Description

This invention relates to reliability enhancement of tracked non-volatile data storage devices.

Tracked data storage devices, such as direct access data storage devices - DASD - in which binary data is written or stored on non volatile recording media such as magnetic or optical disks, are well known in the art, and thus no attempt will be made to describe these devices in detail. It suffices to say that in such devices, and particularly in magnetic disk devices, the ability to properly record or write certain stress sequences of binary "1's" and/or "0's" is a function of the disk-to-head interface and the magnetic recording phenomenon. When an attempt is made to record such stress patterns, the linear track position of the recorded data (i.e. the S-S or N-N magnetic transitions along the track) may shift from the desired position, and as a result, subsequent reading of the data may produce a read-error(s). In addition, it is known that error correction codes are vulnerable to certain data patterns that stress the ability of the ECC to identify and/or correct read-errors. One example of a read stress pattern is a one reoccurring sequence of equal (i.e. repeating) magnitude read-errors.

The present invention relates to the field of tracked storage devices, and provides a scheme for randomizing binary data prior to recording, and for derandomizing the randomized data in a complementary (i.e. similar) fashion upon read back, such that write-patterns that stress the head/media interface are avoided,and such that read-patterns that stress the ECC are avoided.

It is pointed out that the secure storage of encrypted data and the use of interleaved error and PRML codes are known and proposals have been made to spread the mapping of logical word addresses onto real word locations in a volatile chip memory and to allocate each bit of each word of a block of words to a different chip/ROM in an orderly manner - the nth bit of the "nth" chip/ROM.

GB 2017365 and "An Experimental Digital Videotape Recorder" by K Yokoyama, S Nakagawa, and H Katayama, the BKSTS Journal, Nov 1980, p570-580 describe a digital magnetic tape recording system with error correction and modulo-2 addition/subtraction of a random bit stream to/from the data for scrambling/descrambling respectively.

Each context has its own problems and to the knowledge of the present inventors, the concept of randomizing error coded binary data prior to storing the data on disk storage devices avoiding data patterns that stress the head/disk interface and the error code is not to be found in the prior art. In addition, it is believed that the concept of, at a later and a random time (i.e. not real time, as in a data transmission system), reading back the randomized data from the disk, and then derandomizing the data prior to presenting the derandomized data to an ECC network, is not to be found in the prior art.

The use of scrambler/descrambler schemes in real time transmission path digital communication systems for security purposes is known. US-A-4,639,548 is an example. In this patent, an input data signal is scrambled and then applied to a convolutional encoder. The output of the encoder is then applied to a transmission path. The data transmitted through the transmission path to a receiver is applied first to a convolutional encoder, and then to a descrambler. The descrambler is synchronised to the scrambler by the use of an error correction pulse signal that is generated at the location of the descrambler.

US-A-3,775,746 also describes the use of a scrambler/descrambler scheme in connection with a real time digital data transmission system. In this system, the scrambler/descrambler function is provided by a network that includes a delay circuit and an exclusive-or gate. Synchronisation and clocking are not described in this patent.

US-A-3,771,126 describes a data processing and transmission system wherein an analog signal is sampled and encoded by an encoder. A parity generator then adds parity bits to allow for the correction of errors in each block of digital data. The digital data and the parity bits are then converted into a random sequence of pulses by a scrambler, prior to being applied, real time, to a channel. At the output of the channel, the data is descrambled, applied to an error decoder, and then reconstructed into an analog signal, of for use by a user circuit. The scrambler is a conventional self-synchronising scrambler, for example one cell scrambler/descrambler comprising a modulo 2 adder and a one cell delay.

US-A-3,988,538 describes scramblers and descramblers for use in digital data transmission system having a plurality of parallel synchronised data streams. The complementary scramblers/descramblers of this patent are each made up of a series of delay elements (i.e. a shift register) comprising a linear sequential filter having a series of taps (i.e. a tapped delay line) that are summed and then connected in feedback/feedforward fashion to the input of the scrambler/descrambler.

These patents are exemplary of the prior art in which data is scrambled from transmission in real time to a receiver having a descrambler but do not address the problems of using a complementary randomizer/derandomizer combination in a disk data storage device.

The invention provides a method of writing data onto a disk data storage device, the data being written

onto a data storage disk as a series of data fields, each data field including user data and corresponding error correcting code (ECC) data, the method comprising the steps of:

generating ECC data for the user data to produce a string of data words representing the user data and corresponding ECC data for a data field;

and characterised by then randomising said ECC and user data within the data field by multiplying or dividing each data word by data from a sequence of pseudo random data.

The invention also provides a method of reading data from a disk data storage device, the data being stored on the data storage disk as a series of data fields, each data field including user data and corresponding error correcting code (ECC) data, the method comprising the steps of:

reading a data field as a string of data words from the disk and using the user data and corresponding ECC data read back from the data field to identify and correct any errors;

and characterised by the step of derandomizing said ECC and user data within the data field by multiplying or dividing each data word by data from a sequence of pseudo random data, said derandomizing occurring prior to using the user data and corresponding ECC data read back from the data field to identify and correct any errors.

The present invention further provides apparatus able to adopt such read-write methods.

As disclosed hereinafter, randomized data is recorded or stored on a non volatile medium such as a magnetic disk. The DASD device is then used to read back the randomized data to a read channel having a complementary derandomizer. The DASD randomizer/derandomizer combination functions to avoid the writing of data patterns that stress the ability to store the randomized data, and also functions to avoid read-error conditions that stress the ability of the read channel's error correcting code (ECC) to identify and correct errors that may occur when the stored, randomized, data is subsequently read back from the disk.

Binary data, consisting of user data and corresponding ECC data, is randomized prior to being written on non volatile memory such as the magnetic disk of a DASD device, and is derandomized when the data is subsequently read back off of the disk. More specifically, user data and corresponding ECC data is first randomized, then run-length encoded, and then written on the disk by the use of a DASD write channel.

Subsequently, when the randomized and encoded data records are read back off of the disk by the use of the DASD read channel, the records are first decoded and then derandomized in a complementary fashion in order to recover the original user data and corresponding ECC data.

The derandomizer of the read channel is placed intermediate the decoder and the ECC network. In this way, repeating, constant magnitude, read-errors are made to appear as random errors, thus minimising stress to the ECC network's ability to detect and correct these constant magnitude read-errors.

The scheme for randomizing the binary data (i.e. the user data and corresponding ECC data) is constructed and arranged to avoid the writing of data patterns that may stress the ability of the head/disk interface to record the magnetic transitions at the proper linear track 0positions on the disk. Later, when the disk is read, the complementary derandomizing scheme avoids presenting a repeating pattern of read-errors to the read channel's ECC network. Thus, read-errors are not propagated through the read channel.

The write channel randomizer, and the read channel derandomizer, may take the form of a finite field data multiplier and divider, respectively, both of which use the same type of a pseudo random polynomial counter. In the write channel's randomizer, the data is multiplied by the content of a polynomial counter. In the read channel's derandomizer, the randomized data is divided by the content of a polynomial counter, in order to divide out the multiplication effect of the write channel's randomizer.

In accordance with a specific embodiment of the invention, the write channel randomizer and the read channel derandomizer can comprise an 8-bit pseudo random counter(s). When the write/read data consists of a string of 8-bit data words, the 8-bit pseudo random counter can be advanced so as to pass through its 255 distinct and different states (the all-0s state is not used) in a predefined and repeating pseudo random fashion, that is, always in the same known, repeating and random sequence, under the control of a byte counter that advances in response to the sequential writing/reading of the multi-byte data being written to the disk or being read off of the disk.

In a less costly version, the counter comprises an 8-bit pseudo random polynomial counter that is split into two independent 4-bit pseudo random registers or counters (see FIG. 3).

During the writing of data to the disk, the upper four bits (i.e. bits P0-P3) of the counter (i.e. one of the 4-bit registers) are advanced with exclusive-OR feedback that is derived from a first four-bit polynomial, herein called the "forward polynomial", and the lower four bits (i.e. bits P4-P7) of the counter (i.e. the other of the 4-bit registers) are advanced with exclusive-OR feedback that is derived from a second four-bit polynomial, herein called the "reverse polynomial".

During the writing of data to the disk, the upper four bits (i.e bits P0-P3) of the counter randomly passes through 15 different counting states (the state "0000" is not used) in a known, repeating and pseudo random

sequence. The lower four bits (i.e bits P4-P7), however, pass through only 13 of the different counting states.

Subsequently, during the reading of the randomized data, the original data is reproduced by dividing out the multiplication effect of the write channel randomizer. In this case, finite field division is accomplished by a technique known as multiplication by the inverse.

During the reading of data from the disk, the upper four bits (i.e. bits P0-P3) of the counter are again advanced with exclusive-OR feedback that is derived from the above mentioned forward polynomial, and the lower four bits (i.e. bits P4-P7) of the counter are again advanced with exclusive-OR feedback that is derived from the above mentioned reverse polynomial.

During the reading of data from the disk, these upper four bits of the counter (i.e bits P0-P3) pass through only the above mentioned 13 different counting states. The lower four bits of the counter (i.e bits P4-P7), however, pass through the above mentioned 15 different counting states.

Thus, for both the writing of data and the subsequent reading of data, the repetitive counting cycle of the 8-bit polynomial counter consists of 15-times-13 byte cycles, i.e. 195 consecutive data bytes cycles or times.

In summary, in this preferred embodiment data is written to the disk in a randomized form that is provided by the sequential counting of the 8-bit pseudo random polynomial counter, the counter operating to repeat its counting sequence during each sequential field of 195 data bytes being written to the disk - and - the randomized data is subsequently read back from the disk as the 8-bit polynomial counter within the read channel's derandomizer operates to divide out the data-multiplication that was effected during an earlier write operation by the polynomial counter within the write channel's randomizer. Here again, the counter operates to repeat its counting sequence during each sequential field of 195 data bytes being read from the disk.

As seen in FIG. 3, a single polynomial counter 40 can function in both the write channel and the read channel.

The forward and reverse polynomials, as above defined, can be chosen so that the binary content of one 4-bit portion of the counter is the inverse of the binary content of the other 4-bit portion of the counter.

Specifically, the "forward polynomial" is selected to be $g(x) = x^4 + x + 1$, and the "reverse polynomial" the is selected to be $g(x) = x^4 + x^3 + 1$.

These two polynomials are chosen so that if the particular polynomial word of the counter (i.e. bits P0-P7) that was used to randomize a given data byte is reversed, and then used when the randomized byte is subsequently read back, the word-content of the counter (i.e. bits P0-P7) during reading is the inverse of the content of the word-content when writing, and in this way the randomized byte is divided by the original multiplier, and the byte is thus derandomized.

By choosing these two polynomials in this manner, the same counter construction and arrangement can be used for both the write channel's randomizer and the read channel's derandomizer.

In the preferred embodiment of FIG. 3, the procedure for randomizing a write-data byte requires that counter 40 be initialised or seeded to a specific state, in this example to P0-P7 = "00010001". The upper four bits (i.e. CW0-CW3) of write data are now multiplied by the upper four bits (i.e. P0-P3) of the counter, and the lower four bits (i.e. CW4-CW7) of write data are now multiplied by the lower four bits (i.e. P4-P7) of the counter. The counter is advanced to its next randomizing count state, and the above mentioned multiplication repeats for the next write-data byte, using the next pseudo random state of the counter. When the next write field (see 11 of FIG. 1) arrives, the counter is again initialised to "00010001", and the process repeats for this next write-data field.

The procedure for derandomizing a randomized read-data field requires that the counter be initialised or seeded to the state P0-P7 = "10001000" for this example. The upper four bits (i.e. R0-R3) of the read-back data are now multiplied by the inverse order of the lower four bits (i.e. P4-P7) of the counter, and the lower four bits (i.e. R4-R7) of the read-back data are now multiplied by the inverse order of the upper four bits (i.e. P0-P3) of the counter. The counter is then advanced to its next count state for the next read-back data byte, and the above mentioned multiplication-by-the-inverse process repeats. When the next read-back field arrives, the counter is again initialised to "10001000", and the process repeats for this read-back data field.

The present invention will be described further by way of example with reference to embodiments thereof as illustrated in the accompanying drawings, in which:

FIG. 1 shows a representative binary data field of the type that is to be randomized, encoded and then recorded on magnetic media by the use of the invention, and which is subsequently read back, decoded and derandomized by the use of the invention,

FIG. 2 shows the write-channel/read-channel portion of a direct access storage device that is constructed in accordance with the invention,

FIG. 3 shows one embodiment of a randomizer/derandomizer for use in the device of FIG. 2, and

FIGS. 4, 5 and 6 are used to describe the operation of the randomizer/derandomizer of FIG. 3.

The following examples of embodiments of the invention will be described without detailed reference being

made to the construction and arrangement details of a DASD disk file. Devices of this type are well known to those skilled in the art. Since the present invention comprises an improvement(s) to the write-channel/read-channel of these devices, only that portion of a DASD device will be described.

With reference to FIG. 1, the binary data represented therein is to be linearly written to a track(s) of a magnetic recording disk (not shown), after the data has been both randomized and encoded.

The binary data that is read back off of the disk is ideally identical to the written data, and is thus also represented as being identical to that of FIG. 1.

FIG. 1 comprises an exemplary linear disk track 10 having a serial string of individual data fields 11, each individual field of which is here shown as comprising an exemplary 301 bytes. Each data field 11 begins with a start or synchronising area 12 which is, for example, 12 bytes long. Area 12 includes a synchronisation signal that identifies the beginning of each individual data field 11 within track 10. Area 12 also includes other information, such as servo IDs, etc., as is well known to those of skill in the art.

It should be noted that area 12 is not randomized/derandomized by operation of the invention, i.e. area 12 is encoded and then recorded in its encoded form. Subsequently, when a data field 11 is read back from the disk, the not-randomized area 12 is decoded and then detected. This detection event is used to synchronise the read channel's derandomizer 34 (FIG. 2), as will be explained.

The next portion of each data field 11 comprises an area 13 which is, for example, 256 bytes long. This portion 13 of data field 11 contains the variable user information or data that is to be recorded or stored by the DASD device on its non volatile disk memory. The present invention is of particular utility when this user data portion 13 comprises a number of identical repeating bytes, such as the byte "11111111" for example that is shown in the following Table A. Repeating data of this general type is often found in user data that defines a graphic image. It has been found that these repeating byte patterns tend to stress both the magnetic recording phenomenon and conventional ECC algorithms.

The DASD device includes an error correction code (ECC) network 25 (FIG. 2) that analyses the user information 13 to be recorded. This ECC network generates the binary content of an ECC area 14, which is for example 10 bytes long. ECC area 14 is also randomized, encoded, and recorded. Area 14 will be used by the read channel's ECC network 36 (FIG. 2). after decoding and derandomizing, to identify read-errors, and to recover from any such errors as may result when reading recorded data from the disk.

As will be apparent, only areas 13 and 14 are randomized in accordance with the invention.

As is well known to those of skill in the art, certain binary sequences or patterns are to be avoided when writing binary data to a magnetic recording disk, since these binary patterns stress the ability of the head/disk interface to accurately position the magnetic transitions at the desired positions linearly along the track surface of the disk.

In addition, certain stressful read-back data bit patterns appear to cause higher error rates than do random read-back data patterns. If these troublesome read stress patterns are frequently repeated, the read-errors that do occur tend to have equal magnitudes, i.e. each byte that is in error miscompares with the original write data byte in the same bit position for each of the error repetitions. This type of equal-magnitude error condition causes a higher susceptibility to data miscorrection for some ECC codes. That is, certain sequences of detected errors that occur when reading data from the disk stress the ability of the ECC to identify and/or recover from these read-error conditions. In fact, read-error conditions may remain undetected/corrected as a result of reading patterns that stress the ECC.

The present invention provides data randomizing prior to the writing of data areas 13,14 to the disk, and provides similar, complementary, data derandomizing upon subsequently reading the randomized data areas 13,14 from the disk, so as to avoid both magnetic stress conditions and ECC stress conditions. The benefit of the invention is two-fold. First, repeating bytes of customer data are made to appear as random data prior to being written on the disk. Second, even if equal magnitude errors (i.e. multiple errors that are equally bit-spaced) do occur when subsequently reading the randomized data, these errors appear random to the read channel's ECC network, and thus a lower miscorrection rate is achieved.

FIG. 2 is a simplified diagram showing the read and the write channels of a DASD device constructed in accordance with the invention. In this figure the transducing interface that exists between a read/write transducing head(s) and the disk(s) is shown at 20.

The user data portion 13 (i.e. the user data in its not-encoded and not-randomized form) of the exemplary data track of FIG. 1 is present to the write channel by way of conductor 23 during a write operation. This identical data portion 13 will be present on the read channel's output conductor 24 during a subsequent read operation, i.e., the user data 13 that is to be recorded on the disk is presented to conductor 23 during a write operation, and this same user data will be subsequently recovered from conductor 24 during a read operation.

The sync field area 12 of the FIG. 1 track is presented to the read channel by way of conductor or bus 29, and is also present on read channel conductor or bus 39 during a subsequent read operation.

As is well known to those of skill in the art, one of the functions provided in a DASD device is the generation of the FIG. 1 ECC binary data field 14 that is uniquely related to the binary content of the user data portion 13 that exists on conductor 23. This function is provided in FIG. 2 by operation of ECC network 25. The output of network 25 (which includes the operation of other well known circuit means, not shown) comprises an 8-bit wide bus 26, containing a series of 8-bit words (bytes) whose bits are identified for purposes of explanation as bits CW0 through CW7.

In accordance with the invention, each such data byte (i.e. each word comprising bits CW0-CW7) is randomized by the operation of pseudo random randomizing network means 27. The randomized data is then presented to 8-bit bus 28 as a series of randomized bytes, each byte having bits identified for purposes of explanation as W0 through W7.

As is well known to those skilled in the art, a synchronisation data field (see area 12 of FIG. 1) is presented to the write channel of FIG. 2 in order to mark and identify the beginning of each field 11 of disk track 10. This sync information is presented to the write channel by way of conductor 29. In addition to being encoded and then recorded in its encoded form on the disk track, this sync information operates to initialise the operation of randomizer 27 at the beginning of each data field 11 of track 10 (for example, sync field 12 operates to provide a starting seed for the pseudo random polynomial counter 40 of FIG. 3).

In accordance with standard practice, the output of randomizer 27 is combined with the sync information on conductor 29, and both are then subjected to the operation of a run-length-limited encoder 30, for example a 1,7 encoder.

The final operation of the FIG. 2 write channel, which involves the operation of other well known circuit means (not shown), is to present a randomized/encoded serial binary data byte stream to conductor 21, for recording on the disk track(s) in the form of S-S and N-N magnetic transitions.

Later, when reading the randomized/encoded data back from the disk, a serial byte stream is provided to conductor 22. This byte stream is first decoded by operation of a complementary decoder 31.

By virtue of the operation of decoder 31, and by virtue of the operation of other conventional circuit means not shown, the decoded byte stream is provided to 8-bit bus 32 and to sync detector 33 in the form that is seen in FIG. 1. The signal on conductor 39 includes the sync field portion 12, which portion is detected by sync detector network 33. Network 33 operates to synchronise and begin the operation of complementary pseudo random derandomizer 34 at the beginning of each data field 11 (for example, sync field 12 operates to provide a starting seed for the pseudo random polynomial counter 40 of FIG. 3).

For purposes of explanation, the 8 bits presented as an input to complementary derandomizer 34 are identified as bits R0 through R7, and the corresponding derandomized output bits which appear on output 8-bit bus 35 of derandomizer 34 are identified as bits CR0-CR7. As will be apparent, the derandomized data bytes represented by bits CR0-CR7 are identical to the data bytes originally presented to the input of randomizer 27, which original data bytes are represented in the figure as having bits CW0-CW7.

The output of the FIG. 2 read channel comprises complementary ECC network 36, and other well known circuit means not shown. By way of ECC network 36, the data 13 originally presented to conductor 23 is recovered from the read channel's output conductor 24.

FIG. 3 shows a preferred embodiment of the invention wherein the above mentioned pseudo random polynomial counter comprises an 8-bit counter 40 that is split into two registers or counters comprising bits P0-P3 and bits P4-P7, respectively. In this embodiment, counter 40 is shared by the write channel randomizer 27 and the read channel derandomizer 34 of FIG. 1.

Polynomial counter 40 consists of eight latches, having next-count-state logic. The next-state chart and the next-state logic equations for the forward polynomial portion P0-P3 of the counter is given in FIG. 4. The next-state chart and the next-state logic equations for the reverse polynomial portion P4-P7 of the counter is given in FIG. 5.

For example, consider FIGS. 4 and 5 for a write operation, and assume that a write data byte M is currently being randomized. For this explanation, the actual binary content of data byte M is not identified since it is not critical to the explanation. In this case, the current content of the polynomial counter is "00111001". That is, the current content of the forward polynomial counter bits P0-P3 is "0011", as is shown in the byte M position of the "Present State" column of FIG. 4, and the current content of the reverse polynomial counter bits P4-P7 is "1001", as is shown in the byte M position of the "Present State" column of FIG. 5.

For this assumed write operation, the next state of the forward polynomial counter bits P0-P3 is identified in the "Write" column of FIG. 4 as "0110", as is shown in the byte M position of that column, and the next state of the reverse polynomial counter bits P4-P7 is identified in the "Write" column of FIG. 5 as "1011" in the byte M position of that column. Note that in this case, the next state conditions correspond to the entry in the "Present State" column that follows the identified byte M position. This characteristic of FIGS. 4 and 5 is not critical however.

Now assume that a read operation is currently underway and that read-back data byte N is currently being derandomized. Again, the actual data content of byte N is not shown, nor is it critical. In this case, the current content of forward polynomial counter bits P0-P3 is "1110", as is identified at the byte N position in the "Present State" column of FIG. 4, and the current content of reverse polynomial counter bits P4-P7 is "1101", as is identified at the byte N position in the "Present State" column of FIG. 5.

For this assumed read operation, the next state of the forward polynomial counter bits P0-P3 is identified in the "Read" column of FIG. 4 as "1111", as is shown in the byte N position of that column, and the next state of the reverse polynomial counter bits P4-P7 is identified in the "Read" column of FIG. 5 as "0011".

For this preferred embodiment of pseudo random count sequence, note from FIG. 4 that in a read operation, when the present state of the forward polynomial bits P0-P1 is "0001" the next state of these bits will be "1000", i.e. the two entries "0010" and "0100" shown in the "Present State" column are not used. Also note from FIG. 5 that for a write operation, when the present state of the reverse polynomial counter bits P4-P7 is "0011" the next state of these bits will be "0001", i.e. the two entries "0110" and "1100" in the "Present State" column are not used.

Note that while data bytes M and N, as referred to above, have a certain relationship as seen in FIGS. 4 and 5, i.e. they are separated by six intervening data bytes, this relationship is not critical, and in fact bytes M and N have been arbitrarily identified in these figures. In addition, the 4-bit pseudo random sequence shown in the three columns of FIGS. 4 and 5 are not to be taken as a limitation on the invention, since those skilled in the art will readily visualise other distinct sequences that will also function within the scope of the invention as claimed.

The next-state logic equations for forward polynomial counter bits P0-P3 is given in FIG. 4, and the next-state logic equations for reverse polynomial counter bits P4-P7 is given in FIG. 5.

The equation terms such as "P0n" of FIGS. 4 and 5 mean the "next binary value" of the forward polynomial counter bit P0, and the equation terms such as "P0" mean the "current or present binary value" of the forward polynomial counter bit P0.

That is, in the above write example (where write byte M was being randomized, the current value of P0 was equal to "0", and its next state value for the next write byte M+1 was equal to "0") the next state value of "0" for P0 is determined by the Boolean equation given in FIG. 4, which equation requires use of the current values "0", "0" and "1" of forward polynomial bits P0, P1 and P2, respectively.

Note from FIGS. 4 and 5 that P1n, i.e. the next value of P1, equals the current or present value of P2, that the next value of P3 equals the current value of P0, and that the next value of P4 equals the exclusive OR of the current value of P5 and P4.

As for the next value of P2 (i.e. P2n), P5, P6 and P7, these values are given by the Boolean equations of FIGS. 4 and 5, which equation requires the current values of the polynomial bits as is shown in the figures.

The columns and equations of FIGS. 4 and 5 define the pseudo randomness of the preferred embodiment of counter 40, but as those skilled in the art will recognise, other pseudo random binary counting sequences can be originated within the spirit and scope of the invention to satisfy multiplication-by-the-inverse.

During a write operation, the 4-bit register P0-P3 of counter 40 counts in a pseudo random manner as is defined by the above mentioned forward polynomial, and as is shown in the "Write" column of FIG. 4, as the 4-bit register P4-P7 of counter 40 counts in a pseudo random manner as is defined by the above mentioned reverse polynomial, and as is shown in the "Write" column of FIG. 5.

During a read operation, the 4-bit register P0-P3 of counter 40 counts in accordance with the forward polynomial, and as is shown in the "Read" column of FIG. 4, as the 4-bit register P4-P7 of counter 40 counts in accordance with the reverse polynomial, and as is shown in the "Read" column of FIG. 5. In this way, randomizing by a multiplication process is achieved, and derandomization by a complementary multiplication-by-the-inverse process is achieved.

Counter 40 cooperates with four 4-bit by 4-bit multiplier networks 41-44.

Networks 41 and 43 operate to multiply counter input bits B0-B3 and data input bits A0-A3 as is defined by the "FORWARD POLYNOMIAL MULTIPLY" portion of FIG. 6, and networks 42 and 44 operate to multiply counter input bits B4-B7 and data input bits A4-A7 as is defined by the "REVERSE POLYNOMIAL MULTIPLY" portion of FIG. 6.

Networks 41 and 43 are identical in construction. However, note that the counter-input terminals B0-B3 of network 41 and the counter-input terminals B0-B3 of network 41 are reverse connected to polynomial counter 40. That is, B0-B3 of network 41 are connected to counter bits P0-P3 in an advancing order manner, but B0-B3 of network 43 are connected to counter bits P4-P7 in an inverse order manner.

Also, networks 42 and 44 are identical in construction. Again, note that the counter-input terminals B4-B7 of network 42 and the counter-input terminals B4-B7 of network 41 are reverse connected. That is, B4-B7 of network 42 are connected to counter bits P4-P7 in an advancing order manner, but B4-B7 of network 44 are

connected to counter bits P0-P3 in an inverse order manner.

In this embodiment of the invention the write channel's randomizer 27 of FIG. 1 is made up of counter 40, forward-polynomial-multiplier network 41 and reverse-polynomial-multiplier network 42, and the read channel's derandomizer 34 of FIG. 1 is made up of counter 40, forward-polynomial-multiplier network 43 and reverse-polynomial-multiplier network 44.

As shown in FIG. 3, the write channel's 8-bit bus 26 of FIG. 1 splits into two 4-bit buses 26′ and 26″ in order to present the lower four bits CW0-CW3 of each write channel byte to forward-polynomial-multiplier network 41 by way of 4-bit bus 26′, and to present the upper four bits CW4-CW7 of each write channel byte to reverse-polynomial-multiplier network 42 by way of 4-bit bus 26″.

The output of randomizer 40,41,42 comprises 8-bit bus 28. In FIG. 3, bus 28 is formed by the two 4-bit busses 28′ and 28″ that receive the randomized byte as two 4-bit portions W0-W3 and W4-W7 from networks 41 and 42, respectively.

Also as shown in FIG. 3, the read channel's 8-bit bus 32 of FIG. 1 splits into two 4-bit buses 32′ and 32″ in order to present the lower four bits R0-R3 of each read channel byte to forward-polynomial-multiplier network 43 by way of 4-bit bus 32′, and to present the upper four bits R4-R7 of each read channel byte to reverse-polynomial-multiplier network 44 by way of 4-bit bus 32″.

The output of derandomizer 40,43,44 comprises 8-bit bus 35. In FIG. 3, bus 35 is formed by the two 4-bit busses 35′ and 35″ that receive the derandomized byte as two 4-bit portions CR0-CR3 and CR4-CR7 from networks 43 and 44, respectively.

A conductor 40/41 of FIG. 3 provides the synchronisation of counter 40, as the counter functions in either the write channel or the read channel, respectively, as was described relative FIG. 2.

As data fields 13,14 of FIG. 1 are being written to the disk (i.e. counter 40 is now functioning in the write channel's randomizer 40,41,42), byte counter 45 operates to increment counter 40 to its next pseudo random count state upon the processing of each individual byte within write data field 13,14. When the next sync field signal 12 is encountered, as provided by conductor 51 of FIG. 2, counter 40 is initialised or reseeded, and the randomizing process repeats for each individual byte within the next write data field 13,14.

As data fields 13,14 of FIG. 1 are subsequently read back from the disk (i.e. counter 40 is now functioning in the read channel's derandomizer 40,43,44), byte counter 45 again operates to increment counter 40 to its next count state upon the processing of each individual byte within read data field 13,14. When the next sync field signal 12 is detected, as provided by sync detector 33 and conductor 52 of FIG. 2, counter 40 is initialised or reseeded, and the derandomizing process repeats for each individual byte within the next read data field 13,14.

The following Table A shows the beginning and the ending portions of the 196 cycles of counter 40 of FIG. 3 for an exemplary and repeating write data pattern "11111111". While counter 40 can be started or seeded to any one of its 195 valid count states, Table A is made to be consistent with the above explanation where the counter is seeded to the state "00010001" for a write operation, and is seeded to the state "10001000" for a read operation.

TABLE A

Writing			Reading		
(1)	(2)	(3)	(4)	(5)	(6)
pppppppp	cccccccc	wwwwwwww	pppppppp	rrrrrrrr	cccccccc
01234567	wwwwwwww	01234567	01234567	01234567	rrrrrrrr
	01234567				01234567
00010001	11111111	11111111	10001000	11111111	11111111
00100010	11111111	11010111	00111001	11010111	11111111
01000100	11111111	10011110	01101011	10011110	11111111
10001000	11111111	00010101	11001111	00010101	11111111
00111001	11111111	00101010	10110111	00101010	11111111

01101011	11111111	01001101	01011110	01001101	11111111
11001111	11111111	10000011	10100101	10000011	11111111
10110111	11111111	00110110	01111010	00110110	11111111
01011110	11111111	01101100	11101101	01101100	11111111
10100101	11111111	11000001	11110011	11000001	11111111
01111010	11111111	10110010	11010110	10110010	11111111
11101101	11111111	01010100	10011100	01010100	11111111
11110011	11111111	10101000	00010001	10101000	11111111
11010001	11111111	01111111	10000010	01111111	11111111
10010010	11111111	11100111	00110100	11100111	11111111
00010100	11111111	11111110	01001000	11111110	11111111
"	"	"	"	"	"
"	"	"	"	"	"
"	"	"	"	"	"
"	"	"	"	"	"
00011101	11111111	11110100	10011000	11110100	11111111
00100011	11111111	11011000	00011001	11011000	11111111
01000001	11111111	10011111	10001011	10011111	11111111
10000010	11111111	00010111	00111111	00010111	11111111
00110100	11111111	00101110	01100111	00101110	11111111
01101000	11111111	01000101	11001110	01000101	11111111
11001001	11111111	10001010	10110101	10001010	11111111
10111011	11111111	00111101	01011010	00111101	11111111
01011111	11111111	01100011	10101101	01100011	11111111
10100111	11111111	11000110	01110011	11000110	11111111
01111110	11111111	10111100	11100110	10111100	11111111
11100101	11111111	01010001	11111100	01010001	11111111
11111010	11111111	10100010	11010001	10100010	11111111
11011101	11111111	01110100	10010010	01110100	11111111
10010011	11111111	11101000	00010100	11101000	11111111
00010001	11111111	11111111	10001000	11111111	11111111

Column (1) of the table shows the pseudo random counting states of counter 40 for a write operation, while column (4) shows the pseudo random counting states for a read operation.

In this example, the data to be randomized is the reoccurring byte "11111111" as shown in column (2). Data of this type poses a major problem for both magnetic recording stress and error recovery stress, and it is of the type that is found in the recording of graphics data, for example. This data is recovered, by derandomization, as is shown in column (6).

Column (3) shows the randomized bytes that are written to the disk (i.e. the data of column (2) is randomized to produce the data of column (3)), and column (5) shows the read-back of this written data. Thus, columns (3) and (5) are identical.

Note that at the occurrence of the 13th data byte during a write operation (i.e.column (1)) counter bits P4-P7 have completed all 13 of the pseudo random count states, and for the 14th data byte, the bits P4-P7 are

restored to the byte-1 state of "0001". Also, note that at the occurrence of the 15th data byte during a write operation counter bits P0-P3 have completed all 15 of the pseudo random count states, and for the 16th data byte, the bits P0-P3 are restored to the byte-1 state of "0001". This pseudo count sequence for counter bits P0-P7 continues until either the 196th data byte is received, or a synchronisation signal 12 is received, whereupon the counters bit state becomes "00010001".

For a read operation, at the occurrence of the 13th data byte (i.e.column (4)) counter bits P0-P3 have completed all 13 of the pseudo random count states, and for the 14th data byte, the bits P0-P3 are restored to the byte-1 state of "1000". Also, note that at the occurrence of the 15th data byte during a read operation counter bits P4-P7 have completed all 15 of the pseudo random count states, and for the 16th data byte, the bits P4-P7 are restored to the byte-1 state of "1000". This sequence continues until either the 196th data byte is received, or a synchronisation signal 12 is received, whereupon the counters bit state becomes "10001000".

Claims

1. A method of writing data onto a disk data storage device, the data being written onto a data storage disk as a series of data fields (11), each data field including user data (13) and corresponding error correcting code (ECC) data, the method comprising the steps of:
   generating ECC data for the user data to produce a string of data words representing the user data and corresponding ECC data for a data field;
   and characterised by then randomising said ECC and user data within the data field by multiplying or dividing each data word by data from a sequence of pseudo random data.

2. A method as claimed in claim 1, further including:
   run-length encoding the data record after randomization and prior to writing.

3. A method of reading data from a disk data storage device, the data being stored on the data storage disk as a series of data fields (11), each data field including user data (13) and corresponding error correcting code (ECC) data, the method comprising the steps of:
   reading a data field as a string of data words from the disk and using the user data and corresponding ECC data read back from the data field to identify and correct any errors;
   and characterised by the step of derandomizing said ECC and user data within the data field by multiplying or dividing each data word by data from a sequence of pseudo random data, said derandomizing occurring prior to using the user data and corresponding ECC data read back from the data field to identify and correct any errors.

4. A method as claimed in claim 3, further including:
   run-length decoding the read-back data record prior to derandomization.

5. A method as claimed in any preceding claim, wherein the randomising/derandomizing sequence is determined by the state of a register (40), the state of the register being initialised at the beginning of the data record and the read-back data record, and the register being progressively incremented to a next predetermined state as each byte of the multi-byte data record and each byte of the multi-byte read-back data record is progressively processed.

6. A method as claimed in claim 5, wherein the register is an 8-bit register, the method including:
   during data randomizing, multiplying the first four bits of each data record byte by the first four bits of the 8-bit register, and multiplying the last four bits of each data record byte by the last four bits of the 8-bit register, and
   subsequently, during derandomizing of the read-back data record data, performing a multiplication-of-the-inverse process by multiplying the first four bits of each read-back data record byte by the inverse order of the last four bits of the 8-bit counter, and multiplying the last four bits of each read-back record byte by the inverse order of the first four bits of the 8-bit register.

7. A disk data storage device including a non-volatile data storage medium, a transducer arrangement (20) for writing onto and reading from the medium, the data being written onto the medium as a series of data fields (11), each data field including user data (13) and corresponding error correcting code (ECC) data, and a write channel (21) and a read channel (22);

* C L M S *

restored to the byte-1 state of "0001". Also, note that at the occurrence of the 15th data byte during a write operation counter bits P0-P3 have completed all 15 of the pseudo random count states, and for the 16th data byte, the bits P0-P3 are restored to the byte-1 state of "0001". This pseudo count sequence for counter bits P0-P7 continues until either the 196th data byte is received, or a synchronisation signal 12 is received, whereupon the counters bit state becomes "00010001".

For a read operation, at the occurrence of the 13th data byte (i.e.column (4)) counter bits P0-P3 have completed all 13 of the pseudo random count states, and for the 14th data byte, the bits P0-P3 are restored to the byte-1 state of "1000". Also, note that at the occurrence of the 15th data byte during a read operation counter bits P4-P7 have completed all 15 of the pseudo random count states, and for the 16th data byte, the bits P4-P7 are restored to the byte-1 state of "1000". This sequence continues until either the 196th data byte is received, or a synchronisation signal 12 is received, whereupon the counters bit state becomes "10001000".

Claims

1. A method of writing data onto a disk data storage device, the data being written onto a data storage disk as a series of data fields (11), each data field including user data (13) and corresponding error correcting code (ECC) data, the method comprising the steps of:

   generating ECC data for the user data to produce a string of data words representing the user data and corresponding ECC data for a data field;

   and characterised by then randomising said ECC and user data within the data field by multiplying or dividing each data word by data from a sequence of pseudo random data.

2. A method as claimed in claim 1, further including:

   run-length encoding the data record after randomization and prior to writing.

3. A method of reading data from a disk data storage device, the data being stored on the data storage disk as a series of data fields (11), each data field including user data (13) and corresponding error correcting code (ECC) data, the method comprising the steps of:

   reading a data field as a string of data words from the disk and using the user data and corresponding ECC data read back from the data field to identify and correct any errors;

   and characterised by the step of derandomizing said ECC and user data within the data field by multiplying or dividing each data word by data from a sequence of pseudo random data, said derandomizing occurring prior to using the user data and corresponding ECC data read back from the data field to identify and correct any errors.

4. A method as claimed in claim 3, further including:

   run-length decoding the read-back data record prior to derandomization.

5. A method as claimed in any preceding claim, wherein the randomising/derandomizing sequence is determined by the state of a register (40), the state of the register being initialised at the beginning of the data record and the read-back data record, and the register being progressively incremented to a next predetermined state as each byte of the multi-byte data record and each byte of the multi-byte read-back data record is progressively processed.

6. A method as claimed in claim 5, wherein the register is an 8-bit register, the method including:

   during data randomizing, multiplying the first four bits of each data record byte by the first four bits of the 8-bit register, and multiplying the last four bits of each data record byte by the last four bits of the 8-bit register, and

   subsequently, during derandomizing of the read-back data record data, performing a multiplication-of-the-inverse process by multiplying the first four bits of each read-back data record byte by the inverse order of the last four bits of the 8-bit counter, and multiplying the last four bits of each read-back record byte by the inverse order of the first four bits of the 8-bit register.

7. A disk data storage device including a non-volatile data storage medium, a transducer arrangement (20) for writing onto and reading from the medium, the data being written onto the medium as a series of data fields (11), each data field including user data (13) and corresponding error correcting code (ECC) data, and a write channel (21) and a read channel (22);

* C L M S *

said write channel including means for generating ECC data for the user data to produce a string of data words representing the user data and corresponding ECC data for a data field;

said read channel including means for reading a data field as a string of data words from the disk and means for using the user data and corresponding ECC data read back from the data field to identify and correct any errors;

and characterised in that

said write channel includes means for randomizing said ECC and user data within the data field by multiplying or dividing each data word by data from a sequence of pseudo random data;

and said read channel includes means for derandomizing said ECC and user data within the data field by performing inverse processing to that used for the randomization, wherein said derandomizing occurs prior to using the user data and corresponding ECC data read back from the data field to identify and correct any errors.

8. A disk data storage device as claimed in claim 7, wherein the write channel includes;

first data-input means (23) adapted for the reception of binary data to be written to the media, the binary data comprising a plurality of multi-byte data fields,

second data-input means (29) adapted for the reception of an identifying data field for each of the multi-byte data fields,

randomizer means having first input means connected to the first data-input means to thereby receive the multi-byte data fields and second input means connected to the second data-input means and being responsive to an identifying data field to begin a repetitive sequence of data randomizing for each byte of the multi-byte data field, and provide an output comprising a plurality of randomized multi-byte data fields, and

means (21) connecting the transducer arrangement to the output of the randomizer means and to the second data-input means, whereby randomized data comprising a plurality of multi-byte data fields, each data field including an identifying data field, is written on the non volatile media;

the read channel having

read-data input means (21) connected to the transducer arrangement for the reception of the read-back randomized data,

identifying data detector (33) means having an output, and having an input connected to the read-data input means for detecting the identifying data field of each read-data field, and

derandomizer means (34) having first input means connected to the read-date input means, to receive the randomized data, the derandomizer means having second input means connected to the output of the identifying data detector means, the derandomizer means being responsive to detection of an identifying data field to begin a sequence of derandomizing of each byte of the read-data field whereby the derandomization is the inverse of the randomization.

9. A disk data storage device as claimed in claim 7 or claim 8, wherein the medium is a magnetic disk.

10. A disk data storage device as claimed in any of claims 7 to 9, wherein the randomizer means includes means operable to multiply each byte of the multi-byte data field by a pseudo random polynomial function, and wherein the derandomizer means includes complementary means operable to multiple the multi-byte read-data field in accordance with the inverse of the pseudo random polynomial function.

11. A disk data storage device as claimed in any of claims 7 to 10, including

means (30) for run-length encoding the data record after randomization and prior to writing, and

means (31) for run-length decoding the read-back data record in a complementary fashion prior to derandomization.

12. A disk data storage device as claimed in any of claims 7 to 11, including a register and wherein the complementary randomizing/derandomizing sequence is determined by the binary state of the register (40), the state of the register being initialised at the beginning of the data record and the read-back data record, and the register being progressively incremented to a next predetermined state as each byte of the multi-byte data record and each byte of the multi-byte read-back data record is progressively processed.

13. A disk data storage device as claimed in claim 12, wherein the register is an 8-bit register, including:

means (41-43) for, during data randomizing, multiplying the first four bits of each data record byte by the first four bits of the 8-bit register, and multiplying the last four bits of each data record byte by the

* C L M S *

last four bits of the 8-bit register, and

means (42,44) for subsequently, during derandomizing of the read-back data record data, performing a multiplication-of-the-inverse process by multiplying the first four bits of each read-back data record byte by the inverse order of the last four bits of the 8-bit register, and multiplying the last four bits of each read-back record byte by the inverse order of the first four bits of the 8-bit register.

Patentansprüche

1. Ein Verfahren, um Daten auf eine Plattenspeichervorrichtung zu schreiben, wobei die Daten als eine Folge von Datenfeldern 11 auf die Speicherplatte geschrieben werden, die jeweils Benutzerdaten 13 und die zugehörigen Fehlerkorrekturdaten enthalten, wobei das Verfahren einen Schritt enthält, in dem die Fehlerkorrekturdaten zu den Benutzerdaten generiert werden, um eine Folge von Datenwörtern zu erzeugen, welche die Benutzerdaten und die zugehörigen Fehlerkorrekturdaten für ein Datenfeld darstellen, und dadurch gekennzeichnet ist, daß die Fehlerkorrekturdaten und die Benutzerdaten im Datenfeld anschließend pseudozufällig verschlüsselt werden, indem jedes Datenwort mit Daten aus einer Sequenz pseudozufälliger Daten multipliziert oder dividiert wird.

2. Ein Verfahren gemäß Anspruch 1, das ferner einen Schritt enthält, in dem der Datensatz nach der pseudozufälligen Verschlüsselung und vor dem Schreiben lauflängencodiert wird.

3. Ein Verfahren zum Lesen von Daten von einer Plattenspeichervorrichtung, wobei die Daten auf der Speicherplatte als eine Folge von Datenfeldern 11 gespeichert sind, die jeweils Benutzerdaten 13 und die zugehörigen Fehlerkorrekturdaten enthalten, wobei das Verfahren einen Schritt enthält, in dem ein Datenfeld als eine Folge von Datenwörtern von der Platte gelesen wird und die aus dem Datenfeld zurückgelesenen Benutzerdaten und die zugehörigen Fehlerkorrekturdaten zur Erkennung und Korrektur eventueller Fehler verwendet werden, und wobei das Verfahren dadurch gekennzeichnet ist, daß es ferner einen Schritt enthält, in dem die Fehlerkorrekturdaten und die Benutzerdaten im Datenfeld entschlüsselt werden, indem jedes Datenwort mit Daten aus einer Sequenz pseudozufälliger Daten multipliziert oder dividiert wird, wobei diese Entschlüsselung erfolgt, bevor die aus dem Datenfeld zurückgelesenen Benutzerdaten und Fehlerkorrekturdaten zur Erkennung und Korrektur eventueller Fehler verwendet werden.

4. Ein Verfahren gemäß Anspruch 3, das außerdem einen Schritt enthält, in dem die zurückgelesenen Daten vor der Entschlüsselung lauflängendecodiert werden.

5. Ein Verfahren gemäß den vorherigen Ansprüchen, bei dem die Verschlüsselungs- bzw. Entschlüsselungssequenz durch den Stand eines Registers 40 bestimmt wird, das am Anfang des Datensatzes und des zurückgelesenen Datensatzes initialisiert wird, und bei dem das Register mit der Verarbeitung jedes Byte eines aus mehreren Bytes bestehenden Datensatzes oder zurückgelesenen Datensatzes progressiv auf einen nächsten vorgegebenen Stand gebracht wird.

6. Ein Verfahren gemäß Anspruch 5, bei dem das Register ein 8-Bit-Register ist, und in dem bei der Verschlüsselung der Daten die ersten vier Bits jedes Datensatzbytes mit den ersten vier Bits des 8-Bit-Registers und die letzten vier Bits jedes Datensatzbytes mit den letzten vier Bits des 8-Bit-Registers multipliziert werden, und in dem anschließend bei der Entschlüsselung der zurückgelesenen Daten aus dem Datensatz eine Multiplikation mit dem Kehrwert stattfindet, bei der die ersten vier Bits jedes zurückgelesenen Datensatzbytes mit der umgekehrten Reihenfolge der letzten vier Bits des 8-Bit-Zählers und die letzten vier Bits jedes zurückgelesenen Datensatzbytes mit der umgekehrten Reihenfolge der ersten vier Bits des 8-Bit-Registers multipliziert werden.

7. Eine Plattenspeichervorrichtung mit einem nichtflüchtigen Datenspeichermedium, mit einer Transducervorrichtung 20 zum Schreiben auf das Medium und zum Lesen von dem Medium, wobei die Daten als Folge von Datenfeldern 11 auf das Medium geschrieben werden, die jeweils Benutzerdaten 13 und die zugehörigen Fehlerkorrekturdaten enthalten, und mit einem Schreibkanal 21 und einem Lesekanal 22,

wobei der Schreibkanal Mittel zur Generierung der Fehlerkorrekturdaten für die Benutzerdaten enthält, um eine Folge von Datenwörtern zu erzeugen, welche die Benutzerdaten und die zugehörigen Fehlerkorrekturdaten für ein Datenfeld enthalten,

* C L M S *

und der Lesekanal Mittel zum Lesen eines Datenfeldes als Folge von Datenwörtern von der Platte sowie Mittel zur Verwendung der von der Platte zurückgelesenen Benutzerdaten und Fehlerkorrekturdaten eines Datenfeldes zum Erkennen und Korrigieren eventueller Fehler enthält;
und wobei die Plattenspeichervorrichtung dadurch gekennzeichnet ist, daß

der Schreibkanal Mittel zur pseudozufälligen Verschlüsselung der Fehlerkorrektur- und Benutzerdaten im Datenfeld durch Multiplikation oder Division jedes Datenwortes mit Daten aus einer Sequenz von pseudozufälligen Daten enthält,

und der Lesekanal Mittel zur Entschlüsselung der Fehlerkorrektur- und Benutzerdaten im Datenfeld durch Umkehrung des zur Verschlüsselung verwendeten Prozesses enthält, wobei die Entschlüsselung stattfindet, bevor die aus dem Datenfeld zurückgelesenen Benutzerdaten und Fehlerkorrekturdaten zum Erkennen und Korrigieren eventueller Fehler verwendet werden.

8. Eine Plattenspeichervorrichtung gemäß Anspruch 7, bei dem der Schreibkanal folgende Komponenten enthält:

ein erstes Dateneingabemittel 23, das an den Empfang binärer Daten, die auf das Medium geschrieben werden sollen, angepaßt ist, wobei die binären Daten mehrere Datenfelder enthalten, die jeweils aus mehreren Bytes bestehen,

ein zweites Dateneingabemittel 29, das an den Empfang eines kennzeichnenden Datenfeldes für jedes aus mehreren Bytes bestehenden Datenfeldes angepaßt ist,

eine Verschlüsselungsvorrichtung, welche ein an das erste Dateneingabemittel angeschlossenes erstes Eingabemittel enthält, um damit die aus mehreren Bytes bestehenden Datenfelder zu empfangen, sowie ein an das zweite Dateneingabemittel angeschlossenes zweites Eingabemittel, das auf ein kennzeichnendes Datenfeld anspricht, um für jedes Byte eines aus mehreren Bytes bestehenden Datenfeldes eine sich wiederholende Verschlüsselungssequenz zu starten und eine Ausgabe zu erzeugen, die aus mehreren verschlüsselten Datenfeldern mit jeweils mehreren Bytes besteht, sowie

ein Mittel 21 zum Anschluß der Transducervorrichtung an den Ausgang der Verschlüsselungsvorrichtung und an das zweite Dateneingabemittel, wobei die aus mehreren Datenfeldern bestehenden verschlüsselten Daten, die mehrere Datenfelder mit jeweils mehreren Bytes und einem kennzeichnenden Datenfeld enthalten, auf das nichtflüchtige Speichermedium geschrieben werden;
und bei dem der Lesekanal folgende Komponenten enthält:

ein an die Transducervorrichtung angeschlossenes Lesedateneingabemittel 21 zum Empfang der zurückgelesenen verschlüsselten Daten,

ein identifizierendes Datendetektormittel 33 mit einem Ausgang und einem an den Lesedateneingang angeschlossenen Eingang zur Erkennung des kennzeichnenden Datenfeldes jedes zurückzulesenden Datenfeldes, und

ein Entschlüsselungsmittel 34 mit einem an das Lesedateneingangsmittel angeschlossenen ersten Eingangsmittel zum Empfangen der verschlüsselten Daten, das außerdem ein an den Ausgang des identifizierenden Datendetektors angeschlossenes zweites Eingangsmittel besitzt und auf die Erkennung eines kennzeichnenden Datenfeldes anspricht, um eine Sequenz zur Entschlüsselung jedes Bytes des zurückzulesenden Datenfeldes zu starten, wobei die Entschlüsselung die Umkehrung der Verschlüsselung ist.

9. Eine Plattenspeichervorrichtung gemäß Anspruch 7 oder 8, bei der das Speichermedium eine Magnetplatte ist.

10. Eine Plattenspeichervorrichtung gemäß Anspruch 7 bis 9, bei der das Verschlüsselungsmittel ein Mittel enthält, das jedes Byte des aus mehreren Bytes bestehenden Datenfeldes mit einer pseudozufälligen Polynomfunktion multipliziert, und bei der das Entschlüsselungsmittel ein komplementäres Mittel enthält, welches das aus mehreren Bytes bestehende Lesedatenfeld mit dem Kehrwert der pseudozufälligen Polynomfunktion multipliziert.

11. Eine Plattenspeichervorrichtung gemäß Anspruch 7 bis 10, die ein Mittel 30 zur Lauflängencodierung des Datensatzes nach der Verschlüsselung und vor dem Schreiben sowie ein Mittel 31 zur komplementären Lauflängendecodierung des zurückgelesenen Datensatzes vor der Entschlüsselung enthält.

12. Eine Plattenspeichervorrichtung gemäß Anspruch 7 bis 11, die ein Register enthält und bei der die komplementäre Verschlüsselungs-/Entschlüsselungssequenz durch den binären Stand des Registers 40 bestimmt wird, wobei der Stand des Registers am Anfang des Datensatzes und des zurückzulesenden Da-

tensatzes initialisiert wird und das Register mit der Verarbeitung jedes Byte des aus mehreren Bytes bestehenden Datensatzes und jedes Byte des aus mehreren Bytes bestehenden zurückzulesenden Datensatzes progressiv auf den nächsten vorgegebenen Stand gesetzt wird.

13. Eine Plattenspeichervorrichtung gemäß Anspruch 12, bei der das Register ein 8-Bit-Register ist, bestehend aus:

einem Mittel 41 bis 43, das bei der Verschlüsselung der Daten die ersten vier Bits jedes Datensatzbytes mit den ersten vier Bytes des 8-Bit-Registers und die letzten vier Bits jedes Datensatzbytes mit den letzten vier Bits des 8-Bit-Registers multipliziert, und

einem Mittel 42, 44, das bei der nachfolgenden Entschlüsselung des zurückgelesenen Datensatzes eine Multiplikation mit dem Kehrwert durchführt, indem es die ersten vier Bits jedes zurückgelesenen Datensatzbytes mit der umgekehrten Reihenfolge der letzten vier Bits des 8-Bit-Registers und die letzten vier Bits jedes zurückgelesenen Datensatzbytes mit der umgekehrten Reihenfolge der ersten:vier Bits des 8-Bit-Registers multipliziert.

Revendications

1. Procédé d'écriture de données sur un dispositif de mémorisation de données à disque, les données étant écrites sur un disque de mémorisation de données sous forme de séries de champs de données (11), chaque champ de données comprenant des données d'utilisateur (13) et des données de correction d'erreur (ECC) correspondantes, le procédé comprenant les étapes consistant à :
générer des données de correction d'erreur pour les données d'utilisateur afin de produire une chaîne de mots de données représentant les données d'utilisateur et les données de correction d'erreur correspondantes pour un champ de données,
et caractérisé en ce que l'on met alors sous forme aléatoire ledit code de correction d'erreur et lesdites données d'utilisateur à l'intérieur du champ de données en multipliant ou en divisant chaque mot de données par des données provenant d'une séquence de données pseudo aléatoire.

2. Procédé selon la revendication 1, comprenant en outre :
un codage par longueurs de suites répétives, de l'enregistrement des données après les avoir rendues aléatoires et avant l'écriture.

3. Procédé de lecture de données à partir d'un dispositif de mémorisation de données à disques, les données étant mémorisées sur le disque de mémorisation de données sous forme de séries de champ de données (11), chaque champ de données comprenant des données d'utilisateur (13) et des données de code de correction d'erreur (ECC), correspondantes, le procédé comprenant les étapes consistant à :
lire un champ de données sous forme d'une chaîne de mots de données à partir du disque et en utilisant les données d'utilisateur et les données de correction d'erreur correspondantes relues à partir du champ de données afin d'identifier et de corriger toute erreur quelconque,
et caractérisé par l'étape consistant à supprimer la forme aléatoire desdits codes de correction d'erreur et desdites données d'utilisateur à l'intérieur du champ de données en multipliant ou en divisant chaque mot de données par des données provenant d'une séquence de données pseudo aléatoire, ladite suppression de forme aléatoire se produisant avant d'utiliser les données d'utilisateur et les données de code de correction d'erreur correspondantes relues à partir du champ de données afin d'identifier et de corriger toute erreur quelconque.

4. Procédé selon la revendication 3, comprenant en outre :
le décodage par longueurs de suites répétives de l'enregistrement des données relues avant de supprimer la forme aléatoire.

5. Procédé selon toutes revendications précédentes, dans lequel la séquence de mise en forme aléatoire/suppression de forme aléatoire est déterminée par l'état d'un registre (40), l'état du registre étant initialisé au début de l'enregistrement des données et de l'enregistrement des données relues, et le registre étant progressivement incrémenté à un état prédéterminé suivant à mesure que chaque octet de l'enregistrement de données multi-octet et chaque octet de l'enregistrement de données relu multi-octet est progressivement traité.

6. Procédé selon la revendication 5 dans lequel le registre est un registre à 8 bits, le procédé comprenant :
lors de la mise sous forme aléatoire des données, la multiplication des quatre premiers bits de chaque octet d'enregistrement de données par les quatre premiers bits du registre à 8 bits, et la multiplication des quatre derniers bits de chaque octet d'enregistrement de données par les quatre derniers bits du registre à 8 bits, et
ensuite, pendant la suppression de la forme aléatoire des données d'enregistrement de données relues, l'exécution d'un traitement par multiplication inversée en multipliant les quatre premiers bits de chaque octet d'enregistrement de données relues par les quatre derniers bits en ordre inversé du compteur à 8 bits, et en multipliant les quatre derniers bits de chaque octet d'enregistrement relu par les quatre premiers bits en ordre inversé du registre 8 bits.

7. Dispositif de mémorisation de données à disque comprenant un support de mémorisation de données non volatile, un agencement de transducteurs (20) destiné à écrire sur et à lire à partir du support, les données étant écrites sur les supports sous forme d'une série de champs de données (11), chaque champ de données comprenant des données d'utilisateur (13) et des données de codes de correction d'erreur (ECC) correspondantes, et un canal d'écriture (21) et un canal de lecture (22),
ledit canal d'écriture comprenant un moyen destiné à générer des données de codes de correction d'erreur pour les données d'utilisateur afin de produire une chaîne de mots de données en présentant les données d'utilisateur et les données de codes de correction d'erreur correspondantes à un champ de données,
ledit canal de lecture comprenant un moyen destiné à lire un champ de données sous forme d'une chaîne de mots de données provenant du disque et un moyen destiné à utiliser les données d'utilisateur et les données de codes de correction d'erreur correspondantes relues à partir du champ de données pour identifier et corriger toute erreur quelconque,
et caractérisé en ce que
ledit canal d'écriture comprend un moyen pour mettre sous forme aléatoire ledit code de correction d'erreur et lesdites données d'utilisateur à l'intérieur du champ de données en multipliant ou en divisant chaque mot de données par des données provenant d'une séquence de données pseudo aléatoires,
et en ce que ledit canal de lecture comprend un moyen pour supprimer la forme aléatoire desdits codes de correction d'erreur et desdites données d'utilisateur à l'intérieur du champ de données en effectuant le traitement inverse de celui utilisé pour la mise en forme aléatoire, dans lequel ladite suppression de forme aléatoire se produit avant d'utiliser les données d'utilisateur et les données de codes de correction d'erreur correspondantes relues à partir du champ de données afin d'identifier et de corriger toute erreur quelconque.

8. Dispositif de mémorisation de données à disque selon la revendication 7, dans lequel le canal d'écriture comprend,
un premier moyen d'entrée de données (23) adapté à la réception de données binaires à écrire sur le support, les données binaires comprenant une pluralité de champs de données multi-octet,
un second moyen d'entrée de données (29) adapté à la réception d'un champ de données d'identifiant pour chacun des champs de données multi-octet,
un moyen de mise sous forme aléatoire comprenant un premier moyen d'entrée relié au premier moyen d'entrée de données pour recevoir de ce fait les champs de données multi-octet et un second moyen d'entrée relié au second moyen d'entrée de données et répondant à un champ de données d'identification pour lancer une séquence répétitive de mise sous forme aléatoire de données pour chaque octet du champ de données multi-octet, et procurer une sortie comprenant une pluralité de champs de données multi-octet mise sous forme aléatoire, et
un moyen (21) reliant l'agencement de transducteurs à la sortie du moyen de mise sous forme aléatoire et au second moyen d'entrée de données, par lequel les données mises sous forme aléatoire comprenant une pluralité de champs de données multi-octet, chaque champ de données comportant un champ de données d'identifiant, sont écrites sur le support non volatile,
le canal de lecture comportant
un moyen d'entrée de données lues (21) relié à l'agencement de transducteurs pour la réception des données relues mises sous forme aléatoire,
un moyen de détecteur (33) de données d'identifiant comportant une sortie, et comportant une entrée reliée au moyen d'entrée de données lues afin de détecter le champ de données d'identifiant de chaque champ de données lues, et
un moyen de suppression de forme aléatoire (34) comportant un premier moyen d'entrée relié au moyen

* C L M S *

d'entrée de données lues, afin de recevoir les données mises sous forme aléatoire, le moyen de suppression de forme aléatoire comportant un second moyen d'entrée relié à la sortie du moyen de détecteur de données d'identifiant, le moyen de suppression de forme aléatoire répondant à la détection d'un champ de données d'identifiant afin de lancer une séquence de suppression de forme aléatoire de chaque octet du champ de données lues par laquelle la suppression de forme aléatoire est l'inverse de la mise sous forme aléatoire.

9. Dispositif de mémorisation de données à disque selon la revendication 7 ou la revendication 8, dans lequel le support est un disque magnétique.

10. Dispositif de mémorisation de données à disque selon l'une quelconque des revendications 7 à 9, dans lequel le moyen de mise sous forme aléatoire comprend un moyen pouvant être mis en oeuvre pour multiplier chaque octet du champ de données multi-octet au moyen d'une fonction polynomiale pseudo aléatoire, et dans lequel le moyen de suppression de forme aléatoire comprend un moyen complémentaire pouvant être mis en oeuvre pour multiplier les champs de données lues multi-octet conformément à l'inverse de la fonction polynomiale pseudo aléatoire.

11. Dispositif de mémorisation de données à disque selon l'une quelconque des revendications 7 à 10, comprenant,
un moyen (30) pour coder par longueurs de suites répétives l'enregistrement de données après la mise sous forme aléatoire et avant l'écriture, et
un moyen (31) pour décoder par longueurs de suites répétives l'enregistrement de données relues d'une façon complémentaire avant la suppression de la forme aléatoire.

12. Dispositif de mémorisation de données à disque selon l'une quelconque des revendications 7 à 11, comprenant un registre et dans lequel la séquence complémentaire de mise sous forme aléatoire/suppression de forme aléatoire, est déterminée par l'état binaire du registre (40), l'état du registre étant initialisé au début de l'enregistrement de données et de l'enregistrement de données relues, et le registre étant progressivement incrémenté à un état prédéterminé suivant à mesure que chaque octet de l'enregistrement de données multi-octet et que chaque octet de l'enregistrement de données relues multi-octet est progressivement traité.

13. Dispositif de mémorisation de données à disque selon la revendication 12, dans lequel le registre est un registre à 8 bits, comprenant :
un moyen (41 à 43) destiné à, pendant la mise sous forme aléatoire des données, multiplier les quatre premiers bits de chaque octet d'enregistrement de données par les quatre premiers bits du registre à 8 bits, et à multiplier les quatre derniers bits de chaque octet d'enregistrement de données par les quatre derniers bits du registre à 8 bits, et
un moyen (42,44) pour ensuite, pendant la suppression de la forme aléatoire des données d'enregistrement de données relues, effectuer un traitement par multiplication inversée en multipliant les quatre premiers bits de chaque octet d'enregistrement de données relues par les quatre derniers bits en ordre inversé du registre à 8 bits, et en multipliant les quatre derniers bits de chaque octet d'enregistrement relu par les quatre premiers bits en ordre inversé du registre à 8 bits.