Cipher Text Containing Data and Key to Be Transmitted in Network Security

HUEY-MING LEE[1], TSANG-YEAN LEE[1], LILY LIN[2], JIN-SHIEH SU[1]
[1]Department of Information Management, Chinese Culture University
55, Hwa-Kung Road, Yang-Ming-San, Taipei (11114), TAIWAN
[2] Department of International Business, China University of Technology

Abstract: - In this paper, we propose an encryption algorithm to encrypt plaintext to cipher text. We apply the basic computing operations, e.g., inserting dummy symbols, rotating, transposition, shifting and complement, in the proposed algorithm to encrypt plaintext to cipher text. The cipher text contains the plaintext, relative data and tables of encryption and transmits through the network. Since the tables of cipher text are produced randomly, it is difficult to do cryptanalysis.

Key-Words: - Data transmission; Cipher text, Plaintext, Network security

1. Introduction

In 1949, Shannon [21] discussed the theory of security system. In general, the functions of security system are security, authenticity, integrity, non-repudiation, data confidentiality and accessed control [1-3, 23-24]. Diffie and Hellman [5] proposed the concept of pubic key. Rivest et al. [20] proposed public cryptosystem. In 1974, IBM proposed an algorithm to review. In 1977, NBS (National Bureau of Standards, U.S.A) [15-16] suggested this proposed algorithm as data encryption standard (DES). McEliece [12] used algebraic coding theory to propose public key. Merkle [13] presented "One way hash function" and used for digital signature. 1988, Miyaguchi [14] developed fast data enciphered algorithm (FEAL-8). NIST (National Institute of Standards and Technology) [17-18] proposed secure hash standard (SHS). Biham and Shamir [1-3] proposed differential attack, Matsui [10] proposed linear cryptanalysis to attack DES type security system. When new encryption is proposed, cryptanalysis starts to develop to attack.

Lee and Lee [9] used the basic operations of computer to design encryption and decryption algorithm. The proposed algorithms used insertion, rotation, transposition, shift, complement and pack of computer operation. In [10], Lee and Lee use these algorithms to do authentication on grid environment.

In this study, we pack the final symbol table, relative data, relative tables and control byte to produce cipher text. In the cipher text, we also have the fields of format code to design the different combination of tables and data. We should know the location of format code and different combination of format code to do decryption. These processes are more difficult to do cryptanalysis. Also, we implement the proposed algorithms with C language and get the result of processing time.

2. The Proposed Algorithm Description

In order to encrypt plaintext to cipher text, we should solve the following items in this algorithm:
(1)Data uncertainty;
(2)Volume of same data to send;
(3)Change contents of plaintext;
(4)Network transmission;
(5)Position exchange;
(6)Simple computation;
(7)Store key in cipher text.
We explain the solving method of each item as follows under the condition that the plaintext is stored in Symbol Table (ST).
(1)Data uncertainty:
We insert dummy symbols to the trailer of ST to produce Symbol Table with Dummy (STWD).
(2)Volume of the same data to send:
We need the cipher text to be different, set Rotated Byte (RB), rotate the STWD left or right RB times, and then insert RB in the trailer of STWD after rotation to produce Symbol Table after Rotation (STAR).

(3)Change contents of plaintext:
The contents of plaintext will be changed. We set Shift Left Table (SLT) and shift left of each byte of STAR to produce Symbol Table after Shift (STAS).
(4)Network transmission:
When cipher text is transmitted, we should avoid network control code. We complement the value of control code of STAS and set the relative bit of Control Bit Table (CBIT) to 1 else to 0. CBIT may contain network control code also and we pack CBIT to produce Control Byte Table (CBT).
(5) Position exchange:
We set Position Table (PT). From this table (PT), we change the location of STAS to produce cipher text.
(6)Simple computation:
We use the computer basic operations, e.g., shift, complement, insert, pack operations.
(7)Store key in cipher text:
We store the pointers of tables and data to cipher text to do decryption.

3. The Processes of Producing Cipher Text

We present the encryption step in 3.1. In 3.2, we list the relative tables and data used in encryption steps and we pack relative tables, data and final symbol table to cipher text to be used in the decryption. In 3.3, we explain the fields of cipher text. We list the possible combinations of cipher text in 3.4. The decryption method is shown in 3.5. The possible combinations of cipher text are very large and difficult to do decryption as shown in 3.6.

3.1 Encryption Step

Based on Lee and Lee [9], we propose the new encryption algorithm in the following steps.
Step 1: Set symbol table (ST)
(1) Let the length of the plaintext be N Characters;
(2) Store them in the symbol table (ST) as Symbol Table (ST): $S_1S_2\ldots S_N$.
Step2: Insert dummy symbol table to symbol table (ST)
(1) Get any M dummy characters;
(2) Append to symbol table (ST)
(3) Get symbol table with dummy (STWD) as $S_1S_2\ldots S_ND_1D_2\ldots D_M$
Step 3: Set rotated byte and rotate symbol table with dummy (STWD)
(1) Get any character DD.
(2) Set rotated byte, RB, as RB = DD mode (N+M).
(3) If RB is odd then we rotate symbol table with dummy (STWD) to left RB times; else if RB is even then we rotate symbol table with dummy (STWD) to right RB times.
(4) Insert RB to the trailer of above symbol table after rotation.
(5) Get symbol table after rotation (STAR).
For example: if RB=4 then we have symbol table after rotation (STAR) as $D_{M-3}D_{M-2}D_{M-1}D_M S_1S_2\ldots S_ND_1D_2\ldots D_{M-4}$ RB
Step 4: Shift the symbol table after rotation (STAR) to get symbol table after shift (STAS)
(1) Set shift left table (SLT) of each byte, the contained value of shift left table is below 8. There are shown as Shift Left Table: (SLT): $F_1F_2\ldots F_{N+M+1}$
(2) Shift each byte of symbol table after rotation (STAR) according to the contained value of shift left table (SLT).
(3) Get symbol table after shift (STAS) as $SS_1SS_2\ldots SS_{N+M+1}$
(4) Add 32 (e.g., 20_{16}) to each byte of shift left table (SLT) and keep these values.
Step 5: Complement the symbol table after rotation (STAR)
(1) Set control bit table (CBIT) to all 0 and the table length is L= [(N+M)/8+1].
(2) If the value of symbol table after rotation (STAR) is below the certain value (e.g., 20_{16}), we complement the symbol of symbol table after rotation (STAR) to get symbol table after complement (STAC) and set the relative bit of control bit table (CBIT) to 1.
(3) The results of these two tables are as follows:
Control Bit Table (CBIT): $C_1C_2\ldots C_L$
Symbol Table After Complement (STAC): $SS_1SC_2\ldots SC_{N+M+1}$
Step 6: Packed Control Byte Table (CBT).
(1) We take each 7 bits (as eeeeeee) of control bit table (CBIT) from left and set control byte as ee1eeeee to form control byte table (CBT),
(2) The length of control byte table is
K = INT ((N+M)/7)+1.
(3) Get control byte table (CBT) as follows:
Control Byte Table (CBT): $(C1B_1)(C1B_2)\ldots(C1B_K)$
Step 7: Combine symbol table after complement (STAC) and control byte table (CBT) to symbol table after combination (SAC).
(1) Combine symbol table after complement (STAC) and control byte table (CBT).
(2) Get symbol table after combination (SAC) as
$SS_1SC_2\ldots SC_{N+M+1}(C1B_1)(C1B_2)\ldots(C1B_K)$
Step 8: Transpose the symbol table after combination (SAC) to get cipher text.
(1) Set the position table (PT) as $P_1P_2\ldots P_{N+M+1+K}$
(2) Following position table (PT), we change the location of the symbol table after combination (SAC) to produce final symbol table (FST).
(3) Final symbol table (FST) is cipher text (CT) as $SP_1SP_2\ldots SP_{N+M+1+K}$
(4) Add 32 to each byte of position table (PT) and keep these values.

3.2 Relative Tables and Data Used in Encryption Algorithm

Following tables are used for encryption algorithm.
Final Symbol Table (FST) length N+M+1+K
Position Table (PT) length N+M+1+K
Shift Left Table (SLT) length N+M+1
Data of N,M, K length 3
Total length of above is 3N+3M+2K+6

3.3 Fields of Cipher Text

The fields in the cipher text are as follows:
(1) FC: format code in the fixed field.
The FC is the different combinations of pointer field.
(2) PFST: pointer of final symbol table (FST)
(3) PPT: pointer of position table (PT)
(4) PSLT: pointer of shift left table (SLT)
(5) PV: pointer of value of N, M, K
(6) Final symbol table (FST)
(7) Position table (PT)
(8) Shift left table (SLT)
(9) The value of N, M, K

3.4 Cipher Text

The cipher text is the different format depending on format code. The format code is in fixed location of cipher text. The field of pointer is before and after the location of format code. The length of each table is the difference of two pointers. The format code can define the different combinations of pointer. One of the tables may be separated to before and after the format code. Suppose we have three tables (T1, T2, T3) to represent FST, PT and SLT and three pointers (P1, P2, P3) to represent PFST, PPT and PSLT and one pointer (PV) to value (V) (e.g., the value of N, M, K). We can define some value of format code and cipher text as shown in Table 1.

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

3.5 Decryption Algorithm.

Decryption algorithm is the reverse of encryption. The steps of decryption are as follows:
(1)First we know the location of the format code;
(2)When we read the location of format code, we get the format of cipher text;
(3)We get the pointer of FST, PT, SLT and value of N, M, K;
(4)From the pointers, we get the values of FST, PT, SLT and value of N, M, K;
(5)We do decryption from above tables and data.

3.6 Combination Possibility

Encryption Step		Times of Combination
(1)Insert dummy symbol	(STWD	256**M
(2)Set rotate byte	(STAR)	N+M
(3)Shift the symbol table	(STAS)	8**(N+M+1)
(4)Complement the STAR	(STAC, CBIT)	2**(N+M+1)
(5)Packed	(CBT)	2**7*(INT((N+M)/7)+1)
(6)Combine STAC and CBT	(SAC)	1
(7)Transpose SAC	(Cipher text)	(M+N+1)!
(8)Format code		3(M+N)+2K+6
(9)Pointers		4!
(10)Value		3!

The total possible combinations are
256**M*(N+M)*8**(N+M+1)*2**(N+M+1)*2**7*(INT((N+M)/7)+1)*1*(M+N+1)!*(3(M+N)+2K+6)*4!*3!
This number is very large and difficult to get the computational formula.

4. Implementation

In this section, we implement the proposed algorithms. The computing environment is shown in 4.1. The processing time of encryption and decryption are shown in 4.2. In 4.3, we present the discussion of implementation.

4.1 Computing Environment

Computer type: INTEL, Pentium D830
Memory size: DDR 512 MB * 2
Computer Language: C Language

4.2 Executing Results

The processing time of the different combinations of symbol size and executing times are as follows: Table 2 is the encryption processing time. We also get the decryption processing time in Table 3.

4.3 Discussion of Implementation

(1) As the size of symbol table increases, the processing time linearly increases.
(2) As the number of executing times increases, the processing time linearly increases.

Table 2. Encryption processing time

Encryption	Symbol table size (Bytes)			
Times [1)]	8	16	24	32
1M	13.6 [2)]	16.9	19.9	23.3
4M	55.0	66.5	80.0	94.5
8M	108.3	133.8	159.8	189.7
16M	217.0	267.9	320.4	376.1

[1)] M=1000000 processing times.
[2)] processing time in second

Table 3. Decryption processing time

Decryption	Symbol table size (Bytes)			
Times [1)]	8	16	24	32
1M	7.6 [2)]	9.5	11.7	13.0
4M	30.3	37.4	44.5	53.8
8M	57.4	74.5	89.4	106.5
16M	115.0	151.3	181.1	215.1

[1)] M=1000000 processing times.
[2)] processing time in second

5. Conclusion and Discussion

In this study, we use the basic computing operations to design these encryption algorithms. It doesn't need any special hardware. Finally, we make some comments about this study.
(1) To do the decryption, we must know;
(a) the location of format code in cipher text;
(b) the different cipher text content of format code;
(c) the pointers and values of variation to avoid being known.
(2) Each cipher text may have different length and format because it has different format code, the length of dummy
Table 2: Average-users-waiting-time (sec)
table and field of pointers. The length of cipher text is about three times of plaintext, dummy symbols and control bytes, plus format code, values and pointers;
(3) The proposed algorithm in this study is more difficult to cryptanalysis, because the following fields of each transaction have different value in the cipher text;
(a) format code;
(b) final symbol table;
(c) shift left table;
(d) position table.
(4) We can set any length of dummy symbol table;
(5) We can have many pointers for one table;
(6) By the algorithms described in Section 3, we can set up the encryption and decryption mechanism by computers as a useful and security procedures.

Acknowledgement

This work was supported in part by the National Science Council, Republic of China, under grant NSC-95-2745-M-034-007-URD.

References:

[1] Biham, E. and Shamir,A.: "Differential Cryptanalysis of DES-like Cryptosystem", *Advances in Cryptology-CRYPTO '90 Proceedings*, Springer-Verlag Berlin, 1991, pp. 2-21.
[2] Biham, E. and Shamir, A.: "*A Differential Cryptanalysis of the Data Encryption Standard*", Springer Berlin Heidelberg New York, 1993.
[3] Biham, E. and Shamir, A.: "*Differential Cryptanalysis of Data Encryption Standard*", Springer-Verlag Berlin, 1993.
[4] Denning, D.: *Cryptography and Data Security*, Addison-Wesley, 1982.
[5] Diffie, W. and Hellman, M. E.: "New Directions in Cryptography", *IEEE Trans. on Inform.* Theory, 1976, pp. 644-654.
[6] Gilbert, H. and Chase, G.: "A Statistical Attack on the FEAL-8 Cryptosystem" , *Advances in Cryptology-CRYPTO '90 proceedings*, Springer-Verlag Berlin, 1991, pp. 22-2.
[7] Goldreich, O.: "*Foundations of Cryptography: Basic Tools*", Published by the Press Syndicate of The University of Cambridge, The Pitt Building, Trumpington Street, Cambridge, United Kingdom, 2001.
[8] Hardy, D., Carol, W. Walker, L.: "*Applied Algebra: Codes, Ciphers, and Discrete Algorithms*", Library of Congress cataloging-in Publication Data,2003 by Pearson Education, Inc. Pearson Education, Inc. Upper Saddle River, NJ 07458.
[9] Lee, T.-Y., Lee, H.-M.: "Encryption and Decryption Algorithm of Data Transmission in Network Security", *WSEAS Transactions on Information Science and Applications*, Issue 12, Volume 3, 2006, pp. 2557-2562.
[10] Lee, H.-M., Lee, T.-Y., Chen, C.-s., Su, J.-S. "Authentication Algorithm Based on Grid

Environment", Proceeding of the 6th WSEAS International Conference Applied Computer Science (ACOS'07), April 15-17, 2007, pp. 235-239.

[11] Matsui, M.: "Linear cryptanalysis method for DES cipher" In T. Helleseth, Editor, *Advances in Cryptology (CRYPT"O'90). Lecture Notes in Computer Science* No. 765, 1994, pp. 386-397, Springer-Verlag Berlin Heidelberg New York.

[12] McEliece, R.J.: "A Public-Key System Based on Algebraic Coding Theory". Deep Sace Network Progress Report, 44, Jet Propulsion Laboratory, *California Institute of Technology*, 1978, pp. 114-116.

[13] Merkle, R.C.: "*One Way Hash Function and DES*", Proc. Crypto'89,: Springer-Verlag Berlin, pp. 428-446, 1990.

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

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

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

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

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

[19] Pieprzyk, J., Hardjono, T., Seberry, J.: "*Fundamentals of Computer Security*", Springer-Verlag Berlin Heidelberg, 2003.

[20] 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.

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

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

[23] Stallings, W.: "*Cryptography and Network Security: Principles and Practices*", International Edition, Third Edition 2003 by Pearson Education, Inc. Upper Saddle River, NJ 07458.

[24] Stallings, W.: "*Network Security Essentials Application and Standards*", Second Edition 2003 by Pearson Education, Inc. Upper Saddle River, NJ 0745..