Proceedings of International Conference
on Cryptology and Information Security

A Confused Document Encrypting Scheme and Its Implementation

Chu-Hsing Lin and Tien-Chi Lee

Department of Computer and Information Sciences,
TungHai University, Taichung, Taiwan, 407 R.O.C.
E-mail: chlin@s867.thu.edu.tw peter@sun1.cis.thu.edu.tw
Phone: (04) 359-0135 (04) 699-5050
Fax: (04) 359-6557

Abstract

In this paper, we propose a new document protecting scheme, called Confused Document Encrypting Scheme (CDES). The method used by the present scheme is a completely new one for document protections. The main spirit of the secure system is that we do not need send encrypted plaintext directly to message receiver, however, we send a meaningful cheating text and an encrypted special index file to message receiver. The most significant characteristic of this method is that it owns cheating function, and no hackers can exactly know the cheating text he gets is real or false message. Besides, the cryptosystem uses probabilistic idea and IDEA to encrypt the special index file. And we have developed a software package to implement the proposed encrypting and decrypting algorithms.

keywords: *DES, IDEA, probabilistic cryptosystem, document protection, block cipher, character position table, plaintext index file, cheating text.*

1. Introduction

In the past, most researchers in the area of data security have put a great effort on developing cryptosystems which are difficult and complicated to reverse from the ciphertext into the original data. But all the ciphertext form will become meaningless after the original data have been encrypted. When a hacker gets these data, meaningless words, he exactly know that these data has been encrypted. Then if he want, he will try to decrypt these data. In other words, as soon as he gets these encrypted data ,he exactly knows there is some important message he has obtained.

In this paper, we propose that a cheating text instead of encrypted plaintext is sent to receivers. The cheating text can use any kind of text form which is constructed by ASCII code. For example: text articles, source programs, batch files ...etc. And the cheating text can have no relation with the original plaintext. But, the cheating text has a restriction : its number of character's kinds can't less than the number of character's kinds used by the plaintext.

For example, if the plaintext is " I love you! " The kinds of ASCII code's characters are ' I ', 'l', 'o' , 'v' , 'e' , 'space' , 'y' , 'u', '.' , totally 9 kinds of characters . And the cheating text must use some words at least including 9 kinds of ASCII code's characters. We can write a cheating text such as " I have played guitar for a long time." Then according to the plaintext and depending on the cheating text we can make a special index file, called plaintext index file (PIF). Finally, we compress and encrypt the index file. Therefore, the real data we send via the channel is the cheating text and the encrypted index file.

In our document protection system, the security of the plaintext index file is very important. And, we use IDEA method [1,10] to encrypt the index file because of its fast encrypting operation. IDEA is a block cipher [2]. It uses 128-bit secret key to encrypt 64-bit plaintext block and generates 64-bit cipher block. Its encrypting process is similar to DES [3] ,but it is more secure than DES. The most obvious sample is the length of the secret key : DES's secret key is 64 bits, however, the IDEA's is 128 bits. We know that DES has suffered a lot of attacks in recent years, especially coming from " Known-Plaintext Attack " [4]. So we use IDEA in this cryptosystem. Of course, the IDEA is not the only way to encrypt the plaintext index file. The most suitable method should depend on the situation where the cryptosystem was used.

Further, for the public application purpose, we generate the IDEA's key randomly and use any public key system such as El-Gamal's scheme to encrypt this key. By this way, the CDES can be applied in an open system.

2. Review of IDEA

As mentioned previously, we use IDEA to encrypt the plaintext index file. So, before introducing the algorithm of the CDES, we review the IDEA method first. IDEA is a block cipher cryptosystem and it is designed by Xuejia Lai and James Massey in 1990 to substitute DES. IDEA uses 128-bits key to encrypt 64-bit plaintext block and output 64-bit cipher block. The designing goal of the IDEA can be separated into two parts: secret level and easily practicing level .

(1). In secret level, it including:

a). The length of block : The length of block affect the complexity of an efficient encrypting function.

b). The length of secret key: The length of secret key should not be cryptanalyzed easily.

c).Confusion: IDEA uses three kinds operation to confuse data.

(1) Bit-by-Bit XOR .

(2) Addition of integers modulo 2^{16} with inputs and outputs treated as unsigned 16-bit integers.

(3) Multiplication of integers modulo $2^{16}+1$ with inputs and outputs treated as unsigned 16-bit integers, except that a block of all zeros is treated as 2^{16}.

d).Diffusion: IDEA uses Multiplication/ Addition (MA) structure to get its diffusing goal.

(2). In easily practicing level:

IDEA uses subblocks and simple operations to reduce the difficulty of practice. In the following, we'll introduce the encrypting method and decrypting method of the IDEA briefly.

There are two input data for the encrypting function of the IDEA: plaintext and a secret key and the encrypting function is built from the repeated encryption of eight steps and an output transformation. The encrypting function separates the plaintext (64 bits) into four subblocks (16 bits). And the Subkey Generator transfers the secret key (128 bits) into fifty-two 16-bit subkeys. These subkeys are used for encryption and transformation. After processing subblocks and subkeys, we get the ciphertext we want.

As to the decryption, we need the 128-bit secret key and the ciphertext. The decrypting structures are the same as the encrypting structures. They are just different in their processing direction.

In the following, we'll introduce the CDES algorithm formally.

3. Encrypting Algorithm

In our cryptosystem, message sender must prepare two pieces of document. One is a real message and the other is a cheating message. The CDES will use the cheating message to transfer plaintext into the plaintext index file (PIF). Then CDES compress and encrypt the PIF. To avoid intruder to modify the cheating text, we use Digital Signature Algorithm (DSA)[11] to verify the cheating text. Finally, we send out the signed cheating message and the encrypted PIF. In the following, we describe the input, output and steps of the encrypting algorithm for the document sender.

Input : Plaintext and several cheating texts, the kinds of the cheating text's characters must be equal to those of the plaintext's characters at least ; receiver's public key and sender's private key.

Output : Encrypted PIF and the signed cheating text .

Step 1: Use the cheating text to generate the character's position table (CPT).
CPT : It uses each different character in the cheating text as an entry and records all positions of each character's appearance in the cheating text.

Step 2: Using CPT and the plaintext, we can make plaintext index file (PIF) by random.
PIF: It depends on each character in plaintext to find out the same character in CPT and randomly chose a position record of this character.

Step 3: Compress the PIF. Any compressing method can be used.

Step 4: Randomly generate one key with 128 bits. The key is for encrypting the PIF by IDEA.

Step 5: Randomly generate an ID for the cheating text.

Step 6: Encrypt the compressed PIF. Here, we use IDEA method.

Step 7: Encrypt the cheating text's ID with receiver's public key and put it in the head of the PIF. Here , we use El-Gamal public

key system.

Step 8: Use receiver's public key to encrypt the 128-bit key and put it in the 2-nd line of the PIF. Here, we use El-Gamal public key system.

Step 9: Use one-way hash function to hash the cheating text. Here, we use SHA.

Step 10: Use digital signature system to sign the hashed cheating text with sender's private key. Here, we use DSA.

Step 11: Send out the signed cheating text and the encrypted PIF.

In the above Step 7, the reason of encrypting ID independently and putting the ID in the head of the encrypted PIF is to reduce the time required for decrypting process. When we decrypt the encrypted PIF, we don't know which cheating text is corresponding to the PIF. So, we decrypt the ID first and find out the correct cheating text. After finding out the correct cheating text, we decrypt the entirely PIF. By this way, we needn't decrypt the all encrypted PIF every time. We just decrypt it after finding out the corresponding cheating text. Now, we give an example.

Example 1.

We assume that

plaintext : Cat is my pet.

cheating text : Computer security is important.

And according to the cheating text, we generate the CPT:

character	position record
C	1
o	2 , 25
m	3 , 23
p	4 , 24
u	5 , 13
t	6 , 16 , 27 , 30
e	7 , 11
c	12
r	8 , 14 , 26
space	9 , 18 , 21
s	10 ,20 ,
i	15 ,19 , 22
y	17
a	28
n	29
	31

Then we compare each character in plaintext with CPT 's characters and randomly chose a position record to make the PIF.

PIF is : 1 28 16 21 15 20 18 3 17 9 24 7 6 31 (not the only way)

After making the PIF, we give a random ID to the cheating text :

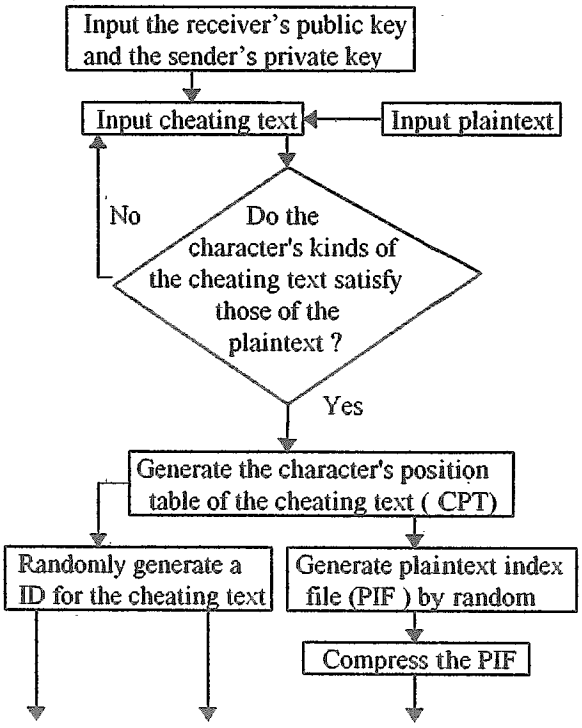
NO:236785
Computer security is important.

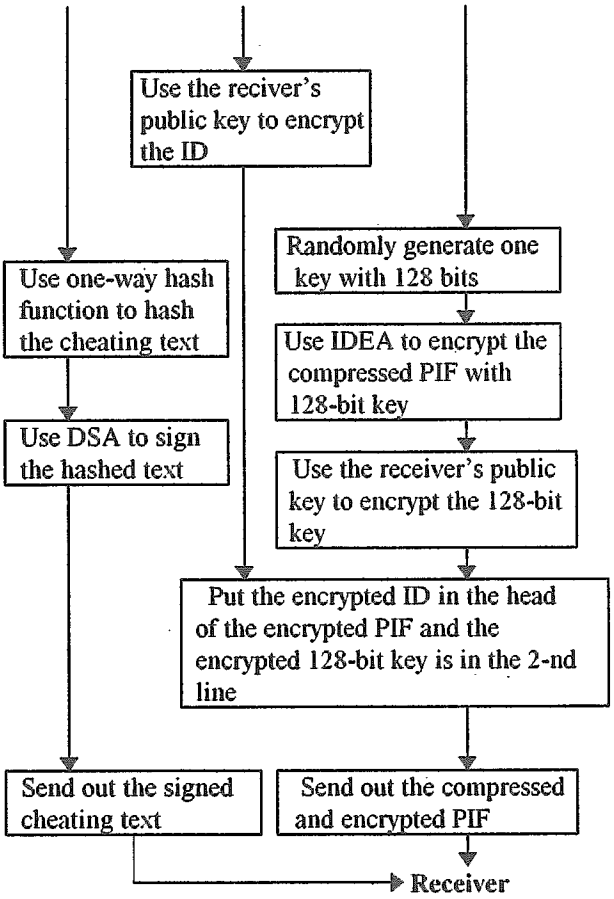
In this time, CDES randomly generate a 128-bit key (ex:12345678abcdefgh) to encrypt the PIF by IDEA. The encrypted PIF is:
2741a28ce9e72278318885b4e2a7f547cc030ec2281
2bb8d4033c39383c28557abf6328e9313f685729bea
e8182675df

Then the CDES use receiver's public key to encrypt the ID and the IDEA's key.
After continual processing, we use SHA then use DSA to sign the cheating text.

Finally, after continually processing, we send out the encrypted PIF and the signed cheating text to the receiver. In the following , we'll show the all encrypting process of the CDES by a flow chart.

Sender :





4. Decrypting Algorithm

The decrypting method of the CDES is very simple. We just reverse the encrypting process' direction and generate CPT again then use the data in the PIF to find out the corresponding characters in the CPT to make the original plaintext.

Input: An encrypted PIF and the corresponding cheating text.
Receiver's private key: To decrypt the IDEA's key and the ID of a cheating text.
Sender's public key: To verify the cheating text if correct.

Output : The original plaintext.

Step 1: Verify each signed cheating text by DSA. Just reserve correct cheating text.

Step 2: Decrypt the cheating text's ID in the head of a given encrypted PIF.

Step 3: Search the corresponding cheating text for the given PIF.

Step 4: If find out the correct cheating text, decrypt the IDEA's key with receiver's private key. If not, wait the correct cheating to come.

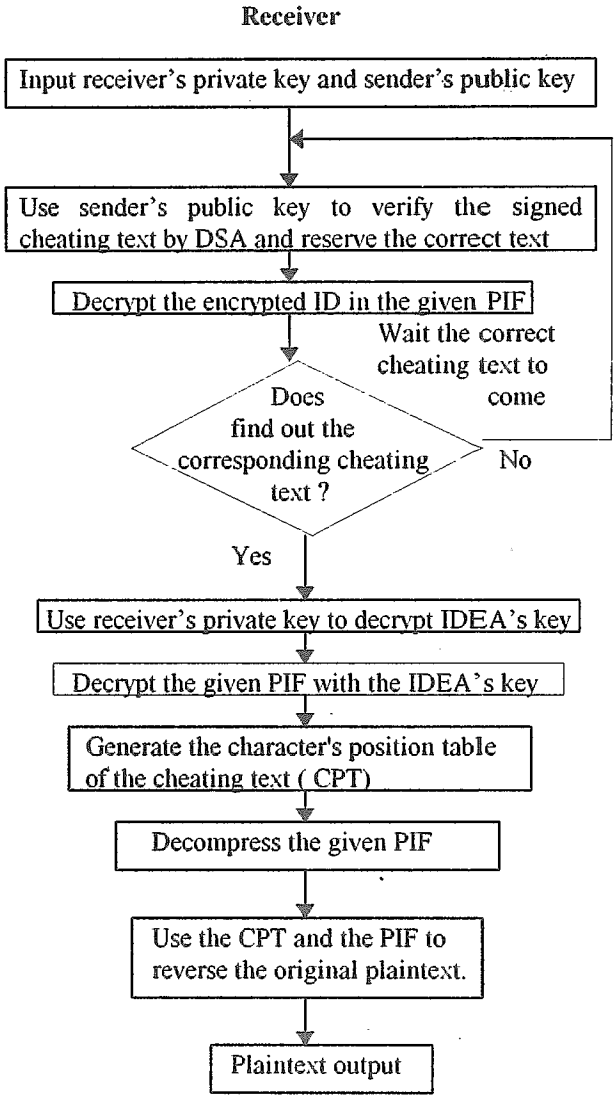
Step 5: According to the IDEA's key, we can decrypt the given encrypted PIF.

Step 6: Use the cheating text to generate the character's position table (CPT).

Step 7: Decompress the given PIF.

Step 8: According to the position record in the PIF, we find out the corresponding characters in the CPT. Finally, we can reverse the original plaintext correctly.

Now, let us describe the decrypting process of the CDES, by the following flow chart.



5. The features and applications of the CDES

There are many advantages in our new cryptosystem. For example, it has good efficiency and multiple protection. Moreover, it owns high extension for other languages. In the following, we list the CDES's features and its applications.

(1) When we want to deliver message to receivers, we needn't send the original message directly. We can send false messages to receivers in the network. By this way, we can send cheating message in purpose to confuse the hackers.

(2) For the same real message, we can use several different false messages to make several different Plaintext Index Files (PIF) and send them to other people. By this way, we can make the hackers misunderstand that these messages are different.

(3) For the PIF, we make it according to the CPT by random so we can have different PIFs from the same plaintext and cheating text.

(4) The method of generating CPT is according to the set of character's kind. In this cryptosystem, we use ASCII code to encode texts which can be represented by ASCII code regardless the country or language they are used.

(5) The delivering function of this cryptosystem uses E-mail system so it is suitable for any platform with E-mail system.

(6) The PIF is an independent file after encoding, so we can chose the most suitable encrypting method for us to encrypt this file.

(7) We not only use the cheating text in the network but also can use the cheating text to store important data in a database. And we can store the corresponding PIF in another database. If a hacker intrudes the database system of the cheating texts, he just thinks that these data are important.

(8) We use hashing function to make CPT , so it has good efficiency to finish encoding.

(9) In this paper, CDES just can process ASCII code. In other words, it just can process English documents. If we want to process Chinese documents , we just change the hash value according to the BIG-5 code in the part of the CPT. As the same way, if we want to process other language, we just have to change the hash value in the CPT. So, the CDES has a high extension for other languages.

6. Security Analysis

The major security consideration of this cryptosystem is cheating the hackers. In order not to cause hackers to suspect the cheating text, we send cheating text and PIF in separation. Therefore, there is no explicit information in the cheating text can cause hackers to suspect.

If the hacker can exactly know what he gets is not real message, he must get the corresponding PIF to reverse the plaintext. Now, the hacker has to faced two problems:

(1) Which PIF is correct ?

(2) How to decrypt the encrypted PIF ?

In this cryptosystem, we use El-Gamal public key system to protect IDEA's key. No matter which attacking method the hacker use, he must face the difficulty of solving desecrate logarithm and spend a lot of time to decrypt all the PIFs he gets.

In the future, public key systems will be applied in the network widely. We know that the PIF is generated by random so the Confused Document Encrypting Scheme could be a kind of Probabilistic Cryptosystem [5,6,7,8,9]. We assume that the cardinality of characters set used in the plaintext are N. The time frequencies of characters appear in the cheating text are $M_1, M_2, ..., M_N$, respectively and the time frequencies of characters appear in the plaintext are $P_1, P_2, ..., P_N$, respectively. The probability of the repeated PIF is $(1 / (M_1)^{P1} * (M_2)^{P2} * ... * (M_N)^{Pn})$.

7. Conclusion

Taking meaningful text as a part of ciphertext is seldom proposed in the past, but we think our idea has great potential in the future. Because the CDES is not only has cheating function but it has multiple protection and good efficiency. We hope in the near feature there'll be more similar idea could be proposed, especially in how to send documents by means of meaningful ciphertext. Finally, we have also developed a software package to implement the proposed idea for sending secure documents.

Reference

[1] X. Lai and J. Massey, " A proposal for a New Block Encryption Standard, " in Proceeding of

EUROCRYPT '90 (Spring-Verlag, Berlin, 1991), pp.389-404.

[2] R. M. Davis, " The Data Encryption Standard in Perspective, " Computer Security and the Data Encryption Standard, National Bureau of Standards Specal Publication Feb. 1978.

[3] NBS FIPS PUB 46, "Data Encryption Standard", National Bureau of Standards , U.S. Department of Commerce, Jan. 1977.

[4] Chi Sung Laih, Lein Harn, and Chin Chen Chang, Contemporary Cryptography and Its Applications , Unalis Corporation, 1995.

[5] S. Goldwasser and S. Micali, "Probabilistic Encryption," Journal of Computer and System Sciences, Vol.28, No.2, pp. 270-299, Apr. 1984.

[6] J. He and K. Lu, "A New Probabilistic Encryption Scheme," Proc. Eurocrypto'88, pp.415-418.

[7] L. Hard and T. Kiesler, " An Efficient Probabilistic Encryption Scheme, " Information Processing Letter, Vol 34, pp.123-129, Apr.1990.

[8] L. Hard and T. Kiesler, "2-bit, Chained, Probabilistic Encryption Scheme," Electrics Letter, Vol.25, No.21, pp.1432-1433, Oct. 1989.

[9] M. O. Rabin, "Digital Signatures and Public-Key Functions as Intractable as Factorization, "MIT laboratory of Computer Science Technical Report, MIT/LCS/TR-212, Jan 1979.

[10] A. Curiger and B. Stuber, "Specification for the IDEA Chip," Technical Report No. 92/03, ETH Zurich: Institute for Integrierte Systeme, 1992.

[11] "The Digital Signature Standard Proposed by NIST," Commun. ACM, Vol.35, No.7, pp.36-40, July 1992.