PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G09C 3/06	A2	(11) International Publication Number: WO 99/10859 (43) International Publication Date: 4 March 1999 (04.03.99)

(21) International Application Number: PCT/US98/17839

(22) International Filing Date: 28 August 1998 (28.08.98)

(30) Priority Data:

08/919,198	28 August 1997 (28.08.97)	US
08/919,212	28 August 1997 (28.08.97)	US
08/919,366	28 August 1997 (28.08.97)	US
08/919,203	28 August 1997 (28.08.97)	US

(71) Applicant: SYNDATA TECHNOLOGIES, INC. [US/US]; West Building, 500 Frank W. Burr Boulevard, Teaneck, NJ 07666 (US).

(72) Inventor: ORRIN, Steven, M.; 43 Conforti Avenue #77, West Orange, NJ 07052 (US).

(74) Agents: KOCH, Robert, J. et al.; Fulbright & Jaworski L.L.P., 801 Pennsylvania Avenue N.W., Washington, DC 20004 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: ENCRYPTION SYSTEM FOR SECURE DATA

(57) Abstract

A data security system for digitized data that can use both encryption and steganographic techniques. Data is encrypted using selected encryption protocols and algorithms. The resulting ciphertext is then treated to a steganographic encoding into a secondary data stream. This encoding is performed by replacing the least significant bit of selected bytes in the secondary data stream with the bits of the ciphertext. Selection of those bytes is performed via ciphertext created using a separate encryption procedure that uses an encryption key as both the key and as the data to be encrypted. The format of the secondary data stream is usually chosen so that it will not be obvious that it has been modified. The modified data can either be stored for later use or transmitted to the intended recipient. Decoding is accomplished by using the cyphertext to find the selected bytes in the modified secondary data stream, extracting the least significant bits, and reassembling those bits into the original data. These encryption techniques are also used for backing up data in a secure manner. The data to be backed up is first encrypted using selected encryption protocols and algorithms. The resulting ciphertext is then split into multiple parts and each part is stored on a separate floppy disk in selected locations that are determined by a separate encryption process which produces a selection cyphertext to select the storage locations. The remaining areas of each floppy disk are then filled in with random data to obscure the identity of the backed up information. Restoration of the original data is accomplished by using the selection cyphertext to locate the stored data on each floppy, reading that data and reassembling it into a data block, and then merging the data blocks from each floppy to recreate the original data.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

ENCRYPTION SYSTEM FOR SECURE DATA

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates to digital data security technologies and communications and more particularly to a method and apparatus for securing data and permitting secure electronic communications relying on encryption and steganographic techniques.

2. Description of the Related Technology

Information and data transfer is growing at an alarming rate. The need for data security has also grown at an equal if not greater speed. The core problems that need to be addressed by any data security system are level of security, ease of use, integratability within the work environment, and mass/ global level usage. By contrast, electronic communications are inherently insecure and open. Confidential and proprietary information and data are transferred regularly through channels which are in need of a secure and user-friendly methodology to facilitate confidentiality. Presently there are two basic approaches to securing information: access control (such as password protection), and encryption (single key, dual key, One Time Pad (OTP), steganography, and hardware). Each of these methods has its own advantages and disadvantages pertaining to implementation, design, level of security, interoperability, development, ease of use, and widespread adoption, use, and appeal.

Password protection is commonly used for access control but has inherent security level shortcomings when applied to data security.

Encryption has been implemented in five basic methodologies:

(a) Single key encryption -- This uses an encryption algorithm along with an encryption key to encrypt and decrypt data. The same key is used in both encoding and decoding. The major shortcoming of single key technology is the transference of the key to the recipient. Most methodologies require a separate secure communication of the key to the recipient/decoder, via either fax, telephone or in person.

(b) Dual key encryption -- Public key encryption solves this problem by utilizing different keys for encryption and decryption. The encryption-public key is given out insecurely to all potential encoders. The decryption-private key is kept by the recipient/decoder and not given out. Encryption is performed on data using the public key and only the private key can decode the data encrypted using its matching public half. Public key technology is primarily based on factorials of large prime numbers that facilitate the public and private key halves. There are presently several patents relating to public key technology, such as those held by RSA (U.S. Patent 4,405,829) and Diffie-Hellman (U.S. Patent 4,200,770). Presently, there are three shortcomings to the public key system in a large network. First, a repository capable of storing millions of public keys needs to be in place to facilitate Internet level global communications security. Second, due to the nature of public keys, they require large streams of data, and a method of authentication needs to be in place to validate the authenticity of the public keys within the repository. Third is the present difficulty and lack of ease of use inherent in today's public key products.

(c) One Time Pad (OTP) is an encryption methodology that provides a high level of security for encrypted information. However, due to its reliance on truly random sequences for the initialization key and the inability of software to provide truly random numbers, OTP is not suitable for the mass market of end users and corporate users for secure communications.

(d) Hardware encryption cards and boxes have been used for secure communications. Hardware encryption provides high level security and key management but is very costly. In addition, hardware encryption systems have not been compatible with other hardware systems, i.e. they lack interoperability. Hardware encryption is ideal for point to point communications or closed systems where cost is not a factor; they range from $1,000 - $25,000 in cost. Hardware encryption systems are typically not suitable for open or mass communication applications.

(e) Steganography is a relatively new method for data security. Steganography, the art and science of hiding the existence of information, has in the past been primarily associated with invisible inks, messages sent via telephone line noise known as TranSec, and red cellophane such as that used in games to reveal information hidden in a red-blue block. Within the past two years, Steganography has migrated to the computer in the form of hiding information in graphical images, sound files, or other media including text files such as Mimic functions. Mimic functions convert plaintext letters into common everyday English words that are then put together to form pseudo-sentences. The resulting message resembles a Madlib™. When steganography is applied to graphic images, it encodes information in the code of a graphic file either in the pixel coding, high bits, or low bits of the coding. The former is useful for small messages but becomes apparent if large amounts of data are hidden. The latter two are more recent but are still in development stages as the encoding generates a noticeable deviation from the standard code of the graphics file. An example of a steganographic system is described by Cooperman et al. (U.S. patent 5,613,004), which is herein incorporated by reference in its entirety. All of these methods lose their security when the system is known and therefore should be used together with key based encryption for additional security.

SUMMARY OF THE INVENTION

This system features a data security method that combines the strength of encryption with the strength of steganography. According to the invention, a software or hardware package may provide both a secure method of communications via electronic communications systems and a secure work environment for data. The features and processes utilized in the invention deliver a cohesive standard by which data can be secured. Further the invention features an easy to use interface that is cohesive with standard protocols and procedures within both the corporate and individual environments. Finally, the interface combined with the high level of security and functionality lends mass market and global level appeal to the invention.

It is an object of the invention to provide an encryption and security program for a personal computer system such as those using the Windows environment. It is a further object to provide an easy to use interface, and an easy to integrate environment. It is a further object to provide for file and document encryption, including selective partial document encryption; combined encryption and steganography for super secure domestic and secure international communications; multiple overwrite for deleted files; Digital Signatures for providing sender verification and document integrity validation; active document blanking for roaming eyes protection in impersonal or hostile work environments; steganographic key management; key list transfers; file compression; a secure key list back-up facility; administrative features; and unique registration and distribution features. Of course, the system also includes provisions for recovering the encrypted and encoded information.

Once the data has been encrypted and/or steganographically encoded, it may be transmitted to another system, kept in the present system, stored on a media, or any combination of those choices. Because of the exposed nature of most communications methods, the greatest need for such security is usually for transmitted data.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows a flow diagram of the encryption process.

Figure 2 shows a flow diagram of public key distribution.

Figure 3 shows a flow diagram of the process of combining encryption with steganography.

Figure 4 shows a flow diagram of the steganographic selection algorithm process for generation of the selection algorithm.

Figure 5 shows a flow diagram of the process of steganographically encoding data into a stream of data bits.

Figure 6 shows an illustration of the floating toolbar user interface.

Figure 7 shows a flow diagram of combining encryption and steganography for the purpose of secure backup to removable media, as may be used for storage of personal or administrator keys.

DETAILED DESCRIPTION OF THE INVENTION

A system according to the invention may encrypt files and documents with a choice of multiple algorithms and methods. One preferred embodiment may be designed with a modular open-ended architecture in an object-oriented scheme that allows the easy insertion of any encryption algorithm, including both single key and public key algorithms. The invention may utilize the modular design with flags that alert the program to the encryption method used for simple decryption and easy upgrade to multiple encryption algorithms and methods. Each algorithm is treated as a modular process that fits into the predefined call of the program. The program calls provide for a key(s) and input/output of data, thereby allowing for any encryption algorithm to be plugged in once configured to meet the specifications of the program calls.

The preferred embodiment uses a two-step process: an encryption algorithm(s) creates the ciphertext, which is then steganographically secured. Both conventional single key encryption and hybrid public/single key encryption systems can be used here. Since multiple types of encoding are contained in the system, each type using potentially different types of keys, key management is also a major feature. The flow charts of Figures 1 - 5 and 7 show the various processes involved in the system.

Figure 1 shows the basic encryption process. In step 2 the data is input into the system through any suitable method (file selection or from within an application via a floating toolbar button as illustrated in Figure 6), where key management is performed in step 3. Key management can be the insertion of a key by the user, a selected key from a key list, or other key selection methodology including randomly generated keys. The key(s) is then used by the encryption process of step 4 to encrypt the retrieved data. The encryption method used is determined by a configuration performed by the user or by a preset default. An encrypted message, or ciphertext, is produced at step 5 as a result of this process. If only basic encryption is desired, the process can end here with the ciphertext being returned to either a file or back into the application the original data came from. However, for greater security and/or innocuous messaging the ciphertext can be subjected to further encoding techniques, as described later.

Figure 2 shows the basic key transfer process utilized by the invention when distributing keys for use in a public key encryption system. In step 7, public keys are selected from a list or database for transfer. Keys are digitally signed by the sender's signing key in step 8 and transferred in step 9 via electronic mail, a network system, or saved to a file for manual transfer. Upon receiving the transferred keys, the recipient system authenticates them using existing public keys, and the recipient's local database of keys is updated in step 10 with the transferred keys. This process enables users to utilize public key encryption as described in Figure 1 for encryption of data and also for key authentication using a horizontal trust model.

Figure 3 shows a two-step encoding process, with encryption followed by steganographic encoding. Steps 12 through 14 correspond to steps 2 through 4 in Figure 1, which produce a ciphertext through standard encryption techniques as described earlier. In step 15, a steganographic selection algorithm (SSA) uses the ciphertext in a bit-to-byte correspondence for selection of replaceable characters/bytes. The replacement produces the steganographically encoded result at step 17. The details of this process are expanded in Figures 4 and 5.

Figure 4 shows the details of the steganographic selection algorithm (step 15 of Figure 3), which uses an encryption key as both a key and as data to be encrypted. A key is chosen in step 19, either randomly generated or chosen by the user, and repeatedly copied in step 20 until the result is the same size as the data stream or media which will be steganographically modified. This result is then encrypted in step 21 with the same key to produce a selection ciphertext. In step 22, each bit of the selection ciphertext is then associated with a byte (or pixel, or other suitable unit) in the data stream. The selected data stream bytes are those whose associated bit in the ciphertext is a logic '1' (or alternately, a logic '0'). In a preferred embodiment, every data stream byte has an associated bit in the selection ciphertext, but for greater security only every nth byte can have an associated bit. While this selection algorithm is described in the context of steganographic data encoding, it can also be used for any other selection applications in which the selected units are to remain secret.

Figure 5 shows the details of the steganographic encoding process (step 16 of Figure 3). The data to be encoded is brought in at step 25, and in step 26 each bit of this data is associated with a selected byte in the data stream or media to be modified. The data stream or media can be a random number stream, multimedia file such as a graphic image or sound file, a text document, or some other form of data. In step 27 the media is brought in for association with the bits in step 26. Using the selection algorithm developed in Figure 4, the least significant bit of each selected byte is replaced in step 28 by the corresponding data bit from step 26. This newly modified data stream or media represents the encoded steg-stream of step 29 which produces the secured message shown in step 17 of Figure 3.

This process can also be described in mathematical terms. The steganographic selection algorithm is a reproducible random selection process utilizing ciphertext as the selection operand, where:

k = any key
d = any target data
n = length in bytes of target data
$\Theta = S_{(X)}$
$S_{(X)}$ = A selection of x

$$f_0(d, f_x(k)) = d_x$$

$f_x(k) =$

$$f_r(n_d,k) = k^n{}_d \quad : \quad n_k{}^n = n_d$$

$$\&$$

$$f_c(k^n{}_d,k) = (k^n{}_d)_k \quad : \quad (k^n{}_d)_k \neq k^n{}_d$$

$\therefore\ f_0(\,(k^n{}_d)_k,d) = d_x$

The invention utilizes an encryption algorithm and resulting ciphertext to enable a selection of data. The selection , $f_0()$, is performed on the data, d, based on the bits from the resulting ciphertext, $f_x(k)$, where k is the encryption key and $f_x()$ is the process by which the key is used as both the encryption key and the plaintext, created by $f_r()$. $f_r()$ takes the encryption key, k, and copies it back to back such that the resulting plaintext is equivalent in size, n, to the target data, n_d, to be selected from. $f_c()$ uses the encryption key, k, to encrypt the plaintext resulting from $f_r()$, or $k^n{}_d$. The resulting ciphertext, $(k^n{}_d)_k$, is then used in a bit-to-data comparison to facilitate selection of units where d is the target data to be selected from and x is the selecting bit from the ciphertext. Either a 1 or 0 can be used as the selecting bit. d_x is the resulting selected units from the data.

The steganographic encoding process can be described mathematically by the following:

m = any medium
d = original data
n = length in bits of original data
$f_s()$ = a selection process
$f_l()$ = a least significant bit conversion

$f_{s2s}(m, m'_d) = m''$

1

$f_s(m) = m'$

$f_l(m',d_{0\text{-}n(d)}) = m'_d$

This process of encoding data within other data can be facilitated by the use of:

(1) any medium, or data stream into which the original data will be encoded, such as that produced by a pseudo random number generator (PRNG);

(2) the selection process;

(3) a key to feed the selection process; and

(4) the original data, to be encoded into the medium.

The invention performs a least significant bit conversion on the medium where the bits from the original data are then inserted into selected bytes of the medium at the least significant bit position. The selection of bytes from the medium is performed by some selection process, such as an algorithm, character key, or other method.

As an example, a random stream generated by a PRNG can be used as the medium. Data is encoded or hidden in the stream by converting the least significant bits from selected bytes of the medium into the bits from the original data, whether in plaintext or ciphertext form.

An example of an encoding is as follows:

Object: Hide the word "it."
The bit sequence for the word "it" is: 01101001 01110100.

The original PRNG stream:

```
aku245q9S53'%14hj2q356@$%d070c6*&56)(noiury24r0q34ikjh3982h'
H*37Hqdhg(dg&h#daghhhaead43y3q8uQ#8y3d203emape3m'zd+w[2oh[2
```

Selection of the characters to be converted (underlined):

```
aku245q9S53'%14hj2q356@$%d070c6*&56)(noiury24r0q34ikjh3982h'
H*37Hqdhg(dg&h#daghhhaead43y3q8uQ#8y3d203emape3m'zd+w[2oh[2
```

The bit sequences of the characters selected:

a	q	j	d	n	i
01100001	01110001	01101010	01100100	01101110	01101001
i	h				
01101001	01101000				

H	*	q	#	8	p
01001000	00101010	01110001	00100011	00111000	01110000
m	[				
01101101	01011011				

The bit sequence for the word "it," 01101001 01110100, is hidden in the selected byte sequences by least significant bit conversion:

01100000 01110001 01101011 01100100 01101111 01101000
01101000 01101001

01001000 00101011 01110001 00100011 00110010 01110001
01101100 01011010

The characters represented by the new byte sequences are:

'qkdohhiH+q#8ql\

These characters are now placed back into the stream from whence they came, giving a new data stream (the replacement characters are underlined):

'ku245q9S53'%14hk2q356@$%d070c6*&56)(oohury24r0q34hkjh3982i'
H+37Hqdhg(dg&h#daghhhaead43y3q8uQ#8y3d203emaqe3l'zd+w[2oh\2

The file to be 'stegged', i.e., the data stream in which the message will be hidden, is selectable by the user. This file can be in many formats, but for effective security it should be the type of file that can be modified without such modification being obvious. In addition to the PRNG file previously mentioned, other preferred formats are: graphics (BMP, GIF, JPG, etc.), sound (WAV, RA, AIF, MIDI, etc.), and video (MPG, MOV, etc.). Depending on the degree of modification, these files might be usable for graphics, sound, or video in their modified form without noticeable performance degradation, thus further obscuring the fact that they contain encoded data.

The system also has a number of other features which give the user a range of selectable security functions:

Toolbar User Interface (TUI) - The system utilizes a floating toolbar that gives the user simple accessibility to security functions throughout the operating system environment. Figure 6 shows an example of selections available from the toolbar. Item 31 is the toolbar's handle which enables both the drag and docking of the toolbar in the environment and, by right mouse clicking on the handle, allows for the configuration of the toolbar and other setup options. Items 32a and 32b are the encrypt and decrypt buttons which allow users to select both files and data from outside the application to be encrypted by the invention. Item 33 is the active window protection button. Item 34 is the place holder for other features configurable for use on the toolbar with the default being the exit function.

Selective Text Encryption - The system allows the user to highlight and select portions of the text, including paragraphs, sentences, even words, to be encrypted within a plaintext document. Delimiters are used so that a user decrypting the selected text does not have to identify its exact boundaries.

Secure Delete - In most computers, deleted files are still retrievable after deletion because only the file's index pointer information has been removed, whereas the file itself is still intact. The system provides a secure deletion facility that will overwrite a file first with a set such as all 0's and then with the set's compliment such as all 1's, and then with one or more overwrites of random data before deleting the file. This protects the user from having files retrieved from the drive after deletion. This function is selectable by the user.

Digital Signatures - The system provides Digital Signature capability to its users. Digital Signatures verify the origin and document integrity using one way hash functions and the Signing key belonging to the user. The system generates a hash sequence based on the contents of a document and then encrypts it with the Signing key. This sequence can be checked by the recipient to validate the sender and the contents of the document by decrypting the hash sequence using the verification key, packaged with the user's public key, and then comparing the hash of the document to the one contained in the Digital Signature. Digital Signatures can be used on both plaintext as well as ciphertext messages.

Partial Screen Blanking - This provides a secure work environment by covering most of the active window with a graphic image. The active window is protected up to approximately the last 3/4 inch of the window. The graphic image acts like a window shade and can be manipulated using the mouse to cover more or less of the active window. This allows the user to continue entering data and have the rest of the document secured from view. This feature is especially useful when the screen is in an unsecured area, where "roving eyes" can be a problem. Partial screen blanking toggles on and off via a button on the TUI and can also be set to require a key to be toggled off.

Key Management - The invention utilizes a special key management system incorporating public and single key technologies with Digital signatures and certificate formats, combined with ease of use to allow easy integration of a public key communication system without the need for a fully implemented public key infrastructure. The invention utilizes hybrid encryption. Data is encrypted using a Pseudo Random Number Generator (PRNG) generated session key to create the message ciphertext. The session key is then encrypted using either a recipient's public key (for transmission security), or a predetermined user local key (for storage security). A header is added to the message ciphertext containing the encrypted session key and other information necessary for decryption purposes. The other information contained in the header includes algorithm identifiers, file name, time and date information, sender name, and a digital signature if that option is selected. The header is divided into protected areas and clear text areas where protected areas are also encrypted and clear text areas are the minimum information needed to decrypt the header and message ciphertext upon receipt without undermining the security of the system.

Public keys are transferred between corespondents via a key transfer utility included in the invention that facilitates a transparent Internet and/or network-based transfer of public key information. The public keys are X.509 certificate formatted in compliance with the standards for compatibility with Certificate Authorities (CA's) and can also be digitally signed by the sender to provide a horizontal Web of Trust infrastructure. Transmitted public keys contain the public half of the public key pair, the verification key, sender username and e-mail address, and any authentication information available to the sender including previous senders' signatures and digital identifications provided by CA's. Public keys are maintained in a database where authentication information is also associated with each public key.

The user's secret information includes the user's private half of the public key pair, the user's signing key and the user's local encryption key. These are secured locally by the user.

The system allows the user to utilize encryption and steganographic capabilities to both secure data and allow for innocuous messaging. As described above, a message or file is encrypted and steganography is used to encode the encrypted file into a graphic image, sound or movie file.

File compression - A file compression utility is provided for the encrypted file to reduce file sizes.

Secure Backup on Removable Media - This provides a method by which users can back up and store off-line on removable media important data such as personal decryption keys, Administrator keys, key pairs, or other highly confidential information and data. As shown in Figure 7, data is input into the system, encrypted, and then encoded onto the removable media. Figure 7 illustrates this process using two floppy diskettes. In a preferred embodiment, this process can be performed to multiple floppies and onto almost any removable media such as recordable CD-ROMs, Zip Disks, Back Up Tapes, etc. It can also be used with multiple locations of remote or local non-removable storage, although this might provide a lesser degree of security. In step 36 the data, which could consist of personal keys or Administrator public key pairs, is input into the system. In step 37 key management is performed to determine the encryption key(s) for the data, such as a public key for a system administrator and a randomly generated session key. The data is then encrypted in step 38 using an encryption algorithm as in Figure 1 utilizing the keys from step 37. At step 39 the encrypted data is split into files corresponding to the number of removable media to be used. This splitting function involves taking bytes from the encrypted data and placing them into separate files such that each byte is placed in a different file than its adjacent bytes. For example, if three files were to be created using the splitting process in step 39, and 'abcdefghijklmno' was the starting file to be stored, the resulting split files would be: 'adgjm', 'behkn', and 'cfilo'. This process eliminates unwanted exposure of partial ciphertexts in the event that the security of one or more of the split files is compromised. In step 40, each split file is written to selected sectors of its removable media. The key produced in step 37 is used to choose the exact sector(s) on the removable media where the files are written. The keys themselves can also be written to one or more of the removable media, such as the first one. Both file(s) and keys can be written without headers. In step 41 the removable media is written to by a pseudo-random number generator (step 43) to "fill" the remaining sectors of the media with random noise. This filling process writes the random data to all sectors of the removable media except those previously written to. The removable media now contains the split files of the encrypted data masked into random noise. This process represents another form of steganography, since it entails the 'hiding' of sensitive data among other, non-sensitive data.

Retrieval of the hidden data can occur by effectively reversing the aforementioned process: Key management is first performed to retrieve the relevant key(s) from the media. Then each split file is retrieved using the proper key(s), the multiple split files are merged, and decryption is performed on the merged file. This is basically a reversal of steps 38-40.

Additional security can be provided through additional means, such as storing each of the backup media in a different location, physically securing the backup media, or keeping the key(s) separate from all the backup media. Such variations are obvious and the invention does not depend on their use.

Administrative Features - The system provides the network administrator or security administrator with several features tuned to the corporate environment. The Administrator can block access of employees to certain features of the system, including secure deletion, steganography, etc. The Administrator is also given the capability to gain emergency access to files and E-mails generated by users on the network.

Decryption - Obviously, every encryption method also includes a decryption capability. The decryption is generally the equivalent of an encryption operation in reverse. The system of this invention can decrypt any message encrypted by the aforementioned methods, using the decryption method appropriate for each type of encryption. Multi-encrypted data is decrypted in the reverse order in which it was encrypted, using any decryption information available from the message itself.

Various means of accomplishing the secure data features described herein are possible. Since most of the procedures involve the manipulation of digital data, the preferred embodiment uses software as a means to perform the described functions. The use of firmware, or a combination of software and firmware, is also envisioned. The preferred embodiment operates in the Windows environment due to the extremely broad market acceptance of the Windows operating system. But the invention will operate equally well in other environments, including single- or multi-user operating systems and single- or multi-processor systems.

The invention may be embodied in other specific forms without departing from the spirit of the invention or its essential characteristics. The present embodiments are to be considered as illustrative and not restrictive, the scope of the invention being indicated by the following claims rather than by the foregoing description. All changes which come within the meaning and the range of equivalency of the claims are intended to be embraced herein.

What is claimed is:

1. A method of providing data security for digital data by using steganographic techniques, comprising the steps of:

providing a first data block comprising a plurality of data bits;

providing a second data block comprising a plurality of discrete units, each of said discrete units comprising a plurality of bits;

selecting a plurality of particular discrete units within the second data block, the number of said plurality of particular discrete units being at least as great as the number of data bits contained in the first data block;

associating each data bit in the first data block with a corresponding selected particular discrete unit in the second data block;

replacing a selected bit within each selected particular discrete unit with said selected particular discrete unit's associated data bit to form a corresponding new particular discrete unit; and

replacing each selected particular discrete unit in the second data block with its corresponding new particular discrete unit to form a modified second data block.

2. The method of claim 1, wherein each discrete unit is a byte.

3. The method of claim 1, wherein said selected bit of each particular discrete unit is the least significant bit.

4. The method of claim 1, wherein the second data block is generated by a pseudo-random number generator.

5. The method of claim 1, wherein the second data block is a file selected from a list comprising a graphics file, a sound file, or a video file.

6. The method of claim 1, wherein the step of associating each data bit in the first data block with a corresponding selected particular discrete unit in the second data block is done in a sequential manner, comprising:

a) associating the first data bit in the first data block with the selected particular discrete unit that occurs first in the second data block;

b) associating the next data bit in the first data block with the selected particular discrete unit that occurs next in the second data block; and

c) repeating step b) for each of the remaining data bits and selected particular discrete units.

7. The method of claim 1, wherein the step of selecting a plurality of particular discrete units comprises:

using an encryption algorithm to create a selection key comprising a series of binary bits; and

associating each binary bit in the selection key with a corresponding discrete unit in the second data block.

8. The method of claim 7, wherein each discrete unit in the second data block that is associated with a logic '1' bit in the selection key is a selected particular discrete unit.

9. The method of claim 7, wherein each discrete unit in the second data block that is associated with a logic '0' bit in the selection key is a selected particular discrete unit.

10. The method of claim 1, further comprising the steps of constructing a copy of the first data block from the modified second data block, said steps comprising:

selecting each said new particular discrete unit in the modified second data block;

extracting the selected bit from each selected new particular discrete unit; and

combining the extracted data bits to form a copy of the first data block.

11. A system for providing data security for digital data by using steganographic techniques, comprising:

a first data block comprising a plurality of data bits;

a second data block comprising a plurality of discrete units, each of said discrete units comprising a plurality of bits;

means for selecting a plurality of particular discrete units within the second data block, the number of said plurality of particular discrete units being at least as great as the number of data bits contained in the first data block;

means for associating each data bit in the first data block with a corresponding selected particular discrete unit in the second data block;

means for replacing a selected bit within each selected particular discrete unit with its associated data bit to form a corresponding new particular discrete unit; and

means for replacing each selected particular discrete unit in the second data block with its corresponding new particular discrete unit to form a modified second data block.

12. The system of claim 11, wherein each discrete unit is a byte.

13. The system of claim 11, wherein said selected bit of each particular discrete unit is the least significant bit.

14. The system of claim 11, further comprising a pseudo-random number generator, the second data block being generated by said pseudo-random number generator.

15. The system of claim 11, wherein the second data block is a file selected from a list comprising a graphics file, a sound file, or a video file.

16. The system of claim 11, wherein the means for associating each data bit in the first data block comprises:

a) means for associating the first data bit in the first data block with the selected particular discrete unit that occurs first in the second data block;

b) means for associating the next data bit in the first data block with the selected particular discrete unit that occurs next in the second data block; and

c) means for repeating b) for each of the remaining data bits and discrete units.

17. The system of claim 11, wherein the means for selecting a plurality of particular discrete units comprises:

an encryption algorithm;

a selection key comprising a plurality of binary bits, said selection key having been created by the encryption algorithm; and

means for associating each of the plurality of binary bits in the selection key with a corresponding discrete unit in the second data block.

18. The system of claim 17, wherein each selected particular discrete unit in the second data block is associated with a logic '1' bit in the selection key.

19. The system of claim 17, wherein each selected particular discrete unit in the second data block is associated with a logic '0' bit in the selection key.

20. The system of claim 11, further comprising:

means for selecting each said new particular discrete unit in the modified second data block;

means for extracting the selected bit from each selected new particular discrete unit; and

means for combining the extracted data bits to form a copy of the first data block.

21. A method of providing data security for digital data by using both encryption and steganographic techniques, comprising the steps of:

encrypting a selection of data to produce a first data block, said first data block comprising a plurality of data bits;

determining the number of said data bits contained in the first data block;

providing a second data block comprising a plurality of discrete units, each of said discrete units comprising a plurality of bits;

selecting a plurality of particular discrete units within the second data block, the number of said plurality of particular discrete units being at least as great as the number of data bits contained in the first data block;

associating each data bit in the first data block with a corresponding selected particular discrete unit in the second data block;

replacing a selected bit within each selected particular discrete unit with said selected particular discrete unit's associated data bit to form a corresponding new particular discrete unit; and

replacing each selected particular discrete unit in the second data block with its corresponding new particular discrete unit to form a modified second data block.

22. The method of claim 21, wherein each discrete unit is a byte.

23. The method of claim 21, wherein said selected bit of each particular discrete unit is the least significant bit.

24. The method of claim 21, wherein the second data block is generated by a pseudo-random number generator.

25. The method of claim 21, wherein the second data block is a file selected from a list comprising a graphics file, a sound file, or a video file.

26. The method of claim 21, wherein the step of associating each data bit in the first data block with a corresponding selected particular discrete unit in the second data block is done in a sequential manner, comprising:

a) associating the first data bit in the first data block with the selected particular discrete unit that occurs first in the second data block;

b) associating the next data bit in the first data block with the selected particular discrete unit that occurs next in the second data block; and

c) repeating step b) for each of the remaining data bits and selected particular discrete units.

27. The method of claim 21, wherein the step of selecting a plurality of particular discrete units comprises:

using an encryption algorithm to create a selection key comprising a series of binary bits; and

associating each binary bit in the selection key with a corresponding discrete unit in the second data block.

28. The method of claim 27, wherein each discrete unit in the second data block that is associated with a logic '1' bit in the selection key is a selected particular discrete unit.

29. The method of claim 27, wherein each discrete unit in the second data block that is associated with a logic '0' bit in the selection key is a selected particular discrete unit.

30. The method of claim 21, further comprising the steps of constructing a copy of the first data block from the modified second data block, said steps comprising:

selecting each said new particular discrete unit in the modified second data block;

extracting the selected bit from each selected new particular discrete unit;

combining the extracted data bits to form a copy of the first data block; decrypting the copy of the first data block to form a copy of the selection of data.

31. A system for providing data security for digital data by using both encryption and steganographic techniques, comprising:

a selection of data;

a first data block, wherein said first data block is created by encrypting the selection of data, said first data block comprising a plurality of data bits;

means for determining the number of data bits contained in the first data block;

a second data block comprising a plurality of discrete units, each of said plurality of discrete units comprising a plurality of bits;

means for selecting a plurality of particular discrete units within the second data block, the number of said plurality of particular discrete units being at least as great as the number of data bits contained in the first data block;

means for associating each data bit in the first data block with a corresponding selected particular discrete unit in the second data block;

means for replacing a selected bit within each selected particular discrete unit with its associated data bit to form a corresponding new particular discrete unit; and

means for replacing each selected particular discrete unit in the second data block with its corresponding new particular discrete unit to form a modified second data block.

32. The system of claim 31, wherein each discrete unit is a byte.

33. The system of claim 31, wherein said selected bit of each particular discrete unit is the least significant bit.

34. The system of claim 31, further comprising a pseudo-random number generator, the second data block being generated by said pseudo-random number generator.

35. The system of claim 31, wherein the second data block is a file selected from a list comprising a graphics file, a sound file, or a video file.

36. The system of claim 31, wherein the means for associating each data bit in the first data block comprises:

a) means for associating the first data bit in the first data block with the selected particular discrete unit that occurs first in the second data block;

b) means for associating the next data bit in the first data block with the selected particular discrete unit that occurs next in the second data block; and

c) means for repeating b) for each of the remaining data bits and selected particular discrete units.

37. The system of claim 31, wherein the means for selecting a plurality of particular discrete units comprises:

an encryption algorithm;

a selection key comprising a plurality of binary bits, said selection key having been created by the encryption algorithm; and means for associating each of the plurality of binary bits in the selection key with a corresponding discrete unit in the second data block.

38. The system of claim 37, wherein each selected particular discrete unit in the second data block is associated with a logic '1' bit in the selection key.

39. The system of claim 37, wherein each selected particular discrete unit in the second data block is associated with a logic '0' bit in the selection key.

40. The system of claim 31, further comprising:

means for selecting each said new particular discrete unit in the modified second data block;

means for extracting the selected bit from each selected new particular discrete unit;

means for combining the extracted data bits to form a copy of the first data block; and

means for decrypting the copy of the first data block to form a copy of the selection of data.

41. A method of securely backing up data, comprising the steps of:

providing a first data block, said first data block comprising a plurality of discrete units;

providing a plurality of memory units;

splitting said first data block into a plurality of split data blocks, the number of said split data blocks being equal to the number of said memory units, wherein each of said split data blocks contains a portion of the discrete units from the first data block;

associating each of said split data blocks with a memory unit;

encoding each of said split data blocks in its associated memory unit.

42. The method of claim 41, wherein the step of providing a first data block comprises encrypting a selection of data to produce said first data block.

43. The method of claim 41, wherein the step of encoding comprises:

for each split data block and associated memory unit, selecting a plurality of locations in said associated memory unit;

associating each of said discrete units in said split data block with one of said plurality of locations;

writing each of said discrete units in its associated location.

44. The method of claim 43, wherein the step of encoding further comprises writing data to other locations that are not associated with said discrete units.

45. The method of claim 44, wherein said plurality of locations are selected by a selection key.

46. The method of claim 44, wherein each of said discrete units is a byte.

47. The method of claim 44, wherein each of said memory units is a removable storage unit.

48. The method of claim 41, further comprising the steps of restoring backed up data from said plurality of memory units, said steps comprising:

retrieving said plurality of split data blocks from said plurality of associated memory units;

merging said plurality of split data blocks to form a copy of said first data block.

49. A system for securely backing up data, said system comprising:

a first data block comprising a plurality of discrete units;

a plurality of memory units;

means for splitting said first data block into a plurality of split data blocks, the number of said split data blocks being equal to the number of said memory units, wherein each of said split data blocks contains a portion of the discrete units from the first data block;

means for associating each of said split data blocks with a memory unit;

means for encoding each of said split data blocks in its associated memory unit.

50. The system of claim 49, further comprising a means for encrypting a selection of data to produce said first data block.

51. The system of claim 49, wherein the means for encoding comprises:

means for selecting a plurality of locations in each of said associated memory units;

means for associating each of said discrete units in said split data block with one of said plurality of locations;

means for writing each of said discrete units in its associated location.

52. The system of claim 51, wherein the means for encoding further comprises means for writing data to other locations that are not associated with said discrete units.

53. The system of claim 52, further comprising a selection key adapted for selecting said plurality of locations.

54. The system of claim 52, wherein each of said discrete units is a byte.

55. The system of claim 52, wherein each of said memory units is a removable storage unit.

56. The system of claim 49, further comprising a means for restoring backed up data from said plurality of memory units, comprising:

means for retrieving said plurality of split data blocks from said plurality of associated memory units;

means for merging said plurality of split data blocks to form a copy of said first data block.

57. A method for selecting a plurality of discrete units with a secure selection key, said method comprising the steps of:

providing an encryption key;

copying said encryption key a plurality of times to form a data block;

encrypting said data block with said encryption key to form a selection key, said selection key comprising a plurality of bits;

providing a plurality of discrete units;

associating each bit of said selection key with one of said plurality of discrete units;

selecting only those discrete units whose associated bit has a particular logic state.

58. The method of claim 57, wherein the step of associating further comprises:

associating each bit of said selection key with every nth discrete unit, where n is an integer.

59. The method of claim 57, wherein said particular logic state is a logic 'one'.

60. The method of claim 57, wherein said particular logic state is a logic 'zero'.

61. The method of claim 57, wherein said discrete unit is a data byte.

62. The method of claim 57, wherein the number of bits in said selection key equals the number of discrete units.

63. A system for selecting a plurality of discrete units with a secure selection key, said system comprising:

an encryption key;

means for copying said encryption key a plurality of times to form a data block;

means for encrypting said data block with said encryption key to form a selection key, said selection key comprising a plurality of bits;

a plurality of discrete units;

means for associating each bit of said selection key with one of said plurality of discrete units;

means for selecting only those discrete units whose associated bit has a particular logic state.

64. The system of claim 63, wherein the means for associating further comprises:

means for associating each bit of said selection key with every nth discrete unit, where n is an integer.

65. The system of claim 63, wherein said particular logic state is a logic 'one'.

66. The system of claim 63, wherein said particular logic state is a logic 'zero'.

67. The system of claim 63, wherein said discrete unit is a data byte.

68. The system of claim 63, wherein the number of bits in said selection key equals the number of discrete units.

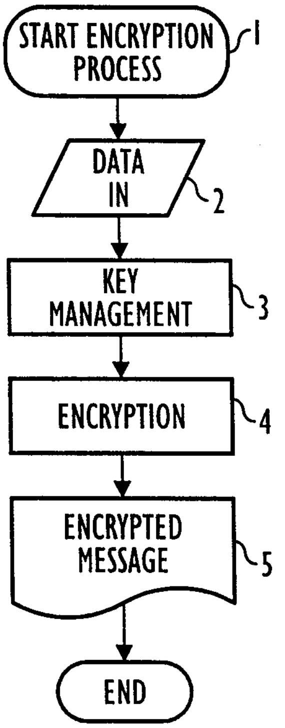
START ENCRYPTION PROCESS
1
DATA IN
2
KEY MANAGEMENT
3
ENCRYPTION
4
ENCRYPTED MESSAGE
5
END

FIG. 1

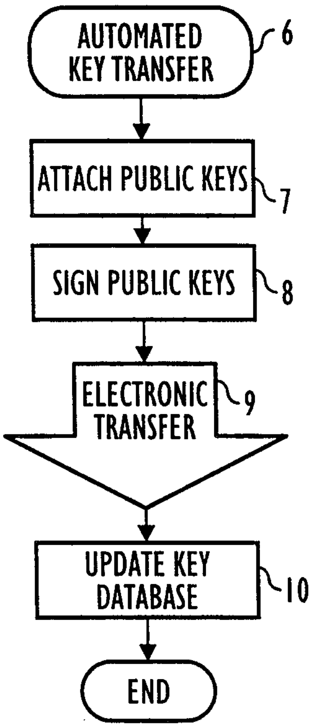
AUTOMATED KEY TRANSFER
6
ATTACH PUBLIC KEYS
7
SIGN PUBLIC KEYS
8
ELECTRONIC TRANSFER
9
UPDATE KEY DATABASE
10
END

FIG. 2

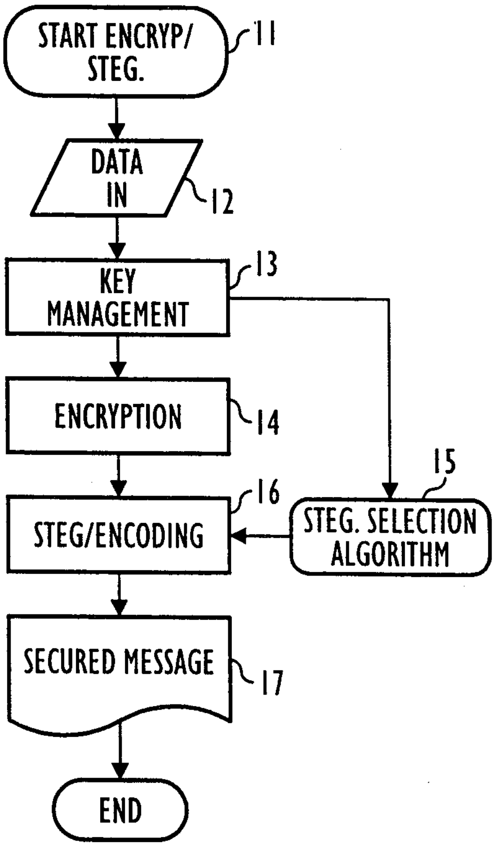
START ENCRYP/ STEG.
11
DATA IN
12
KEY MANAGEMENT
13
ENCRYPTION
14
15
STEG. SELECTION ALGORITHM
16
STEG/ENCODING
SECURED MESSAGE
17
END

FIG. 3

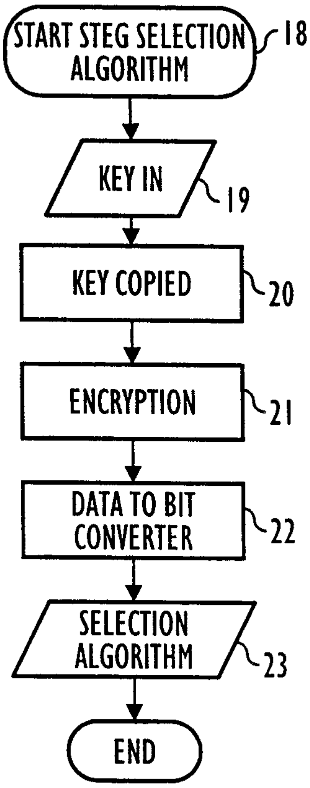
START STEG SELECTION ALGORITHM
18
KEY IN
19
KEY COPIED
20
ENCRYPTION
21
DATA TO BIT CONVERTER
22
SELECTION ALGORITHM
23
END

FIG. 4

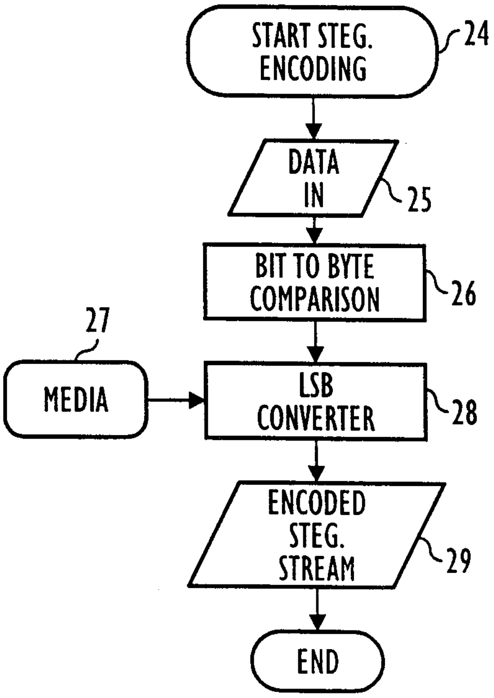
START STEG. ENCODING
24
DATA IN
25
BIT TO BYTE COMPARISON
26
27
MEDIA
LSB CONVERTER
28
ENCODED STEG. STREAM
29
END

FIG. 5

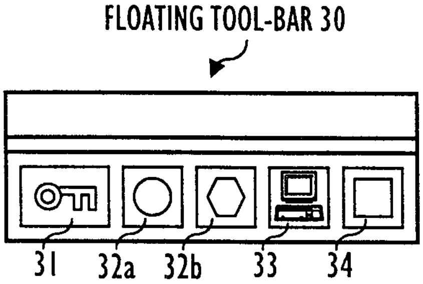
FLOATING TOOL-BAR 30
31
32a
32b
33
34

FIG. 6

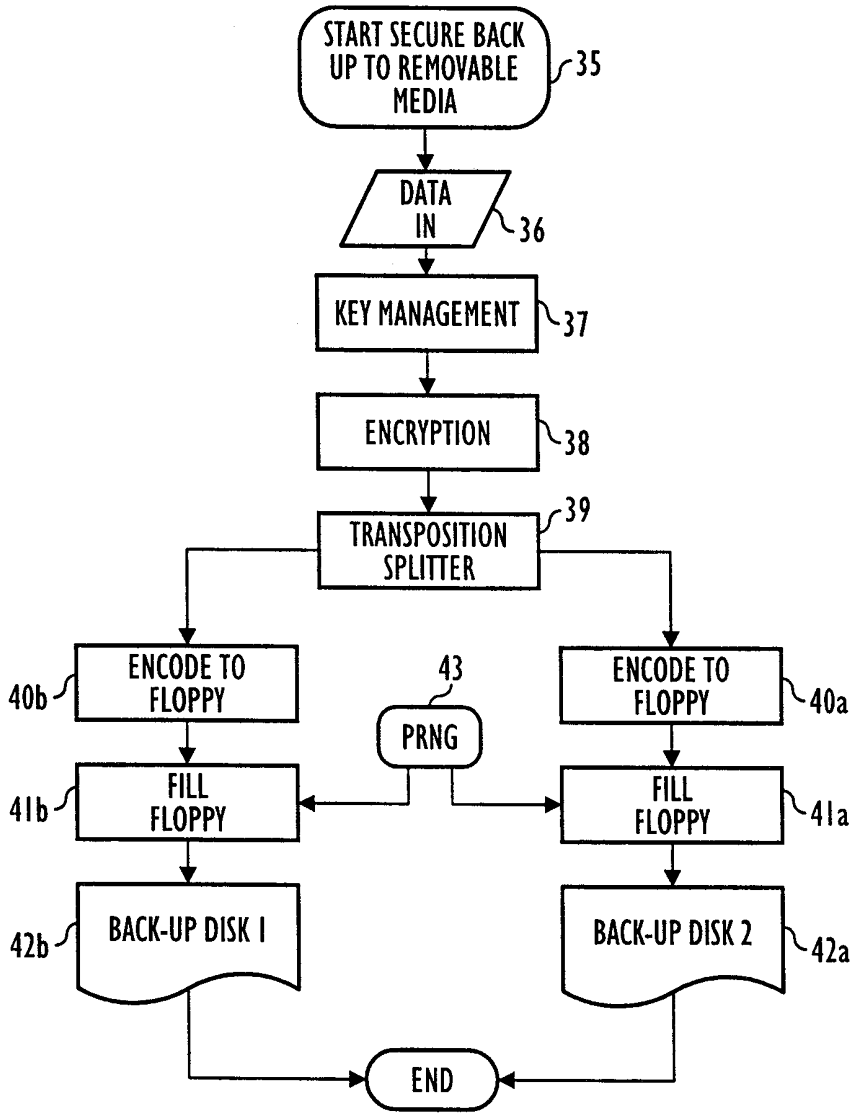
START SECURE BACK UP TO REMOVABLE MEDIA
35
DATA IN
36
KEY MANAGEMENT
37
ENCRYPTION
38
TRANSPOSITION SPLITTER
39
ENCODE TO FLOPPY
40b
43
PRNG
ENCODE TO FLOPPY
40a
FILL FLOPPY
41b
FILL FLOPPY
41a
BACK-UP DISK 1
42b
BACK-UP DISK 2
42a
END

FIG. 7