

Processing Logical Access Control Command in Computer System

Tsang-Yean Lee, Huey-Ming Lee, Wu-Yee Chen, Heng-Sheng Chen
Department of Information Management, Chinese Culture University
55, Hwa-Kung Road, Yang-Ming-San, Taipei (11114), TAIWAN
{tylee, hmlee, wuuyee, chenhs}@faculty.pccu.edu.tw

Abstract

Access control includes physical and logical access control. Physical access control protects the damage, theft and losing of assess. In this paper, we propose the method to protect logical access control. If we want to control the access information system in security, we can encrypt the executing command to cipher text and send it to the computer system. The computer system decrypts the cipher text to the original executing command. The computer system also checks executing command of availability to process.

1. Introduction

In general, the functions of security system are security, authenticity, integrity, non-repudiation, data confidentiality and access control [1-3, 13]. Diffie and Hellman [5] proposed the concept of public key. Rivest et al [12] proposed public cryptosystem. McEliece [7] used algebraic coding theory to propose public key. Merkle [8] presented "One way hash function" and used for digital signature. 1988, Miyaguchi [9] developed fast data encipherment algorithm (FEAL-8). NIST (National Institute of Standards and Technology) [10-11] proposed secure hash standard, namely SHS. All of these are encryption algorithm. Lee and Lee [5] used insertion, rotation, transposition, shift, complement and pack of the basic computer operations to design the encryption and decryption algorithm. We pack the final symbol table, relative data, relative tables and control byte to generate cipher text.

In Dhillon [4], information system needs to be secured at technical, formal, and informal level. Managing information system security is the implementation of the range of control to confidentiality, integrity and availability of data. Businesses use complex technological controls to protect the information held in their computer system.

Most of these controls have been in the area of access control and authentication.

In this paper, we propose the logical access control in security to encrypt the processing command to cipher text and send it to the computer system. The computer system decrypts the cipher text to the original command and checks this command to process.

2. Describing the Proposed Method

2.1 Processing Command

We divide the processing command into two kinds, saying, normal and access control command. Access control command has a special character in the first character (e.g., &), otherwise it is normal command. The format of access control command is as Table 1.

Table 1. Access Control Command

&	2-Digits	Cipher text
---	----------	-------------

In Table 1, 2-digits represent the location of format code, we will explain it in encryption algorithm. Cipher text is the encryption of processing command.

2.2 User Access Control Information Data Base

In the computer system, we use user-id as key to build UACIDB (User Access Control Information Data Base) for each user. We use UACIDB to validate users to access the computer system. The format of UACIDB is as shown in Table 2.

Table 2. User Access Control Information Data Base (UACIDB)

User-id	Password	Location of format code	Access level
---------	----------	-------------------------	--------------

The field of location of format code uses to encrypt the plaintext and decrypts the cipher text. Access level

controls user access right.

When user processes the request permission command, he/she receives location of format code and stores it to UADB (User Access Data Base) in local system as shown in Table 3.

Table 3. User Access Data Base (UADB)

User-id	Password	Location of format code
---------	----------	-------------------------

2.3 User Sends Processing Command

Users send the commands to host and the commands have the following formats.

A. Normal command

Users send normal command to host to process. The format of command is shown as Table 4.

Table 4. Normal Command

Command

B. Request permission command

When user wants to access control command, he/she sends this command to receive permission from the system. The system uses the default value of location of format code to encrypt user-id and password to produce cipher text. After receiving cipher text, the system decrypts it to get user-id and password. The system uses user-id as key to check password in UACIDB. If it is yes, the system returns location of format code and store it to UADB (User Access Data Base). The format of request permission command is shown as Table 5.

Table 5. Request Permission Command

&	00	Cipher text
---	----	-------------

Where 00 is the default value of location of format code and the cipher text is the encryption of user-id and password.

C. Access control command

When user wants to send access control command, he/she uses the value of *dd* (location of format code in UADB) to encrypt user-id, password and command to cipher text and sends cipher text out. The format of access control command is shown as Table 6.

Table 6. Access Control Command

&	<i>dd</i>	Cipher text
---	-----------	-------------

In Table 6, *dd* is the location of format code and cipher text is the encryption of user-id, password and command.

2.4 The User and System Operations

A. Send normal command.

User: User sends normal command as Table 4.

Host: Host processes normally.

B. Send request permission command to host and get permission.

User: User uses the default location of format code to encrypt user-id and password to produce cipher text. The format is shown as Table 5. He/she sends the cipher text of request permission command to host.

Host: Host uses default location of format code to decrypt cipher text, gets user-id and password, and then uses user-id as key to check password with UACIDB. If password is not correct, it returns error and exits. We get the new value of location of format code and store it to UACIDB and return this value to user.

User: If user receives error message, he/she corrects the errors and sends the corrected command again. If user receives the value of location of format code, he stores it to UADB.

C. Send access control command.

User: User encrypts user-id, password and command with the value of location of format code in UADB to produce cipher text as Table 6, sends this cipher text to host.

Host: Host receives the cipher text, reads “&” in the first character, and decrypts cipher text with the location of format code in UACIDB to get user-id, password and command. Host uses user-id as key to check if password and location of format code are the same as in UACIDB. If yes, host checks the command level with user’s access level. If user can access, host begins to process this command. If no, host returns error.

User: User continues to process.

2.5 System Command Analyzer Must Be Changed

System command analyzer must be changed. If “&” is the first character, he must decrypt the cipher text with the value of location of format code.

2.6 Produce Plaintext to Do Encryption

In request permission command, we combine user-id and password to produce plaintext. In access control command, we combine user-id, password and command to produce plaintext. We use this plaintext to encrypt and produce cipher text.

3. Algorithm of Cipher Text Containing Data and Key

We present the encryption step in Section 3.1. In Section 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 Section 3.3, we explain the fields of cipher text. We list the possible combinations of cipher text in Section 3.4. The decryption method is shown in Section 3.5.

3.1 Encryption Steps

Based on Lee and Lee [6], we propose the encryption algorithm in the following steps.

A. 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): $S_1S_2 \dots S_N$.

B. Data uncertainty;

Insert dummy symbol table to symbol table (ST)

- (a) Get any M dummy characters;
- (b) Append to symbol table (ST)
- (c) Get symbol table with dummy (STWD) as $S_1S_2 \dots S_N D_1 D_2 \dots D_M$

C. Avoidance of volume of same data or serial data to send:

Set rotated byte and rotate symbol table with dummy (STWD)

- (a) Get any character DD.
- (b) Set rotated byte, RB, as RB = DD mode (N+M).
- (c) If RB is odd then we rotate symbol table with dummy (STWD) to left RB times; else if we rotate symbol table with dummy (STWD) to right RB times.
- (d) Insert RB to the trailer of the above symbol table after rotation.
- (e) Get symbol table after rotation (STAR).
For example: if RB=4 then we have symbol table after rotation (STAR) as $D_{M-3}D_{M-2}D_{M-1}D_M S_1 S_2 \dots S_N D_1 D_2 \dots D_{M-4} RB$

D. Change contents of plaintext:

Shift the symbol table after rotation (STAR) to get symbol table after shift (STAS)

- (a) 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): $F_1F_2 \dots F_{N+M+1}$
- (b) Shift each byte of symbol table after rotation (STAR) according to the contained value of shift left table (SLT).
- (c) Get symbol table after shift (STAS) as $SS_1SS_2 \dots SS_{N+M+1}$

E. Position exchange:

Transpose the symbol table after shift (STAS) to get cipher text.

- (a) Set the position table (PT) as $P_1P_2 \dots P_{N+M+1}$
- (b) Following position table (PT), we change the location of the symbol table after shift (STAS) to produce symbol table after transposition (STAT).
- (c) Get symbol table after transposition (STAT) as $ST_1ST_2 \dots ST_{N+M+1}$

F. Produce cipher text.

Combine symbol table after transposition, left shift table, position table and variable of N and M.

- (a) Get the location of format code and value of format code.
- (b) Following the value of format code and the explanation of fields (3.3) of cipher text (3.4), we get cipher text.

G. Network transmission:

We compress the cipher text to avoid the transmission code.

3.2 Relative Tables and Data Used in Encryption Algorithm

Following tables are used for encryption algorithm.

SLT (Shift Left Table)	length N+M+1
PT (Position Table)	length N+M+1
STAT	length N+M+1
Data of N,M	length 2

Total length of above is $3N+3M+5$

where N is the length of plaintext, M is the length of dummy symbol.

3.3 Fields of Cipher Text

The fields in the cipher text are as follows:

1. FC: format code in the fixed field.

The value of FC is the different combinations of pointer field.

2. Pointers

- (1) PSLT: pointer of shift left table (SLT)
- (2) PPT: pointer of position table (PT)
- (3) PSTAT: pointer of symbol table after transposition (STAT)
- (4) PV: pointer of value of N, M

3. Tables and data

- (1) Shift left table (SLT)
- (2) Position table (PT)
- (3) Symbol table after transposition (STAT)
- (4) The value of N, M

- (5) From the pointers, we get the values of SLT, PT, STAT and value of N, M;
- (6) From position table (PT), we change the position to get symbol table after shift (STAS).
- (7) From shift left table (SLT), we shift each byte of STAS to get symbol table after rotation (STAR).
- (8) From the trailer of STAR, we get rotate byte (RB). The remainder of STAR is called STAR1. If RB is odd, we rotate STAR1 right RB times, else we rotate STAR1 left RB times. We get symbol table with dummy (STWD).
- (9) The first N characters of STWD is the plaintext.

3.4 Cipher Text

The cipher text has the different format depending on the vale of 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 SLT, PT and STAT and three pointers (P1, P2, P3) to represent PSLT, PPT and PSTAT and one pointer (PV) to value (V) (e.g., the value of N, M). We can define some value of format code and cipher text as shown in Table 7.

T1, T2, T3 and V may represent different combination of SLT, PT, STAT and value of N, M ; the values of pointers may increase by some value to avoid the value 1.

For example: The format code equals to 1. Suppose T1= SLT, T2=PT, T3= STAT, P1= PSLT, P2=PPT, and P3= PSTAT, then the cipher text is as (SLT) PSLT FC PPT PSTAT PV (PT) (STAT) (V).

Table 7. 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

3.5 Decryption Algorithm

Decryption algorithm is the reverse of encryption. The steps of decryption are as follows:

- (1) Decompress the cipher text.
- (2) We know the location of the format code;
- (3) When we read the location of format code, we get the format of cipher text;
- (4) We get the pointer of SLT, PT, STST and value of N, M;

4. Conclusion and Discussion

In this study, we used the basic computing operations to design these encryption and decryption algorithms. We use these algorithms to handle access control command. Finally, we make some comments about this study.

- (1) We install encryption code in user's site and decryption code in host site. We use them to encrypt the processing command and decrypt cipher text.
- (2) Each access control command has encryption process to produce cipher text. It is more secure.
- (3) Host decrypts the cipher text and gets password and command to check. It is easier to control.
- (4) Host uses more time to analyze command.
- (5) Each access control command uses cipher text to transmit in the network. It is in secure and safe.
- (6) In transmission, we know the location of format code and cipher text. We must decrypt cipher text to get user-id, password and command. It is difficult to reproduce cipher text of command.

5. References

- [1] E. Biham, and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystem, Advances in Cryptology-CRYPTO" '90 Proceedings, Berlin: Springer-Verlag, pp. 2-21, 1991
- [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, Berlin: Springer-Verlag, 1993
- [4] Dhillon, Gurpreet., Principles of Information Systems Security:Text and Cases, John Wiley & Sons, Inc. (2007)
- [5] W. Diffie, and M.E. Hellman, "New Directions in Cryptography", IEEE Trans. on Inform. Theory, pp. 644-654, 1976

- [6] T.-Y. Lee, H.-M. Lee, "Encryption and Decryption Algorithm of Data Transmission in Network Security", WSEAS Transactions on Information Science and Applications, Issue 12, Vol.3, pp.2557-2562, 2006
- [7] R.J. McEliece, "A Public-Key System Based on Algebraic Coding Theory", Deep Space Network Progress Report, 44, Jet Propulsion Laboratory, California Institute of Technology, pp. 114-116, 1978
- [8] R.C. Merkle, "One Way Hash Function and DES," Proc. Crypto'89, Berlin Springer-Verlag, pp.428-446, 1990
- [9] S. Miyaguchi, "The FEAL-8 Cryptosystem and Call for Attack," Advances in Cryptology-CRYPTO'89 proceedings, Berlin: Springer Verlag, pp. 624-627, 1990
- [10] National Institute of Standards and Technology (NIST), FIPS PUB 180: Secure Hash Standard (SHS), May 11, 1993
- [11] National Institute of Standards and Technology (NIST). NIST FIPS PUB 185, Escrowed Encryption Standard, February, 1994
- [12] R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public -Key Cryptosystems", Communications of the ACM, Vol. 21, No. 2, pp. 120-160, Feb, 1978
- [13] Stallings, W. "Cryptography and Network Security: Principles and Practices", International Edition, Third Edition by Pearson Education, Inc. Upper Saddle River, NJ 07458 (2003)