Authentication Algorithm Based on Grid Environment

HUEY-MING LEE, TSANG-YEAN LEE, CHENG-SHENG CHEN, JIN-SHIEH SU
Department of Information Management, Chinese Culture University
55, Hwa-Kung Road, Yang-Ming-San, Taipei (11114), TAIWAN

Abstract: - We propose the encryption and decryption algorithm to produce authenticator. In local execute grid node, we use this authenticator to build user information data base and do authentication application. In supervisor grid node, we use the authenticator to build remote user information data base and do authentication application. When these algorithms install in all grid nodes, we can run authentication in all system more secure and effective.

Key-Words: - Authentication; Decryption; Encryption,

1 Introduction

The term "Grid" was coined in the mid 1990s to denote a proposed distributed computing infrastructure for advanced science and engineering [2]. In grid environment, users may access the computational resources at many sites [1]. 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 optimize.

In general, the functions of security system are security, authenticity, integrity, non-repudiation, data confidentiality and access control [12]. Rivest et al. [10] 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). All of these are encryption algorithm. Lee and Lee [6] used insertion, rotation, transposition, shift, complement and pack of the basic computer operations to design encryption and decryption algorithm.

In this paper, we propose the authentication encryption algorithm in the execute grid node. We apply user-id and password to generate authenticator to send and create user information data base in the supervisor node. Supervisor has authentication decryption algorithm to decrypt authenticator to get user-id. We use user-id as key to access user information data base and verify users from execute grid node.

2 Framework of the Proposed Authentication

In this section, we present the framework of the proposed authentication. Based on grid computing architecture, we divide grid nodes into supervisor grid node (S0), backup supervisor grid node (B1) and execute grid node (Xi). We also present the supervisor authentication (SA) on the supervisor grid node, execute authentication (EA) on the backup supervisor and execute grid node, as shown in Fig. 1.

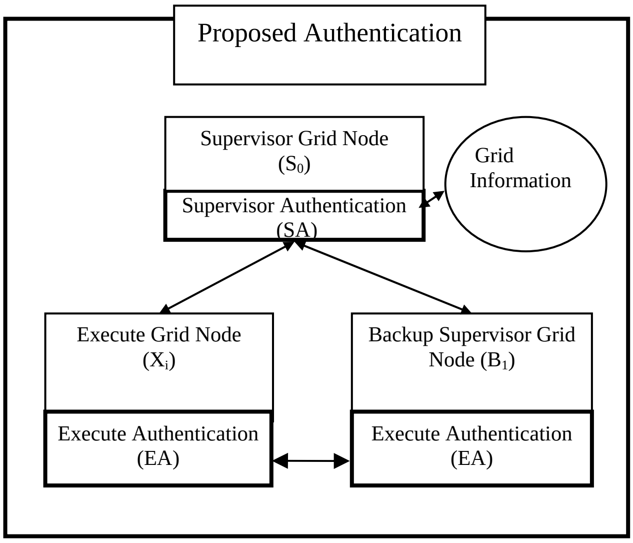
Proposed Authentication
Supervisor Grid Node (S_0)
Supervisor Authentication (SA)
Grid Information
Execute Grid Node (X_i)
Backup Supervisor Grid Node (B_1)
Execute Authentication (EA)
Execute Authentication (EA)

Fig. 1. Architecture of the proposed authentication

2.1 Supervisor Grid Node

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

The functions of these modules are as the follows:

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

(2)Supervisor remote authentication module (SRAM):
Supervisor remote authentication module (SRAM) uses to access remote authentication. Calls remote authentication decryption component (RADC) to decrypt authenticator from execute grid node and gets user-id and authenticator. The operations of SRAM are as follows:

(i) Supervisor create remote (SCR): it uses user-id and authenticator to create remote user information data base (RUIDB) and write to log file (LG).

(ii) Supervisor delete remote (SDR): it uses user-id as key to delete remote user information data base (RUIDB) and write to log file.

(iii) Supervisor replace remote (SRR): it uses user-id as key to replace remote user information data base (RUIDB) by new authenticator and write to log file (LG).

(iv) Supervisor access remote (SAR): it uses user-id as key to verify that if the user authenticator exists from remote user information data base (RUIDB) and then returns message to request execute grid node or request execute grid node to be transferred.

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

We use user-id as key to create RUIDB (remote user information data base). The contents are user-id, authenticator and node name (if it needs). The format is as table 1.

Table.1. RUIDB (Remote User Information Data Base)

User-Id	Authenticator

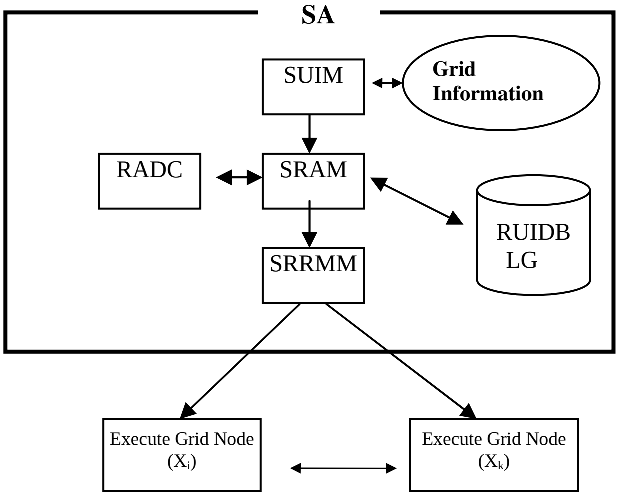

Fig. 2. Framework of the SA

2.2 Execute Grid Node

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): It uses to process user's requests from local or remote supervisor or remote execute grid nodes. If it receives data from local operation, it calls local authentication module (LAM) to process local authentication. If it receives data to remote operation, it calls execute send authentication module (ESAM) to send to supervisor or calls execute access supervisor/execute request module (EASERM) to process supervisor or other execute grid node request processes.

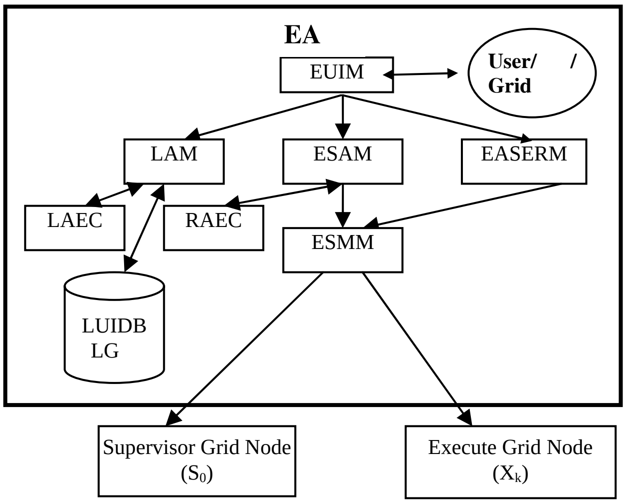

Fig. 3. Framework of the EA

(2).Local authentication module (LAM): Local authentication module (LAM) uses to access local authentication. Calls local authentication encryption component (LAEC) to encrypt input data (user-id and password) to get authenticator.

The operations of LAM are as follows:

(i) Local create (LC): Local create (LC) receives user information from the local node to create local user information data base (LUIDB) and writes to log file (LG).

(ii) Local delete (LD): Local delete (LD) receives user information from the local node to delete local user information data base (LUIDB) and writes to log file.

(iii) Local replace (LR): Local replace (LR) checks old user information from the local user information data base (LUIDB). If it exists, then it will replace local user information data base (LUIDB) by new authenticator and write to log file.

(iv) Local access (LA): Local access (LA) receives user information from the local node to verify from local user information data base (LUIDB) to check user exist.

(v) Local message (LM): Local message component (LM) processes returned message to users.

(3).Execute send authentication module (ESAM): It call remote authentication encryption component (RAEC) to encrypt user-id and password to produce authenticator and send the authenticator to supervisor grid node.

(4).Execute access supervisor/execute request module (EASERM): It processes supervisor or other execute grid node request process. If the message is encrypted, then it calls remote authentication decryption component (RADC) to decrypt message first. After processing, it returns messages to request node.

(5).Execute send message module (ESMM): It sends message to supervisor grid node or request grid node.

We use user-id as key to create LUIDB (local user information data base). The contents are user-id, authenticator. The format is as table 1.

2.3 Backup Supervisor Grid Node

Backup supervisor grid node works as the execute grid node normally. It replaces supervisor, when it decides that the supervisor can not work.

3 Produce Authenticator Encryption and Decryption 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).

3.1 Encryption Algorithm Locality (LAEC Local Authentication Encryption Component)

The encryption algorithm of LAEC is as follows:

3.1.1. 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

3.1.2. 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 $SS_1SS_2\ldots SS_{N+2}$

3.1.3. Position exchange:

(1) Transpose symbol table after shift.

Get transposition table (TT) as $T_1T_2\ldots T_{N+2}$. Following the transposition table (TT), we transpose the symbol table after shift (STAS) and get symbol table after transposition (STAT) as ST_1 $ST_2\ldots$ ST_{N+2}.

This is local to produce authenticator.

3.2 Encryption Algorithm Remote (Remote Authentication Encryption Component, RAEC)

3.2.1. Create symbol table

The same as Section 3.1.1.

3.2.2. Change contents of plaintext.

Same as Section 3.1.2

3.2.3. Network transmission:

(1) Complement the symbol table after rotation

(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 rotation (STAR) is below the certain value (ex. 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.

(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 $SS_1 \ldots SS_{N+2}C1B_1 \ldots C1B_K$

3.2.4. 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 $SP_1SP_2 \ldots SP_{N+2+K}$. This is authenticator.

3.2.5 Message sends out

The format of message is as length of authenticator, authenticator, grid name.

3.3 Decryption Algorithm (Remote Authentication Decryption Component, RADC)

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). Position exchange: Using transposition operation.

(3).Network transmission: Using pack and complement operations.

(4). Restore contents: Using shift and rotate operations.

3.4 Message

The message transmitted is length of authenticator, authenticator, node name, new authenticator (option). We decrypt authenticator to get user-id and use the user-id as key to build user information data base.

4 Performance (Encryption and Decryption only)

In this section, we use INTEL, Pentium D830 DDR to implement these algorithms. In local, we only need encryption. In supervisor, we need decryption. Table 2 is the result of processing time in supervisor.

Table 2. Encryption and Decryption Processing Time

	Encryption (Bytes)			Decryption (Bytes)		
Times[*1]	8	16	32	8	16	32
1M	6.42[*2]	7.98	10.98	5.59	7.61	11.48
4M	25.58	32.00	43.92	23.02	30.66	45.20
8M	54.14	63.98	87.95	45.66	61.77	91.84

[*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 and decryption algorithms. We don’t need any special hardware. Finally, we make some comments about this study.

(1) In local authentication, we only use user-id and password to call local authentication encryption component (LAEC) to produce authenticator and we do not need decryption process. In the remote authentication, the execute grid node calls remote authentication encryption component (RAEC) to encryption user-id and password, then it sends to supervisor. Supervisor calls remote authentication decryption component (RADC) to decrypt authenticator.

(2) In remote, we should have the following tables and values to do decryption:

(a) Position table (PT)

(b) Shift Left Table (SLT)

(c) Length of plaintext to encryption

(3) If the length of authenticator is short, we can double the symbol table.

(4) 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.

(5) Using basic operations, we don't need complex computation.

(6) From the Table 2, we have the processing time of proposed encryption and decryption is smaller than other's algorithms.

(7) In supervisor grid node, we can also do local authentication.

Acknowledgment

This work was supported in part by the National Science Council, Republic of China, under grant NSC95-2745-M-0034-007-URD. The authors gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References:

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

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

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

[4] Lee, H-.M., Lee, T.-Y., Yang, C.-H. and Hsu, M.-H.: *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] Lee, H-.M., Lee, T.-Y., Hsu M.-H., "*A Process Schedule Analyzing Model Based on Grid Environment* ", KES 2006, Part III. LNAI 4253, 2006, pp. 938-947.

[6] 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, Vol. 3, 2006, pp.2557-2562

[7] McEliece, R.J. "*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

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

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

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

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

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