



- (51) **International Patent Classification:**
G06F 21/62 (2013.01) *G06F 3/06* (2006.01)
- (21) **International Application Number:**
PCT/MY2013/000156
- (22) **International Filing Date:**
5 September 2013 (05.09.2013)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
PI2012004046 11 September 2012 (11.09.2012) MY
- (71) **Applicant:** MIMOS BERHAD [MY/MY]; Technology Park Malaysia, 57000 Kuala Lumpur (MY).
- (72) **Inventors:** SEA CHONG SEAK; C/o Mimos Berhad, Technology Park Malaysia, 57000 Kuala Lumpur (MY). NOR IZYANI DAUD; c/o Mimos Berhad, Technology Park Malaysia, 57000 Kuala Lumpur (MY).
- (74) **Agent:** MIRANDAH, Patrick; Patrick Mirandah Co. (malaysia) Sdn Bhd, Suite 3b-19-3 Plaza Sentral, Jalan Stesen Sentral 5, 50470 Kuala Lumpur (MY).
- (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, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

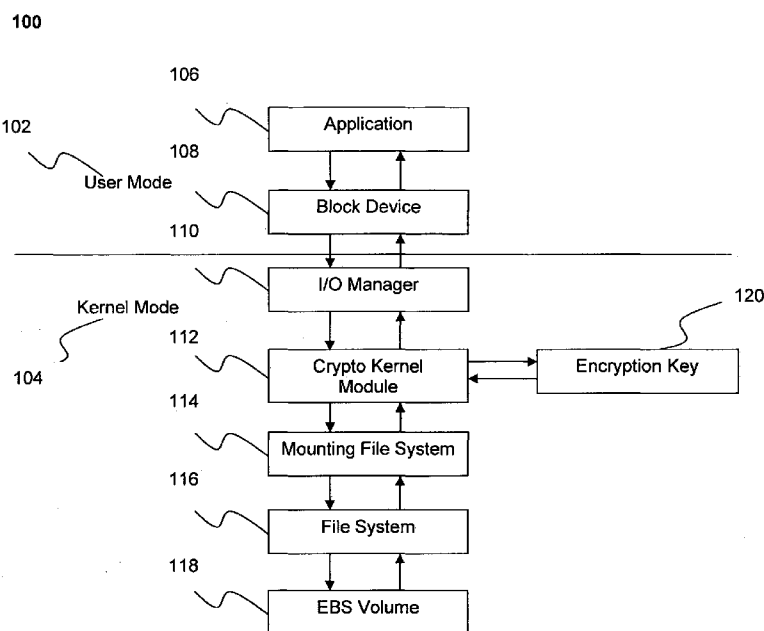
Declarations under Rule 4.17:

- *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))*

(54) **Title:** MANAGEMENT OF STORAGE ENCRYPTION OVER NETWORK-BASED ELASTIC BLOCK STORE VOLUME



(57) **Abstract:** The system and method of the present invention manages storage encryption over network-based or cloud based Elastic Block Store (EBS) Volume so as to provide flexibility and transparency on the encryption algorithm that can be universally acceptable to any provider and user. The system (100) of the present invention manages storage encryption over network based Elastic-Block Store (EBS) Volume (118) with a user mode (102) and a kernel mode (104); said user mode (102) comprising of an application (106) and a block device (108); said kernel mode (104) comprising of at an I/O Manager (110) for receiving mounting instruction and presenting decrypted data to requester; a crypto kernel module (112) for retrieving encryption key from key file; a file system (116) an EBS Volume (118). The methodology of the present invention is being initiated by initializing and configuring EBS Volume (202). Thereafter, encrypted EBS Volume (204) is mounted on the system and data is written on EBS Volume and said data is further encrypted (206). The encrypted data is read from EBS Volume and said encrypted data will be decrypted to obtain decrypted data

(208). The encrypted EBS Volume (210) can be unmounted from the system when the volumes are not required. The present invention is transparent to user in which the data that is written to the EBS disk are encrypted transparently with the random generation of encryption key and stored in the disk. Confidentiality is allowed and reinforced in the EBS Volume by encryption of the data being stored in the data storage and decryption of the encrypted data using correct symmetric key provided by the authorized user during retrieval process.

MANAGEMENT OF STORAGE ENCRYPTION OVER NETWORK-BASED ELASTIC BLOCK STORE VOLUME

FIELD OF INVENTION

5

The present invention relates to a system and method for managing storage encryption over network-based or cloud based Elastic Block Store (EBS) Volume to provide flexibility and transparency on the encryption algorithm that can be universally acceptable to any provider and user.

10

BACKGROUND ART

15

Cloud computing environment has two major types of data storage. The first type of storage is known as Instance Storage, which is non-persistent and data will be loss after an instance termination. The other type is persistent network-based Elastic Block Store (EBS) Volume, which can be attached to any running instances or also can be used as a persistent boot medium. EBS Volumes are off-instance storage that persists independently from the life of Virtual Machine (VM) instance.

20

Elastic Block Store (EBS) provides highly available and highly reliable storage volumes that can be attached to a running VM instance and exposed as a device within the VM instance. EBS is particularly suited for applications that require a database, file system, or access to raw block level storage. EBS also allows user to create storage volumes in any sizes that can be mounted and attached as devices by VM instances. Multiple volumes can be mounted on the same VM instance. The storage volumes behave like raw, unformatted block devices and the user is required to supply device names and block device interface. User can create a file system on top of the EBS Volumes, or use them in any other way as a block device (just like a normal hard drive).

25

30

At present, data stored in cloud computing environment using EBS is not encrypted; the administrator of EBS may possibly view and/or open and/or modify the data contents. Users are not confident to keep important data in the EBS volume due to the low security level.

The present invention proposes a system and method to provide confidentiality as only data owner or authorized users are allowed to view the contents in a cloud computing environment. In the present invention, when the system creates or writes a file to the storage encryption in EBS volume, the data which writes to EBS disk are encrypted transparently with the random generation of encryption key and stored in the disk. The encryption key is stored in the key file and said encryption key will be protected by symmetric key provided by the user. When the correct symmetric key is used to decrypt the encryption key, this encryption key will be used to decrypt data read from the disk. The encryption data will be decrypted transparently by the system and return to user mode application.

The subject matter claimed herein is not limited to embodiments that solve any disadvantages or that operate only in environments such as those described above. Rather, this background is only provided to illustrate one exemplary technology area where some embodiments described herein may be practice.

SUMMARY OF INVENTION

The present invention provides a system (100) and a method (200) for managing storage encryption over network based Elastic-Block Store (EBS) Volume. One aspect of the present invention provides a system (100) for managing storage encryption over network based Elastic-Block Store (EBS) Volume (118) with at least one user mode (102) and at least one kernel mode (104); said user mode (102) comprising of at least one application (106) and at least one block device (108); said kernel mode (104) comprising of at least one I/O Manager (110) for receiving mounting instruction and presenting decrypted data to requester; at least one crypto kernel module (112) for retrieving encryption key from key file and performing encryption and decryption on data; at least one file system (116) and at least one EBS Volume (118). The at least one crypto kernel module (112) for retrieving encryption key from key file having means for hashing password (P) through user input to obtain symmetric key to decrypt encrypted key from key file; calculate message digest (H) by hashing received password (P); reading key file to retrieve encrypted key (EK); and retrieving encrypted key (R) using H as a symmetric decryption key.

Another aspect of the present invention provides for a method (200) for managing storage encryption over network based Elastic-Block Store (EBS) Volume. The method comprises steps of initializing and configuring EBS Volume (202); mounting encrypted EBS Volume (204); writing data on EBS Volume and encrypting data (206); reading encrypted data from EBS Volume and decrypting said encrypted data (208); and unmounting encrypted EBS Volume (210). The methodology for initializing and configuring EBS Volume further comprises steps for random generation of encryption key wherein said encryption key encrypts data for writing on EBS volume and said encryption key is used to decrypt data which are decrypted by the system.

A further aspect of the present invention provides a further method (300) for initializing and configuring EBS Volume. The said method comprises steps of selecting EBS Volume (302); selecting encryption algorithm (304); loading Encryption Kernel Module (306); generating at least one encryption key (308); storing encryption key file in Virtual Machine (VM) (310); and initializing and formatting EBS Volume (312). The

method for generating at least one encryption key further comprises steps of receiving password (P) through user input (402); calculating message digest (H) by hashing received password (P) (404); generating encryption key (R) randomly (406); encrypting encryption key (R) using message digest (H) to produce encrypted key (EK) (408); and storing the encrypted key into key file (410).

Another aspect of the present invention provides a further method (500) for mounting encrypted EBS Volume. The said method comprises steps of sending instruction to I/O Manager (502); retrieving encryption key from key file by Crypto Kernel Module (504); retrieving encryption key from key file (506); and mounting file system (508). The method for retrieving encryption key from key file by Crypto Kernel Module further comprises steps of receiving password (P) through user input (602); calculating message digest (H) by hashing received password (P) (604); reading key file to retrieve encrypted key (EK) (606); retrieving encrypted key (R) using (H) as a symmetric decryption key (608); and forwarding retrieved encryption key (R) to crypto kernel module (610).

A further aspect of the present invention provides a further method (700) for writing data on EBS Volume and encrypting data. The said method comprises steps of requesting to write data on EBS Volume (702); performing write system call to I/O Manager (704); forwarding data to crypto kernel module and encrypting data (706); forwarding encrypted data to file system (708); and storing encrypted data on EBS Volume (710). The method for encrypting data forwarded to crypto kernel module further comprises steps of retrieving encryption key (EK) (802); and encrypting data to obtain encrypted data (ED) using EK as a symmetric encryption key (804).

Another aspect of the present invention provides a further method for reading encrypted data from EBS Volume and decrypting said encrypted data. The said method comprises steps of requesting to read data from EBS Volume (902); performing read system call to I/O Manager (904); reading data from EBS Volume and decrypting data (906); forwarding decrypted data to I/O Manager (908); and presenting decrypted data to requester (910). The method for decrypting data upon

reading data from EBS Volume further comprises steps of retrieving encryption key (EK) (1002); and decrypting data to obtain decrypted data (DD) using EK as a symmetric decryption key (1004).

- 5 The present invention consists of features and a combination of parts hereinafter fully described and illustrated in the accompanying drawings, it being understood that various changes in the details may be made without departing from the scope of the invention or sacrificing any of the advantages of the present invention.

BRIEF DESCRIPTION OF ACCOMPANYING DRAWINGS

To further clarify various aspects of some embodiments of the present invention, a more particular description of the invention will be rendered by references to specific
5 embodiments thereof, which are illustrated in the appended drawings. It is appreciated that these drawings depict only typical embodiments of the invention and are therefore not to be considered limiting of its scope. The invention will be described and explained with additional specificity and detail through the accompanying drawings in which:

10 FIG. 1.0 illustrates the block diagram of the system of the present invention.

FIG. 2.0 is a flowchart illustrating the method for managing storage encryption over network based Elastic-Block Store (EBS) Volume.

15 FIG. 3.0 is a flowchart illustrating the method for initializing and configuring EBS Volume.

FIG. 4.0 is a flowchart illustrating the method for generating at least one encryption key.

FIG. 5.0 is a flowchart illustrating the method for mounting encrypted EBS Volume.

20

FIG. 6.0 is a flowchart illustrating the method for retrieving encryption key from key file by Crypto Kernel Module

FIG. 7.0 is a flowchart illustrating the method for writing data on EBS Volume and
25 encrypting data.

FIG. 8.0 is a flowchart illustrating the method for encrypting data forwarded to crypto kernel module.

30 FIG. 9.0 is a flowchart illustrating the method for reading encrypted data from EBS Volume and decrypting said encrypted data.

FIG. 10.0 is a flowchart illustrating the method for decrypting data upon reading data from EBS Volume.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention provides a system and method for managing storage encryption over network-based or cloud based Elastic Block Store (EBS) Volume. Hereinafter, this specification will describe the present invention according to the preferred embodiments. It is to be understood that limiting the description to the preferred embodiments of the invention is merely to facilitate discussion of the present invention and it is envisioned without departing from the scope of the appended claims.

Reference is first being made to FIG. 1.0. FIG. 1.0 illustrates the block diagram of the system of the present invention. As illustrated in FIG. 1.0, the system (100) of the present invention manages storage encryption over network based Elastic-Block Store (EBS) Volume (118) with a user mode (102) and a kernel mode (104). The said user mode (102) comprising of an application (106) and a block device (108) and the said kernel mode (104) comprising of Input Output (I/O) Manager (110), a crypto kernel module (112), a file system (116) and a EBS Volume (118).

EBS Volume (118) is an off-instance storage that persists independently from the life of Virtual Machine (VM) instance. EBS provides highly available, highly reliable storage volumes that can be attach to a running VM instance and exposed as a device within the VM instance. EBS is particularly suited for applications that require a database, a file system, or access to raw block level storage. EBS allows user to create storage volumes in any sizes that can be mounted as devices by VM instances. Multiple volumes can be mounted to the same VM instance. The storage volumes behave like raw, unformatted block devices, with user required to supplied device names and a block device interface. User can create a file system on EBS volumes, or use them in any other way as a block device (just like a normal hard drive).

The I/O Manager (110) of the present invention receives mounting instruction and presents decrypted data to requester while the crypto kernel module (112) retrieves encryption key from key file. The crypto kernel module (112) retrieves encryption key from key file by the following means wherein the said system receives user input (i.e. a password (P)). The said password is hashed ($H = \text{Hash}(P)$). Key file is read by retrieving the encrypted key (EK) and the encryption key (R) is retrieved using H as a symmetric

decryption key. Thereafter, the encryption key (R) is forwarded to the crypto kernel module.

Reference is now being made to FIG. 2.0, FIG. 3.0 and FIG. 4.0 respectively. FIG. 2.0 is a flowchart illustrating the method for managing storage encryption over network based Elastic-Block Store (EBS) Volume while FIG. 3.0 is a flowchart illustrating the method for initializing and configuring EBS Volume and FIG. 4.0 is a flowchart illustrating the method for generating at least one encryption key. The methodology for managing storage encryption over network based Elastic-Block Store (EBS) Volume is initiated by first initializing and configuring EBS Volume (202) wherein initializing and configuring EBS Volume further comprises random generation of encryption key wherein said encryption key encrypts data for writing on EBS volume and said encryption key is used to decrypt data which are decrypted by the system. EBS Volume is first selected (202) as a startup to initialize and configure EBS Volume. Thereafter, the encryption algorithm (304) is selected and Encryption Kernel Module is loaded (306) to generate encryption key (308). The encryption key is generated by the following means wherein the system receives user input (402) (i.e. a password (P)). The said password is hashed ($H = \text{Hash}(P)$). Encryption key (R) is generated randomly (406) and encrypted key (EK) is generated by encrypting randomly generated encryption key (R) using H as a symmetric encryption key (408) (i.e. Encrypted Key (EK) = $\text{Enc}_H(R)$). The encrypted key (ER) will be stored into the key file in the Virtual Machine (310, 410). Thereafter, EBS Volume will be initialized and formatted accordingly (312).

Upon initializing and formatting the EBS Volume, the encrypted EBS Volume (204) is mounted on the system. Reference is now being made to FIG. 5.0 and FIG. 6.0 respectively. FIG. 5.0 is a flowchart illustrating the method for mounting encrypted EBS Volume and FIG. 6.0 is a flowchart illustrating the method for retrieving encryption key from key file by Crypto Kernel Module. The methodology of mounting the encrypted EBS Volume comprises of first sending instruction to the I/O Manager (502). Thereafter, the encryption key is retrieved from the key file by Crypto Kernel Module (504) wherein user input (i.e. password (P)) is received (602) and the said password (P) is hashed (604) (i.e. $H = \text{Hash}(P)$). The key file is read by retrieving the encrypted key (EK) (606) and encryption key (R) is retrieved by using H as a symmetric decryption key (608) (i.e. Encryption Key (R) = $\text{Dec}_H(\text{EK})$). Thereafter, the retrieved encryption key (R) is

forwarded to crypto kernel module (610). Subsequently, the system of the present invention retrieves the encryption key from the key file (506) and mounts the file system (508).

- 5 Upon mounting the encrypted EBS Volume, data is written on EBS Volume and further data is being encrypted (206). Reference is now being made to FIG. 7.0 and FIG. 8.0 respectively. FIG. 7.0 is a flowchart illustrating the method for writing data on EBS Volume and encrypting data and FIG. 8.0 is a flowchart illustrating the method for encrypting data forwarded to crypto kernel module. Request to write data on EBS
10 Volume (702) is received and thereafter a write system call is initiated to I/O Manager (704). Thereafter, data is forwarded to crypto kernel module and the said data is encrypted (706) by retrieving the encryption key (EK) (802) and data is encrypted to obtain encrypted data (ED) using EK as a symmetric encryption key (804) (i.e. Encrypted Data (ED) = $\text{Enc}_{\text{EK}}(\text{data})$). The encrypted data is forwarded to the file system
15 (708) and store in the EBS Volume (710).

- Reference is now being made to FIG. 9.0 and FIG. 10.0 respectively. FIG. 9.0 is a flowchart illustrating the method for reading encrypted data from EBS Volume and decrypting said encrypted data and FIG. 10.0 is a flowchart illustrating the method for
20 decrypting data upon reading data from EBS Volume. Once the encrypted data is stored in the EBS Volume, the encrypted data will be read from the EBS Volume and said encrypted data will be decrypted accordingly (208). Request to read data from EBS Volume (902) is received and thereafter a read system call is initiated to I/O Manager (904). Thereafter, data will be read from EBS Volume and the said data is decrypted
25 (906) by retrieving the encryption key (EK) (1002) and data is decrypted to obtain decrypted data (DD) using EK as a symmetric decryption key (1004) (i.e. Decrypted Data (DD) = $\text{Dec}_{\text{EK}}(\text{ED})$). The decrypted data is forwarded to the I/O Manager (908) and said decrypted data (DD) will be presented to the requester. Subsequently, the encrypted EBS Volume (210) can be unmounted from the system if the volumes are not
30 required.

The present invention provides management of storage encryption over network-based EBS Volume in which data being encrypted by encryption algorithm. The present invention provides flexibility on encryption algorithm in which the said algorithm is not

limited to any provider. Further, the present invention is transparent to user in which the data that is written to the EBS disk are encrypted transparently with the random generation of encryption key and stored in the disk. The present invention allows and re-enforce confidentiality in the EBS Volume by encryption of the data being stored in the data storage and decryption of the encrypted data using correct symmetric key provided by the authorized user during retrieval process.

The present invention may be embodied in other specific forms without departing from its essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore indicated by the appended claims rather than by the foregoing description. All changes, which come within the meaning and range of equivalency of the claims, are to be embraced within their scope.

CLAIMS

1. A method (200) for managing storage encryption over network based Elastic-Block Store (EBS) Volume, the method comprising steps of:

5 initializing and configuring EBS Volume (202);
 mounting encrypted EBS Volume (204);
 writing data on EBS Volume and encrypting data (206);
 reading encrypted data from EBS Volume and decrypting said encrypted data (208); and
10 unmounting encrypted EBS Volume (210)
 characterized in that
 initializing and configuring EBS Volume further comprises steps for random generation of encryption key wherein said encryption key encrypts data for writing on EBS volume and said encryption key is used
15 to encrypt data which are decrypted by the system.

2. A method (300) according to Claim 2, wherein initializing and configuring EBS Volume further comprises steps of:

20 selecting EBS Volume (302);
 selecting encryption algorithm (304);
 loading Encryption Kernel Module (306);
 generating at least one encryption key (308);
 storing encryption key file in Virtual Machine (VM) (310); and
 initializing and formatting EBS Volume (312).

25

3. A method (400) according to Claim 3, wherein generating at least one encryption key further comprises steps of:

 receiving password (P) through user input (402);
 calculating message digest (H) by hashing received password (P) (404);
30 generating encryption key (R) randomly (406);
 encrypting encrypted key (EK) using H as a symmetric encryption key (408); and
 storing the encrypted key into key file (410).

30

4. A method (500) according to Claim 2, wherein mounting encrypted EBS Volume further comprises steps of:

5 sending instruction to I/O Manager (502);
 retrieving encryption key from key file by Crypto Kernel Module (504);
 retrieving encryption key from key file (506); and
 mounting file system (508).

5. A method (600) according to Claim 5, wherein retrieving encryption key from key file by Crypto Kernel Module further comprises steps of:

10 receiving password (P) through user input (602);
 calculating message digest (H) by hashing received password (P) (604);
 reading key file to retrieve encrypted key (EK) (606);
 retrieving encrypted key (R) using H as a symmetric decryption key (608);
 and
15 forwarding retrieved encryption key (R) to crypto kernel module (610).

6. A method (700) according to Claim 2, wherein writing data on EBS Volume and encrypting data further comprises steps of:

20 requesting to write data on EBS Volume (702);
 performing write system call to I/O Manager (704);
 forwarding data to crypto kernel module and encrypting data (706);
 forwarding encrypted data to file system (708); and
 storing encrypted data on EBS Volume (710).

- 25 7. A method (800) according to Claim 7, wherein encrypting data forwarded to crypto kernel module further comprises steps of:

 retrieving encryption key (EK) (802); and
 encrypting data to obtain encrypted data (ED) using EK as a symmetric encryption key (804).

- 30 8. A method (900) according to Claim 2, wherein reading encrypted data from EBS Volume and decrypting said encrypted data further comprises steps of:

 requesting to read data from EBS Volume (902);
 performing read system call to I/O Manager (904);

reading data from EBS Volume and decrypting data (906);
forwarding decrypted data to I/O Manager (908); and
presenting decrypted data to requester (910).

- 5 9. A method (1000) according to Claim 9, wherein decrypting data upon reading data from EBS Volume further comprises steps of:

retrieving encryption key (EK) (1002); and
decrypting data to obtain decrypted data (DD) using EK as a symmetric
decryption key (1004).

10

10. A system (100) for managing storage encryption over network based Elastic-Block Store (EBS) Volume (118) with at least one user mode (102) and at least one kernel mode (104); said user mode (102) comprising of at least one application (106) and at least one block device (108); said kernel mode (104) comprising of at least one I/O Manager (110) for receiving mounting instruction and presenting decrypted data to requester; at least one crypto kernel module (112) for retrieving encryption key from key file; at least one file system (116) and at least one EBS Volume (118)

15

characterized in that

20

the at least one crypto kernel module (112) for retrieving encryption key from key file having means for:

hashing password (P) through user input to obtain symmetric
key to decrypt encrypted data from key file;
calculating message digest (H) by hashing received password (P);
25 reading key file to retrieve encrypted key (EK); and
retrieving encryption key (R) using H as a symmetric
decryption key.

25

1/6

100

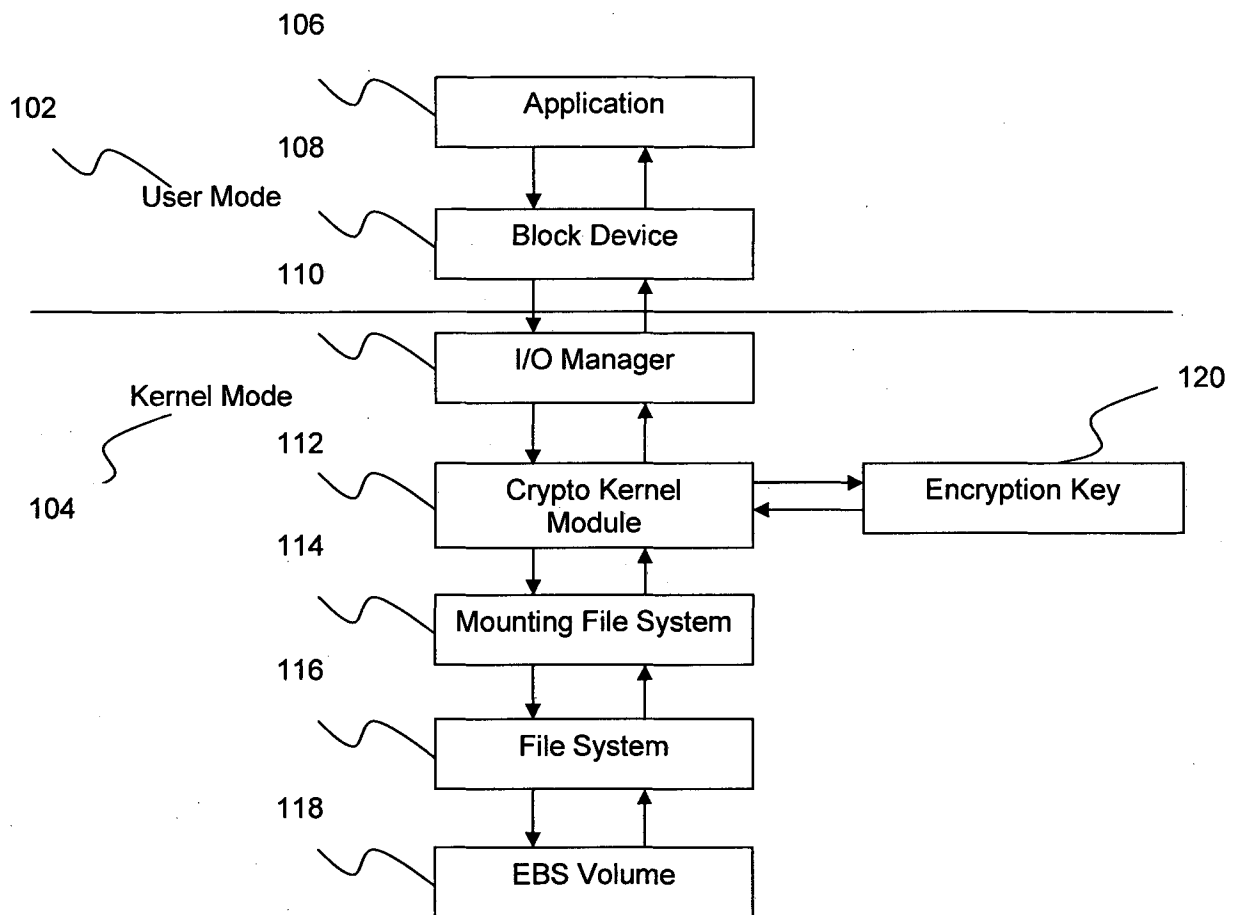


FIG. 1.0

2/6

200

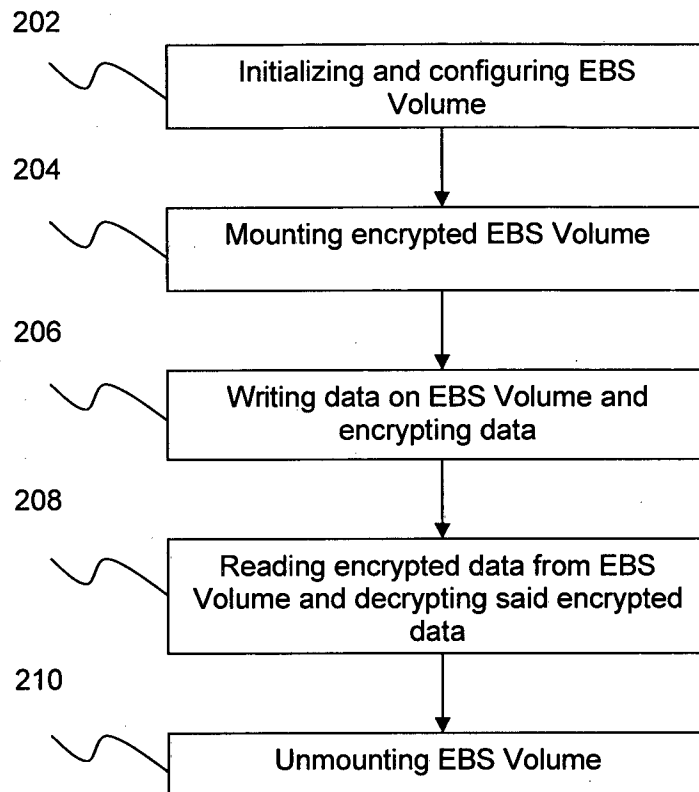


FIG. 2.0

3/6

300

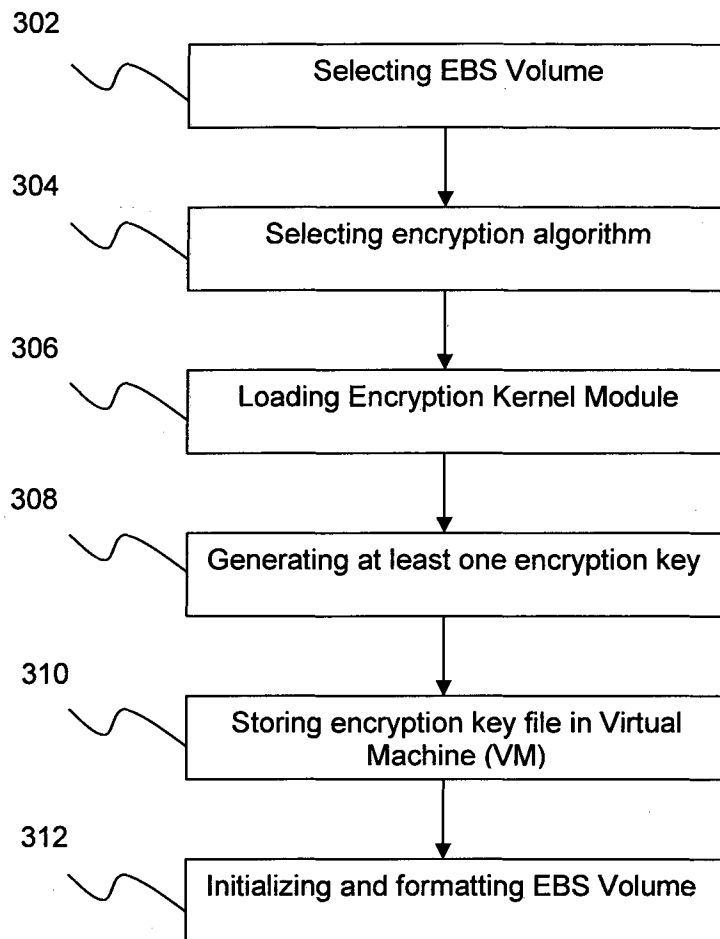


FIG. 3.0

4/6

400

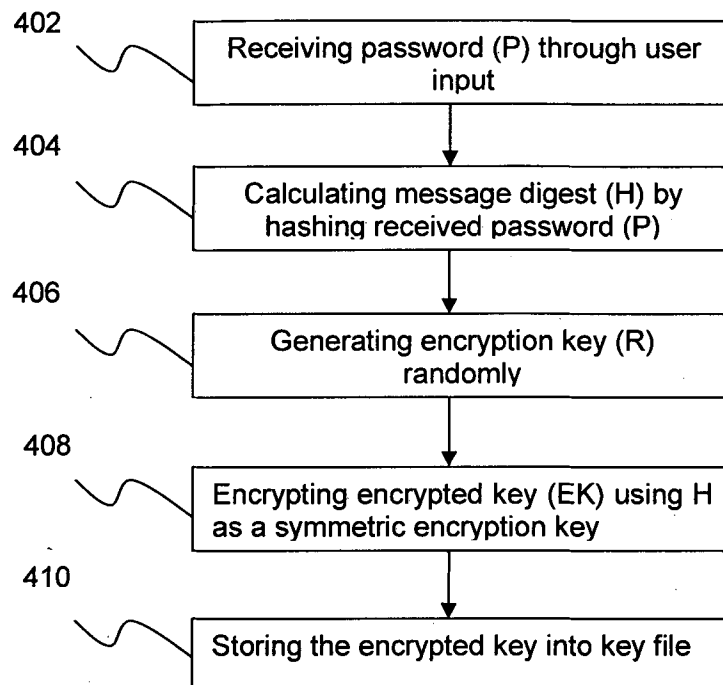


FIG. 4.0

500

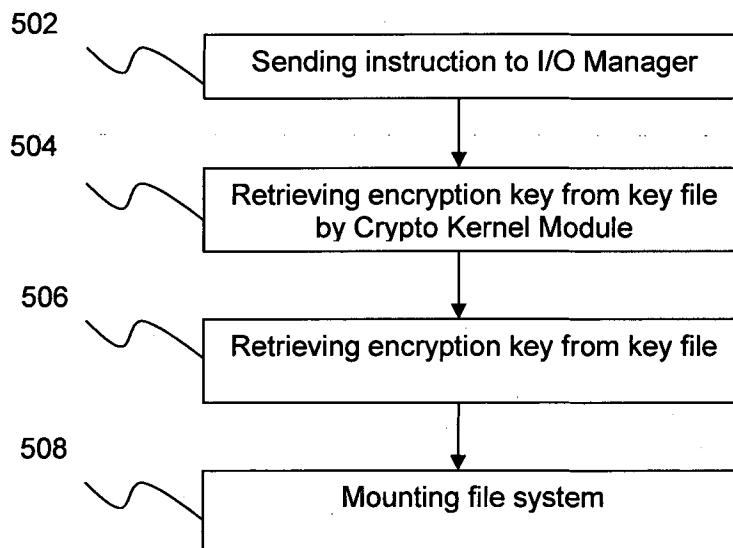


FIG. 5.0

5/6

600

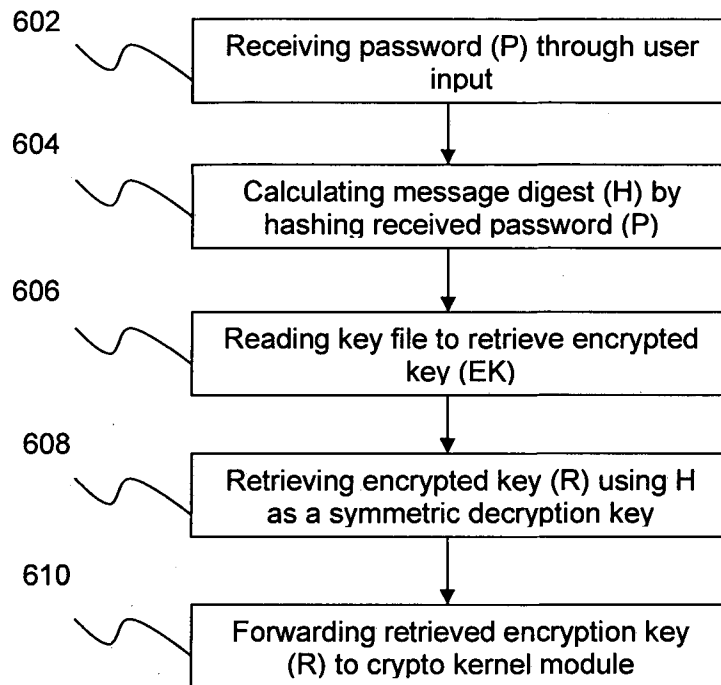


FIG. 6.0

700

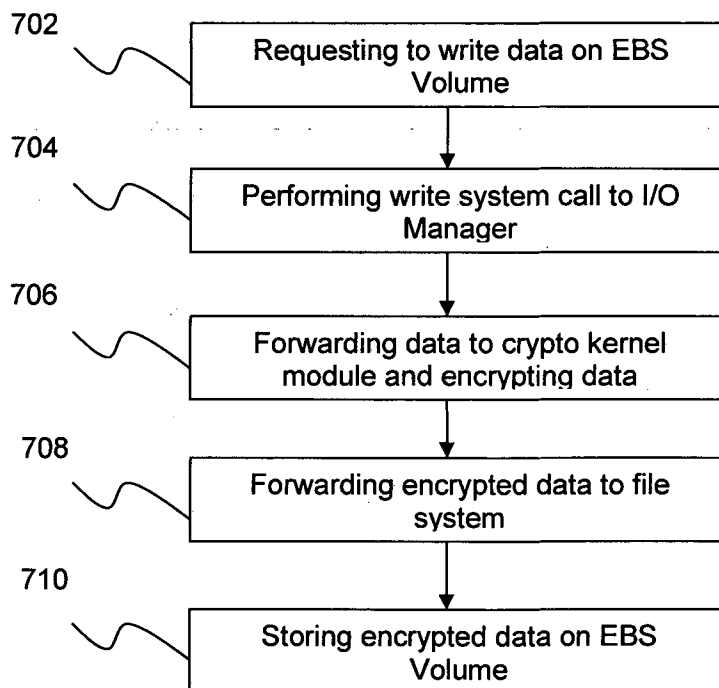


FIG. 7.0

6/6

800

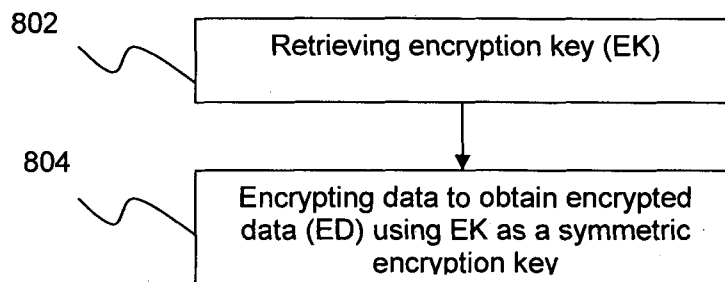


FIG. 8.0

900

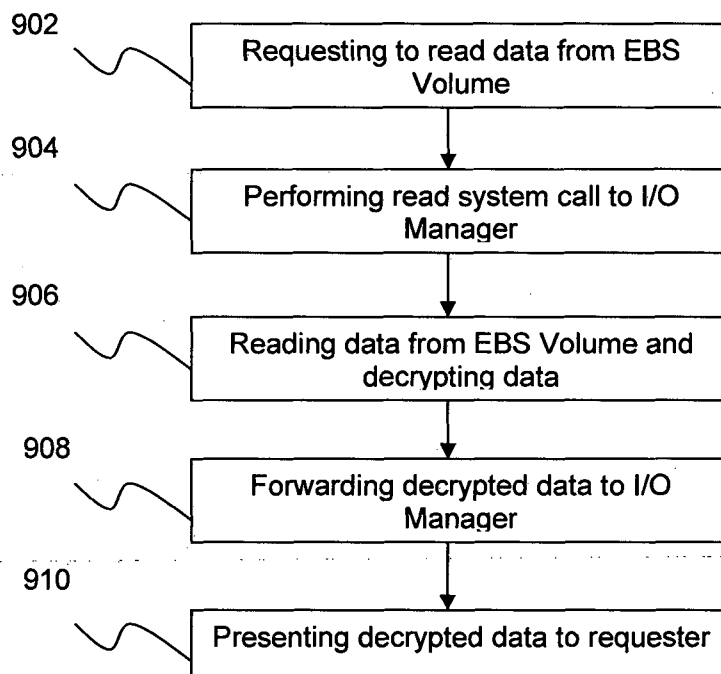


FIG. 9.0

1000

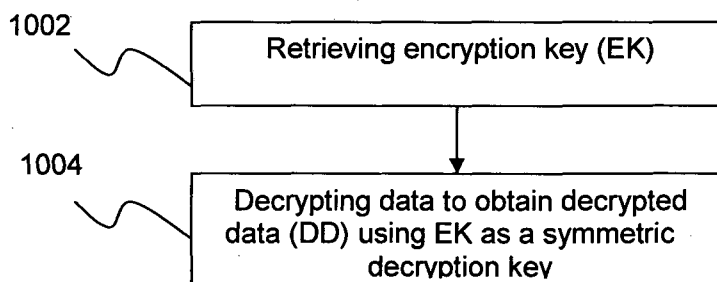


FIG. 10.0

INTERNATIONAL SEARCH REPORT

International application No
PCT/MY2013/000156

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F21/62 G06F3/06
ADD.

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

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

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

EP0-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	Erik Peterson: "Encrypting your data on Amazon EC2", 26 November 2011 (2011-11-26), pages 1-9, XP055066779, Retrieved from the Internet: URL: http://silvexis.com/2011/11/26/encrypting-your-data-on-amazon-ec2/ [retrieved on 2013-06-14] the whole document ----- -/-	1-10



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

11 December 2013

Date of mailing of the international search report

20/12/2013

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Mäenpää, Jari

INTERNATIONAL SEARCH REPORT

International application No
PCT/MY2013/000156

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	Clemens Fruhwirth ET AL: "Secret messages, hard disk encryption with DM-Crypt, LUKS, and cryptsetup", Linux Magazine, 1 December 2005 (2005-12-01), XP055092966, Retrieved from the Internet: URL:http://www.linux-magazine.com/content/download/62715/485838/version/1/file/Hard_Disk_Encryption_DM-Crypt_LUKS_cryptsetup.pdf [retrieved on 2013-12-11] the whole document -----	1-10