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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 June 2012 (14.06.2012)

WIPO | PCT

(10) International Publication Number
WO 2012/076266 A1

(51) **International Patent Classification**:
G06F 21/02 (2006.01) *G06F 9/455* (2006.01)

(21) **International Application Number**:
PCT/EP2011/069323

(22) **International Filing Date**:
3 November 2011 (03.11.2011)

(25) **Filing Language**: English

(26) **Publication Language**: English

(30) **Priority Data**:
10194400.7 9 December 2010 (09.12.2010) EP

(71) **Applicant** *(for all designated States except US)*: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, New York, Armonk, N.Y., New York 10504 (US).

(72) **Inventors; and**

(75) **Inventors/Applicants** *(for US only)*: **FONTIGNIE, Jacques** [BE/CH]; IBM Suisse SA, 8 Chemin de Blandonnet, CH-1214 Vernier (CH). **MARINELLI, Claudio** [IT/IT]; IBM Italia SpA, Via Sciangai 53, I-00144 Roma (IT). **VUILLEUMIER STUECKELBERG, Marc** [CH/CH]; IBM Research GmbH, Saeumerstrasse 4, CH-8803 Rueschlikon (CH). **PICHETTI, Luigi** [IT/IT]; IBM Italia SpA, Via Sciangai 53, I-00144 Roma (IT).

(74) **Agent**: **KUISMA, Dr. Sirpa**; IBM Deutschland Management & Business Support GmbH, IBM-Allee 1, 71139 Ehningen (DE).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, 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, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, 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.

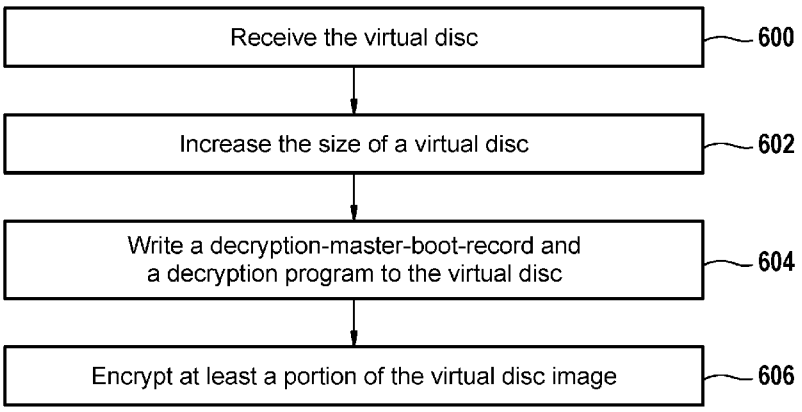
(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, 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, ML, MR, NE, SN, TD, TG).

Published:

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

(54) **Title**: COMPUTER-READABLE STORAGE MEDIUMS FOR ENCRYPTING AND DECRYPTING A VIRTUAL DISC

Fig. 6



(57) **Abstract**: A computer-readable storage medium (916) containing machine executable instructions that when executed by a processor cause the processor to encrypt a virtual disc; wherein the virtual disc comprises a virtual disc image; and wherein execution of the machine executable instructions cause the processor to: receive (600) the virtual disc; increase (602) the size of the virtual disc; write (604) a decryption-master- boot-record and a decryption program to the virtual disc; encrypt (606) at least a portion of the virtual disc image, wherein the decryption program comprises decryption-machine- executable-instructions for decrypting the at least partially encrypted virtual disc image in accordance with a cryptographic key (944).

WO 2012/076266 A1

D E S C R I P T I O N

COMPUTER-READABLE STORAGE MEDIUMS FOR ENCRYPTING AND DECRYPTING A VIRTUAL DISC

Field of the invention

The invention relates to virtualization software for computer systems. In particular the invention relates to the protecting the contents of a virtual disc by encrypting and decrypting the virtual disc.

Background

Setting up or installing large numbers of servers and computer systems with specific software and applications using conventional physical resources has become more and more costly. In recent years this process has be simplified by the use of virtualization technologies to provide virtual disc or operating systems with preconfigured software packages and system configurations.

If a large number of like configured servers is needed, the same virtual disc can be used over and over again. Virtual disc images may be distributed via the internet. However the packets containing the virtual disk image may contain critical data and licenses. In addition the unauthorized use of virtual disc images needs to be prevented. Currently the entire disc image is encrypted before being distributed.

Summary of the invention

The invention provides for a computer-readable storage medium containing instructions for encrypting a virtual disc, a computer-readable storage medium containing instructions for decrypting an encrypted virtual disc, and an encrypted-virtual-disc computer-readable storage medium containing the virtual disc in the independent claims. Embodiments are given in the dependent claims.

Encrypting the entire virtual disk images before distribution has several disadvantages. First a virtual disc image in one format may not be converted into another without being decrypted. Secondly, an administrator is needed to decrypt and install the virtual disc image. This could be a problem, because a cryptographic key or credentials for decrypting the virtual disc image need to be provided to the administrator. The end user or operator of the virtual disc system may or may not wish to share the cryptographic key or credentials with the administrator.

Embodiments of the invention may solve these and other problems by placing an encryption-master-boot-record on the encrypted virtual disc image along with a decryption program for decrypting an at least partially encrypted virtual disc image. The encryption-master-boot-record is a master boot record which is used for booting the virtual machine. The virtual machine its self then decrypts the encrypted virtual disc image using the decryption program and cryptographic credentials. This eliminates the need to provide the operator with the cryptographic credentials.

Embodiments of the invention may have the advantage that the virtual disk may be encrypted or decrypted from the virtual

disc. Embodiments of the invention may have the advantage that the method can be set up to encrypt only used blocks. This results in faster encryption and a reduction in the amount of data to encrypt. Embodiments of the invention may have the advantage that a virtual disk can be encrypted and decrypted on the fly.

Additionally, only portions of the virtual disk may be encrypted. For instance, the blocks of the virtual disc may be selectively encrypted. The portions of the virtual disc which contain data or records specific to a particular format of virtual disc can be left unencrypted. This allows the conversion of the virtual disc without decrypting the virtual disc.

A computer-readable storage medium as used herein encompasses any tangible storage medium which may store instructions which are executable by a processor of a computing device. The computer-readable storage medium may be referred to as a computer-readable non-transitory storage medium. The computer-readable storage medium may also be referred to as a tangible computer-readable medium. In some embodiments, a computer-readable storage medium may also be able to store data which is able to be accessed by the processor of the computing device.

Examples of computer-readable storage media include, but are not limited to: a floppy disc, a magnetic hard disc drive, a solid state hard disc, flash memory, a USB thumb drive, Random Access Memory (RAM) memory, Read Only Memory (ROM) memory, an optical disc, a magneto-optical disc, and the register file of the processor. Examples of optical discs include Compact Discs (CD) and Digital Versatile Discs (DVD), for example CD-ROM, CD-RW, CD-R, DVD-ROM, DVD-RW, or DVD-R discs. The term

computer-readable storage medium also refers to various types of recording media capable of being accessed by the computer device via a network or communication link. For example a data may be retrieved over a modem, over the internet, or over a local area network.

Computer memory is an example of a computer-readable storage medium. Computer memory is any memory which is directly accessible to a processor. Examples of computer memory include, but are not limited to: RAM memory, registers, and register files.

Computer storage is an example of a computer-readable storage medium. Computer storage is any non-volatile computer-readable storage medium. Examples of computer storage include, but are not limited to: a hard disc drive, a USB thumb drive, a floppy drive, a smart card, a DVD, a CD-ROM, and a solid state hard drive. In some embodiments computer storage may also be computer memory or vice versa.

A computing device or computer system as used herein refers to any device comprising a processor. A processor as used herein encompasses an electronic component which is able to execute a program or machine executable instruction. References to the computing device comprising "a processor" should be interpreted as possibly containing more than one processor. The term computing device should also be interpreted to possibly refer to a collection or network of computing devices each comprising a processor. Many programs have their instructions performed by multiple processors that may be within the same computing device or which may even distributed across multiple computing device.

A user interface as used herein is an interface which allows a user or operator to interact with a computer or computer system. A user interface may provide information or data to the operator and/or receive information or data from the operator. The display of data or information on a display or a graphical user interface is an example of providing information to an operator. The receiving of data through a keyboard, mouse, trackball, touchpad, pointing stick, graphics tablet, joystick, gamepad, webcam, headset, gear sticks, steering wheel, pedals, wired glove, dance pad, remote control, and accelerometer are all examples of receiving information or data from an operator.

Virtualization software, a virtualization program, and a virtualization module as used herein all refer to software or computer executable instruction which allow a computer system to run a virtual computer system. A virtual machine or virtual computer system as used herein encompasses a computer system which is implemented virtually or simulated by software running on a computer system.

A cryptographic key or cryptographic credential as used herein encompasses a key, credential, or password which may be used by a decryption algorithm to decrypt a data file.

A virtual disc as used herein encompasses data which may be used by a virtualization program as a virtual disc image. A virtual disc may contain a file system which may be accessed by the virtualization system or virtual system. A virtual disc may also contain a bootable operating system.

In one aspect the invention provides for a computer-readable storage medium containing machine executable instructions that when executed by a processor cause a processor to encrypt a

virtual disc. The virtual disc comprises a virtual disc image. The virtual disc image is an image of an existing disc file system or a disc file system which is constructed for the purpose of creating the virtual disc image. Execution of the machine executable instructions causes the processor to receive the virtual disc. Execution of the machine executable instructions further causes the processor to increase the size of the virtual disc. The virtual disc may essentially be a file stored on the computer-readable storage medium or a different computer-readable storage medium. The size of the virtual disc may be increased by adding blocks at the beginning or the end of the virtual disc. Execution of the machine executable instructions further cause the processor to write a decryption-master-boot-record and a decryption program to the virtual disc. The decryption-master-boot-record is a master boot record that a virtual computer or computer system boots into when using the virtual disc image. The decryption-master-boot-record allows the virtual computer system to run the decryption program for decrypting the virtual disc. Execution of the machine executable instructions further causes the processor to encrypt at least a portion of the virtual disc image. The decryption program comprises decryption-machine-executable-instructions for decrypting the at least partially encrypted virtual disc image in accordance with a cryptographic key. The virtual disc image is encrypted such that the virtual disc image may be decrypted using the decryption program in accordance with the cryptographic key.

In some embodiments the entire virtual disc image is encrypted. In other embodiments only certain portions of the virtual disc image are encrypted. For instance if portions of the virtual disc are unused, these portions of the disc image need not be encrypted. It may also be able to be determined if certain portions of the virtual disc contain sensitive

information which will be desirable to protect by encryption for instance application programs or sensitive data. The decryption of the virtual disc image could be speeded by selectively using those portions of the virtual disc which need to be protected and not encrypting those portions which do not need to be protected.

Embodiments of the invention have several advantages. For instance, adding the decryption-master-boot-record to the virtual disc and the decryption program enables an end user to perform the decryption of the virtual disc image. This eliminates the need for an administrator to perform this task.

In another embodiment the virtual disc image is divided into first and second parts. The virtual disc is divided into first, second, third, fourth, and fifth portions. The virtual disc image originally spans the first, second, and third portions of the virtual disk. The second part of the virtual disc image is stored in a third portion of the virtual disc. Execution of the instructions further cause the processor to copy the first part of the virtual disc image to a fourth portion of the virtual disc. The first part of the virtual disc image is copied or moved from the first and second portions of the virtual disc to the fourth portion of the virtual disc. The decryption-master-boot-record is written to the first portion of the virtual disc. When the virtual disc is loaded into a virtual system and the virtual system boots into the virtual disc the decryption-master-boot-record will cause the virtual system to run the decryption program.

The decryption program is written to the second portion of the virtual disc. As mentioned above, The first part of the virtual disc image is copied from the first and second portions of the virtual disc. The first part of the virtual

disc image is copied from the first and second portions of the virtual disc before the decryption-master-boot-record and the decryption program are written to the first and second portions of the virtual disc respectively. Execution of the instructions further causes the processor to at least partially encrypt the first and second parts of the virtual disc. The size of the virtual disc is increased to create the fourth portion of the virtual disc and a fifth portion of the virtual disc. The size of the fifth portion is larger than or equal to the second portion. The combined size of the first and second portions is less than or equal to the size of the fourth portion. This embodiment of the invention may be advantageous because the decryption-master-boot-record is in the first portion and will cause the virtual system to boot into the decryption program.

In another embodiment the first part of the virtual disc image is encrypted together. In this embodiment the entire first part of the virtual disc image is encrypted as a single encrypted data file.

In another embodiment the second part of the virtual disc image is encrypted together. In this embodiment the second part of the virtual disc image is encrypted as a single data file.

In another embodiment the virtual disc image is divided into blocks. As used herein a block is a portion or sub-division of data of a disc or a virtual disc image. The data in a block is addressable by the disc or the virtual disc. The blocks are selectively encrypted in accordance with a predetermined block encryption list. The block encryption list is a list of blocks which are to be encrypted during the encryption of the virtual disc image. For instance an operator could determine which

blocks of the virtual disc image contain data which is desired to be protected by encryption. For instance these blocks may contain sensitive data or information. Likewise these blocks may contain applications for which a license is to be purchased. If the program is transmitted across the internet it would be desirable to protect the executable version of the code or data.

In another embodiment the virtual disc image is divided into blocks. Execution of the instructions causes the processor to examine each of the blocks and create a list of unused blocks. Particular blocks are encrypted only if they are not found in the list unused blocks. This embodiment is particularly advantageous because the computer-readable storage medium avoids encrypting blocks which are not used. Since the blocks are not used there is no need to protect them. In some virtual file systems unused data may be part of the file system but not used. By not encrypting these portions of the file system the virtual disc may be smaller.

The aforementioned embodiments of the computer-readable storage medium also provide for other aspects of the invention. For instance a computer system is provided for by the invention which contains or comprises the machine readable instructions contained on a computer-readable storage medium according to an embodiment of the invention. Likewise execution of the machine executable instructions causes the processor to perform various steps or actions which also provide for a method and computer-implemented methods. The executable instructions on the computer-readable storage medium also provides for a computer program product and/or a computer system.

In another aspect the invention provides for a computer-readable storage medium containing machine executable instructions that when executed by a processor cause the processor to decrypt an encrypted virtual disc. The virtual disc comprises a decryption-master-boot-record, a decryption program, and an at least partially encrypted virtual disc image. The decryption program comprises decryption-machine-executable-instructions for decrypting the at least partially encrypted virtual disc image in accordance with a cryptographic key. Execution of the machine executable instructions causes the processor to receive the encrypted virtual disc.

Execution of the machine executable instructions further causes the processor to boot a virtual machine using the decryption-master-boot-record. Execution of the machine executable instructions further causes the processor to receive the cryptographic key. The order of receiving the cryptographic key is not critical. For instance the processor could receive the cryptographic key at any point before the virtual disc image is decrypted. Execution of the machine executable instructions further cause the processor to decrypt the at least partially encrypted virtual disc image in accordance with the cryptographic key and the decryption program. The machine executable instructions cause the processor to boot the virtual machine and the virtual machine boots into the operating system on the encrypted virtual disc via the decryption-master-boot-record. This then causes the virtual machine to run the decryption program. The decryption program then decrypts the at least partially encrypted virtual disc image. Both the cryptographic key and the decryption program are needed for decrypting the at least partially encrypted virtual disc image.

In another embodiment the decryption of the at least partially encrypted virtual disc image is performed during deployment of the virtual machine. In the current state of the art an administrator will receive an encrypted virtual disc and the administrator is responsible for decrypting it. This is however undesirable in many circumstances because the end user or operator of the virtual machine relies on an administrator to perform the decryption. Embodiments of the invention may have the advantage that the end user or operator can perform the decryption his or herself.

In another embodiment the virtual disc comprises a first portion containing the decryption-master-boot-record. The virtual disc further comprises a second portion containing the decryption program. The virtual disc further comprises a third portion containing a second part of the virtual disc image. The virtual disc further comprises a fourth portion containing a first part of the virtual disc. The first part of the virtual disc may contain in some embodiments a master boot record for booting into an operating system contained in the virtual disc image. This master boot record in the fourth portion may be used to boot the virtual machine once the decryption of the at least partially encrypted virtual disc is completed.

The virtual disc comprises a fifth portion containing storage space. The size of the fifth portion is larger than the second portion. The combined size of the first and second portions is less than or equal to the size of the fourth portion. The combined size of the first and second portions is less than or equal to the size of the fourth portion. The at least partially encrypted disc image is decrypted by decrypting the second part of the virtual disc image. The at least partially encrypted virtual disc image is further decrypted by copying

the decryption program to the fifth portion of the virtual disc. The at least partially encrypted virtual disc image is decrypted by decrypting a portion of the first part of the virtual disc image.

The at least partially encrypted virtual disc image is further decrypted by copying the decrypted portion of the first part of the virtual disc image to the second portion of the virtual disc. The virtual disc image is further decrypted by decrypting the remainder of the first part of the virtual disc image. The virtual disc image is further decrypted by copying the decrypted remainder of the first part of the virtual disc image to the first portion of the virtual disc. Performing the decryption in this manner may have the advantage that the decryption can be interrupted at any point in time. For instance the fifth portion may contain a data file which maintains a status of the decryption process.

In another embodiment execution of the instructions further causes the processor to erase data in the fourth and fifth portions of the virtual disc after copying the decrypted remainder of the first part of the virtual disc image to the first portion of the virtual disc.

In another embodiment execution of the instructions further causes the virtual machine to reboot after decrypting the at least partially encrypted virtual disc image.

The aforementioned embodiments of the computer-readable storage medium also provide for other aspects of the invention. For instance a computer system is provided for by the invention which contains or comprises the machine readable instructions contained on a computer-readable storage medium according to an embodiment of the invention. Likewise

execution of the machine executable instructions causes the processor to perform various steps or actions which also provide for a method and computer-implemented methods. The executable instructions on the computer-readable storage medium also provides for a computer program product and/or a computer system.

In another aspect the invention provides for an encrypted-virtual-disc computer-readable storage medium containing a virtual disc. The virtual disc comprises a decryption-master-boot-record, a decryption program, and an at least partially encrypted virtual disc image. The decryption program comprises machine executable instructions for decrypting the at least partially encrypted virtual disc in accordance with a cryptographic key. In other words the combination of the decryption program and the cryptographic key are used for decrypting the at least partially encrypted virtual disc image.

The decryption program comprises machine executable instructions that when executed by a processor cause the processor to receive a cryptographic key. The cryptographic key may in some embodiments be prompted to be entered by the decryption program or the cryptographic key may be passed to the decryption program by another program. For instance virtualization software for running a virtual computer system may pass a cryptographic key on to the decryption program. Further execution of the machine executable instructions of the decryption program cause the processor to decrypt the at least partially encrypted virtual disc image in accordance with the cryptographic key and the decryption program.

In another embodiment the virtual disc comprises a first portion containing the decryption-master-boot-record. The

virtual disc further comprises a second portion containing the decryption program. The virtual disc further comprises a third portion containing a second part of the virtual disc image. The virtual disc further comprises a fourth portion containing a first part of the virtual disc. The virtual disc further comprises a fifth portion containing storage space. The size of the fifth portion is larger than or equal to the second portion. The combined size of the first and second portions is less than or equal to the size of the fourth portion.

In another embodiment the at least partially encrypted virtual disc image is decrypted by decrypting the second part of the virtual disc image. The virtual disc is further decrypted by copying the decryption program to the fifth portion of the virtual disc. The virtual disc image is further decrypted by decrypting a portion of the first part of the virtual disc image. The virtual disc is further decrypted by copying the decrypted portion of the first part of the virtual disc image to the second portion of the virtual disc. The virtual disc is further decrypted by copying the decrypted portion of a first part of the virtual disc image to the second portion of the virtual disc. The virtual disc image is further decrypted by decrypting the remaining of the first part of the virtual disc image. The virtual disc image is further decrypted by copying the decrypted remainder of the first part of the virtual disc image to the first portion of the virtual disc.

In another embodiment the virtual disc contains a decryption-status-data-file for storing the progress of the decryption of the at least partially encrypted virtual disc image. Execution of the machine executable instructions of the decryption program further cause the processor to update the decryption-status-data-file during decryption of the at least partially encrypted virtual disc image. Execution of the machine

executable instructions of the decryption program further cause the processor to check the decryption-status-data-file when starting the decryption of the at least partially encrypted virtual disc image. By checking the status of the decryption-status-data-file the decryption can be started at an intermediate point if the decryption was originally interrupted.

In another embodiment execution of the instructions further cause the processor to erase data in the fourth and fifth portion of the virtual disc after copying the decrypted remainder of the first part of the virtual disc image to the first portion of the virtual disc.

In another embodiment execution of the instructions further cause a virtual machine executing the decryption program to reboot after decrypting the at least partially encrypted virtual disc image.

The aforementioned embodiments of the computer-readable storage medium also provide for other aspects of the invention. For instance a computer system is provided for by the invention which contains or comprises the machine readable instructions contained on a computer-readable storage medium according to an embodiment of the invention. Likewise execution of the machine executable instructions causes the processor to perform various steps or actions which also provide for a method and computer-implemented methods. The executable instructions on the computer-readable storage medium also provides for a computer program product and/or a computer system.

Brief description of the drawings

In the following, preferred embodiments of the invention will be described in greater detail by way of example only making reference to the drawings in which:

Fig. 1 illustrates the decryption of a virtual disc 100 according to an embodiment of the invention,

Fig. 2 illustrates an example of decryption during importation of the virtual system,

Fig. 3 illustrates the decryption of the virtual disc image during deployment,

Figs. 4a-4e illustrates a method of block-based encryption of a virtual disc image according to an embodiment of the invention,

Figs. 5a-5d illustrates the decryption of the virtual disc image encrypted in Figs. 4a-4e,

Fig. 6 shows a flow diagram which illustrates a method of encrypting a virtual disc image according to a further embodiment of the invention,

Fig. 7 shows a flow diagram which illustrates a method of decrypting a virtual disc image according to a further embodiment of the invention,

Fig. 8 shows a flow diagram which illustrates a method of decrypting a virtual disc image according to a further embodiment of the invention, and

Fig. 9 illustrates a first computer system for encrypting a virtual disc and a second computer system for decrypting a virtual disc.

Detailed description

In the following, like numbered elements in these figures are either similar elements or perform an equivalent function. Elements which have been discussed previously will not necessarily be discussed in later figures if the function is equivalent.

Fig. 1 illustrates the decryption of a virtual disc 100 according to an embodiment of the invention. The virtual disc 100 comprises a virtual disc image 102 which is encrypted and a decryption program 104 for decrypting the virtual disc image 102. In the Fig. is also shown a computer system 106 which contains virtualization software for running virtual computer systems or machines. Shown in Fig. 1 is an operator 108 and a user 110. The Fig. shown in Fig. 1 illustrates the actions taken by the operator 108 and the user 110 when using a virtual disc 100 according to an embodiment of the invention. The steps shown in Fig. 1 are for the first booting of the virtual disc and its decryption. Step 1 is labeled 112. In this step the operator 108 stores the virtual disc 100 on the computer system 106. In step 2, 114 the user 110 boots the virtual machine using the virtualization software on a computer system 106. In step 3, 116 the decryption program 104 starts and requests credentials or a password from the user 110. In step 4, 118 the user 110 provides the credentials or password to the decryption program 104. In step 5, 120 the

decryption program 104 decrypts the virtual disc image 102 using the credentials or password. In step 6, 122 the virtual machine reboots and the virtual machine boots from the decrypted virtual disc image 102.

During use of a virtual disc 100 according to an embodiment of the invention two different use scenarios are possible. There may be decryption during the import of the virtual disc image 102 or there may be decryption during deployment of the virtual disc image 102. If the virtual disc image 102 is decrypted during the importation of the virtual system the operator knows the credentials and passes this to the program or programs for managing the virtual systems. The programs for managing the virtual system then import the image and decrypt it on the fly. For the second possibility for decryption during deployment, the operator does not know the credentials and asks to import the images without decrypting. The virtual disc image is stored in a database of virtual systems at deployment time the user is prompted for the credentials.

Fig. 2 shows an example of decryption during importation of the virtual system. Shown in Fig. 2 is a computer system 200 which functions as a virtual system image server 202. The virtual system image server 202 serves images of virtual systems when requested by an operator 208. Also stored or able to be accessed by the computer system 200 is a virtual disc repository 204 which is a repository of virtual discs which are accessible via the virtual system image server 202. There is also a decryption module 206 which is equivalent to the decryption program 104 shown in Fig. 1. In step 1 the operator 208 downloads a virtual disc. In step 2, 212 the operator requests the importation of the virtual disc into the system managed by the virtual system image server 202. In step 3, 214 the virtual system image server 202 requests credentials or a

password from the operator 208. In step 4, 216 the operator 208 provides the credentials or passwords to the virtual system image server 202. In step 5 the virtual system image server 202 imports the virtual disc image from the virtual disc and decrypts the virtual disc image on the fly using the credentials or passwords provided by the operator 208.

Fig. 3 illustrates the decryption of the virtual disc image during deployment. Shown in this figure is a computer system 200 with a virtual system image server 202 which manages virtual disc images stored in a virtual disc repository 204. In the example shown in Fig. 3 there is a second computer system 300 which is used for decryption during the deployment of the virtual disc image. The second computer system 300 is for running a target virtual system 302. Within the target system 302 is an operating system deployment tool 301. The operating system deployment tool 301 is provided to deploy a virtual system in a virtual disc image.

Also within the target virtual system is a decryption module 303. The decryption module is a software module or decryption program for decrypting a virtual disc image using a password or a cryptographic credential. An operator 304 and a user 306 are shown. In a first step 1 designated by reference 308, the operator 304 requests or triggers the deployment of a virtual system on the second computer system 300. In a second step 2 (designated by reference 310) the target or virtual system boots on the operating system deployment tool 301. In a third step 3 (designated by reference 312), the operating system deployment tool 301 requests cryptographic password or credentials from the user 306. In a fourth step 4 (designated by reference 314), the user 306 provides the cryptographic password or credentials to the operating system deployment tool 301. In a fifth step 5 (also referred to as 316), the

operating tool downloads and decrypts the virtual disc image using the cryptographic password or credentials and the decryption module 303. In step 6 (also referred to as 318) after the virtual disc image has been decrypted the deployment of the virtual system continues on the decrypted virtual disc image.

Figures 4a to 4e illustrate a method of block-based encryption of a virtual disc image according to an embodiment of the invention. In Fig. 4a, a virtual disc 400 and a virtual disc image 402 are shown. The blocks which make up the virtual disc image are labeled 1-n. To encrypt the virtual disc image next the user starts an encryption tool or program. In a first step the decryption tool increases the size of the virtual disc. This is illustrated in Fig. 4b. At the end of the virtual disc 400 a region of empty operating system blocks 404 is created. In a next step the virtual disc image is divided into a first part 406 and a second part 408. The first part of the virtual disc image 406 is copied to the empty operating system blocks 404 at the end of the virtual disc 400.

In Fig. 4d, it is shown that the encryption tool copies a decryption master boot record 410 and a decryption program 412 to a first and second part of the virtual disc 400. Fig. 4e illustrates the final step. The encryption tool or software encrypts the first part of the virtual disc image 406′ and encrypts the second part of the virtual disc image 408′. All of the blocks of the encrypted virtual disc image 406′, 408′ may be encrypted or the blocks may be selectively encrypted. In Fig. 4e the virtual disc 400 is also shown as being divided into five portions. The first portion of the virtual disc 414 contains the decryption-master-boot-record 410. The second portion of the virtual disc 416 contains the decryption program 412. The third portion of the virtual disc 418

contains the encrypted 408′ second part of the virtual disc image. The fourth portion of the virtual disc 420 contains the encrypted 406′ first part of the virtual disc image. The fifth portion of the virtual disc 422 is at the end of the virtual disc 400. In various embodiments the fifth portion 422 may contain data recording the decryption state, journaling data, temporary data used in the decryption, and combinations thereof.

Figs 5a-5d illustrates the decryption of the virtual disc image 406′, 408′ of a virtual disc 400 when booted from a virtual machine. Fig. 5a is identical with Fig. 4e. In a first step the virtual machine boots on the virtual disc 400 and boots into the virtual disc master boot record 410. Next the master boot record 410 loads the decryption program 412. The decryption program 412 then requests cryptographic credentials or a password for use for decrypting the encrypted virtual disc image 406′, 408′. In Fig. 5b the decryption process is illustrated. Two different views of the virtual disc 400 are shown. Blocks labeled 500 are decrypted blocks of the second part of the virtual disc image. Blocks labeled 502 are encrypted blocks of the second part of the virtual disc image. In the top view shown in Fig. b only the block labeled 4 is a decrypted block 500. The remainder of the second part of the virtual disc image is encrypted. The bottom part of Fig. 5b shows that all blocks of the second part of the virtual disc image 408 are decrypted blocks 500.

Fig. 5c shows further progress in decrypting the virtual disc 400. After all blocks of the second part of the virtual disc image 408 have been decrypted the decryption program 412 is copied to the fifth portion 422 of the virtual disc. Next a portion of the first part of the virtual disc image 406′ is decrypted and copied to the second portion 416 of the virtual

disc. The remainder 506 of the first part of the virtual disc image 406′ is decrypted and copied to the first portion 414 of the virtual disc. The remainder of the first part of the virtual disc image 506 in this embodiment has overwritten the master boot record 410. The encrypted first part of the virtual disc image 406′ and the decryption program 412 may be overwritten leaving empty operating system blocks 404. The Fig. shown in 5d is equivalent with that shown in Fig. 4b. This shows how the method illustrated in Fig. 5 has been used to decrypt the at least partially encrypted virtual disc image 402 of the virtual disc 400.

Fig. 6 shows a flow diagram which illustrates an embodiment of encrypting a virtual disc image according to the invention. In step 600 a virtual disc is received. The virtual disc comprises a virtual disc image. In step 602 the size of the virtual disc is increased. In step 604 a decryption-master-boot-record and a decryption program are written to the virtual disc. In step 606 at least a portion of the virtual disc image is encrypted.

Fig. 7 shows a flow diagram which illustrates a method of decrypting a virtual disc according to an embodiment of the invention. In step 700 an encrypted virtual disc is received. In step 702 a virtual machine is booted using a decryption-master-boot-record contained on the virtual disc. In step 704 a cryptographic key is received. The virtual disc comprises an at least partially encrypted virtual disc. In step 706 the at least partially encrypted virtual disc is decrypted using a decryption program which is on the virtual disc. The decryption program uses the cryptographic key for decrypting with the decryption program for performing the decryption of the at least partially encrypted virtual disc.

Fig. 8 shows a flow diagram which illustrates a method of decrypting an encrypted virtual disc according to a further embodiment of the invention. In step 800 an encrypted virtual disc is received. In step 802 a virtual machine is booted using the decryption-master-boot-record. In step 804 a cryptographic key is received. In step 806 a second part of the virtual disc image is decrypted using the cryptographic key and a decryption program which is located on the virtual disc. In step 808 the decryption program is copied to a fifth portion of the virtual disc. In step 810 a portion of a first part of the virtual disc image is decrypted. In step 812 the decrypted portion of the first part of the virtual disc image is copied to the second portion of the virtual disc. In step 814 the remainder of the first part of the virtual disc image is decrypted. In step 816 the decrypted remainder of the first part of the virtual disc image is copied to the first portion of the virtual disc. In step 818 the virtual machine is rebooted. The method illustrated in Fig. 8 is analogous to the method illustrated by Fig. 5.

Fig. 9 shows two computer systems, a first computer system 900 for encrypting a virtual disc and a second computer system 902 for decrypting a virtual disc. There is a network communication 904 between the first computer system 900 and the second computer system 902. The first computer system has a network interface 906 for connecting to the computer network 904 and the second computer system 902 has a network interface 908 for connecting to the computer interface 904. The network connection 904 can be any standard computer interface such as an Ethernet connection or an internet connection. The first computer system 900 has a processor 910 that is connected to a user interface 912 the network interface 906. The processor 910 is also connected to computer storage 914 and computer memory 916.

Within the computer storage 914 is an unencrypted virtual disc 918. The unencrypted virtual disc contains a unencrypted virtual disc image. Also within the computer storage 914 is a decryption-master-boot-record 920. Also within the computer storage 914 is a decryption program 922. Also within the computer storage 914 is an encrypted virtual disc 924. The encrypted virtual disc 924 contains a decryption-master-boot-record 920, a decryption program 922, and an at least partially encrypted virtual disc image 923. The encrypted virtual disc 924 may also contain an at least partially encrypted virtual disc image. The computer memory 916 contains an encryption tool 926. An encryption tool 926 is a software module or program containing machine executable instructions that cause the processor 910 to create the encrypted virtual disc 924 using the unencrypted virtual disc 918, the decryption-master-boot-record 920, and the decryption program 922. The encryption tool 926 may be used to implement the methods illustrated in Figs. 4 and 6. In some embodiments the computer memory 916 also contains a cryptographic module and a cryptographic credential generation module 930. The cryptographic module 928 is used for encrypting the unencrypted virtual disc 918. The cryptographic credential generation module 930 is an optional module and may be used for generating cryptographic credentials. For instance the cryptographic credential generation module may be used to generate cryptographic key pair for an asymmetric encryption algorithm.

The second computer system 902 also contains a processor 932. The processor 932 is connected to the network interface 908 and the user interface 934. The processor 932 is also connected to computer storage 936 and computer memory 938. The computer storage 936 contains the encrypted virtual disc 924

from the first computer system 900. In this embodiment the network connection 904 was used to transfer the encrypted virtual disc 924. Also within the computer storage 936 is an encryption cryptographic key 944. Computer memory 938 contains a virtualization module 942. The virtualization module 942 allows the processor 932 to run and operate a virtual computer system. As can be seen all that is needed to decrypt the encrypted virtual disc 924 is the virtualization module 942 and the cryptographic key 944. This Fig. also illustrates how an end user may be able to decrypt the encrypted virtual disc 924 without the aid of an operator.

List of Reference Numerals

- - - - - - - - - - - - - - - - - - -

100	virtual disc
102	virtual disc image
104	decryption program
106	computer system
108	operator
110	user
112	step 1
114	step 2
116	step 3
118	step 4
120	step 5
122	step 6
200	computer system
202	virtual system image server
204	virtual disc respository
206	decryption module
208	operator
210	step 1
212	step 2
214	step 3
216	step 4
218	step 5
300	second computer system
301	operating system deployment tool
302	target virtual system
303	decryption module
304	operator
306	user
308	step 1
310	step 2
312	step 3

314 step 4
316 step 5
318 step 6
400 virtual disc
402 virtual disc image
404 empty operating system blocks
406 first part of virtual disc image
406′ encrypted first part of virtual disc image
408 second part of virtual disc image
408′ encrypted second part of virtual disc image
410 decryption master boot record
412 decryption program
414 first portion of virtual disc
416 second portion of virtual disc
418 third portion of virtual disc
420 fourth portion of virtual disc
422 fifth portion of virtual disc
500 decrypted blocks of second part of virtual disc image
502 encrypted blocks of second part of virtual disc image
504 portion of the first part of the virtual disc image
506 remainder of the first part of the virtual disc image
900 first computer system
902 second computer system
904 network connection
906 network interface
908 network interface
910 processor
912 user interface
914 computer storage
916 computer memory
918 unencrypted virtual disc
920 decryption-master-boot-record
922 decryption program
923 encrypted virtual disc image

924 encrypted virtual disc
926 encryption tool
928 cryptographic module
930 cryptographic credential generation module
931 cryptographic key pair
932 processor
934 user interface
936 computer storage
938 computer memory
940 decrypted virtual disc
942 virtualization module
944 decryption cryptographic key

C L A I M S

1. A computer-readable storage medium (916) containing machine executable instructions that when executed by a processor cause the processor to encrypt a virtual disc; wherein the virtual disc comprises a virtual disc image; and wherein execution of the machine executable instructions cause the processor to:
   - receive (600) the virtual disc;
   - increase (602) the size of the virtual disc;
   - write (604) a decryption-master-boot-record and a decryption program to the virtual disc;
   - encrypt (606) at least a portion of the virtual disc image, wherein the decryption program comprises decryption-machine-executable-instructions for decrypting the at least partially encrypted virtual disc image in accordance with a cryptographic key (944).

2. The computer-readable storage medium of claim 1, wherein the virtual disc image is divided into first (406) and second parts (408), wherein the second part of the virtual disc image is stored in a third portion (418) of the virtual disc, wherein execution of the instructions further causes the processor to copy the first part of the virtual disc image to a fourth portion (420) of the virtual disc; wherein the decryption-master-boot-record is written to a first portion (414) of the virtual disc, wherein the decryption program is written to a second portion (416) of the virtual disc, wherein the first part of the virtual disc image is copied from the first and second portions of the virtual disc; wherein execution of the instructions further causes the processor to at least partially encrypt the first (406') and second (408') parts of the virtual

disc, wherein the size of the virtual disc is increased to create the fourth portion of the virtual disc and a fifth portion (422) of the virtual disc, wherein the size of the fifth portion is larger than the second portion, and wherein the combined size of the first and second portions is less than or equal to the size of the fourth portion.

3. The computer-readable storage medium of claim 1 or 2, wherein the first part of the virtual disc image is encrypted together and/or wherein the second part of the virtual disc image is encrypted together.

4. The computer-readable storage medium of claim 1 or 2, wherein the virtual disc image is divided into blocks, wherein the blocks are selectively encrypted in accordance with a predetermined block encryption list.

5. The computer-readable storage medium of claim 1, 2, or 4, preceding claims, wherein the virtual disc image is divided into blocks, wherein execution of the instructions cause the processor to examine each of the blocks and create a list of unused blocks, and wherein particular blocks are encrypted only if they are not found in the list of unused blocks.

6. A computer-readable storage medium (938) containing machine executable instructions that when executed by a processor cause the processor to decrypt an encrypted virtual disc; wherein the virtual disc comprises: a decryption-master-boot-record, a decryption program, and an at least a partially encrypted virtual disc image; wherein the decryption program comprises decryption-machine-executable-instructions for decrypting the at least partially encrypted virtual disc image in accordance with a

cryptographic key (944), wherein the execution of the machine executable instructions cause the processor to:

- receive (700, 800) the encrypted virtual disc;
- boot (702, 802) a virtual machine using the decryption-master-boot-record;
- receive (118, 216, 314, 704, 804) the cryptographic key;
- decrypt (120, 218, 316, 706, 806, 808, 810, 812, 814, 816, 818) the at least partially encrypted virtual disc image in accordance with the cryptographic key and the decryption program.

7. The computer-readable storage medium of claim 6, wherein the decryption of the at least partially encrypted virtual disc image is performed during deployment of the virtual machine.

8. The computer-readable storage medium of claim 6 or 7, wherein the virtual disc comprises a first portion containing the decryption-master-boot-record, a second portion containing the decryption program, a third portion containing a second part of the virtual disc image, a fourth portion containing a first part of the virtual disc, and a fifth portion containing storage space, wherein the size of the fifth portion is larger than or equal to the second portion, wherein the combined size of the first and second portions is less than or equal to the size of the fourth portion, wherein the at least partially encrypted virtual disc image is decrypted by:

- decrypting (806) the second part of the virtual disc image;
- copying (808) the decryption program to the fifth portion of the virtual disc;

- decrypting (810) a portion of the first part of the virtual disc image;
- copying (812) the decrypted portion of the first part of the virtual disc image to the second portion of the virtual disc;
- decrypting (814) the remainder of the first part of the virtual disc image; and
- copying (816) the decrypted remainder of the first part of the virtual disc image to the first portion of the virtual disc.

9. The computer-readable storage medium of claim 8, wherein execution of the instructions further causes the processor to erase data in the fourth and fifth portion of the virtual disc after copying the decrypted remainder of the first part of the virtual disc image to the first portion of the virtual disc.

10. The computer-readable storage medium of any one of claims 6 through 9, wherein execution of the instructions further causes the virtual machine to re-boot after decrypting the at least partially encrypted virtual disc image.

11. An encrypted-virtual-disc computer-readable storage medium (914, 936) containing a virtual disc (924); wherein the virtual disc comprises: a decryption-master-boot-record (414), a decryption program (416), and an at least partially encrypted virtual disc image (406', 408', 923); wherein the decryption program comprises machine executable instructions for decrypting the at least partially encrypted virtual disc image in accordance with a cryptographic key (944); wherein the decryption program comprises machine executable instructions that when executed by a processor (932) cause the processor to:

- receive (118, 216, 314, 704, 804) the cryptographic key;
- decrypt (120, 218, 316, 706, 806, 808, 810, 812, 814, 816) the encrypted at least partially encrypted virtual disc image in accordance with the cryptographic key and the decryption program.

12. The virtual-disc computer-readable storage medium of claim 11, wherein the virtual disc comprises:
- a first portion (414) containing the decryption-master-boot-record;
- a second portion (416) containing the decryption program,
- a third portion (418) containing a second part of the virtual disc image (408');
- a fourth portion (420) containing a first part of the virtual disc image (406'); and
- a fifth portion (422) containing storage space, wherein the size of the fifth portion is larger than or equal to the second portion, wherein the combined size of the first and second portions is less than or equal to the size of the fourth portion.

13. The virtual-disc computer-readable storage medium of claim 12, wherein the at least partially encrypted virtual disc image is decrypted by:
- decrypting (806) the second part of the virtual disc image;
- copying (808) the decryption program to the fifth portion of the virtual disc;
- decrypting (810) a portion (504) of the first part of the virtual disc image;

- copying (812) the decrypted portion of the first part of the virtual disc image to the second portion of the virtual disc;
- decrypting (814) the remainder of the first part of the virtual disc image; and
- copying (816) the decrypted remainder (506) of the first part of the virtual disc image to the first portion of the virtual disc.

14. The virtual-disc computer-readable storage medium of claim 11 or 13, wherein the virtual disc contains a decryption-status-data-file for storing the progress of the decryption of the at least partially encrypted virtual disc image, wherein execution of the machine executable instructions of the decryption program further causes the processor to:
- update the decryption-status-data-file during decryption of the at least partially encrypted virtual disc image; and
- check the decryption-status-data-file when starting the decryption of the at least partially encrypted virtual disc image.

15. A method for encrypting a virtual disc comprising a virtual disc image, the method comprising:
- receiving (600) the virtual disc;
- increasing (602) the size of the virtual disc;
- writing (604) a decryption-master-boot-record and a decryption program to the virtual disc;
- encrypting (606) at least a portion of the virtual disc image, wherein the decryption program is provided to decrypt the at least partially encrypted virtual disc image in accordance with a cryptographic key (944).

Fig. 1

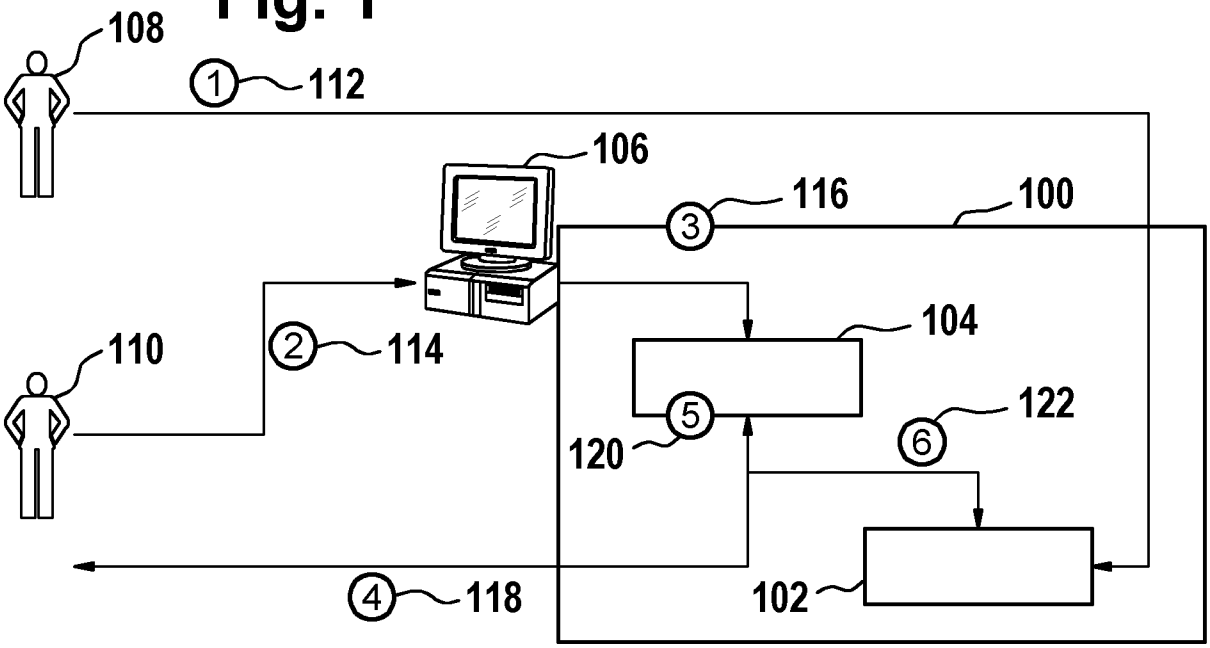


Fig. 2

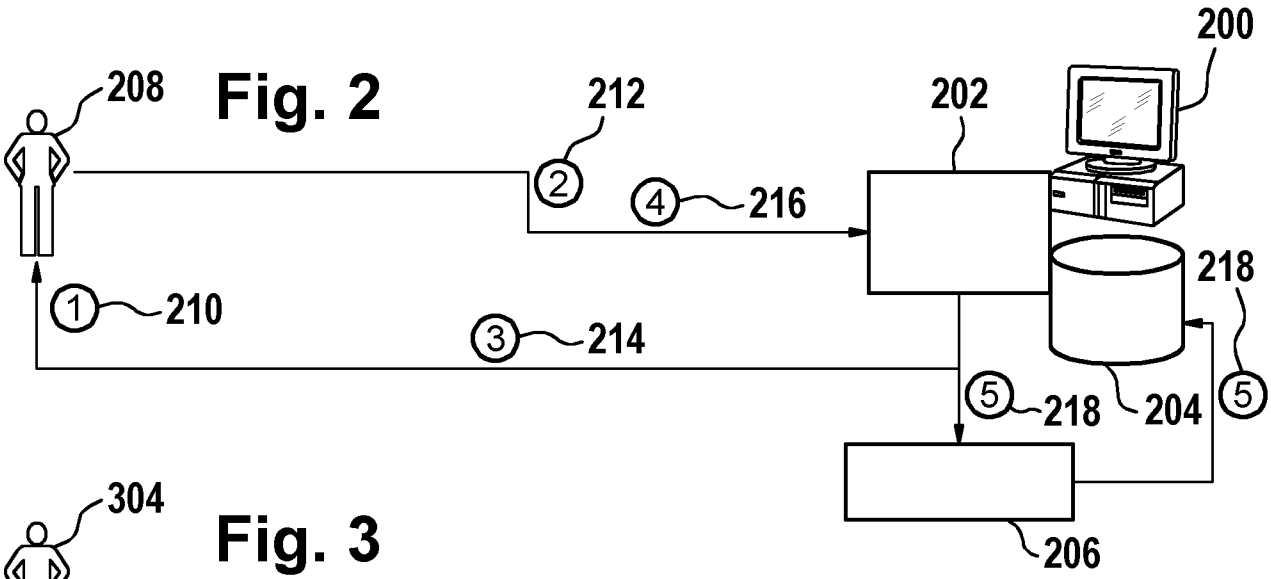
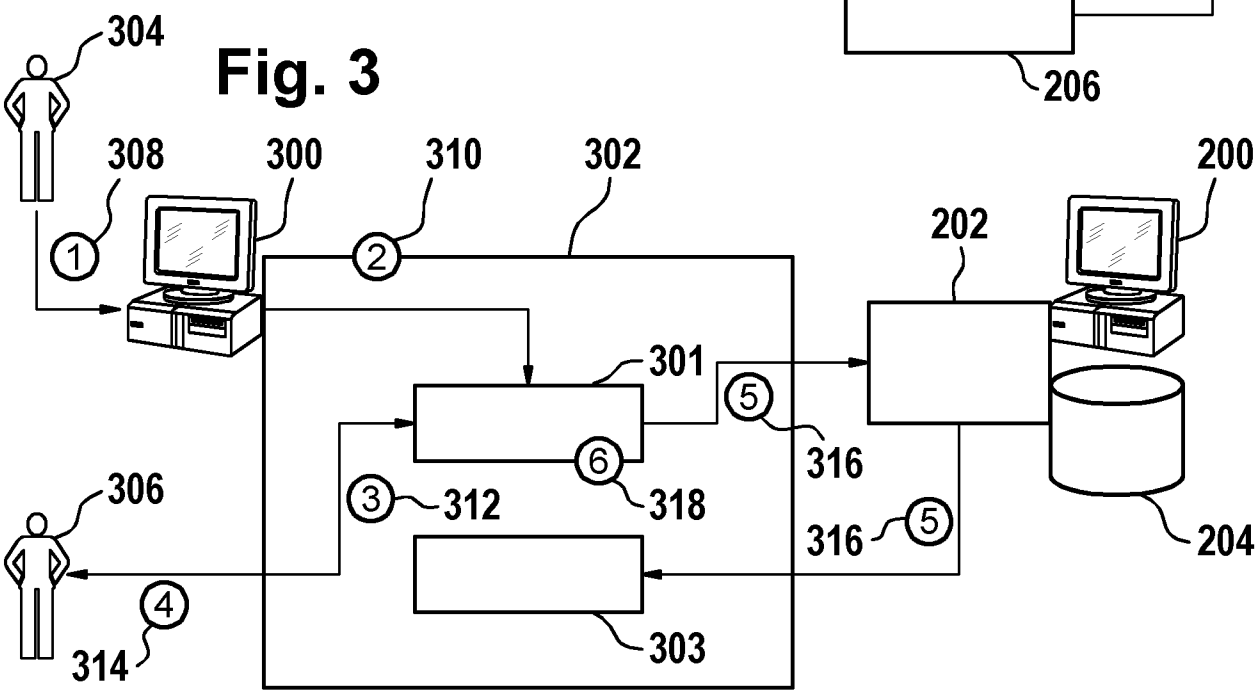


Fig. 3



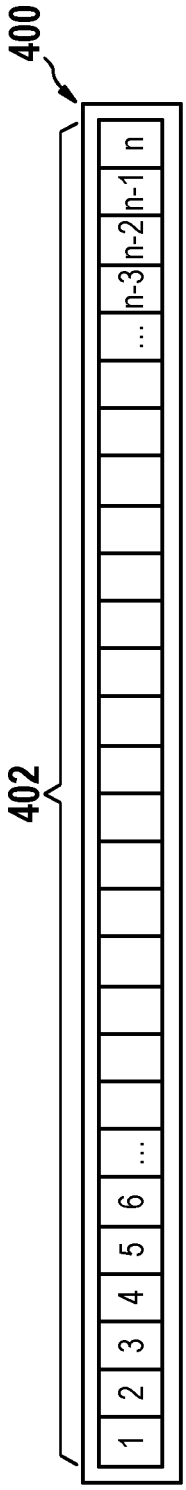


Fig. 4A

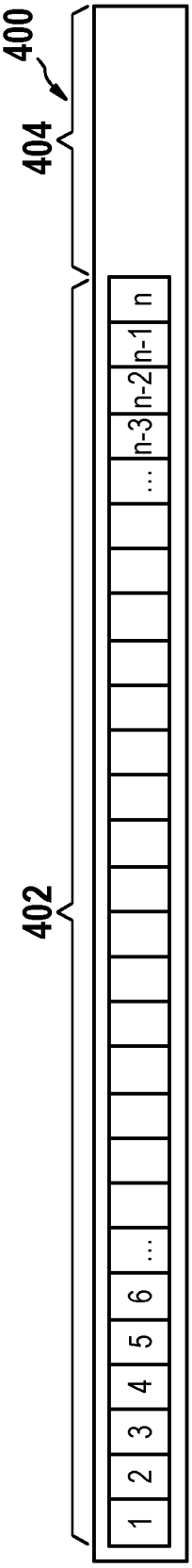


Fig. 4B

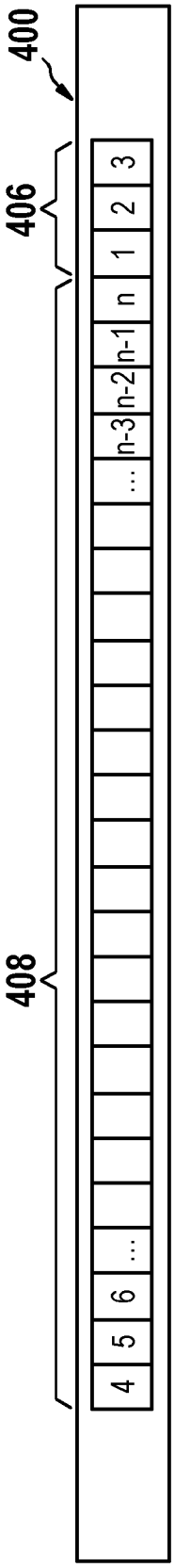


Fig. 4C

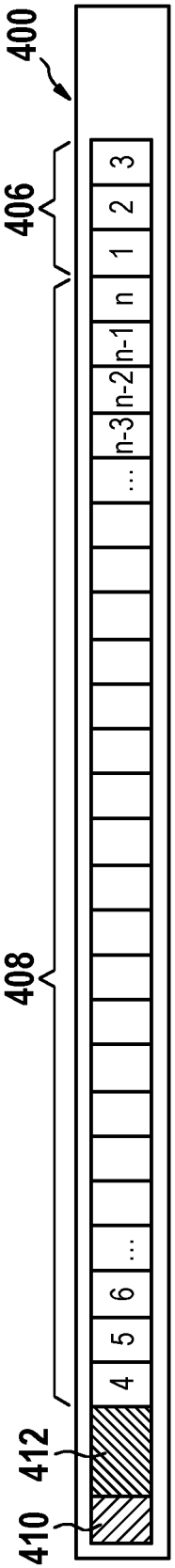


Fig. 4D

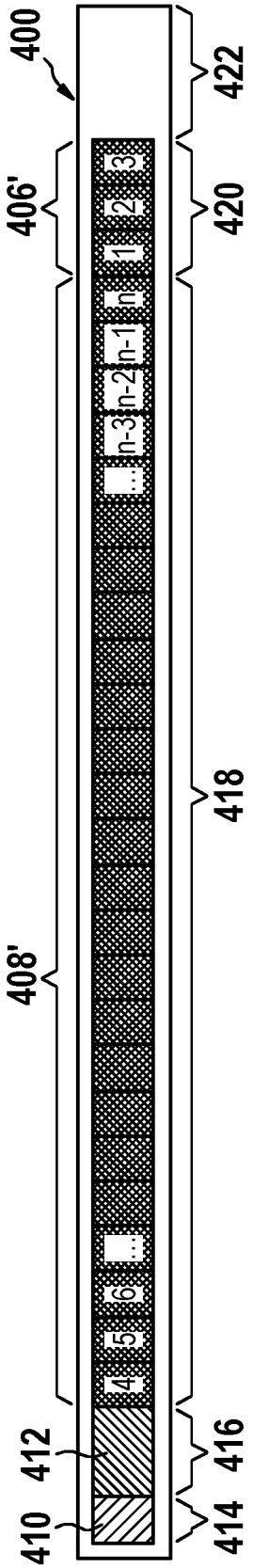


Fig. 4E

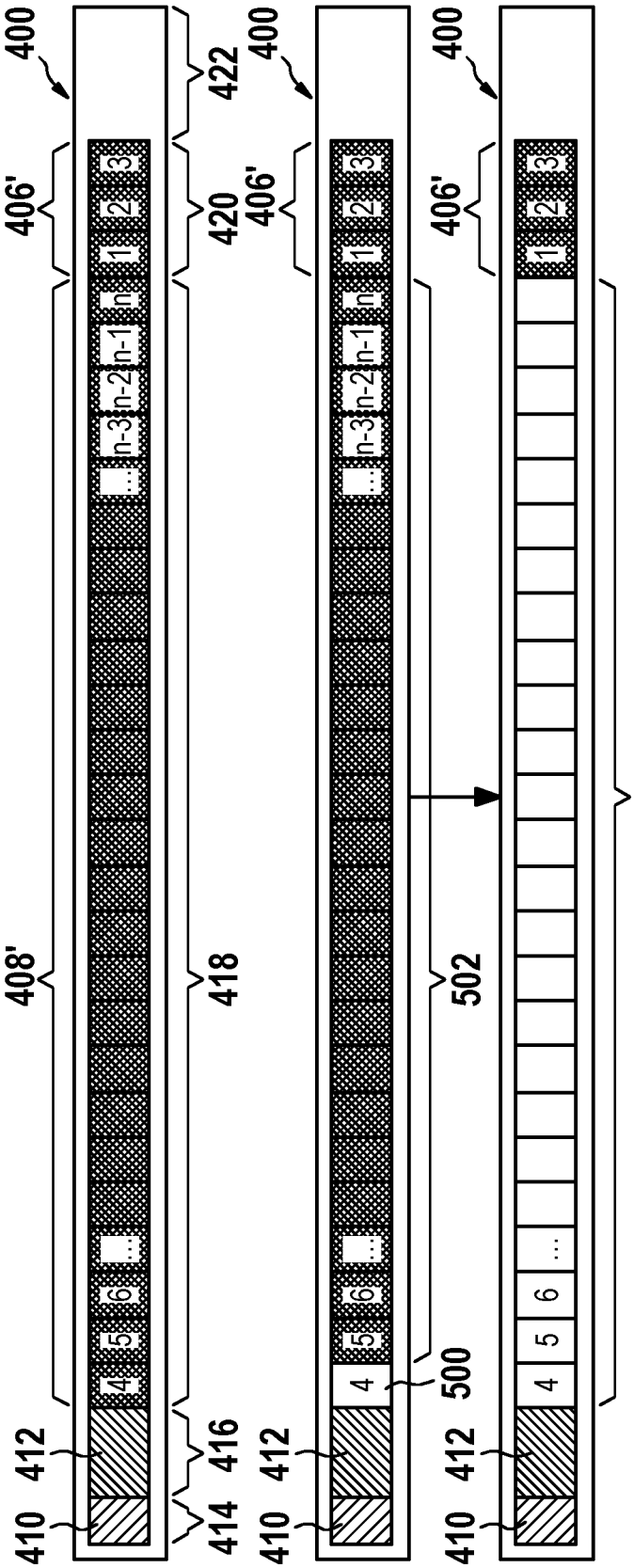


Fig. 5A

Fig. 5B

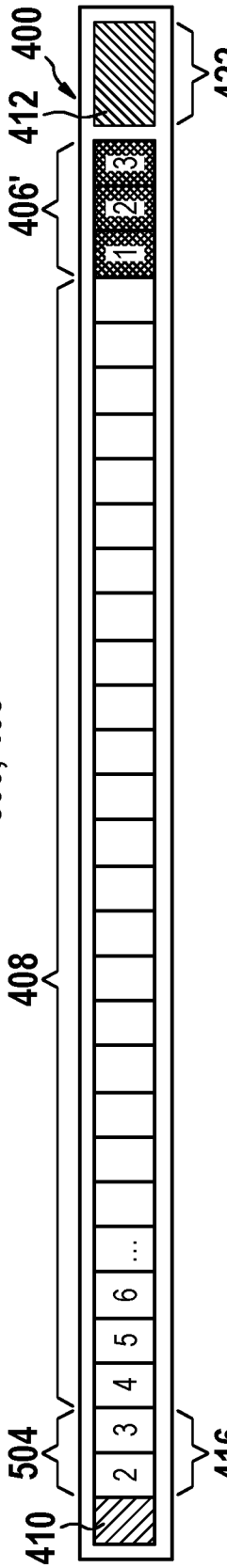


Fig. 5C

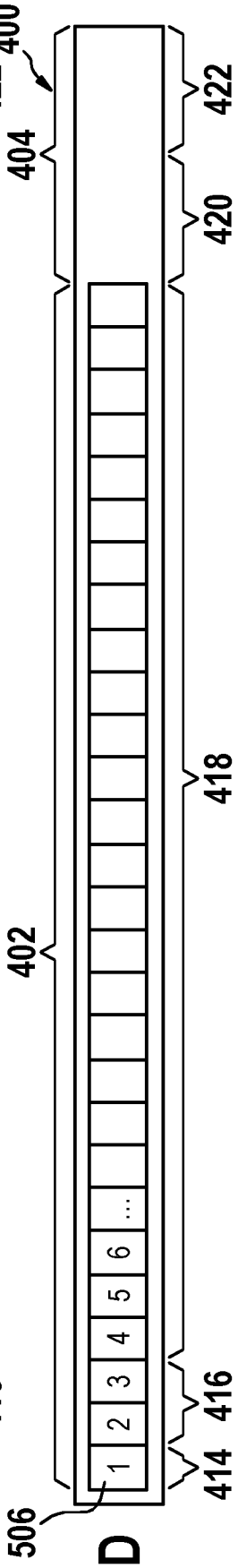


Fig. 5D

Fig. 6

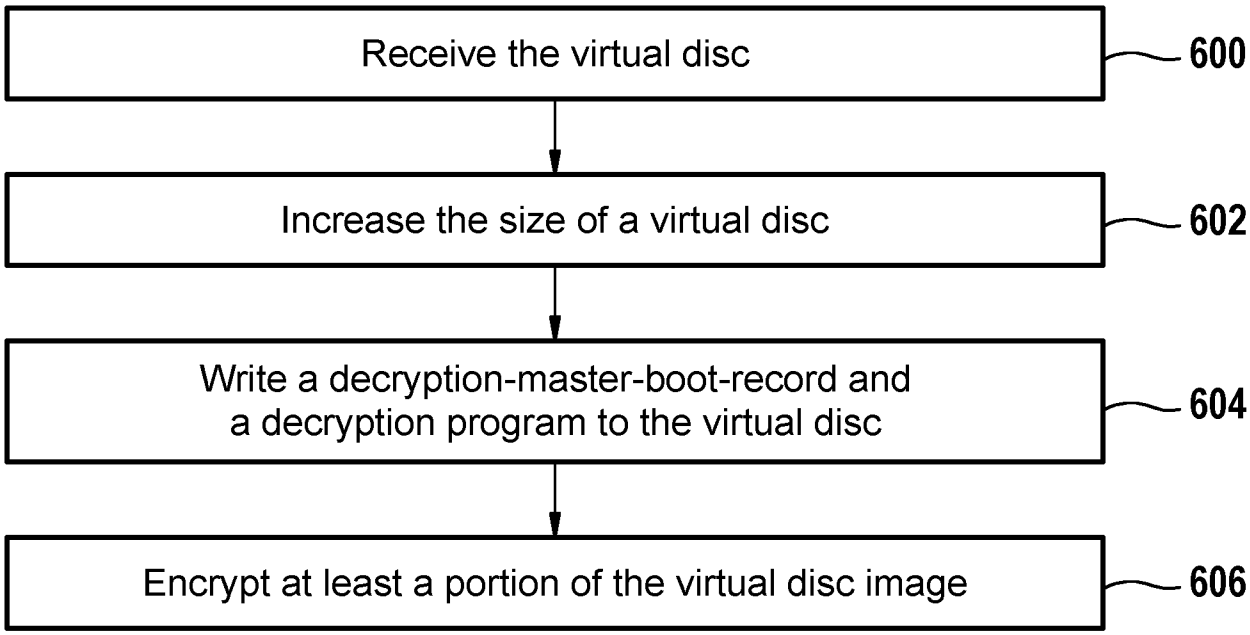

Receive the virtual disc
600
Increase the size of a virtual disc
602
Write a decryption-master-boot-record and a decryption program to the virtual disc
604
Encrypt at least a portion of the virtual disc image
606

Fig. 7

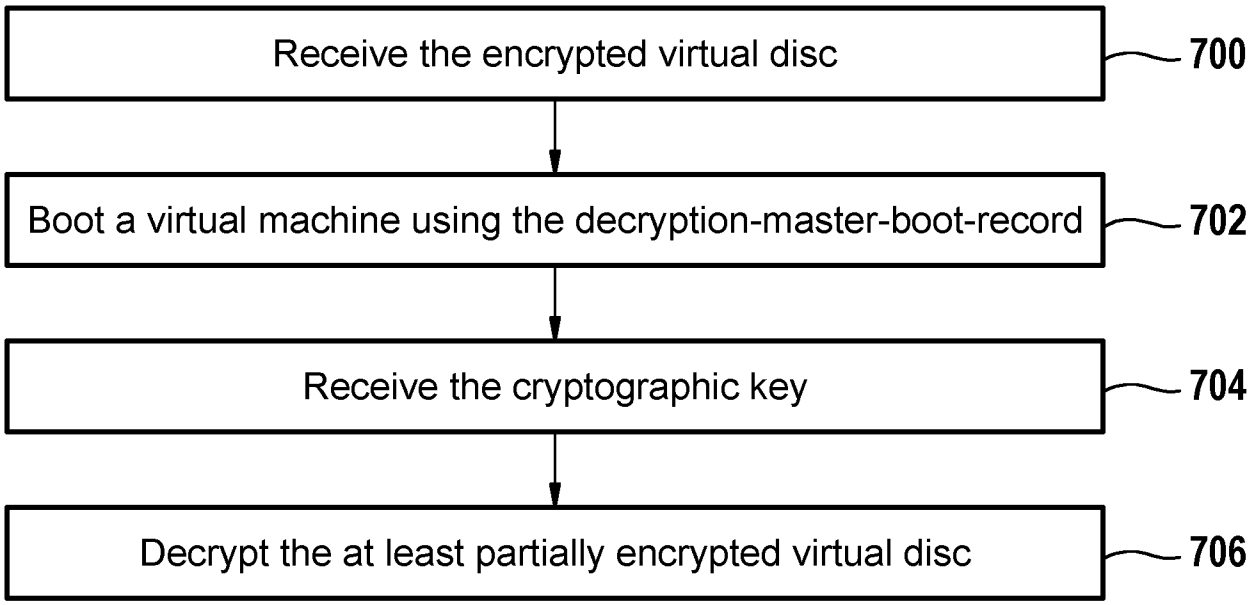

Receive the encrypted virtual disc
700
Boot a virtual machine using the decryption-master-boot-record
702
Receive the cryptographic key
704
Decrypt the at least partially encrypted virtual disc
706

Fig. 8

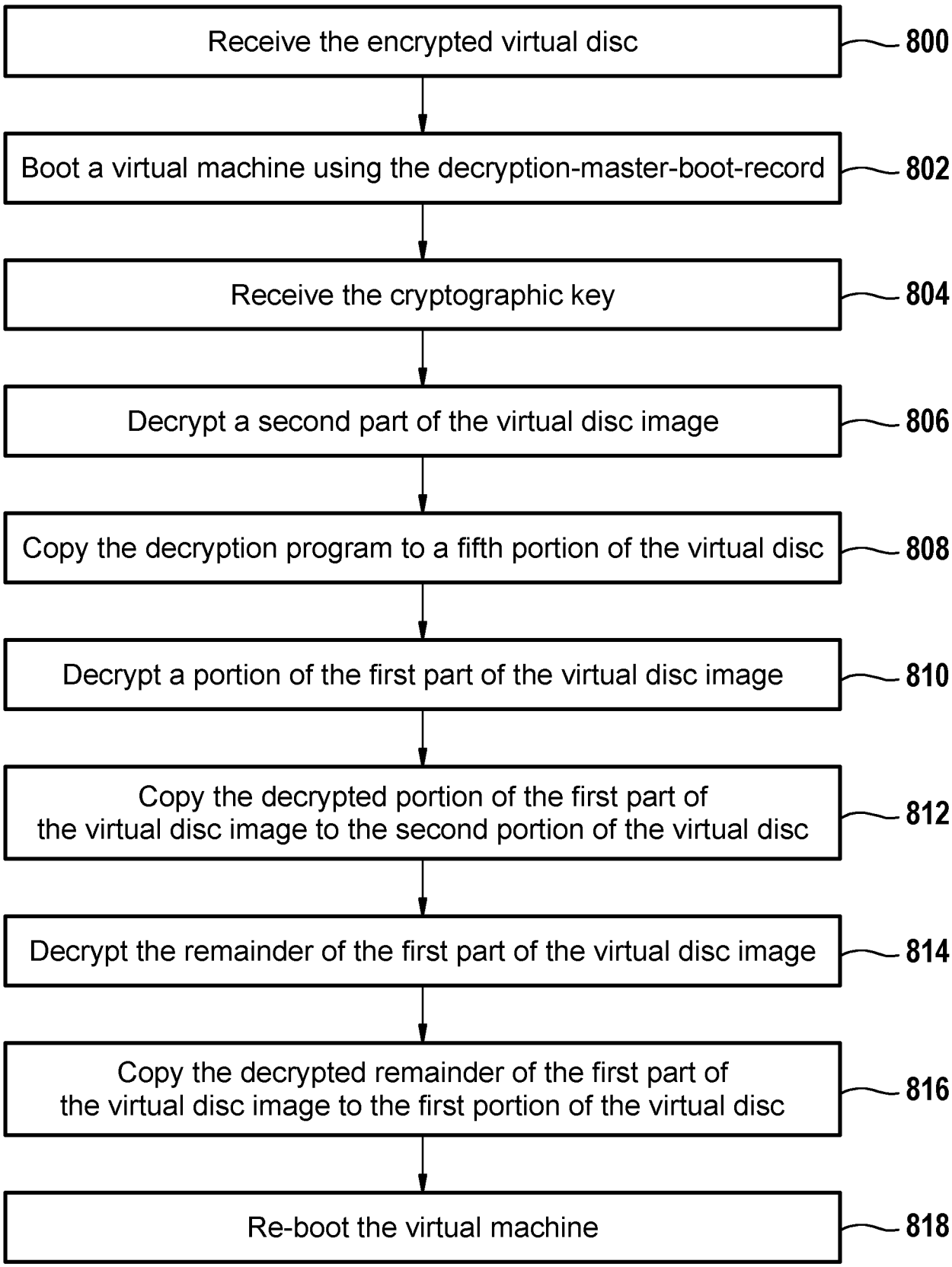
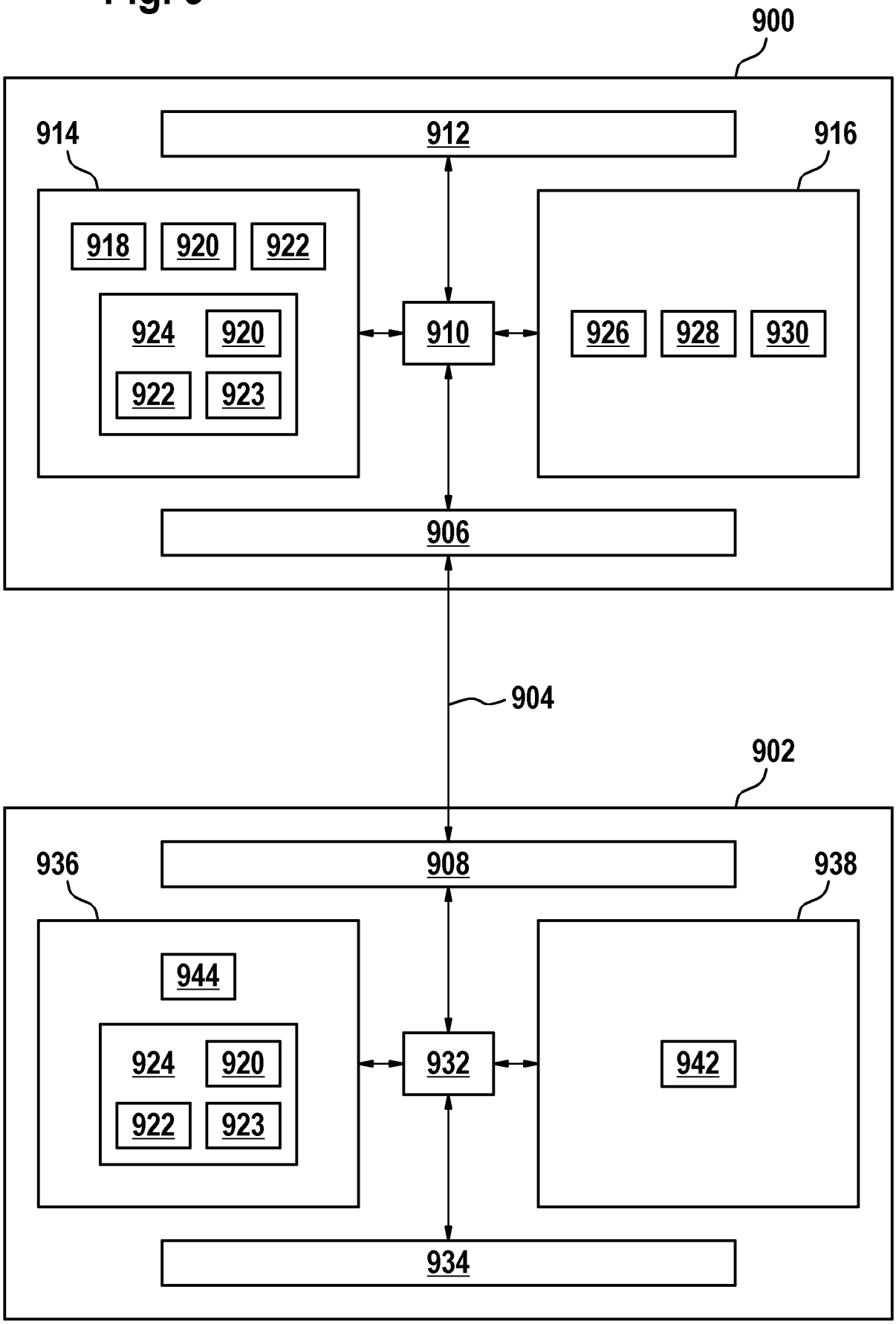

Receive the encrypted virtual disc
800
Boot a virtual machine using the decryption-master-boot-record
802
Receive the cryptographic key
804
Decrypt a second part of the virtual disc image
806
Copy the decryption program to a fifth portion of the virtual disc
808
Decrypt a portion of the first part of the virtual disc image
810
Copy the decrypted portion of the first part of the virtual disc image to the second portion of the virtual disc
812
Decrypt the remainder of the first part of the virtual disc image
814
Copy the decrypted remainder of the first part of the virtual disc image to the first portion of the virtual disc
816
Re-boot the virtual machine
818

Fig. 9


900
914
912
916
918
920
922
924
920
922
923
910
926
928
930
906
904
902
936
908
938
944
924
920
922
923
932
942
934

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2011/069323

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F21/02 G06F9/455
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 practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2008/049008 A2 (MANAGE IQ INC [US]; FITZGERALD JOSEPH [US]; BARENBOIM OLEG [US]) 24 April 2008 (2008-04-24) paragraph [0010] paragraph [0027] - paragraph [0035] paragraph [0040] - paragraph [0046] paragraph [0054] paragraph [0077] paragraph [0084] - paragraph [0088] paragraph [0098] - paragraph [0105] abstract; claims 1-48	1-15

A	US 2004/030822 A1 (RAJAN VIJAYAN [US] ET AL) 12 February 2004 (2004-02-12) the whole document	1-15

	-/--	

[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
28 February 2012	08/03/2012

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, Fax: (+31-70) 340-3016	Powell, David

1

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

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2011/069323

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 7 428 636 B1 (WALDSPURGER CARL A [US] ET AL) 23 September 2008 (2008-09-23) the whole document	1-15
A	EP 0 770 950 A2 (SCM MICROSYSTEMS INC [US]) 2 May 1997 (1997-05-02) the whole document	1-15

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2011/069323

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2008049008 A2	24-04-2008	NONE	
US 2004030822 A1	12-02-2004	AU 2003252181 A1	25-02-2004
		CN 1688981 A	26-10-2005
		EP 1543424 A2	22-06-2005
		HK 1082304 A1	03-07-2009
		JP 2005535962 A	24-11-2005
		US 7076509 B1	11-07-2006
		US 2004030822 A1	12-02-2004
		WO 2004015522 A2	19-02-2004
US 7428636 B1	23-09-2008	US 7428636 B1	23-09-2008
		US 2008320316 A1	25-12-2008
EP 0770950 A2	02-05-1997	EP 0770950 A2	02-05-1997
		US 6075858 A	13-06-2000