VNU Journal of Science, Natural Sciences and Technology 24 (2008) 170-178

Security of information processing based on grid environment

Huey-Ming Lee[1,*], Tsang-Yean Lee[1], Lily Lin[2]

[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, 56, Sec. 3, Hsing-Lung Road, Taipei (116), Taiwan*

Received 11 November 2007, received in revised form 20 November 2007

Abstract. Grid computing architecture was defined to be a complete physical layer. Based on the grid computing architecture, we divided grid nodes into supervisor grid node and execute grid nod. The data transfer in network must be in secure. In this study, we propose the encryption and decryption algorithm in each grid node to keep information processing in security. We create user information database both in supervisor and execute grid nodes. We use them to verify user processing in system. When these algorithms install in all grid nodes, we can keep processing be secure in all system.

Keywords: Decryption algorithm, Encryption algorithm, Grid computing, Security

1. Introduction

The term "Grid" was coined in the mid 1990s to denote a proposed distributed computing infrastructure for advanced science and engineering [1]. In grid environment, users may access the computational resources at many sites [2]. Lee *et al.* [3] proposed a dynamic supervising model which can utilize the grid resources, e.g., CPU, storages, etc., more flexible and optimal. Lee *et al.* [4, 5] proposed a dynamic analyzing resources model which can receive the information about CPU usages, number of running jobs of each grid node resource to achieve load-balancing and make the plans and allocations of the resources of collaborated nodes optimize.

In general, the functions of security system are security, authenticity, integrity, non-repudiation, data confidentiality and access control [6-9]. Rivest *et al.* [10] proposed public cryptosystem. McEliece [11] used algebraic coding theory to propose public key. Merkle [12] presented "One way hash function" and used for digital signature. Miyaguchi [13] developed the fast data encipherment algorithm (FEAL-8). All of these are encryption algorithm. Lee and Lee [14] used the basic computer operations, such as insertion, rotation, transposition, shift, complement and pack, to design encryption and decryption algorithm.

* Corresponding author.
E-mail: hmlee@faculty.pccu.edu.tw

In this paper, we propose the method to send information to other execute grid nodes through supervisor grid node. Supervisor checks the user to do the processes. We also propose encryption algorithm to encrypt information to produce cipher text and send it to supervisor. Supervisor uses sender format code to decrypts the cipher text to produce information. Once supervisor has checked, it uses received format code to encrypt information to produce cipher text and sends to the received execute grid node. The received execute grid node uses decryption algorithm to produce original information. Via the proposed algorithms, we can receive and send information in secure in network transmission.

2. Propose method description

The information is sent from one execute grid node to other execute grid node. We send information to supervisor grid node to check and verify. When it is correct, we send information to received executed grid node. The information is encrypted to produce cipher text and to be sent. When cipher text has received, we decrypt to produce original information. We explain the processes as follows.

2.1. Execute grid node

In the execute grid nodes, they have the following operations to do:

1) Sign on procedure first time

When the execute grid node signs on first time, it uses default format code to encrypt user-id and password and sends to supervisor grid node. It receives format code from supervisor and saves to create EUIDB (Execute User Information Data Base). The contents of EUIDB are as Table 1.

Table 1. EUIDB (Execute User Information Data Base)

User-id	Password	Format code

When user wants to send information, it uses format code in EUIDB to encryption user-id and password. When supervisor returns correct, it can send information to users.

2) Request permission from supervisor

When he wants to send information to other users, he inputs user-id and password to get permission from supervisor. We use format code in EUIDB to encrypt password and send to supervisor to process.

3) Change password

When user wants to change password, he inputs user-id, old password and new password. We use format code in EUIDB to encrypt password and send to supervisor to process.

4) Delete user

When user wants to delete entry in supervisor, he inputs user-id and password. We use format code in EUIDB to encrypt password and send to supervisor to process and delete the entry in EUIDB.

5) Send information to user in other execute grid node

When he wants to send information to other user, he types user-id, received-user-id and information. We use format code in EUIDB to encrypt received-user-id and information to produce cipher text and send to supervisor to process.

6) Receive information from supervisor grid node

When it receives cipher text from supervisor, it uses format code to decrypt cipher text to get information.

7) Exit from supervisor grid node

When user wants to log out, it sends user-id to supervisor.

2.2. Supervisor grid node

In the supervisor grid node, it handles information processing. It has following operations to do.

1) Receive new user sign on

When the new user signs on, it receives cipher text. It uses default format code to decrypt cipher text to get user-id and password. It uses user-id as key to access supervisor user information data base. If user exists and returns error code, otherwise he assigns a format code to user and creates an entry in the SUIDB (supervisor user information data base) as Table 2 and return format code. It creates an entry in the RUIDB (running user information data base) and inserts access time as Table3.

Table 2. SUIDB (Supervisor User Information Data Base).

User-id	Password	Format code

Table 3. RUIDB (Running user information data base).

User-id	Password	Format Code	Access Time

2) Receive user request

It receives the cipher text and uses use-id as key to find the format code in the SUIDB. If the user does not exist, it returns error code. It uses this format code to decrypt cipher text to get password. When the password is not the same as in SUIDB, it returns error code and exits. It creates an entry in the RUIDB and returns permission to access.

3) Receive information

When it receives the cipher text of information, it uses user-id as key to find the format code in the RUIDB. If the user-id does not exist, it will return error code and exist. It uses the format code to decrypt cipher text to find received-user-id and information. It uses receive-user-id as key to find the format code of this received-user-id. If the user does not exist, it will return error code to user of sender and exit. It uses format code of received-user-id to encrypt user-id and information to produce cipher text. It sends the cipher text to received-user-id. We update access time field in RUIDB.

4) Receive return message from receive user

When it receives return message from received-user-id, it uses the user-id as key to find the format code and decrypt to find original user-id and message. It uses the format code of original user-id to encrypt message to produce cipher text and return to original user. We update access time field in RUIDB.

5) Force to process sign out

When user does not process a periodical time, supervisor releases the entry in RUIDB.

3. Framework of the proposed model

In this section, we present the framework of the proposed security of information process model based on grid environment. Based on the grid computing architecture, we divide grid nodes into supervisor grid node (S0) and execute grid node (Xi). We also present the supervisor information process module (SIPM) on the supervisor grid node, execute information process module (EIPM) on the execute grid node, as shown in Fig. 1.

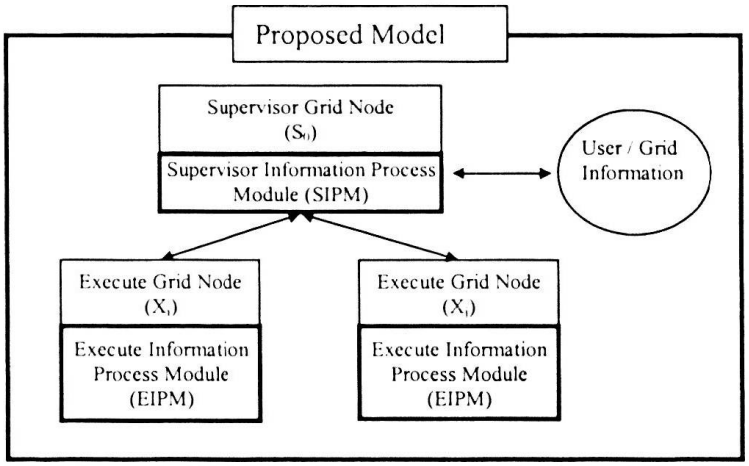

Fig. 1. Framework of the proposed model.

3.1. Supervisor grid node

We present the supervisor information process module (SIPM) on the supervisor grid node. The components in this module are shown in Fig. 2.

The functions of these components are as the follows:

1) Supervisor receive information component (SRIC):

SRIC receives information from the execute grid node. It calls information decryption component (IDC) to decrypt cipher text to get information. Calls SPIC (Supervisor Process Information Component).

2) Supervisor process information component (SPIC):

SPIC processes the request of execute grid nodes. We have the following actions.

(1) Type N. Use user-id as key to check SUIDB (Supervisor User Information Data Base). If user-id exists, it will return error code and exit. If user-id does not exist, it creates an entry with user-id, password and new format code in SUIDB and returns format code. We create an entry in RUIDB (Running User Information Data Base) as Table 3.

(2) Type P. We check user-id and password in SUIDB. If it is not correct, it returns error code and exits. We create an entry in RUIDB.

(3) Type U. We check user-id and password in SUIDB. If it is not correct, it will return error code and exit. We change password in SUIDB and store new format code and return format code. We create an entry in RUIDB.

(4) Type D. We check user-id and password in SUIDB. If it does not correct, it will return error code and exit. We delete user-id in SUIDB and return message.

(5) Type E. We delete the entry in RUIDB

(6) Type S. We use received-user-id as key to check in SUIDB. If it does not exist, it will return error code and exits. It uses format code of received-ser-id to call IEC to encrypt information to produce cipher text and send to received-user-id.

In each process, we change connect time in RUIDB when required and write the text to log file.

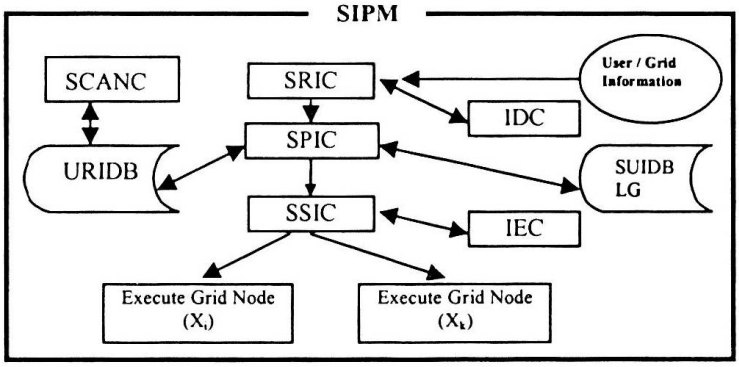

Fig. 2. Architecture of the SIPM.

3) Supervisor check active node component (SCANC):

SCANC processes periodically. If user does not connect for a period, supervisor deletes the entry in URIDB.

4) Supervisor send information component (SSIC):

SSIC sends information to grid node.

3.2. Execute grid node

We present the execute information process module (EIPM) on the execute grid node in this section. The components in this module are shown in Fig. 3.

The functions of these components are as the follows:

1) Execute receive information component (ERIC):

ERIC receives information. If it receives from supervisor, it calls EPSIC (Execute Process Supervisor Information Component), otherwise it calls EPUIC (Execute Process User Information Component).

2) Execute process user information component (EPUIC):

EPUIC processes to send user information to supervisor. It has the following formats.

(1) First time sign on. Set code as N and type user-id and password.

(2) Request permission. Set code as P and type user-id and password.

(3) Change password. Set code as U and type user-id, old password and new password.

(4) Send information. Set code as S and type user-id, received-user-id and information.

(5) Exit. Set code E and type user-id to exit from supervisor.

In (1), we use default format code. In (2) to (4), we get format code in EUIDB (Execute User Information Data base). We call IEC (Information Encryption Component) to encryption information to produce cipher text. Then call ESIC.

3) Execute process supervisor information component (EPSIC):

EPSIC calls IDC. IDC uses format code to decrypt cipher text to get information. From the receive code, it has following process.

(1) Code N. Receive format code and store to EUIDB (Execute User Information Data Base).

(2) Code P. Receive permission from supervisor.

(3) Code R. Receive return code from supervisor.

(4) Code S. Receive information from supervisor. This information comes from other user and returns message to user.

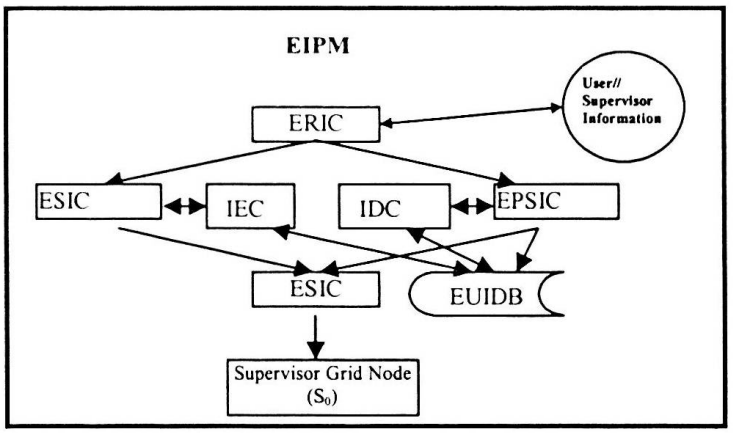

Fig. 3. Architecture of EIPM.

4) Execute send information component ESIC):

Execute send information component ESIC) sends information or return code to supervisor.

I. Encryption and decryption algorithm

1.1. Encryption algorithm (IEC Information encryption component)

The information has the following format as Table 4.

Table 4. Information

Code	User-id	Information

Information has different fields separated by comma. After processes the encryption, we produce the following format as Table 5 to send out.

Table 5. Information send out

Code	User-id	Cipher Text

We use the basic computer operations to design this algorithm. We explain each encryption step in Section 4. We let the length of information to be N and it is plaintext.

1). Encryption step

The encryption steps are as follows:

(1) Build the tables. The steps are as follows:

Step 1: Store plaintext to symbol table. From plaintext, we set symbol table ST as N to store plaintext.

Step 2: Set shift count to SC. SC is 1 to 7. We left shift every byte of ST to SC places. We set SC to SC+32.

Step 3: Insert M dummy symbol to trail of ST. We get any M (=INT (N/10)+1) dummy symbol and insert to the trail of symbol table. The length of symbol table is N+M.

Step 4: Set rotate byte and rotate symbol table. Get any character DD_1, DD_2. Set rotated byte RB_1, as $RB_1 = DD_1$ mode ((N+M)/2) and $RB_2 = DD_2$ mode ((N+M)/2). We divide ST into two equal parts, saying SP1 and SP2, lengths of SP1 SP2 are equal or length (SP1)=length (SP2)+1. We rotate SP1 to left RB_1 times and rotate SP2 to right RB_2 times. Insert RB_1, RB_2 to the trailer of combination of new SP1 and SP2. Get symbol table after rotation (STAR).

Step 5: Complement the symbol table after rotation. Set control bit table (CBIT) to all 0 and byte length to L= [(N+M+2)/8+1]. If the value of STAR is below the certain value (ex. 20_{16}), we complement the symbol of STAR to get symbol table after complement (STAC) and set the relative bit of CBIT to 1.

***Step 6*: *Packed control byte table*.** To form control byte table (CBT), we take each 7 bits (as *eeeeeee*) of CBIT from left and set control byte as *eee1eeee*. The length of CBT is K=[(N+M+2)/7] +1.

(2) Build background symbol table (BST)

Step 1: Reserve table

Set S to format code. We set number L=2*N+S. We reserve table size as L.

Step2: Set value of table. Set above table as random value between 20_{16} to $F0_{16}$

(3) Build cipher text

We have STAC (symbol table after complement), CBT (control byte table), SC, N, M, and K.

From format code, we store SC, STAC and CBT to BST and BST is cipher text.

2). Format code

We may define some value of format code as showing Table 6.

3). Message format

The format of sending message has fields as Code, User-id and cipher text.

4). Algorithm description

In this algorithm, we have solved the following items.

(1) Data uncertainty;

(2) Brute-force by volume of data to send;

(3) Change contents of plaintext;

(4) Network transmission;

(5) Simple computation.

5). Combination possibility

Encryption Step		Times of Combination
(1)	Shift the symbol table	8**(N)
(2)	Insert dummy symbol	256**M
(3)	Set rotate byte and rotate	((N+M)/2)**2
(4)	Complement the STAR	2**(N+M+2)
(5)	Packed	2**7*(INT ((N+M+1)/7)+1)
(6)	Reserve Table	240**(0.7N)
(7)	Format code	240

The total possible combinations are 8**(N)*256**M*(N+M)*
((N+M)/2)**2**2**(N+M+2)
*2**7*(INT((N+M+1)/7)+1)*240**(0.7N)*240

This number is large. It is difficult to decrypt.

4.2. Decryption algorithm (IDC Information Decryption Component)

Decryption is the reversed order of encryption. Before decryption, we should know the values S of format code in execute user information data base and U (length of user-id +1 (Code)). We get the L (length of message). We can compute the length of tables as follows

Table 6. Contents of format code and cipher text.

Format Code	Cipher text Content
1	SC,STAC,CBT
2	SC, dd, STAC, dd, CBT
3	STAC, SC, CBT
...	....
>127	Store in reverse order

where dd is the character skipped.

The length of symbol table N=1/2*(L-S-U).

The length of dummy symbol M= INT(N/10)+1.

The length of CBT K=[(N+M+2)/7] +1.

From different format code and above values, we can get SC, STAC and CBT.

1) The steps of decryption algorithm are as follows:

Step 1: Get from cipher text (CT). We get N, M, K, SC, STAC and CBT.

Step 2: Pack control bit table (CBIT). We retrieve 7 bits (skip the 5th bit from left of each byte) from each CBT. We pack above bits to form the CBIT and length L=[(N+M+2)/8]+1.

Step 3: Complement symbol table after complement (STAC). From each bit of CBIT, if the value of relative bit is 1, we complement the corresponding byte of STAC and get symbol table after rotation (STAR).

Step 4: Rotate symbol table after rotation (STAR). Get rotated byte $RB_1=STAR_{N+M+1}$ and $RB_2 = STAR_{N+M+2}$. We divide first N+M symbols of STAR to two equal parts, saying SP1 and SP2, lengths of SP1 and P2 are equal or length (SP1) =length (SP2) +1. We rotate SP1 to right RB_1 times and rotate SP2 to left RB_2 times. We combine SP1 and SP2 to get symbol table after shift (STAS).

Step 5: Shift the symbol table after shift (STAS). Set SC=8-(SC-32). We left shift each byte of first N bytes of STAS and get the plaintext. This is the original plaintext.

5. Conclusion and discussion

In this study, we use the basic computing operations to design the encryption and decryption algorithms. It doesn't need any special hardware. Finally, we make some comments about this study.

a) To do the encryption, we must know format code to produce cipher text.

b) Each cipher text may have different length and format because it has different format code and the length of dummy symbol table.

c) To do decryption, we must know format code, shift count and different format to decrypt cipher text to plaintext.

d) 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) shift count, (c) rotation (d) background table of random data.

e) Message processes through encryption and decryption are more secure.

f) Give permission from supervisor and do information process.

Acknowledgements. This work was supported in part by the National Science Council, Republic of China, under Grant NSC-96-2745-M-034-002-URD.

References

[1] I. Foster, C. Kesselman, S. Tuecke, "GRAM: Key concept", Available: http://www-unix.globus.org/toolkit/docs/3.2/gram/key/index.html, July 31, 1998

[2] I. Foster, C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", *International Journal of Supercomputer Application* Vol. 11 No. 2 (1997) 115.

[3] H.M. Lee, C.C. Hsu, M.H. Hsu, "A Dynamic Supervising Model Based on Grid Environment", *Knowledge-Based Intelligent Information & Engineering Systems,* LNCS 3682, Springer-Verlag, (2005) 1258.

[4] H.M. Lee, T.Y. Lee, C.H. Yang, M.H. Hsu, "An Optimal Analyzing Resources Model Based on Grid Environment", *WSEAS Transactions on Information Science and Applications*, Issue 5, Vol. 3 (2006) 960.

[5] H.M. Lee, T.Y.Lee, M.H. Hsu, "A Process Schedule Analyzing Model Based on Grid Environment", *Knowledge-Based Intelligent Information & Engineering Systems*, Part III, LNAI 4253, Springer-Verlag, (2006) 938.

[6] E. Biham, A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystem", *Advances in Cryptology-*

CRYPTO '90 Proceedings, Berlin: Springer-Verlag, (1991) 2.

[7] E. Biham, A. Shamir, "*A Differential Cryptanalysis of the Data Encryption Standard*", Springer, Berlin Heidelberg New York, 1993

[8] E. Biham, A. Shamir, "*Differential Cryptanalysis of Data Encryption Standard*", Berlin: Springer-Verlag, 1993.

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

[10] R.L. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public –Key Cryptosystems", *Communications of the ACM*, Vol. 21 No. 2 (1978) 120.

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

[12] R.C. Merkle, "One Way Hash Function and DES," *Proc. Crypto '89*, Berlin Springer-Verlag (1990) 428.

[13] S. Miyaguchi, "The FEAL-8 Cryptosystem and Call for Attack," *Advances in Cryptology-CRYPTO '89 proceedings*, Springer-Verlag (1990) 624.

[14] 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 (2006) 2557.