Cipher Text to Be Transmitted and Cryptanalysis in Network Security

Tsang-Yean Lee, Huey-Ming Lee, and Nai-Wen Kuo

Abstract—In this paper, we propose an encryption algorithm to encrypt plaintext to cipher text. We divide plaintext into numeric and non-numeric fields, also, we pack the numeric fields to produce packed numeric table and combine it with symbol fields to produce the new plaintext. 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 produced the cipher text which contains the plaintext, relative data and tables of encryption is transmitted to the receiver through the network. We also propose the cryptanalysis about these algorithms. It can be shown that the proposed algorithm is more secure in network security.

Keywords—Data transmission; Cipher text, Plaintext, Network security

I. INTRODUCTION

In 1949, Shannon [22] 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, 24-25]. Diffie and Hellman [5] proposed the concept of pubic key. Rivest et al. [21] proposed public cryptosystem. In 1974, IBM proposed an algorithm to review. In 1977, NBS (National Bureau of Standards, U.S.A) [16-17] suggested this proposed algorithm as data encryption standard. McEliece [13] used algebraic coding theory to propose public key. Merkle [14] presented "One way hash function" and used for digital signature. 1988, Miyaguchi [15] developed fast data enciphered algorithm (FEAL-8). NIST (National Institute of Standards and Technology) [18-19] proposed secure hash standard. Biham and Shamir [1-3] proposed differential attack, Matsui [12] proposed linear cryptanalysis to attack DES type security system. When the new encryption is proposed, cryptanalysis starts to be developed to attack.

Lee and Lee [10] used the basic computer's operations, e.g., insertion, rotation, transposition, shift, complement and pack, to design encryption and decryption algorithm. Lee and Lee [11] used these algorithms to do authentication on grid environment.

Lee and Lee [9] propose cipher text containing data and key to be transmitted in network security. In this study, we extend the method to separate the plaintext to numeric and non-numeric fields and use the encryption algorithm to produce cipher text. The cipher text contains the final symbol table, relative data, relative tables, control byte and the field of format code to design the different combination of tables and data. In order to decrypt, we should know the location of format code and different combination of format code. We also propose the cryptanalysis about these algorithms. We can show that the proposed processes are more difficult to do cryptanalysis.

Manuscript received May 13, 2007, Revised November 24, 2007
This work was supported in part by the National Science Council, Republic of China, under grant NSC 96-2745-M-034-002-URD.
T.-Y. Lee is with Department of Information Management, Chinese Culture University, TAIWAN (e-mail: tylee@faculty.pccu.edu.tw).
H.-M. Lee is with Department of Information Management, Chinese Culture University, TAIWAN (corresponding author, phone: +886-937-893-845; fax: +886-2-2777-4723; e-mail: hmlee@faculty.pccu.edu.tw).
N.-W. Kuo is with Department of Information Management, Chinese Culture University, TAIWAN (e-mail: neven}@faculty.pccu.edu.tw).

II. THE PROPOSED ALGORITHM DESCRIPTION

In order to encrypt the plaintext to cipher text, we should solve the following items in the proposed algorithm.
(1) Separating numeric and non-numeric fields;
(2) Burst force by volume of same data to be sent;
(3) Data uncertainty;
(4) Position exchange;
(5) Change contents of plaintext;
(6) Network transmission;
(7) Simple computation;
(8) Store key in cipher text.
We explain the solving method of each item as follows under the condition that the plaintext is stored in the plaintext table (PT).

(1) Separating numeric and non-numeric fields.
(a) We separate the plaintext to numeric and non-numeric table. We store the original position to numeric and non-numeric position table.
(b) We pack numeric table to produce the packed numeric table. This will be about half size of the original numeric fields.

(2) Burst force by volume of the same data to be sent:
We need the cipher text to be different with the same plaintext to be sent.
(a) We set different rotated byte (RB), rotate above four tables left or right RB times, and then insert RB in the trail of each table.
(b) After combination of these four tables, we set rotated byte (RB), rotate left or right RB times, and then insert RB in the trail of table.
(c) We may have different combinations of these tables.

(3) Data uncertainty:

We get different length of dummy symbols and insert these dummy symbols to the trail of table.

(4) Position exchange:
We set position table. From this table, we change the location of above table to produce cipher text.

(5) Change contents of plaintext:
The contents of plaintext will be changed. We set shift left table (SLT) and shift left of each byte of above table.

(6) Network transmission:
When cipher text is transmitted, we should avoid network control code. We create control block table. If the value of table is the communication control code, we complement the value of table and set the relative bit in control block table to 1 else to 0.

(7) Simple computation:
We use the computer basic operations, e.g., shift, complement, insert, pack operations.

(8) Store key in cipher text:
We store the pointers of tables and tables to produce cipher text.

III. THE PROCESSES OF PRODUCING CIPHER TEXT

We present the encryption step in Section A. In Section B, we list the relative tables and data used in encryption steps and we pack relative tables, data and final symbol table to produce the cipher text. We explain the fields of cipher text in Section C. We list the possible combinations of cipher text in Section D. The decryption method is shown in Section E. The possible combinations of cipher text are very large and difficult to do decryption as shown in Section F.

A. Encryption Step

Based on Lee and Lee [9], we propose the encryption algorithm as the following steps.

Step 1: Set plaintext table (PT)

(1) Let the length of the plaintext be LP characters;

(2) If each byte of plaintext is numeric, we store it to numeric table (NT) and posit position to numeric position table (NPT) else store it to the symbol table (ST) and symbol position table (SPT). NL is the length of numeric table and SL is the length of symbol table. The length of plaintext (LP) is LP=NL+SL.

(3) We pack numeric table as packed numeric table (PNT).

(4) The tables are as the following:

Packed Numeric Table (PNT) $N_1N_2\ldots N_{PL}$
PL is the length of packed numeric table.
PL=INT((NL+1)/2), LP=NL+SL.

Symbol Table (ST) $S_1S_2\ldots S_{SL}$
SL is the length of symbol table.

Numeric Position Table (NPT) $NP_1NP_2\ldots NP_{NL}$
NL is the length of numeric position table.

Symbol Position Table (SPT) $SP_1SP_2\ldots SP_{SL}$
SL is the length of symbol position table.

Step2: Burst force by volume of the same data to be sent

(1) We set different rotated byte (RB), rotate above four tables left or right RB times depending on sign of RB. We insert RB to the trail of each table.

(2) The tables are as the following:

Rotate Packed Numeric Table (RPNT)
$RN_1RN_2\ldots RN_{PL}RB$.

Rotate Symbol Table (RST)
$RS_1RS_2\ldots RS_{SL}RB$.

Rotate Numeric Position Table (RNPT)
$RNP_1RNP_2\ldots RNP_{NL}RB$.

Rotate Symbol Position Table (RSPT)
$RSP_1RSP_2\ldots RSP_{SL}RB$.

(3) We get the combination byte (CB). The combination byte determinates the order of these four tables. Following the combination byte, we combine above four tables and combination byte as text table (TT). The text table is as following:

Text Table (TT): $T_1T_2\ldots T_{L-1}CB$ where L= PL+ SL+NL+SL+5

(4) We set rotated byte (RB), rotate text table (TT) left or right RB times, and then insert RB in the trail of text table to produce rotate text table (RTT). The table is as follows:

Rotate Text Table (RTT) $RT_1RT_2\ldots RT_LRB$ The length of RTT is L+1

Step 3: Data uncertainty:

(1) Get any M dummy characters;

(2) Append to rotate text table (RTT);

(3). Get text table with dummy (TTWD) as $RT_1RT_2\ldots RT_LRT_{L+1}\ D_1D_2\ \ldots D_M$

Step 4: Position exchange:

(1) Set the position table (PT) as $P_1P_2\ldots P_{L+1+M}$

(2) Following position table (PT), we change the location of the text table with dummy (TTWD) to produce text table after position (TTAP).

Step 5: Change contents of plaintext:

(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): $F_1F_2\ldots F_{L+1+M}$

(2) Shift each byte of text table after position (TTAP) according to the contained value of shift left table (SLT).

(3) Get text table after shift (TTAS) as $SS_1SS_2\ \ldots SS_{L+1+M}$

(4) Add 32 (e.g., 20_{16}) to each byte of shift left table (SLT) to avoid network transmission code and keep these values.

Step 6: Network transmission:

(1) Set control bit table (CBIT) to all 0 and the table length is LC= [(L+M)/8+1].

(2) Avoid network control code. If the value of text table after shift (TTAS) is below the certain value (e.g., 20_{16}), we complement the symbol of text table after shift (TTAS) to get text table after complement (TTAC) 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): $C_1C_2\ \ldots C_{LC}$

Text Table After Complement (TTAC):
$TC_1\ TC_2\ldots TC_{L+1+M}$

(4) We take each 7 bits (as *eeeeeee*) of control bit table from left and set control byte as *ee1eeeee* to form control byte table (CBT). The 1 in ee1eeeee is to avoid network control code. The length of control byte table is K = INT ((L+M)/7)+1.

(5) Get control byte table (CBT) as follows:
Control Byte Table (CBT): $(C1B_1)(C1B_2)...(C1B_K)$

Step 7: Combine text table after complement (TTAC) and control byte table (CBT) to text table after combination (TTAC).

(1) Combine text table after complement (TTAC) and control byte table (CBT) to produce final text table (FTT), which is
$TC_1\ TC_2 \ldots TC_{L+1+M}\ (C1B_1)(C1B_2)...(C1B_K)$
The length of FTT is L+1+M+K.

B. Relative Tables and Data Used in Encryption Algorithm

Table 1 are used for encryption algorithm.

Table 1. Tables and length

Tables and Value	Length
Final Text Table	L+1+M+K
Position Table	L+1+M
Shift Left Table	L+1+M
L, M, K, PL, NL, SL, CB	7

Total length is 3L+3M+K+10.

C. 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) Final text table (FTT)
(3) Position table (PT)
(4) Shift left table (SLT)
(5) VD the value of L, M, K, NPL, NL, SL, CB
(6) PFTT: pointer of final text table (FTT)
(7) PPT: pointer of position table (PT)
(8) PSLT: pointer of shift left table (SLT)
(9) PV: pointer of value of L, M, K, PL, NL, SL, CB

D. 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 the front and rear of the location of format code. The length of each table is the difference of two pointers. The format code can be defined to be the different combinations of pointer. One of the tables may be separated to front and rear of the format code. We can define some value of format code and cipher text as shown in Table 2.

Table 2. Cipher Text Content

Format Code	Cipher text Content
1	FTT PFTT FC PPT PSLT PV PT SLT VD
2	FTT PFTT(1) FC PFTT(2) PPT PSLT PV PFTT PT SLT VD
3	FTT PT PFTT PPT FC PSLT PV SLT VD
4	FTT PT PFTT PPT (1) FC PPT(2) PSLT PV PT SLT VD
>127	Store in reverse order

E. 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) From the contents of format code, we get the format of cipher text;
(3) We get the pointer of FTT, PT, SLT and VD;
(4) From the pointers, we get the values of FTT, PT, SLT and value of L, M, K, PL, NL, SL, CB. We can get LC=int((L+M)/7) and LP=NL+SL.
(5) Separate final text table (FTT) to text table after complement (TTAC) and control byte table (CBT);
(6) From control byte table (CBT), we get control bit table (CBIT);
(7) From control bit table (CBIT) and text table after complement (TTAC), we get text table after shift (TTAS);
(8) From shift left table (SLT) (subtract 32 from each value) and text table after shift (TTAS), we get text table with position (TTWP);
(9) From position table (PT) and text table after position (TTAP), we get text table with dummy (TTWD);
(10) From text table with dummy (TTWD) and value of M, we get rotated text table (RTT);
(11) From the trail byte (rotated byte), we rotate rotated text table (RTT) to get text table (TT);
(12) From text table (TT), combination byte (CB) and value of PL, SN, NL, SN, we get rotated numeric table (RNT), rotated symbol table (RST), rotated numeric position table (RNPT) and rotated symbol position table (RSPT);
(13) From each trial byte (rotate byte), we rotate it to get packed numeric table (PNT), symbol table (ST), numeric position table (NPT) and symbol position table (SPT).;
(14) From packed numeric table (PNT), we unpack to get numeric table (NT);
(15) From numeric table, symbol table (ST), numeric position table (NPT) and symbol position table (SPT), we get plaintext.

F. Combination Possibility

In each encryption step and tables, we have following combinations as table 3.

From table 3, the total possible combinations are
2*PL*SL*NL*SL*24*(2*(PL+2SL+NL)*256**M*(L+1+M)!*8**(L+M+1)*2**(L+1+M)*2**7*(INT((L+M)/7)+1)*1*3(M+N+1)!*(3(L+M)+K+10)* 4!*7!
This number is very large and difficult to get the computational formula.

IV. CRYPTANALYSIS

A. Announce

We announce the encryption steps of this algorithm as follows:

(1). Separation numeric and symbol tables;
(2). Rotate tables;
(3). Insert dummy symbols;
(4). Transpose tables;
(5). Shift tables;
(6). Complement tables;
(7). Produce cipher text.

B. Knowing data

After encryption, we can know only the following data:

(1). Plaintext;
(2). The length of plaintext;
(3). Cipher text;
(4). The length of cipher text;
(5). The length of numeric field and location;
(6). The length of non-numeric field and location;
(7). The encryption steps.

Table 3. Combinations

Encryption Step		Times of Combination
(1)Set symbol tables	(ST)	1
(2)Rotate	(RTT)	2*PL*SL*NL*SL*24*(2*(PL+2SL+NL)
(3)Insert dummy symbol	(TTWD)	256**M
(4)Position exchange	(TTWP)	(L+1+M)!
(5)Shift the symbol table	(TTAS)	8**(L+1+M)
(6)Complement the TTAS	(TTAC, CBIT)	2**(L+1+M)
(7)Packed	(CBT)	2**7*(INT((L+M)/7)+1)
(8)Combine STAC and CBT	(SAC)	1
(9)Format code		3(L+M)+K+10
(10)Pointers		4!
(11)Value		7!

C. Keep data in secure

We must keep the following data in secure.

(1) Location of format code;
(2) The combination of format code;
(3) The increment value of each symbol;
(4) The combinations of four tables.

D. Cryptanalysis description

The cryptanalysis of this encryption algorithm may be as the follows:

(1) Compute length of tables and dummy symbol;
(2) We use brute-force to test. We can use the same plaintext to do encryption and get the different cipher text. We use these cipher text to analyze. The steps are as following:
(a) Analyze cipher text to find the location of format code in the cipher text;
(b) From the value of format code find the format of cipher text.
(c) Locate the pointer of each table;
(d) Locate each table;
(e) Reverse the encryption algorithm to find tables and data of following order;.
Final Text Table (FTT);
Shift Left Table (SLt);
Position Table (PT);
Value of L, M, K, NPL, SN, NL, SL, CB;
Convert control block table to control bit table;
Complement text table
Text Table After Shift (TTAS);
Text Table After Position (TTAP);
Text Table With Dummy (TTWD);
Rotate Text Table (RTT);
Numeric Table, Numeric Position Table, Symbol Table, Symbol Position Table.
Plaintext
(3) From the combination possibility of Section 3.6, we know it is difficult to cryptanalysis.

E. The protective method

The protective methods of above are as the follows:

(1) We Keep the document of encryption and decryption algorithm in secure;
(2) We keep the encryption and decryption source or object program in secure;
(a) We separate the subprograms into different files and set different password, and protect to be stolen whole programs;
(b) We link together before execution and download to site in the fixed period.
(3) We download the binary code by changing the location of format code to every site in the fixed period.

V. IMPLEMENTATION

In this section, we implement the proposed algorithms. The computing environment is shown in Section A. The processing time of encryption and decryption are shown in Section B. In Section C, we present the discussion of implementation.

A. Computing Environment

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

B. Executing Results

The processing time of the different combinations of symbol size and executing times are as follows: Table 4 is the encryption processing time. We also get the decryption processing time as shown in Table 5.

Table 4. Encryption processing time

Encryption	Symbol table size (Bytes)		
Times [1)]	8	16	32
1M	30.66 [2)]	37.67	52.81
4M	123.97	152.34	210.51
8M	247.52	305.56	423.36
16M	498.81	612.84	844.63

[1)] M=1000000 processing times.

[2)] processing time in second

Table 5. Decryption processing time

Decryption	Symbol table size (Bytes)		
Times [1)]	8	16	32
1M	20.50 [2)]	27.00	42.75
4M	86.59	108.23	176.74
8M	162.63	221.75	357.64
16M	330.97	472.42	699.73

[1)] M=1000000 processing times.

[2)] processing time in second

C. 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.

VI. 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) We separate plaintext to numeric and non-numeric fields and pack numeric fields. Through rotation, there will have different sequence of cipher text;
(2) We store of tables in cipher text and use to decrypt. These tables are set randomly and have different combinations
(3) Each cipher text may have different length and format, because it has different format code, the length of dummy table and field of pointers;
(4) We can set any length of dummy symbol table;
(5) We can have many pointers for one table;
(6) In order 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;
(7) Compare to Lee and Lee [9], the processing time is about 2.3 times. The length of cipher text is about 2 times. The proposed method in this study is more secure.

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, H.-M., Lee, T.-Y., Lin Lily, Su Jin-Shieh.: Cipher Text Containing Data and Key to Be Transmitted in Network Security, Proceeding of the 11th WSEAS International Multiconference CSCC (CIRCUITS, SYSTEMS, COMMUNICATIONS, COMPUTERS), Agios Nikolaos, Crete Island, Greece, July 23-28, 2007, pp.275-279.

[10] 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.

[11] Lee, T.-Y., Lee, H.-M., Authentication Algorithm Based on Grid Environment, Proceeding of the 6th WSEAS International Conference Applied Computer Science (ACOS'07), Hangzhou, China, April 15-17, 2007, pp. 235-239.

[12] 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.

[13] 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.

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

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

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

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

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

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

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

[21] 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.

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

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

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

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

Tsang-Yean Lee received his Master degree in electrical engineering from National Taiwan University at Taipei, Taiwan in 1969. He is currently an associate professor at the Department of Information Management at Chinese Culture University of Taiwan. His research interests are operating system, information security, and grid computing.

Huey-Ming Lee is a professor in the Department of Information Management at the Chinese Culture University. He got his Ph.D. from the School of Computer Science and Engineering at the University of New South Wales in Australia. His research interests are in the field of fuzzy sets theory and its applications, operation research, grid computing, software engineering, and information systems. His papers appeared in European Journal of Operational Research, Fuzzy Sets and Systems, Information Sciences, International Journal of Innovative Computing, Information & Control, International Journal of Reliability, Quality and Safety Engineering, Journal of Information Science and Engineering.

Nai-Wen Kuo received his Ph.D. degree from the Institute of Management Science at Tamkang University. He is currently an Assistant Professor at the Department of Information Management at Chinese Culture University of Taiwan. His research interests are system dynamics, management information systems.