONT-END LOGIC
806
814
EXECUTION LOGIC
EXECUTION UNIT 816-1
EXECUTION UNIT 816-2
EXECUTION UNIT 816-n
RETIREMENT LOGIC 820
BACK-END LOGIC 818
PROCESSOR CORE 800

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2016/048104

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/60(2013.01)i, G06F 21/78(2013.01)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)
G06F 21/60; G06F 21/56; G06F 17/30; G06F 12/00; G06F 11/00; G06F 21/78

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords:malicious, comparation, modification, file type

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2015-0058987 A1 (F-SECURE CORPORATION) 26 February 2015 See paragraphs [0020], [0022]-[0025]; claims 1, 4, 8; and figures 2, 4.	1-21
Y	US 2011-0082838 A1 (JARNO NIEMELA) 07 April 2011 See paragraph [0007]; claim 1; and figures 1, 2.	1-21
A	US 2008-0086513 A1 (THOMAS EDWARD O`BRIEN) 10 April 2008 See paragraphs [0025]-[0033], [0052]-[0056]; and figures 3-5.	1-21
A	US 2013-0311708 A1 (CHIEN-FU LEE) 21 November 2013 See paragraph [0009]; and figure 1A.	1-21
A	US 8352522 B1 (YI-HUNG CHENG) 08 January 2013 See claim 1; and figure 2.	1-21

☐ 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 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
29 November 2016 (29.11.2016)	**29 November 2016 (29.11.2016)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea
Facsimile No. +82-42-481-8578

Authorized officer
CHIN, Sang Bum
Telephone No. +82-42-481-8398

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2016/048104

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2015-0058987 A1	26/02/2015	GB 2517483 A	25/02/2015
		GB 2517483 B	22/07/2015
		US 9292687 B2	22/03/2016
US 2011-0082838 A1	07/04/2011	EP 2486506 A1	15/08/2012
		EP 2486506 B1	06/05/2015
		WO 2011-042306 A1	14/04/2011
US 2008-0086513 A1	10/04/2008	US 7769731 B2	03/08/2010
US 2013-0311708 A1	21/11/2013	TW 201348965 A	01/12/2013
		TW I498738 B	01/09/2015
		US 8954692 B2	10/02/2015
US 8352522 B1	08/01/2013	US 9378369 B1	28/06/2016