

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

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
20 September 2001 (20.09.2001)

PCT

(10) International Publication Number
WO 01/69392 A2

(51) International Patent Classification⁷: **G06F 12/00**

(74) Agent: **SHAPIRO, Stuart, B.**; Epstein, Edell, Shapiro & Finnan, LLC, Suite 400, 1901 Research Boulevard, Rockville, MD 20850 (US).

(21) International Application Number: PCT/US01/08482

(22) International Filing Date: 16 March 2001 (16.03.2001)

(81) Designated States (*national*): AU, CA, JP, MX.

(25) Filing Language: English

(84) Designated States (*regional*): European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

(26) Publication Language: English

(30) Priority Data:
60/189,932 16 March 2000 (16.03.2000) US

Published:

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

(71) Applicant and

(72) Inventor: **BODEN, Scott, T.** [US/US]; 1600 Ludington Lane, LaJolla, CA 92037-6338 (US).

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



WO 01/69392 A2

(54) Title: METHOD AND APPARATUS FOR SECURE AND FAULT TOLERANT DATA STORAGE

(57) **Abstract:** A computer system including software manipulates data prior to storage on media disposed within a system storage device. The data is initially stored sequentially within a first array, while a pseudo-random number sequence is generated in accordance with a seed value to identify storage locations for associated data bits within a second array. The second array is stored sequentially on the media to randomly distribute the data across that media. In order to retrieve the data in original form, the entire contents of the media are retrieved and stored in a third array. The sequence is reproduced in accordance with the seed value, while data bits are retrieved in the order of the sequence and stored in temporary storage to recover the data. The seed value may further serve as a password to maintain the data in a secure fashion.

METHOD AND APPARATUS FOR SECURE AND FAULT TOLERANT DATA STORAGE

CROSS REFERENCE TO RELATED APPLICATIONS

This application claims priority to U.S. Provisional Patent Application Serial No. 60/189,932, entitled "Psuedo-Random Data Convolution Algorithm" and filed March 16, 2000, the disclosure of which is incorporated herein by reference in its entirety.

BACKGROUND OF THE INVENTION

1. Technical Field

The present invention pertains to data storage systems. In particular, the present invention pertains to a data storage system that stores information in a pseudo-random manner across media of a mass storage device (e.g., CD-ROM drive, floppy disk drive and drives for other removable or non-removable optical or magnetic type disks) to enhance data longevity, error correction and security.

2. Discussion of Related Art

Various storage devices are currently utilized with computer systems to store data. These storage devices typically receive removable storage media (e.g., CD-ROMs, floppy diskettes, Zip disks, etc.) generally having a rotatable disk with a plurality of tracks and sectors defined therein to contain data. The storage media typically store data from files in a sequential manner (e.g., physically proximate each other), while the files, in turn, are sequentially stored on the media. Although the sequential data arrangement employed by these devices may enhance data access rates, a defective or damaged disk surface may produce data errors of a magnitude sufficient to prevent recovery of data stored within or proximate the damaged or defective disk sections. These types of errors are typically limited to a few files, however, data within those files is generally not recoverable. The data loss may result in severe consequences, especially when the discarded data is of a secure or critical nature. In addition, the information stored on the storage media is generally accessible to any users

1 employing a storage device compatible with that media, thereby facilitating compromise of
2 stored information integrity.

3 The related art has attempted to overcome the aforementioned problems by providing
4 systems and/or methods that distribute data in various fashions within a memory device. For
5 example, U.S. Patent No. 4,789,902 (Shimura) discloses an image signal processing method
6 for storing a series of image signals on a recording medium or transmitting the image signals
7 to a receiving device, and reproducing the image from the stored or transmitted image signals.
8 The image signals are arrayed in the array sequence of scanning lines on an image. The series
9 of image signals is divided into predetermined units in the course of storing or transmitting
10 those signals, while the image signals are stored or transmitted by changing the sequence of
11 the respective units so that the units which were adjacent to each other do not adjoin each
12 other. The image is reproduced by rearranging the units in the original sequence in the course
13 of image reproduction.

14 U.S. Patent No. 5,276,826 (Rau et al) discloses a computer system having a multi-
15 module memory system. Access to the memory modules for reading or writing are undertaken
16 in parallel. The memory system is addressed by input addresses and includes a map unit for
17 transforming the input addresses to output addresses in a pseudo-random manner to distribute
18 memory accesses uniformly among the memory modules. The contention resulting from
19 multiple concurrent attempts to access the same memory module is thereby reduced. The map
20 unit performs address transforms that are repeatable so that the same input address maps to
21 the same output address and that are one-to-one such that each input address maps to one and
22 only one output address.

23 U.S. Patent No. 5,305,324 (Demos) discloses an error correction and detection
24 interface between a high speed data channel and a high capacity digital data recording tape
25 system. The interface includes a data scrambling and translation scheme which provides an
26 additional layer of error correction to the data as it is recorded. The data scrambling and
27 translation scheme permits the correction of normally uncorrectable large error bursts on
28 digital tape devices.

29 U.S. Patent No. 5,799,033 (Baggen) discloses an error protected transmission method.
30 Data is transmitted via a signal containing a number of simultaneously active modulated
31 frequency channels. The data is encoded in an error protecting code, while successive data

1 items are mapped pseudo-randomly to different frequency channels. The pseudo-random
2 mapping is realized by writing the data items into memory in one order and reading them from
3 memory in another order. Successive signals are each modulated in this way. The memory
4 locations vacated upon reading data items for the modulation of one signal are filled by data
5 items for modulating the next successive signal. This is maintained by permuting the order
6 of the memory locations in which the data items are written for each successive signal.

7 The related art suffers from several disadvantages. In particular, the Demos system
8 utilizes plural memory modules to arrange data in accordance with a predetermined offset
9 scheme, thereby significantly increasing system complexity and cost. The Shimura method
10 distributes or rearranges data that is partitioned or grouped into units, while the Rau et al
11 system employs a pseudo-random mapping of a memory input address to a memory output
12 address for achieving relatively uniform access of a plurality of memory modules and employs
13 data typically grouped and stored in the form of data words having several bits. Thus,
14 logically adjacent data within a unit or word is stored at adjacent locations. These techniques
15 basically limit distribution of data within memory and, with respect to storage media, increase
16 the risk of unrecoverable data loss in the event defects occur in the storage media where the
17 data is stored. The Baggen method employs a pseudo-random interleaving scheme where the
18 data is distributed with respect to frequency channels, while the above described systems
19 and/or methods generally utilize offset and/or interleaving schemes to distribute data. These
20 manners of distributing data tend to limit and provide a relatively uneven distribution of data
21 within the memory space, thereby increasing risk of unrecoverable data loss with respect to
22 storage media in the event of storage media defects as described above. In addition, the
23 systems and methods described above do not provide a manner for a user to selectively control
24 access to the distributed information.

25 The present invention overcomes the aforementioned problems of the related art by
26 distributing data in a pseudo-random fashion across storage media. Since the random
27 distribution enables sequentially stored data bits to be associated with different files, the
28 impact of a media surface defect is typically limited to a few bits within selected files. These
29 errors may be corrected with various conventional error correction techniques, thereby
30 permitting recovery of impacted data. In addition, the random distribution may be determined
31 in accordance with a particular seed value that is further required to recover the data in

1 original form. The seed value basically serves as an access code or password to retrieve the
2 data, while enabling storage of data in a secure manner. Thus, the present invention provides
3 data storage with enhanced security and fault tolerance.

4 **OBJECTS AND SUMMARY OF THE INVENTION**

5 Accordingly, it is an object of the present invention to distribute data in a random
6 fashion across storage media to provide enhanced data security and fault tolerance.

7 It is another object of the present invention to randomly distribute data across storage
8 media based on a seed value that serves as a security measure to selectively enable access to
9 the stored data.

10 Yet another object of the present invention to store data across storage media in a
11 manner to detect and correct errors within the data.

12 The aforesaid objects may be achieved individually and/or in combination, and it is
13 not intended that the present invention be construed as requiring two or more of the objects
14 to be combined unless expressly required by the claims attached hereto.

15 According to the present invention, a computer system including software manipulates
16 data prior to storage on storage media disposed within a system storage device. The data is
17 initially stored in a sequential manner within a first array. A random number within a pseudo-
18 random number sequence is generated for each sequential bit within the first array and serves
19 as an index to identify a storage location or position for the associated bit within a second
20 array. The sequence of random numbers is generally a non-repeating sequence that is
21 generated in accordance with a seed value. The data from the first array is thus distributed in
22 a pseudo-random fashion across the second array in accordance with the generated pseudo-
23 random number sequence. The second array is subsequently stored in a sequential manner
24 on the storage media, thereby randomly distributing the data across that media.

25 In order to retrieve the data in original form, the entire contents of the storage media
26 are retrieved and stored in a third array. The pseudo-random number sequence is reproduced
27 in accordance with the seed value with each generated random number serving as an index
28 within the third array to retrieve a corresponding data bit. The data bits are retrieved in
29 accordance with the reproduced pseudo-random number sequence and stored in temporary
30 storage to recover the data in original form. The seed value for the random number generator
31 may further serve to maintain the data in a secure fashion since the seed value is required to

1 distribute the data in a pseudo-random fashion and recover the data in original form. Further,
2 the random data distribution limits errors due to media defects to only a few bits within
3 several files, where the data bits may be recovered with conventional error correction
4 techniques.

5 The present invention may be employed with storage media having greater storage
6 capacities than the available computer system Random Access Memory (RAM) by
7 distributing data in a pseudo-random fashion as described above within sequential sections
8 of the media or by utilizing operating system virtual memory to accommodate the storage
9 media capacity. In addition, plural instances of randomly distributed data may be stored on
10 the storage media to enhance error detection and recovery. Since each data bit is stored on
11 the storage media in plural instances, the data bit value appearing within a predetermined
12 quantity of instances may be considered the appropriate data bit value. Thus, the system may
13 detect and correct errors within the data in accordance with the appropriate data bit values.

14 The above and still further objects, features and advantages of the present invention
15 will become apparent upon consideration of the following detailed description of specific
16 embodiments thereof, particularly when taken in conjunction with the accompanying drawings
17 wherein like reference numerals in the various figures are utilized to designate like
18 components.

19 **BRIEF DESCRIPTION OF THE DRAWINGS**

20 Fig. 1 is a view in perspective of an exemplary computer system for storing data in a
21 pseudo-random fashion across storage media and retrieving the data in original form
22 according to the present invention.

23 Figs. 2a - 2b are a procedural flow chart illustrating the manner in which the computer
24 system distributes data in a pseudo-random fashion across storage media according to the
25 present invention.

26 Fig. 3 is a procedural flow chart illustrating the manner in which the computer system
27 retrieves data in original form from storage media having the data distributed thereon in
28 a pseudo-random fashion according to the present invention.

29 **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS**

30 An exemplary computer system of the present invention for storing data in a pseudo-
31 random fashion across storage media and retrieving that data in original form is illustrated in

Fig. 1. Specifically, computer system 10 is typically implemented by a conventional personal or other computer system preferably equipped with a display or monitor 12, a base 14 (e.g., including the processor, memories and internal or external communication devices (e.g., modem, network cards, etc.)), a keyboard 16 and optional mouse 18 or other input device. Base 14 may further include one or more internal or external storage devices or disk drives 20 (e.g., drives accommodating CD-ROMs, floppy diskettes, Zip disks, etc.) to store data or information. These drives typically accommodate removable storage media (e.g., CD-ROMs, floppy diskettes, Zip disks, etc.) preferably employing a rotatable disk having plural tracks and sectors defined therein to store information in a sequential manner. Computer system 10 includes software for facilitating pseudo-random data distribution across the media surface and associated data retrieval therefrom and appropriate components (e.g., processor, disk storage or hard drive, etc.) having sufficient processing and storage capabilities to effectively execute the software. The computer system preferably includes a Windows environment, but may alternatively utilize any of the major platforms (e.g., Linux, Macintosh, Unix, OS2, etc.). The computer system, under software control, implements the data storage system of the present invention for distributing data in a pseudo-random fashion across storage media and subsequently retrieving that data in original form.

The present invention is typically utilized with storage devices having read and write capabilities and employing a rotatable disk-based media with plural tracks and sectors defined therein for storing data (e.g., CD-ROM, floppy diskette, Zip disk, etc.) as described above. The computer system may be implemented by any processing system or embedded device capable of interfacing a storage device or disk drive. Computer system 10 includes software to store data in a pseudo-random fashion across storage media disposed within a computer system storage device and retrieve that data from the storage media in original form as described below. Basically, the present invention evenly distributes data bits across the surface of the selected storage media. When data bits for a particular file are distributed in this fashion, the entire media or disk is generally required to fail to lose all the data stored thereon. For example, if five percent of a disk surface having data stored thereon in accordance with the present invention is damaged, five percent of data in each of the stored files is lost. However, the lost data is recoverable via conventional error correction techniques. With respect to sequential storage of data, all data in five percent of the stored

1 files is lost, without possibility of recovery. The present invention benefits from the
2 localization of physical disk damage to a portion of the disk. Thus, the present invention
3 enables recovery of lost data due to deep scratches or gouges (e.g., even circular along track
4 paths) or other severe disk damage (e.g., a hole drilled through the media) that occupy a low
5 percentage of disk surface area.

6 The manner in which the computer system stores data in a pseudo-random fashion
7 across storage media disposed within a computer system storage device is illustrated in Figs.
8 2a - 2b. Initially, data is selected by a user for storage on storage media disposed within a
9 system storage device. The selected data is loaded into an array or other data structure,
10 typically a Random Access Memory (RAM) buffer array, at step 30. For example, if one
11 hundred twenty-eight files, each having ten thousand bytes of data, are to be stored, the files
12 (e.g., 1,280,000 bytes of data) are placed in the buffer array. A sequence counter is initialized
13 at step 32, while the system determines at step 34 an exponential value of two (e.g., the value
14 of two raised to a particular power) that is closest to and exceeds the bit capacity of the
15 storage media. For example, a 3.5 inch floppy diskette has a storage capacity of
16 approximately 1.44 megabytes or 11.52 megabits (i.e., 1.44 megabytes multiplied by eight
17 since each byte contains eight bits). The exponential value of two closest to and greater than
18 11.52 megabits is 2^{24} . Thus, the sequence counter for the 3.5 inch floppy disk may be
19 implemented by a twenty-four bit binary counter having a maximum of 16,777,216 states.
20 The exponential value defines a range upper limit for generating a sequence of non-repeating
21 random numbers as described below. The computer system automatically ascertains the
22 storage capacity of the storage media in order to determine the exponential value. The
23 sequence counter maintains a quantity of random numbers generated to indicate when each
24 number within the exponential value range (e.g., typically the range of zero to the exponential
25 value minus one) has been generated.

26 The system prompts a user at step 36 to enter a seed value for generating a pseudo-
27 random number sequence. If a seed value is entered as determined at step 38, the entered seed
28 value is utilized by a pseudo-random number generator to generate a non-repeating sequence
29 of random numbers as described below. Otherwise, a default seed value is retrieved at step
30 40 for utilization by the random number generator to generate the sequence. The seed value
31 may be any value within the exponential value range. For example, since 2^{24} or 16,777,216

1 is the exponential value greater than the storage capacity of the 3.5 inch floppy disk (e.g.,
2 having a storage capacity of 1.44 megabytes) as described above, any of approximately 2^{24}
3 seed values may be utilized. Further, since a Zip disk has a storage capacity of approximately
4 one-hundred megabytes, any value within an excess of 1.6 billion seed values may be utilized.
5 The default seed value is used for general interoperability between systems (e.g., the default
6 seed value is the same to enable reproduction of the pseudo-random sequence and retrieval
7 of information across plural systems), while custom seed values may be utilized to effectively
8 encrypt the data for enhanced data security as described below.

9 Data bits within the buffer array are sequentially retrieved and stored in a pseudo-
10 random fashion within a holding array or other data structure having a capacity similar to that
11 of the intended storage media (e.g., 1.44 megabytes in the case of a 3.5 inch floppy diskette).
12 In particular, the system determines the presence of unprocessed data within the buffer array
13 at step 42. If unprocessed data is present, the system retrieves the next data bit within the
14 buffer array at step 46. Otherwise, a filler or pad bit is retrieved by the system at step 44. The
15 filler bit is basically employed when storing a quantity of data less than the storage media
16 capacity to facilitate completion of the pseudo-random number sequence and to ensure usage
17 of virtually the entire storage media (e.g., the filler bit is typically stored at residual locations
18 within the storage media not receiving data from the buffer array). The random number
19 generator utilizes the seed value at step 48 to generate a random number within a non-
20 repeating pseudo-random number sequence, while the sequence counter is incremented at step
21 50 to maintain the quantity of random numbers generated within the sequence. The generated
22 random numbers in the sequence have values within the exponential value range and each
23 generally serve as an index into the holding array to identify the storage location for an
24 associated retrieved or filler bit as described below. The random number may be generated
25 by an operating system or via a call to a software function (e.g., a standard function call within
26 the 'C' programming language). A pseudo-random sequence is utilized to enable
27 reproduction of particular sequences in response to utilization of corresponding seed values.
28 In other words, a particular seed value enables generation of the same corresponding pseudo-
29 random number sequence. This feature enables the sequence to be reproduced for retrieving
30 data in original form as described below.

1 When the generated random number is less than or equal to the media bit storage
2 capacity (e.g., the random number is within the media storage capacity space) as determined
3 at step 52, the system stores the associated retrieved or filler bit within the holding array
4 location identified by the generated random number at step 56. If the generated random
5 number exceeds the media bit capacity (e.g., the random number is outside the media storage
6 capacity space), the system determines at step 54 whether or not the sequence counter is
7 greater than or equal to the exponential value (e.g., whether or not each of the values within
8 the exponential value range has been generated). If the sequence counter is less than the
9 exponential value (e.g., additional values have not been generated), the system generates the
10 next random number within the sequence at step 48. In effect, a generated random number
11 exceeding the media bit capacity identifies a storage location beyond the bounds of the storage
12 media and holding array. Accordingly, the system generates successive random numbers
13 within the pseudo-random number sequence until ascertaining a value within the media bit
14 and holding array capacities, while generated values outside the media and holding array
15 capacities are basically ignored.

16 When the counter attains the determined exponential value, each value within the
17 exponential value range has been generated. Thus, each retrieved data bit has been stored in
18 a location within the holding array identified by a corresponding random number. When the
19 system determines at steps 54 or 58 that the counter has attained a value equal to or greater
20 than the determined exponential value (e.g., all values within the exponential value range have
21 been generated), the holding array is written sequentially to the storage media at step 60. The
22 physical writing of the data to the storage media is in a track/sector format, however, the
23 arrangement of data within the holding array basically serves to produce a random data
24 distribution of the original data across the storage media. The resulting distribution evenly
25 distributes each bit from a file across the surface of the storage media. Although defects
26 encountered on the storage media surface may affect a few bits of several files, these errors
27 may be routinely corrected with conventional error correction techniques, such as Reed-
28 Solomon.

29 The manner in which the computer system retrieves data in original form from storage
30 media having the data distributed thereon in a pseudo-random fashion is illustrated in Fig. 3.
31 Initially, the contents of the storage media are retrieved and stored in a working array or other

1 data structure at step 70. The working array has a capacity to store the entire contents of the
2 storage media. A sequence counter is initialized at step 72, while the system determines at
3 step 74 an exponential value of two (e.g., the value of two raised to a particular power) that
4 is closest to and exceeds the media bit capacity in substantially the same manner described
5 above. The computer system automatically ascertains the storage capacity of the storage
6 media in order to determine the exponential value.

7 The system prompts a user at step 76 for the seed value utilized to generate the
8 pseudo-random number sequence for randomly distributing data across the storage media.
9 The seed value basically serves as a password to enhance data security since the data order
10 is virtually impossible to attain without the original seed value utilized to initially distribute
11 that data. If a seed value is entered as determined at step 78, the entered seed value is utilized
12 by a pseudo-random number generator to generate the pseudo-random number sequence as
13 described below. Otherwise, a default seed value is retrieved at step 80 for use by the random
14 number generator to produce the sequence. The default seed value is generally a common
15 value to facilitate sharing of the storage media between plural locations or systems as
16 described below.

17 The random number generator utilizes the seed value at step 82 to generate a random
18 number within a non-repeating pseudo-random number sequence and having a value within
19 the exponential value range as described above. The random number generator basically
20 reproduces the same sequence utilized to randomly distribute the data, thereby enabling the
21 system to retrieve that data in the proper order. The sequence counter is incremented at step
22 84 to maintain the quantity of random numbers generated. If the generated random number
23 is less than or equal to the media bit storage capacity as determined at step 86, the system
24 retrieves the bit within the working array from a location identified by the generated random
25 number, and stores the retrieved bit in a sequential manner within a temporary storage area
26 on the system at step 88. When the generated random number exceeds the media bit capacity,
27 the system determines at step 87 whether or not the sequence counter is greater than or equal
28 to the exponential value (e.g., whether or not each of the values within the exponential value
29 range has been generated). If the counter is less than the exponential value (e.g., additional
30 values have not been generated), the system generates the next random number within the
31 sequence at step 82. In effect, a generated random number exceeding the media bit capacity

1 identifies a storage location beyond the bounds of the storage media and working array.
2 Accordingly, the system repeatedly generates successive random numbers in the pseudo-
3 random number sequence until ascertaining a value within the media storage and working
4 array capacities to identify the next sequential data bit, while generated random numbers
5 outside the media storage and working array capacities are basically ignored.

6 When the sequence counter reaches the determined exponential value, each value
7 within that range has been generated. Thus, each bit stored on the media has been retrieved
8 and placed in the appropriate order within the temporary storage area. When the system
9 determines that the sequence counter has attained a value equal to or greater than the
10 exponential value (e.g., all values within the exponential value range have been generated)
11 at steps 87 or 90, the files in the temporary storage area are selectively recovered at step 92.
12 Basically, a file structure is displayed to the user to enable selection of files. The system
13 retrieves the selected files from the temporary storage area for permanent storage to user
14 specified locations.

15 The present invention may further be employed with storage media having capacities
16 that exceed the capacity of available RAM or the buffer array associated with the computer
17 system. In particular, the system basically partitions the storage media space into memory
18 blocks each having a storage capacity similar to or less than that of the available RAM or
19 buffer array. The data is similarly partitioned into sections having a quantity of data sufficient
20 to be accommodated by a corresponding memory block. Each data section is distributed in
21 a random fashion across the corresponding memory block in substantially the same manner
22 described above. The memory blocks having randomly distributed data are sequentially
23 stored on the storage media. For example, the present invention may accommodate a Zip
24 disk having a capacity of one-hundred megabytes by storing data on the disk in five sequential
25 blocks, each having a five megabyte capacity and data distributed in a random fashion within
26 that block. This provides the advantage of rapid formatting, but distributes data in a random
27 fashion only within the blocks (e.g., not across the storage media).

28 Alternatively, operating system virtual memory may be utilized for storage media
29 having capacities exceeding those of the computer system available RAM or buffer array.
30 Basically, the virtual memory feature of an operating system interchanges RAM memory
31 blocks with hard disk blocks to simulate additional system RAM. The system may randomly

1 distribute data across the storage media as described above by utilizing virtual memory (e.g.,
2 the simulated RAM) to accommodate the storage media capacity. This enables random data
3 distribution across the storage media, but reduces formatting efficiency due to overhead
4 generated by the memory block interchanges providing the virtual memory.

5 The present invention may further enhance error recovery performance by providing
6 additional error correction schemes. Initially, the entire capacity of the storage media is
7 utilized by the present invention regardless of the quantity of data to be stored. Accordingly,
8 storage space on the storage media is available when the quantity of data to be stored is less
9 than the storage media capacity. For example, the system may store two-hundred kilobytes
10 of data across a floppy diskette (e.g., having a 1.44 megabyte storage capacity) in a random
11 fashion as described above, thereby enabling the floppy diskette to have 1.24 megabytes of
12 available storage (e.g., the available storage typically stores filler bits as described above).
13 The present invention may utilize the available storage to repeatedly store the data in a
14 random fashion as described above (e.g., additional copies of the data may be placed and/or
15 appended to the data within the buffer array and processed for storage across the media as
16 described above). A header is placed toward the initial location of the storage media to
17 indicate the block size of the originally stored data. This information is utilized to determine
18 when stored data starts to repeat within the storage media.

19 The system utilizes a consensus scheme to identify incorrect or erroneous bits within
20 the data. Specifically, each data bit is typically stored on the storage media at plural locations
21 due to the storage of the additional data copies. The value for a data bit is retrieved from each
22 instance of the data (e.g., from the data and corresponding additional data copies) stored on
23 the storage media. The retrieved values are compared where the value for the data bit
24 appearing within a predetermined quantity of data instances may be considered to be the
25 correct value. For example, when a data bit is stored at six locations on the storage media
26 (e.g., when the data and five additional copies thereof are stored on the storage media), the
27 value of the data bit that appears within four or more data instances may be considered to be
28 the correct value. If a data bit appears twice on the media (e.g., when the data and one
29 additional copy thereof is stored on the storage media), the value of the data bit within the data
30 instance having no errors (e.g., whether or not recoverable) may be considered to be the

1 correct value. The stored data bits may be subsequently updated with the appropriate values
2 for error recovery.

3 Operation of the present invention is described. The present invention is typically
4 implemented by a personal computer having a Windows type environment and a software
5 module executable by a user. The user executes the software module and selects files for
6 storage, preferably within a pop-up or other window. Subsequently, the user indicates the
7 destination storage device having storage media disposed therein to receive the selected data.
8 The system automatically determines the storage media capacity and the pseudo-random
9 number sequence length (e.g., exponential value) sufficient to perform the random data
10 distribution. The user is prompted to enter a seed value. If no seed value is entered, a default
11 seed value is utilized as described above. The system performs the data distribution and
12 storage as described above. This may be accomplished as a background task (e.g., in a
13 Windows environment). The status of the operation may be displayed, while a completion
14 indication is typically displayed upon termination of the operation.

15 In order to retrieve randomly distributed information from storage media, the user
16 executes the software module. The system prompts the user for the seed value utilized to
17 store the original data. If no seed value is entered, the default seed value is utilized as
18 described above. The system retrieves the data and places the data in original form into a
19 temporary storage area on the system (e.g., a system hard drive) as described above. A file
20 structure is displayed to the user to facilitate selection of files for permanent recovery (e.g.,
21 storage to a desired location).

22 The software of the present invention is typically implemented in the 'C' programming
23 language, however, any suitable high or low level language may be utilized, especially those
24 that may be ported to all common computers. The present invention may be utilized with any
25 removable disk architecture having any storage capacity in substantially the same manner
26 described above. Further, storage devices or drives may be manufactured to utilize the
27 random data distribution and retrieval features of the present invention as a native interface.

28 It will be appreciated that the embodiments described above and illustrated in the
29 drawings represent only a few of the many ways of implementing a method and apparatus for
30 secure and fault tolerant data storage.

1 The computer system of the present invention may be implemented by any personal
2 or other type of computer system (e.g., IBM-compatible, Apple, Macintosh, laptop, palm
3 pilot, main frame, minicomputer, microcomputer, etc.) or processing device or circuitry
4 capable of interfacing a storage device. The computer system of the present invention may
5 include any commercially available operating system (e.g., Windows, OS/2, Unix, Linux,
6 etc.). The computer system of the present invention may further include any commercially
7 available or custom software, any quantity of any types of input devices (e.g., keyboard,
8 mouse, voice recognition, etc.) and any quantity of any types of storage devices (e.g., CD-
9 ROM drive, DVD drive, floppy diskette drive, Zip drive, hard disk drive, etc.). The present
10 invention may be utilized with any quantity of any types of removable or non-removable
11 storage media (e.g., magnetic media, optical media, magneto-optical media, tapes, disks,
12 memory devices or circuits, etc.) of any shape, size or storage capacity. The media may store
13 data in any fashion and include any quantity of data storing mechanisms (e.g., tracks, sectors,
14 etc.). The present invention may be utilized to store any type or quantity of information, and
15 may distribute the data in any desired portions or units having any quantity of bits (e.g., bit,
16 byte, word, etc.). Further, the present invention may accommodate any quantity of drives
17 having the same or different storage media.

18 It is to be understood that the software for the computer system of the present
19 invention may be implemented in any desired computer language and could be developed by
20 one of ordinary skill in the computer arts based on the functional descriptions contained in the
21 specification and flow charts illustrated in the drawings. The computer system of the present
22 invention may alternatively be implemented by hardware or other processing circuitry. The
23 various functions of the computer system may be distributed in any manner among any
24 quantity of computer or processing systems or circuitry and/or among any quantity of software
25 and/or hardware modules. The software and/or algorithms described above and illustrated in
26 the flow charts may be modified in any manner that accomplishes the functions described
27 herein. For example, the counters may maintain the amount of data stored instead of the
28 exponential value to control termination of storage and retrieval. The holding array in this
29 case is initialized with filler bits, while the present invention only stores or retrieves data in
30 selected locations to reduce processing (e.g., the loop is executed for the amount of data
31 instead of the exponential value range).

1 The present invention may reside on any communications network (e.g., LAN, WAN,
2 Internet, Intranet, etc.), while end-user computer systems may include any conventional or
3 other communications devices to communicate over the network to utilize or access the
4 present invention and receive the randomly distributed data or data in original form.

5 The data storage system of the present invention may be implemented by any quantity
6 of computer systems, and may reside on a server, end-user or other third-party computer
7 system or any combination of these computer systems. The software of the present invention
8 may be available on recorded medium (e.g., floppy diskettes, CD-ROM, memory devices,
9 etc.) for use on stand-alone systems or systems connected by a network, or may be
10 downloaded (e.g., in the form of carrier waves, packets, etc.) to systems from a network.

11 The arrays and temporary storage area of the present invention may be of any quantity
12 and of any suitable storage capacity. The arrays and area may alternatively be implemented
13 by any type of data structure (e.g., queue, stack, linked list, record, etc.) or memory device
14 (e.g., RAM, hard disk, etc.), and may be stored within any suitable system memory or storage
15 device at any desired locations. The sequence counter may be implemented by any quantity
16 of any type of hardware or software counter and may maintain any desired values. The seed
17 value may be any value within any desired range. The filler or pad bit may be of any desired
18 value, and may include any quantity of bits to accommodate the particular units of data being
19 stored (e.g., bit, byte, word, etc.).

20 The present invention may utilize any type of conventional or other pseudo-random
21 or random number generator to generate the sequence. The present invention may utilize any
22 conventional or other error correction techniques to recover lost data or correct erroneous
23 data. The present invention may store data on the media in any fashion and may be utilized
24 in combination with any error correction schemes and/or data distribution schemes (e.g.,
25 offset, interleaving, etc.). The present invention may utilize any types of prompts (e.g., line
26 prompts, windows, menus, etc.) to query the user and receive any type of information. The
27 present invention may display files or other data in any manner or arrangement (e.g., list,
28 window, etc.) to facilitate selection by a user via any suitable input device (e.g., mouse, voice,
29 keyboard, etc.).

30 The present invention may partition the media and data in any fashion to accommodate
31 the media storage capacity. The memory blocks and data sections may be of any quantity and

1 include any desired storage capacity. The memory blocks may be stored on the media in any
2 desired fashion (e.g., sequential, interleaved, etc.). The present invention may utilize any
3 virtual or other system memory to accommodate the media storage capacity. The present
4 invention may store any quantity of additional instances of data on the storage media in any
5 desired manner or arrangement and utilize any type of consensus or other scheme to detect
6 and correct errors. The consensus scheme may utilize any predetermined quantity of data
7 instances (e.g., a majority, a super majority, mathematical formula, etc.) or any error threshold
8 for instances having errors (e.g., the instance data value is utilized if the instance has a
9 quantity of errors below the threshold) to determine the appropriate value for data. The
10 header may include any desired information to indicate characteristics of the stored data.

11 The present invention is not limited to the specific applications disclosed herein, but
12 may be utilized in substantially the same manner described above to encrypt or secure data.
13 For example, the present invention may be utilized to encrypt data for any communication
14 applications. The data may be arranged as described above and transmitted across any type
15 of network (e.g., LAN, WAN, wireless, packet, etc.) where the receiver utilizes the seed value
16 to decrypt the transmission.

17 From the foregoing description it will be appreciated that the invention makes
18 available a novel method and apparatus for secure and fault tolerant data storage wherein data
19 is distributed in a pseudo-random fashion across storage media to store the data in a secure
20 manner and enable recovery of data in the event of media defects.

21 Having described preferred embodiments of a new and improved method and
22 apparatus for secure and fault tolerant data storage, it is believed that other modifications,
23 variations and changes will be suggested to those skilled in the art in view of the teachings
24 set forth herein. It is therefore to be understood that all such variations, modifications and
25 changes are believed to fall within the scope of the present invention as defined by the
26 appended claims.

What is Claimed is:

1 1. A system for distributing data across storage media in a manner to facilitate
2 fault tolerant data storage, said system comprising:

3 a storage unit to store data on storage media received within said storage unit, wherein
4 said storage media has a memory space encompassing a storage capacity of said storage
5 media; and

6 a processor to manipulate and facilitate storage of said data across said storage media
7 in a manner to provide fault tolerant storage of said data with respect to defects arising within
8 said storage media, said processor including:

9 a data retrieval module to selectively retrieve data for storage on said storage
10 media, wherein said data includes a plurality of data portions;

11 a data distribution module to distribute said data portions within said memory
12 space in a pseudo-random fashion; and

13 a storage module to facilitate storage of said distributed data on said storage
14 media via said storage unit to produce a pseudo-random distribution of said data across said
15 storage media.

1 2. The system of claim 1 wherein said data distribution module includes:

2 a sequence generation module to generate a sequence of random numbers, wherein
3 said random numbers are selectively associated with corresponding data portions, and wherein
4 each associated random number has a value within a range of said memory space and
5 identifies a location within that space to store said corresponding data portion; and

6 a data arrangement module to store each said data portion within said memory space
7 location identified by said associated random number.

1 3. The system of claim 2 wherein said sequence generation module produces said
2 sequence in accordance with a seed value.

1 4. The system of claim 3 wherein said data distribution module further includes:

2 a user module to prompt a user to enter said seed value and to retrieve said seed value
3 entered by said user;

4 wherein said sequence generation module includes:

5 a seed module to produce said sequence in accordance with said entered seed
6 value in response to said user entering that value; and

7 a default seed module to retrieve a default seed value and to produce said
8 sequence in accordance with said default seed value in response to said user failing to enter
9 said seed value.

1 5. The system of claim 1 wherein said data distribution module further includes:

2 a data load module to store said data portions sequentially in a first storage space; and

3 a data transfer module to retrieve data portions sequentially from said first storage
4 space and to store those data portions in said pseudo-random fashion within a second storage
5 space having a storage capacity of said storage media;

6 wherein said storage module includes:

7 a write module to facilitate sequential storage of said distributed data portions
8 from said second storage space to said storage media, thereby distributing said data across said
9 storage media in said pseudo-random fashion.

1 6. The system of claim 3 wherein said processor further includes:

2 a data recovery module to retrieve data portions from said storage media and arrange
3 said data portions in a manner to produce said data in original form.

1 7. The system of claim 6 wherein said data recovery module includes:

2 a sequence reproduction module to reproduce said sequence of random numbers
3 utilized for said pseudo-random data distribution, wherein said random numbers are
4 selectively associated with locations within said memory space, and wherein each associated
5 random number identifies a location within said memory space to retrieve a corresponding
6 data portion; and

7 a data organization module to retrieve each data portion from said memory space
8 location identified by said associated random number, wherein said data portions are retrieved
9 in order of said sequence to arrange said data in original form.

1 8. The system of claim 7 wherein said data recovery module further includes:
2 a user module to prompt a user to enter said seed value utilized for said pseudo-
3 random distribution and to retrieve said seed value entered by said user;
4 wherein said sequence reproduction module includes:
5 a seed module to reproduce said sequence in accordance with said entered
6 seed value in response to said user entering that value; and
7 a default seed module to retrieve a default seed value and to reproduce said
8 sequence in accordance with said default seed value in response to said user failing to enter
9 said seed value.

1 9. The system of claim 6 wherein said data recovery module further includes:
2 a data load module to retrieve data portions sequentially from said storage media and
3 to store those data portions sequentially in a first storage space; and
4 a restoration module to retrieve data portions in the order of and from locations within
5 said first storage space identified by said random number sequence to produce said data in
6 original form and to sequentially store said ordered data portions in a temporary storage area.

1 10. The system of claim 9 wherein said data recovery module further includes:
2 a data selection module to facilitate selection of said restored data by said user; and
3 a data transfer module to retrieve said selected data from said temporary storage area
4 and to store said selected data at a user specified location.

1 11. The system of claim 1 wherein said data distribution module includes:
2 a partition module to selectively partition said memory space into data blocks and to
3 selectively partition said data into data sections, wherein each data section is associated with
4 a corresponding data block; and

5 a block distribution module to distribute data portions of each data section in said
6 pseudo-random fashion within an associated data block;

7 wherein said storage module includes:

8 a block storage module to facilitate sequential storage of each said data block
9 onto said storage media.

1 12. The system of claim 5 wherein said processor includes virtual memory and at
2 least one of said first and second storage spaces is at least partially defined within said virtual
3 memory.

1 13. The system of claim 1 wherein said processor further includes:
2 a copy module to facilitate storage of additional instances of said data onto said
3 storage media in response to said storage media having available storage space;
4 a verification module to determine the value of a particular data portion within each
5 said data instance;
6 a value module to determine the appropriate value for said data portion in accordance
7 with said values retrieved from said data instances; and
8 an update module to correct said data portion with said determined value.

1 14. The system of claim 13 wherein said value module includes:
2 a consensus module to determine said appropriate value based on a data portion value
3 appearing within a predetermined quantity of data instances.

1 15. The system of claim 13 wherein said value module includes:
2 a consensus module to determine said appropriate value in accordance with a data
3 instance having a quantity of errors below an error threshold.

1 16. The system of claim 6 wherein said data recovery module includes:
2 a security module to enable said retrieval of said data in original form in response to
3 a user entering an appropriate password.

1 17. A program product apparatus having a computer readable medium with
2 computer program logic recorded thereon for distributing data across storage media in a
3 manner to facilitate fault tolerant data storage, wherein said storage media has a memory
4 space encompassing a storage capacity of said storage media, said program product apparatus
5 comprising:

6 a data retrieval module to selectively retrieve data for storage on said storage media,
7 wherein said data includes a plurality of data portions;

8 a data distribution module to distribute said data portions within said memory space
9 in a pseudo-random fashion; and

10 a storage module to facilitate storage of said distributed data on said storage media to
11 produce a pseudo-random distribution of said data across said storage media.

1 18. The apparatus of claim 17 wherein said data distribution module includes:

2 a sequence generation module to generate a sequence of random numbers, wherein
3 said random numbers are selectively associated with corresponding data portions, and wherein
4 each associated random number has a value within a range of said memory space and
5 identifies a location within that space to store said corresponding data portion; and

6 a data arrangement module to store each said data portion within said memory space
7 location identified by said associated random number.

1 19. The apparatus of claim 18 further including:

2 a data recovery module to retrieve data portions from said storage media and arrange
3 said data portions in a manner to produce said data in original form.

1 20. The apparatus of claim 19 wherein said data recovery module includes:

2 a sequence reproduction module to reproduce said sequence of random numbers
3 utilized for said pseudo-random data distribution, wherein said random numbers are
4 selectively associated with locations within said memory space, and wherein each associated
5 random number identifies a location within said memory space to retrieve a corresponding
6 data portion; and

7 a data organization module to retrieve each data portion from said memory space
8 location identified by said associated random number, wherein said data portions are retrieved
9 in order of said sequence to arrange said data in original form.

1 21. The apparatus of claim 17 wherein said data distribution module includes:
2 a partition module to selectively partition said memory space into data blocks and to
3 selectively partition said data into data sections, wherein each data section is associated with
4 a corresponding data block; and
5 a block distribution module to distribute data portions of each data section in said
6 pseudo-random fashion within an associated data block;
7 wherein said storage module includes:
8 a block storage module to facilitate sequential storage of each said data block
9 onto said storage media.

1 22. The apparatus of claim 17 further including:
2 a copy module to facilitate storage of additional instances of said data onto said
3 storage media in response to said storage media having available storage space;
4 a verification module to determine the value of a particular data portion within each
5 said data instance;
6 a value module to determine the appropriate value for said data portion in accordance
7 with said values retrieved from said data instances; and
8 an update module to correct said data portion with said determined value.

1 23. The apparatus of claim 22 wherein said value module includes:
2 a consensus module to determine said appropriate value based on a data portion value
3 appearing within a predetermined quantity of data instances.

1 24. The apparatus of claim 22 wherein said value module includes:
2 a consensus module to determine said appropriate value in accordance with a data
3 instance having a quantity of errors below an error threshold.

1 25. The apparatus of claim 19 wherein said data recovery module includes:
2 a security module to enable said retrieval of said data in original form in response to
3 a user entering an appropriate password.

1 26. A carrier signal having computer program logic embedded therein for
2 distributing data across storage media in a manner to facilitate fault tolerant data storage,
3 wherein said storage media has a memory space encompassing a storage capacity of said
4 storage media, said carrier signal comprising:

5 a data retrieval module to selectively retrieve data for storage on said storage media,
6 wherein said data includes a plurality of data portions;

7 a data distribution module to distribute said data portions within said memory space
8 in a pseudo-random fashion; and

9 a storage module to facilitate storage of said distributed data on said storage media to
10 produce a pseudo-random distribution of said data across said storage media.

1 27. The carrier signal of claim 26 wherein said data distribution module includes:
2 a sequence generation module to generate a sequence of random numbers, wherein
3 said random numbers are selectively associated with corresponding data portions, and wherein
4 each associated random number has a value within a range of said memory space and
5 identifies a location within that space to store said corresponding data portion; and

6 a data arrangement module to store each said data portion within said memory space
7 location identified by said associated random number.

1 28. The carrier signal of claim 27 further including:

2 a data recovery module to retrieve data portions from said storage media and arrange
3 said data portions in a manner to produce said data in original form.

1 29. The carrier signal of claim 28 wherein said data recovery module includes:

2 a sequence reproduction module to reproduce said sequence of random numbers
3 utilized for said pseudo-random data distribution, wherein said random numbers are
4 selectively associated with locations within said memory space, and wherein each associated

5 random number identifies a location within said memory space to retrieve a corresponding
6 data portion; and

7 a data organization module to retrieve each data portion from said memory space
8 location identified by said associated random number, wherein said data portions are retrieved
9 in order of said sequence to arrange said data in original form.

1 30. The carrier signal of claim 26 wherein said data distribution module includes:
2 a partition module to selectively partition said memory space into data blocks and to
3 selectively partition said data into data sections, wherein each data section is associated with
4 a corresponding data block; and

5 a block distribution module to distribute data portions of each data section in said
6 pseudo-random fashion within an associated data block;

7 wherein said storage module includes:

8 a block storage module to facilitate sequential storage of each said data block
9 onto said storage media.

1 31. The carrier signal of claim 26 further including:

2 a copy module to facilitate storage of additional instances of said data onto said
3 storage media in response to said storage media having available storage space;

4 a verification module to determine the value of a particular data portion within each
5 said data instance;

6 a value module to determine the appropriate value for said data portion in accordance
7 with said values retrieved from said data instances; and

8 an update module to correct said data portion with said determined value.

1 32. The carrier signal of claim 31 wherein said value module includes:

2 a consensus module to determine said appropriate value based on a data portion value
3 appearing within a predetermined quantity of data instances.

1 33. The carrier signal of claim 31 wherein said value module includes:

2 a consensus module to determine said appropriate value in accordance with a data
3 instance having a quantity of errors below an error threshold.

1 34. The carrier signal of claim 28 wherein said data recovery module includes:
2 a security module to enable said retrieval of said data in original form in response to
3 a user entering an appropriate password.

1 35. A system for distributing data across storage media in a manner to facilitate
2 fault tolerant data storage, wherein said storage media has a memory space encompassing a
3 storage capacity of said storage media, said system comprising:

4 data retrieval means for selectively retrieving data for storage on said storage media,
5 wherein said data includes a plurality of data portions;

6 data distribution means for distributing said data portions within said memory space
7 in a pseudo-random fashion; and

8 storage means for storing said distributed data on said storage media to produce a
9 pseudo-random distribution of said data across said storage media.

1 36. The system of claim 35 wherein said data distribution means includes:
2 sequence generation means for generating a sequence of random numbers, wherein
3 said random numbers are selectively associated with corresponding data portions, and wherein
4 each associated random number has a value within a range of said memory space and
5 identifies a location within that space to store said corresponding data portion; and

6 data arrangement means for storing each said data portion within said memory space
7 location identified by said associated random number.

1 37. The system of claim 36 further including:
2 data recovery means for retrieving data portions from said storage media and arranging
3 said data portions in a manner to produce said data in original form.

1 38. The system of claim 37 wherein said data recovery means includes:

2 sequence reproduction means for reproducing said sequence of random numbers
3 utilized for said pseudo-random data distribution, wherein said random numbers are
4 selectively associated with locations within said memory space, and wherein each associated
5 random number identifies a location within said memory space to retrieve a corresponding
6 data portion; and

7 data organization means for retrieving each data portion from said memory space
8 location identified by said associated random number, wherein said data portions are retrieved
9 in order of said sequence to arrange said data in original form.

1 39. The system of claim 35 wherein said data distribution means includes:

2 partition means for selectively partitioning said memory space into data blocks and
3 for selectively partitioning said data into data sections, wherein each data section is associated
4 with a corresponding data block; and

5 block distribution means for distributing data portions of each data section in said
6 pseudo-random fashion within an associated data block;

7 wherein said storage means includes:

8 block storage means for storing each said data block sequentially onto said
9 storage media.

1 40. The system of claim 35 further including:

2 copy means for storing additional instances of said data onto said storage media in
3 response to said storage media having available storage space;

4 verification means for determining the value of a particular data portion within each
5 said data instance;

6 value means for determining the appropriate value for said data portion in accordance
7 with said values retrieved from said data instances; and

8 update means for correcting said data portion with said determined value.

1 41. The system of claim 40 wherein said value means includes:

2 consensus means for determining said appropriate value based on a data portion value
3 appearing within a predetermined quantity of data instances.

1 42. The system of claim 40 wherein said value means includes:
2 consensus means for determining said appropriate value in accordance with a data
3 instance having a quantity of errors below an error threshold.

1 43. The system of claim 37 wherein said data recovery means includes:
2 security means for enabling said retrieval of said data in original form in response to
3 a user entering an appropriate password.

1 44. A method of distributing data across storage media in a manner to facilitate
2 fault tolerant data storage, wherein said storage media has a memory space encompassing a
3 storage capacity of said storage media, said method comprising the steps of:

4 (a) selectively retrieving data for storage on said storage media, wherein said data
5 includes a plurality of data portions;

6 (b) distributing said data portions within said memory space in a pseudo-random
7 fashion; and

8 (c) storing said distributed data on said storage media to produce a pseudo-random
9 distribution of said data across said storage media.

1 45. The method of claim 44 wherein step (b) further includes:

2 (b.1) generating a sequence of random numbers, wherein said random numbers are
3 selectively associated with corresponding data portions, and wherein each associated random
4 number has a value within a range of said memory space and identifies a location within that
5 space to store said corresponding data portion; and

6 (b.2) storing each said data portion within said memory space location identified by
7 said associated random number.

1 46. The method of claim 45 wherein step (b.1) further includes:

2 (b.1.1) generating said sequence of random numbers in accordance with a seed value.

1 47. The method of claim 46 wherein step (b.1) further includes:

2 (b.1.1) prompting a user to enter said seed value and retrieving said seed value entered
3 by said user;

4 (b.1.2) generating said sequence in accordance with said entered seed value in
5 response to said user entering that value; and

6 (b.1.3) retrieving a default seed value and generating said sequence in accordance with
7 said default seed value in response to said user failing to enter said seed value.

1 48. The method of claim 44 wherein step (b) further includes:

2 (b.1) storing said data portions sequentially in a first storage space; and

3 (b.2) retrieving data portions sequentially from said first storage space and storing
4 those data portions in said pseudo-random fashion within a second storage space having a
5 storage capacity of said storage media;

6 wherein step (c) further includes:

7 (c.1) storing said distributed data portions from said second storage space
8 sequentially onto said storage media, thereby distributing said data across said storage media
9 in said pseudo-random fashion.

1 49. The method of claim 46 further including:

2 (d) retrieving data portions from said storage media and arranging said data
3 portions in a manner to produce said data in original form.

1 50. The method of claim 49 wherein step (d) further includes:

2 (d.1) reproducing said sequence of random numbers utilized for said pseudo-random
3 data distribution, wherein said random numbers are selectively associated with locations
4 within said memory space, and wherein each associated random number identifies a location
5 within said memory space to retrieve a corresponding data portion; and

6 (d.2) retrieving each data portion from said memory space location identified by said
7 associated random number, wherein said data portions are retrieved in order of said sequence
8 to arrange said data in original form.

1 51. The method of claim 50 wherein step (d.1) further includes:

2 (d.1.1) prompting a user to enter said seed value utilized for said pseudo-random
3 distribution and retrieving said seed value entered by said user;

4 (d.1.2) reproducing said sequence in accordance with said entered seed value in
5 response to said user entering that value; and

6 (d.1.3) retrieving a default seed value and reproducing said sequence in accordance
7 with said default seed value in response to said user failing to enter said seed value.

1 52. The method of claim 49 wherein step (d) further includes:

2 (d.1) retrieving data portions sequentially from said storage media and storing those
3 data portions sequentially in a first storage space; and

4 (d.2) retrieving data portions in the order of and from locations within said first
5 storage space identified by said random number sequence to produce said data in original
6 form and sequentially storing said ordered data portions in a temporary storage area.

1 53. The method of claim 52 wherein step (d) further includes:

2 (d.3) facilitating selection of said restored data by said user; and

3 (d.4) retrieving said selected data from said temporary storage area and storing said
4 selected data at a user specified location.

1 54. The method of claim 44 wherein step (b) further includes:

2 (b.1) selectively partitioning said memory space into data blocks and selectively
3 partitioning said data into data sections, wherein each data section is associated with a
4 corresponding data block; and

5 (b.2) distributing data portions of each data section in said pseudo-random fashion
6 within an associated data block;

7 wherein step (c) further includes:

8 (c.1) storing each said data block sequentially onto said storage media.

1 55. The method of claim 48 wherein at least one of said first and second storage
2 spaces is at least partially defined within virtual memory.

- 1 56. The method of claim 44 further including:
2 (d) storing additional instances of said data onto said storage media in response
3 to said storage media having available storage space;
4 (e) determining the value of a particular data portion within each said data
5 instance;
6 (f) determining the appropriate value for said data portion in accordance with said
7 values retrieved from said data instances; and
8 (g) correcting said data portion with said determined value.

- 1 57. The method of claim 56 wherein step (f) further includes:
2 (f.1) determining said appropriate value based on a data portion value appearing
3 within a predetermined quantity of data instances.

- 1 58. The method of claim 56 wherein step (f) further includes:
2 (f.1) determining said appropriate value in accordance with a data instance having
3 a quantity of errors below an error threshold.

- 1 59. The method of claim 49 wherein step (d) further includes:
2 (d.1) enabling said retrieval of said data in original form in response to a user
3 entering an appropriate password.

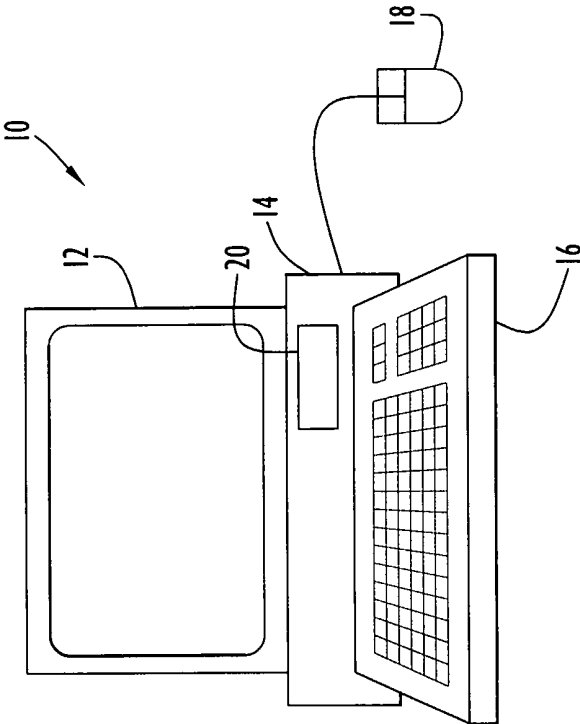


FIG. 1

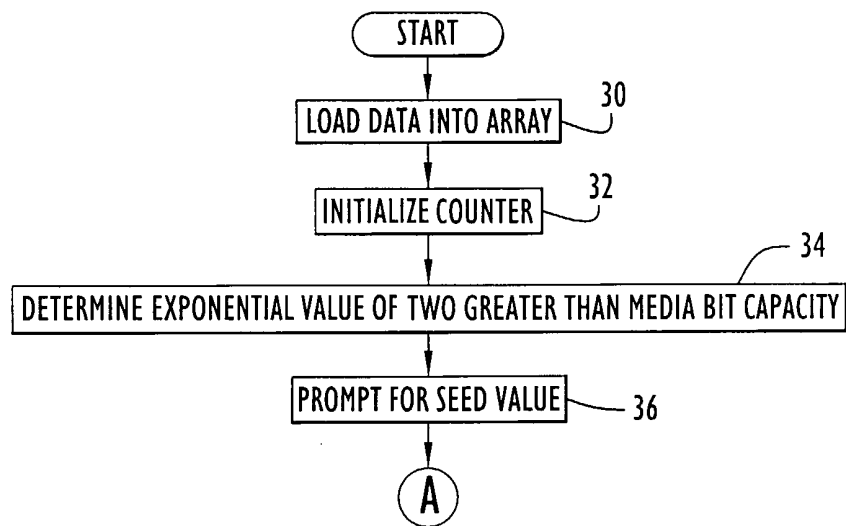


FIG2a

