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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
26 January 2017 (26.01.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/014727 A1

(51) International Patent Classification:
G06F 21/62 (2013.01) *H04L 9/08* (2006.01)

(21) International Application Number:
PCT/US2015/040949

(22) International Filing Date:
17 July 2015 (17.07.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP** [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).

(72) Inventor: **YORK, Justin E.**; 11445 Compaq Center Dr. W., Houston, Texas 77070 (US).

(74) Agents: **GONCALVES, Brian** et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79 , Fort Collins, CO 80528 (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).

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

[Continued on next page]

(54) Title: DATA TAMPER DETECTION

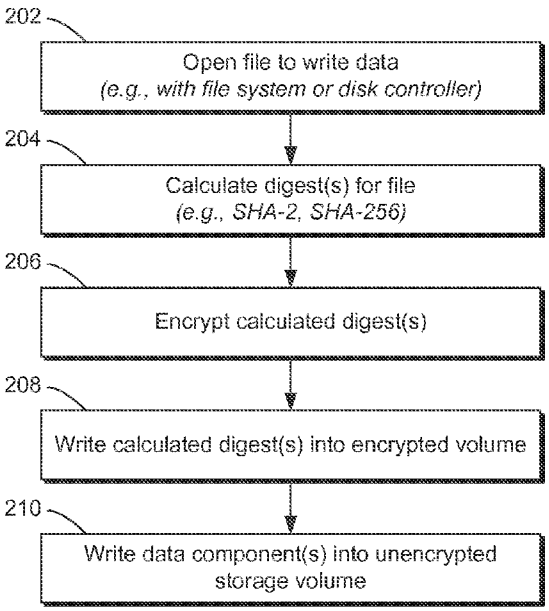

FIG. 2

(57) Abstract: Examples include a system for data tamper detection including an encrypted storage volume and an unencrypted storage volume, a digest calculation engine to calculate a digest for a file, an encryption engine to encrypt the calculated digest for the file, and a storage engine to store the encrypted calculated digest in the encrypted storage volume and to store a data component for the file in the unencrypted storage volume. In some examples, the file is read into a memory space, a digest calculation is performed on the file, a saved digest calculation is loaded from an encrypted disk volume, and the digest calculation performed on the file is compared with the saved digest calculation.

WO 2017/014727 A1

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

Published:

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

DATA TAMPER DETECTION

BACKGROUND

[0001] Computing systems, devices, and electronic components may utilize content in the form of digital files. A computer system may create files, store files, or receive files on, for example, a disk drive. Such files may contain data designated at various levels of sensitivity, and may be subject to tampering.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] The following detailed description references the drawings, wherein:

[0003] FIG. 1 is a block diagram of a system for data tamper detection, according to an example;

[0004] FIG. 2 is a flowchart of writing a file to support data tamper detection upon reading the file, according to an example;

[0005] FIG. 3 is a flowchart of detecting data tampering of a file; and

[0006] FIG. 4 is a block diagram of a computing device to detect data tampering, according to an example.

DETAILED DESCRIPTION

[0007] Various examples described below provide a system for data tamper detection including an encrypted storage volume and an unencrypted storage volume. To write a file, a digest calculation engine may calculate a digest for the file, an encryption engine may encrypt the calculated digest for the file, and a storage engine may store the encrypted calculated digest in the encrypted storage volume and store a data component for the file in the unencrypted storage volume.

[0008] In some examples, to read the file and detect data tampering, the file is read

into a memory space, a digest calculation is performed (or "recalculated") on the file, a saved digest calculation is loaded from an encrypted disk volume, and the digest calculation performed on the file is compared with the saved digest calculation. If the digest calculation performed on the file matches the saved digest calculation, access to a memory space containing the file may be granted.

[0009] The adoption of technology has increased the importance of security in computing systems, with such systems routinely storing personal and sensitive data in the consumer, commercial, and industrial sectors, as well as by governments. Data encryption may be used as a technique to prevent access to sensitive data, as well as to prevent modification of the data or "tampering" with the data. For example, if an encrypted file is altered, the file may no longer decrypt back to its original form.

[0010] Although data encryption may provide a mechanism to secure data, it may be a computationally expensive operation that can greatly reduce the throughput of a computing system, and may require additional capital upgrades or hardware encryption offload to service the larger workloads of such a system. For example, encrypting a file and writing the encrypted file may require substantially more resources such as CPU, memory, and disk access than writing an unencrypted file. In some systems, such as a network boot target where storage of a large number of frequently accessed files of very large size may be common, such an impact may be particularly troubling.

[0011] In some cases, users of data may not require a level of data security that requires full encryption of a file, and thus encryption of files would unnecessarily impact system performance. However, such users may desire to determine or detect whether a file has been modified or tampered with after it was written, while minimizing any performance impact.

[0012] Referring now to the drawings, FIG. 1 is a block diagram of a system for data tamper detection, according to an example.

[0013] In the example of FIG. 1, a data tamper detection system 100 may run or execute on a computing system, device, and/or electronic component (hereinafter "computing device"). As used herein, a computing device may be a server, blade enclosure, desktop computer, laptop (or notebook) computer, workstation, tablet computer, mobile phone, smart device, or any other processing device or equipment including a processing resource.

[0014] A computing device may store or create files, run an operating system, and/or run applications such as file storage tools, word processing tools, spreadsheet tools, presentation tools, programming tools, communications tools, utilities, games, or other applications. For example, the applications running on a computing device or data tamper detection system 100 may include engines, controllers, circuitry, or modules to write or read files to unencrypted and/or encrypted disk volumes, and to detect data tampering of such files.

[0015] A computing device may also include a machine-readable storage medium or storage device 102 and/or a processing resource 128, and may comprise or be encoded with instructions stored on a machine-readable storage medium and executable by the processing resource, as discussed below in more detail with respect to FIGS. 2-4. In some examples, the instructions may be implemented as engines comprising any combination of hardware, e.g., circuitry, and programming to implement the functionalities of the engines, as described below.

[0016] In examples described herein, such combinations of hardware and programming may be implemented in a number of different ways. For example, the programming for engines, such as an encryption engine or digest calculation engine, may be processor executable instructions stored on a non-transitory machine-readable storage medium and the hardware for the engines may include a processing resource and/or circuitry to execute those instructions.

[0017] Data tamper detection system 100 may include storage device 102, which may be a hard disk drive, a solid state disk drive, flash memory, or any other type of machine-readable storage or memory. Storage device 102 may include an encrypted storage volume 104 and an unencrypted storage volume 112, or any combination of encrypted and unencrypted storage volumes, to split encrypted data from unencrypted data. Storage device may store files, as described herein, and may also store the machine-readable instructions to implement the data tamper detection system 100, such as storage of instructions for encryption, digest calculation, or other data tamper detection functions described herein. Volumes 104 and 112 and may reside on the same storage device, or across separate storage devices or platforms.

[0018] Encrypted storage volume 104 may be a volume, partition, or other area of storage device 102 dedicated or accessible to data tamper detection system 100 for purposes of storing files or file parts intended to be encrypted. In an example, files stored on the storage device 102 as part of data tamper detection system 100 are stored with an encrypted digest calculation, as described below in more detail with respect to FIGS. 2-4. In some examples, the encrypted digest calculation is stored in parts, such as saved digest calculations 106, 108, and 110.

[0019] Storage volume 112, which may be unencrypted, may be a volume, partition, or other area of storage device 102 dedicated or accessible to data tamper detection system 100 for purposes of storing files or file parts that are not intended to be encrypted. In an example, files stored on the storage device 102 as part of data tamper detection system 100 are stored with an encrypted digest calculation, as described above, with the remainder of the file, e.g., the data component or payload, stored unencrypted, as described below in more detail with respect to FIGS. 2-4. In some examples, the unencrypted data components are stored in parts, such as data components 114, 116, and 118.

[0020] Data tamper detection system 100 may also include a digest calculation engine 130 to calculate a digest for a file stored on storage device 102, such as digests 106-110 discussed above, or digests 120-124 discussed below. In some examples, a digest may be a cryptographically derived value or a short summary of a file created when the file is written, and that can be used at a later stage such as when reading the file to determine if the file has been tampered with since it was last written. A digest may be calculated using, for example, the Secure Hash Algorithm ("SHA") at various strengths, block sizes, or hash lengths, such as SHA-2 or SHA-3.

[0021] In other examples, other cryptographic hash functions may be used to calculate the digest or as part of calculating the digest. The calculated digests may be stored in a small, fixed size, regardless of the file size of the overall file, e.g., regardless of the size of the data components associated with the digest.

[0022] Data tamper detection system 100 may also include an encryption engine 132 to encrypt parts of storage device 102, such as storage volume 104 or digest calculations 106, 108, and 110. Data, such as calculated digests, may be encrypted with any disk-level or file system-level encryption technique. As described herein, some data associated with a file may be stored in unencrypted form.

[0023] Data tamper detection system 100 may also include a storage engine 134 to store data or files to, e.g., storage device 102. Storage engine 134 may be, in examples, part of a disk controller or file system.

[0024] Data tamper detection system 100 may also include, store, read, or fetch current or new digest calculations, e.g., digest calculations 120, 122, and 124, which may be calculated (or "recalculated") when a file is read or when a file read attempt is executed, as discussed below in more detail with respect to FIGS. 3 and 4.

[0025] In some examples, data tamper detection system 100 may also include an encryption key or keys, which may be stored on a storage device, or may be stored

remote to the storage device. In the case of a key stored remotely, if a storage device were physically removed from data tamper detection system 100 or a computing device, the encrypted data could not be accessed.

[0026] FIG. 2 is a flowchart of writing a file to support data tamper detection upon reading the file, according to an example.

[0027] In block 202, a file is opened, accessed, or created to write data to the file or file wrapper. The file may be opened with, for example, a file system or a disk controller, and may be stored on, for example, storage device 102.

[0028] In block 204, a digest or digests are calculated for the file. As described above, a digest may be calculated using, for example, the Secure Hash Algorithm at various strengths, block sizes, or hash lengths, such as SHA-2 or SHA-3, or other cryptographic hash functions.

[0029] In block 206, the calculated digest or digests are encrypted. As discussed above, the digests may be encrypted with any disk-level or file system-level encryption technique.

[0030] In block 208, the calculated digest or digests may be stored in an encrypted volume on a disk, such as encrypted volume 104, as a single digest or multiple digests, e.g., as digests 106-110 or any other combination.

[0031] In block 210, a data component or components associated with the digest may be stored in an unencrypted disk volume, such as unencrypted storage volume 112. A data component may be defined as the entire file to be stored without the digest, which may be stored separately in encrypted form as discussed above. The data component may be broken into parts, such as data components 114-118, or may be stored in a single part.

[0032] FIG. 3 is a flowchart of detecting data tampering of a file.

[0033] In block 302, a file is opened with a file system driver. As discussed in the

example of FIG. 4, the file may also be opened with a disk controller or other tool for opening, accessing, or reading files.

[0034] In block 304, the file is read into memory. In some examples, block 304 may comprise reading the unencrypted part of the file into memory, e.g., by accessing unencrypted volume 112.

[0035] In block 306, a digest calculation or calculations on the file read into memory are performed. The digest calculation may be performed as discussed above, e.g., using SHA or another cryptographic hash routine. In an example, the digest calculation or recalculation of block 306 would run the same cryptographic routine as used to calculate the last or saved digest calculation when the file was written, e.g., in block 204.

[0036] The digest calculation of block 306 may be represented by, for example, the "current" or "new" digest calculations 120-124 as shown in FIG. 1, such that the digests are current or new as of the time the data tamper detection system 100 is reading the file. The current or new digest calculations may also be referred to as "recalculated" digests as discussed above. The digest calculations 120-124 may be stored, e.g., on storage device 102, or may be stored temporarily, e.g., in memory, until data tamper detection system 100 can verify that the file has not be tampered with prior to reading.

[0037] In block 308, the saved digest calculation or calculations stored in the file are loaded or fetched. The saved digest calculations may be represented by digest calculations 106-110.

[0038] In decision block 310, a determination is made as to whether the current or new digest calculations, e.g., calculations 120-124, match the saved digest calculations, e.g., calculations 106-110. If the current digest calculation does match the stored digest calculation, the file has not been tampered with and the flow proceeds to block 312. In block 312, the memory space containing the file, e.g., the file or unencrypted data component read into memory in block 304, is made available to the file system and/or disk

controller.

[0039] If the current digest calculation does not match the stored digest calculation, the file has been tampered with, corrupted, or altered and the flow proceeds to block 314. In block 314, the file or unencrypted data component read into memory in block 304 is not made available to the file system and/or disk controller. In some examples, an alert to a user, process, or other output may be generated. In some examples, the unencrypted data component may also be cleared from, for example, memory.

[0040] FIG. 4 is a block diagram of a computing device to detect data tampering, according to an example.

[0041] The computing system 402 including data tamper detection system 400 of FIG. 4 may comprise a power source 404, a memory or storage medium 406, a processing resource or processor 408, and a disk controller 410. Files and/or data 412 may be read or written by disk controller 410, which may communicate or interface with a file system.

[0042] As used herein, a processing resource may be at least one of a central processing unit (CPU), a semiconductor-based microprocessor, a graphics processing unit (GPU), a field-programmable gate array (FPGA) configured to retrieve and execute instructions, other electronic circuitry suitable for the retrieval and execution of instructions stored on a machine-readable storage medium, or a combination thereof. Processing resource 402 may fetch, decode, and execute instructions, e.g., instructions 416-426, stored on memory or storage medium 406 to perform the functionalities described herein. In other examples, the functionalities of any of the instructions of memory or storage medium 406 may be implemented in the form of electronic circuitry, in the form of executable instructions encoded on a machine-readable storage medium, or a combination thereof.

[0043] As used herein, a "machine-readable storage medium" may be any electronic, magnetic, optical, or other physical storage apparatus to contain or store information such

as executable instructions, data, and the like. For example, any machine-readable storage medium described herein may be any of Random Access Memory (RAM), volatile memory, non-volatile memory, flash memory, a storage drive (e.g., a hard drive), a solid state drive, any type of storage disc (e.g., a compact disc, a DVD, etc.), and the like, or a combination thereof. Further, any machine-readable storage medium described herein may be non-transitory. In examples described herein, a machine-readable storage medium or media is part of an article (or article of manufacture). An article or article of manufacture may refer to any manufactured single component or multiple components. The storage medium may be located either in the computing device executing the machine-readable instructions, or remote from but accessible to the computing device (e.g., via a computer network) for execution.

[0044] In some examples, instructions 414-426 may be part of an installation package that, when installed, may be executed by processing resource 408 to implement the functionalities described herein in relation to instructions 416-426. In such examples, memory or storage medium 406 may be a portable medium, such as a CD, DVD, or flash drive, or a memory maintained by a server from which the installation package can be downloaded and installed. In other examples, instructions 416-426 may be part of an application, applications, or component(s) already installed on a computing device 402 including a processing resource.

[0045] In some examples, memory 406 may be separate from a machine-readable storage medium, as described herein, and may be volatile storage utilized by system 400 for performing the processes as described herein, for example. In some examples, a memory may temporarily store data portions while performing processing operations on them, such as calculating a digest.

[0046] The instructions in the memory or machine-readable storage of system 400 may comprise a data tamper engine 414. In block 416 of data tamper engine 414, the

instructions may access a file stored in an encrypted volume and in an unencrypted volume. The instructions may perform a new digest calculation on the file in block 418, and fetch a saved digest calculation from the encrypted volume in block 420.

[0047] In an example, the instructions may compare the new digest calculation with the saved digest in block 422. If the new digest calculation matches the saved digest calculation, the instructions of block 424 may permit access to the data components associated with the file stored in the encrypted volume via, for example, disk controller 410. The instructions in block 426 may then instruct the disk controller 410 to output the file to the file system, or to make the memory or storage device space containing the file accessible to the file system.

[0048] In some examples, the system of FIG. 4 may be or may communicate with a network boot target. If tampering is detected, e.g., if the digests do not match, the system of FIG. 4 may halt delivery of a boot image to protect a network boot target. In such examples, any performance impact is minimized as such systems typically use large files that change infrequently, and thus only a secure digest calculation is executed at the time the file is read without significant computational overhead that would be incurred with decryption.

[0049] Although the instructions of FIGS. 2-4 show a specific order of performance of certain functionalities, the instructions of FIGS. 2-4 are not limited to that order. For example, the functionalities shown in succession may be performed in a different order, may be executed concurrently or with partial concurrence, or a combination thereof.

[0050] All of the features disclosed in this specification (including any accompanying claims, abstract and drawings), and/or all of the elements of any method or process so disclosed, may be combined in any combination, except combinations where at least some of such features and/or elements are mutually exclusive.

CLAIMS

What is claimed is:

1. A system for data tamper detection, comprising:

a processing resource to write to a file;

a storage device comprising an encrypted storage volume and an unencrypted storage volume;

a digest calculation engine to calculate a digest for the file;

an encryption engine to encrypt the calculated digest for the file; and

a storage engine to store the encrypted calculated digest in the encrypted storage volume and to store a data component for the file in the unencrypted storage volume.

2. The system of claim 1, wherein the digest calculation engine comprises a secure hash routine.

3. The system of claim 1, wherein the storage device is a network boot target.

4. The system of claim 1, wherein an encryption key to encrypt the calculated digest for the file is stored remote to the storage device.

5. The system of claim 1, wherein the storage engine is a file system.

6. The system of claim 1, wherein the storage engine is a disk controller.

7. A method for detecting data tampering of a file, comprising:

opening a file with a file system driver;

reading the file into a memory space;

performing, with a processor, a digest calculation on the file;

loading a saved digest calculation from an encrypted disk volume;

comparing the digest calculation performed on the file with the saved digest calculation loaded from the encrypted disk volume; and

in the event that the digest calculation performed on the file does not match the saved digest calculation loaded from the encrypted disk volume, denying access to the memory space containing the file.

8. The method of claim 7, further comprising halting delivery of a boot image.

9. The method of claim 7, further comprising transmission of a tamper alert.

10. The method of claim 7, wherein performing the digest calculation comprises executing a secure hash routine.

11. An article comprising at least one non-transitory machine-readable storage medium comprising instructions executable by a processing resource of a data tamper detection system to:

access, with a disk controller, a file stored in an encrypted volume and an unencrypted volume on a storage device;

perform a new digest calculation on the file;

fetch a saved digest calculation from the encrypted volume;

compare the new digest calculation with the saved digest calculation from the encrypted volume; and

permit access to a data component associated with the file stored in the unencrypted volume.

12. The article of claim 11, further comprising instructions to decrypt the saved digest calculation from the encrypted volume.

13. The article of claim 11, further comprising a decryption key stored independent of the storage device to decrypt the saved digest calculation.

14. The article of claim 11, further comprising instructions to output the data component associated with the file to a file system.

15. The article of claim 11, wherein the new digest calculation comprises a secure hash routine.

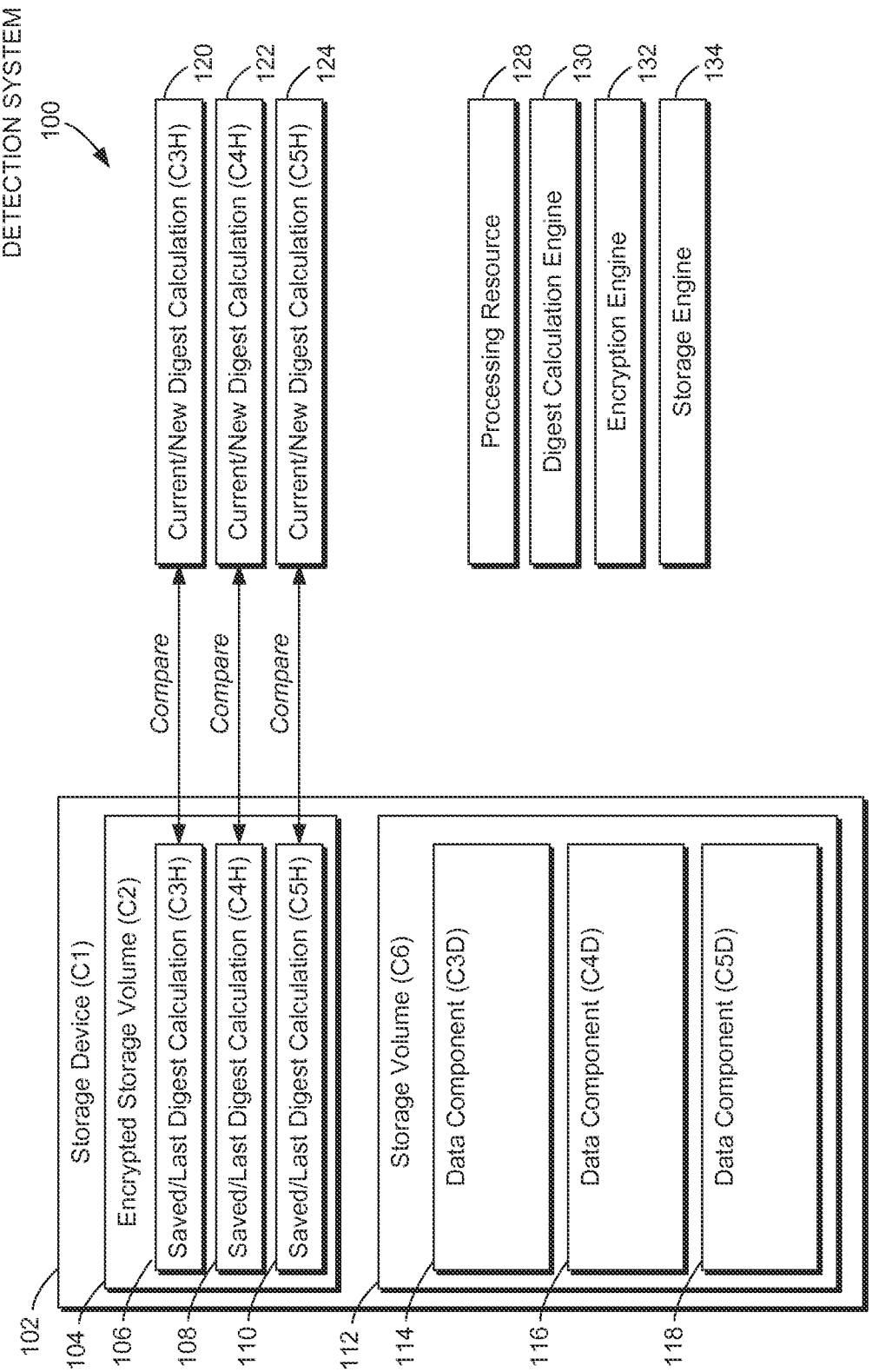
DATA TAMPER DETECTION SYSTEM
100
Storage Device (C1)
102
Encrypted Storage Volume (C2)
104
Saved/Last Digest Calculation (C3H)
106
Saved/Last Digest Calculation (C4H)
108
Saved/Last Digest Calculation (C5H)
110
Storage Volume (C6)
112
Data Component (C3D)
114
Data Component (C4D)
116
Data Component (C5D)
118
Compare
Compare
Compare
Current/New Digest Calculation (C3H)
120
Current/New Digest Calculation (C4H)
122
Current/New Digest Calculation (C5H)
124
Processing Resource
128
Digest Calculation Engine
130
Encryption Engine
132
Storage Engine
134

FIG. 1

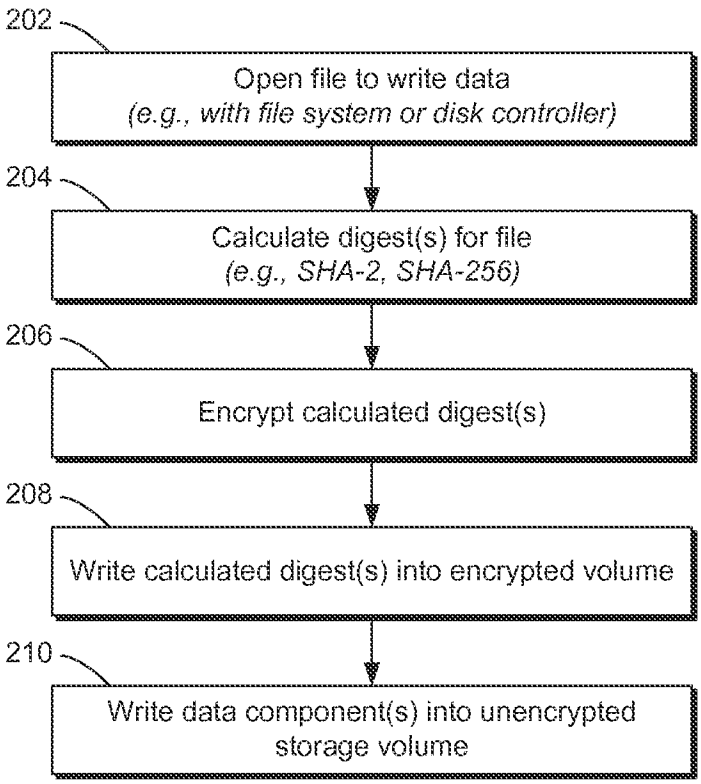
202
Open file to write data
(e.g., with file system or disk controller)
204
Calculate digest(s) for file
(e.g., SHA-2, SHA-256)
206
Encrypt calculated digest(s)
208
Write calculated digest(s) into encrypted volume
210
Write data component(s) into unencrypted storage volume

FIG. 2

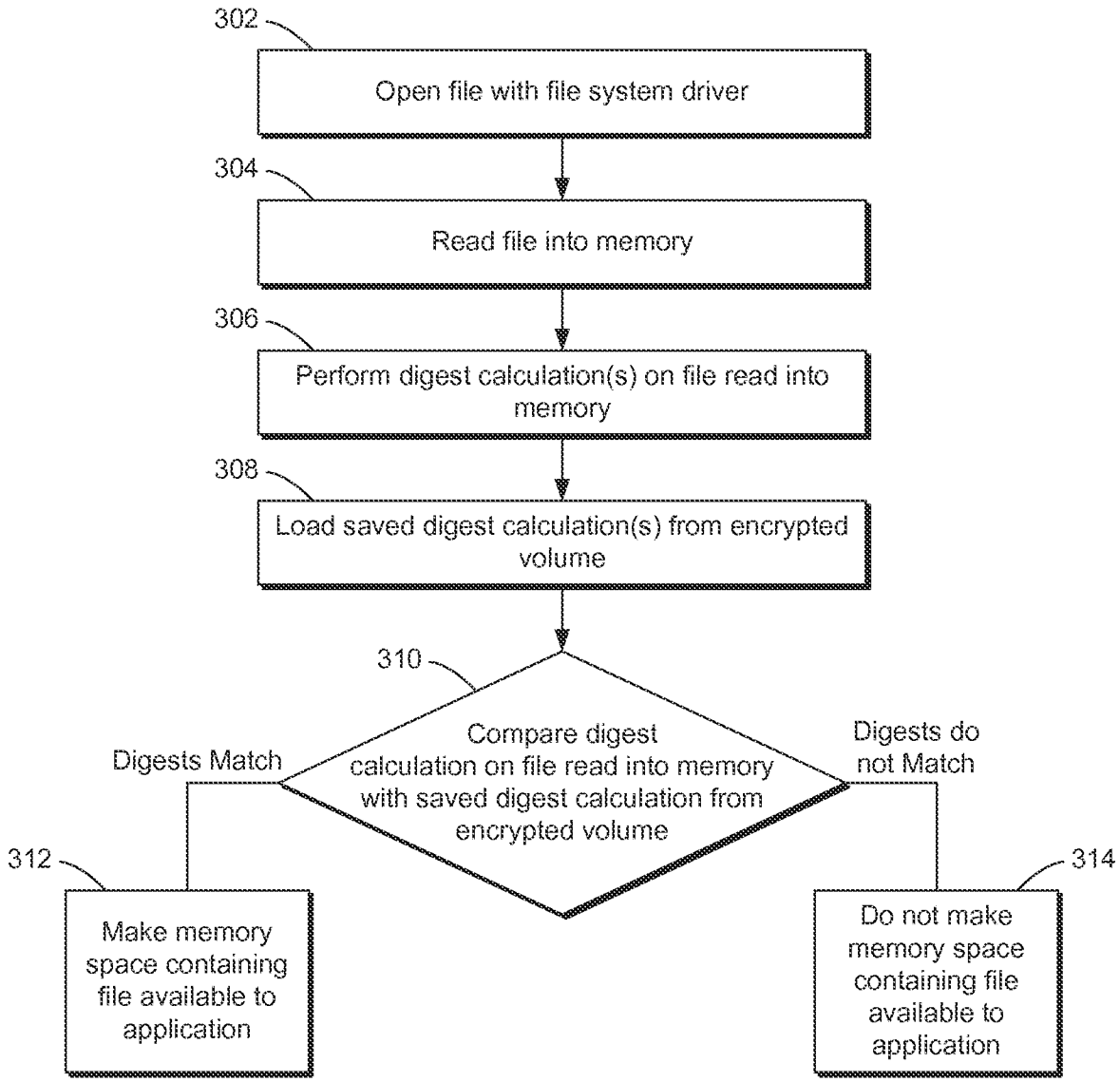
302
Open file with file system driver
304
Read file into memory
306
Perform digest calculation(s) on file read into memory
308
Load saved digest calculation(s) from encrypted volume
310
Compare digest calculation on file read into memory with saved digest calculation from encrypted volume
Digests Match
Digests do not Match
312
Make memory space containing file available to application
314
Do not make memory space containing file available to application

FIG. 3

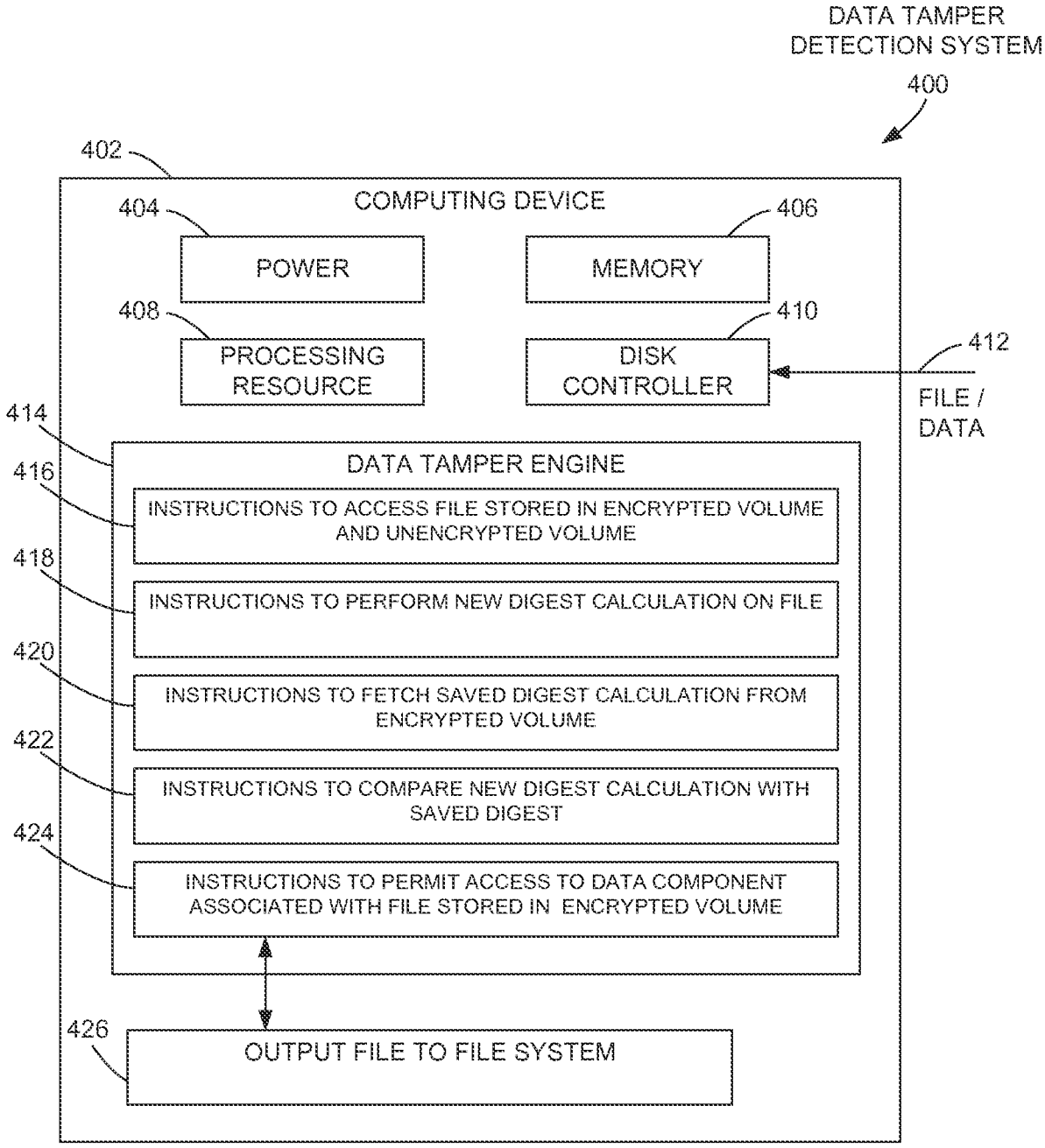
DATA TAMPER DETECTION SYSTEM
400
402
COMPUTING DEVICE
404
POWER
406
MEMORY
408
PROCESSING RESOURCE
410
DISK CONTROLLER
412
FILE / DATA
414
DATA TAMPER ENGINE
416
INSTRUCTIONS TO ACCESS FILE STORED IN ENCRYPTED VOLUME AND UNENCRYPTED VOLUME
418
INSTRUCTIONS TO PERFORM NEW DIGEST CALCULATION ON FILE
420
INSTRUCTIONS TO FETCH SAVED DIGEST CALCULATION FROM ENCRYPTED VOLUME
422
INSTRUCTIONS TO COMPARE NEW DIGEST CALCULATION WITH SAVED DIGEST
424
INSTRUCTIONS TO PERMIT ACCESS TO DATA COMPONENT ASSOCIATED WITH FILE STORED IN ENCRYPTED VOLUME
426
OUTPUT FILE TO FILE SYSTEM

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/040949

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/62(2013.01)i, H04L 9/08(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)
G06F 21/62; G06F 11/30; H04L 9/08; G06F 15/16; H04L 9/18; G06F 17/30; G06F 12/14; H04L 9/00

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: file, digest, calculation, tamper, encrypt, storage

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 8205094 B2 (CORINNE DIVE-RECLUS) 19 June 2012 See column 2, lines 21-24; column 3, lines 34-38; column 7, lines 15-32; and claims 1, 5, 8, 14.	1-15
A	US 2006-0130154 A1 (WAI LAM et al.) 15 June 2006 See paragraphs [0016]-[0017]; and figure 1.	1-15
A	US 2010-0205446 A1 (MICHAEL FREDERICK KENRICH et al.) 12 August 2010 See paragraphs [0044]-[0045]; and figure 5.	1-15
A	US 2009-0158037 A1 (CHUNG-I LEE et al.) 18 June 2009 See paragraphs [0013]-[0014]; and figure 1.	1-15
A	US 7039713 B1 (DAVID VAN GUNTER et al.) 02 May 2006 See column 2, lines 3-32; and figure 6.	1-15

☐ 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
12 April 2016 (12.04.2016)	**12 April 2016 (12.04.2016)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 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/US2015/040949

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 8205094 B2	19/06/2012	AT 326718 T	15/06/2006
		AU 2003-234040 A1	12/12/2003
		DE 60305315 D1	22/06/2006
		DE 60305315 T2	08/02/2007
		DK 1512060 T3	18/09/2006
		EP 1512060 A1	09/03/2005
		EP 1512060 B1	17/05/2006
		ES 2265105 T3	01/02/2007
		GB 0212318 D0	10/07/2002
		GB 0312199 D0	02/07/2003
		GB 2390918 A	21/01/2004
		GB 2390918 B	27/10/2004
		JP 2005-527905 A	15/09/2005
		JP 2010-205270 A	16/09/2010
		JP 4526383 B2	18/08/2010
		JP 4975127 B2	11/07/2012
		US 2005-0216907 A1	29/09/2005
		WO 03-100583 A1	04/12/2003
US 2006-0130154 A1	15/06/2006	None	
US 2010-0205446 A1	12/08/2010	US 7707427 B1	27/04/2010
		US 8301896 B2	30/10/2012
US 2009-0158037 A1	18/06/2009	CN 101459661 A	17/06/2009
		CN 101459661 B	16/05/2012
US 7039713 B1	02/05/2006	None	