



- (51) **International Patent Classification:**
G06F 21/02 (2006.01) *G06F 21/24* (2006.01)
- (21) **International Application Number:**
PCT/EP2010/007657
- (22) **International Filing Date:**
16 December 2010 (16.12.2010)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant** (for all designated States except US): **NEC EUROPE LTD.** [DE/DE]; Kurfuersten-Anlage 36, 69115 Heidelberg (DE).
- (72) **Inventors; and**
- (75) **Inventors/Applicants** (for US only): **HESSLER, Alban** [CH/DE]; Kettengasse 2, 69117 Heidelberg (DE). **UGUS, Osman** [TR/DE]; Elfendgrund 8a, 64367 Mühlthal (DE).
- (74) **Agent:** **ULLRICH & NAUMANN**; Schneidmühlstrasse 21, 69115 Heidelberg (DE).
- (81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

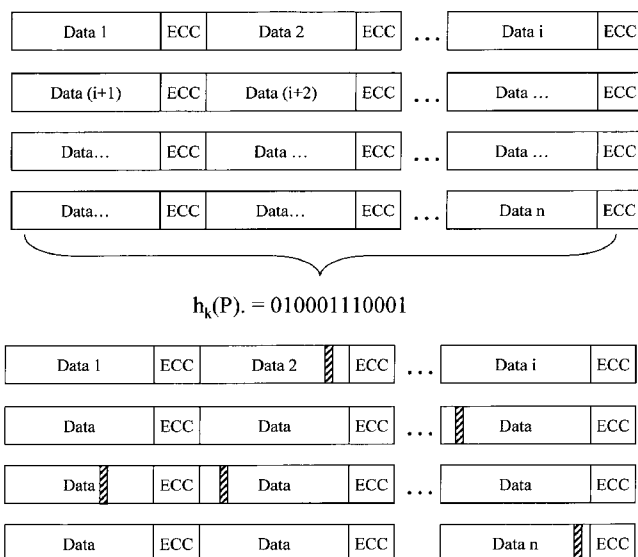
AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, 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, 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:** METHOD FOR PROVIDING A TAMPERING DETECTION FUNCTION FOR A STORAGE MEDIUM, STORAGE MEDIUM AND DEVICE FOR WRITING DIGITAL DATA TO A STORAGE MEDIUM



(57) **Abstract:** A method for providing a tampering detection function for a storage medium, in particular a portable digital storage device, wherein digital data stored on said storage medium is divided into a plurality of data blocks with each data block having associated an error correction code, is characterized in the steps of calculating a hash code by applying a hash function to the stored data, performing a mapping of the bits of said hash code to said data blocks, and based on the mapping, embedding said hash code into said data blocks in form of bit errors. Furthermore, a device for writing digital data to a storage medium is disclosed.

Fig. 4

METHOD FOR PROVIDING A TAMPERING DETECTION FUNCTION FOR A STORAGE MEDIUM, STORAGE MEDIUM AND DEVICE FOR WRITING DIGITAL DATA TO A STORAGE MEDIUM

The present invention relates to method for providing a tampering detection function for a storage medium, in particular a portable digital storage device, wherein digital data stored on said storage medium is divided into a plurality of data blocks with each data block having associated an error correction code.

The present invention also relates to a storage medium, in particular a portable digital storage device, being equipped with a tampering detection function, wherein digital data stored on said storage medium is divided into a plurality of data blocks with each data block having associated an error correction code.

Furthermore, the present invention relates to a device for writing digital data to a storage medium, comprising an error correction component such that said digital data to be stored on said storage medium is divided into a plurality of data blocks with each data block having associated an error correction code.

In the digital world, copying or modifying large amount of data is extremely simple and cost effective. Hence, a storage medium, especially a portable one like, e.g., CD-ROM, memory stick, flash memory, etc., can easily contain information about thousands or millions of items, which can be critical for the owner. Thus, loss or modifications of the data of a storage medium can have important consequences for businesses, organizations, or individuals. Making sure that the original data was not altered during its transport is not trivial, and is a key requirement in many areas, for example in forensics.

In prior art it is known to protect storage media by means of a built-in error correction mechanism to ensure that the data remain readable after minor damage of the media, or just to prevent the negative effect of aging. Such mechanisms are known as Error Correcting Codes (ECC), which are codes that are appended to the original data, and which permit to correct up to a certain amount of errors (see for reference ISO/IEC 10149 / ECMA-130). Typically, the longer the code

overhead is, the more errors can be recovered. A variant of ECCs are EDCs (Error Detection Codes), to “detect” errors. They can often be built on the same code, and are used most of the time in combination with ECC to offer robust data storage.

U.S. patent 6,597,648 describes an optical disk having a data duplication prevention function. To enable detection of unauthorized copying of the optical disk security information is embedded into the ECC. The main idea is to inject errors on purpose at some specific places. A copy machine generally automatically detects those errors and removes them. Hence only the original copy has errors, and a verifier can distinguish an original disk from a copy. However, the proposed scheme is ineffective in ensuring integrity of the stored data. In particular, when reading a copy of the original optical disk, which was fabricated according to the method of US 6,597,648 B1, it is impossible to detect whether the stored data have been modified, i.e. any tampering with the originally stored data remains unrevealed.

It is therefore an object of the present invention to improve and further develop a method, a storage medium and a device for writing digital data to a storage medium of the initially described type in such a way that any modification and/or tampering made on the data stored on the medium can be reliably detected to ensure that the original data was not altered, e.g., during its transport.

In accordance with the invention, the aforementioned object is accomplished by a method comprising the features of claim 1. According to this claim such a method is characterized in the steps of calculating a hash code by applying a hash function to the stored data, performing a mapping of the bits of said hash code to said data blocks, and based on the mapping, embedding said hash code into said data blocks in form of bit errors.

Furthermore, the above mentioned object is accomplished by a storage medium comprising the features of claim 12. According to this claim such a storage medium is characterized in that the data includes a hash code that is embedded into said data blocks in form of bit errors, wherein the hash code has been

calculated by applying a hash function to the stored data, and by performing a mapping of the bits of said hash code to said data blocks.

Finally, the above mentioned object is accomplished by a device for writing digital data to a storage medium comprising the features of claim 13. According to this claim such a device is characterized in that it comprises a generator component being configured to calculate a hash code by applying a hash function to the data to be stored, and to perform a mapping of the bits of said hash code to said data blocks, and injection means being configured to embed said hash code, based on the mapping, into said data blocks in form of bit errors.

According to the invention it has first been recognized that the ECC verification can be considered as a hidden communication channel, which can be employed to embed in the data storage a hash code of the original data. Therefore, the present invention presents a data modification detection mechanism for storage media with built-in error correction mechanism which conceals itself in the error correction layer of the medium. The hash code of data being protected is embedded data as if they are natural errors. Any modification on data will produce different hash value and therefore different error distribution in the data. Hence, an unknowledgeable attacker will modify the data without knowing he will raise a flag if the data is verified at a later point in time. Even if the attacker knows about the protection, he is not able to modify the data without the original device having accomplished the original data storage.

The present invention presents a solution that embeds integrity and/or authentication of data into the error correcting code of a storage medium, while hiding the presence of this mechanism to a malicious party. With the presence of the systematically injected bit errors, which can be regarded as security bits, on the storage medium, a malicious person who is not in possession of the generator device cannot modify the data without being detected. If an attempt is made to modify the data, a verifier device will detect it, and the verifier device can warn the user of the media storage that the data has been modified since it left the generator device.

As a further advantage the present invention produces almost no overhead, and only slightly increases the risk of data loss due to the increasing number of bit errors. Hence, the method according to the invention slightly limits the efficiency of the ECC by injecting errors. The integrity scheme may therefore itself contain error correcting codes in order to cope with "natural" errors.

Moreover, the present invention additionally provides a weak protection against copying of storage media. Furthermore, it is backwards compatible to existing reading devices, since they will still be able to read the data of storage media that employ the method according to the present invention, although they won't be able to verify the data. However, the invention provides only poor support for securing stream of data. For example, if one would like to add or modify one block of data, this would affect completely the hash, and thus requiring the modifications of most of the security bits. Hence, the invention is mostly practical for data that is written or modified very occasionally, which is the case of the use cases presented.

According to a preferred embodiment it may be provided that for each bit of the hash code that is equal to '1' at least one bit error is injected into the data block that (according to the mapping) corresponds to the respective bit of the hash code. In the same way, in case a bit of the hash code is equal to '0', the corresponding data block remains unchanged, i.e. no bit error is injected into said data block. The injection of more than one bit error per data block embeds a higher security into the data. However, there is of course a trade off between the number of bit errors, the number of bit errors that the ECC can correct, and the natural rate of bit errors of the media. As will be apparent to a skilled person, a reverse embodiment, in which bit errors are injected in case the respective bit of the hash code is equal to '0' and in which no bit error is injected in case the respective bit of the hash code is equal to '1', is also possible.

In the simplest form it may be provided that each bit of the hash code is mapped to exactly one data block. In this approach the number of data blocks, e.g. 256, has to be adapted to the number of bits of the hash code. Alternatively, in particular with respect to an enhanced flexibility, each bit of the hash code may be mapped to a predefined number of data blocks. This would be one possibility to take into

consideration that each kind of storage medium and each kind of digital data stored on the medium comes along with a typical rate and distribution of bit errors. For instance, a spreading of the bit errors in a more natural way can be achieved by considering only every i 'th data block for the mapping, with i being an integer larger or equal 2. For instance, it may be provided that the bits of the hash code are mapped only to every third data block.

According to a preferred embodiment it may be provided that the positions of the embedded bit errors within a data block are chosen such that the bit error positions transmit information about the hash code. This means that the positions of the embedded bit errors carry additional information with respect to the hash code. For instance, by choosing the positions of embedded bit errors appropriately, a spreading of the bit errors in a more natural way can be achieved. for example, by embedding an error to the i 'th bit of the data block, where i is the decimal representation of a block (i.e. an aggregated number) of bits of the hash code.

By combining the above mentioned different approaches of mapping and bit error position selection in a suitable way it is even possible to adapt to rather specific patterns of natural bit errors, which on some media occur for example in bursts. By suitably adapting where bit errors are injected into the data bits, it is possible to follow those patterns, at least to a certain extent, such that the tampering detection mechanism remains entirely undetected.

With respect to an effective verification of the data stored on the storage medium it may be provided that in a first step for each data block the associated error correction code is executed. In a next step, a first hash code may be generated by applying the hash function to the corrected data. A second hash code may be generated by considering a bit of the hash code as '0' in case the respective data block did not have any bit errors, and by considering a bit of the hash code as '1' in case the respective data block contained at least one bit error. Naturally, in case that in the generation process the bit error embedding has been performed in a reverse way by the generator component as described above (i.e. with bit errors being injected in case the respective bit of the hash code is equal to '0' and with

no bit error being injected in case the respective bit of the hash code is equal to '1'), the second hash code has to be calculated also during the verification process in a reverse fashion. In any case, after having generated both hash codes, a bitwise comparison of both strings may be conducted. If both strings match in each bit, the stored data can be regarded as not having been modified. Otherwise, i.e. in case the first and the second hash code are different, the verifier knows that the data has been tampered with.

Embedding a hash code as described so far offers protection to any unknowledgeable attacker. As the hash function is public – in fact, due the restricted number of existing hash functions, an attacker can at least easily figure out which one was deployed – an unknowledgeable attacker could replace the data with his own data and hash result. In that case, the proposed mechanism only offers integrity protection. With respect to further enhanced security, it may be provided that a keyed hash code is employed. The deployment of a keyed hash function significantly increases the protection of the data. However, any verifier to whom the secret key is known can also create valid authentication information. Therefore, to even further enhance the protection the deployment of a signed hash function or a public key cryptography signature proves to be beneficial. Thereby, devices that can generate the content (by possessing the private key), and devices that can verify it (by possessing the public key) would be separated distinctively.

Advantageously, the input of the hash function includes information about the IDs of the data blocks to which it is applied. By this means it is possible to avoid that groups of blocks are copied from one part of the storage media to another.

There are several ways how to design and further develop the teaching of the present invention in an advantageous way. To this end, it is to be referred to the patent claims subordinate to patent claims 1 and 13 on the one hand, and to the following explanation of a preferred example of an embodiment of the invention illustrated by the drawing on the other hand. In connection with the explanation of the preferred example of an embodiment of the invention by the aid of the drawing,

generally preferred embodiments and further developments of the teaching will be explained. In the drawing

- Fig. 1 is a schematic view illustrating the risks associated with removable digital storage media,
- Fig. 2 is a schematic view illustrating a simplified memory mapping of data on a digital storage medium,
- Fig. 3 is a schematic view illustrating the functional principle of ECC (Error Correction Code) verification, and
- Fig. 4 is a schematic view illustrating an embodiment of the present invention.

Fig. 1 illustrates schematically the risks associated with removable storage media and the problems underlying the present invention. In the illustration of Fig. 1, a digital storage medium, which is a removable data storage in form of a compact disk 1, is moved from a location A to a location B. The general problem is that on the way from location A to location B the data stored on the storage medium may be altered by a third party 2. In particular, as illustrated in Fig. 1, a third party 2 may modify, insert and/or delete original data stored on the storage medium.

Making sure that the original data was not altered during its transport is not trivial, and is a key requirement in many areas. Therefore, offering a mechanism which detects any modification made on the data stored on a (removable) storage medium, while at the same time hiding to a potential attacker that the data embeds such protection, will prove of great value for any critical data.

Fig. 2 illustrates schematically a simplified memory mapping of data on a digital storage medium. The overall data is divided into a multitude of data blocks. In Fig. 2 a total number of n data blocks is illustrated. Each data block has associated a corresponding error correction code which allows reconstructing the original data

in case of the occurrence of natural bit errors. It is to be noted, that in the context of the present invention the sequence of the data blocks is not of any special importance. For instance, in contrast to the illustration of Fig. 2, the data blocks may be written on the storage medium in a disordered fashion, which would be typical in case of flash media.

Fig. 3 illustrates schematically the functional principal of the error correction codes ECC. A total of three data blocks is illustrated, with each data block having associated an ECC. When being read, the ECC will correct errors in the associated data block and will reconstruct the original data. In the illustration of Fig. 3, data block 1 and data block 3 are free of errors. However, data block 2 contains one bit error, as indicated by the hatched area. The ECC will correct it when being read, so there is no impact on the application.

Fig. 4 illustrates schematically an embodiment of the present invention. The data structure shown in the upper part of Fig. 4 is the same data structure as described in detail in connection with Fig. 2. From now on this data structure is called a page P . In case the overall data stored on the medium consists of more than one such page, the procedural steps described in detail below will be executed for each page separately.

As to provide a tampering detection function, according to the present invention a hash code is calculated by applying a hash function to the stored data. In the embodiment of Fig. 4 a keyed hash function h_k is employed. Hence, being applied to the illustrated page P , the expression $h_k(P)$ is calculated. As an exemplary value, in the context of the embodiment of Fig. 4 it is assumed that the resulting hash code is $h_k(P) = 010001110001$.

In a next step the bits of the resulting hash code are mapped to the data blocks. In the illustrated embodiment each bit of the hash code is mapped to exactly one data block. Based on this mapping, the hash code is embedded into the data blocks in form of bit errors, which is illustrated in the lower part of Fig. 4 in form of the hatched areas.

According to the invention, the ECC verification is considered as a hidden communication channel. This hidden communication channel is used to embed in the data storage a hash code of the original data. The hash code may be generated by employing an unencrypted hash function, a keyed hash function or a signed hash function, depending on the desired level of protection.

The verification can be carried out by way of executing the following steps:

- 1) For each data block of the page, execute the ECC correction process. If a block is without error, consider the bit of the hash for this data block to be '0', otherwise consider it as '1'.
- 2) Calculate the hash, keyed hash, or signed hash of the corrected page. If the overall data contains more than one page, compute a hash value for each corrected page.
- 3) Compare the string generated in step 1 with the one of step 2. If they match, the data is valid. Otherwise, it has been tampered with.

Adding hidden evidence that the data has been tampered in accordance with the present invention can prove beneficial for numerous applications. For instance, tampering evidence of digital data can assure the detection of any modification on data stored in a digital medium, for instance on a CD, DVD, USB stick, etc. In this context, main areas of concern would be government infrastructures, enterprises and B2B data exchange. In addition, the present invention may be applied in various other fields. For instance, by applying the present invention software modifications can be detected, e.g. in slot machines in casinos or in e-voting machines. Further, in the field of forensics, e.g. pictures taken from a scene of an accident or a crime can be authenticated with forensics information (for instance date, unique camera ID, etc.). Further applications are possible in the field of digital watermarking or to trace a responsible employee in case of an incident, e.g. when a CD with confidential data is found, shared, etc.

The implementation of the invention would most probably need the modification of the firmware of the removable storage writer/reader device. Also, the drivers would have to be modified in order to incorporate key management, settings, verification procedures, etc. In a practical realization, three classes of devices may be

employed: generator, verifier, and reader, wherein any device can be a combination of the three. The generator will be able to write or modify content to the data storage in a legible way. A verifier can read the memory from the data store, and verify that the data was not tampered with. A reader can read the data, but is not able to check the integrity/authenticity of the data. Advantageously, if a keyed hash code is utilized, the generator and verifier share a key if they are not combined into a single device. If public cryptography is applied, the generator must possess the private key and the verifier the public one.

In the following, applications that can build on top of the present invention are considered in some more detail. To this end the function P is considered, which is the application of the invention on a storage S . $P(S)$ means that the present invention was applied and storage S embeds security error bits according to the invention. As written, it can basically come in three different forms: $P(S)$, $P_K(S)$, $P_{Sig}(S)$ which are respectively that the error bits embed a hash, a keyed hash or a public key cryptography signature.

Tampering evidence of digital data

Sensitive data needs to be sent from one company location to another. For that purpose, it is written to a removable medium S such as CD, and embeds security bits, such as $P(S)$. At the reception of the removable medium, the recipient runs the verification algorithm. If it fails, or no security bits are found, then the data was modified on its way to the recipient.

Tampering evidence of software

Similar to early protection of CDs, but the errors are here dependant on the content, and not just fixed to some fixed location. If the abstract data format is composed of error correcting codes, the present invention can be used to protect software that is running from a Flash memory, which is usual for embedded system. One could apply $P_K(S)$ to such system. If an attacker rewrites part of the memory, the security bits won't be found, or be incorrect in the rewritten part. Hence, the modification of the program will be detected. In casinos, program chips are often read and compared to the original program, to avoid such attacks. The present invention would here provide additional protection to this kind of frauds.

Forensics

Recording devices such as cameras could take advantage of present invention to protect data that they register (typically on Flash memory). For example, a camera used by forensics could embed a private key to sign the data, and ensure that a picture remained untouched when reaching the lab or to end up as evidence for court. For example, $P_{sig}(S)$ could be applied for the memory card. The court could then use the public key to ensure that the evidence remained untouched.

Digital watermarking

Comparable to forensics, a company might like to prevent data loss and data leakage by embedding a private key, i.e. by applying $P_{sig}(S)$, for every removable media writer in its company (by for example having a modified firmware on each employee laptop). If a lost USB or CD ROM is found, the originator of the signature could be brute forced, and thus find the incriminating device which wrote that data.

It is to be noted that the present invention is related to ECC codes. This is in contrast to what is done in the watermarking area, where errors are also injected, but instead of being corrected, they are just expected to have the least impact on the data. Also, if there are codes to detect errors, like for example in the ZIP format, this is not enough to apply the present invention, since the errors will spread throughout the decoding, and hence have dramatic impact on the application.

Many modifications and other embodiments of the invention set forth herein will come to mind the one skilled in the art to which the invention pertains having the benefit of the teachings presented in the foregoing description and the associated drawings. Therefore, it is to be understood that the invention is not to be limited to the specific embodiments disclosed and that modifications and other embodiments are intended to be included within the scope of the appended claims. Although specific terms are employed herein, they are used in a generic and descriptive sense only and not for purposes of limitation.

Claims

1. Method for providing a tampering detection function for a storage medium, in particular a portable digital storage device, wherein digital data stored on said storage medium is divided into a plurality of data blocks with each data block having associated an error correction code,
c h a r a c t e r i z e d i n the steps of
calculating a hash code by applying a hash function to the stored data,
performing a mapping of the bits of said hash code to said data blocks, and
based on the mapping, embedding said hash code into said data blocks in form of bit errors.
2. Method according to claim 1, wherein for a bit of said hash code being equal to '1' at least one bit error is injected into the corresponding data block, and wherein for a bit of said hash code being equal to '0' no bit error is injected into the corresponding data block, or vice versa.
3. Method according to claim 1 or 2, wherein each bit of said hash code is mapped to one of said data blocks.
4. Method according to claim 1 or 2, wherein each bit of said hash code is mapped to a predefined number of said data blocks.
5. Method according to any of claims 1 to 4, wherein the normal rate of bit errors for the stored digital data is taken into account.
6. Method according to any of claims 1 to 5, wherein only each i-th data block is considered for the mapping, with i being an integer larger or equal 2.
7. Method according to any of claims 1 to 6, wherein the positions of said embedded bit errors inside a data block are chosen such that said positions transmit information about said hash code.

8. Method according to any of claims 1 to 7, wherein a verification algorithm for said storage medium includes the steps of
- for each of said data blocks, executing the associated error correction code,
 - generating a first hash code by applying said hash function to the corrected data,
 - generating a second hash code by considering a bit of the hash code as '0' in case the respective data block is without any bit error and considering a bit of the hash code as '1' in case the respective data block contains at least one bit error, and
 - comparing bitwise said first and said second hash code.
9. Method according to claim 8, wherein the stored data is regarded as not having been modified in case said first and said second hash code are identical, and wherein the stored data is regarded as having been modified in case said first and said second hash code are different.
10. Method according to any of claims 1 to 9, wherein said hash code is a keyed hash code.
11. Method according to any of claims 1 to 9, wherein said hash code is a signed hash code.
12. Method according to any of claims 1 to 11, wherein the input of said hash function includes information about the IDs of the data blocks to which it is applied.
13. Storage medium, in particular a portable digital storage device, being equipped with a tampering detection function, wherein digital data stored on said storage medium is divided into a plurality of data blocks with each data block having associated an error correction code,
- c h a r a c t e r i z e d i n that the data includes a hash code that is embedded into said data blocks in form of bit errors,
- wherein the hash code has been calculated by applying a hash function to the stored data, and by performing a mapping of the bits of said hash code to said data blocks.

14. Device for writing digital data to a storage medium, in particular by applying a method according to any of claims 1 to 12, comprising

an error correction component such that said digital data to be stored on said storage medium is divided into a plurality of data blocks with each data block having associated an error correction code,

c h a r a c t e r i z e d i n that the device further comprises

a generator component being configured to calculate a hash code by applying a hash function to the data to be stored, and to perform a mapping of the bits of said hash code to said data blocks, and

injection means being configured to embed said hash code, based on the mapping, into said data blocks in form of bit errors.

15. Device according to claim 14, further comprising a verifying component being configured

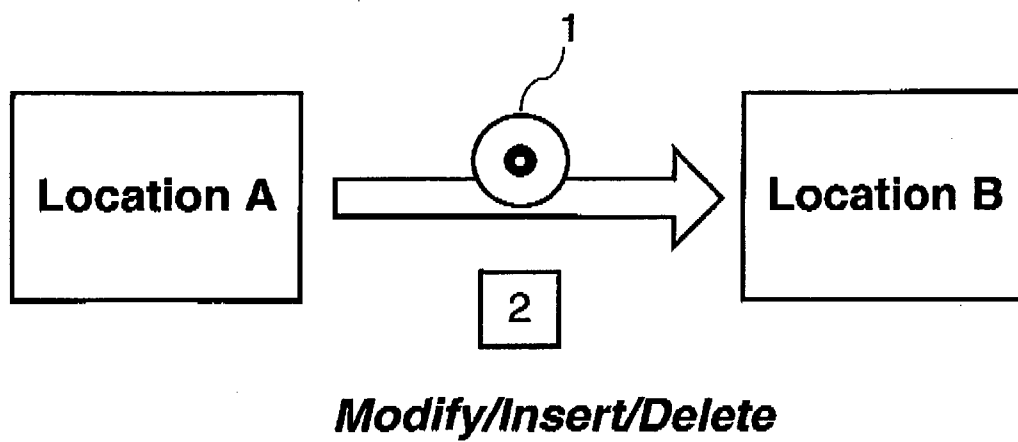
to execute for each of said data blocks the associated error correction code,

to generate a first hash code by applying said hash function to the corrected data,

to generate a second hash code by considering a bit of the hash code as '0' in case the respective data block is without any bit error and considering a bit of the hash code as '1' in case the respective data block contains at least one bit error, or vice versa, and

to compare bitwise said first and said second hash code.

1/3

**Fig. 1**

2/3

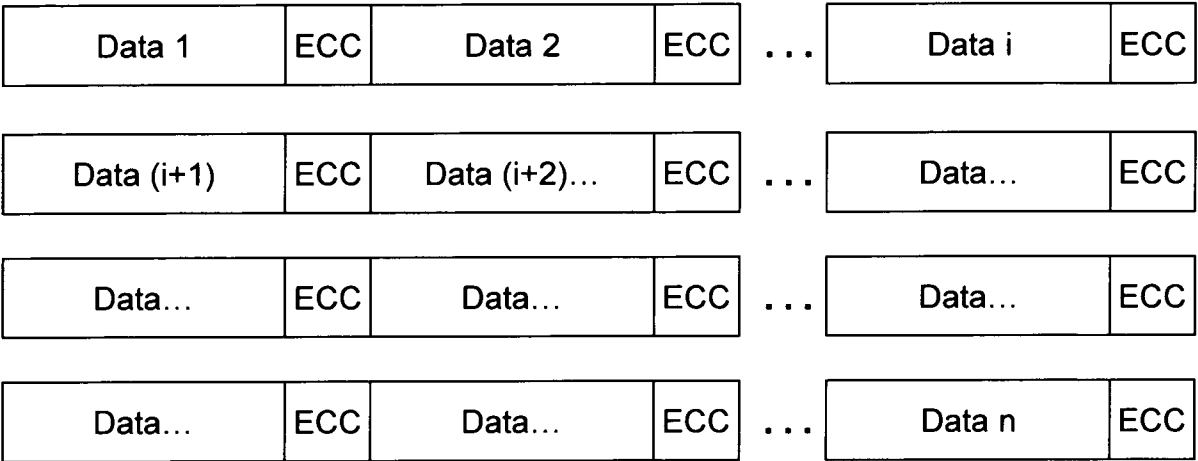


Fig. 2

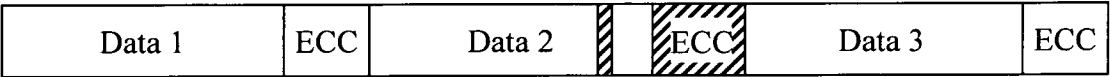
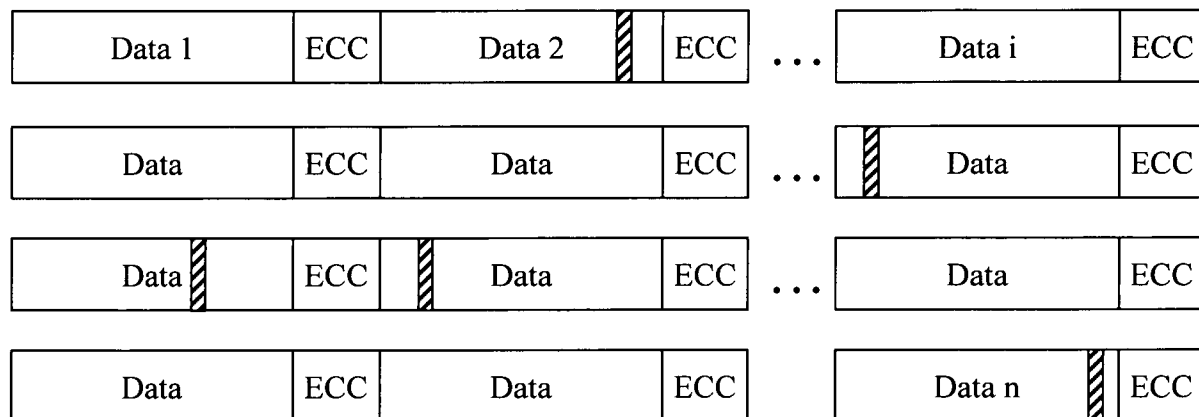
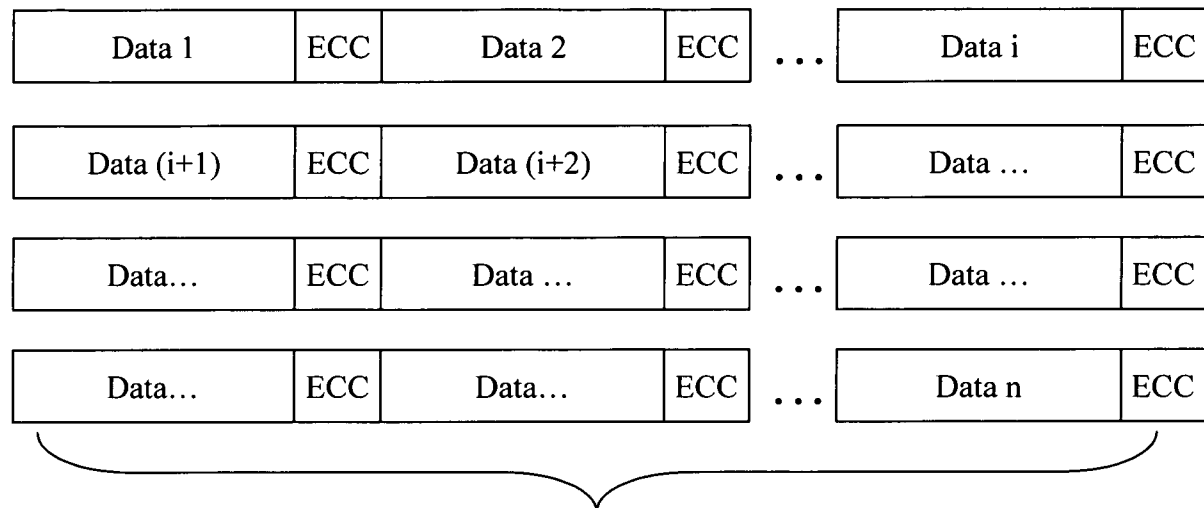


Fig. 3

3/3

**Fig. 4**

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2010/007657

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F21/02 G06F21/24
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)

EP0-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	GB 2 348 585 A (IBM [US]) 4 October 2000 (2000-10-04) abstract pages 5-8; figures 3-5 -----	1-15
X	PETER WAYNER: "Summary", DISAPPEARING CRYPTOGRAPHY: INFORMATION HIDING: STEGANOGRAPHY & WATERMARKING, vol. 3rd edition, 17 December 2008 (2008-12-17), pages 1-2, XP007918120, DOI: HTTP:// PROQUEST.SAFARIBOOKSONLINE.COM/PRINT?XMLID =978012374 4791/CH ISBN: 9780123744791 [retrieved on 2011-04-06] Section 3.4 ----- -/--	1-15

☒ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

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

"E" earlier 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

7 April 2011

Date of mailing of the international search report

14/04/2011

Name and mailing address of the ISA/

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

Authorized officer

Schäfer, Andreas

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2010/007657

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 2006/253722 A1 (NIELSEN HANS H [US] ET AL) 9 November 2006 (2006-11-09) abstract -----	1-15

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2010/007657

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
GB 2348585	A	04-10-2000	CN 1262509 A	09-08-2000
			JP 2000224411 A	11-08-2000
			US 6564322 B1	13-05-2003

US 2006253722	A1	09-11-2006	NONE	
