(19) **United States**

(12) **Patent Application Publication** (10) Pub. No.: US 2017/0293535 A1

NILES et al. (43) **Pub. Date:** **Oct. 12, 2017**

(54) **SYSTEM AND METHOD FOR BACKING UP DATA**

(71) Applicant: **FALCONSTOR, INC.**, Melville, NY (US)

(72) Inventors: **Ronald S. NILES**, Teaneck, NJ (US); **Wai T. LAM**, Jericho, NJ (US)

(73) Assignee: **FALCONSTOR, INC.**

(21) Appl. No.: **15/633,044**

(22) Filed: **Jun. 26, 2017**

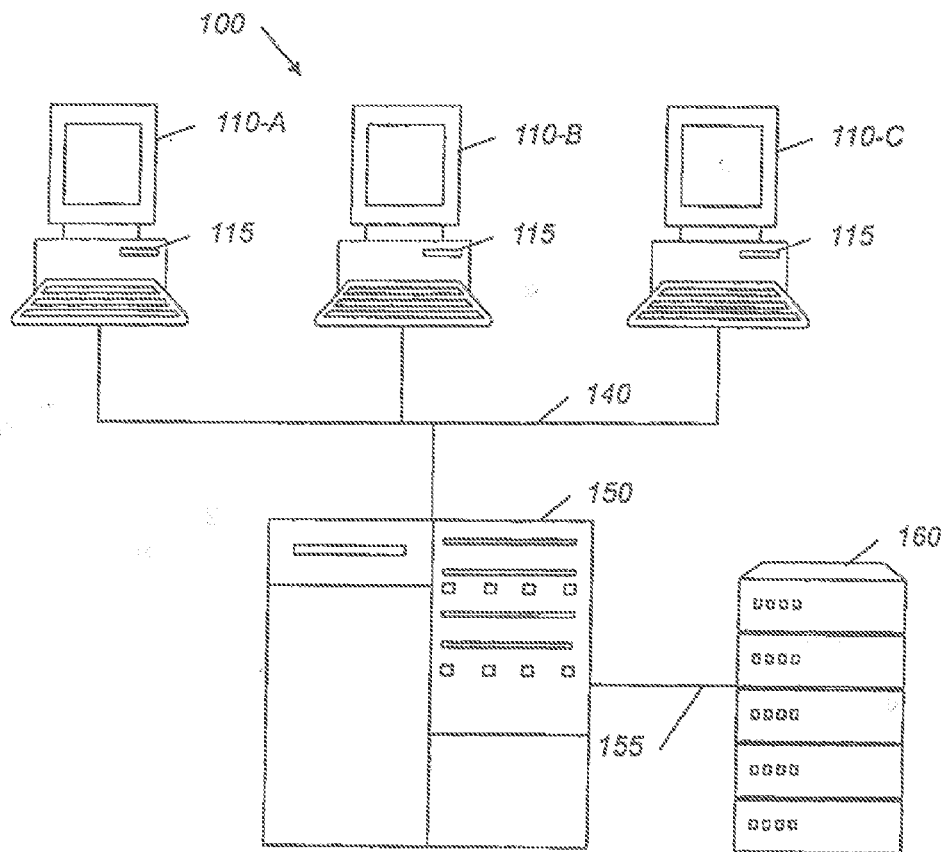**Related U.S. Application Data**

(60) Continuation of application No. 14/709,197, filed on May 11, 2015, now Pat. No. 9,690,794, which is a continuation of application No. 13/758,705, filed on Feb. 4, 2013, now Pat. No. 9,032,171, which is a continuation of application No. 13/134,065, filed on May 27, 2011, now Pat. No. 8,370,594, which is a continuation of application No. 11/440,910, filed on May 25, 2006, now Pat. No. 7,954,157, which is a division of application No. 10/348,834, filed on Jan. 22, 2003, now Pat. No. 7,055,008.

Publication Classification

(51) **Int. Cl.**
G06F 11/14	(2006.01)
H04L 9/06	(2006.01)
G06F 21/56	(2006.01)

(52) **U.S. Cl.**
CPC *G06F 11/1464* (2013.01); *G06F 21/564* (2013.01); *H04L 9/0643* (2013.01)

(57) **ABSTRACT**

A hash-optimized backup system and method takes data blocks and generates a probabilistically unique digital fingerprint of the content of each data block using a substantially collision-free algorithm. The process compares the generated fingerprint to a database of stored fingerprints if the generated fingerprint matches a stored fingerprint, the data block is determined to already have been backed up, and therefore does not need to be hacked up again. Only if the generated fingerprint does not match a stored fingerprint is the data block backed up, at which point the generated fingerprint is added to the database of stored fingerprints. Because the algorithm is substantially collision-free, there is no need to compare actual data content if there is a hash-value match. The process can also be used to audit software license compliance, inventory software, and detect computer-file tampering such as viruses and malware.
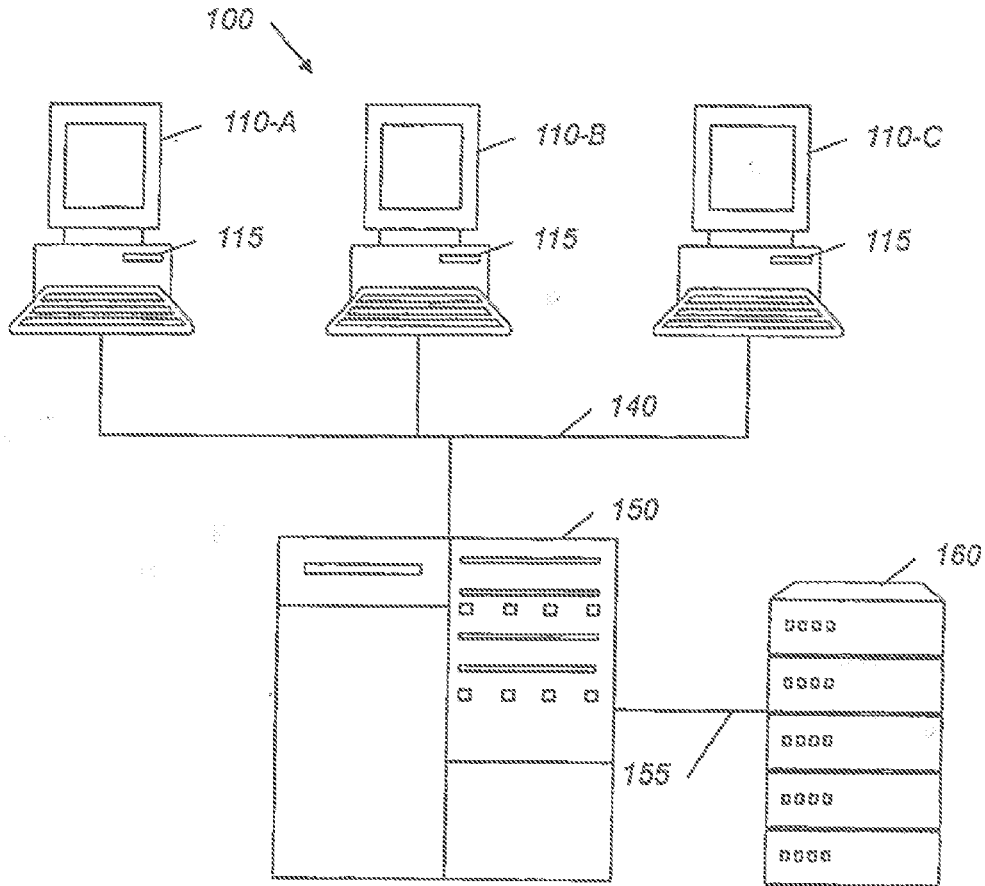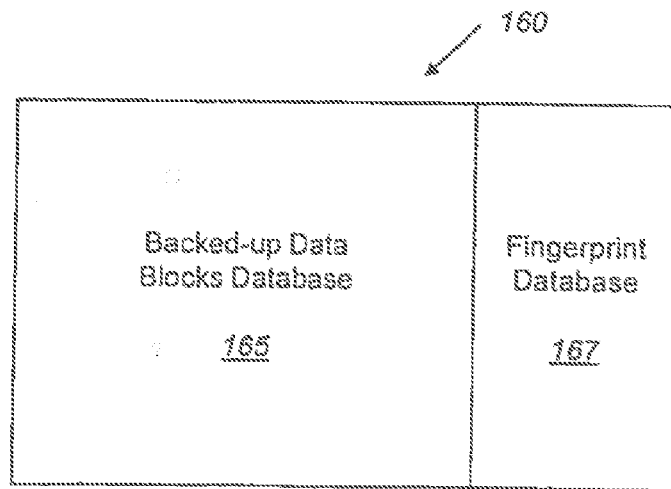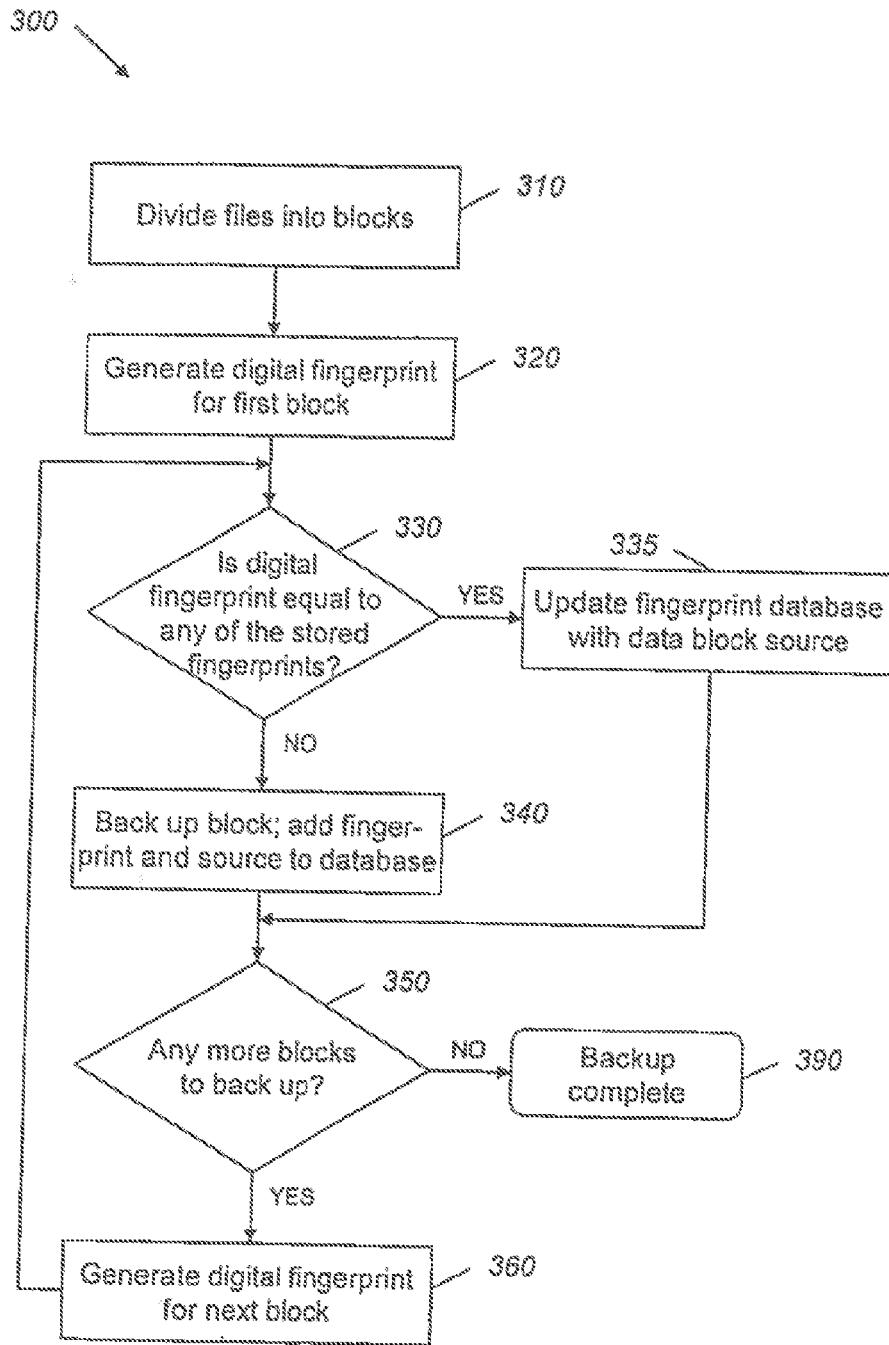
100

110-A   *110-B*   *110-C*

115   *115*   *115*

140

150

160

155

FIG. 1

160

Backed-up Data
Blocks Database

165

Fingerprint
Database

167

FIG. 2

300

Divide files into blocks — *310*

Generate digital fingerprint
for first block — *320*

330
Is digital
fingerprint equal to
any of the stored
fingerprints?

YES → *335* Update fingerprint database
with data block source

NO

Back up block; add finger-
print and source to database — *340*

350
Any more blocks
to back up?

NO → Backup
complete — *390*

YES

Generate digital fingerprint
for next block — *360*

FIG. 3

160

165   167

Data Block #	Fingerprint (410)	Comp./Loc. (420-1)	Comp./Loc. (420-2)	Comp./Loc. (420-3)	Comp./Loc. (420-...)
1	FP 1	A Loc	R Loc		
2	FP 2	A Loc	B Loc	D Loc	
3	FP 3	A Loc			. . .
4	FP 4	A Loc	A Loc		
.	.	.	.	.	.
A	FP A	A Loc			
A+1	FP A+1	B Loc	C Loc	D Loc	
A+2	FP A+2	B Loc	D Loc		. . .
A+3	FP A+3	B Loc	G Loc	H Loc	
.	.	.	.	.	.
B	FP B	B Loc			
B+1	FP B+1	C Loc	H Loc	M Loc	. . .
.	.	.	.	.	.
Q	FP Q	Q Loc			
Q+1	FP Q+1	R Loc	R Loc	R Loc	. . .
.	.	.	.	.	.
R	FP R	R Loc			. . .

FIG. 4

SYSTEM AND METHOD FOR BACKING UP DATA

RELATED APPLICATIONS

[0001] The present application is a U.S. patent application Ser. No. 14/709,197, which was filed on May 11,2015 and will issue on Jun. 27, 2017 bearing patent application Ser. No. 9,690,794, which is a continuation of U.S. patent application Ser. No. 13/758,705, which was filed on Feb. 4, 2013 and issued on May 12,2015 bearing U.S. Pat. No. 9,032,171; which is a continuation of U.S. patent application Ser. No. 13/134,065, which was filed on May 27,2011 and issued on Feb. 5, 2013 bearing U.S. Pat. No. 8,370,594; which is a continuation of U.S. patent application Ser. No. 11/440,910, which was filed on Mar. 25,2006 and issued on May 31,2011 bearing U.S. Pat. No. 7,954,157; which is a division of U.S. patent application Ser. No. 10/348,834, which was filed on Jan. 22,2003 and issued on May 30,2006 bearing U.S. Pat. No. 7,055,008; all of which are assigned to the assignee of the present invention and are incorporated by reference herein.

FIELD OF THE INVENTION

[0002] Embodiments of the invention relate generally to a system and method backing up and storing unique data through the use of digital fingerprints. Other embodiments of the invention relate to auditing of license restrictions, inventorying an enterprise, and detecting file tampering, including virus and malware detection, through the use of digital fingerprints.

BACKGROUND OF THE INVENTION

[0003] Over time in a typical computer environment, large amounts of data are typically written to and retrieved from storage devices connected to the computer. As more data are exchanged with the storage devices. It becomes increasingly difficult for the data owner to reproduce these data if the storage devices fell. One way of protecting data is by backing up the data to backup media (e.g., tapes or disks). Such backup is typically performed manually or automatically at preset Intervals using backup software. The backup media are then stored away in a safe location. Continuous backups result In a collection of backup media. Because of space constraints, however, backup media are generally only kept for a finite period of time, and then the oldest backup medium is written over with the newest backup data. The length of this time period, or "backup window," thus depends on the number of backup media and the amount of data each medium contains.

[0004] Making full backups of a system is very time-consuming. One way to reduce the need for full backups, thereby increasing the backup window, is to perform incremental or differential backups between full backups. An incremental backup backs up only files that have changed since the last full or incremental backup. A differential backup backs up every file that has changed since the last full backup. The difference between the two is shown in the following example. Assume a full backup is performed weekly, e.g., every Saturday night, and the incremental or differential backup Is performed nightly. In order to restore data corrupted or lost on Friday, a system using incremental backup requires the full backup from the previous Saturday, as well as each incremental backup from the intervening five

nights. A system using differential backup also requires the full backup from the previous Saturday, but only requires the differential backup made on Thursday night, because that, includes all the files changed since the previous Saturday. Thus, each incremental backup takes less time and stores less data than a differential backup, but a differential backup allows corrupted or lost data to be restored more easily and quickly.

[0005] One method of performing an incremental or differential backup is to set an "archive flag" for each file after it is backed up. If the file is changed (or is new), the archive flag is reset. Then, during the subsequent backup, the backup software only looks for files whose archive flags have been reset.

[0006] One disadvantage of incremental and differential backup is that the scope of the archive flag Is limited to an individual computer. When backing up two or more different computers, such as those found in a networks the files on both computers must be backed up. An archive flag system backs up the files on a first computer, and any identical files loaded on the second computer will have their archive flag reset on that computer, indicating that those files should also be backed up. Such backup of identical files on the second computer is a duplication of space and effort, however, because only one backup copy of any specific file need be available. Another disadvantage of the archive flag system is that if a large file is modified only slightly, the archive flag will be reset, no matter how small the change is, and the entire file will have to be backed up again.

[0007] This latter limitation is addressed by U.S. Pat. No. 5,559,991 to Kanfi, issued Sep. 24,1996. That patent discloses performing an incremental backup by dividing a file into blocks, generating a signature for each block, and backing up the block if the signature differs from a signature generated for an earlier version of the block. If the signature is the same, no backup is necessary. The backup computer (i.e., the computer controlling the backup) associates each block with the file from which it came. The advantage of this backup process is that if a large file is only slightly modified, only the modified blocks will be backed up, not the whole file. However, the process is limited to backing up versions of specific, named flies on individual computers, even if the identical file (or data block) is located on the same computer but under a different name or it is located on another computer on the same network.

[0008] Another reference attempts to solve this last limitation. U.S. Pat. No. 6,374,266 to Shnelvar, issued Apr. 16,2002, discloses dividing data to be backed up into data units, generating a hash value for each data unit, and backing up the data unit if the hash value does not match a hash value saved in a table. If the generated hash value does match one in the table, the method compares the actual data in the data unit to the data associated with the hash value in the table. If the data are the same, the data in the data unit are not backed up; if the data are not the same, then the data unit is hacked up, and the table is updated to reflect the addition. This method is able to back up data from multiple computers and does not back up identical data units that reside on different computers.

[0009] The method of the Shnelvar patent, however, is not efficient because whenever there is a hash-value match, that method compares the actual data in the data unit to the data associated with the hash value in the table. In Shnelvar, a hash-value match can occur when the data units giving rise

to the hash values are the same or when there is a hash-value "collision"—when the data units are different but the generated hash values are the same. Shnelvar performs a data comparison because of the possibility of hash-value collisions. However, in a system in which much of the data does not change between backups, there will be numerous hash-value matches, and the backup will spend a significant amount of time comparing the actual data, especially if the data are not local to the computer being backed up, or are only available over a low-speed

SUMMARY OF THE INVENTION

[0010] The inefficiency of the Shnelvar patent can be avoided by using a substantially collision-free hash-optimized backup process. A hash-optimized backup process takes data blocks and generates a probabilistically unique digital fingerprint of the content of that data block. The process compares the generated fingerprint to a database of stored fingerprints and, if the generated fingerprint matches a stored fingerprint, the data block is determined to already have been backed up, and therefore does not need to be backed up again. Only if the generated fingerprint does not match a stored fingerprint Is the data block backed up, at which point the generated fingerprint is added to the database of stored fingerprints. Because the algorithm is substantially collision-free, there Is no need to compare actual data content if there is a hash-value match.

[0011] More particularly, a method of the present invention generates a digital fingerprint for the data, block using a substantially collision-free algorithm, compares the digital fingerprint to digital fingerprints stored in a database, and if the digital fingerprint does not match one of the stored digital fingerprints, the method backs up the data block and adds to the database the digital fingerprint and the location of the data block. If the digital fingerprint matches one of the stored digital fingerprints, the method updates the database to associate the digital fingerprint with the data block location. Preferably, the substantially collision-free algorithm is a hash function such as MD5 or SHA-1, because these hash functions have been used and studied over several years, during which time their collision-resistance has been widely tested and documented.

[0012] A system of the present invention includes a backup server connected to a source computer via a communication path. The backup server has access to a digital fingerprint database. The source computer generates a digital fingerprint of the data block using a substantially collision-free algorithm, and the backup server backs up the data block if the digital fingerprint does not match any digital fingerprint stored in the database. In such a case, the backup server may add to the database the digital fingerprint, and the location of the data block. If the digital fingerprint matches a digital fingerprint stored in the database, the backup server updates the database to associate the digital fingerprint with the data block location. The database may be stored in a storage device connected to the backup server. There may also be a storage device connected to the backup server for storing the backed-up data blocks. The communication path between the source computer and the backup server may be a network, in which case the network may be, e.g., a local area network (LAN), a wide-area network (WAN), a metropolitan area network (MAN), or an internetwork of computers, such as the Internet. A "storage device" can mean a

disk drive, a memory-based storage system, an optical disk, or a logical partition within a data storage device.

[0013] Another method in accordance with the invention allows for auditing license restrictions of a computer program in an enterprise computing environment. This method generates a digital fingerprint of at least one file in the computer program using a substantially collision-free algorithm, generates a digital fingerprint for each file on each computer in the enterprise using the substantially collision-free algorithm, compares the digital fingerprints from the enterprise files with the digital fingerprint of the computer program file, and counts the number of fingerprint matches. A similar method in accordance with the invention allows for inventorying a computer program in an enterprise computing environment In both of these methods, the file may be divided into data blocks, and a digital fingerprint generated for each data block.

[0014] Digital fingerprints could also be used to implement system management and prevent users from installing new programs without authority, or at least to monitor such installations.

[0015] Further methods in accordance with the invention allow for detecting file tampering, such as a computer virus, on a computer. One method generates a digital fingerprint, for each file on the computer using a substantially collision-free algorithm and then compares the digital fingerprints of the computer files with digital fingerprints of the computer files generated when the files were previously saved. If the digital fingerprint of that file differs from the digital fingerprint generated when that file was previously saved, a computer virus or other tampering may exist on the file.

[0016] Another method generates a digital fingerprint of a computer virus itself using a substantially collision-free algorithm, generates a digital fingerprint for each file on the computer using the substantially collision-free algorithm, and compares the digital fingerprints from the computer files -with the digital fingerprint of the computer virus, if any digital fingerprint of the computer files matches the digital fingerprint of the computer virus, the virus exists on the computer.

[0017] These methods for detecting computer viruses can be generalized to detect any unwanted type of "malicious software" (a.k.a. "malware"), such as adware, spyware, worms, and other software installed without permission.

[0018] Embodiments of the invention recognizes that many files within a networked computing environment are identical and therefore much backup efficiency can be realized If only one copy of each file is backed op. The invention also recognizes that files can be uniquely identified using a digital fingerprint. Further efficiencies are realized because generating the digital fingerprint can be performed quickly—more quickly than the backing up of the files themselves. The invention also operates on subsections of files (i.e., data blocks), thus increasing backup efficiency by only backing up portions of large files that have changed, without backing up the whole file.

[0019] Additional advantages of the invention will be set forth in the description which follows, and In part will be apparent from the descriptions or may be learned by practice of the invention. The advantages of the invention may be realized and obtained by means of the instrumentalities and combinations particularly pointed out In the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] The accompanying drawings, in which like reference numerals represent like parts, are incorporated in and constitute a part of the specification. The drawings illustrate presently preferred embodiments of the invention and, together with the general description given above and the detailed description given below, serve to explain the principles of the invention.

[0021] FIG. 1 is a block diagram illustrating a system for backing op data in accordance with an embodiment of the present invention;

[0022] FIG. 2 is a block diagram illustrating storage within a storage device in accordance with an embodiment of the present invention;

[0023] FIG. 3 Is a flowchart depicting backing up data In accordance with an embodiment of the present invention; and

[0024] FIG. 4 is a schematic depicting the contents of the backup databases in accordance with an embodiment of the present Invention.

DETAILED DESCRIPTION

[0025] The present invention generates a probabilistically unique digital fingerprint of the content of each data block of a file. If the generated fingerprint matches a fingerprint that is already stored, the block is not backed up and the digital fingerprint is associated with the data block. If no match occurs, the block is backed up and the digital fingerprint is associated with the data block.

[0026] FIG. 1 illustrates an embodiment of the present invention. System 100 includes any number of computers 110 (three of which, 110-A, 110-B, 110-C, are shown in FIG. 1) connected to backup server 50 via network. 140. As shown in FIG. 1, computers 110 are optionally connected to each other over network 140. Connected to backup server 150 via communication path 155, which may be, for example, a Fibre Channel or SCSI connection, is storage device 160. Backup server 150 is a central computer whose main function is to back up or archive data stored on other computers and servers on a computer network. Network 140 may be, for example, a LAN, a WAN, a MAN, or an internetwork of computers, such as the Internet. Storage device 160 acts as the backup (or archive) device for computers 110 and has a large capacity (e.g., terabytes). In order to satisfy the backup needs for system 100, there may be more than one storage device 160 connected to backup server 150. In system 100, computers 110 typically include local storage 115 (e.g., a hard disk drive) for saving data and files between backups. Alternatively, computers 110 may have no local storage and be part of a storage area network (SAM) in which case another server (not shown) connected to network 140, such as a file server or a data server, stores data and files on a primary storage device connected to that server.

[0027] As shown In FIG. 2, storage device 160 includes storage for at least backed-up data blocks in database 165 (a "data" database) and digital fingerprints in database 167 (a "metadata" database). Database 16 includes digital fingerprints and information relating (e.g., cross-referencing) the digital fingerprints to the data blocks. Database 167 also includes the location (i.e., which computer 110 and the location on that computer) from which the data blocks came, so that the data can fee restored if the local storage is lost or

destroyed. Because data blocks appearing on more than one computer are not backed up, storage device 160 can back up much more data and files than prior art backup systems.

[0028] Backup server 150 typically includes software that can schedule and Initiate periodic backups. If computers 110 include local storage 115, an "agent" residing on each computer 110 scans all the files on that computer, dividing the files Into data blocks and computing digital fingerprints for each block. (Alternatively, the agent may reside on backup server ISO and perform these tasks via network 140. If computers 110 do not include local storage, an agent residing on the data or file server of the SAN performs these tasks.) For each block, each computer 110 contacts backup server 150, which compares the digital fingerprint of that block to those In database 167 and determines if there is a match. If there is a match, there is no need to copy the data, block to storage device 160 because the data block is already there. In such a case, database 167 will be updated to include a cross-reference from the digital fingerprint to the current data block's source or location. If there is no match, the data block is copied to storage device 160, and the digital fingerprint and data block location are added to database 167. The backup thus consists of backed-up data blocks database 165, a list of the digital fingerprints, and the data block or blocks associated with each of the digital fingerprints (along with the data block's origin information).

[0029] The flowchart 300 in FIG. 3 shows how the backup process operates. First, in step 310, each file may be divided into data blocks. The size of these blocks may be fixed or variable, depending on the operating system or the system administrator's preferences. Fixed blocks are easier to manage, but may waste space. Variable blocks make a better use of the available backup space, but are somewhat more difficult to keep track of. In addition, the size of the blocks may vary from file to file. For instance, one option may be to have each file contain a set number of blocks, N—the size of each block from a larger file of size S1 would be S1/N and the size of each block from a smaller file of size S2 would be S2/N, where S1/N>S2/N. A special case of a variable-sized block Is the whole file itself (i.e., where N=1), however, it Is likely more advantageous to have smaller-sized blocks in order to avoid having to save large files that change only slightly between backups. In addition, the size of the blocks may be limited by the requirements of the specific algorithm used to create the digital fingerprint.

[0030] Once the files are divided into data blocks, step 320 generates a digital fingerprint for each data block. The algorithm for generating the fingerprint is preferably a hash function. A bash function performs a transformation on an Input and returns a number having a fixed length—the hash value. Properties of a hash function as used in the present invention are that it should (1) be able to take a variable-sized input and generate a fixed-size output, (2) compute the hash value relatively easily and quickly for any input value, and (3) be substantially (or "strongly") collision-free, Hash functions satisfying these criteria include the MD5 and SHA-1 algorithms, although others are available or will be available in the future.

[0031] The MD5 ("message digest 5") algorithm was created by Professor Ronald Rivest of MIT and RSA Laboratories, It generates a 16-byte (128-bit) hash value. It is designed to run on 32-bit computers. Earlier algorithms created by Professor Rivest, MD2 and MD4, developed in 1989 and 1990, respectively, also produce 128-bit hash

values, but have been shown not to be substantially collision-free. MD5 was created in 1991 and is slightly slower than MD4, but more secure. MD5 is substantially collision-free. Using MD5, fingerprints may be generated at high speed on most computers.

[0032] The SHA-1 ("secure hash algorithm") algorithm was developed in 1994 by the U.S. National Institute of Standards and Technology (NIST). It generates a 20-byte (160-bit) hash value. The maximum input length of a data block to the SHA-1 algorithm is 2^{64} bits (~1.8×10^{19} bits). The design of SHA-1 is similar to that of MD4 and MD5, but because its output is larger, it is slightly slower than MD5, but more collision-free.

[0033] Before performing the first backup for backup server 150, data database 165 and metadata database 167 are empty. Thus, there can be no fingerprint matching as called for in step 330. Instead, flowchart 300 proceeds directly to step 340 to back up the data block in backed-up data blocks database 165 and record in database 167 the digital fingerprint and the source of the data block (i.e., the file path, including which computer 110 and where the data block resides on the computer). Step 350 asks whether any more data blocks need to be backed up. If so, then step 360 generates the digital fingerprint for the next block In the same manner as was done in step 320. Now, because data database 165 and metadata database 167 are not empty, step 330 compares the digital fingerprint of the data block being backed up to the stored digital fingerprint. Because the hash function generating the digital fingerprint is substantially collision-free, if there is a match, it is assumed that the data block has been backed up already and therefore step 335 only has to update database 167 to associate that digital fingerprint with the source of the data block. If there is no match, step 340 backs up the data block In backed-up data blocks database 165 and records In database 167 the digital fingerprint and the source of the data block. This loop of steps 360, 330, 335/340, and 350 continues until there are no more data blocks on any of the computers 110 to back up. In that case, step 350 returns NO and the backup is complete in step 390.

[0034] FIG. 4 shows one way of illustrating the contents of databases 165 and 167 during a backup of computers 110-A to 110-R. Database 165 includes a list of all the data blocks that have been backed up. Database 167 includes column 410 for the digital fingerprint for each data block and columns 420-1, 420-2, 420-3, etc., for the source(s) (i.e., computer and block location) of those data blocks. Thus, the first data block, 1, is placed in database 165, and its digital fingerprint, FP 1, is placed in database 167, along with the source computer, 110-A, and the location, "Loc". "Loc" may be a memory or a cluster location in the source computer, and uniquely identifies the data block's source location at the time of backup. Consecutive data blocks 1, 2,3, etc. do not have to come from adjacent locations in the source computer. For example, a file may be made up of 200 data blocks that will be placed in database 165 consecutively, but their locations on the source computer do not have to be consecutive.

[0035] The next data block whose fingerprint does not match any fingerprint in column 410 is numbered data block 2 and placed in database 165. Its fingerprint, FP 2, is placed in column 410 along with the location in computer 110-A. If the fingerprint of the data block does match an existing fingerprint, the data block Itself Is not backed up again, but

the location In computer 110-A is noted in column 420-2 (see, e.g., the entry for data block 4, which is located in two places in computer 110-A). Although only three columns 420 are shown in FIG. 4, there can be many columns 420, each one recording the location of the same data block at different locations on the same computer or on different computers. Thus, data block 1 also exists on computer 110-R. and data block 2 also exists on computers 110-B and 110-D. The process continues until all of the A (where A is an integer) unique data blocks from computer 110-A have been backed up.

[0036] The next block to be backed up comes from computer 110-B. If this block is unique, it is numbered data block A+1 and is placed in database 165. Its fingerprint, FP A+1, is then placed in column 410 along with the location in computer 110-B. If the fingerprint of the data block matches an existing fingerprint, however, the data block itself is not backed up again, but the location in computer 110-B is noted in the next appropriate column 420 (see, e.g., the entry for data block 2, which is located in computers 110-A and 110-B). Thus, data block A+1 also exists on computers 110-C and 110-D, data block A+2 also exists on computer 110-D, and data block A+3 also exists on computers 110-G and 110-H. This process continues until all of the B A (where B is an integer) unique data blocks from computer 110-B have been backed up.

[0037] The next block to be backed up comes from computer 110-C. If this block is unique, it is numbered data block B+1 and is placed in database 165. Its fingerprint, FP B+1, is then placed in column 410 along with the location in computer 110-C. If the fingerprint of the data block matches an existing fingerprint, however, the data block itself is not backed up again, but the location in computer 110-C is noted in the next appropriate column 420 (see, e.g., the entry for data block A+1, which is located in computers 110-B, 110-C, and 110-D). Thus, data block B+1 also exists on computers 110-H and 110-M. This process continues until all of the C-B (where C is an integer) unique data blocks from computer 110-C have been backed up. The process then continues in the same manner for each of the computers from 110-D to 110-R.

[0038] The process of the present invention is more efficient than that of the prior art because the substantially collision-free nature of the hash function allows a hash-value match to represent a data block that is already backed up, and the actual data do not have to be compared to confirm that that is so.

[0039] In additions because each backup generates a digital fingerprint for all the data blocks on the system being backed up, it is a fall backup, unlike the traditional incremental or differential backup, and the fingerprint database 167 and backed-up data blocks database 165 exist permanently. In the event of lost or destroyed data, data restoration can be performed using only the most recent backup.

[0040] The present invention has many applications. One application is within an enterprise environment, such as a networked office, in which a local area network 140 connects many computers 110. These computers 110 are generally configured similarly, typically having the same operating system files and basic application (e.g., word processors, e-mail, spreadsheet presentation, etc.) programs. Because of the redundancy of these files, there is no need to back up all of the files on all of the computers—doing so would take up much-needed backup space. Using the pres-

ent invention, however, allows an organization to make one backup copy of these identical files and then note the locations of those files on the other computers **110**. If a single backup server **150** were used to back up all of the computers in the organization, the database would very quickly build up a list of the most common duplicated files. Once the Initial backup is made in backed-up data blocks database **165**, the data blocks (and the files made up by those blocks) never have to be backed up again. When new computers **110** are deployed, it is likely that all of the initial files on those computers are already contained in the backed-up data blocks database **165**, thus making the initial backup of a new or existing computer extremely efficient, especially in a centralized computing environment.

[0041] This application can be extended to a MAN or a WAN, which network extends beyond the physical boundaries of a floor or a building. Because digital fingerprints can be efficiently delivered to backup server **150** for comparison to database **167** and because not every data block must be backed up, the present Invention Is especially suited for such remote operation.

[0042] The invention, however, is not limited to backing up files from, computers that are related to each other, such as those within an organization. The invention can be used to back up computers (related or non-related) over the Internet. In such an application, the Internet is network **140**. Even though the computers **110** may not be from the same organization, they may contain many identical files, such as operating system files and popular software packages. Each computer **110** can have a backup agent that scans the files on that computer, dividing the files into data blocks and computing digital fingerprints for each block. For each block, computer **110** transmits over network **140** the fingerprint to backup server **150**, which compares the digital fingerprint to those in database **167** and determines if there is a match. If there is a match, database **167** updates the fingerprint to include the source, but there is no need to copy the data block to storage device **160** because the data block is already there. If there is no match, the data block is transmitted over network **140** to backup server **150** and copied to storage device **160**, and the digital fingerprint and source are added to database **167**. Prior art backup methods that compare the actual data when a match is found could not be used in such an application because there is usually not enough bandwidth between computer **110** and backup server **150** to perform such a backup efficiently. Prior art schemes therefore generally contemplate having the source computer and the backup server near each other or connected by high-bandwidth lines.

[0043] Embodiments of the invention can also be used in auditing. In such an application, auditors can assess the backup record of any computer within an enterprise to determine what is on that computer, without actually having to restore a tape. Auditors could easily ensure that computers in the enterprise have the proper operating system service packs installed without having to visit those machines. If the auditors wanted to know how many computers have a specific software package installed to verify the enterprise is in compliance with licensing requirements, they can determine that by comparing the fingerprints of one or more files within the various popular software packages in question with fingerprints generated from files in the computers in the enterprise. This could be performed by generating finger-

prints for data blocks or files as a whole. Each match would count as having the particular software package in question.

[0044] This idea can be extended for use as an inventory tool. A system administrator could set up a database of the fingerprints of one or more files within the various popular software packages and then compare the database to the fingerprints of the files or data blocks of each computer in the system (or even of external computers, e.g., over the internet). Again, each match would count as having the particular software package in question.

[0045] Embodiments of the present invention can be used to manage enterprise systems. Digital fingerprints can be used to prevent users from installing new programs without authority, or at least to monitor such installations.

[0046] Embodiments of the present invention can be used to defect viruses or other file tampering. Including any unwanted type of "malicious software" (a.k.a. "malware"), such as adware, spyware, worms, and other software installed without permission. When loading a file having a known digital fingerprint, the file's fingerprint (or those of a file's data blocks) can be checked to see that it has not been changed since the file was last saved. If the fingerprint has changed, the file is likely to have been tampered with or Infected by a virus. In addition, a digital fingerprint for a virus-infected file (or data block) may be determined, thereby allowing detection of the virus by fingerprint alone.

[0047] There are thus many benefits of embodiments of the present invention, in backup, embodiments of the invention improve backup efficiency and recovery (restoration) speed by reducing backup redundancy. Already backed-up files are readily identifiable. Blocks of data smaller than a whole file are backed up, thus reducing the need to backup whole files having minor changes. In an enterprise environment, in which many of the computers have the same files, the invention eliminates the need to save all of the files on all of the computers, only the files that are unique to each computer. Data can be backed up across a network such as the Internet with relative ease and speed.

[0048] Additional advantages and modifications will readily occur to those skilled in the art. Therefore, the present invention in its broader aspects is not limited to the specific embodiments, details, and representative devices shown and described herein. Accordingly, various changes, substitutions, and alterations may be made to such embodiments without departing from the spirit or scope of the general inventive concept as defined by the appended claims.

We Claim:

1. A method for backing up data comprising:

selecting a file for backup;

dividing the file into one or more data blocks;

generating a digital fingerprint for at least one of the one or more data blocks using a substantially collision-free algorithm;

comparing the digital fingerprint to digital fingerprints stored in a database; and

if the digital fingerprint does not match one of the stored digital fingerprints, backing up the data block and adding to the database the digital fingerprint and the data block location.

2. The method of claim **1**, comprising updating the database to associate the digital fingerprint with the data block location, if the digital fingerprint matches one of the stored, digital fingerprints.

3 The method of claim **2**, wherein the algorithm comprises a hash function.

4. The method of claim **3**, wherein the hash function is MD5 or SHA-1.

5. A system for backing up data comprising:

a backup server having access to a digital fingerprint database; and

a source computer connected to the backup server via a communication path, the source computer being configured to:

select a file for backup;

divide the file into one or more data blocks;

generate a digital fingerprint of at least one of the one or more data blocks using a substantially collision-free algorithm;

wherein the backup server is configured to back up the at least one data block if the digital fingerprint does not match any digital fingerprint stored in the database.

6. The system of claim **5**, wherein the data block has a location on the source computer and the backup server is configured to add to the database the digital fingerprint and the data block location.

7. The system of claim **5**, wherein the data block has a location on the source computer and if the digital fingerprint matches a digital fingerprint stored in the database, the backup server is configured to update the database to associate the digital fingerprint with the data block location.

8. The system of claim **5**, further comprising a storage device connected to the backup server for storing the database.

9. The system of claim **5**, further comprising a storage device connected to the backup server for storing the backed-op data blocks.

10. The system of claim **5**, wherein the communication path comprises a network.

11. The system of claim **9**, wherein the network comprises a local area network, or a wide area network, and/or the Internet.

12. A method for detecting file tampering on a computer, comprising:

selecting a file;

dividing the selected file into one or more data blocks;

generating a digital fingerprint of at least one of the one or more data blocks;

generating a first digital fingerprint for the at feast one data block on the computer using a substantially collision-free algorithm at a first time;

generating second digital fingerprint for the at least one data block on the computer using the substantially collision-free algorithm at the second time after the first time;

comparing the first digital fingerprint of a data block generated at the first time with second digital fingerprint of a data block generated at the second time; and

determining whether tampering exists on a respective first file if the digital fingerprint of the at least one data block file differs from the second digital fingerprint of the corresponding at least one second data block.

13. The method of claim **12**, wherein the file tampering comprises a computer virus.

14. A method for detecting a computer virus on a computer, comprising:

generating a digital fingerprint of a computer virus using a substantially collision-free algorithm;

generating a digital fingerprint for a file on the computer using the substantially collision-free algorithm;

comparing the digital fingerprint of the computer file with the digital fingerprint of the computer virus; and

determining whether the computer file is a virus based on the comparison.

15. The method of claim **14**, further comprising dividing each file into data blocks and generating a digital fingerprint of at least one data block.

* * * * *