

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
5 October 2006 (05.10.2006)

PCT

(10) International Publication Number
WO 2006/103679 A2

(51) International Patent Classification:
G06F 9/00 (2006.01)

(21) International Application Number:
PCT/IL2006/000411

(22) International Filing Date: 30 March 2006 (30.03.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/667,035 1 April 2005 (01.04.2005) US

(71) Applicant (for all designated States except US): **GED-I LTD.** [IL/IL]; 13 Sigalit Street, 42203 Netanya (IL).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **ATZMON, Elisha** [IL/IL]; 13 Berliner Street, 69057 Tel-aviv (IL). **COHEN, Giora** [IL/IL]; 1 Carmel Street, 45815 Bnei Dror (IL). **SAAR, David** [IL/IL]; 13 Sigalit Street, 42203 Netanya (IL).

(74) Agent: **REINHOLD COHN AND PARTNERS**; P.o.b. 4060, 61040 Tel-aviv (IL).

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

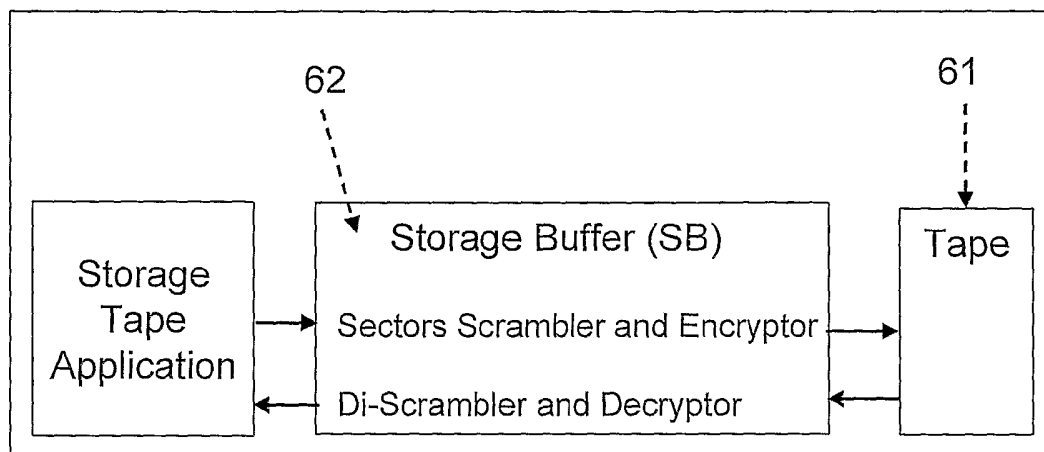
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report

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

(54) Title: A METHOD FOR DATA STORAGE PROTECTION AND ENCRYPTION



(57) Abstract: A method for encrypting and/or hiding data to a storage device is given. The method is based on any of the following algorithms: 1. Formatting any storage uniquely while storing the information about the unique format to a Geometrical Cell Location Table (GCLT). 2. Storage-sectors, based on standard formatting are uniquely shuffled while creating Shuffled LBA (SLBA). 3. A method of encrypting a sector-content by using more than one key. 4. "Dating" prevention of the sectors content in a hard disk drive by electro-magnetic instruments. This is achieved by renewing data of randomly selected sectors. 5. A method for keeping the encrypted content of a hard disk drive in its maximal encryption power complexity by preserving the hard drive disk fully filled with encrypted data (even the empty sectors are filled with encrypted data). By implementing the above algorithms to a hard disk drive the encryption complexity power stays constant in time.

WO 2006/103679 A2

A METHOD FOR DATA STORAGE PROTECTION AND ENCRYPTION

5

FIELD OF THE INVENTION.

The present invention is in the general field of data storage protection and encryption.

10

BACKGROUND OF THE INVENTION

Data or information encryption has a long history of thousands of years. Today's communication and information world, which follow so far the Moore law, increases the need for security.

15

Data resides on storage media accessible to and used by applications, that run on computers that are connected by networks (LAN, WAN, Internet, Intranet, etc). The common security concept is of layers of security, starting by securing the outer circle (networks) and gradually securing the inner circles such as the computers (by access control products) and the applications themselves.

20

Today's solutions for the outer circles include the intrusion prevention systems (like firewalls), application solutions (like anti-spy, anti-virus, etc.) and physical access control (like keys – biometric, etc.). There are also solutions for the most inner circle i.e. the one of the information itself, based on encryption. However, as will be explained, current solutions are not suitable for large storage media such as disks and tapes.

25

The problem with current encryption methods for data storage is that almost all methods were derived from encryption methodologies that were specially developed for data transfer (i.e. communication) rather than data storage (i.e. stable, static, etc.). There are many differences between the two and the main ones are summarized in the following table:

30

Table 1. - Information nature of communication vs. storage

	Subject	Communication	Storage
1.	Data lifetime	Short Duration	Long Duration
2.	Unit Size	Packets	Files, Sectors
3.	Data sensitivity	Usually Low	High
4.	Applicability of Statistical analysis in decryption methods	Low	High
5.	Data ownership	Mostly Shared between the sender and the receiver	For use by the owner only

Therefore the encryption solutions for data communication were basically
5 developed to be fast and be applicable for a few packets (leading to relatively small key).

National regulations also enforce the communication software manufacturers
(and therefore the end users too) to apply only relatively small key for reasons
known to all – the authorities must be able to decrypt the data in reasonable
10 time. It should be noted that in the communication world a reasonable time is as close as possible to real time, since the nature of the transferred information is so, especially the information that bothered the authorities.

However, stored information, as presented in the **Table 1** is of a different nature
15 than of communication. Stored information is usually privately protected by law (i.e. non shared), is found on large storage media that contain very large amounts of sensitive data (especially in every large organizations).

Here are some fundamental facts regarding data storage media: the information
20 is very sensitive; the storage media contain enormous amounts of data, which would ease the possibility of decryption based on advanced statistical analysis methods and the data is (as a rule) permanent or at least stored for relatively long durations.

These fundamental facts lead us to realize that encryption methods using relatively short keys (as being used in communication) are not sufficient (i.e. do not provide adequate protection) for data storage media.

- 5 There is thus a need in the art for a powerful encryption method in order to prevent an unauthorized object to understand the stored information (after reaching an access to it). There is further need in the art to provide an encryption technique which combines a few levels of data protection and encryption.
- 10 There follows a brief description on how data is stored on a disk.

Disk Structure

- As seen in **Figure 1**, a hard disk can contain a few physical plates (1) (**Figure 1.a**), each physical plate is divided to tracks (2) (**Figure 1.b**), where each track
- 15 (2) has its own fixed radius. The tracks are then divided to sectors (sometimes called also blocks) (3) (**Figure 1.c**). The sectors' division of each of the tracks is azimuthally a division of the circle to a fixed number. Once, all the tracks contained the same number of sectors, but today different tracks can have different numbers of sectors (**Figure 1.f**). Essentially, the disk-tracks and disk-
- 20 sectors are an outcome of a logical division and are disk dependant. The known disk Format is a disk geometrical mapping of all the areas as defined by the track-sectors. The physical disk (hard drive) can be divided to logical units which are called partitions, where each partition is labeled, and virtually behaves as a physical unit.

25

Related art:

“Encryption Anywhere Hard Disk” of GuardianEdge Technologies Inc.

Abstract:

- 30 Encryption Anywhere Hard Disk uses a government certified AES cryptographic algorithm and a 256-bit encryption key to protect 100% of the data on a Windows hard drive – including the master boot record, the operating system files, applications and all data files. The program even encrypts unused

sectors of the hard drive to assure air-tight security of computer and network systems.

US patent application no. 20030070083, by Nessler Kai Wilhelm "Method and
5 device for encryption/decryption of data on mass storage device".

Abstract:

The present invention provides a method and device performing
encryption/decryption of data on a mass storage media. A plurality of different
encryption/decryption algorithms with associated keys can be utilized on
10 different storage areas of the media, such as blocks/sectors on a hard disk drive,
thereby increasing the security of the data significantly. In addition, the present
invention provides a method and device that combines said keys with random
numbers for a further increase of the security of the data. In a preferred
embodiment of the invention, the block/sector number is used to select an
15 algorithm with associated key and random number. The present invention can
also be utilized to encrypt/decrypt e-mails, networking traffic etc. and other
types of electronic data. In addition, said device provides an authentication
session when booting a computer from a hard disk drive, and can even be used
to select a certain operating system and/or environment associated with a certain
20 key carrier by changing the Master Boot Record of the disk system.

SUMMARY OF THE INVENTION

25 The invention provides for a method for formatting of a disk that includes at
least one track and the said track is capable of being divided into at least two
geometrical cells (GCs), comprising: (a) dividing at least one of the tracks into
at least two neighboring geometrical cells (GCs); at least one of said GCs is
bordered by at least different internal radius and/or different outer radius
30 compared to the respective radius of a neighboring cell of said track; (b) storing
Geometrical Cell Location Table (GCLT), for accessing data in said cells, said
accessing includes inserting, updating or deleting data.

The present invention further provides for a method for formatting a disk that includes at least one track and the said track is capable of being divided into at least two geometrical cells (GCs), comprising: (a) dividing at least one of the tracks into at least two neighboring geometrical cells (GCs); at least one of said
5 GCs defining a different angular size compared to the angular size of a neighboring GC of said track; (b) storing Geometrical Cell Location Table (GCLT), for accessing data in said cells, said accessing includes inserting, updating or deleting data

10 In accordance with certain aspects of the invention there is provided a counterpart system.

Yet further provided by the present invention is a method for formatting a disk that includes at least one track and the said track is capable of being divided into
15 at least two geometrical cells (GCs), comprising: (a) dividing at least one of the tracks into geometrical cells (GCs); at least one of said GCs is bordered by at least different internal radius and/or different outer radius compared to the respective radius of a neighboring cell of said track; (b) at least one of said GCs defining a different angular size compared to the angular size of a neighboring
20 GC of said track; (c) storing Geometrical Cell Location Table (GCLT), for accessing data in said cells, said accessing includes inserting, updating or deleting data.

Further provided by the present invention is a method for encrypting data stored
25 in a disk, the disk includes at least two geometrical cells (GCs), comprising: (a) the geometrical cells (GCs) constituting sectors and constructing Logical Block Address (LBA), for facilitating access of sector of interest; (b) reshuffling at least two of said sectors addresses, giving rise to Shuffled LBA (SLBA), such that a sector from among said at least one sectors having a sector address in said
30 LBA will have a different address in said SLBA; said SLBA facilitating access to sector of interest.

Further provided by the present invention is a method for encrypting data stored in a disk, the disk includes at least one track and track is capable of being

divided into at least two geometrical cells (GCs), comprising: (a) dividing the disk into tracks and geometrical cells (GCs) constituting sectors and constructing Logical Block Address (LBA), for facilitating access to sector of interest; (b) encrypting data in at least one of said sectors using at least two encryption keys.

The present invention further provides a method for encrypting data stored in a disk, the disk includes at least one track and the said track is capable of being divided into at least two geometrical cells (GCs), comprising: (a) dividing the disk into tracks and geometrical cells (GCs) constituting sectors and constructing Logical Block Address (LBA), for facilitating access of sector of interest; (b) encrypting data in at least two of said sectors using at least two encryption algorithms.

Further provided by the present invention is a method of encrypting data stored to a tape comprising: (a) buffering at least two sectors before storing them to the tape, (b) shuffling the said buffered sector-addresses found in the buffer, (c) storing the information on the said shuffling to a Tape Sectors Shuffling Table (TSST), and (d) storing the shuffled sectors in their new order to the tape.

BRIEF DESCRIPTION OF THE DRAWINGS

For a better understanding, the invention will now be described by way of example only, with reference to the accompanying drawings, in which:

Fig 1A -F. – illustrates a Disk Architecture - Physical and Logical Structure, in accordance with the prior art;

Fig 2-2A. – illustrates schematically Radial Modification of Sectors Location, in accordance with an embodiment of the invention;

Fig 3-3A. - illustrates schematically Angular Modification by Two Type of Sectors, in accordance with an embodiment of the invention;

Fig 4A-4B. - illustrates schematically Randomly Scrambled Logical
5 Sectors, in accordance with an embodiment of the invention;

Fig 5A-5B. - illustrates schematically an Algebraically Encrypted Sectors after Random Scrambling, in accordance with an embodiment of the invention;

10

Fig 6. - illustrates schematically a Tape Encryption Schematic, in accordance with an embodiment of the invention;

Fig 7. - illustrates schematically a CD/DVD Encryption Schematic, in
15 accordance with an embodiment of the invention;

Fig 8. - illustrates schematically the major three levels of Protection & Encryption, in accordance with an embodiment of the invention;

20 **Fig 9.-** illustrates schematically Randomly Scrambled Sectors Distributed Over Two Tracks, in accordance with an embodiment of the invention;

Fig 10.- illustrates schematically Scramblers and Encryption Board Major
25 Modules, in accordance with an embodiment of the invention; and

Fig 11A-11B.- illustrates schematically Protected/Encrypted Storage in Operation Mode, in accordance with an embodiment of the invention.

30

DETAILED DESCRIPTION OF THE INVENTION

In accordance with an aspect of the invention, there is provided Geometrical Format Modification (GFM). As described above in the background chapter,
5 any disk is formatted in the form illustrated in **Figure 1**. The fact that a disk is characterized by cylindrical symmetry causes us to prefer cylindrical coordinates in elaborating on the disk format. The cylindrical coordinates will be denoted by $[r, \theta]$, where the r -coordinate and the θ -coordinate stand for the radial dependency and the angular dependency, respectively.

10

As seen in **Figure 1**., the formatting of the disk (hence the format operation) divides the disk (or a partition) geometrically to cells (**3**) (i.e. sectors – see **Figure 1.c**), where each Geometrical Cell (GC) is bordered by two fixed successive radii (e.g. r_i & r_{i+1} – (**6**), (**7**) **Figure 1.c**) and fixed successive angles
15 (e.g. θ_j & θ_{j+1} – (**8**), (**9**) **Figure 1.c**). Once, all tracks had the same number of GCs., and moreover the alignment of the θ -division was preserved through all of the tracks (**Figure 1.c**). This of course caused disk-area waste, since sectors in the outer tracks were physically longer. On the other hand, within fixed angular disk speed, each GC is passed through, under the disk arm reader/writer head, at
20 the exact time lap. Today, the disk formatting methods enable more sectors on the outer tracks (**Figure 1.f**), however on each specific track, the angular division to sectors is of equal θ gap. Thus all sectors are identical in area and shape upon each specific given track. Within this context it should be mentioned that the disk format operation is eventually done on two levels: Low Level
25 Format and High Level Format. As regards to the Low Level Format for disks, it is done by the disk manufacturer and encoded to hardware disk controller, in order to improve disk performance. However, as regards recordable media such as diskettes, CDs, DVDs, etc, the Low Level Format is part of the whole format operation.

30

In accordance with certain embodiment of the invention, one applies r -modifications to the standard formatting method, while the angular distribution is kept standard. One possible modification with regard to the r frame is:

Some of the GCs on some of the tracks have different radii, whereby the term "some of the one" means "any number out of the total number". See e.g. (21) & (22) **Figure 2** that represent two possible track modifications.

5 In accordance with certain embodiments of the invention, one applies θ -modifications to the standard formatting method, while the radial distribution is kept standard. Some possible modification with regard to the θ frame eventually means sectors with different angular sizes. It follows that some of the possibilities are:

10 Some predetermined GC sizes, randomly or periodically alternated within each of the tracks. The order can be either the same for all the tracks, or different for each of the tracks. See e.g. **Figure 3.**, two angular GC sizes exist (represented by A & B (31), (32) **Figure 3.** respectively).

15 The GCs have no standard sizes. Each GC size is randomly fixed.

In accordance with certain embodiments of the invention, one applies both r-modification & θ -modification simultaneously. Some of the options are:

Any combination of r-modification and θ -modification, whether, the disk is
20 fully "covered" by the GCs, or even contains empty holes as seen e.g. in (23) **Figure 2.** or any GCs disk coverage, other than described, including any option of non periodic GCs coverage. See e.g. **Figure 2.**

In accordance with certain embodiments of the invention, one applies empty
25 physical spaces on the disk, i.e. empty areas. The empty areas can be of the same shape and size or of different shapes and sizes (e.g. (23) **Figure 2**)

In accordance with certain embodiments of the invention all the information of the new modified formatting method (i.e. GFM) is stored in a way to enable
30 access, reading and writing to the disk. We will denote this table by the term "GCLT" – Geometrical Cell Location Table.

The GCLT will always have a zero coordinate on the disk (e.g. see (24) **Figure 2** & (33) **Figure 3**) that is automatically recognized. The GCs location can be easily calculated based on GCs sizes and the GCLT. As an example of GCLT,

see **Figure 2.A** and **Figure 3.A**.

Figure 2.A is related to the example of radial modification. So the first row describes the initial locations that start the tracks, while the zeros represent a move towards the center while stepping forward to the next sector (GC) or the ones in outward direction respectively, both of them while moving in anti-clockwise direction. Whenever there is a gap the coordinate of the next track GC is given. So the left column represents the track starts from the disk-sector number 0, sector (GC) number 3 in the radial direction, while the right column represents the track starts in disk-sector number 0, sector (GC) number 4 in radial direction. Figure 3.A is the GCLT related to case of angular modification. Each column is related to different track started from the cell coordinate given in the first row, while advancing in anti-clockwise direction. The zeros represent sectors (GCs) of type A, while the ones represent the sectors (GCs) of type B.

15 In accordance with certain embodiments of the invention the reading/writing process from GFM disk is performed by applying some optimization methods. These optimization methods are necessary to reduce the time per operation. The main criteria will be then minimal read/write header movement. Detailed description will be further elaborated in special sections referring to this issue.

20 However, it is clear that any of the above Disk Format Modifications will be paid in terms of latency and more intensive read/write arm movements.

In accordance with certain embodiments of the invention, each disk (or a partition) can be formatted uniquely. In that case, there will be no other disk (or partition) that is formatted identically (i.e. with identical GCLT). In order to reach a unique format disk one has to implement a faithful random number generator, where the generated numbers are used for the GCLT creation. For example, one can think of mapping the even numbers to short GC sectors, while odd numbers will be mapped to long GC sectors (as demonstrated in **Figure 3** where (31) point to short GC sector, while (32) point to long GC sector). So the short-long GC sectors' sizes will be randomly ordered upon the tracks. Each of the above other methods will need its implementation.

In accordance with certain embodiments of the invention, the GFM as described so far enables hiding the data on a disk, if the disk format uniqueness is kept and the GCLT is accessible only to the certified data owner. This of course is due to the fundamental fact that without the GCLT availability to the operation system,
5 the physical location of GCs is not available to the operation system. Therefore, the GCLT should be stored in at least one of the following ways:

Not on the same GFM disk, but elsewhere and accessible to the operating system.

10 It can be on the same disk but in an exact known location to the operating system.

It can be encrypted, to prevent easy use of an uncertified object.

It should be clear that if the disk is being scanned or analyzed physically in a lab by sensitive electromagnetic instrumentation, the stored data on a GFM disk and
15 even the GCs shapes and sizes, in principle, can be revealed. This enables disk rescue operation in case of disk functionality failure. However, this is not a trivial process and might take very long time.

Since all the above described with regard to the GFM method is related to Low
20 Level Format, one implements the GFM by modifying the disk controller hardware, when it is applied to disks. For Diskette, CD, DVD- like media it is done through the Low Level Format executed by the Format operation (which an integral part of any operating system).

25

In accordance with an aspect of the invention, there is provided the Randomly Scrambled Logical Sectors (RSLS). These sectors are addressed by the LBA - Logical Block Address - each sector (i.e. a block) is assigned a unique "sector number". In essence, the sectors are numbered 0, 1, 2, etc. up to (N-1), where N
30 is the number of sectors on the disk. As described in the opening section, information that is stored on disk (or a partition) is stored in sectors. In the next few paragraphs, the methods of "hiding" the data on disk will be presented. The main idea is based on redirecting read/write operations to randomly selected locations, such that the end result is a completely scrambled disk. Instead of

accessing sectors by their natural address, the addresses now get converted (redirected) to new addresses. The means by which the sectors are redirected is stored in an address lookup-table, which is hence termed "Shuffled LBA" (SLBA).

5 With this method, we do not touch the sectors physical (geometrical) location on disk as described in the former sections about the GFM method, but rather keep the standard formatting structure. However, it will be further clarified that both methods can be combined.

10 In accordance with certain embodiment of the invention, one scrambles the LBA. The term LBA without scrambling simply refers to the natural numbers 0,1,2 ...n-1 that represent the sector numbers (or serial numbers).

Randomly Scrambled LBA means randomly remapping or re-addressing of the
15 sectors serial numbers. So each sector after the scrambling process is re-addressed. The mapping between the original LBA and the scrambled LBA is stored by another table called SLBA (or Shuffled LBA). See Figure 4A which describes the sector address before scrambling (e.g. sector address 40 (40,A) & sector address 41 (41,A)) and Figure 4B after the scrambling (e.g. sector address
20 40 (40,B) & sector address 41 (41,B)).

In accordance with certain embodiment of the invention, the scrambling process of the LBA can be done by scrambling only part of the LBA. So, one can scramble e.g. only the sectors which are related to specific disk partition. Yet
25 another example can be inner scrambling of a given number of successive sectors.

In accordance with certain embodiment of the invention the scrambling process will be executed by (but not limited to) any random number generator.
30 Eventually one would prefer random number generators which are certified by the regulations. Due to the fact that the number of sectors scrambled can reach more than hundreds of thousands, the scrambling uniqueness is quite assured (by using verified random generator). Therefore, each SLBA can be considered as unique due to the combinatorial factor.

It is clear that the higher the number of randomly re-addressed sectors, the higher the number of SLBA possibilities. Thus, SLBA entropy increases with the number of re-addressed sectors, where the term entropy, means the

5 Logarithm of the number of indistinguishable equivalent states (which is $N!$, where N stands for the number of sectors scrambled and “!” represents Factorial operation). However, one should be careful here. The indistinguishable states are really equivalent if no data is stored on the re-addressed sectors. Any stored data on the re-addressed sectors, decreases the SLBA entropy. This should be

10 analyzed at two levels. The first level is the logical level, while the second level is the physical level. At the logical level, one considers the following as distinguisher objects (and therefore entropy decreases): first - the very fact that a sector is “populated” by data distinguishes between the sectors with data and those without. Second - the fact that a data with special features is stored (e.g. a

15 file) on more than one sector, would lead to find out the order of the sectors while trying to rebuild the sectors order, by reconstructing the file. At the physical level - if one tracks the disk with a very sensitive electromagnetic tool, it is possible to differentiate between “older” stored data and “newer” data. So these “quasi-dating” capabilities also decrease initial SLBA entropy. As part of

20 certain embodiments of the invention a solution for this entropy decrease will be given in the next sections.

The fact that the sector addresses are scrambled does not prevent one from reading the sectors content one by one; each of them contains at least 512 Bytes.

25 The scrambling just interrupts the operation system from working properly. Even if working irregularly, by activating reading of the sectors one by one, all the stored information will be revealed. If the data is a simple ASCII then it will be immediately revealed. Other known file structures such as Word, XSL, PDF, PS, etc. can be easily revealed too. Therefore, important information can be

30 extracted from the disk content. The same problem exists for the GFM method, if one applies sensitive electromagnetic tools.

In order to keep the entropy as high as (or closer as possible to) the initial entropy, we apply in accordance with a certain embodiment of the invention, the following three steps:

In order to resolve the abovementioned issue of empty sectors, we suggest that “all” the randomly re-addressed sectors be “populated” with stored data. By this one means, even the “empty sectors” will have data (which of course is irrelevant to the disk users). So *ab initio* even an empty disk will be full of stored data. However, in the SLBA it will be indicated that “all” the sectors are actually empty. By the term “all” we mean “most of the sectors, where the higher is the better, up to all of the sectors”.

In order to resolve the abovementioned issues that sectors are readable one by one and that their content can be used for re-ordering of the sectors and the fact that it easy to distinguish between real data and fake data (as suggested above), all the data will be encrypted by a strong (if possible or needed regulated) algebraic encryption method (such as AES-128, AES-256, Camellia, etc.). Within this context the method of GED Algebraic Encryption Method (GAEM) will be further presented in more details later in this document. **Figure 5.** illustrates the combination of sectors addresses scrambling with the implementation of algebraic encryption, where the different gray colored sectors demonstrate that the sectors are differently (and even uniquely as will be presented later in GAEM) encrypted.

In order to resolve the abovementioned issue of the electro-magnetic sensitive analysis as a tool to weaken the encryption, the solution is by using an interference method based on randomly rewriting (i.e. renewing) of the data on the disk (or on the scrambled sectors only). This kind of operation can be executed in system idle time, regularly, or by any other user defined policy. Such an operation disrupts (and can even eliminates) the physical “dating” capabilities as described above. We called such an operation as Electro-Magnetic Dating Interference (EMDI). The EMDI will be further discussed in more details later in this document.

In accordance with a certain embodiment of the invention, the SLBA does not include only the information about the sector addresses scrambling, but also additional information as described in the former section (e.g. information about the encryption). It is also possible that it will be not as part of the SLBA itself,

but will be in other tables, but it is clear that those tables will be linked to each other during system operation. The SLBA with the said extra data (whether it is in one table or in few tables) will be marked as GSLBA (which stands for Generalized SLBA). **Figure 5.A** demonstrates GSLBA. The first column represents the original sectors. Whenever any referring to the original sectors is executed (e.g. reading/writing) the operation is executed on the shuffled sectors in the second column. The information on the encryption keys used and their order for each of the sectors is found in the 3rd column. The keys' number are associated with yet another table of keys. The 4th column represents whether the sector is already occupied or free (where in both cases the sectors are filled with data).

In accordance with a certain embodiment of the invention the GSLBA can be stored encrypted (preferably, but it is not a must), on another media (see **Figure 11**); or hidden and encrypted on the same encrypted media. It is noted that using two media, a higher level of security is conferred, because in order to break the disk encryption one has to have access to both medias (the encrypted disk and the one which contains the GSLBA).

In accordance with a certain embodiment of the invention a similar method of sector addresses scrambling is applied to tapes ((61) **Figure 6.**). However, since the tape geometry is totally different, the implementation method is different too. The way we apply the sector addresses scrambling method to the tape, is by creating a Storage Buffer (SB) ((62) **Figure 6.**). Before the stream of data is written to the tape, it is kept temporarily within the SB. There, the scrambling method is implemented and then it is sent forward scrambled to the tape. As regards the SB size, on one hand it is clear that the larger the SB is, it is better from the entropy point of view; on the other hand the larger is the SB, the longer is the latency of the data transfer to the tape. However, tapes are usually used for backup, rather than online immediately available data storage. So, as regards the SB size, an optimized size is fixed. The issues discussed above relate to weaknesses due to the ability of reading the data sector by sector and that the file type can be recognized, and (where as above described) the solution that should be based on data encryption is relevant to tape sector address scrambling solution as well.

The process of reading from tape is identical to the writing but in reversed order of inverse operations. That is reading the data to the SB, descrambling, decrypting and forwarding it to the user application. (See **Figure 6**).

- 5 In accordance with a certain embodiment of the invention we apply a similar method of sector addresses scrambling to any CD like and/or any DVD like and/or Non-Voltaic Memory like (e.g. Flash Memory) media ((71) **Figure 7**). The way we apply the sector scrambling is by creating a Storage Buffer (SB) ((72) **Figure 7**). Before the stream of data is written on the media, it is kept
- 10 temporarily within the SB. There, the scrambling method is implemented and then it is sent forward scrambled to the media. As regards the SB size, on the one hand it is clear that the larger the SB is, it is better from the entropy point of view; on the other hand the larger the SB, the longer is the latency of the data transfer to the media. The issues discussed above relate to weaknesses due to the
- 15 ability of reading the data sector by sector and that the file type can be recognized, and the above described solutions are relevant here as well. The reading process from media is executed in exactly the same way as the writing but in the reversed order of inverse operations. That is reading the data to the SB, descrambling, decrypting and forward it to the user application. This RSLs
- 20 (and/or GFM) method, especially when applied to any CD like and/or any DVD like media can be very useful as a method for RIGHTS PROTECTION. Since, without the GSLBA (and/or the GCLT in case of GFM), the rights protected data on the media is practically useless.
- 25 In accordance with an aspect of the invention, there is provided Global Encrypted Disk (GED) and GED Algebraic Encryption Method (GAEM). These are terms used for a disk (or a disk partition) which is fully encrypted (e.g. by the GFM and/or RSLs methods as described in the former sections). Moreover, all the sectors and/or GCs of the disk contain data (so no empty sector and/or
- 30 GCs will be found on the disk or the partition); therefore the term "Global" is used. The term Algebraic Encryption Method represents an encryption method which actually is inter-bit- manipulations, while preserving the length of the encrypted data. Known Algebraic Encryption Methods are DES, AES, RSA, Camellia, etc. In accordance with a certain embodiment of the invention,

regarding the prevention of revealing each sector's content both in the GFM method and the RSLs method; an algebraic encryption is applied to each of the sectors content.

- 5 Thus, implementing algebraic encryption involves a key. A use of one key means that all the information will be encrypted with one key based upon one algebraic method. This could be problematic. The weakness of a one key use is that by successful decryption of one sector, all the GED will be revealed. Moreover, one can deliberately store to the disk predetermined information
10 while weakening the encryption power to a great extent.

Thus, as a certain embodiment of the invention, it is preferable to apply more than one algebraic encryption key and even use more than one algebraic encryption method. As regards to how to implement such a method, one
15 considers the following methods:

- A method with one algebraic encryption method, with more than one key, where the keys are randomly chosen from a set of keys. Thus a single sector can be encrypted in an all key combinatorial option. It is preferable (but not necessary) that the set of keys will be predefined and will be as part of the
20 GCLT and/or GSLBA.
e.g. assume a sector of 4096 bit, an algebraic encryption method with 256 bit key and a set of 100 keys. So first, one sees that the sector should be encrypted by 16 keys. The keys are randomly picked from the set of the 100 keys. As one can see the combinatorial factors as regards the number of key orders is
25 enormous.

A method with more than one algebraic encryption method, where each of the algebraic encryption methods (independently) involves one or more keys as described in the former above method. The algebraic encryption methods can be combined in few ways.

- 30 A method where each sector is encrypted by few algebraic encryption methods, where the first n -th bits are encrypted by one method with n -bits key long, and then the rest of the bits within the sector are encrypted in the same way by the other methods.

A method where some of the sectors will be encrypted by one method and

some by other methods, but the methods will be randomly switched between sectors.

A method where each sector is fully encrypted by one method and then encrypted by yet another encryption method. The methods will be randomly
5 chosen from a set of methods.

Figure 5 with all the different gray colored sectors, demonstrates that each of the sectors is differently algebraically encrypted. As one can notice by the embodiment, the whole disk is filled with encrypted data. **Figure 5.A** further
10 demonstrates an example of GSLBA which contains the relevant data for operation.

In accordance with an aspect of the invention, there is provided Electro-Magnetic Dating Interference (EMDI). It is known that Electro-Magnetic
15 Instruments (EMI) used by companies that deal with disk rescuing/recovering, are so sensitive that they enable extraordinary capabilities (e.g. data recovery in cases such as: post disk reformatting, physically deleted data, and even sometimes in cases where physically a sector has been already rewritten by new data).

20 Furthermore, there is a fully described method on how the bits of a flash memory can be directly read from the memory cells, without the need of the logical module usually associated with the non-voltaic memory module.

As regards disks, full scanning of a disk by EMIs can chronologically order, up
25 to a certain level, the content aging. It can be done by Electro-Magnetic Intensity Analysis. Fading, intensity decaying, smearing and other physical features can be used as data aging indicators. Therefore, they can greatly reduce the power of the sector scrambling algorithm (RSLS). Even the geometrical format modification (GFM), can be detected by the EMIs. Although this might
30 be a long and complex process, it is possible!

Therefore, further interference actions are desired. The straightforward interference action (as already noted) which is also a certain embodiment of the invention is to randomly select sectors (or GCs in the case of GFM) and

refresh/renew the sector content. It should be mentioned again that we apply also the GAEM (see above) method as part of “keeping the entropy as high as it was in the beginning”. Now, given that all the sectors (and/or GCs) on the disk/partition contain real encrypted data [even the “not used yet” sectors (and/or GCs)]; it follows that the refreshing/renewing of sectors (and/or GCs) content, includes all the sectors (and/or GCs) of the disk/partition. This refreshing/renewing operation misleads the process of chronological ordering by EMIs. So, the result outcome of EMIs attempts for “chronological dating” will fail.

10

In accordance with some aspects of the invention, there are provided some optimization methods and operative aspects. The methods described so far for data storage protection and encryption can be summarized as three levels (or layers) of operation (see **(81), (82), (83) - Figure 8.** – respectively). All three levels are executed in order to preserve the entropy at a satisfactorily high level, starting from the beginning of the disk usage. Keeping the entropy high all the way cannot be done without this affecting performance. In this section some of the operations that are applied in order to improve the disk (and/or the tape) performance, while reducing the performance cost to an acceptable value are described. Although some optimization is performed regularly by the storage operating system, due to the modifications to the disk (and/or tape) operation that were implemented, some of the storage system operations have to be modified and even blocked, e.g. there is no point in performing a Defrag operation on a scrambled disk.

25

First Initialization – **(Figure 11.A)** In accordance with a certain embodiment of the invention, all information with regard to the storage (a disk and/or a tape) **((111) Figure 11.A)** protection and encryption is prepared before the storage is available for live operation **((112), (113) - Figure 11.A)**. It is clear that if a new hot swap disk is inserted to the storage, those initialized operations will be performed without interrupting the live storage system. Only upon completion of the initialized preparation the disk will be ready for the whole encrypted protected storage.

30

Furthermore, within the initialization process as regards the sector addresses

scrambling (as regards the RSLS method), we apply in accordance with a certain embodiment of the invention the method of Neighboring -Tracks Sector Scrambling (NTSS). If one had scrambled all the sectors on a given disk, the disk electromagnetic arm-header that reads and writes the information to the disk would have moved upon the whole disk, while slowing the disk operation tremendously. The NTSS method means that first one should fix the required encryption power. Then, based on that, one finds how many sectors should be scrambled at once to reach the desired encryption power. Therefore the sectors on the disk will be divided to groups, where each group contains at least the number of sectors desired (but not much more than this number), and all of the sectors in each of the groups are either sequential (before the scrambling) therefore found upon vicinity tracks, or taking an integral number of neighboring tracks with all the sectors found on them. This means that the scrambling sectors in each of the groups is controlled, and the disk read/write header movement will be physically in limited disk zone in average. **Figure 9** demonstrates the NTSS idea. The colored tracks represent the tracks from which the sectors were randomly picked and scrambled. So e.g. **(91) Figure 9** represents the two outer tracks within the sectors address were scrambled, while **(92) Figure 9** represents the same for the two inner tracks.

20

Operation – In accordance with a certain embodiment of the invention, all the tables that contain all the information on the storage encryption and protection (e.g. the GCLT and/or GSLBA), are cached to very fast memory modules ((117) **Figure 11.B**), and will always be accessed to the encryption/decryption operation module ((115) **Figure 11.B**). Furthermore, in accordance with a certain embodiment of the invention, dedicated hardware module which has the desired features boost the encryption/decryption operation to the storage. The necessary modules (but not limited to these only) (see **Figure 10.**) are:

Data communication module ((101) **Figure 10.**) that can handle the data content to be written to (or to be read from) the storage (e.g. PCI, PCI-X, PCI-E, SCSI, ISCSI, ATA, PATA, SATA, FC, etc.), and that can communicate with the modules where the GCLT and/or GSLBA) are found.

Fast cache memory ((102) **Figure 10.**) for embedding the encryption/decryption tables, and for any other use of the CPU.

CPU (and/or other processing modules such a DSP) ((103) **Figure 10.**) for fast executing the algebraic encryption and all the other live operations are also needed for the first initialization of the “to be protected/encrypted” storage as described above. The CPU is also needed for the “idle time operation” as
5 described with regard to randomly rewriting/renewing of sectors (and/or GCs).

Bootable Non-Voltaic memory ((104) **Figure 10.**), where the protecting and the encrypting/decrypting software is stored.

Read/Write optimization - It is known that optimization methods are
10 implemented by different operating systems, in order to maintain high performance of the storage. Among the optimized methods, one finds:

A preference for sequential sectors writing in order to reduce the arm moves to a minimum required. This of course also improves the read operation. This is implemented by sectors/blocks (to be stored to the disk) accumulation for a very
15 short time, organizing them in increasing or decreasing series of their logical addresses , before storing them on the disk.

Defrag operation causes the data to be stored to the disk as continuously as possible, where files will be continuously/sequentially written to the disk. It also differs between temporary data that are kept on different locations since it is
20 assumed that it will soon be deleted.

Automatic storage operation, that automatically transfers old data that is not accessed for a long time to tapes archiving, where the accessed time is much longer. Thus new disk empty places are available for new data to be stored.

Some of the raids configurations are designed to improve the disks
25 read/write performance.

The abovementioned optimization operations are just examples, and other methods are possible. However, all of them were developed for normal storage operation, i.e. without the storage protection/encryption operations. Therefore,
30 some of the optimization methods cannot be implemented at all (and even should be forbidden, e.g. the defrag operation), some can be implemented with some modifications, and some new methods have been specially developed. Therefore, in accordance with a certain embodiment of the invention, optimization methods compatible with protected/encrypted storage

methodology, are presented. For disk reading/writing optimizations there are a few aspects to be considered. Firstly, the reading/writing execution time that one prefers to be the shortest as possible, should be considered. Secondly, a minimal number of moves for the reading/writing arm should be considered.

- 5 Implementing this aspect ensures longer disk life, and minimum of noise during operation. Further, it should be noticed that any optimization should be strongly related to the disk physical features. For example – the disk RPM (Rounds Per Minutes), the arm speed, the number of plates/heads, the disk density, etc. all should be considered and be part of the optimization algorithm as implemented
- 10 by the low level disk software driver installed to the OS. Furthermore, different GFM's methods will lead to different optimization methods.

- It is of extreme importance to apply an optimization method on the performance, especially while implementing GFM and/or RSL's methods. In lab experiments
- 15 of RSL's implementation starting with no optimization and with totally scrambled 20GB disk about 90% extra latency was obtained, as compared to normal work of reading/writing from a disk (not speaking of the loud noise heard due to the arm moves). After applying an optimization algorithm the extra latency (as well as the noise) was reduced to about 10%. Further implementing
- 20 the NTSS method above discussed, can reduce the extra latency to a value of less than 1%.

- The present invention has been described with a certain degree of particularity, but those versed in the art will readily appreciate that various alterations and
- 25 modifications may be carried out, without departing from the scope of the following Claims:

CLAIMS:

1. A method for formatting of a disk that includes at least one track and the said track is capable of being divided into at least two geometrical cells
5 (GCs), comprising:
 - (a) dividing at least one of the tracks into at least two neighboring geometrical cells (GCs); at least one of said GCs is bordered by at least different internal radius and/or different outer radius compared to the respective radius of a neighboring cell of said track;
 - 10 (b) storing Geometrical Cell Location Table (GCLT), for accessing data in said cells, said accessing include inserting or updating or deleting data.
2. The method according to Claim 1, wherein said GCLT is stored in different medium than said disk.
- 15 3. The method according to Claim 1, wherein said GCLT is encrypted.
4. A method for formatting a disk that includes at least one track and the said track is capable of being divided into at least two geometrical cells (GCs), comprising:
 - 20 (a) dividing at least one of the tracks into at least two neighboring geometrical cells (GCs); at least one of said GCs defining a different angular size compared to the angular size of a neighboring GC of said track;
 - (b) storing Geometrical Cell Location Table (GCLT), for accessing data in said cells, said accessing includes inserting, updating or deleting
25 data.
5. The method according to Claim 4, wherein said GCLT is stored in a different medium than said disk.
6. The method according to Claim 4, wherein said GCLT is encrypted.
7. A method for formatting a disk that includes at least one track, said track is
30 capable of being divided into at least two geometrical cells (GCs), comprising:
 - (a) dividing at least one of the tracks into geometrical cells (GCs); at least one of said GCs is bordered by at least different internal radius and/or different outer radius compared to the respective radius of a neighboring cell of said track;

(b) at least one of said GCs defining a different angular size compared to the angular size of a neighboring GC of said track;

(c) storing Geometrical Cell Location Table (GCLT), for accessing data in said cells, said accessing include inserting or updating or deleting data.

5

8. The method according to Claim 7, wherein the GC stipulated in said (a) and (b) is one GC and the neighboring GC stipulated in said (b) is a second GC.

9. The method according to Claim 7, wherein said GCLT is stored in a different medium than said disk.

10 10. A method for encrypting data stored in a disk, the disk includes at least two geometrical cells (GCs), comprising:

(a) the geometrical cells (GCs) constituting sectors and constructing Logical Block Address (LBA), for facilitating access of sector of interest;

(b) reshuffling at least two of said sectors addresses, giving rise to

15

Shuffled LBA (SLBA), such that a sector from among said at least one sectors having a sector address in said LBA will have a different address in said SLBA; said SLBA facilitating access of sector of interest.

11. The method according to Claim 10, wherein said SLBA is stored in a different medium than said disk.

20 12. The method according to Claim 10, wherein said SLBA is encrypted.

13. The method according to Claim 1, further comprising shuffling said Geometrical Cell Location Table, giving rise to Shuffled Geometrical Cell Location Table (SGCLT), such that a GC from among said at least one GCs having a GC address in said GCLT will have a different address in said SGCLT; said GCLT facilitating access of cell of interest.

25

14. The method according to Claim 4, further comprising shuffling said Geometrical Cell Location Table, giving rise to Shuffled Geometrical Cell Location Table (SGCLT), such that a GC from among said at least one GCs having a GC address in said GCLT will have a different address in said SGCLT; said GCLT facilitating access of cell of interest.

30

15. The method according to Claim 7, further comprising shuffling said Geometrical Cell Location Table, giving rise to Shuffled Geometrical Cell Location Table (SGCLT), such that a GC from among said at least one GCs

having a GC address in said GCLT will have a different address in said SGCLT; said GCLT facilitating access of cell of interest.

16. The method according to Claim 10, further comprising encrypting data in at least one of said sectors.

5 17. The method according to any one of Claim 1, further comprising encrypting data in at least one of said cells.

18. The method according to any one of Claim 4, further comprising encrypting data in at least one of said cells.

10 19. The method according to any one of Claim 7, further comprising encrypting data in at least one of said cells.

20. The method according to Claim 17, wherein said encrypting includes encrypting data in at least one cell utilizing at least two separate keys.

15 21. The method according to any one of Claim 17, wherein said encrypting includes encrypting data in at least two cells utilizing different encryption algorithms.

22. The method according to Claim 18, wherein said encrypting includes encrypting data in at least one cell utilizing at least two separate keys.

20 23. The method according to any one of Claim 18, wherein said encrypting includes encrypting data in at least two cells utilizing different encryption algorithms.

24. The method according to Claim 19, wherein said encrypting includes encrypting data in at least one cell utilizing at least two separate keys.

25 25. The method according to any one of Claim 19, wherein said encrypting includes encrypting data in at least two cells utilizing different encryption algorithms.

26. A method for encrypting data stored in a disk, the disk includes at least one track and track is capable of being divided into at least two geometrical cells (GCs), comprising:

- 30 (a) dividing the disk into tracks and geometrical cells (GCs) constituting sectors and constructing Logical Block Address (LBA), for facilitating access of sector of interest;
- (b) encrypting data in at least one of said sectors using at least two encryption keys.

27. A method for encrypting data stored in a disk, the disk includes at least one track and the said track is capable of being divided into at least two geometrical cells (GCs), comprising:
- (a) dividing the disk into tracks and geometrical cells (GCs) constituting sectors and constructing Logical Block Address (LBA), for facilitating access of sector of interest;
 - (b) encrypting data in at least two of said sectors using at least two encryption algorithms.
28. The method according to Claim 27, further comprising encryption at least one of said sectors using at least two keys.
29. The method according to Claim 1, further comprising, renewing content of GCs, according to a predefined criterion.
30. The method according to Claims 4, further comprising, renewing content of GCs, according to a predefined criterion.
31. The method according to Claims 7, further comprising, renewing content of GCs, according to a predefined criterion.
32. The method according to Claims 10, further comprising, renewing content of GCs, according to a predefined criterion.
33. The method according to Claim 26, further comprising renewing content of GCs, according to a predefined criterion.
34. The method according to Claim 27, further comprising renewing content of GCs, according to a predefined criterion.
35. A method for encrypting data stored in a disk, the disk includes at least one track and the said track is capable of being divided into at least two geometrical cells (GCs), comprising:
- (a) dividing the disk into tracks and geometrical cells (GCs) constituting sectors and constructing Logical Block Address (LBA) for facilitating access of sector of interest;
 - (b) storing data in said sectors;
 - (c) renewing content of selected sectors according to a predefined criterion, thereby hindering the possibility of tracing update of data based on aging.
36. The method according to Claims 13, wherein said SGCLT is stored in a different medium than said disk.

37. The method according to Claims 13, wherein said SGCLT is encrypted.
38. A method of encrypting data stored to a tape comprising:
- (a.) buffering at least two sectors before storing them to the tape.
 - (b.) shuffling the said buffered sector-addresses found in the buffer.
 - 5 (c.) storing the information on the said shuffling to a Tape Sectors Shuffling Table (TSST).
 - (d.) storing the shuffled sectors in their new order to the tape.
39. The method according to Claim 38 wherein said TSST is stored in different medium than said tape.
- 10 40. The method according to Claim 38 wherein said TSST is encrypted.
41. The method according to Claim 38 wherein said shuffled sectors their content is encrypted.

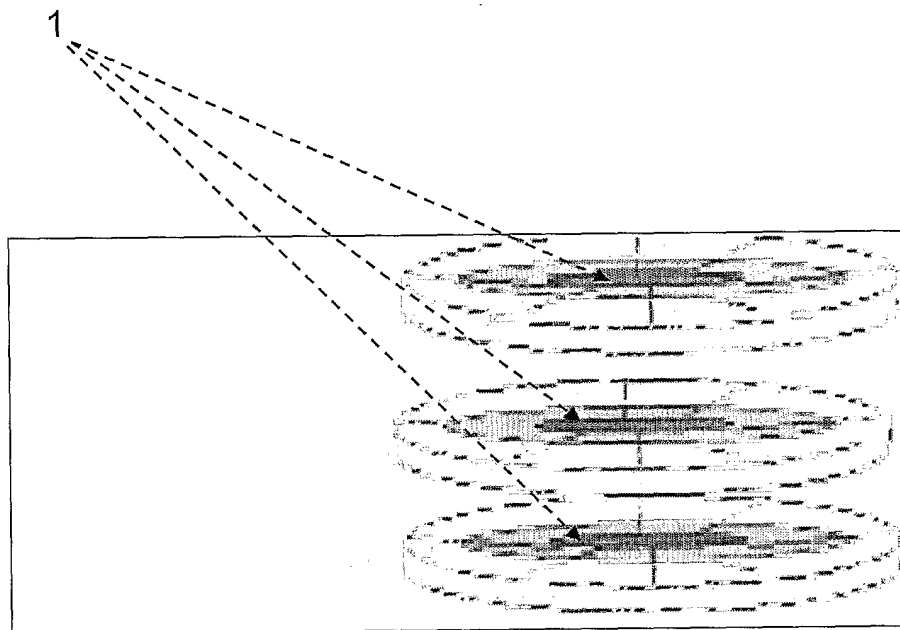


Figure 1A

2/17

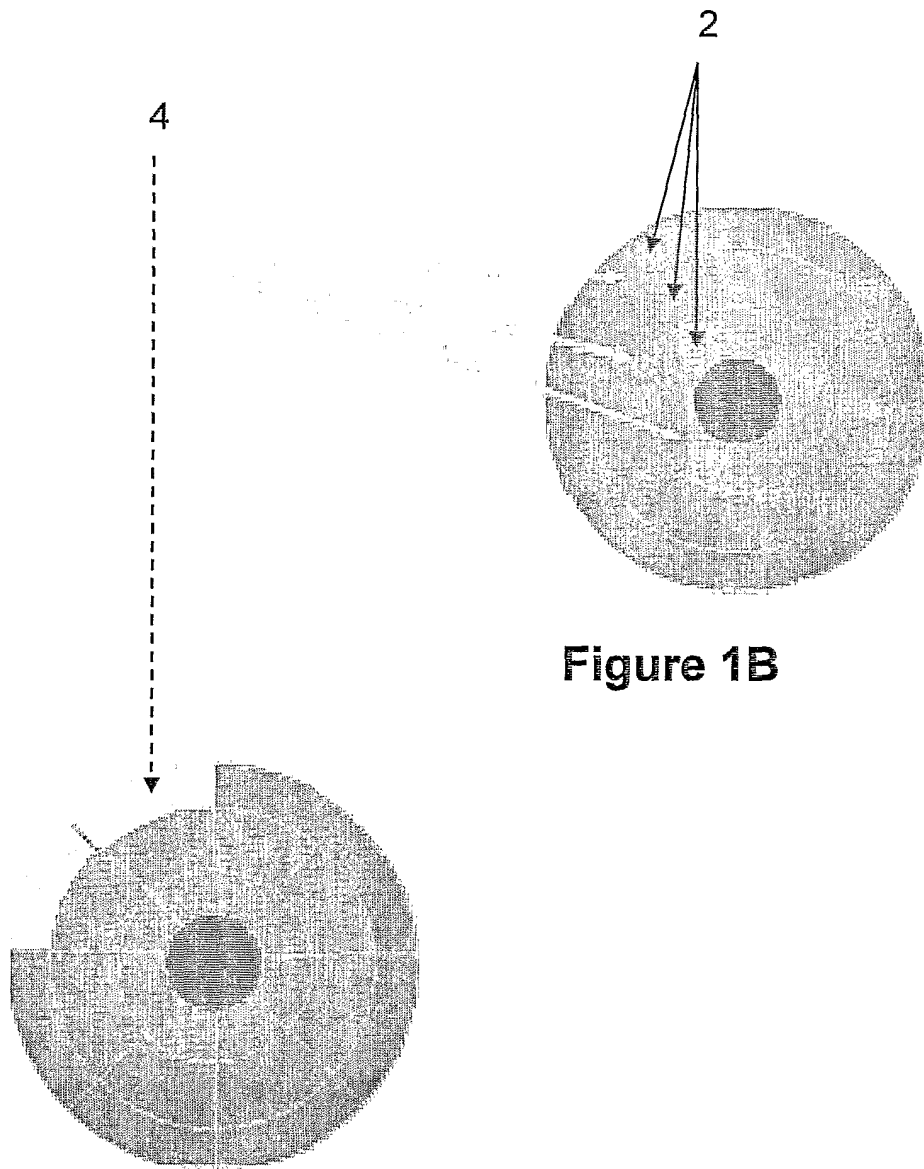


Figure 1B

Figure 1D

3/17

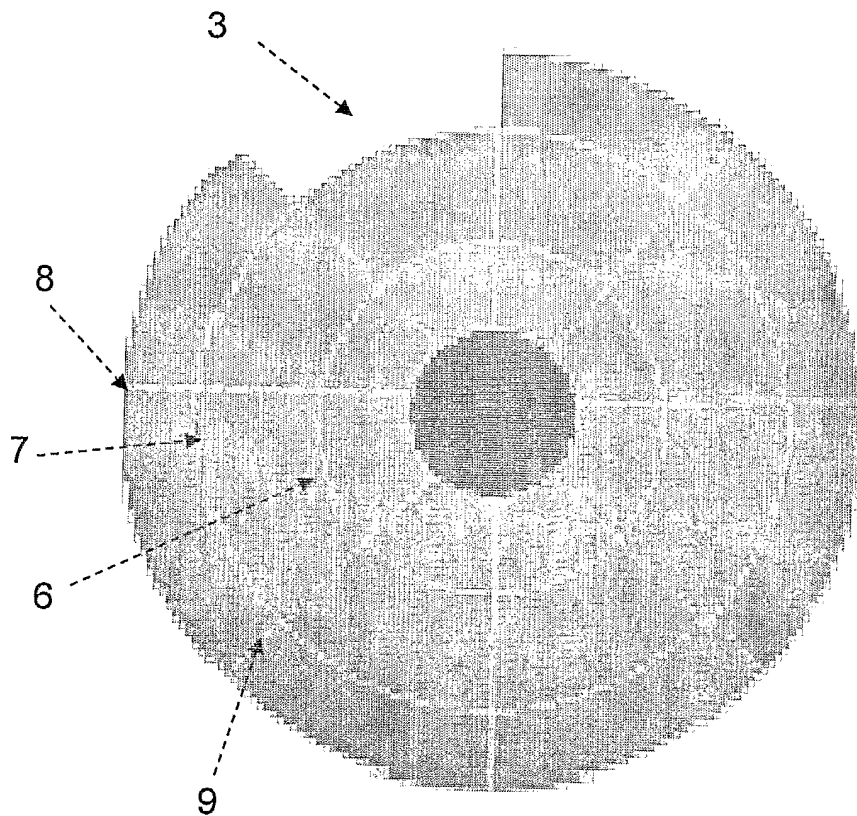


Figure 1C

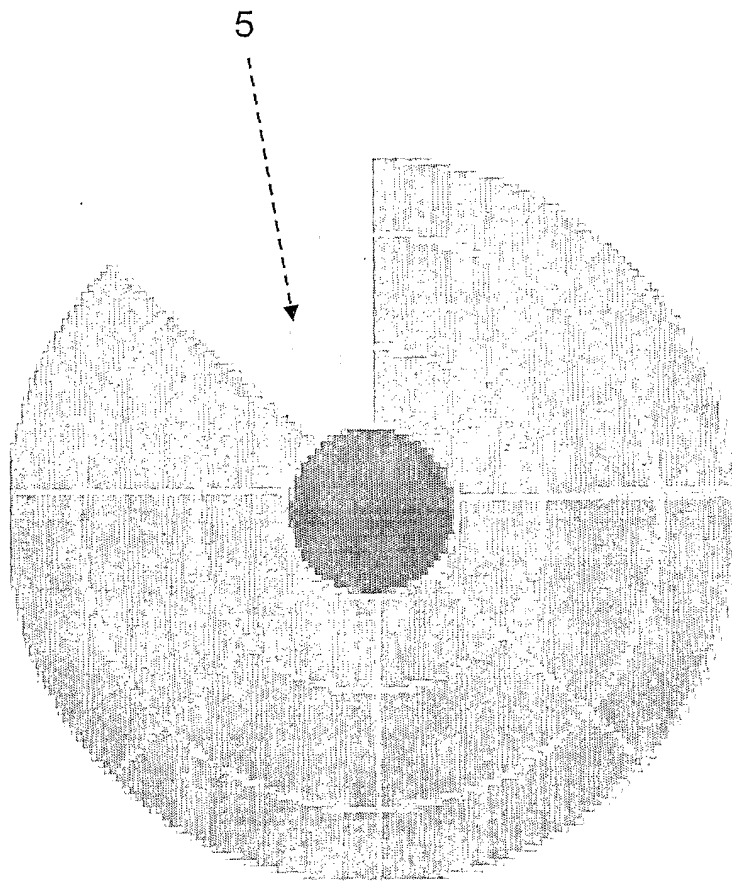


Figure 1E

5/17

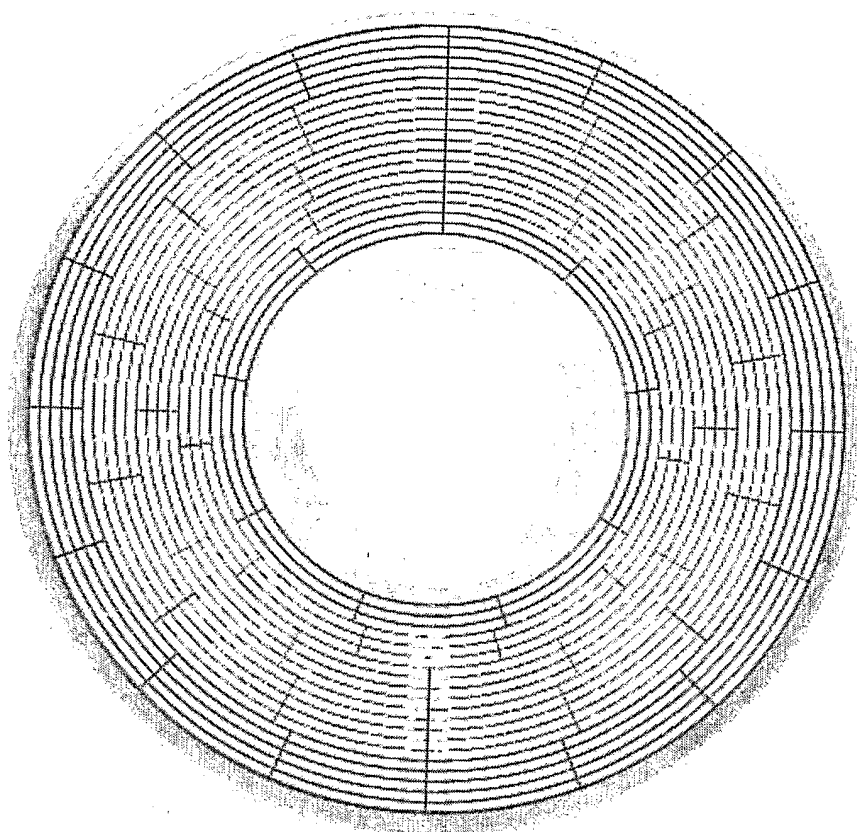


Figure 1F

6/17

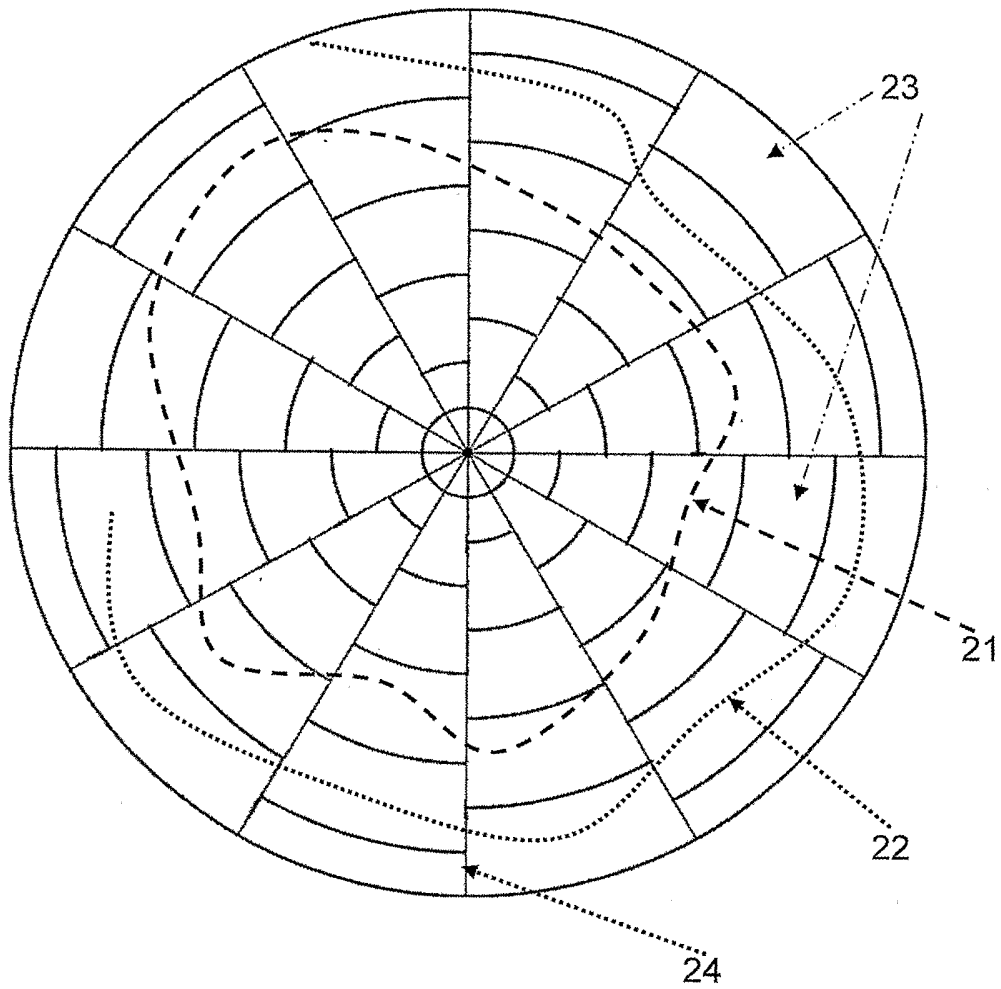


Figure 2

0,3	0,4
0	0
0	1
1	0
0	0
1	1
1	1
1	8,4
0	0
0	1
1	0
0	

Figure 2A

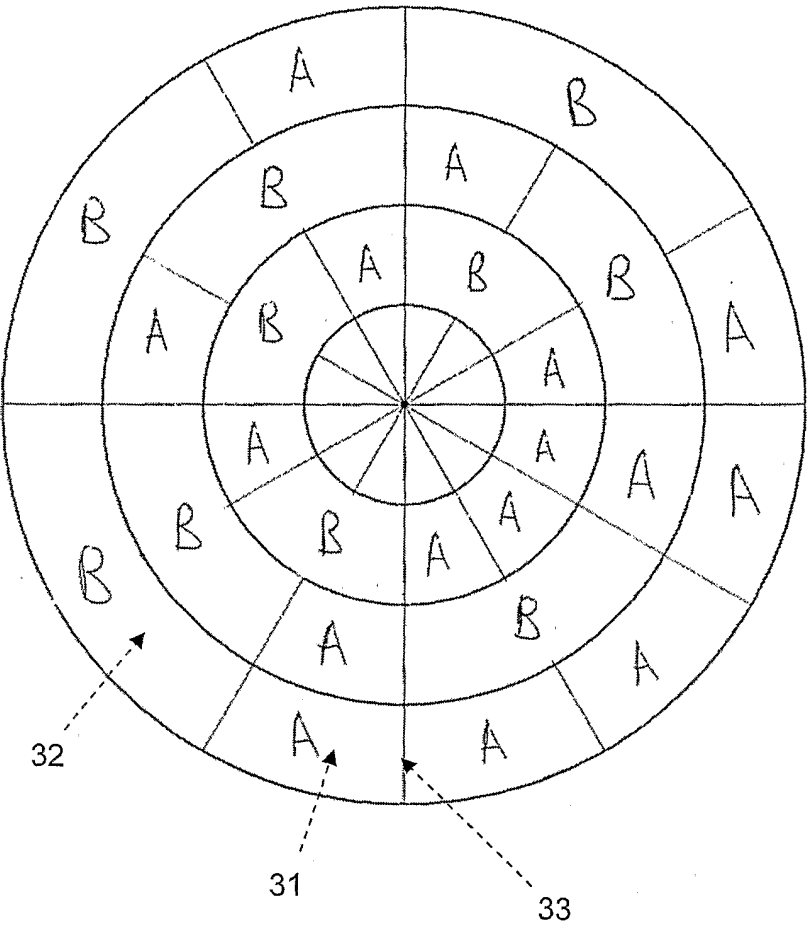
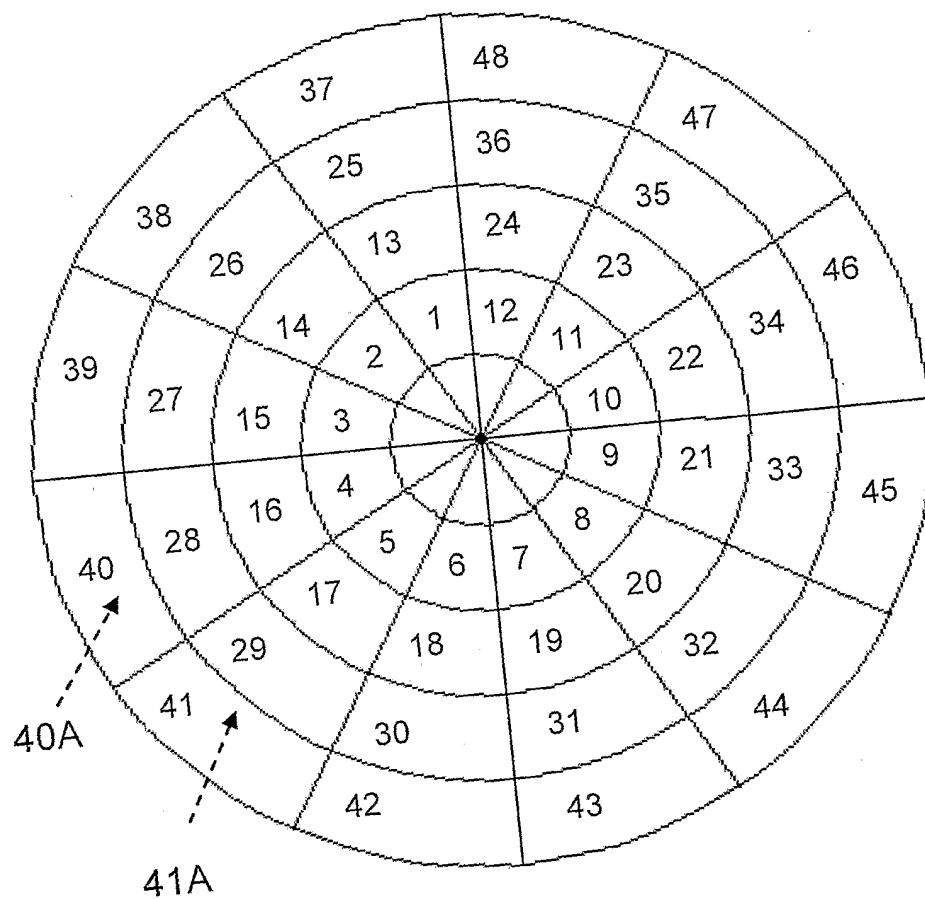
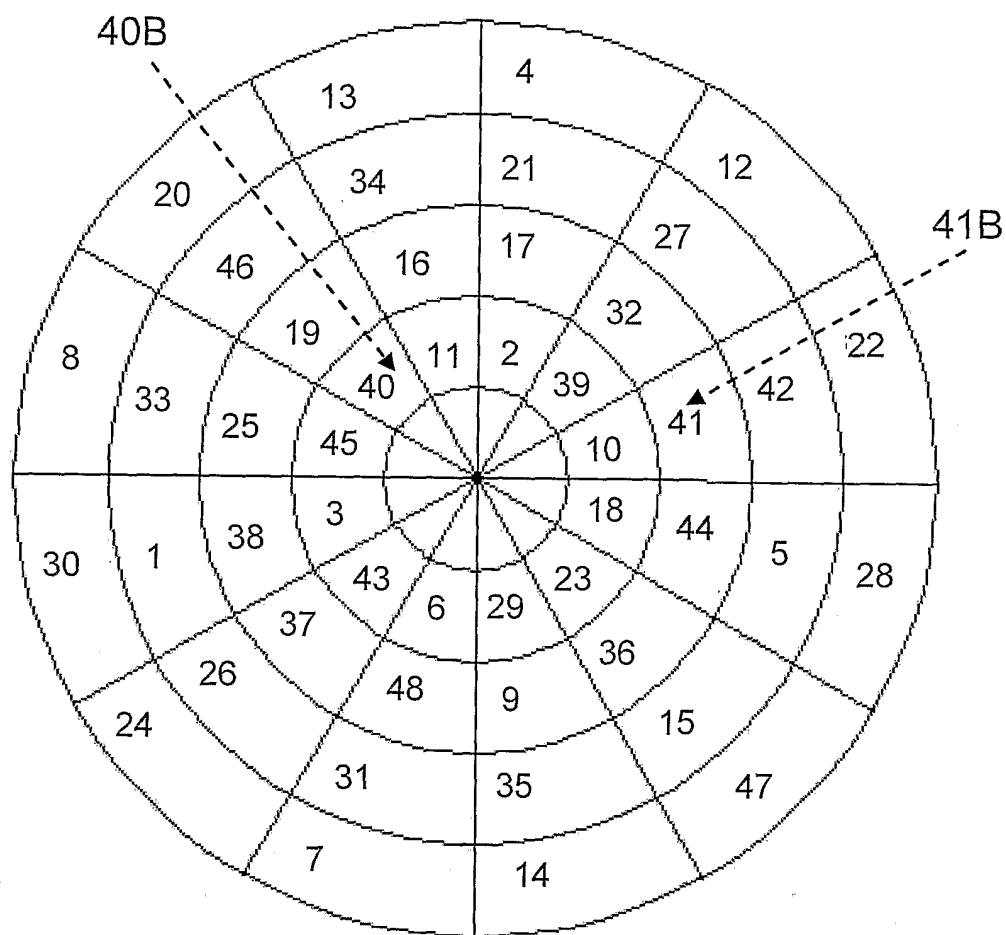


Figure 3

0,1	0,2	0,3
0	1	0
0	0	0
0	1	0
0	0	0
1	1	1
0	0	0
1	1	1
0	0	1
1		0

Figure 3A

**Figure 4A**

**Figure 4B**

12/17

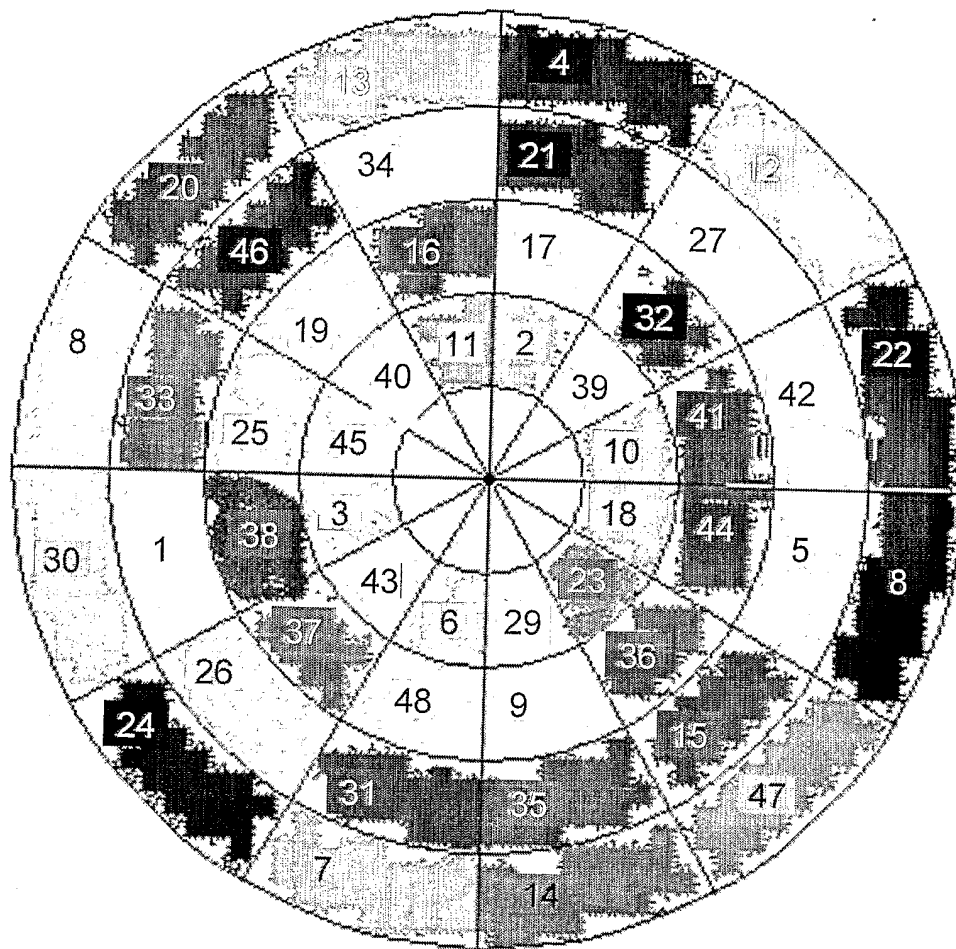


Figure 5A

LBA	SLBA	Encryption Keys	Status
1	2	2,19	0
2	1	27,3	0
3	6	15,1	1
4	4	6,21	1
5	3	3,3	0
6	5	14,13	0

Figure 5B

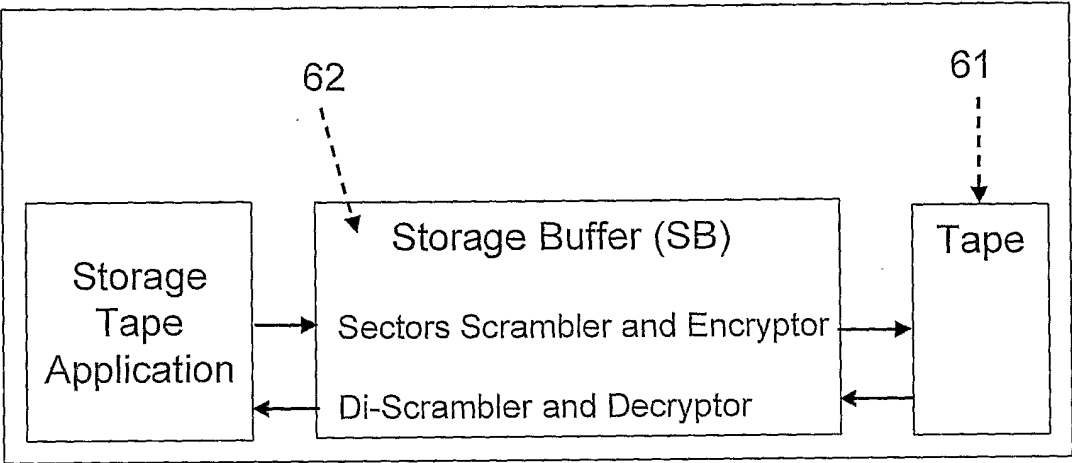


Figure 6

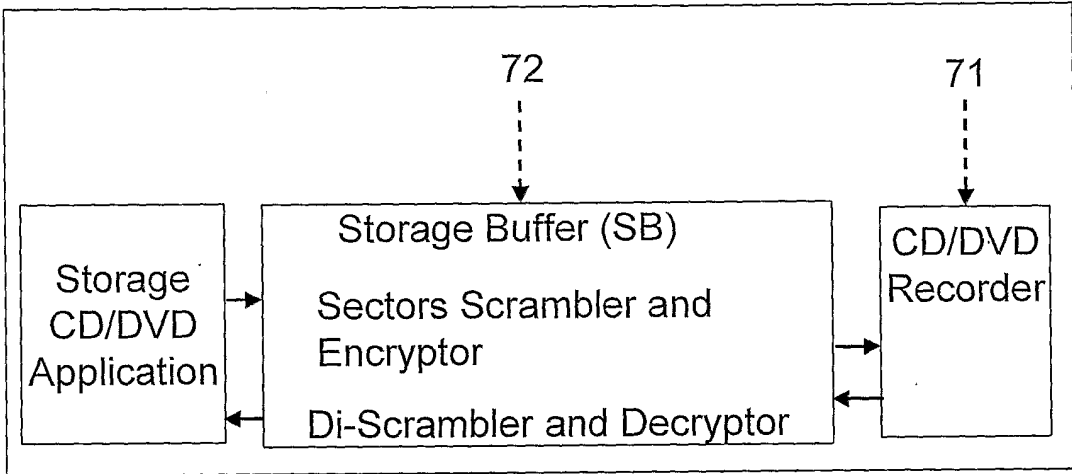


Figure 7

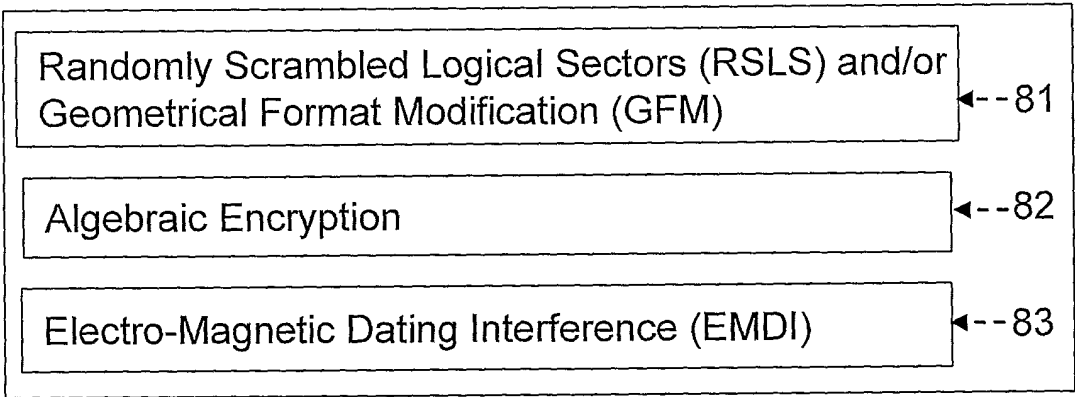
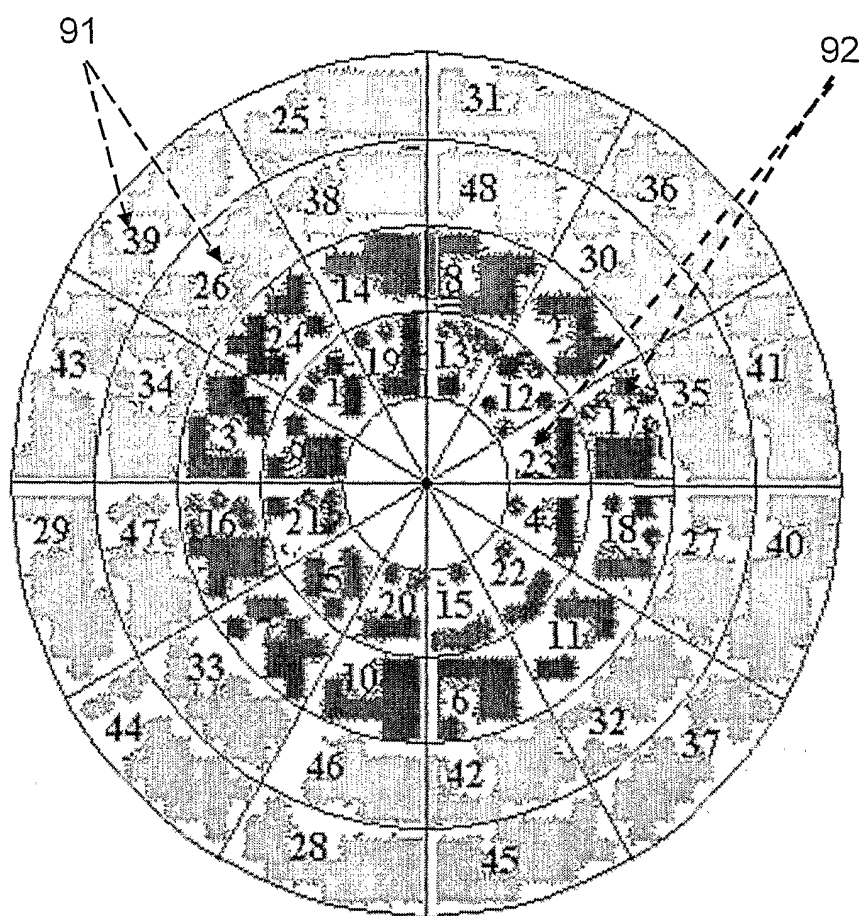


Figure 8

**Figure 9**

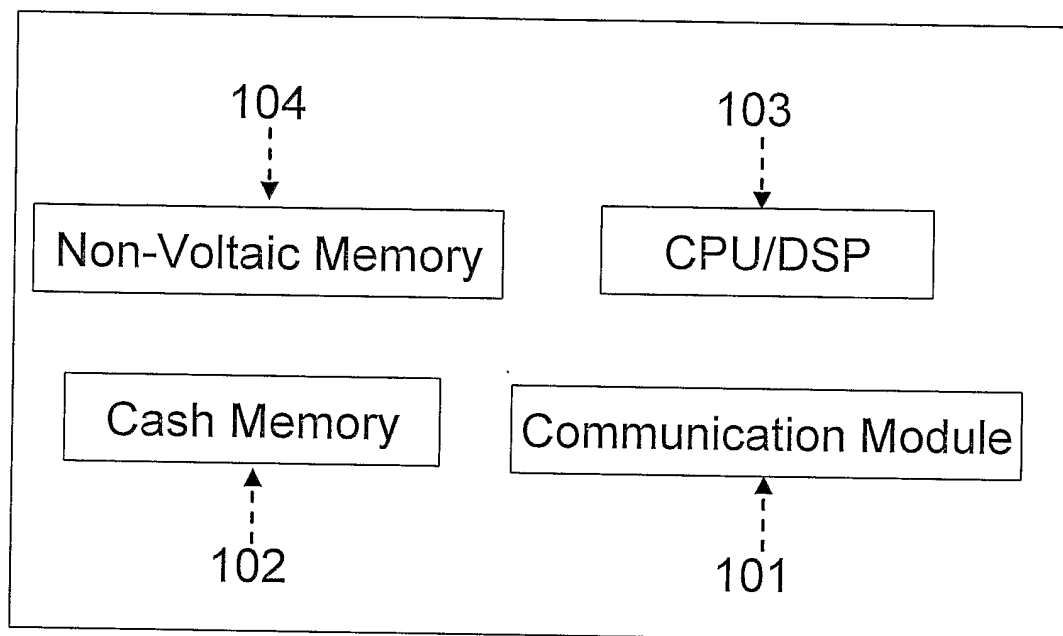
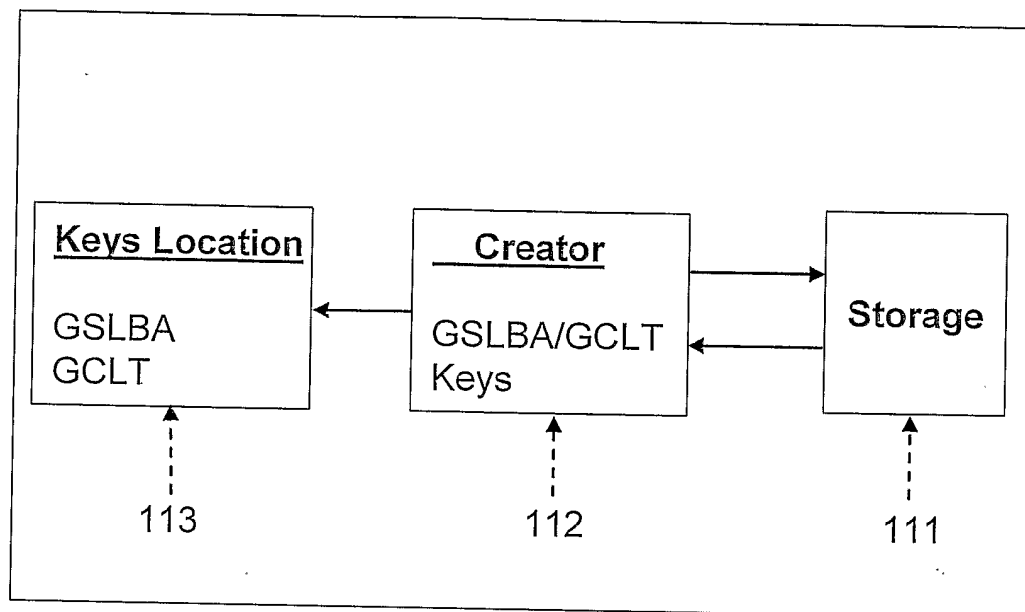
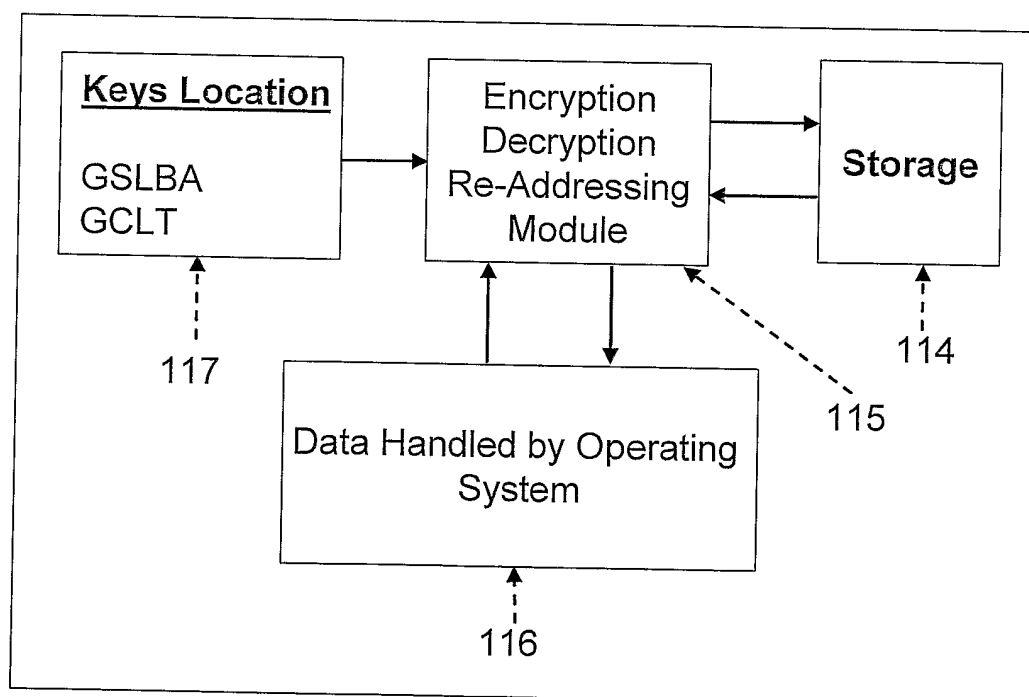


Figure 10

**Figure 11A****Figure 11B**