A Document Protection Scheme using Innocuous Messages as Camouflage

[1]CHING-SHENG HSU, [2]SHU-FEN TU, [3]YOUNG-CHANG HOU
Department of Information Management
[1]Ming Chuan University
5 De Ming Rd., Gui Shan District, Taoyuan County 333, TAIWAN
dsf3@faculty.pccu.edu.tw
[2]Chinese Culture University
No.55, Huagang Rd., Shihlin District, Taipei City 11114, TAIWAN
[3]Tamkang University
151 Ying-Chuan Road, Tamshui, Taipei County 251, TAIWAN
ychou@mail.im.tku.edu.tw

Abstract: - Lin and Lee proposed a document protection scheme which utilized a meaningful document to cover the secret document. Although some researchers extended and improved their scheme, the main drawbacks of Lin and Lee's scheme are still left unsolved. The aim of our study is to propose a new document protection scheme to solve these drawbacks. Instead of encoding the secret codes into index numbers, we generate the cipher message through a series of comparisons between cheating codes and the logic operator XOR. Compared with other studies, ours have the following advantages: firstly, the selection of the cheating document needs not to be restricted to the character set of the secret document; secondly, the length of the encoded file is the same as that of the secret document; thirdly, the codes of the cipher message are almost uniformly distributed so is difficult to analyze without the key; fourthly, with the help of inner codes, our scheme is applicable to documents in any languages; finally, our scheme performs efficiently and is easy to implement.

Key-words: - Document Protection Scheme, Steganography, Cryptography, Encoding Method

1. Introduction

Confidential messages usually face a risk of being eavesdropped while being transmitted or exchanged over unsecured media. Therefore more and more researchers pay attention on how to protect the secret message in recent years. Cryptography is a kind of method to protect a secret message, such as text, images, audio, …, etc. The secret message is encoded with a predetermined key and then transferred publicly, and the key is transferred through a secure channel. Generally speaking, the encoded message is hard to be recovered without the private key within limited time and resources. However, the content of encoded message is meaningless hence is highly probable to attract eavesdroppers' interest. While cryptography is about scrambling the content of secret messages, steganography is about concealing their existence.

Steganography is a technique for cheating eavesdroppers by embedding secret messages into un-suspicious objects, called carrier. Since the traffic on the network is very heavy, it is impossible for eavesdroppers to eavesdrop messages one by one. Therefore one can use seemingly innocuous message to avoid eavesdroppers' notices. The format of the carrier can be text, images, audio, …, etc.

In the past, to hide text into another text, some people tried to change the layout of passage or to put the specified characters in specified positions of the passage [3]. Nevertheless, the size of secret messages was restricted by the size of the cheating message (*i.e.* carrier); that is, the length of the cheating message has to be much longer than that of the secret message. In 1998, Lin and Lee [1] proposed a document protection scheme, which allowed the size of the cheating message to be smaller than that of the secret message. However, their method is only suitable for single-byte characters; therefore, Yeh and Hwang [4-5] converted each character into inner code to make Lin and Lee's method to be applicable to documents containing double-byte characters. Both of Lin et al. and Yeh et al. applied IDEA to encrypt the encoded file before delivering it to receivers. Considering that the meaningless of the encrypted data may attract eavesdroppers' attention, Wang and Lu [6] developed the idea a little further. They used a gray-scale image to conceal the encoded data; hence only meaningful files were sent out. Yeh et al. and Wang et al. may bring about an improvement in some measure, but the major drawbacks of Lin and Lee's scheme are still left unsolved. Firstly, they cannot take any text as cheating message to encode the secret message. Secondly, the length of the cheating message has to be longer than that of the secret message. Thirdly, the cipher message is much larger than the secret message hence has to be compressed. Finally, the cipher message is not secure enough hence has to be encrypted again by the cryptosystem IDEA. A more detailed description about these drawbacks is given in the next section.

The purpose of this paper is to propose a new document protection scheme without those major drawbacks. That is, our scheme can use any text as cheating message to encode the secret message, and the length of the cheating message is not necessary to be longer than that of the secret message. Moreover, the codes of the cipher message are almost uniformly distributed so is difficult to analyze without the key. Besides, the length of the cipher message is the same as that of the secret message. In section 2, we will review Lin and Lee's scheme and the improvements on their scheme proposed by other researchers. In section 3, we describe the encoding and decoding algorithm of the proposed scheme, and the analysis of the security and performance are given in the same section. Section 4 is the experimental results and discussions about the proposed scheme and other researchers' schemes. Finally, the conclusion is given in section 5.

2. Literature Review

In 1998, Lin and Lee [1] proposed a document protection scheme, called Confused Document Encrypting Scheme (CDES), which can be seen as an integration of steganography and cryptography. At first, one has to select a meaningful cheating message randomly and index each character sequentially. Next, to encode a character, say *P*, of the secret message, called plaintext, one searches for the same character in the cheating message. If more than one character is the same as *P*, one of them is selected randomly, and its index is recorded in a plaintext index file (PIF). In other words, the character *P* is encoded into a number, which represents the position of the same character in the cheating message. Assume that the secret message is “I love you” and that the cheating message is “I have played guitar for a long time”. The third character “l” of the secret message can be encoded as 9 or 28, which are indices of “l” in the cheating message. By doing so, the secret message was transformed into a Plain-text Index File (PIF). Then PIF is compressed and encrypted by IDEA. Finally, the cheating message and encrypted PIF was delivered to the receiver. To decode the secret message, one just picks characters from the cheating message according to the content of PIF.

Lin and Lee’s method is suitable for the language with small character set, such as English. The number of characters in English is at most 128, while that of characters in Chinese is at least 5401. To apply Lin and Lee’s method to double-byte character-based language, such as Chinese, Yeh and Hwang [4-5] utilized the encoding method, Big5, to convert Chinese characters into hexadecimal codes; hence the character set is reduced to sixteen elements, ‘0’-‘9’ and ‘A’-‘F’. Both of the secret and cheating messages are converted into inner codes first. Then each inner code of the secret message is encoded according to the inner codes of the cheating message by means of CDES. After that, the secret message is transformed into a Chinese Document Index File (CDIF). The CDIF is compressed and encrypted by IDEA as well. In the decoding phase, one has to convert each character of the cheating message into inner code first. Then each inner code of the secret message is picked from the converted cheating message according to the content of CDIF. Finally, the secret message was recovered by means of the appropriate encoding method.

Since the encrypted index file may attract eavesdroppers’ interest, Wang and Lu [6] utilized an information hiding method proposed in [7] to conceal it. To encode a document containing both English and Chinese characters, they select two cheating messages, in English and in Chinese, respectively. Then, English characters are encoded into a PIF by Lin and Lee’s method, whereas Chinese characters are encoded into a CDIF by Yeh and Hwang’s method. The two index files were encrypted by IDEA and concealed into the least significant bits of a gray-scale image. Finally, two cheating messages and the camouflage image were delivered to the receiver.

There are some major and obvious drawbacks of CDES and other researchers’ methods. Firstly, the cheating message has to contain all elements of the character set of the secret message. In other words, the character set of the secret message is a subset of that of the cheating message. Otherwise, the characters not existed in the cheating message cannot be encoded. When such situation happens, they have to look for another cheating message suitable for the secret message. Hence, the selection of the cheating message is not truly random. Consider the following example quoted from [5].

Secret message：二八四師向林口集結。

Cheating message：【職籃消息】羅興樑於職籃五年將暫披達欣戰袍。

The hexadecimal codes of the secret message and cheating messages are as follows.

Secret codes: A4 47 A4 4B A5 7C AE 76 A6 56 AA 4C A4 66 B6 B0 B5 B2 A1 43

Cheating codes: A1 69 C2 BE C4 78 AE F8 AE A7 A1 6A C3 B9 BF B3 BC D9 A9 F3 C2 BE C4 78 A4 AD A6 7E B1 4E BC C8 A9 DC B9 46 AA 59 BE D4 B3 54 A1 43

Obviously, the code ‘0’ in the secret message does not exist in the cheating message. This situation can only be solved by adding more characters to the cheating message or looking for another cheating message.

Secondly, if some character of the secret message exists only once in the cheating message, and if by any chance that character occurs many times in the secret message, the same number will occur many times in the index file as well. Therefore, the CDES doesn't conform to security of a cryptosystem since the probability of each character occurring in the cheating message is different. On the contrary, if some characters of the cheating message occur many times but never exist in the secret message, it is obvious that such characters are useless for encoding the secret message. Such cases happened many times in the examples presented in Wang and Lu's paper. Accordingly, we have sufficient reason of thinking that the number of cheating characters should be much more than that of secret characters to release the problems above.

Thirdly, in fact, the character set of a double-byte character-based language also contains the single-byte ASCII characters. Therefore, a document containing both English and Chinese characters can be converted into inner codes by the same encoding method, Big5. It is not necessary to separately encode English and Chinese characters with two cheating message in different languages. Therefore, Wang and Lu took a superfluous action to encode a bilingual document.

Fourthly, in the above researches, each character of the secret message is encoded into an integer. In many systems, the size of the integer data type ranges from two to eight bytes. Nevertheless, the size of an ASCII character is only 7 bits, and that of a hexadecimal code is only 4 bits. Therefore, the size of PIF or CDIF is much larger than of the secret message. That is why the index file has to be compressed before sending it. It is more ideal that the size of the encoded result is the same as that of the plaintext.

In conclusion, though the length of the cheating message is not necessary to be longer than that of the secret message in CDES, nevertheless, in fact, the length of the character message usually has to be longer than that of the secret message as we mentioned above. At least, the content of the cheating message is still restricted by the secret message. Hence not any cheating message can be used to encode the secret message. Moreover, the index file may leave a clue to the secret message hence is not secure enough. From the above literatures review, we can see that Yeh and Hwang and Wang and Lu only show a slight improvement on CDES but don't solve the major drawbacks. In this paper, we proposed a new document protection scheme without these drawbacks. To make our scheme applicable to characters in any language, we also encode the inner codes of the secret message instead of processing the characters directly.

3. The Proposed Scheme

3.1 The Encoding Phase

To encode a secret message, we randomly select a cheating message and convert both of them into hexadecimal inner codes by appropriate encoding method. Suppose that $S = (S_1, S_2, \ldots, S_n)$, $H = (H_1, H_2, \ldots, H_m)$, and $C = (C_1, C_2, \ldots, C_n)$ denote the hexadecimal codes of the secret message, cheating message, and cipher message, respectively. Note that a hexadecimal code ranges from '0' to '9' and 'A' to 'F', so it can be represented as a 4-bit binary string. For each code S_i, we can generate a corresponding 4-bit code D_i (= $(d_1d_2d_3d_4)_2$) from H. To generate each bit of D_i, we randomly select two codes h and h' ($h \neq h'$) from H using a pseudo random number generator with a seed *key*. Then $d_k = 1$ if $h > h'$ and $d_k = 0$ if $h < h'$ for $k = 1..4$. If $h = h'$, h' is abandoned and the selection is continued until the two codes are different. After D_i is generated, we can encode S_i into

C_i according to the following equation:

$$C_i = S_i \oplus D_i, \qquad (1)$$

where '$\oplus$' denotes the "XOR" operator. In this way, the secret message can be transformed into a cipher message C. Note that the seed of the pseudo random number generator is a private key and must be send out through a secure channel.

```
Algorithm  Encoding
Inputs:  (1) The hexadecimal codes of the cheating message $H = (H_1, H_2, \ldots, H_m)$
         (2) The hexadecimal codes of the secret message $S = (S_1, S_2, \ldots, S_n)$
         (3) The seed key of the pseudo random number generator
Output:  The cipher message $C = (C_1, C_2, \ldots, C_n)$
Step 1:  Set $i = 1$.
Step 2:  Let $D_i$ denote a 4-bit binary string $(d_1d_2d_3d_4)_2$. For $k = 1..4$, do Step 3.
Step 3:    Randomly select two different codes $h$ and $h'$ from $H$ with the pseudo random number generator seeded by key. Then $d_k = 1$ if $h > h'$, $d_k = 0$ if $h < h'$ and reselect $h'$ again if $h = h'$.
Step 4:  Encode $S_i$ into $C_i$ according to $C_i = S_i \oplus D_i$ and set $i = i + 1$.
Step 5:  Repeat Step 2 to Step 4 until $i > n$.
```

3.2 The Decoding Phase

Before recovering the secret message, the receiver has to convert the corresponding cheating message into hexadecimal codes by an appropriate encoding method in advance. To decode S, the receiver has to generate D from H by the same procedure presented in the previous section. After D is generated, we can decode S according to the following equation:

$$S_i = C_i \oplus D_i, \qquad (2)$$

where $i = 1..n$. Using the same seed *key* of the pseudo random number generator, we can recover the correct D corresponding to S. Finally, S is converted into secret message by the appropriate encoding method.

```
Algorithm  Decoding
Inputs:  (1) The cipher message $C = (C_1, C_2, \ldots, C_n)$
         (2) The hexadecimal codes of the cheating message $H = (H_1, H_2, \ldots, H_m)$
         (3) The seed key of the pseudo random number generator
Output:  The hexadecimal codes of the secret message $S = (S_1, S_2, \ldots, S_n)$
Step 1:  Set $i = 1$.
Step 2:  For $k = 1..4$, do Step 3.
Step 3:    Randomly select two different codes $h$ and $h'$ from $H$ using the pseudo random number generator seeded by key to generate $D_i = (d_1d_2d_3d_4)_2$. That is, $d_k = 1$ if $h > h'$, $d_k = 0$ if $h < h'$ and reselect $h'$ again if $h = h'$.
Step 4:  Decode the hexadecimal code $S_i$ of the secret message according to $S_i = C_i \oplus D_i$ and set $i = i + 1$.
Step 5:  Repeat Step 2 to Step 4 until $i > n$.
```

After the hexadecimal codes of the secret message are recovered, we have to use the same encoding method to convert them into human-readable secret message.

3.3 Security Analysis

Before discussing the security of our scheme, we explain the probability of $h > h'$ and that of $h < h'$ in detail. Suppose that the set of all possible hexadecimal codes in H is $\{\lambda_i \mid i = 1..16\}$, where $\lambda_i < \lambda_{i+1}$. We assume that $\Pr(\lambda_i)$ denotes the probability of code λ_i appeared in H. Now, we compute the probability of $h < h'$. If the randomly selected $h = \lambda_i$, then the randomly selected h' must be any one of $\lambda_{i+1}, \lambda_{i+2}, \ldots, \lambda_{16}$. Therefore, the probability of $h < h'$ can be denoted by

$$\alpha = \Pr(\lambda_1) \times \sum_{k=i+1}^{16} \Pr(\lambda_k)$$

Next, we compute the probability of $h > h'$. If the randomly selected $h = \lambda_i$, then the randomly selected h' must be any one of $\lambda_{i-1}, \lambda_{i-2}, \ldots, \lambda_1$. Similarly, the probability of $h < h'$ can be denoted by

$$\beta = \Pr(\lambda_i) \times \sum_{k=1}^{i-1} \Pr(\lambda_k)$$

Accordingly, the expect probability of $h < h'$ can be represented by

$$\mathrm{E}(\alpha) = \frac{1}{15}\sum_{i=1}^{15}\left(\Pr(\lambda_i) \times \sum_{k=i+1}^{16}\Pr(\lambda_k)\right),$$

and the expect probability of $h > h'$ can be represented by

$$\mathrm{E}(\beta) = \frac{1}{15}\sum_{i=2}^{16}\left(\Pr(\lambda_i) \times \sum_{k=1}^{i-1}\Pr(\lambda_k)\right).$$

Theorem 1 The expect probability of $h < h'$ and that of $h > h'$ are equal.

Proof.

Let

$$A = \sum_{i=1}^{15}\left(\Pr(\lambda_i) \times \sum_{k=i+1}^{16}\Pr(\lambda_k)\right)$$

and let

$$B = \sum_{i=2}^{16}\left(\Pr(\lambda_i) \times \sum_{k=1}^{i-1}\Pr(\lambda_k)\right).$$

Thus, the expect probabilities can be rewritten as

$$\mathrm{E}(\alpha) = \frac{A}{15}$$

and

$$\mathrm{E}(\beta) = \frac{B}{15}.$$

Now, we show that $A = B$ as follows:

$$B = \sum_{i=2}^{16}\left(\Pr(\lambda_i) \times \sum_{k=1}^{i-1}\Pr(\lambda_k)\right)$$

$$= \Pr(\lambda_2) \times \sum_{k=1}^{1}\Pr(\lambda_k) + \Pr(\lambda_3) \times \sum_{k=1}^{2}\Pr(\lambda_k) + \ldots$$

$$+ \Pr(\lambda_{16}) \times \sum_{k=1}^{15}\Pr(\lambda_k)$$

$= \Pr(\lambda_2) \times (\Pr(\lambda_1)) + \Pr(\lambda_3) \times (\Pr(\lambda_1) + \Pr(\lambda_2)) + \ldots + \Pr(\lambda_{16}) \times (\Pr(\lambda_1) + \Pr(\lambda_2) + \ldots + \Pr(\lambda_{15}))$

$= \Pr(\lambda_1) \times (\Pr(\lambda_2) + \Pr(\lambda_3) + \ldots + \Pr(\lambda_{16})) + \Pr(\lambda_2) \times (\Pr(\lambda_3) + \Pr(\lambda_4) + \ldots + \Pr(\lambda_{16})) + \ldots + \Pr(\lambda_{15}) \times (\Pr(\lambda_{16}))$

$$= \sum_{i=1}^{15}\left(\Pr(\lambda_i) \times \left(\Pr(\lambda_{i+1}) + \Pr(\lambda_{i+2}) + \ldots + \Pr(\lambda_{16})\right)\right)$$

$$= \sum_{i=1}^{15}\left(\Pr(\lambda_i) \times \sum_{k=i+1}^{16}\Pr(\lambda_k)\right) = A$$

Therefore, we have that

$$\mathrm{E}(\beta) = \frac{B}{15} = \frac{A}{15} = \mathrm{E}(\alpha).$$

Suppose (Π, X, K, E, Δ) is a cryptosystem where Π, X, K, E, and Δ are finite sets of possible plaintexts, ciphertexts, keys, encryption functions, and decryption functions, respectively. For each key $K \in$ K, there is an encryption rule $e_K \in$ E and a corresponding decryption rule $d_K \in \Delta$. Each $e_K : \Pi \rightarrow$ X and $d_K :$ X $\rightarrow \Pi$ are functions such that $d_K(e_K(x)) = x$ for every plaintext element $x \in \Pi$. In our scheme, the cipher message C_i (= $(c_1c_2c_3c_4)_2$) is generated by performing the logic operation XOR on a code D_i (= $(d_1d_2d_3d_4)_2$) generated from H and S_i (= $(s_1s_2s_3s_4)_2$). Therefore, the bits of S_i can correspond to plaintexts, the bits of C_i can correspond to ciphertexts, and the bits of D_i can correspond to keys. It is clear that Π = X = K = {0, 1} in our scheme. For each key $K \in$ K, the encryption and decryption rules are defined as $e_K(x) = x \oplus K$ and $d_K(y) = y \oplus K$, respectively. Each time we make a comparison between two distinct randomly selected codes h and h' to generate a key K. That is, $K = 1$ if $h > h'$ and $K = 0$ if $h < h'$. We have proved that the two events $h > h'$ and $h < h'$ have equal probability. Therefore, the probability distribution of K can be defined as Pr(**K** = K) = 1/2 for every $K \in$ K, where **K** denotes a random variable defined on K. Then, we have that |K| = |X| = |Π| = 2. Accordingly, every key is used with equal probability 1/|K| = 1/2, and for every $x \in \Pi$ and $y \in$ X, there is a unique key K such that $e_K(x) = y$. Hence, without the appropriate seed of the pseudo random number generator, one can have difficulty to gain information about the secret message by analyzing the cipher message.

3.4 Performance Analysis

The input size, denoted as n, of the encoding algorithm is the number of elements in S. For each element, at least four comparisons and one logic operation XOR are done. Through the whole loop, there are about $4 \times n$ comparisons and n logic operations are done. Therefore, the time complexity of the encoding algorithm is $O(n)$. Since the input size and structure of the decoding algorithm are similar to those of the encoding algorithm, its time complexity is $O(n)$ as well.

4. Experiment Results and Discussions

4.1 Experiment Results

Our scheme can use any text as cheating message, so we use two different cheating messages, which respectively lack of code '8' and code '7', to encode the secret message. Table 1 is the secret message and its hexadecimal codes converted by the encoding method "Big5". Table 2 and Table 3 show two different cheating messages and their hexadecimal codes respectively. Table 4 and Table 5 are the cipher messages of the secret message using the two cheating messages as camouflage respectively.

Table 1. The secret message and its hexadecimal codes

The secret message
內政部警政署函：本署將於4月1日起展開爲期一個星期之治平專案，本次治平專案檢肅對象詳見附件。請所屬各單位針對本專案研擬一份施行計劃，並在2月底前提報本署參考，以做爲專案結束後，各單位工作檢討之基準。各單位之所屬同仁於治平專案期間之表現，亦爲本年度績效考核之重點。
Hexadecimal codes of the secret message
A4 BA AC 46 B3 A1 C4 B5 AC 46 B8 70 A8 E7 A1 47 A5 BB B8 70 B1 4E A9 F3 34 A4 EB 31 A4 E9 B0 5F AE 69 B6 7D AC B0 B4 C1 A4 40 AD D3 AC 50 B4 C1 A4 A7 AA 76 A5 AD B1 4D AE D7 A1 41 A5 BB A6 B8 AA 76 A5 AD B1 4D AE D7 C0 CB B5 C2 B9 EF B6 48 B8 D4 A8 A3 AA FE A5 F3 A1 43 BD D0 A9 D2 C4 DD A6 55 B3 E6 A6 EC B0 77 B9 EF A5 BB B1 4D AE D7 AC E3 C0 C0 A4 40 A5 F7 AC 49 A6 E6 AD 70 B9 BA A1 41 A8 C3 A6 62 32 A4 EB A9 B3 AB 65 B4 A3 B3 F8 A5 BB B8 70 B0 D1 A6 D2 A1 41 A5 48 B0 B5 AC B0 B1 4D AE D7 B5 B2 A7 F4 AB E1 A1 41 A6 55 B3 E6 A6 EC A4 75 A7 40 C0 CB B0 51 A4 A7 B0 F2 B7 C7 A1 43 A6 55 B3 E6 A6 EC A4 A7 A9 D2 C4 DD A6 50 A4 AF A9 F3 AA 76 A5 AD B1 4D AE D7 B4 C1 B6 A1 A4 A7 AA ED B2 7B A1 41 A5 E7 AC B0 A5 BB A6 7E AB D7 C1 5A AE C4 A6 D2 AE D6 A4 A7 AD AB C2 49 A1 43

Table 2. The cheating message 1 and its hexadecimal codes

The cheating message 1
梵高颱風直撲北臺灣，十九日晚間北市宣佈明日正常上班上課，但北市高中職暑期輔導課停課一天。
Hexadecimal codes of the cheating message 1
B1 EB B0 AA BB E4 AD B7 AA BD BC B3 A5 5F BB 4F C6 57 A1 41 A4 51 A4 45 A4 E9 B1 DF B6 A1 A5 5F A5 AB AB C5 A7 47 A9 FA A4 E9 A5 BF B1 60 A4 57 AF 5A A4 57 BD D2 A1 41 A6 FD A5 5F A5 AB B0 AA A4 A4 C2 BE B4 BB B4 C1 BB B2 BE C9 BD D2 B0 B1 BD D2 A4 40 A4 D1 A1 43

Table 3. The cheating message 2 and its hexadecimal codes

The cheating message 2
全市除基隆河五、六、九號水門及淡水河三號水門，將

視雨勢及水位變化宣布關閉時間，其餘水門晚間十一時關閉。
Hexadecimal codes of the cheating message 2
A5 FE A5 AB B0 A3 B0 F2 B6 A9 AA 65 A4 AD A1 42 A4 BB A1 42 A4 45 B8 B9 A4 F4 AA F9 A4 CE B2 48 A4 F4 AA 65 A4 54 B8 B9 A4 F4 AA F9 A1 41 B1 4E B5 F8 AB 42 B6 D5 A4 CE A4 F4 A6 EC C5 DC A4 C6 AB C5 A5 AC C3 F6 B3 AC AE C9 B6 A1 A1 41 A8 E4 BE 6C A4 F4 AA F9 B1 DF B6 A1 A4 51 A4 40 AE C9 C3 F6 B3 AC A1 43

Table 4. The cipher message of the secret message using the cheating message 1 as camouflage

91 81 1D D3 88 FA 3B 60 1F FF 69 8F D5 98 54 9A 9C A8 0D AB EC 1B BA 44 C3 99 3C EC B9 BA E1 6E 71 12 21 C4 77 47 65 FC 59 D7 52 4A 3B C3 AF 14 3F F8 57 4D 9E 74 48 74 93 6E 38 D6 F8 24 B1 8F 59 CF BE 3A C0 1A 33 8C DD 30 EA D9 EE 12 ED 39 C5 0D 3F B0 39 6D DC 86 B6 1A EC AB 18 23 D1 EC 53 44 40 1F 91 93 21 42 06 D4 14 4E 66 9C 5D 8C DB F8 19 75 B9 F1 DC EE 15 1C 79 D9 BA CD C2 29 F4 50 B9 7E D9 FF 29 B1 9E 1E 0C F6 32 A3 D0 2C 27 D0 A4 49 AD 8F E8 9D 61 3E B2 7E B7 E7 82 1D C3 46 F0 95 68 88 CD 5E 65 FE 9A 74 F8 5B AE E4 75 91 BD 5F EC D6 9F 9B 58 6D 06 5F F8 43 27 CE 52 9E F4 FF A8 0E BB D7 BB 51 56 1C 05 F9 68 71 E9 53 BC 94 2E BB 4D 52 9C 8A F6 F7 6A AB 1C 69 7C 3D BE 95 DC A3 20 70 D2 56 90 53 27 B6 A6 B5 2B 90 C4 D2 0F F5 B5 9B 6F 91 6D 13 98 56 72 F1 9E BC F2

Table 5. The cipher message of the secret message using the cheating message 2 as camouflage

5F E9 7B 11 4C 10 39 C2 DD 75 01 8B 15 18 98 9E 50 66 E1 21 2E B5 F2 08 A9 B1 96 2A 13 D0 4F 66 1D D2 2F 4E D9 C7 41 18 D3 79 90 28 93 A1 2B FC 95 D8 1D E7 18 92 A2 78 3F 8C BC 52 BA C6 79 8B BB 45 38 FC 66 50 5B 6C DD 7E 02 95 E6 3C 01 DD 43 41 F1 B4 D5 2D DC AE 12 B0 E2 ED 9C C3 3F 8C F7 A6 04 11 D1 71 69 66 CA FA 1A E2 0A 30 9F A2 7B 54 57 D3 DD D5 3A CC 97 9A 97 DD DA 4F AE 4F F2 92 DB 96 3D 5B 61 F1 16 D2 4A 7C 1C CF 5C 4E 07 38 00 C5 ED E9 8A 5B A7 1E 18 FC 19 0D 0E 7F E9 4E 7A 59 0A E6 41 9C 43 BC B2 76 5E 5B C8 2E 91 5D 15 9B 0A 14 B1 17 30 C1 4A 3B 9E 2D 01 C6 38 9E 9E BB 48 8E D3 B3 F7 99 54 10 65 37 48 FB CB 31 B4 D0 08 95 C9 78 1E EE D0 1B 66 A9 56 ED 98 95 D2 17 5A 41 02 D2 9C DE BC 37 AD F4 06 51 0D 16 02 70 C3 7D 7F B7 C3 FF C7 97 14 10 FA 1B D8 50 9C

4.2 Discussions

Table 6 shows the frequencies of every hexadecimal codes occurred in the cheating message 1, the cheating message 2, and the secret message. Observing Table 6, we can see that the code ‘8’ and the code ‘7’ never occur in the cheating message 1 and the cheating message 2, respectively. Nevertheless, the two codes show more than once in the secret message. The way to handle this problem by CDES is to look for another cheating message or add more characters to the cheating message until the code ‘8’ exists. This problem seems to be solved, but we have to change the cheating message constantly whenever this situation happens. In addition, because we cannot know the hexadecimal codes of the cheating message before converting it, it is highly probable that we have to try several times till a suitable cheating message is found. Moreover, it may happen that the new cheating message indeed includes the code ‘8’ or ‘7’, but lacks of another code. Accordingly, from what has been mentioned above, being unable to take any character as a cheating message is a serious problem of CDES. Besides, each code of the secret message is encoded into an integer, so the index file may be too large and need to be compressed. In our scheme, we use both the comparison operation and the logic operation “XOR” to encode the hexadecimal codes of the secret message, so we can use any characters as cheating message. Additionally, the cipher message has the same size as the secret message.

Table 6. Frequencies of the codes in the cheating message and the two secret messages

Hexadecimal codes	Frequencies in the cheating message 1	Frequencies in the cheating message 2	Frequencies in the secret message
0	5	3	23
1	13	12	30
2	5	5	11
3	2	6	20
4	20	31	48
5	16	12	29
6	4	11	29
7	6	0	32
8	0	5	11
9	4	8	12
A	36	47	106
B	33	22	65
C	6	15	29
D	11	4	33
E	6	9	28
F	9	14	12
Sum	176	204	518

In CDES, if the secret message has more elements than the cheating message has, the same index numbers will show many times in the index file. For instance, the code ‘6’ shows 29 times in the secret message, but only four times in the cheating message 1. Therefore, only four different numbers can be used to encode the code ‘6’. Since this kind of index file may leave clues to eavesdroppers, the authors utilize IDEA to encode it again. On the contrary, in our scheme, the cipher message is worthless to guess the secret message. Therefore, the cipher message generated by our scheme needs not to be encoded by other cryptosystems again.

Wang and Lu select an English message and a Chinese message to encode English and Chinese characters, respectively. Therefore, two index files are generated: one corresponds to English characters and the other corresponds to Chinese characters. In fact, the character set of Big5 reserves the codes between 0x21 and 0x7E for the character set of ASCII [2], so the message containing both English and Chinese can be converted by the same encoding method (*i.e.* Big5). It is not necessary to look for different cheating messages to cope with English and Chinese secret messages.

5. Conclusions

In this paper, we present a method to utilize a meaningful message to cover a secret message. The cheating and secret messages are converted into hexadecimal codes first. For each code of the secret message S_i, we use the comparison operation to generate the corresponding hexadecimal code D_i from the hexadecimal codes of the cheating message H. Each code of the secret message S_i is “XOR”-ed with its corresponding code D_i to generate the final cipher code C_i.

We arrive at the conclusion that the drawbacks of CDES are solved in our scheme. We can choose any characters as a cheating message, and the size of the cipher message is the same as that of the secret message. Hence it is not necessary to compress the cipher message. In addition, our scheme provides more security than other researchers do. Moreover, the performance of the proposed scheme is good since the time complexity is $O(n)$, and the implementation is easy as well. By converting the message into inner codes, we can apply our scheme to encode messages in any language.

6. Acknowledgment

This work was supported in part by a grant from the National Science Council of the Republic of China under the project NSC 96-2221-E-034-016-MY2 and

97-2221-E-130-019-.

References:

[1] C.H. Lin and T.C. Lee, A Confused Document Encrypting Scheme and Its Implementation, *Computers & Security*, Vol.17, No.6, 1998, pp. 543-551.

[2] K. Lunde, *Chinese, Japanese, Lorean & Vietnamese Information Processing*, O'Reilly & Associates, Inc., 1999

[3] F.J. Neil and J. Sushil, Exploring Steganography: Seeing the Unseen, *IEEE computer*, Vol.31, No.2, 1998, pp.26-34.

[4] W.H. Yeh and J.J. Hwang, Hiding Digital Information Using a Novel System Scheme, *Computers & Security*, Vol.20, No.6, 2001a, pp.533-538.

[5] W.H. Yeh and J.J. Hwang, A Scheme of Hiding Secret Chinese Information in Confused Documents, *Journal of Information Management*, Vol.7, No.2, 2001b, pp.183-191.

[6] S. J. Wang and C K. Lu, A Scheme of Non-sensible Document in Transit with Secret Hiding, *Journal of Information Management*, Vol.9, No.2, 2003, pp.169-182.

[7] S.J. Wang and K.S. Yang, A Scheme of High Capacity Embedding on Image Data Using Modulo Mechanism, *Proceedings of The Second International Workshop on Information Security Applications*, 2001, pp.299-309.