INTERNATIONAL JOURNAL OF COMPUTERS
Issue 3, Volume 1, 2007

Processing Authentication Based on Grid Environment

TSANG-YEAN LEE, HUEY-MING LEE, JIN-SHIEH SU, and HENG-SHENG CHEN

***Abstract*—**In this study, we propose the encryption algorithm to produce authenticator. The grid nodes are divided to supervisor and execute grid nodes. We use this authenticator to create execute user information data base in execute grid node and remote user information data base in supervisor grid node. We use the authenticator to run authentication application. When these authentication applications install in all grid nodes, we can run the system more secure**.**

***Keywords*—**Authentication, Authenticator, Decryption, Encryption.

I. 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 to achieve load-balancing and make the plans and allocations of the resources of collaborated nodes

In general, the functions of security system are security, authenticity, integrity, non-repudiation, data confidentiality and access control [6]. Rivest et al. [7] proposed public cryptosystem. McEliece [8] used algebraic coding theory to propose public key. Merkle [9] presented "One way hash function" and used for digital signature. Miyaguchi [10] developed fast data encipherment algorithm (FEAL-8). All of these are encryption algorithm. Lee and Lee [11] used \insertion, rotation, transposition, shift, complement and pack of the basic computer operations to design encryption and decryption algorithm. Lee *et al.* [12] proposed authentication algorithm based on grid environment.

Based on Lee *et al.* [4], we propose the authentication application in the grid nodes in this paper. We encrypt user-id and password to generate cipher text of authenticator and store it to remote user information database in the supervisor grid node. We send authenticator to the execute grid node to store in execute user information database. If the supervisor wants to decrypt the cipher text of authenticator, then it has authentication decryption algorithm to get user-id and password. We use it to verify users from both grid nodes.

II. PROPOSED AUTHENTICATION DESCRIPTION

Based on grid computing architecture,, we divide grids to supervisor, executive and backup grid nodes. Users in the execute grid node send commands to supervisor grid node to process the authenticator operation. If the supervisor verifies authenticator to be correct, it returns message to the execute grid node. Via the supervisor's authentication, users in the execute grid nodes can continue to process.

A. Supervisor Grid Nod

In the supervisor grid node, it processes user's commands and needs the following tables, database and operations.

1. Supervisor Tables and Database

The supervisor receives commands as Table 1 from execute grid node. It uses user-id as key to compare authenticator in RUIDB (Remote User Information Data Base) as Table 2. The user in the execute grid node sends new command as Table 3 to the supervisor. The supervisor encrypts user-id and password to produce authenticator and stores authenticator in the RUIDB.

Table 1. Receive command

Command	User-Id	authenticator

Table 2. Remote user information data base

User-Id	Authenticator

Table 3. New command

New	User-Id	Password

2. Supervisor Operations

In the supervisor grid node, it processes the following operations.

(1) New Operation

Manuscript received Jan. 3, 2007; Revised received April 5, 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).

J.-S. Su is with Department of Applied Mathematics, Chinese Culture University, TAIWAN (e-mail: suston@tpts8.seed.net.tw)

H.-S. Chen is with Department of Information Management, Chinese Culture University, TAIWAN (e-mail: chenhs@faculty.pccu.edu.tw).

When new user comes as Table 3, the supervisor encrypts user-id and password to produce authenticator. It uses user-id as key to save authenticator in RUIDB as Table 2.

(2) Update Operation

When user wants to change his authenticator, he sends update command as Table 4 to the supervisor. The supervisor uses user-id as key to check authenticator the same as in RUIDB. If yes, it encrypts user-id and new password to produce authenticator, replaces authenticator in RUIDB and returns authenticator to execute grid node.

Table 4 Updater command

Update	User-Id	Authenticator	New Password

(3) Delete Operation

When user does not want to process, he sends delete command as Table 5 to supervisor grid node. The supervisor uses user-id as key to check user-id and the authenticator in Table 5 the same as in RUIDB. If they are the same, the supervisor deletes the entry of user-id in RUIDB.

Table 5 Delete command

Delete	User-Id	Authenticator

(4) Permission Operation

When user wants to get permission, he sends permission command as Table 6 to the supervisor. The supervisor uses user-id as key to check authenticator in Table 6 the same as in RUIDB. If yes, it returns correct message, else returns error.

Table 6 Permission command

Permission	User-Id	Authenticator

B. Execute Grid Node

In the executive grid node, user sends commands to supervisor grid node to process. The executive grid node creates EUIDB (Execute User Information Data Base) and processes the following operations

1. Execute Tables and Database

In the executive grid node, user sends new / update commands to supervisor grid node to get authenticator. The executive grid node stores authenticator to EUIDB as Table 7.

Table 7 Execute user information data base

User-Id	Authenticator

User can send new / update / delete / permission command to the supervisor grid node to process. The format of command is in Section 2.1.1.

2. Execute Operation

The operations in the execute grid node processes the following operations.

(1) New Operation

The execute grid node sends new command as Table 3 to supervisor to get authenticator. It saves the authenticator to EUIDB as Table 7.

(2) Update operation

When user in the execute grid node wants to change authenticator, the execute grid node sends update command as Table 4 to supervisor grid node. The executive grid node receives new authenticator from the supervisor and stores authenticator in EUIDB.

(3) Delete Operation

When user does not access again, the execute grid node sends delete command as Table 5 to supervisor grid node. If supervisor verifies authenticator correct, supervisor deletes the entry of this user in RUIDB. The execute grid node deletes the entry in EUIDB too.

(4) Permission Operation

When user wants to access, he sends permission command as Table 6 to supervisor grid node to process. When user gets permission from supervisor, he can continue to process commands.

III. Framework of the Proposed Authentication

In this section, we present the framework of the proposed authentication. We present the supervisor authentication (SA) on the supervisor grid node (S_0), execute authentication (EA) on the backup supervisor (B_1) and execute grid node (X_i), as shown in Fig. 1.

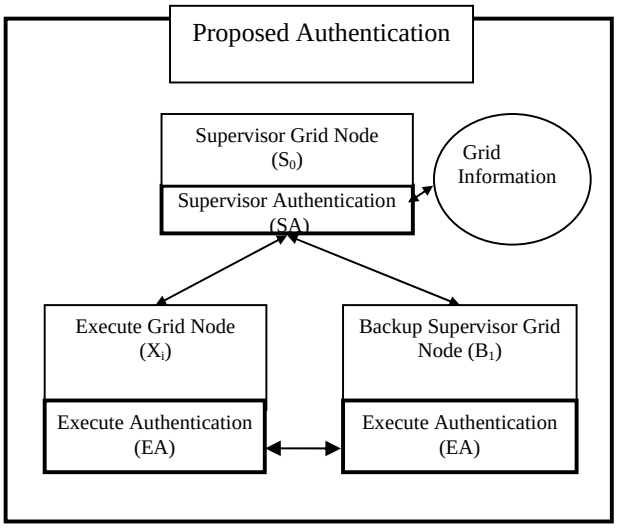

Fig. 1. Architecture of the proposed authentication

A. Supervisor Grid Nodes

We present the supervisor authentication (SA) on the supervisor grid node. There are three modules in this authentication as shown in Fig. 2.

The functions of these modules are as the follows:

(1) Supervisor user interactive module (SUIM):
SUIM processes user request from remote execute grid nodes. It calls supervisor remote authentication module (SRAM) to process.

(2) Supervisor remote authentication module (SRAM):
SRAM accesses remote authentication. The operations of SRAM are as follows:

(i) Supervisor create remote (SCR): it calls authentication encryption component (AEC) to encrypt user-id and password to produce authenticator and stores it to remote user information data base (RUIDB) and write Table 4 to log file (LG).

(ii) Supervisor replaces remote (SRR): it uses user-id as key to check authenticator in supervisor the same as in RUIDB. If yes, it calls AEC to encrypt user-id and new password to produce authenticator and replaces authenticator in RUIDB and write the replace operation to log file (LG).

(iii) Supervisor delete remote (SDR): it uses user-id as key to check the user-id and authenticator in Table 5 the same as in RUIDB. If both are the same, the supervisor deletes the entry of this user-id in RUIDB and writes the delete operation to log file.

(iv) Supervisor permit remote (SPR): it uses user-id as key to verify that the authenticator in Table 6 the same as in RUIDB. If yes, it returns permission message to request execute grid.

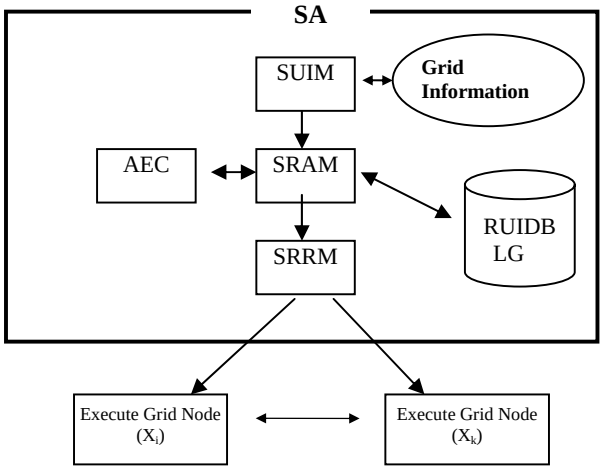

Fig. 2. Framework of the SA

(3). Supervisor return remote message (SRRM): SRRM returns the result messages to execute grid node.

B. Execute Grid Nodes

We present the execute authentication (EA) on the execute grid node as shown in Fig. 3.

The functions of these modules are as the follows:

(1) Execute user interactive module (EUIM): EUIM processes user's requests from local or remote supervisor grid nodes. If user sends commands to supervisor, the execute grid node calls execute process authentication module (EPAM) to process. If execute grid node receives data from supervisor, it calls execute receive authentication module (ERAM) to process.

(2) Execute processes authentication module (EPAM): EPAM accesses the execute authentication operation. The operations of EPAM are as follows:

(i) Execute create (EC): EC sends new command as Tale 3 to supervisor to request authenticator.

(ii) Execute replace (ER): ER sends update command as Table 4 to supervisor to change authenticator.

(iii) Execute delete (ED): ED sends delete command as Table 5 to supervisor to delete authenticator in RUIDB and EUIDB.

(iv) Execute permit (EP): EP sends permit command to supervisor to check authenticator.

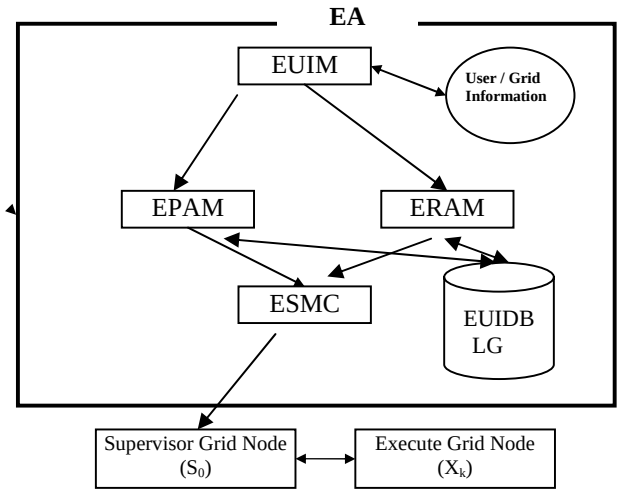

Fig. 3. Framework of the EA

(3) Execute receive authentication module (ERAM): ERAM receives return message from supervisor and processes the following operations.

(i) Execute receive new module (ERNM): ERNM receives authenticator and saves authenticator to EUIDB

(ii) Execute receive update module (ERUM): ERUM receives authenticator and replaces it in EUIDB.

(iii) Execute receive delete module (ERDM): ERDM receives correct message, and deletes user's entry in EUIDB.

(iv) Execute receive permit module (ERPM): ERPM receives permission from supervisor and continue to process.

(4) Execute send message module (ESMM): ESMM sends message to supervisor grid node.

We use user-id as key to create EUIDB (Execute User Information Data Base). The EUIDB contains user-id and authenticator. The format is as Table 7.

C. Backup Supervisor Grid Node

Backup supervisor grid node works as the execute grid node normally. If the backup supervisor does not receive messages from the supervisor for a period, then it becomes the supervisor

IV. Produce Authenticator Encryption Algorithm Description

In order to encrypt plaintext to cipher text, we should solve the following items.

(1) Change contents of plaintext;
(2) Volume of same data to send;
(3) Network transmission;
(4) Position exchange;
(5) Data uncertainty;
(6) Simple computation;
(7) Store key in cipher text.

In the proposed algorithm, we have solved (1) (2) (3) (4) (6) of above items. Because authenticator must be unique for each user, we do not have (5) and (7).

A. Encryption Algorithm (AEC Authentication Encryption Component)Grid Nodes

The encryption algorithm of AEC is as follows:

1. Create symbol table of plaintext.
   (1) The plaintext is the combination of user-id and password.
   (2) Let user-id be $U_1U_2\ldots U_U$, password be $P_1P_2\ldots P_P$.
   (3) Store them in the symbol table (ST) as $U_1U_2\ldots U_U$ $P_1P_2\ldots P_P$., and N=U+P
2. Change contents of plaintext:
   (1) Set rotated byte and rotate symbol table.
   Set rotated byte $RB_1= P_{P-1}P_P$ mode (N/2) and $RB_2 = P_{P-3}P_{P-2}$ mode (N/2). We divide symbol table (ST) to two equal parts, saying SP1 and SP2, length (SP1)= length(SP2) or length(SP1) =length(SP2)+1. We rotate SP1 to left RB_1 times and rotate SP2 to right RB_2 times. Insert RB1, RB2 to the trailer of combination of new SP1 and SP2. Get symbol table after rotation (STAR) as $SP1_1\ldots SP2_1\ldots RB_1\ RB_2$
   (2) Shift the symbol table
   (i) Get shift left table (SLT) of each byte, the contained value of shift left table is between 0 to 8, as shown below: Shift Left Table: (SLT): $F_1F_2\ldots F_{N+2}$
   (ii) Shift each byte of symbol table after rotation (STAR) according to the contained value of shift left table (SLT).
   (iii) Get symbol table after shift (STAS) as $SS_1SS_2\ \ldots SS_{N+2}$
3. Network transmission:
   (1) Complement the symbol table after shift (STAS)
   (i) Set control bit table (CBIT) to all 0 and byte length is L= [(N+1)/8+1].
   (ii) If the value of symbol table after shift (STAS) is below the certain value (ex. 20_{16}), we complement the symbol of symbol table after shift (STAS) to get symbol table after complement (STAC) and set the relative bit of control bit table (CBIT) to 1.
   (iii)The results of these two tables are as follows:
   Control Bit Table (CBIT): $C_1C_2\ \ldots C_L$
   Symbol Table after Complement (STAC): $SS_1SC_2\ldots SS_{N+2}$
   (2) Packed control byte table
   (i) To form control byte table (CBT), we take each 7 bits (as eeeeeee) of control bit table (CBIT) from left and set control byte as ee1eeeee. The length of control byte table is K=[(N+1)/7] +1.
   (ii)Get control byte table (CBT) as $(C1B_1)(C1B_2\ )...(C1B_K)$
   (3) Combine symbol table after complement and control byte table to symbol table after combination
   (i)Combine symbol table after complement (STAC) and control byte table (CBT).
   (ii)Get symbol table after combination (SAC) as $SS_1\ldots SS_{N+2}C1B_1\ldots C1B_K$
4. Position exchange:
   (1) Transpose the symbol table after combination to get cipher text.
   (i) Set the position table (PT) as $P_1P_2\ldots P_{N+2+K}$
   (ii) Following position table (PT), we change the location of the symbol table after combination (SAC).
   (iii) Get cipher text (CT) as $SP_1SP_2\ldots SP_{N+2+K}$.
5. Produce authenticator
   We combine CT (Cipher Text), SLT (Shift Left Table), PT (Position Table), U and P to produce authenticator.

B. Decryption Algorithm (ADC Authentication Decryption Component)

When ADC is required, we must know SLT (Shift Left Table) PT (Position Table) and the values of U (length of user-id) and P (length of password). Decryption is the reverse order of encryption. We get authenticator from message.

The steps of decryption algorithm are as follows:

(1) Get the authenticator
(2) Get CT (Cipher Text), SLT (Shift Left Table), PT (Position Table), U and P.
(3) Position exchange: Using transposition operation.
(4) Network transmission: Using pack and complement operations.
(5) Restore contents: Using shift and rotate operations.

C. Combination Possibility

If we want to cryptanalysis, the combination of each encryption step has the following:

Encryption Step	Times of Combination
Set Symbol Table	1
Set rotate byte	(P+U)/2* (P+U)/2
Shift Left	8**(U+P+2)
Complement	2**(U+P+2)

Packed	2**7*(INT((U+P+1)/7+1)
Transposition	(U+P+2)!

The total possible combinations are 1*(PTU)/2*(PTU)/2 * 8** U+P+2)*2**(U+P+2)*2**7*(INT((U+P+1)/7+1)*(U+P+2)! This number is very large and is difficult to get the computational formula.

V. PERFORMANCE (ENCRYPTION AND DECRYPTION)

In this section, we use INTEL, Pentium D830 DDR to implement these algorithms. The processing time of these encryption and decryption are in Table 8.

Table 8. Encryption and Decryption Processing Time

	Encryption (Bytes)		Decryption (Bytes)	
Times[*1]	8	32	8	32
1M	6.42[*2]	10.98	5.59	11.48
4M	25.58	43.92	23.02	45.20
8M	54.14	87.95	45.66	91.84

[*1]M=1000000 processing times,
[*2] processing time in second

VI. CONCLUSION AND DISCUSION

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

1) In authentication, we only use user-id and password to call authentication encryption component (AEC) to produce authenticator.
2) If the length of authenticator is short, we can double the symbol table.
3) The reasons of difficult cryptanalysis are as follows:
   (i)Through rotation and transposition, when plaintext is on sequence numbers, the authenticator has changed and it may be different position. It is difficult to process cryptanalysis.
   (ii) Through rotation and left shift, the content has changed.
   (iii)Through complement, we can avoid control codes of transmission.
4) Using basic operations, we don't need complex computation.
5) From the Table 8, we have the processing time of proposed encryption and decryption is smaller than other's algorithms.
6) In supervisor grid node, we can also do local authentication.

REFERENCES

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

[2] I. Foster, C. Kesselman, *Globus: "*A Metacomputing Infrastructure Toolkit, *" International Journal of Supercomputer Application*, Vol. 11, No. 2, 1997, pp. 115-128.

[3] H.-M. Lee, C.-C. Hsu, and M.-H. Hsu, "A Dynamic Supervising Model Based on Grid Environment," *Knowledge-Based Intelligent Information & Engineering Systems, LNCS 3682/2005*, Springer-Verlag, 2005, pp.1258-1264.

[4] H.-M. Lee, T.-Y. Lee, C.-H. Yang, and M.-H. Hsu, " An Optimal Analyzing Resources Model Based on Grid Environment," *WSEAS Transactions on Information Science and Applications*, Issue 5, Vol. 3 , 2006, pp. 960-964.

[5] H.-M. Lee, T.-Y. Lee, and M.-H. Hsu, "A Process Schedule Analyzing Model Based on Grid Environment *", Knowledge-Based Intelligent Information & Engineering Systems, LNAI 4253/2006*, 2006, pp. 938-947.

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

[7] 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, Feb. 1978, pp. 120-126.

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

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

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

[11] T.-Y. Lee, and 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, pp.2557-2562

[12] H-.M. Lee, T.-Y. Lee, H.-S. Chen, and J.-S. Su, "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

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.

Jin-Shieh Su is a lecture in the Deaprtment of Applied mathematics at the Chinese Culture University. His research interests are in the field of fuzzy sets theory and its applications, operation research, and information systems. His papers appeared in European Journal of Operational Research, Computers and Operation Research, International Journal of Innovative Computing, Information & Control, Journal of Information Science and Engineering.

Heng-Sheng Chen is a director of the Information and Communication Center at the Chinese Culture University. His research interests are in the field information systems.