(12) UK Patent Application (19) GB (11) 2 435 333 (13) A

(43) Date of A Publication 22.08.2007

(21) Application No: 0602012.7

(22) Date of Filing: 01.02.2006

(71) Applicant(s):
Hewlett-Packard Development Company L.P., 20555 S.H.249, Houston, Texas 77070, United States of America

(72) Inventor(s):
Gregory Keith Trezise
Jonathan Peter Buckingham
Andrew Hana

(74) Agent and/or Address for Service:
Hewlett-Packard Limited
IP Section, Filton Road, Stoke Gifford, BRISTOL, BS34 8QZ, United Kingdom

(51) INT CL:
G06F 12/14 (2006.01) ***G06F 1/00*** (2006.01)
G06F 3/06 (2006.01) ***G06F 21/02*** (2006.01)
G11B 20/00 (2006.01)

(52) UK CL (Edition X):
G4A AAP A23A
G5R RHB

(56) Documents Cited:
EP 1615368 A1 **EP 1440439 A1**
EP 1020856 A2 **US 20050278257 A1**
US 20030074319 A1

(58) Field of Search:
UK CL (Edition X) **G4A, G5R, H4P**
INT CL **G06F, G11B, H04L**
Other: **WPI, EPODOC**

(54) Abstract Title: **Data transfer device**

(57) A data transfer device for transferring data to and from a removable data storage item. When transferring data to the removable data storage item, the data transfer device encrypts the data using an encryption key, and additionally encrypts predetermined reference data using the encryption key. The data transfer device then stores the encrypted data and the encrypted reference data to the removable data storage item. When transferring data from the removable data storage item, the data transfer device retrieves the encrypted data from the removable data storage item, decrypts the encrypted data and outputs the decrypted data. In the event that an error occurs during decryption, the encrypted reference data stored on the removable data storage item may be retrieved and used to determine the cause of the error. In particular, the encrypted reference data may be used to determine whether the correct encryption key has been used for decryption. The reference data may comprise a copy of the encryption key. The data transfer device may be a tape drive and the removable data storage item may be a tape cartridge.

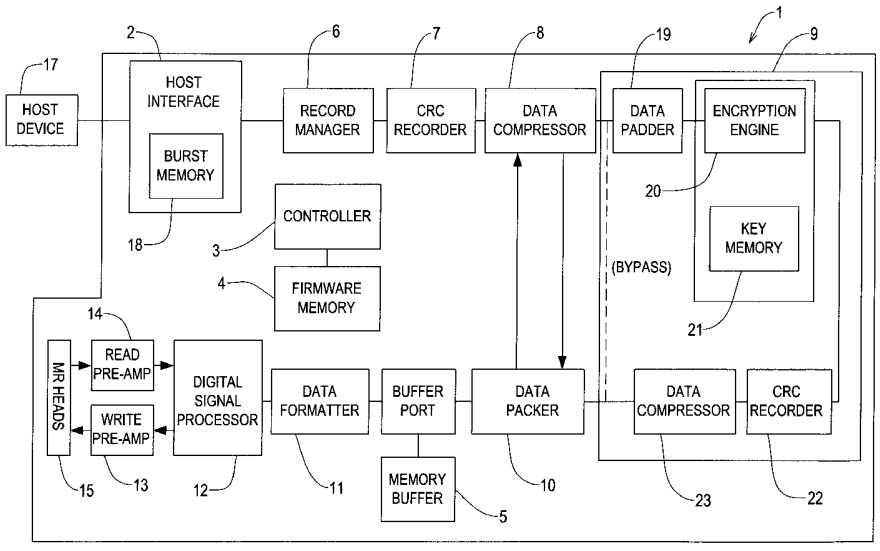

FIG. 1

GB 2 435 333 A

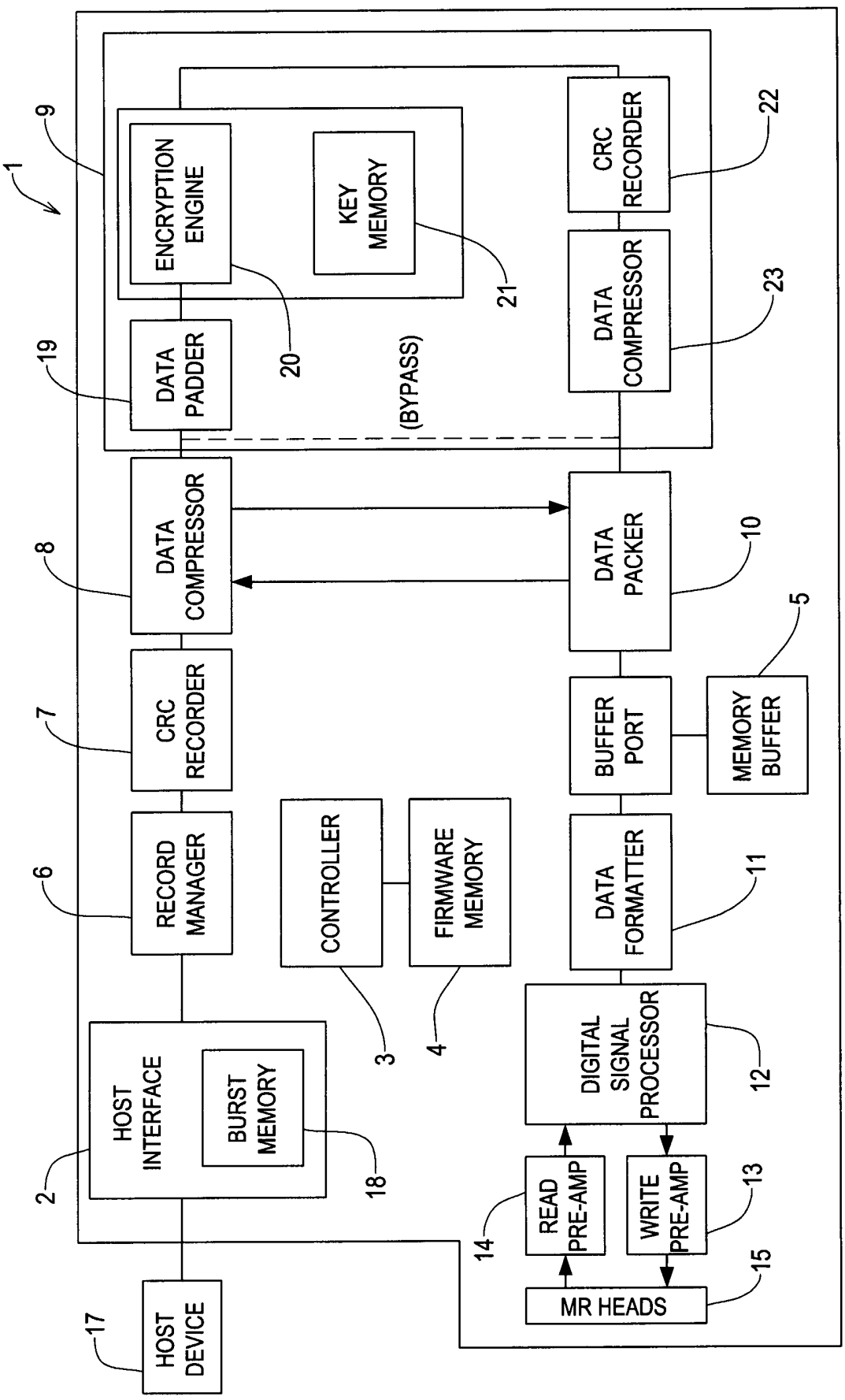
1
9
ENCRYPTION ENGINE
KEY MEMORY
CRC RECORDER
22
21
20
DATA PADDER
(BYPASS)
DATA COMPRESSOR
23
19
DATA COMPRESSOR
DATA PACKER
8
10
CRC RECORDER
BUFFER PORT
MEMORY BUFFER
5
7
RECORD MANAGER
CONTROLLER
FIRMWARE MEMORY
DATA FORMATTER
6
11
HOST INTERFACE
BURST MEMORY
DIGITAL SIGNAL PROCESSOR
2
18
3
4
12
READ PRE-AMP
WRITE PRE-AMP
13
14
MR HEADS
15
HOST DEVICE
17

FIG. 1

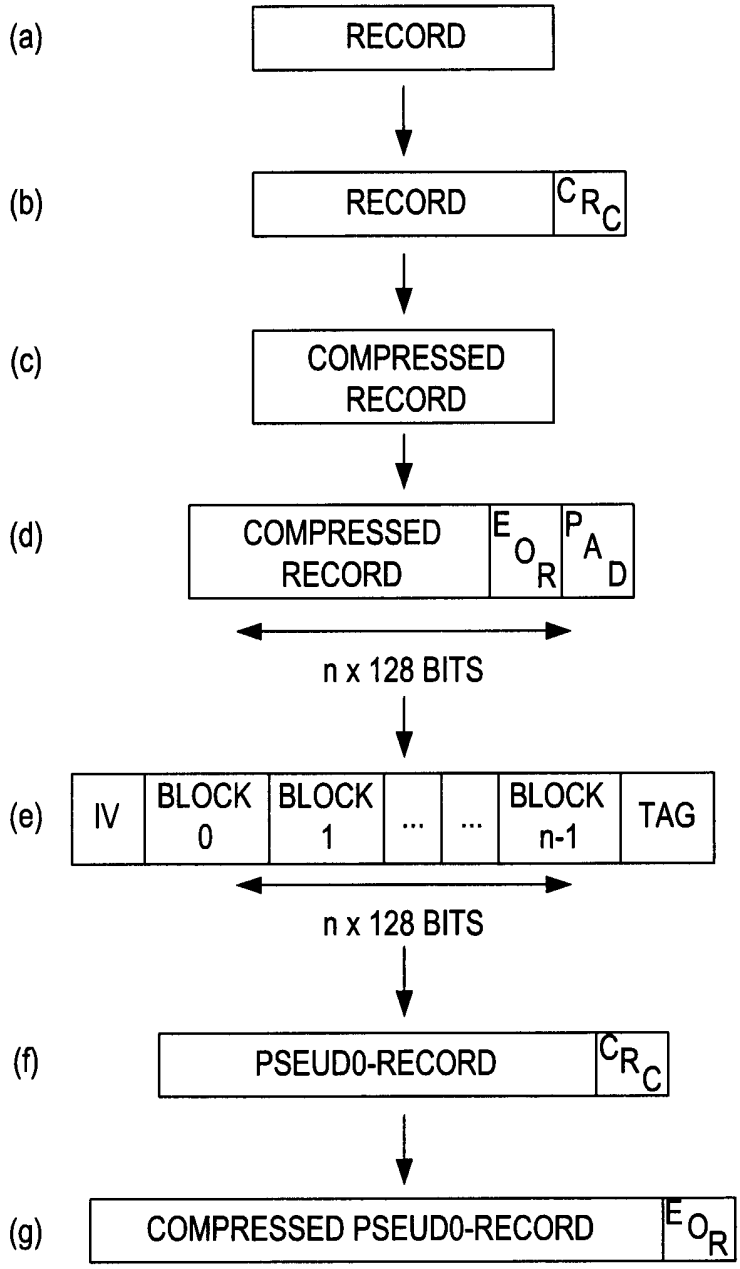
(a)
RECORD
(b)
RECORD
CRC
(c)
COMPRESSED RECORD
(d)
COMPRESSED RECORD
EOR
PAD
n x 128 BITS
(e)
IV
BLOCK 0
BLOCK 1
...
...
BLOCK n-1
TAG
n x 128 BITS
(f)
PSEUD0-RECORD
CRC
(g)
COMPRESSED PSEUD0-RECORD
EOR

FIG. 2

DATA TRANSFER DEVICE

FIELD OF THE INVENTION

The present invention relates to a data transfer device for transferring data between a host device and a removable data storage item, wherein data are encrypted or decrypted by the data transfer device during data transfer.

BACKGROUND OF THE INVENTION

Data backup is a valuable tool in safeguarding important data. Data are generally backed-up onto removable data storage items, such as tape cartridges or optical discs, such that the backup data may be stored at a different geographical location to the primary data.

By storing important data onto removable data storage items, security issues become a consideration. For example, a visitor to a site might easily pocket a tape cartridge storing large amounts of commercially sensitive data.

Many backup software packages provide the option of encrypting data prior to backup. A drawback with this approach, however, is that the same software package must be used in order to retrieve and decrypt the backup data. Accordingly, backup data cannot be recovered using other legitimate systems where the backup software is not provided. Additionally, software encryption increases the time required to backup data and consumes valuable computer resources.

SUMMARY OF THE INVENTION

In a first aspect, the present invention provides a data transfer device for transferring data to a removable data storage item, the data transfer device being operable to: receive data to be stored to the removable data storage item; encrypt the data using an encryption key; encrypt predetermined reference data using the encryption key; and store the encrypted data and the encrypted reference data to the removable data storage item.

Preferably, the reference data comprises a copy of the encryption key.

Advantageously, the data transfer device is operable to store the encrypted data as one or more data blocks, and to store a copy of the encrypted reference data with each data block.

Conveniently, each data block comprises a data region and an information table, and a copy of the encrypted reference data is stored in the information table.

Preferably, the data transfer device is operable to store the encrypted data in a format selected from one of the generations of LTO and DDS/DAT formats.

Advantageously, the data is received as one or more records and a copy of the reference data is provided within a record, and the data transfer device is operable to encrypt the records, including the record comprising the reference data, using the encryption key and to store the encrypted records to the removable data storage item.

Conveniently, the data transfer device is operable to compress the data prior to encryption.

Preferably, the data transfer device stores a copy of the encryption key and is suitable for transferring data from the removable data storage item, the data

transfer device being operable to: retrieve the encrypted data from the removable data storage item; decrypt the encrypted data using the encryption key stored by the data transfer device; and output the decrypted data.

Advantageously, the data transfer device additionally stores a copy of the predetermined reference data and is operable to: retrieve the encrypted reference data from the removable data storage item; compare the reference data retrieved from the removable data storage item with the reference data stored by the data transfer device; and generate an error message in the event that the reference data retrieved from the removable data storage item does not substantially correspond with the reference data stored by the data transfer device.

Conveniently, retrieving the encrypted reference data, comparing the reference data and generating an error message are performed in the event that the encrypted data cannot be successfully decrypted using the encryption key stored by the data transfer device.

Preferably, the data transfer device in comparing reference data is operable to: encrypt the reference data stored by the data transfer device using the encryption key; and compare the encrypted reference data retrieved from the removable data storage item with the encrypted reference data created by the data transfer device.

Alternatively, the data transfer device in comparing reference data is operable to: decrypt the encrypted reference data using the encryption key stored by the data transfer device; and compare the decrypted reference data with the reference data stored by the data transfer device.

Advantageously, the data transfer device in comparing reference data is operable to determine the number of locations at which the reference data

retrieved from the removable data storage item and the reference data stored by the data transfer device differ, and the data transfer device is operable to generate the error message in the event that the number of locations is greater than a predetermined value.

Conveniently, the data transfer device is a tape drive and the removable data storage item is a tape cartridge.

In a second aspect, the present invention provides a data transfer device for transferring data to a removable data storage item, the data transfer device comprising: means for receiving data to be stored to the removable data storage item; means for encrypting the data using an encryption key; means for encrypting predetermined reference data using the encryption key; and means for storing the encrypted data and the encrypted reference data to the removable data storage item.

Preferably, the reference data comprises a copy of the encryption key

Advantageously, the data transfer device comprises: means for storing a copy of the reference data; means for retrieving the encrypted reference data from the removable data storage item; means for comparing the reference data retrieved from the removable data storage item with the reference data stored in the means for storing, and means for generating an error message in the event that the reference data retrieved from the removable data storage item does not substantially correspond with the reference data stored in the means for storing.

In a third aspect, the present invention provides a method of operating a data transfer device for transferring data to a removable data storage item, the method comprising: receiving data to be stored to the removable data storage item; encrypting the data using an encryption key, encrypting predetermined

reference data using the encryption key; and storing the encrypted data and the encrypted reference data to the removable data storage item.

Preferably, the reference data comprises a copy of the encryption key.

Advantageously, the method further comprises: storing a copy of the reference data in memory; retrieving the encrypted reference data from the removable data storage item; comparing the reference data retrieved from the removable data storage item with the reference data stored in the memory; and generating an error message in the event that the reference data retrieved from the removable data storage item does not substantially correspond with the reference data stored in the memory.

In a fourth aspect, the present invention provides a computer program product storing computer program code executable by a data transfer device to perform the aforementioned method.

BRIEF DESCRIPTION OF THE DRAWINGS

In order that the present invention may be more readily understood, embodiments thereof will now be described, by way of example, with reference to the accompanying drawings, in which:

Figure 1 is a schematic block diagram of a tape drive embodying the present invention; and

Figure 2 illustrates a record at various stages of formatting by the tape drive of Figure 1.

DETAILED DESCRIPTION

The tape drive 1 of Figure 1 comprises a host interface 2, a controller 3, a firmware memory 4, a memory buffer 5, a record manager 6, a CRC recorder 7,

a data compressor 8, a data encryptor 9, a data packer 10, a data formatter 11, a digital signal processor 12, write 13 and read 14 pre-amplifiers, and magneto-resistive heads 15. With the exception of the data encryptor 9 and the software stored in the firmware memory 4, the components of the tape drive 1 are identical to those employed in conventional linear-tape-open (LTO) tape drives.

The host interface 2 controls the exchange of data between the tape drive 1 and a host device 17. Control signals received from the host device 17 by the interface 2 are delivered to the controller 3, which, in response, controls the operation of the tape drive 1. Data received from the host device 17 typically arrives in high speed bursts and the host interface 2 includes a burst memory 18 for storing data received from the host device 17.

The controller 3 comprises a microprocessor, which executes instructions stored in the firmware memory 4 to control the operation of the tape drive 1.

The record manager 6 retrieves data from the bust memory 18 of the host interface 2 and appends record boundaries. The CRC recorder 7 then appends a cyclic redundancy check (CRC) to each record. Each of the protected records is then compressed by the data compressor 8 using LTO scheme-1 (ALDC) compression. The integrity of the compressed records is then verified by the data compressor 8, after which the compressed records are delivered to the data encryptor 9.

The data encryptor 9 comprises a data padder 19, an encryption engine 20, a key memory 21, a CRC recorder 22 and a data compressor 23. The CRC recorder 22 and data compressor 23 of the data encryptor 9 shall be referred to hereafter as the encrypt CRC recorder 22 and encrypt data compressor 23 so as to distinguish them from the other CRC recorder 7 and data compressor 8.

As described below, the data encryptor 9 employs block encryption, each block having 128 bits. The data padder 19 therefore appends an end-of-record (EOR) codeword to each compressed record and pads each compressed record with redundant data (e.g. with zeros) such that each compressed record is an integral number of 128 bits.

The encryption engine 20 employs a Galois Counter Mode (GCM) encryption algorithm to encrypt each padded, compressed record. The key memory 21 may be volatile or non-volatile, depending on the intended applications of the tape drive 1, and stores a 256-bit encryption key that is used by the encryption engine 20. Other keys such as a 128 or a 192 bit key may also be used. The Galois/Counter Mode is specified in "The Galois/Counter Mode of Operation" by David A. McGrew and John Viega available from NIST/CSRC.

The encryption engine 20 divides each padded, compressed record into blocks of 128 bits. Each block is then encrypted using the encryption key held in key memory 21 and a counter value.

After data encryption, the encryption engine 20 appends an initialisation vector (sometimes referred to as an initial vector) to the beginning of the blocks of ciphertext and an authentication tag to the end of the blocks of ciphertext to create a pseudo-record. The initialisation vector is the counter value for the first block of ciphertext of the pseudo-record (i.e. block number = 0), whilst the authentication tag is generated in accordance with the GCM specification and comprises a form of checksum data generated over the data of a record. The tag may also be generated over any additional authenticated data (AAD) which may or may not be prefixed to each record. The tag, AAD and prefixing AAD are all concepts enshrined in the GCM and IEEE1619.1 standards.

The pseudo-record, comprising the IV, blocks of ciphertext and authentication tag, is delivered to the encrypt CRC recorder 22, which appends a CRC to the

pseudo-record to create a protected pseudo-record. The protected pseudo-record is then delivered to the encrypt data compressor 23, which compresses the protected pseudo-record using LTO scheme-2 (no-compress) compression. Owing to encryption, the pseudo-record comprises random data and therefore the pseudo-record is incompressible. It is for this reason that scheme-2 compression is employed. Although no compression is actually achieved, the compressed pseudo-record consists of LTO codewords (e.g. compression, scheme and reset codewords). Consequently, the compressed pseudo-record is LTO compliant

The compressed encrypted pseudo-record is then delivered to the data packer 10, which appends an EOR codeword to the compressed pseudo-record and packs sequential compressed pseudo-records together to form a compressed data stream, which is then written to the memory buffer 5

Figure 2 illustrates a record received from the host device 17 at various stages of formatting by the tape drive 1. Figure 2(a) illustrates the record as received by the tape drive 1, which may be of any size. Figure 2(b) illustrates the record after processing by the CRC recorder 6, and Figure 2(c) illustrates the protected record after compression by the data compressor 7. Figure 2(d) illustrates the compressed record after formatting by the data padder 19. Figure 2(e) illustrates the pseudo-record created after encryption. Figure 2(f) illustrates the pseudo-record after processing by the encrypt CRC recorder 22, and Figure 2(e) illustrates the protected pseudo-record after compression by the encrypt data compressor 23 and the data packer 10. LTO format specifies also that records must be padded to a 32 bit boundary hence the potential use of a 4-byte pad appended to the end of the pseudo-record.

As in conventional LTO tape drives, the controller 3 then divides or partitions the compressed data stream into data chunks of a predetermined size (e.g. 403884 bytes for LTO1/LTO2 and 1616940 for LTO3/LTO4) and appends a

data set information table (DSIT) to each data chunk to create a data set. In LTO1, LTO2, LTO3 and LTO4 formats, the DSIT comprises 468 bytes. Each data set is then delivered to the data formatter 11, which ECC-encodes the data set, randomises the ECC-encoded data to remove long sequences, and RLL encodes the randomised data. The RLL-encoded data are then processed by the digital signal processor 12 and delivered, via the write pre-amplifier 13, to write head elements 15 which write the data set to a magnetic tape.

The read process is basically the reverse of the write process. In response to a request to retrieve a particular record, the tape drive 1 first locates the relevant data set or group of data sets. The data set is then read from the tape by read head elements 16 which generate an analogue signal. The analogue signal is then amplified by the read pre-amplifier 14 and processed by the digital signal processor 12 to generate a digital data stream. The digital data stream is then RLL-decoded, unscrambled and ECC-decoded by the data formatter 11 to create the data set.

The chunk of data corresponding to the data region of the data set is then delivered to the data packer 10, which unpacks the chunk of data to create one or more compressed pseudo-records. The location of each compressed pseudo-record is determined by the EOR codewords previously appended by the data packer 10 during data storage.

Each compressed pseudo-record is then decompressed by means of the encrypt data compressor 23. The CRC appended to each pseudo-record is discarded by the encrypt data compressor 23 and the resulting pseudo-records are delivered to the encryption engine 20, which then decrypts the pseudo-records. The encryption engine 20 uses the encryption key stored in key memory 21 and the initialization vector stored at the beginning of each pseudo-record to decrypt the pseudo-records and generate in response padded, compressed records.

The padded, compressed records are then delivered to the data compressor 8, which decompresses the records. Owing to the presence of the EOR codeword, the data compressor 8 ignores any padding to the compressed records.

The controller 3 then reads each of the retrieved records in turn until the requested record is identified, whereupon it is delivered to the host device 11 via the host interface 2

Each read and write command issued by the host device 17 is accompanied by an encryption key. The tape drive 1, upon receiving a command, stores the accompanying encryption key in key memory 21 for subsequent use by the encryption engine 20. Data to be written or read by the tape drive 1 are then encrypted or decrypted using the encryption key stored in key memory 21.

When a new command is received by the tape drive 1, the accompanying encryption key overwrites the contents of the key memory 21. When no encryption or decryption is required, the read or write command is accompanied by a blank encryption key. Alternatively, once a command has been completed by the tape drive 1, the encryption key may be erased from the key memory 21 such that subsequent commands are performed without data encryption/decryption occurring.

Rather than an encryption key accompanying each read and write command, the host device 17 may alternatively issue a special SET_KEY command, which includes the encryption key to be stored. The tape drive 1, in response to receiving the SET_KEY command, stores the received encryption key to key memory 21 for subsequent use by the encryption engine 20. In this alternative embodiment, the contents of the key memory 21 are unchanged until such time as a new SET_KEY command is received and the contents of the key memory 21 are overwritten.

By permitting different encryption keys to be used during operation, data may be stored to tape using a plurality of different encryption keys so as to further increase data security.

If a new encryption key is received during data write, the controller 3 appends an end-of-marker codeword followed by redundant data to the compressed data stream such that the current, partial data chunk (i.e. data region) is padded to the required, predetermined size (i.e. 403884 bytes for LTO1/LTO2 and 1616940 for LTO3/LTO4). By padding the compressed data stream in this manner whenever a new encryption key is received, each data set comprises only records that have been encrypted using the same encryption key. Consequently, locating and retrieving a record within a particular data set requires the provision of only one encryption key.

If an incorrect encryption key is employed by the tape drive 1 when retrieving encrypted data from tape, the data compressor 8 will generate an error on the basis that the record length and/or the CRC data of the decompressed record is incorrect. The data compressor 8, however, will also generate an error during data retrieval if the encrypted data read back from the tape is itself corrupt, regardless of whether or not the correct encryption key has been used. Accordingly, it is not always possible to reliably determine the cause of a data retrieval error.

In order that the tape drive 1 may reliably determine the cause of a data retrieval error, a self-encrypted copy of the encryption key is stored to tape along with any data encrypted using that particular encryption key. In the event that an error occurs during subsequent data retrieval, the copy of the self-encrypted encryption key is retrieved from tape. A copy of the encryption key stored in key memory 21 is then encrypted by the encryption engine 20 to

create a second self-encrypted encryption key, and the two self-encrypted encryption keys are compared.

If the two self-encrypted encryption keys correspond, a positive verification is made that the encryption key stored in key memory 21 is the correct encryption key for the data being retrieved. The cause of the error generated during data retrieval can therefore be positively identified as corrupt data read back from tape. Conversely, if the two keys do not correspond, the cause of the error can be positively identified as the use of an incorrect encryption key for the data being retrieved.

Embodiments for storing a self-encrypted copy of the encryption key to tape, and subsequently retrieving the self-encrypted key from tape in order to determine the cause of a data retrieval error, will now be described.

In a first embodiment, the host device 17 delivers to the tape drive 1 a copy of the self-encrypted encryption key in the form of a data record. This data record, which shall be referred to hereafter as an encrypt data record, is formatted by the tape drive 1 in the same manner as that for a conventional data record, i.e. the encrypt data record is compressed by the data compressor 8, encrypted by the data encryptor 9, and appended to the compressed data stream by the data packer 10

The host device 17 delivers an encrypt data record at the beginning of a data write process. In particular, the host device 1 delivers the encrypt data record prior to any user data records. Additionally, the host device 1 delivers an encrypt data record whenever a change in encryption key occurs.

Since a new data set is created at the beginning of each write process or whenever a change in the encryption key has occurred, an encrypt data record will be recorded as the first record of a data set. The tape drive 1 updates the

directory of the tape such that each data set storing an encrypt data record is labelled as such. Accordingly, when encrypted data are to be later retrieved, the data set storing the relevant encrypt record can be quickly and easily identified.

When data are to be retrieved from tape, the host device 17 reads the contents of the tape directory. Using the contents of the tape directory, the host device 17 determines the location of the data set(s) that comprises the requested data record(s). The host device 17 then issues commands to the tape drive 1 requesting that the identified data set(s) be retrieved from tape and delivered to the host device 17. In response to the data read command, the tape drive 1 retrieves the identified data set(s) from tape, processes the data set(s) in the manner described above, and delivers the data records contained therein to the host device 17. If the tape drive 1 returns an error during data retrieval, the host device 17 determines, from the contents of the tape directory, the location of the data set that comprises the relevant encrypt data record. The host device 17 then delivers a KEY_CHECK command to the tape drive 1 along with the location of the data set storing the encrypt data record. In response, the tape drive 1 retrieves the relevant data set and processes the data set so as to obtain the encrypt data record. The controller 3 then extracts the self-encrypted encryption key from the encrypt data record and compares this against a self-encrypted copy of the encryption key that is stored in key memory 21.

If the two self-encrypted keys correspond, a positive verification is made of the encryption key stored in key memory 21 and the tape drive 1 returns a KEY_CORRECT message to the host device 17. Since the two keys correspond, the initial error generated by the tape drive 1 during data retrieval cannot be attributed to an incorrect encryption key being used for decryption. Consequently, in response to receiving a KEY_CORRECT message, the host

device 17 outputs an error indicating that the requested data record(s) is corrupt.

If the two self-encrypted keys do not correspond, the tape drive 1 determines whether or not the copy of the self-encrypted encryption key stored on tape is itself corrupt. Two possible examples for determining whether the self-encrypted encryption key stored on tape is corrupt will now be described.

In the first example, the self-encrypted encryption key stored to tape is protected with an error correction code (ECC) or a data redundancy check. The encrypt data record then comprises both the self-encrypted encryption key and the ECC. If the self-encrypted encryption key read back from tape is corrupt and cannot be repaired by means of the ECC, the tape drive 1 delivers a KEY_CORRUPT message to the host device 17. If, however, the ECC or redundancy check indicates that the self-encrypted encryption key read back from tape is not corrupt, the tape drive 1 delivers a KEY_INCORRECT message to the host device 17.

In response to receiving a KEY_CORRUPT message, the host device 17 outputs an error indicating that the tape is corrupt. In response to receiving a KEY_INCORRECT message, the host device 17 outputs an error indicating that the data record(s) being retrieved was encrypted using a different encryption key to that stored in key memory 21.

In the second example, the number of bytes that differ between the two self-encrypted keys is used to determine whether the self-encrypted key retrieved from tape is corrupt. Corruption of the self-encrypted key retrieved from tape is likely to be restricted to a small number of bytes. In contrast, two different encryption keys (or two different self-encrypted encryption keys) are likely to differ at many or all byte locations. Accordingly, in the second example, the controller 3 determines the number of byte locations at which a difference

exists between the self-encrypted encryption key retrieved from tape and a self-encrypted copy of the encryption key stored in key memory 21. The tape drive 1 additionally stores a predetermined number, which shall be referred to hereafter as the BYTE_ERROR_THRESHOLD. The value of BYTE_ERROR_THRESHOLD will depend upon the tape format or implementation being employed. If the number of differing byte locations determined by the controller 3 is lower than or equal to the BYTE_ERROR_THRESHOLD, this suggests a corrupt rather than incorrect encryption key read back from tape and the tape drive 1 therefore delivers a KEY_CORRUPT message to the host device 17. If, however, the number of differing byte locations determined by the controller 3 is greater than the BYTE_ERROR_THRESHOLD, this suggests an incorrect rather than corrupt encryption key read back from tape and the tape drive 1 therefore delivers a KEY_INCORRECT message to the host device 17.

As with the first example, the host device 17 outputs an error indicating that the tape is corrupt in response to receiving a KEY_CORRUPT message, and outputs an error indicating that the data record(s) being retrieved was encrypted using a different encryption key to that stored in key memory 21 in response to receiving a KEY_INCORRECT message.

In both examples, the tape drive 1 delivers to the host device 17 one of three possible messages in response to a KEY_CHECK command. The first message is KEY_CORRECT, which indicates that the encryption key stored in key memory 21 is the correct key for the data being retrieved. The second message is KEY_INCORRECT, which indicates that the encryption key stored in key memory 21 is not the correct key for the data being retrieved. The third message is KEY_CORRUPT, which indicates that the self-encrypted key retrieved from tape is corrupt. Consequently, the host device 17 is able to determine the cause of a data retrieval error. In particular, the host device 17

is able to determine whether data being retrieved is corrupt or whether the encryption key being used for decryption is incorrect.

The two examples described above are not exclusive and may be combined to better determine whether or not the self-encrypted encryption key stored on tape is corrupt.

In an alternative embodiment for writing and retrieving a copy of the self-encrypted key to tape, the DSIT of each data set includes a field that stores a copy of the self-encrypted encryption key For example, the first 228 bytes of the DSIT is reserved for manufacturer use and would provide a suitable location for storing a copy of the self-encrypted encryption key, which occupies only 16 bytes (for an encryption key of 128 bits) or 32 bytes (for an encryption key of 192 or 256 bits).

The controller 3, when appending a DSIT to the data region of a data set (i.e. when appending a DSIT to each data chunk of the compressed data stream), copies the self-encrypted encryption key to the DSIT of the data set. Consequently, each data set securely stores a copy of the encryption key used to encrypt the data stored within the data region.

When data are to be retrieved from tape, the host device 17 first reads the contents of the tape directory. Using the contents of the tape directory, the host device 17 determines the location of the data set(s) that comprises the requested data record(s). The host device 17 then issues a command to the tape drive 1 requesting that the identified data set(s) be retrieved from tape and delivered to the host device 17. If the tape drive 1 returns an error during data retrieval, the host device 17 in response delivers a KEY_CHECK command to the tape drive 1 along with the identity of the data set. In response, the tape drive 1 retrieves the self-encrypted encryption key from the DSIT of the data set

and compares this against a self-encrypted copy of the encryption key that is stored in key memory 21.

In the same manner as that described above for the first embodiment, the tape drive 1 then determines whether the self-encrypted key retrieved from tape is corrupt, or whether it corresponds or differs from a self-encrypted copy of the key stored in key memory. In particular, the tape drive 1 returns one of three possible messages in response to a KEY_CHECK command, namely KEY_CORRECT, KEY_INCORRECT and KEY_CORRUPT.

In the embodiments described above, the tape drive 1 returns an error signal to the host device 17 should a problem occur during data retrieval. The host device 17 in response delivers a KEY_CHECK command to the tape drive 1, which in response returns to the host device 17 one of three possible messages, namely KEY_CORRECT, KEY_INCORRECT and KEY_CORRUPT. In an alternative embodiment, the tape drive 1 upon detecting an error during data retrieval immediately enters key-checking mode. In particular, the tape drive 1 does not wait for a KEY_CHECK command to be received but instead behaves in the manner described above as if a KEY_CHECK command had been received. If the self-encrypted key is stored as an encrypt data record, the tape drive 1 determines, from the tape directory, the location of the relevant record, retrieves the record, and extracts the self-encrypted key from the record. If the self-encrypted encryption key is stored in the DSIT of a data set, the tape drive 1 retrieves the from the DSIT of the data set for which an error has occurred.

Since both KEY_CORRECT and KEY_CORRUPT messages are indicative of a corrupt tape, the tape drive 1 need only return one of two possible messages to the host device 17 in the event that an error occurs data retrieval. The first message is TAPE_CORRUPT (which replaces both KEY_CORRECT and KEY_CORRUPT) and is generated should the tape drive 1 determine that the

encryption key stored in key memory 21 is correct or the encryption key retrieved from tape is corrupt). The second message is KEY_INCORRECT and is generated, as described above, should the tape drive 1 determine that the encryption key stored in key memory 21 is incorrect. Consequently, the tape drive 1, in response to receiving a read command, returns to the host device 17 the requested data, a TAPE_CORRUPT message, or a KEY_INCORRECT message.

In the first of the two embodiments described above, the tape drive 1 receives an encrypt data record comprising a self-encrypted encryption key. However, rather than receiving an encrypt data record, the tape drive 1 may alternatively create the encrypt data record whenever a new write request or a new encryption key is received.

In the second of the two embodiments described above, a copy of the self-encrypted encryption key is written to the DSIT of each data set. However, a single copy of the self-encrypted encryption key may alternatively be written to the tape cartridge, e.g. written to the tape header portion of the tape or to a cartridge memory. In this alternative embodiment, all data stored to the tape cartridge are encrypted using the same encryption key. In order to prevent data encrypted using different keys from being stored to a particular tape cartridge, the tape drive 1 prevents the encryption key stored in key memory 21 from being changed until such time as a new tape cartridge has been inserted, or until the contents of the tape cartridge have been erased

Rather than creating a self-encrypted copy of the encryption key stored in key memory 21 every time the encryption key is to be verified, the controller 3 optionally stores a self-encrypted copy of the encryption key in the memory buffer 5 (or some other memory location) for subsequent use by the controller 3. Whenever a new encryption key is received by the tape drive 1, the controller 3 causes the new encryption key to be stored in the key memory 21.

Additionally, the controller 3 causes the encryption key to be encrypted by the encryption engine 20, and the resulting self-encrypted encryption key is then stored to the memory buffer 5 (or some other memory location) for subsequent use.

In the embodiments described above, a self-encrypted encryption key is retrieved from tape and compared against a self-encrypted copy of the encryption key stored in key memory 21. If the two self-encrypted encryption keys correspond, a positive verification is made that the encryption key stored in key memory 21 is the correct encryption key for the data being retrieved. In an alternative embodiment, the self-encrypted encryption key retrieved from tape is decrypted by the encryption engine 20, using the encryption key stored in key memory 21. The resulting decrypted key is then compared against the encryption key stored in key memory 21. If the two keys correspond (i.e. if the decrypted key retrieved from tape corresponds to the encryption key stored in key memory 21), a positive verification is made that the encryption key stored in key memory 21 is the correct encryption key for the data being retrieved. In this alternative embodiment, the encrypt data record stores a copy of the encryption key rather than a self-encrypted copy of the encryption key.

It is assumed that the encryption key stored in key memory 21 is the correct encryption key for the data being retrieved. Accordingly, a comparison of the encryption key stored on tape with that stored in key memory 21 is made only in the event that an error is generated during data retrieval, e.g. when an error is generated by the data compressor 8. This has the advantage that data may be retrieved from tape even if the self-encrypted encryption key stored on tape is corrupt. Nevertheless, in an alternative embodiment, a comparison of the key stored on tape is made with the key stored in key memory 21 prior to the retrieval of data records. In this alternative embodiment, the tape drive 1 retrieves the data set(s) of the requested data record(s) only in the event that the two keys correspond.

In the embodiments described above, a self-encrypted copy of an encryption key is stored to tape. Since the encryption key serves as both the plaintext and encryption key, it is extremely difficult for a third party to extract the encryption key from the ciphertext. In an alternative embodiment, rather than storing an encrypted copy of the encryption key, an encrypted copy of predetermined reference data is instead stored to tape. In order to verify that that the encryption key stored in key memory 21 is the correct key for the data being retrieved, the controller 3 encrypts the reference data using the encryption engine 20 and compares the result with the encrypted reference data stored on tape. The reference data is not intended to be kept secret and it is assumed to be publicly available. For example, the reference data may be derived from the serial number of the tape cartridge onto which data are to be stored, or it may be a fixed value stored within the tape drive 1 on non-volatile memory. Since the plaintext in this alternative embodiment is now known, the encryption algorithm employed by the encryption engine 20 creates the ciphertext (i.e an encrypted copy of the reference data) in a manner which does not permit the encryption key to be easily recovered. A suitable algorithm includes GCM encryption.

The tape drive 1 may be regarded as involving two formatting steps. In the first step, records received by the tape drive 1 are compressed and then encrypted to create pseudo-records In the second step, the pseudo-records are subjected to conventional LTO formatting, i.e. the pseudo-records are protected, compressed using an LTO scheme, and packed together to form a compressed data stream The tape drive 1 may therefore be regarded as converting records into encrypted pseudo-records which are then formatted by the tape drive 1 using conventional LTO formatting.

By creating pseudo-records, which are then formatted using conventional LTO formatting, data sets stored to tape by the tape drive 1 can be read back using

conventional LTO tape drives, i.e. LTO tape drives not having means to encrypt or decrypt data. When a particular record is requested by a host device, a conventional LTO tape drive will locate and retrieve the relevant data set of group of data sets from the tape. The retrieved data set(s) is then formatted in a conventional manner by the LTO tape drive to extract one or more pseudo-records, each pseudo-record comprising an encrypted record. The pseudo-records are then delivered to the host device 17, whereupon they can be decrypted using software resident on the host device 17. The tape drive 1 therefore has the very real benefit that data stored to tape by the tape drive 1 are encrypted and yet can nevertheless be read back by conventional tape drives and decrypted using software resident on a host device.

The tape drive 1 may optionally deliver pseudo-records to the host device 17 should the tape drive 1 determine that the encryption key stored in key memory 21 is unsuitable for decrypting the pseudo-records.

Additionally, the tape drive 1 may optionally include a bypass (see Figure 1) such that the data encryptor 9 is ignored by the tape drive 1 during data write or data read. Bypass of the data encryptor 9 may occur should no encryption key be stored in key memory 21, or if the controller 3 receives a command from the host device 17 to bypass encryption. With the exception of the data encryptor 9, the components of the tape drive 1 are identical to those of a conventional LTO tape drive. In particular, the data compressor 8 of the tape drive 1 employs an LTO compression scheme. Consequently, when the data encryptor 9 is bypassed, the tape drive 1 functions as a conventional LTO tape drive and records to be stored and/or retrieved are formatted using conventional LTO formatting.

Whilst the data encryptor 9 employs a Galois Counter Mode encryption algorithm, other encryption algorithms may alternatively be employed, including block cipher, stream cipher, symmetric and asymmetric encryption. In the case

of asymmetric encryption, the key memory 21 optionally stores a decryption key in addition to an encryption key.

Although an embodiment of the present invention has been described with reference to the LTO format, the present invention is equally applicable to other data formats, particularly those data formats in which data to be stored are received as records. In particular, the pseudo-records created by the encryption engine 20 can be formatted as conventional records using alternative tape formats, such as DDS (including DAT 72 and DAT 160), SDLT, DLT and proprietary IBM formats. By using conventional tape formatting to format and write the pseudo-records to tape, data stored to tape by the tape drive 1 can be read back using conventional tape drives.

Moreover, although embodiments of the present invention has been described with reference to a tape drive 1, it will be appreciated that the present invention is equally applicable to other types of data transfer devices including, but not limited to, optical drives.

With the data transfer device embodying the present invention, the encryption and decryption of backup data is moved from the host device to the data transfer device. Moreover, by storing an encrypted copy of the encryption key along with any data encrypted using the encryption key, the data transfer device is able to reliably determine the cause of any error that might occur during data retrieval.

When used in this specification and claims, the terms "comprises" and "comprising" and variations thereof mean that the specified features, steps or integers are included. The terms are not to be interpreted to exclude the presence of other features, steps or components.

The features disclosed in the foregoing description, or the following claims, or

the accompanying drawings, expressed in their specific forms or in terms of a means for performing the disclosed function, or a method or process for attaining the disclosed result, as appropriate, may, separately, or in any combination of such features, be utilised for realising the invention in diverse forms thereof.

CLAIMS

What is claimed is:

1. A data transfer device for transferring data to a removable data storage item, the data transfer device being operable to:

receive data to be stored to the removable data storage item;

encrypt the data using an encryption key;

encrypt predetermined reference data using the encryption key; and

store the encrypted data and the encrypted reference data to the removable data storage item.

2. A data transfer device according to claim 1, wherein the reference data comprises a copy of the encryption key.

3. A data transfer device according to claim 1 or 2, wherein the data transfer device is operable to store the encrypted data as one or more data blocks, and to store a copy of the encrypted reference data with each data block.

4 A data transfer device according to claim 3, wherein each data block comprises a data region and an information table, and a copy of the encrypted reference data is stored in the information table.

5. A data transfer device according to any preceding claim, wherein the data transfer device is operable to store the encrypted data in a format selected from one of the generations of LTO and DDS/DAT formats.

6. A data transfer device according to any preceding claim, wherein the data is received as one or more records and a copy of the reference data is provided within a record, and the data transfer device is operable to encrypt the

records, including the record comprising the reference data, using the encryption key and to store the encrypted records to the removable data storage item

7. A data transfer device according to any preceding claim, wherein the data transfer device is operable to compress the data prior to encryption.

8. A data transfer device according to any preceding claim, wherein the data transfer device stores a copy of the encryption key and is suitable for transferring data from the removable data storage item, the data transfer device being operable to:

retrieve the encrypted data from the removable data storage item;

decrypt the encrypted data using the encryption key stored by the data transfer device; and

output the decrypted data

9. A data transfer device according to claim 8, wherein the data transfer device additionally stores a copy of the predetermined reference data and is operable to.

retrieve the encrypted reference data from the removable data storage item;

compare the reference data retrieved from the removable data storage item with the reference data stored by the data transfer device; and

generate an error message in the event that the reference data retrieved from the removable data storage item does not substantially correspond with the reference data stored by the data transfer device.

10. A data transfer device according to claim 9, wherein retrieving the encrypted reference data, comparing the reference data and generating an error message are performed in the event that the encrypted data cannot be

successfully decrypted using the encryption key stored by the data transfer device

11. A data transfer device according to claim 9 or 10, wherein the data transfer device in comparing reference data is operable to:

encrypt the reference data stored by the data transfer device using the encryption key; and

compare the encrypted reference data retrieved from the removable data storage item with the encrypted reference data created by the data transfer device.

12. A data transfer device according to claim 9 or 10, wherein the data transfer device in comparing reference data is operable to:

decrypt the encrypted reference data using the encryption key stored by the data transfer device; and

compare the decrypted reference data with the reference data stored by the data transfer device.

13. A data transfer device according to any one of claims 9 to 11, wherein the data transfer device in comparing reference data is operable to determine the number of locations at which the reference data retrieved from the removable data storage item and the reference data stored by the data transfer device differ, and the data transfer device is operable to generate the error message in the event that the number of locations is greater than a predetermined value.

14. A data transfer device according to any preceding claim, wherein the data transfer device is a tape drive and the removable data storage item is a tape cartridge.

15. A data transfer device for transferring data to a removable data storage item, the data transfer device comprising:

means for receiving data to be stored to the removable data storage item;

means for encrypting the data using an encryption key;

means for encrypting predetermined reference data using the encryption key; and

means for storing the encrypted data and the encrypted reference data to the removable data storage item

16 A data transfer device according to claim 15, wherein the reference data comprises a copy of the encryption key.

17. A data transfer device according to claim 15 or 16, wherein the data transfer device comprises:

means for storing a copy of the reference data;

means for retrieving the encrypted reference data from the removable data storage item;

means for comparing the reference data retrieved from the removable data storage item with the reference data stored in the means for storing; and

means for generating an error message in the event that the reference data retrieved from the removable data storage item does not substantially correspond with the reference data stored in the means for storing.

18. A method of operating a data transfer device for transferring data to a removable data storage item, the method comprising:

receiving data to be stored to the removable data storage item;

encrypting the data using an encryption key;

encrypting predetermined reference data using the encryption key; and

storing the encrypted data and the encrypted reference data to the removable data storage item.

19 A method according to claim 18, wherein the reference data comprises a copy of the encryption key.

20. A method according to claim 18 or 19, wherein the method further comprises:

storing a copy of the reference data in memory;

retrieving the encrypted reference data from the removable data storage item;

comparing the reference data retrieved from the removable data storage item with the reference data stored in the memory; and

generating an error message in the event that the reference data retrieved from the removable data storage item does not substantially correspond with the reference data stored in the memory.

Amendments to the Claims have been filed as follows

What is claimed is:

1. A data transfer device for transferring data to a removable data storage item, the data transfer device being operable to:

receive data to be stored to the removable data storage item;

encrypt the data using an encryption key;

encrypt predetermined reference data using the encryption key; and

store the encrypted data and the encrypted reference data to the removable data storage item.

2. A data transfer device according to claim 1, wherein the reference data comprises a copy of the encryption key.

3. A data transfer device according to claim 1 or 2, wherein the data transfer device is operable to store the encrypted data as one or more data blocks, and to store a copy of the encrypted reference data with each data block.

4. A data transfer device according to claim 3, wherein each data block comprises a data region and an information table, and a copy of the encrypted reference data is stored in the information table.

5. A data transfer device according to any preceding claim, wherein the data transfer device is operable to store the encrypted data in a format selected from one of the generations of LTO and DDS/DAT formats.

6. A data transfer device according to any preceding claim, wherein the data is received as one or more records and a copy of the reference data is provided within a record, and the data transfer device is operable to encrypt the records,

including the record comprising the reference data, using the encryption key and to store the encrypted records to the removable data storage item.

7. A data transfer device according to any preceding claim, wherein the data transfer device is operable to compress the data prior to encryption.

8. A data transfer device according to any preceding claim, wherein the data transfer device stores a copy of the encryption key and is suitable for transferring data from the removable data storage item, the data transfer device being operable to:

retrieve the encrypted data from the removable data storage item;

decrypt the encrypted data using the encryption key stored by the data transfer device; and

output the decrypted data.

9. A data transfer device according to claim 8, wherein the data transfer device additionally stores a copy of the predetermined reference data and is operable to:

retrieve the encrypted reference data from the removable data storage item;

compare the reference data retrieved from the removable data storage item with the reference data stored by the data transfer device; and

generate an error message in the event that the reference data retrieved from the removable data storage item does not substantially correspond with the reference data stored by the data transfer device.

10. A data transfer device according to claim 9, wherein retrieving the encrypted reference data, comparing the reference data and generating an error message are performed in the event that the encrypted data cannot be successfully decrypted using the encryption key stored by the data transfer device.

11. A data transfer device according to claim 9 or 10, wherein the data transfer device in comparing reference data is operable to:

encrypt the reference data stored by the data transfer device using the encryption key; and

compare the encrypted reference data retrieved from the removable data storage item with the encrypted reference data created by the data transfer device.

12. A data transfer device according to claim 9 or 10, wherein the data transfer device in comparing reference data is operable to:

decrypt the encrypted reference data using the encryption key stored by the data transfer device; and

compare the decrypted reference data with the reference data stored by the data transfer device.

13. A data transfer device according to any one of claims 9 to 11, wherein the data transfer device in comparing reference data is operable to determine the number of locations at which the reference data retrieved from the removable data storage item and the reference data stored by the data transfer device differ, and the data transfer device is operable to generate the error message in the event that the number of locations is greater than a predetermined value.

14. A data transfer device according to any preceding claim, wherein the data transfer device is a tape drive and the removable data storage item is a tape cartridge.

15. A data transfer device according to any preceding claim, wherein the predetermined reference data is derived from the serial number of the removable data storage item onto which data are to be stored.

16. A data transfer device according to any preceding claim, wherein the predetermined reference data is a fixed value stored within the data transfer device in memory.

17. A method of operating a data transfer device for transferring data to a removable data storage item, the method comprising:

receiving data to be stored to the removable data storage item;

encrypting the data using an encryption key;

encrypting predetermined reference data using the encryption key; and

storing the encrypted data and the encrypted reference data to the removable data storage item.

18. A method according to claim 17, wherein the reference data comprises a copy of the encryption key.

19. A method according to claim 17 or 18, wherein the method further comprises:

storing a copy of the reference data in memory;

retrieving the encrypted reference data from the removable data storage item;

comparing the reference data retrieved from the removable data storage item with the reference data stored in the memory; and

generating an error message in the event that the reference data retrieved from the removable data storage item does not substantially correspond with the reference data stored in the memory.

The Patent Office

For Innovation

33

Application No: GB0602012.7 **Examiner:** Mr Adam Tucker

Claims searched: 1-20 **Date of search:** 17 May 2006

Patents Act 1977: Search Report under Section 17

Documents considered to be relevant:

Category	Relevant to claims	Identity of document and passage or figure of particular relevance
X	1-20	EP 1020856 A2 (Yamaha) See in particular the abstract and paragraphs 1 & 8
X	1-20	US 2003/0074319 A1 (Jacquette) See the whole document and in particular paragraphs 6, 7, 64, 74-78 & 82
X	1-20	EP 1440439 A1 (Philips Electronics) See in particular the claims, page 1, page 4 lines 15-19 and page 6 lines 15-17
X	1, 15 & 18 at least	EP 1615368 A1 (Sony Corp.) See in particular and at least claims 9-11 and paragraph 64
X	1, 15 & 18 at least	US 2005/0278257 A1 (Barr et al.) See the whole document and in particular the abstract and claims and paragraphs 12, 22 & 25

Categories:

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

Field of Search:

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC^{X} :

G4A; G5R; H4P

Worldwide search of patent documents classified in the following areas of the IPC

G06F; G11B; H04L

The following online and other databases have been used in the preparation of this search report

WPI, EPODOC