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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 June 2006 (22.06.2006)

PCT

(10) International Publication Number
WO 2006/065626 A1

(51) International Patent Classification:
G11B 20/10 (2006.01) *G06F 12/14* (2006.01)
G11B 20/12 (2006.01)

(21) International Application Number:
PCT/US2005/044468

(22) International Filing Date:
7 December 2005 (07.12.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/636,360 14 December 2004 (14.12.2004) US
11/295,214 5 December 2005 (05.12.2005) US

(71) Applicant *(for all designated States except US)*: **NETWORK APPLIANCE, INC.** [US/US]; 495 East Java Drive, Sunnyvale, CA 94089 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **McGOVERN, William, P.** [US/US]; 495 East Java Drive, Sunnyvale, CA 94089 (US).

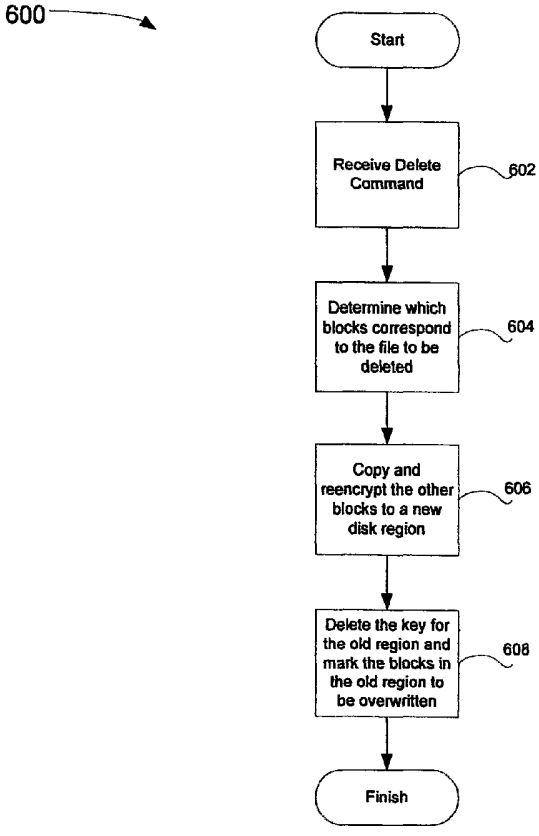
(74) Agents: **VINCENT, Lester, J.** et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 7th Floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT,

[Continued on next page]

(54) Title: RENDERING DISK DATA UNRECOVERABLE USING ENCRYPTION



(57) Abstract: A technique for sanitizing data storage devices, such as magnetic disks, is disclosed. Logical data storage units such as files or portions thereof may be individually deleted and sanitized on a disk. A disk is divided into physical disk regions, each comprising one or more blocks. The contents of the disk are encrypted using a separate encryption key for each physical disk region. If a file or other data structure located in a first disk region and encrypted using a first encryption key is to be deleted, the logical portions (i.e., blocks) of that region that do not belong to the file are re-encrypted using a second encryption key, and the first encryption key is deleted.

WO 2006/065626 A1

RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

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

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

RENDERING DISK DATA UNRECOVERABLE USING ENCRYPTION

[0001] This application claims the benefit of U.S. provisional patent application no. 60/636,360, filed on December 14, 2004 and entitled, "Disk Sanitation Using Encryption," which is incorporated herein by reference.

FIELD OF THE INVENTION

[0002] The invention generally relates to data storage and, more specifically, the invention relates to sanitization of disks using encryption.

BACKGROUND

[0003] When data is deleted from a magnetic disk such as a hard drive, the data can often be recovered. A hard drive typically comprises many addressable "blocks," or individual units of data. A file or other logical data storage unit typically comprises data written to the blocks and an entry in a file structure that includes pointers that point to the blocks storing the data. The delete function of many file systems only removes the pointers; the data itself remains intact. Even after a low-level format of a hard drive, data stored on the drive may be recoverable. In many applications, however, it may be undesirable for certain data (e.g., sensitive or confidential data) to remain recoverable after deletion. In such applications, it may be desirable to "sanitize" a mass storage device, such as a magnetic disk.

[0004] Generally, sanitization involves making the data blocks on the disk unreadable so that sensitive data is unrecoverable. Several techniques for "sanitizing" a magnetic disk are known today. One way to sanitize a hard drive is to physically destroy the drive. For example, the drive may be dismantled or otherwise physically altered. Another physical method is to degauss the disk by applying a powerful alternating magnetic

field to the disk. The degaussing technique changes the orientation of the magnetic particles on the disk platter.

[0005] If the drive is to be reused, it can be sanitized by writing over the data already on the disk. This approach is known as "media overwrite" sanitization. Media overwrite sanitization may be as simple as writing logic zeros to every bit on a drive, or writing different predetermined or random patterns to the drive. Writing over the drive once is known as a "single pass" overwrite. Writing over the drive multiple times is known as "multiple pass" overwrite. Different users require different levels of sanitization. For example, a user storing sensitive information, such as confidential trade secrets, may want to perform a greater number of passes.

[0006] Several different "patterns" have been developed to perform media overwrite sanitization. A pattern is the sequence of bits (ones and zeros) that is written to the drive in order to prevent recovery of deleted data. The "Guttman" pattern is an example of a pattern used by many hard drive sanitization applications. Using a multiple pass overwrite, different patterns may be used for each pass. For example, the first pass may use a particular pattern, where the second pass may use the pattern's complement, and the third pass may use random data.

[0007] **Figure 1** illustrates an example of a system 100 that can be used for storing data. The system 100 includes a storage server 102 that manages a volume 104 which comprises one or more physical magnetic drives. The data stored in the volume 104 is encrypted. The encrypted data can only be accessed using an appropriate encryption key, which can be a random series of bits (typically between 40 and 512 bits) used to encode the data stored on the volume 104. The clients 106 may access the volume 104, to read to write data, through a cryptographic interface 108 and the storage server 102. The read and write requests and

associated data are encrypted and decrypted by the cryptographic interface 108.

[0008] The volume 104 can be sanitized by discarding the encryption key, since the data on the volume 104 is unreadable without they key, and typical encryption standards (such as the advanced encryption standard (AES) encryption standards including AES-256 and AES-512) are computationally infeasible to compromise with currently existing technology. Removing the cryptographic interface 108 would provide further protection against undesired recovery of the deleted (encrypted) data.

[0009] A shortcoming of known sanitization techniques and system 100 is that they sanitize only at a disk-wide level of granularity. When a hard drive or other storage device is being retired or removed from use, the entire device (e.g., the entire disk drive) is sanitized to protect the data. However, in some instances, it may be desirable to sanitize only a portion of the disk drive. For example, storage users that are subject to government regulations regarding the retention of data may want to delete and sanitize only the particular files that the users are permitted to delete. The regulations may require that the user retain the other files. In some instances, it may be desirable to be able to sanitize only a portion of a file, e.g., only certain blocks. Further, in the system 100 of Figure 1, the cryptographic interface 108 undesirably introduces additional hardware between the storage server 102 and the clients 106.

SUMMARY

[0010] The present invention includes a method and apparatus for disk sanitization using encryption. The method includes encrypting data stored on a disk, including a set of blocks, by using a first encryption key. The method further includes, in response to a request to delete the set of

blocks, re-encrypting blocks stored on the disk other than the set of blocks, by using a second encryption key, and not re-encrypting the set of blocks, and deleting the first encryption key.

[0011] Other aspects of the invention will be apparent from the accompanying figures and from the detailed description which follows.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] One or more embodiments of the present invention are illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

[0013] **Figure 1** illustrates a system for sanitizing a magnetic data storage device such as a hard drive;

[0014] **Figure 2** illustrates a storage server that may be used to implement embodiments of the present invention;

[0015] **Figure 3** shows the architecture of a storage server according to certain embodiments of the invention;

[0016] **Figure 4** illustrates the operating system of the storage server of **Figure 3**, according to certain embodiments of the invention;

[0017] **Figure 5** illustrates physical disk regions including stored data; and

[0018] **Figure 6** is a flowchart describing a process for deleting and sanitizing a file, according to embodiments of the invention.

DETAILED DESCRIPTION

[0019] Described herein are methods and apparatuses for disk sanitization using encryption. Note that in this description, references to "one embodiment" or "an embodiment" mean that the feature being referred to is included in at least one embodiment of the present invention.

Further, separate references to "one embodiment" or "an embodiment" in this description do not necessarily refer to the same embodiment. However, such embodiments are also not mutually exclusive unless so stated, and except as will be readily apparent to those skilled in the art from the description. For example, a feature, structure, act, etc. described in one embodiment may also be included in other embodiments. Thus, the present invention can include a variety of combinations and/or integrations of the embodiments described herein.

[0020] According to embodiments of the invention, logical data storage units such as files or portions thereof (e.g., individual blocks) may be individually deleted and sanitized. The disk on which the files are located is divided into physical disk regions, where each region can include one or more blocks of data. The contents of the disk are encrypted using a separate encryption key for each physical disk region (which can mean a separate encryption key for each block, if a region is defined as a single block). If a file or other data structure located in a first disk region and encrypted using a first encryption key is to be deleted, the logical portions (i.e., blocks) of that region that do not belong to the file are re-encrypted using a second encryption key, and the first encryption key is then deleted. The first encryption key may be deleted using overwrite sanitization or other techniques. It is to be understood that the files or data structures being deleted may span more than one disk region.

[0021] **Figure 2** illustrates a storage server that may be used to implement embodiments of the present invention. A system 200 shows a storage server 202 having a client 204 connected thereto. Using a network attached storage (NAS) configuration, the client 204 may communicate with the storage server 202 using various communication protocols such as the Network File System (NFS) or Common Internet File System (CIFS) protocols.

[0022] The client 204 accesses and uses a volume 206 for data service. The volume 206 may comprise one or more drives, including one or more magnetic disks such as hard drives. The client 204 communicates with the volume 206 through an encryption layer 208. The encryption layer 208 encrypts and decrypts incoming and outgoing data. The volume 206 may be encrypted using several encryption keys. According to one embodiment of the invention, each physical disk region in the volume 206 has a unique encryption key. The encryption layer 208 includes hardware and software components to aid in encryption and to store the encryption keys. The encryption layer 208 includes a cryptographic key database 210 that includes each unique encryption key for each disk region in the volume 206.

[0023] The storage server 202 may be any one of several different types of storage servers, including those that employ a NAS or Storage Area Network (SAN) approach. For example, the storage server 202 may be a filer server, or filer, that stores data in the form of files.

[0024] **Figure 3** shows the architecture of a filer 300 such as the storage server 202 according to certain embodiments of the invention. Note that certain standard and well-known components which are not germane to the present invention are not shown. The filer 300 includes a processor 302 and main memory 304, coupled together by a bus system 306. The bus system 306 in **Figure 3** is an abstraction that represents any one or more separate physical buses and/or point-to-point connections, connected by appropriate bridges, adapters and/or controllers. The bus system 306, therefore, may include, for example, a system bus, a Peripheral Component Interconnect (PCI) bus, a HyperTransport or industry standard architecture (ISA) bus, a small computer system interface (SCSI) bus, a universal serial bus (USB), or an

Institute of Electrical and Electronics Engineers (IEEE) standard 1394 bus (sometimes referred to as "Firewire").

[0025] The processor 302 is the central processing unit (CPU) of the filer 300 and, thus, controls the overall operation of the filer 300. In certain embodiments, the processor 302 accomplishes this by executing software stored in main memory 304. The processor 302 may be, or may include, one or more programmable general-purpose or special-purpose microprocessors, digital signal processors (DSPs), programmable controllers, application specific integrated circuits (ASICs), programmable logic devices (PLDs), or the like, or a combination of such devices.

[0026] The main memory 304, which is generally some form of random access memory (RAM), stores the operating system 308 of the filer 300. Techniques of the present invention may be implemented within the operating system 308, as described further below. The operating system 308 may be, for example, the ONTAP operating system by Network Appliance, Inc., of Sunnyvale, CA (NetApp®). Also connected to the processor 302 through the bus system 306 are a network adapter 310 and a storage adapter 312. The network adapter 310 provides the filer 300 with the ability to communicate with remote devices, such as clients and/or another filer, over a network and may be, for example, an Ethernet adapter. The storage adapter 312 allows the filer to access the external mass storage devices and may be, for example, a Fibre Channel (FC) adapter or SCSI adapter.

[0027] A cryptographic module 314 is coupled to the bus 306. The cryptographic module 314 may be considered a part of the encryption layer 208 of **Figure 2**. The cryptographic module 314 may be implemented in software or as a hardware accelerator, such as an integrated circuit (IC) mounted on a system motherboard, a daughterboard or adapter card. The cryptographic module 314 performs

the encryption and decryption functions of the system 300. The cryptographic module 314 accesses the cryptographic key database 210 (see **Figure 2**) that includes the several encryption keys for every disk region of the storage devices.

[0028] The master key 316 is an integrated circuit (IC) mounted on the motherboard or a daughterboard or expansion card of the system 300. The IC storing the master key 316 may implement physical security and tamper resistant measures, such as being encased in epoxy, to prevent compromise of the stored encryption key. The master key 316 contains a master encryption key that is used to encrypt the cryptographic key database 210. Since the master key 316 is a hardware component, the cryptographic key database 210 cannot be accessed by any system other than the system 200. Further, since the master key 316 cannot be removed and used in another system, the data encrypted using the master key 316 will remain secure.

[0029] **Figure 4** illustrates the operating system 308 of the filer 300 of **Figure 3**, according to certain embodiments of the invention. As can be seen, the operating system 308 includes a number of layers. The core of the operating system 308 is the file system 402. The file system 402 is a programmatic entity that imposes structure on an address space of one or more physical or virtual storage devices, such as disks, so that the operating system 308 may conveniently read and write data containers, such as files and blocks, and related metadata. The file system 402, among other responsibilities, executes read and write operations on the mass storage devices in response to client requests, maintains directories, and manages consistency point operations. An example of a file system suitable for this purpose is the Write Anywhere File Layout to (WAFL) file system from Network Appliance, such as used in the NetApp® Filers. The file system 402 in certain embodiments operates on

blocks of data of a predetermined exemplary size, such as 4 Kbytes. Also shown in **Figure 4** is the logical data path 404 from clients to mass storage devices, through the file system 402.

[0030] The operating system 308 also includes a user interface 406, through which a network administrator or other user can control and/or configure the filer (e.g., remotely from a management station). The user interface 406 may generate a command line interface and/or a graphical user interface for this purpose. On the client side the operating system 308 includes a network access layer 408 and, at the lowest level, a media access layer 410. The network access layer 408 implements any of various protocols used to communicate with client devices, such as network file system (NFS), common Internet file system (CIFS) and/or hypertext transport protocol (HTTP). The media access layer 410 includes one or more drivers which implemented the protocols used to communicate over the network, such as Ethernet.

[0031] On the storage device side, the operating system 308 includes a storage access layer 412 and, at the lowest level, a driver layer 414. The storage access layer 412 implements a disk storage protocol such as RAID, while the driver layer 414 implements a lower-level storage device access protocol, such as Fibre Channel or SCSI.

[0032] The file system 208 also includes a cryptographic module 416. The cryptographic module 416 interfaces with the cryptographic module 314 of **Figure 3** through the file system 402 and the storage access layer 412. The cryptographic module 416 sends instructions and data to and receives data form the cryptographic module 314. The cryptographic module 416 further provides an interface between the cryptographic key database 210 of **Figure 2** and the cryptographic module 314.

[0033] **Figure 5** illustrates an example of how data may be stored on a disk divided into multiple physical disk regions. A disk 500 is divided into

several physical disk regions 502. The disk 500 may comprise a part of the volume 206 of **Figure 2**. The disk regions 502 denote physical locations on the platters of the disk 500. Each disk region 502 may comprise a predetermined amount of storage, for example 2 megabytes (MB). Each disk region 502 may store one or more files (or other logical data structures). Each disk region 502 is encrypted using a unique encryption key. The encryption keys are stored in the cryptographic key database 210 (see **Figure 2**), which may be stored on the disk 500. The cryptographic key database is encrypted using the master key 316 (see **Figure 3**).

[0034] Each disk region 502 may further be divided into smaller physical denominations, such as blocks. For example, a first disk region 502a comprises several blocks 504. Each block 504 may be a file or a portion of a file, such as a 4 Kbyte segment of a file. Note that a region 502 can also be defined as a single block. A file is a logical structure that includes pointers to physical blocks on the disk 500. For example, the data comprising the file 'A' is stored in the blocks 504c, 504d, and 504e. When referencing the file 'A', the file system follows several pointers to the blocks 504c, 504d, and 504e. It is understood that although the file 'A' is shown in contiguous blocks 504c, 504d, and 504e, that the blocks of a specific file may be stored in a noncontiguous fashion. For example, the file 'C' is stored in the two nonadjacent blocks 504g and 504i. It is further understood that a file may include blocks from several disk regions 502. However, for simplicity, the file 'A' is described in terms of the three blocks 504c, 504d, and 504e.

[0035] **Figure 6** is a flowchart describing a process 600 for deleting a file and sanitizing a file, in accordance with the technique introduced here. The process 600 will be described with reference to the example of **Figure 5**. The process 600 is described in terms of a file 'A' stored in

three blocks 504c, 504d, and 504e in a first disk region 502a. The first disk region 502a is encrypted using a first encryption key that is stored in the cryptographic key database 210 (see **Figure 2**).

[0036] In operation 602, a delete command is received by the operating system 208 (see **Figure 2**). The delete command may be issued by a user, an application, a system process, etc. According to one embodiment, an application may want to delete certain data at specific times. For example, a compliant data storage system may want to delete compliant data as soon as it expires. The delete command may be issued to delete a file or other data structure. According to the example shown here, a request to delete the file 'A' is received by the operating system 308 (see **Figure 3**).

[0037] In operation 604, it is determined which blocks correspond to the file (or other data structure) to be deleted. As can be seen in **Figure 6**, the file 'A' occupies the set of blocks 504c, 504d, and 504e. Other files have data stored in the remaining set of blocks 504a, 504b, and 504f-504i. A file can be stored using blocks in more than one disk region 502. However, the file 'A' is shown stored in only the first disk region 502a for clarity.

[0038] In operation 606, the remaining set of blocks in disk region 502a (i.e., blocks 502a, 502b, and 504f-504i, but not blocks 504c, 504d and 504e) are re-encrypted in place, by using a second (new) encryption key for disk region 502a. During the re-encryption process, the blocks are first decrypted using the first encryption key, and then encrypted using the new encryption key. The blocks 504c-504e comprising the file 'A' are not re-encrypted.

[0039] In operation 608, the first encryption key is deleted, and the blocks 504c-504e representing the file are marked as available, so that they may be overwritten. The first encryption key is deleted by overwriting

the key in the cryptographic key database. The first encryption key may be overwritten using any one of several known techniques, including using any one of the well-known patterns (such as a "Guttman" pattern), writing zeroes over the key ("zeroing" the key), etc. According to one embodiment, the first encryption key may be overwritten only once, since the data that comprises an encryption key must be completely intact to be useful. Therefore, more robust sanitization techniques may not be necessary. Further, the cryptographic key database is encrypted using the master key 316 (see **Figure 3**). The master key 316 is physically located within the system 300 of **Figure 3**, and therefore a user or client must have access to the system 300 to gain access to the first encryption key. If a malicious user only has access to the disk 500 (see **Figure 5**), the user will be unable to decrypt the cryptographic key database, and therefore will be unable to obtain the first encryption key. Once the first encryption key is discarded, the file 'A' has been sanitized, even though the blocks 504c-504e may not have been overwritten, since the data stored in the disk region 502a is unreadable without the first encryption key.

[0040] The technique introduced above has been described in the context of a network attached storage (NAS) environment. However, these techniques can also be applied in various other contexts. For example, the techniques introduced above can be applied in a storage area network (SAN) environment. A SAN is a highly efficient network of interconnected, shared storage devices. One difference between NAS and SAN is that in a SAN, the storage server (which may be an appliance) provides a remote host with block-level access to stored data, whereas in a NAS configuration, the storage server provides clients with file-level access to stored data. Thus, the techniques introduced above are not limited to use in a file server or in a NAS environment.

[0041] Software to implement the technique introduced here may be stored on a machine-readable medium. A "machine-accessible medium", as the term is used herein, includes any mechanism that provides (i.e., stores and/or transmits) information in a form accessible by a machine (e.g., a computer, network device, personal digital assistant (PDA), manufacturing tool, any device with a set of one or more processors, etc.). For example, a machine-accessible medium includes recordable/non-recordable media (e.g., read-only memory (ROM); random access memory (RAM); magnetic disk storage media; optical storage media; flash memory devices; etc.), etc.

[0042] The term "logic", as used herein, can include, for example, hardwired circuitry, programmable circuitry, software, or any combination thereof.

[0043] This invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident to persons having the benefit of this disclosure that various modifications changes may be made to these embodiments without departing from the broader spirit and scope of the invention. The specification and drawings are accordingly to be regarded in an illustrative rather than in a restrictive sense.

CLAIMS:

What is claimed is:

1. A method of disk sanitization comprising:

encrypting data stored on a disk, including a set of blocks, by using a first encryption key; and

in response to a request to delete the set of blocks,

re-encrypting blocks stored on the disk other than the set of blocks, by using a second encryption key, and not re-encrypting the set of blocks; and

deleting the first encryption key.

2. The method of claim 1, wherein the disk comprises a plurality of regions, and wherein said encrypting data stored on the disk comprises encrypting each of the plurality of regions with a different encryption key.

3. The method of claim 1, wherein deleting the first encryption key comprises:

overwriting the first encryption key.

4. The method of claim 3, wherein overwriting the first encryption key comprises:

overwriting a portion of a cryptographic key database including the first encryption key, wherein the cryptographic key database is encrypted using a master key.

5. The method of claim 1, wherein re-encrypting blocks stored on the disk other than the set of blocks comprises using a cryptographic module.

6. The method of claim 1, wherein the set of blocks comprises a file.

7. The method of claim 1, wherein re-encrypting blocks stored on the disk other than the set of blocks comprises:

decrypting blocks stored on the disk other than the set of blocks, by using the first encryption key; and

encrypting the blocks stored on the disk other than the set of blocks, by using the second encryption key.

8. The method of claim 1, further comprising marking the set of blocks so that the first disk region may be overwritten.

9. The method of claim 1, wherein the set of blocks represents a file.

10. The method of claim 1, wherein the set of blocks represents a portion of a file.

11. A system to perform disk sanitization, the system comprising:

a disk including data encrypted using a first encryption key; and

a processor configured to respond to a command to delete a first set of encrypted blocks on the disk, the first set of encrypted blocks representing a file or a portion thereof, by:

decrypting blocks on the disk other than the first set of encrypted blocks,

using a second encryption key to re-encrypt the blocks on the disk other than the first set of encrypted blocks, and not decrypting the first set of encrypted blocks, and

deleting the first encryption key.

12. The system of claim 11, wherein the disk comprises a plurality of regions, and wherein the processor is further configured to encrypt data stored on the disk by encrypting each of the plurality of regions with a

different encryption key, and wherein the first set of blocks is contained within a single region of the plurality of regions.

13. The system of claim 11, wherein the processor is at least part of a cryptographic module.

14. The system of claim 11, wherein the first and second cryptographic keys are stored in a cryptographic key database on the disk.

15. The system of claim 14, wherein the processor deletes the first encryption key by performing a media overwrite of the first encryption key.

16. The system of claim 15, wherein the media overwrite is a single pass overwrite.

17. The system of claim 14, wherein the cryptographic database is encrypted using a master key.

18. The system of claim 17, wherein the master key is accessible to the cryptographic module.

19. A method for performing media sanitization of a disk that includes a plurality of regions, the method comprising:

encrypting data on the disk by using a different encryption key to encrypt each of the plurality of regions, the data on the disk including a file, wherein said encrypting includes encrypting the file using a first encryption key;

storing the first encryption key in a cryptographic key database;

receiving a command to delete the file; and

in response to the command to delete the file,

identifying a first set of blocks on the disk which belong to the file and a second set of blocks on the disk which do not belong to the file;

re-encrypting the second set of blocks by using a second encryption key and not re-encrypting the first set of blocks; and

deleting the first encryption key by overwriting a portion of the cryptographic key database corresponding to the first encryption key.

20. The method of claim 19, further comprising:

encrypting the cryptographic key database using a master key.

21. The method of claim 19, further comprising:

marking the first set of blocks so that the first set of blocks may be overwritten.

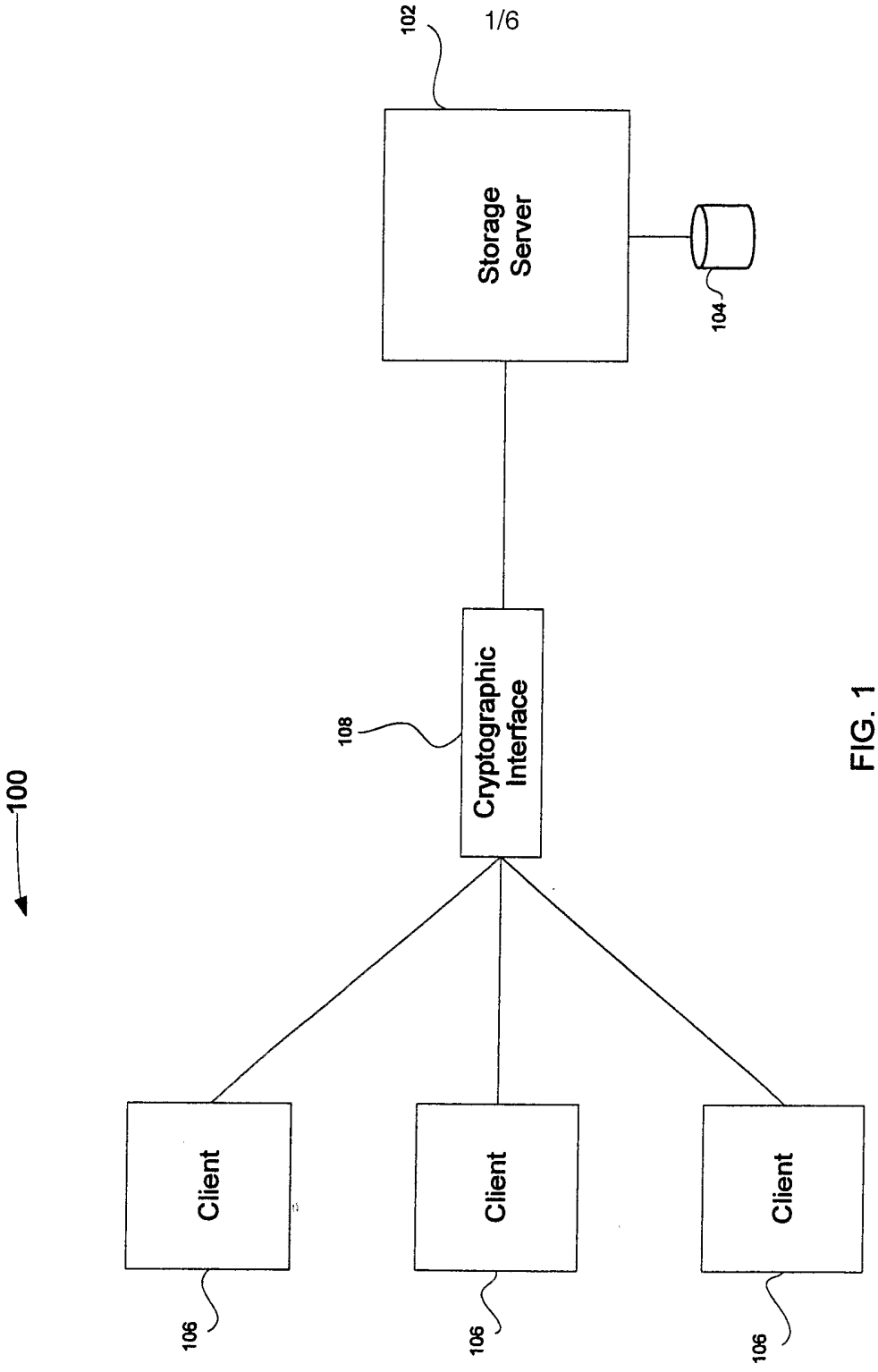

100
102
Storage Server
104
108
Cryptographic Interface
Client
Client
Client
106
106
106

FIG. 1

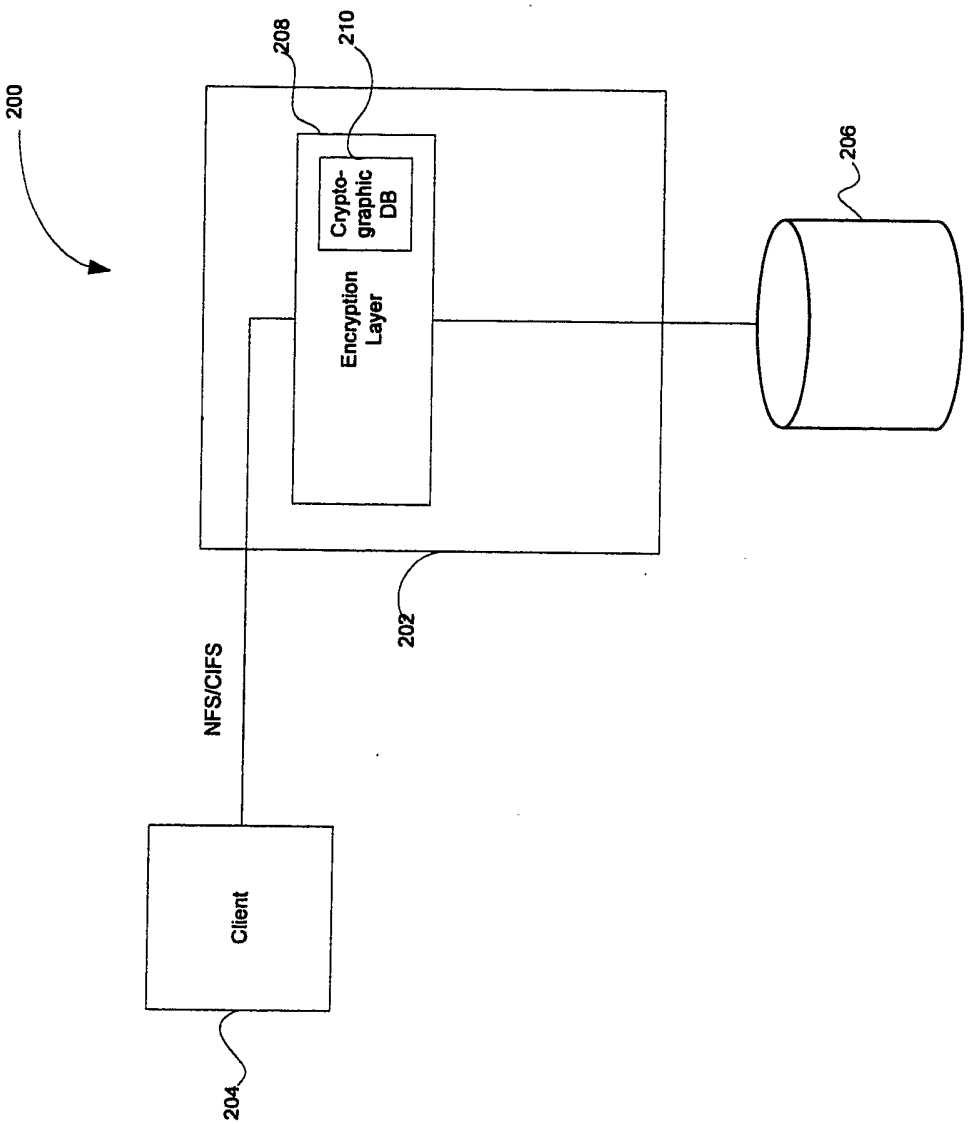

200
208
210
Crypto-graphic DB
Encryption Layer
206
202
NFS/CIFS
Client
204

FIG. 2

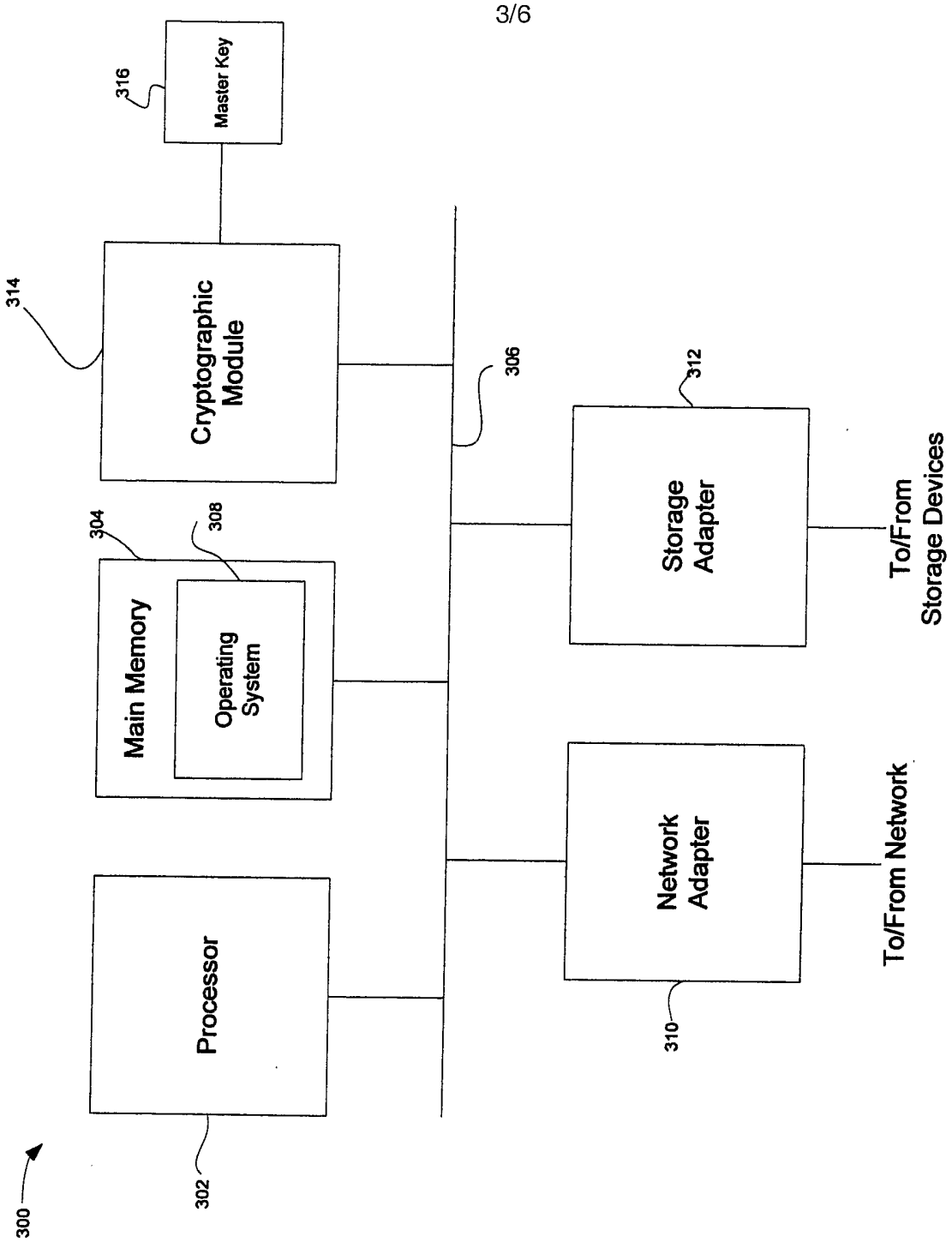

Master Key
316
Cryptographic Module
314
306
312
Storage Adapter
To/From Storage Devices
304
308
Main Memory
Operating System
Network Adapter
To/From Network
Processor
310
302
300

FIG. 3

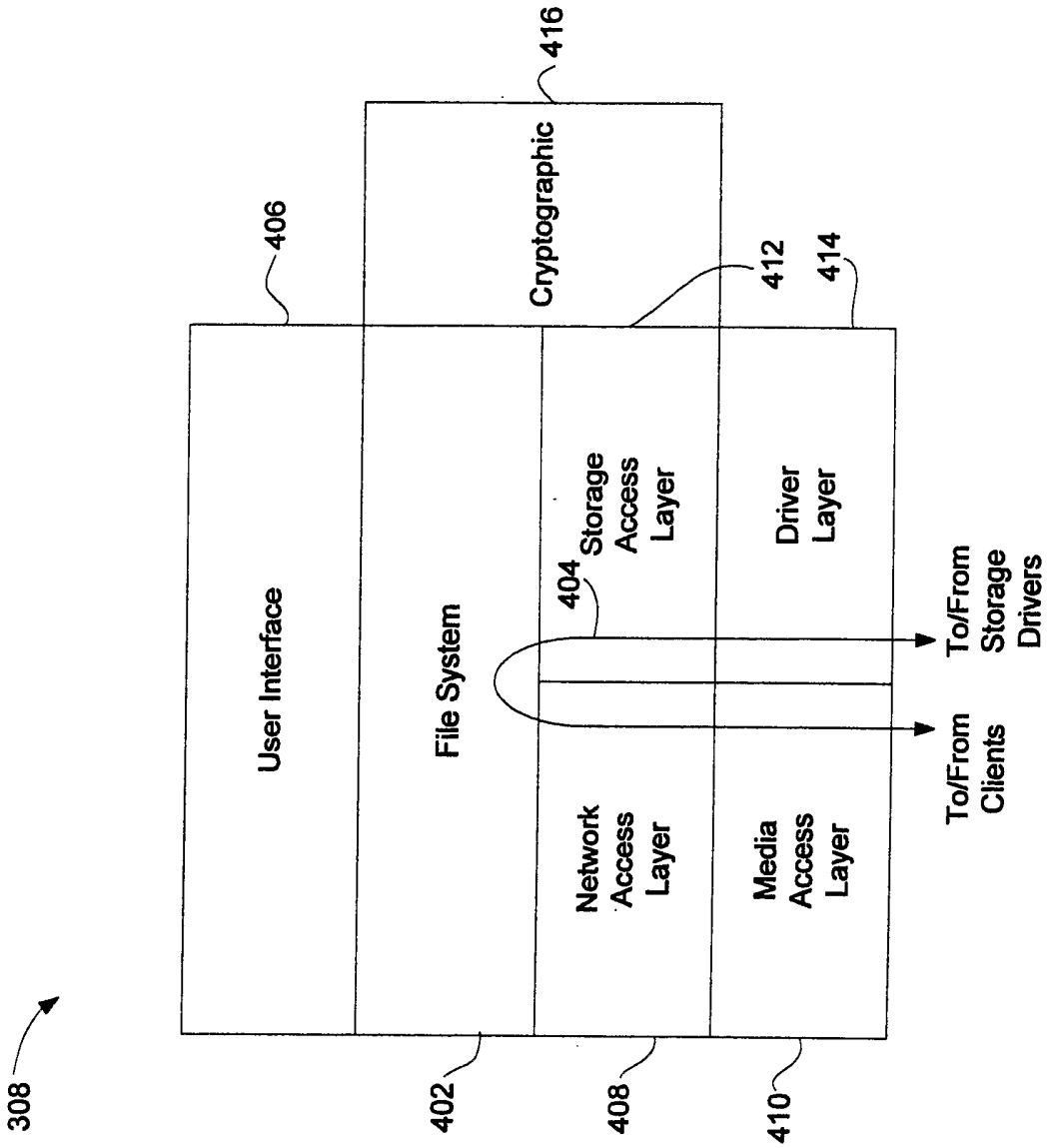

308
User Interface
406
File System
402
Cryptographic
416
Storage Access Layer
404
Network Access Layer
408
412
Driver Layer
414
Media Access Layer
410
To/From Storage Drivers
To/From Clients

FIG. 4

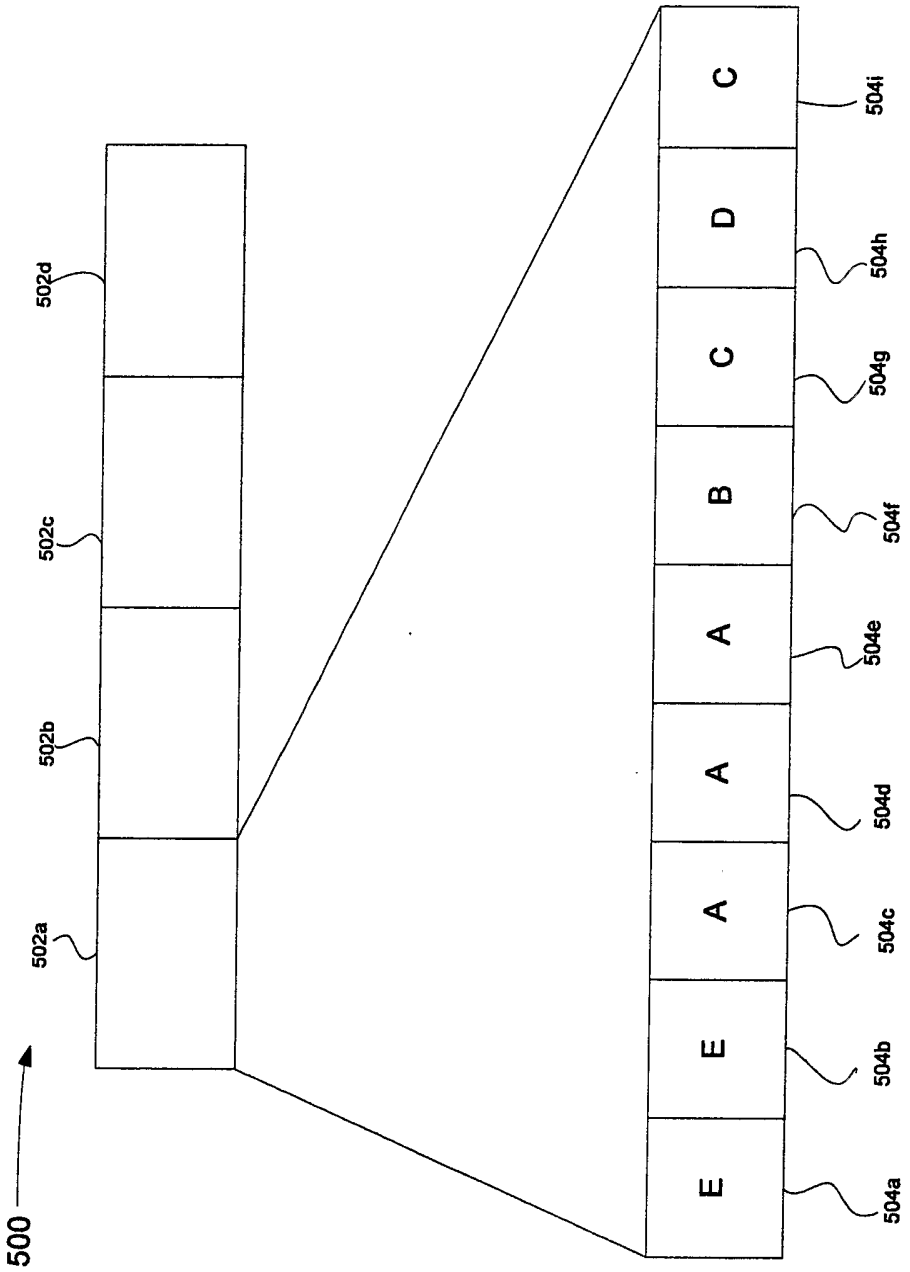

500
502a
502b
502c
502d
E
E
A
A
A
B
C
D
C
504a
504b
504c
504d
504e
504f
504g
504h
504i

FIG. 5

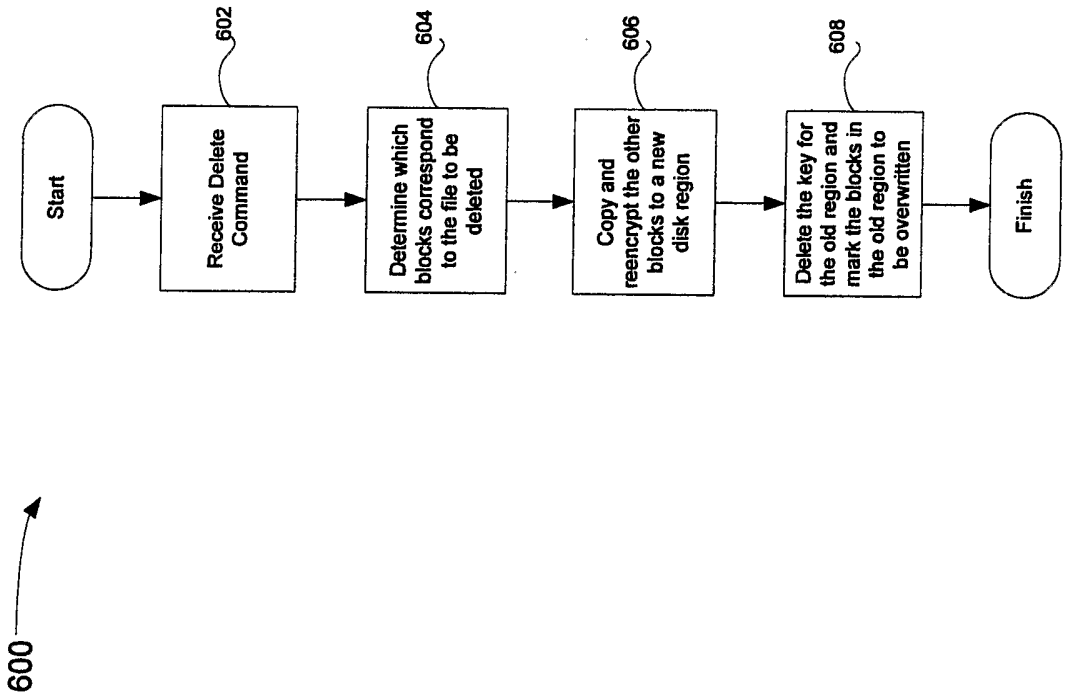

600
Start
Receive Delete Command
602
Determine which blocks correspond to the file to be deleted
604
Copy and reencrypt the other blocks to a new disk region
606
Delete the key for the old region and mark the blocks in the old region to be overwritten
608
Finish

FIG. 6

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2005/044468

A. CLASSIFICATION OF SUBJECT MATTER
INV. G11B20/10 G11B20/12 G06F12/14

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)
G11B G06F

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

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 03/067438 A (INFINEON TECHNOLOGIES AG; GAMMEL, BERNDT, M; KUENEMUND, THOMAS; SEDLAK) 14 August 2003 (2003-08-14) page 2, line 1 - line 14	1-21
A	EP 0 869 460 A (PITNEY BOWES INC) 7 October 1998 (1998-10-07) page 8, line 40 - page 9, line 4	1-21
A	EP 1 233 414 A (MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD) 21 August 2002 (2002-08-21) paragraphs [0021], [0075] paragraphs [0170], [0174], [0203]	1-21

-/—

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
27 April 2006	12/05/2006

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Ogor, M

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2005/044468

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6 134 660 A (BONEH ET AL) 17 October 2000 (2000-10-17) abstract column 5, line 63 - column 6, line 2 column 6, line 37 - line 41	1-21
A	EP 1 122 910 A (MITSUBISHI CORPORATION) 8 August 2001 (2001-08-08) paragraphs [0018], [0019] paragraphs [0021] - [0023]	1,11,19
A	US 2003/005313 A1 (GAMMEL BERNDT ET AL) 2 January 2003 (2003-01-02) paragraph [0039]	1,11,19
A	EP 1 058 254 A (MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD) 6 December 2000 (2000-12-06) paragraphs [0188], [0189]; figure 35	1,11,19

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2005/044468

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 03067438 A	14-08-2003	DE 10205316 A1	28-08-2003
		EP 1472606 A2	03-11-2004
		TW 591924 B	11-06-2004
		US 2005044392 A1	24-02-2005
EP 0869460 A	07-10-1998	CA 2231082 A1	04-09-1998
		DE 69815575 D1	24-07-2003
		DE 69815575 T2	29-04-2004
		US 6131090 A	10-10-2000
EP 1233414 A	21-08-2002	CN 1371081 A	25-09-2002
		JP 2002244926 A	30-08-2002
		US 2002126843 A1	12-09-2002
US 6134660 A	17-10-2000	NONE	
EP 1122910 A	08-08-2001	AU 6123599 A	01-05-2000
		CA 2347480 A1	20-04-2000
		CN 1330819 A	09-01-2002
		WO 0022777 A1	20-04-2000
		JP 2002101089 A	05-04-2002
US 2003005313 A1	02-01-2003	AT 249664 T	15-09-2003
		CN 1423801 A	11-06-2003
		WO 0154083 A1	26-07-2001
		DE 50003679 D1	16-10-2003
		ES 2207567 T3	01-06-2004
		JP 2003521053 T	08-07-2003
EP 1058254 A	06-12-2000	CN 1355919 A	26-06-2002
		DE 60011958 D1	12-08-2004
		DE 60011958 T2	25-08-2005
		WO 0067257 A2	09-11-2000
		TW 540039 B	01-07-2003
		US 6938162 B1	30-08-2005