Algorithm of Cipher Text Containing Key to Produce Cipher Text Transmitted in Network Security

HOMER WU, CHONG-YEN LEE, WUU-YEE CHEN, TSANG-YEAN LEE
Department of Information Management, Chinese Culture University
55, Hwa-Kung Road, Yang-Ming-San, Taipei (11114), TAIWAN

Abstract: - A plaintext is separated into two parts, a fixed length part and a variable length part. The second part of the plaintext is encrypted to a cipher text using a key. In this paper, an encryption algorithm which encrypts the first part of the plaintext and the key of the second part to a cipher text is proposed. Basic computing operations, such as inserting dummy symbols, rotating, transposition, shifting, and complementation, are applied to encrypt plaintext to cipher text. The key used to encrypt the second part of cipher text can be retrieved by decryption of the first part of cipher text. It is secure for the cipher text transmitted through the network since the tables of cipher text are produced randomly and it is difficult to carry out cryptanalysis.

Key-Words: - Data transmission, Cipher text, Plaintext, Encryption, Decryption

1 Introduction

In general, functions of a security system are security, authenticity, integrity, non-repudiation, data confidentiality, and accessed control [1-3, 18-19]. Diffie and Hellman [4] invented the concept of pubic key. Rivest et al [15] advanced a public cryptosystem. In 1960, IBM started the planning of computer cryptology research, and in 1974, IBM proposed an algorithm to review. In 1977, NBS (National Bureau of Standards, U.S.A) [11-12] suggested this algorithm as data encryption standard (DES). McEliece [8] used algebraic coding theory to propose public key. Merkle [9] presented “One way hash function” and used for digital signature. 1988, Miyaguchi [10] developed fast data encipherment algorithm (FEAL-8). NIST (National Institute of Standards and Technology) [13-14] proposed secure hash standard (SHS). Biham and Shamir [1-3] proposed differential attack, Matsui [7] proposed linear cryptanalysis to attack DES type security system.

Lee and Lee [5-6] used the basic operations of computer to design encryption and decryption algorithm. These algorithms used insertion, rotation, transposition, shifting, complementation, and packing computer operations. In this paper, we propose the cipher text contains two parts. The first is a fixed length part and it contains the key to do encryption of the second part which is with variable length. The key is the combination of the location and the value of the format code in cipher text. Different values of the format code make different combination of tables and data. The location of the format code and different combination of format code need to be known before the decryption is proceeded. These processes are more difficult to perform cryptanalysis. Implementations of these algorithms are written in C language and results of processing time are discussed in the paper.

2 Description of Algorithm

The plaintext contains two parts. The first part is a fixed length part which contains plaintext and the encryption key of the second part. The second part could be with variable length. The encryption key, which included in the first part, is the location and the value of format code of the second part and it is used to produce cipher text of the second part. These two parts can then be encrypted together to generate a cipher text to be transmitted. Computer operations including insertion, rotation, transposition, shifting, complementation, and packing are applied to design the encryption algorithm. With the decryption procedure, the first part is decrypted to get the plaintext and the key of the second part when the cipher text is received. The second part of cipher text can then be decrypted using the key to get the original plaintext.

The encryption algorithm is separated into two parts. The following impacts need to be considered in the encryption algorithm of the first part of plaintext and all these impacts should be taken well consideration under the condition that the plaintext is stored in TT (Text Table).

(1) Same cipher texts are created by sending the same data. The problem is that by comparing extensive cipher texts and their original

plaintexts can expose the encryption rules. In order to avoid this drawback, it is important to make sure that different cipher texts are generated with the same plaintext. Setting RB (rotated byte), rotating TT left or right RB times, and then inserting RB into the trailer of TT after rotation to produce TTAR (Text Table after Rotation) can achieve the purpose.

(2) The content of plaintext changes dynamically by setting SLT (Shift Left Table) and shifting left of each byte of TTAR to produce TTAS (Text Table after Shift). This technique makes sure that contents of plaintext to be encrypted are different even with same plaintexts.

(3) Network control codes are inserted into the message when the cipher text is transmitted over the network. The control codes could cause problems when the cipher text is transmitted. In order to solve the problem, we complement the values of control codes of TTAS and set the relative bits of CBIT (Control Bit Table) to 1 else set to 0. CBIT may contain control codes and they are packed to produce CBT (Control Byte Table).

(4) The position of cipher text is changed when sequential order of plaintext is sent. Setting PT (Position Table) and by following this table, the location can be changed to produce cipher text.

(5) The algorithm uses basic computer operations such as shifting, complementing, insertion, and packing to accomplish the job with simple computations.

In the encryption algorithm of the second part of plaintext, additional features under the condition that the plaintext is stored in TT (Text Table) need to be considered:

(1) The content of the cipher text cannot be traced if dummy symbols are inserted to the tail of TT to produce TTWD (Text Table with Dummy).

(2) Different combinations of the second part are necessary. The value is treated as a key to be stored in cipher text to process decryption.

3 Processes of Producing Cipher Text

The encryption algorithms of the first part and the second part are presented in section 3.1 and section 3.2. Relative tables and data used in encryption steps and packed cipher text used in the decryption are also shown in this section. Section 3.3 illustrates transmitted cipher text. Section 3.4 introduces the decryption method.

3.1 First Part Encryption Algorithm

The encryption algorithm presented in Lee and Lee [5-6] is modified to encrypt the first part of the plaintext and has the following steps.

3.1.1 Creating text table (TT)

There are two steps in creating a TT.

(1) Let the length of the plaintext (plaintext, location of format code, value of format code) be N Characters.

(2) Store these characters in TT as T_1, T_2, …,T_N (T_i is the i^{th} character in TT).

3.1.2 Changing contents of plaintext

There are two steps in changing contents of plaintext.

(1) Set rotated byte and rotate TT

(i) Get last 2 characters $T_{N-1}T_N$ from TT.

(ii) Set rotating byte as RB = $T_{N-1}T_N$ mod N.

(iii) Generate rotated text table. If RB is odd then rotate TT to the left RB times; else if RB is even then rotate TT to the right RB times.

(iv) Insert RB to the tail of the above rotated text table to produce text table after rotation (TTAR). TTAR then contains N+1 bytes as $R_1R_2\ldots R_{N+1}$. For example: if RB=3 then we have TTAR as $R_1=T_4$, $R_2=T_5$, $R_3=T_6$, …, $R_{N-4}=T_{N-1}$, $R_{N-3}=T_N$, $R_{N-2}=T_1$, $R_{N-1}=T_2$, $R_N=T_3$, R_{N+1}=RB.

(2) Shift TTAR to get text table after shifting (TTAS)

(i) Set shift left table (SLT) of size N+1 bytes as $F_1F_2\ldots F_{N+1}$. Values of F_i ($1 \le i \le N+1$) are between 0 and 7.

(ii) Shift each byte of TTAR according to the value of relative F_i in SLT.

(iii) Get TTAS as $S_1S_2\ldots S_{N+1}$ (S_i is R_i shifts F_i positions, $1 \le i \le N+1$).

3.1.3 Transmitting through the network

In order to transmit the cipher text through the network correctly, the following steps need to be performed.

(1) Complement TTAS to get text table after complementation (TTAC)

(i) Set a bit k_i for each byte S_i in TTAS to form control bit table (CBIT). There are N+1 bits in CBIT and all bits are set to 0's initially.

(ii) The length of CBIT $L=\lceil(N+1)/8\rceil$ bytes and CBIT is $K_1K_2\ldots K_L$.

(iii) If the value of S_i ($1 \leq i \leq N+1$) in TTAS is below the predetermined threshold value (e.g., 20_{16}), it is complemented to r_i and the relative bit k_i in CBIT is set to 1. The text of TTAS is then complemented to generate TTAC as $r_1r_2\ldots r_{N+1}$.

(2) Pack CBIT to form Control Byte Table (CBT)

(i) Extract every 7 bits (ex. eeeeeee) in CBIT from left-hand side and insert a 1-bit to create a control byte k as eee1eeee.

(ii) The length of CBT $Y=\lceil(N+1)/7\rceil$ bytes and CBT is $c_1c_2\ldots c_Y$.

(3) Combine TTAC and CBT to text table after combination (TAC) as $r_1r_2\ldots r_{N+1}c_1c_2\ldots c_Y$.

3.1.4 Exchanging position

Cipher text (CT) is produced by transposing TAC and the steps are:

(1) Set position table (PT) as $P_1P_2\ldots P_{N+1+Y}$.

(2) Following PT, change the location of TAC to get CT as $C_1C_2\ldots C_{N+1+Y}$.

3.2 Second Part Encryption Algorithm

The encryption algorithm of the second part of the plaintext has 5 steps. They are (1) Create TT; (2) Data uncertainty; (3) Change contents of plaintext; (4) Network transmission; and (5) Position exchange. Except for (2) Data uncertainty, the others are similar to encryption steps in section 3.1. Relative tables and data of the second part encryption algorithm of the plaintext are also introduced.

3.2.1 Data uncertainty in encryption step

In step (2), Dummy symbol table is inserted TT to generate text table with dummy (TTWD) as follows.

(1) Get any M dummy characters $D_1D_2\ldots D_M$.

(2) Append these dummy characters to TT to generate TTWD as $T_1T_2\ldots T_ND_1D_2\ldots D_M$.

In steps (3) to (5), these steps are similar to the steps (2) to (4) in the first part encryption algorithm by replacing TT by TTWD and N by N+M to get cipher text (CT) as $C_1C_2\ldots CT_{N+M+1+Y}$.

3.2.2 Tables and data of the algorithm

Tables used in second part encryption algorithm are shown in Table 1.

Total data length of N, M, and Y is 3. The total length of tables and data is 3N+3M+2Y+6.

Table 1. Second Part Encryption Algorithm Tables

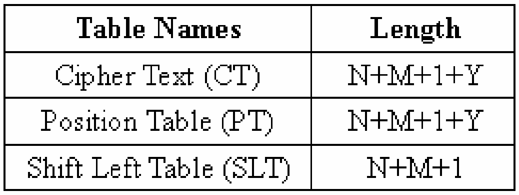


Table Names	Length
Cipher Text (CT)	N+M+1+Y
Position Table (PT)	N+M+1+Y
Shift Left Table (SLT)	N+M+1

3.2.3 Second part cipher text contents

Fields in the cipher text content are as follows:

(1) Format code (FC) in the fixed field
The value of FC is in different combination of pointer fields.

(2) Pointers
(i) PCT: pointer of CT
(ii) PPT: pointer of PT
(iii) PSLT: pointer of SLT
(iv) PV: pointer of value of N, M, Y

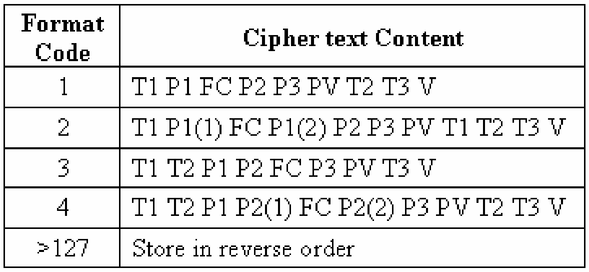
(3) Tables
(i) Cipher Text (CT)
(ii) Position table (PT)
(iii) Shift left table (SLT)

(4) Data
(i) Values of N, M, Y

3.2.4 Second Part Cipher Text

The cipher text is in different formats depending on the format code. The format code is in a fixed location of the cipher text. These two values, the value of format code and the value of the location of the format code, are resided in the first part of plaintext. Fields of pointers are both before and after the location of format code. The length of each table is the difference of corresponding two pointers. The format code can define the different combinations of pointers. A table may be separated into two parts, one before the format code and the other after the format code. Suppose there are three tables T1, T2, T3 to represent CT, PT and SLT and three pointers P1, P2, P3 to represent PCT, PPT and PSLT and one pointer (PV) of value V (like the value of N, M, Y). We can define some values of format code and cipher text as Table 2.

Table 2. Cipher Text Content



Format Code	Cipher text Content
1	T1 P1 FC P2 P3 PV T2 T3 V
2	T1 P1(1) FC P1(2) P2 P3 PV T1 T2 T3 V
3	T1 T2 P1 P2 FC P3 PV T3 V
4	T1 T2 P1 P2(1) FC P2(2) P3 PV T2 T3 V
>127	Store in reverse order

T1, T2, T3 and V may represent different combination of CT, PT, SLT and values of N, M, Y. The values of pointers may increase by some value to avoid the value 1. For example: The format code equals to 1. Suppose T1=CT, T2=PT, T3=SLT, P1=PCT, P2=PPT, and P3=PSLT, then the cipher text is as (CT) PCT FC PSLT PPT PV (SLT) (PT) V.

3.3 Transmitted Cipher Text

By combining the first part cipher text and the second part cipher text produces the transmitted cipher text.

3.4 Decryption algorithm

Decryption algorithm is the reverse of encryption. SLT, PT, N, and Y information is stored in the first part of cipher text. The decryption algorithm of the first part is as follows:

(1) Get the first part of cipher text
(2) Position exchange: Using transposition operation.
(3) Network transmission: Using packing and complementing operations.
(4) Restore contents: Using shifting and rotating operations to retrieve plaintext, location of format code, and value of format code.

Decryption of the second part of cipher text is as follows:

(1) The value of the format code is known from the location of format code.
(2) From the value of the format code, the format of cipher text and pointers of PCT, PPT, PSLT and PV are identified.
(3) From pointers, values of CT, PT, SLT and values of N, M, Y are known.
(4) The decryption process is then proceeded from above tables and data.

4 Implementation

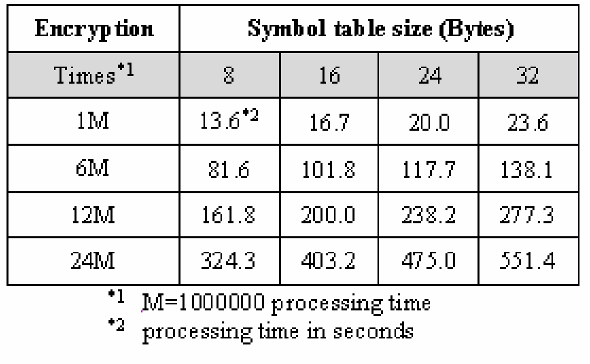
The implementation and experiments of the second part encryption algorithm are performed using INTEL, Pentium D830. The processing times of encryption and decryption with difference combinations of symbol sizes and executing times are shown in Table 3 and Table 4.

5 Conclusion and Discussion

In this study, encryption algorithms of two parts are developed with basic computing operations. Main feature of the algorithm is it is difficult to carry out cryptanalysis and it can save computational time as well. Some comments should be noted as follows.

(1) Information of SLT, PT, N, and Y in the first part encryption algorithm must be known in order to perform decryption process of the first part. In spite of the above information, decryption process of the second needs additional information in the second part encryption algorithm such as:
   (i) location of the format code in cipher text.
   (ii) different cipher text content of the format code.
   (iii) pointers and values of variation to avoid being known.
(2) The second part cipher text may have different length and format since it has different format code, the length of dummy table, and fields of pointers.
(3) The proposed algorithm in this study is more difficult to perform cryptanalysis, because the following fields of each transaction have different values in the cipher text.
   (i) format code
   (ii) cipher text
   (iii) shift left table
   (iv) position table
   (v) location of format code

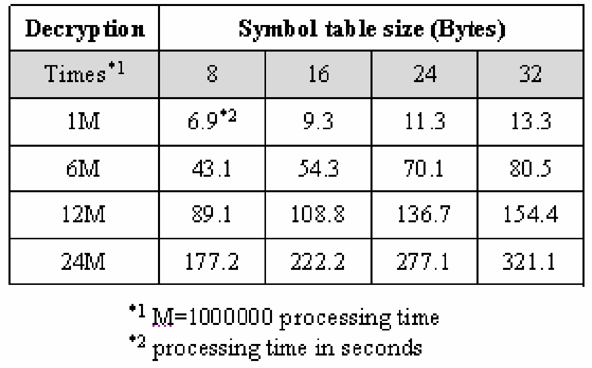
Table 3. Second Part Encryption Processing Times



Encryption	Symbol table size (Bytes)			
Times[*1]	8	16	24	32
1M	13.6[*2]	16.7	20.0	23.6
6M	81.6	101.8	117.7	138.1
12M	161.8	200.0	238.2	277.3
24M	324.3	403.2	475.0	551.4

[*1] M=1000000 processing time
[*2] processing time in seconds

Table 4. Second Part Decryption Processing Times



Decryption	Symbol table size (Bytes)			
Times[*1]	8	16	24	32
1M	6.9[*2]	9.3	11.3	13.3
6M	43.1	54.3	70.1	80.5
12M	89.1	108.8	136.7	154.4
24M	177.2	222.2	277.1	321.1

[*1] M=1000000 processing time
[*2] processing time in seconds

References:

[1] Biham, E. and Shamir, A, Differential Cryptanalysis of DES-like Cryptosystem, *Advances in Cryptology-CRYPTO '90 Proceedings,* Springer-Verlag, 1991, pp.2-21.

[2] Biham, E. and Shamir, A., *A Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.

[3] Biham, E. and Shamir, A., *Differential Cryptanalysis of Data Encryption Standard*, Springer-Verlag, 1993.

[4] Diffie, W. and Hellman, M. E., New Directions in Cryptography, *IEEE Trans. on Inform. Theory*, 1976, pp. 644-654.

[5] Lee, T.-Y., Lee, H.-M., Wu, H, Data Transmission Encryption and Decryption Algorithm in Network Security, *The 6th WSEAS International Conference on Simulation, Modeling and Optimization*, Lisbon, Portugal, September 2006, pp. 22-24.

[6] Lee, T.-Y. and Lee, H.-M., Encryption and Decryption Algorithm of Data Transmission in Network Security, *WSEAS Transactions on Information Science and Applications*, Vol.3, No.12, 2006, pp. 2557-2562.

[7] Matsui, M., Linear cryptanalysis method for DES cipher, In *Advances in Cryptology (CRYPT"O'90). Lecture Notes in Computer Science*, No. 765, 1994, pp. 386-397.

[8] McEliece, R.J., A Public-Key System Based on Algebraic Coding Theory, *Deep Sace Network Progress Report*, No.44, Jet Propulsion Laboratory, California Institute of Technology, 1978, pp. 114-116.

[9] Merkle, R.C., One Way Hash Function and DES, *Proceedings of. Crypto'89*, Springer-Verlag, 1990, pp.428-446.

[10] Miyaguchi, S., The FEAL-8 Cryptosystem and Call for Attack, *Advances in Cryptology-CRYPTO'89 proceedings*, Springer-Verlag, 1990, pp. 624-627.

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

[12] National Bureau of Standards, NBS FIPS PUB 81, *Data Modes of Operation*, National Bureau of Standards, U. S. A. Department of Commerce, Jan.1980.

[13] National Institute of Standards and Technology (NIST). FIPS PUB 180, *Secure Hash Standard (SHS)*, May 11, 1993.

[14] National Institute of Standards and Technology (NIST). NIST FIPS PUB 185, *Escrowed Encryption Standard*, February 1994.

[15] Rivest, R.L., Shamir, A., and Adleman, L., A Method for Obtaining Digital Signatures and Public–Key Cryptosystems, *Communications of the ACM*, Vol.21, No.2, 1978, pp. 120-126.

[16] Shannon, C. E., Communication Theory of Security Systems, *Bell System Technical Journal*, Vol.28, 1949, pp. 657-715.

[17] Shimizu, A. and Miyaguchi, S., Fast Data Enciphrment Algorithm FEAL, *Advances in Cryptology-EUROCRYPT '87 Proceedings,* Springer-Verlag, 1987, pp. 267-278.

[18] Stallings, W., *Cryptography and Network Security: Principles and Practices*, 3rd Edition, Pearson Education, Inc., 2003.

[19] Stallings, W., *Network Security Essentials Application and Standards*, 2nd Edition, Pearson Education, Inc., 2003.