Computers & Security Vol.20, No.6, pp.533-538, 2001

0167-4048/01$20.00

Hiding Digital Information Using a Novel System Scheme

Wen-Hung Yeh[1,2], and Jing-Jang Hwang[2]

[1,2] Telecommunication Laboratories, Chunghwa Telecom Co., Ltd[1], Yang-Mei, Taoyuan, Taiwan 326, ROC
wenhung@ms.chttl.com.tw
[2] Institute of Information Management, National Chiao Tung University[2], Hsinchu, Taiwan 30050, ROC
jjhwang@cc.nctu.edu.tw

This work builds on Lin and Lee's document protection scheme, termed as Confused Document Encrypting Scheme (CDES), to present a novel strategy that could hide digital information and provide secret communication between two communication parties. Lin and Lee invented a document protection scheme to transmit large quantities of secret information. However, their design strategy is only suitable for single-byte character-based text document. The proposed scheme presents a novel strategy that modifies the restrictions in Lin and Lee's work. Any kind of digital data, not only the plain text, can be employed as a carrier or as a secret in the proposed system. The proposed method utilizes the hexadecimal representation of digital data in a manner that ensures it will have many practical applications. In addition, the procedure is easy to implement and capable of transmitting large quantities of secret information.

Keywords Cryptography, Confused Document Encrypting Scheme (CDES), Cheating File Database (CFD), Hexadecimal Code Index Table (HexCIT), Secret Content Index File (SCIF), Steganography

Introduction

Cryptography and steganography have recently become two significant branches of computer security. Cryptography scrambles a message (encryption) to make it meaningless to humans so eavesdroppers cannot recover the original message (decryption) without the key. However, the scrambled message always incurs suspicion. Steganography, derived from a Greek word that literally means "covered writing"[5], is distinct from cryptography since it tries to hide the message. The "invisible" message generated with steganography will not arise suspicious as an eavesdropper can view the message without realizing it contains an "invisible" message in it.

A German spy sent the following message during WWII [5]:

> *Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by-products, ejecting susets and vegetable oils.*

The message can be decoded by extracting the second letter in each word to produce the hidden message:

> *Pershing sails from NY June 1.*

Bender et al. [8], proved people can embed data to digital media for identification, annotation and copyright. Digital media includes text, image, audio, or anything that can be represented as bit streams. Three methods are considered when embedding data into plain text document: open space, syntactic and semantic methods. Open space methods embed bit stream by utilizing the space characters in a plain text document; syntactic methods manipulate punctuation to embed bit stream; while semantic methods substitute the words themselves for embedding private information. For example, open space methods can be employed to

interpret one space character between words as a bit value "0", and two space characters as "1", then a secret character "A", with binary representation 01000000, can be embedded in the following message [8]:

A□meat□□without□wine□is□like□a□day□□without sunshine

The least significant bit (LSB) insertion [3] is a common, simple approach for hiding a secret in an image. For example, the least significant bits of a color image can be utilized to store your own bit streams. Although this will change the source image data, it is "invisible" to eyes. Hiding information in images has become a popular research topic recently and has developed into a new branch referred as "digital watermark" [2][4][5] because of its copyright protection applications.

None of the systems examined above can process large amounts of secret information. However, Lin and Lee's Confused Document Encrypting Scheme [1] can transmit large quantities of secret information. Their method sends a meaningful cheating text that does not incur suspicion along with an encrypted special index file. The receiver can reconstruct the secret information as soon as he receives the cheating text and the index file. Nevertheless, Lin and Lee's method has one restriction: the cheating text cannot have fewer characters than the characters used by the secret text. For example, if the secret text is: "I love you", then the cheating text must include these eight kinds of ASCII code characters: "I", "l", "o", "v", "e", "y", "u", and "space." Moreover, although their method is easily implemented for single byte character-based text documents such as English, it is not suitable for double-byte character-based language, such as Chinese, Japanese, and Korean. To generate the meaningful cheating text for the double-byte character-based secret information is hard by employing their method. In addition, their implementation is limited to plain-text cheating text because it cannot use an image file, an audio file, or other kinds of digital files.

This work presents a novel system scheme that modifies the restrictions in Lin and Lee's method. The proposed scheme can utilize kinds of digital files, not only the plain text, to cheat eavesdroppers. The procedure is easy to implement and capable of transmitting large amounts of secret information. The rest of this paper is organized as follows. Section 2 describes the proposed system in detail. The security and performance issues are discussed in Section 3 and 4 respectively. Conclusions are finally made in Section 5.

System Architecture

A cheating file database (CFD) is employed to provide confidential communication. The CFD must be manually or automatically established before starting any confidential communication between communication parties. Any file in the database can be used as the cheating content for any size of secret information to be transferred from one party to the other. The sender chooses one file from CFD and generates a corresponding index file for each transmission. The index file is then sent to the receiver instead of the original secret information to prevent interception. The sender may use different cheating file at each transmission session. The particular cheating file used for a particular transfer will be indicated to the receiver by embedding file identifier of the cheating file into the transmission.

Each record in CFD consists of a cheating file, a file identifier, and a network IP address that indicate the negotiation results with the peer site. The network IP address at the sender site verifies the receiver's IP address whereas the network IP address is the sender's address at the receiver's site. The content of the CFD could be changed each day, each week, each month, or any other period agreed to by the two communicating parties.

The simplest approach to manage CFD is to directly configure CFD by keying all the contents into the

File	File ID	Network IP address
Index.html	N0000001	140.113.23.3
MSLOGO.gif	N0000002	140.113.23.3
BkMusic.mid	N0000003	140.113.23.3
Bkground.gif	N0000004	140.113.23.3
Banner1.gif	N0000005	140.113.23.4
Banner2.gif	N0000006	140.113.23.4

Table 1: Sample of Cheating File Database at the sender site (140.113.23.10)

File	File ID	Network IP address
Index.html	N0000001	140.113.23.10
MSLOGO.gif	N0000002	140.113.23.10
BkMusic.mid	N0000003	140.113.23.10
Bkground.gif	N0000004	140.113.23.10
Banner1.gif	N0000005	140.113.23.4
Banner2.gif	N0000006	140.113.23.4

Table 2: Sample of Cheating File Database at the receiver site (140.113.23.3)

database. However, the automatic configuration must be supported for widespread deployment. Thus, the sender must negotiate with the receiver to complete the CFD.

All the files chosen in the CFD must remain in both the sender's and the receiver's site. One simple way to achieve this is to visit some specified web pages and to employ the web page files as the necessary cheating files. For example, the sender could provide the web pages and request the receiver visit these web pages. Thus, the pages' html files, the background music files, and the image files used in these web pages are downloaded to the receiver's computer. Certain FTP (File Transfer Protocol) tools can also help obtain these files. After both the sender and the receiver have received all the necessary files, the sender must bind a unique identifier to each file that will play as a cheating file in the confidential communication and send the binding results to the receiver to complete the CFD.

The sender must select a cheating file from the CFD and generate a Hexadecimal Code Index Table (HexCIT) according to this selected file to start the confidential communication. Two hexadecimal codes, from 0x00 to 0xFF, are applied to represent each byte of the cheating file. A HexCIT stores the positions where the hexadecimal characters ('0'–'9', 'A'–'F') appear in the file's hexadecimal representation. The scheme employs the position to indirectly indicate the hexadecimal character for information hiding. Different positions may be used to indicate the same hexadecimal character because each hexadecimal character may appear at many places in the file's hexadecimal representation. Therefore, the character set has only sixteen elements from '0' to '9', and 'A' to 'F'.

For example, assuming that the sender selects "MSLOGO.gif" from the CFD (Table 1), the image of "MSLOGO.gif" is displayed in Fig. 1 while Table 3 illustrates the hexadecimal representation of the first 512 bytes. Table 4 presents a HexCIT generated by the system according to the hexadecimal representation in Table 3.

Fig. 1: MSLOGO.gif

47	49	46	38	39	61	A7	00	24	00	E6	FF	00	08	08	08
10	10	10	18	18	18	18	21	21	10	18	18	08	10	10	00
08	08	39	42	4A	5A	6B	7B	52	63	73	18	21	29	42	4A
52	63	73	84	29	31	39	21	29	31	31	39	42	C0	C0	C0
73	84	9C	7B	8C	A5	8C	9C	B5	84	94	AD	6B	7B	94	63
73	8C	5A	63	73	39	42	52	63	6B	7B	A5	B5	D6	9C	AD
CE	94	A5	C6	8C	9C	BD	42	4A	5A	84	94	B5	7B	8C	AD
52	5A	6B	4A	52	63	31	39	4A	9C	AD	D6	94	A5	CE	7B
84	9C	5A	63	7B	73	7B	94	6B	73	8C	63	6B	84	52	5A
73	4A	52	6B	7B	84	A5	94	9C	BD	8C	94	B5	84	8C	AD
9C	A5	CE	73	7B	9C	94	9C	C6	31	31	39	29	29	31	4A
4A	5A	21	21	29	42	42	52	39	39	4A	31	31	42	18	18
21	29	29	39	10	10	18	08	08	10	00	00	08	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	21	F9	04
01	00	00	10	00	2C	00	00	00	00	A7	00	24	00	40	07
FF	80	10	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E
8F	90	91	92	83	1C	0A	40	97	98	98	3C	1C	84	21	99
9F	40	15	1B	03	9F	05	9C	93	A8	A9	AA	8B	22	A0	40
18	10	17	9F	37	A7	8D	1A	0D	AE	99	3D	2E	15	B9	99
1E	10	0F	9F	06	12	1A	96	C3	21	29	B8	9F	01	1B	12
00	BE	B9	01	29	1A	24	3A	A0	22	10	36	A0	28	83	B2

Table 3: Hex Representation of the first 512 bytes of Fig. 1

Character	Position	Character	Position
0	15,16,19,20, …	8	8,28,30,32, …
1	12,33,35,37,39, …	9	4,10,70,92, …
2	17,47,49, …	A	13,74,76,96, …
3	7,9,69,84,86, …	B	78,80,136,145, …
4	1,3,5,18, …	C	123,125,127,136, …
5	75,81,97,140, …	D	152,187,192,206, …
6	6,11,22,77, …	E	21,194,254,326, …
7	2,14,79,85, …	F	23,24,797,833,834, …

Table 4: Part of the HexCIT of Fig. 1

Next, the sender will generate a Secret Content Index File (SCIF) for the secret information according to the HexCIT. The SCIF is employed to encrypt the original secret information with a sequence of numbers and is sent over the network during transmission process. Thus, the eavesdropper cannot decrypt the secret information without knowing the HexCIT the SCIF referred to. To demonstrate the process of constructing the SCIF, the secret information is assumed to be:

"This is secret information."

Table 5 summarizes the hexadecimal representation of the secret information. Each hexadecimal representational character of the secret information is compared with the entry in the specified HexCIT, and a position number is randomly chosen to represent this hexadecimal character if it matches. Thus, a sequence of numbers is obtained to construct the SCIF.

54 68 69 73 20 69 73 20 73 65 63 72 65 74 20 69 6E 66 6F 72 6D 61 74 69 6F 6E

Table 5, Hexadecimal Representation of Secret Content

Table 6 illustrates a feasible number sequence according to Tables 4 and 5. Finally, the sender transmits both the selected file identifier and the resultant SCIF, instead of the original secret information, to the receiver.

75, 1, 6, 8, 7, 10, 2, 9, 17, 15, 22, 4, 85, 86, 47, 15, 79, 9, 77, 75, 11, 86, 79, 17, 22, 97, 2, 5, 47, 16, 22, 4, 6, 21, 11, 22, 77, 24, 14, 49, 6, 152, 6, 35, 85, 5, 77, 70, 6, 833, 11, 326

Table 6, A random chosen sequence of numbers

The receiver searches the CFD for the specified cheating file after receiving both the file identifier and the SCIF. The receiver can then generate the HexCIT of the cheating file by the same process employed at the sender's site. Furthermore, the receiver produces a Reversed Index Table (RIT) according to the HexCIT. The HexCIT, indexed by the hexadecimal characters, binds position information to each character as illustrated in Table 4 while the RIT employs the position information as the index to translate a number to a hexadecimal character. Table 7 illustrates the results of the reversed index of Table 4.

POSITION	CHARACTER
1	4
2	7
3	4
4	9
5	4
6	6
7	3
8	8
9	3
10	9
11	6
12	1
13	A
14	7
15	0
16	0
17	2
. . . .	
75	5
. . . .	

Table 7, Reversed Index of Table 4

The receiver can obtain each number from the SCIF and search the RIT to translate the number to a hexadecimal character because the SCIF is composed of a sequence of numbers. For example, the first two numbers, 75 and 1, can be mapped to two hexadecimal characters '54' and indicate a ASCII character 'T' by searching the RIT. Thus, the hexadecimal representation of the original secret information could be reconstructed. Fig. 2 demonstrates the entire process.

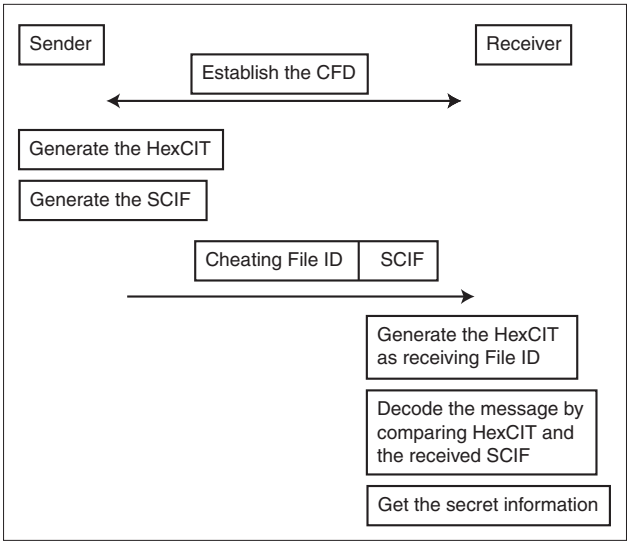

Fig. 2: The transmission process of the proposed scheme

The secret information does not have to be limited to text message in the proposed scheme. Anything that can be digitized can be processed. Indeed, both the secret information and the cheating file can appear as an image, an audio or video, a Microsoft Word document, or as some other digital format.

Security Issues

The proposed system is most vulnerable when it automatically runs the CFD setup process because all of the communications are exposed to the hackers. However, a hacker may intuitively chose to ignore meaningful information (cheating files) when receiving it as he or she may not desire such information.

The CFD can be distributed by some public key cryptosystem, such as an RSA [6] public key cryptosystem, to enhance the strength of the automatic CFD setup process. Those cryptosystems are proved to be very secure and are typically used to distribute session keys. Thus, the eavesdropper must spend a lot of time to steal the CFD with current technology. Obtaining the secret information is nearly impossible for the eavesdropper because the two communication parties can negotiate to change the content of CFD periodically. [7] discusses these cryptosystems and their associated risks and advantages.

The manual CFD setup process appears to be very secure since no third party could understand the real contents of the CFD. Moreover, the sender transmits only the file identifier and the SCIF after the CFD setup process. So it is impossible for the eavesdropper to obtain the original secret information even if he or she captured all the data during transmission since he or she does not know the meaningful cheating contents in advance.

In addition, the position number must be randomly determined during the process of generating the SCIF in order to generate two distinct SCIF even if the same secret information is sent twice. This procedure could help strengthen the security of the proposed system.

Performance Issues

Two hexadecimal characters are employed to represent each byte of the cheating file and the position's information is employed to indirectly indicate the hexadecimal character in the developed system. The meaningful cheating content cannot exceed two-giga bytes (232/2) since the position's information is stored by a 32-bit integer. Although the size of the meaningful cheating content must be limited, two giga-bytes are more than adequate for any kind of meaningful cheating content.

Furthermore, The generated SCIF is eight times the size of the original secret information since each byte of the original secret information must be represented by two integers. Several well-known compression tools can, if necessary, compress the SCIF to lighten the network traffic before sending it to the receiver.

Conclusion

This work present a novel system scheme that can transmit large quantities of secret information and provide secure communication between two communication parties without the restrictions in Lin and Lee's document protection system. Any kind of digital data can be employed as secrets or as cheating files to transmit digital secret information. The size of the secret is unlimited while the size of the cheating file is limited to two-giga bytes. However, two-giga bytes are more than adequate for any kind of cheating file.

Both steganography and cryptography are woven into the proposed scheme. The meaningful cheating content employing the concept of steganography is sent over the network without incurring suspicion while the SCIF using the position to indirectly indicate the original information is like to encrypt the secret message. In addition, the proposed procedures are simple and easy to implement. Also, the developed system has many practical personal and militaristic applications for both point-to-point and point-to-multi-point communications.

Reference

[1] Chu-Hsing Lin, Tien-Chi Lee, "A Confused Document Encrypting Scheme and its Implementation", Computers & Security, Vol. 17, No. 6, pp543-551, 1998

[2] Fred Mintzer, Gordon W. Braudaway, and Alan E. Bell, "Opportunities for Watermarking Standards", Communication of the ACM, Vol. 41, No. 7, pp.57-64, July. 1998.

[3] I. Cox et al., "A Secure, Robust Watermark for Multimedia", Proc. First Int'l Workshop Information Hiding, Lecture Notes in Computer Science No. 1, 174, Springer-Verlag, Berlin, pp.185-206, 1996.

[4] Nasir Memon and Ping Wah Wong, "Protecting Digital Media Content", Communication of the ACM, Vol. 41, No. 7, pp.35-43, July. 1998.

[5] Neil F. Johnson, Sushil Jajodia, "Exploring Steganography : Seeing the Unseen", IEEE Computer pp.26-34, Feb. 1998.

[6] Rivest, R. L., Shamir, A., and Adleman, L. M., "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", Communication of the ACM, Vol. 21, No. 2, pp.120-126, Feb. 1978.

[7] Schneier, B., "Applied Cryptography", 2nd Edition, John Wiley & Sons Inc, 1996

[8] W. Bender, D. Gruhl, N. Morimoto, A. Lu, "Techniques for Data Hiding", IBM Systems Journal, Vol. 35, No. 3&4, pp.313-335, 1996