Enhancement of User Authentication for Grid Computing Security

Komathi.P and Balakumar.P

Abstract--***Recently, Authentication protocol has been recognized as an important factor for grid computing security. This paper described a new simple and efficient Grid authentication system providing user anonymity. Our system is based on hash function, and mobile users only do symmetric encryption and decryption. In our system, it takes only one round of messages exchange between the mobile user and the visited network, and one round of message exchange between the visited network and the corresponding home network. The proposed architecture possesses several desirable emerging properties that enable it to provide an improved level of security for grid computing systems.***

Index Terms-- **grid computing security, Authentication protocol, anonymity, Access control**

I. INTRODUCTION

WHEN it comes to computational grids, some old challenges that have always existed in the realm of computing security still remain. Security is always a balance of vulnerabilities and threats.

Grids can be used to harness computational horsepower, provide access to unified data, or other intensive tasks. From a security manager's viewpoint, a corporate grid represents a high-value target for anyone who would want to gain unauthorized access. They need to be protected not only because they are high-value assets representing lots of hardware and software, but because they often serve a strategic function that's central to success.

At the same time, security managers understand that successful security is about tradeoffs. Tighten security too much, and it'll become harder for the R&D folks to share their findings with Engineering. Make it too hard to share information, and pretty soon all kinds of ad-hoc systems will start popping up that provide back doors to the system.

II. RELATED WORK

The use of a user's identity as the basis for delegation in distributed systems has venerable roots in existing security practice. However, it is the fundamental source of the cited scalability and flexibility problems. To solve these issues requires an acceptance that in very large distributed systems it will be impossible to base authorisation on individual user identity, the security policy for any given machine must not require a list of all possible remote users.

It is straightforward (in theory) to group users into roles; remote processing can then be authorised on behalf of groups. An individual user's session can be given temporary authority by an authorisation service

Acting for the group and individual accountability can still be traced via a session pseudonym. Remote machines need only to associate privileges with temporary accounts assigned to the group. This scheme can deal with some of the issues: user names are managed locally, providing more flexibility in forming and changing groups, sessions are identified pseudonymously enhancing the prospect of privacy.

Model between resource providers and the consumers presented a trust enhanced security solution, in which the trust decision could be used for Grid security enhancement decentralized

Grid security infrastructure (GSI) in Globus Toolkit uses PKI technologies to handle authentication, single sign-on, and trust delegation. However, it is not capable of assessing local security condition in a Grid site. The trust model we proposed the main aim to assess local security conditions to match with dynamically changed job security demands. We introduce the trust index of a Grid site, which is determined by site reputation and self-defense capability attributed to the site track record, risk conditions, hardware and software defenses deployed at a Grid site.

III. OBJECTIVE

The grid computing is the major resources sharing environment. The nodes are dynamically connected and Disconnected with the grid environment. The user authentication is the important factors for security. The system is designed with the following objectives. To assess local security conditions to match with dynamically changed job security demands. To protect hardware and software from unauthorized access. Support user authentication and security for ad hoc grid environment. Use home agent and foreign agent for the authentication process.

IV. METHODOLOGY

A hash function is a reproducible method of turning some

Komathi.P, II Year M.E (CSE), Department of Computer Science and Engineering, Annai Mathammal Sheela Engineering College, Affiliated to Anna University, Tamilnadu, India komathibe@yahoo.co.in

Balakumar.P, Assistant Professor, Department of Computer Science and Engineering, Annai Mathammal Sheela Engineering College, Affiliated to Anna University, Tamilnadu, India p_balakumar@yahoo.com

kind of data into a small number that may serve as a digital fingerprint of the data. The algorithm chops and mixes the data to create such fingerprints. The fingerprints are called hash sums, hash values, hash codes or simply hashes.

Timestamp can refer to a time code or to a digitally signed timestamp whose signer vouches for the existence of the signed document or content at the time given as part of the digital signature.

In cryptography, a public key certificate is an electronic document which incorporates a digital signature to bind together a public key with an identity — information such as the name of a person or an organization, their address, and so forth. The certificate can be used to verify that a public key belongs to an individual.

V. PROPOSED SCHEME

Assumptions:

We assume that the home agent *HA* and the foreign agent *FA,* pre-share a common secret key $K_{HF.}$

In the initialization phase, a new mobile user *MN* submits his/her identity ID_{MN} and a chosen password PW_{MN} to the home agent *HA* for registration. According to the submitted values ID_{MN} and PW_{MN}.

HA the performs following steps:

1. *HA* uses its private key *N* to generate by computing $r_1=h(ID_{MN} \| N)$where h() is a one-way hash function with the output sized 512 bits, e.g. SHA-512.
2. Compute $r_2=h(r_1 \| h(PW_{MN}))$.
3. Compute the hashing value h(N).
4. *HA* issues a smart card with { ID_{MN} , $ID_{HA,}$ $r_{1,}$ $r_{2,}$ h(N),h()} and delivers the smart card to *MN* through a secure channel.

When a mobile user *MN* wants to roam in a new foreign network. The foreign agent *FA* needs to authenticate *MN* through *MN*'s home agent *HA*. First *MN*, inserts his/her smart card into the device and keys in the password PW^*_{MN}. The smart card performs the following operations:

1. It generates the shadow SID of the user's identity by computing $SID=h\ (h(N)\| T_{MN}\ )$ EXOR ID_{MN} where T_{MN} is a timestamp contains the current date and time.
2. Compute $r^*_2 =h(r_1 \| h(PW^*_{MN}))$,$v1=r^*_2$ EXOR $h(r1\|T_{MN})$, and $v2= h(r_2$ EXOR $T_{MN})$.
3. Select a random integer *n* and generate the secret key x_0 by computing $x_0=h(n)$.
4. Compute $c=x_0$ EXOR $h(ID_{MN} \| T_{MN})$ and send *FA* the message $m1=(SID,V!,V2,T_{MN},ID_{HA})$.

After receiving the login request, *FA* first verifies the timestamp T_{MN} with the current date and time T'. If $(T'-T_{MN}) >=\Delta T$, where ΔT denotes the expected valid time interval for transmission delay, terminates the connection. Otherwise,

FA continues the following steps:

1. According to ID_{HA}, *FA* uses the common secret key K_{HF} , shared with H*A*, and the timestamp T_{FA} to generate s1,s2by computing $s1=T_{MN}$ EXOR $h(K_{FH} \| T_{FA})$,$s2=c$ EXOR $h(K_{FH} \| T_{FA})$.
2. Send *HA* a message m2={SID,v1,v2.s1,s2, T_{FA} }to verify whether the user *MN* is legal or not.

After receiving the message m2,*HA* verifies the timestamp T_{FA} first, if the timestamp is valid,

HA will continue the following authentication steps.

1. Compute $T_{MN}=s1$ EXOR $h(K_{FH} \| T_{FA})$ and $C=s2$ EXOR $h(K_{FH} \| T_{FA})$.
2. Obtain the user's identity ID_{MN} by computing ID_{MN} = SID EXOR $h(h(N)\| T_{MN}\ )$.
3. Verify the format of ID_{MN} If the format is not valid, *HA* terminates the connection.
4. Compute $r_1=h(ID_{MN} \| N)$ and $r^*_2 = r_1$ EXOR $h(r_1 \| T_{MN}\ )$.
5. Compute $v2^*=h(T_{MN}$ EXOR $r^*_{2\ )}$ and check whether $v2^*=v2$ or not. If they are equal, it means that the password PW^*_{MN} will equal to PW_{MN} The system executes the following steps. Otherwise, it sends *FA* a message for acknowledging that *MN* is an illegal user.
6. Obtain x_0 from C by computing x_0 =C EXOR $h(ID_{MN} \| T_{MN})$.
7. Generate the timestamp T_{HA} . Then, compute K= x_0 EXOR h ($K_{HF} \| T_{HA}$ *).* And send FA the message m3= {K, T_{HA} }for acknowledging that MN is a legal user.

VI. SYSTEM ARCHITECTURE

Access control system mainly consists of two sub systems:

1. Authentication server
2. Terminal proxy system.

1. AUTHENTICATION SERVER:

1.Identification terminal's operator service, which is a multiple thread service and uses certificate interface of ECA system and related authentication protocols to identify every operator who is try to logon in terminal systems. Identification terminal's operator service is activated by terminal proxy's operator identification service and turn into sleeping state as soon as there are connections attached to it.

2. Query operator identity status module, which is a multiple-thread process and whose goal is to finish querying the certificate status of every terminal's operator

Who has entered into terminal through the interface of certificate query system provided by ECA system. This module is activated by terminal proxy's event of requesting to query status of operator's identity. This module not only provides service of terminal's identity authentication but also can carry out query certificate status, which is provided for some operations required authentication.

3. Module of query privilege of terminal's operator, which is a multiple-thread process, and whose goal is to query privilege of terminal's operator through the interface of

privilege publishing service provided by ECA system. This module is triggered by the terminal proxy's event of request operation privilege query.

4. Module of writing login and operation log is a multiple-thread process to write down records of terminal user's login and operation. The log can be used to timely analyzing and monitoring and to afterwards query. This module is triggered by terminal proxy's event of request writing log. Log record is delivered by UDP segment and stored into log database server.

5. Module of operation log audit is to audit the behavior of terminal operator through some related rules. This module can alert illegal operation and can timely monitor terminal proxy.

6. Monitor procedure of terminal proxy: Supervises terminal proxy procedure' state to prevent terminal operators from intentionally or unconsciously killing the terminal proxy procedure through sending a special UDP segment to authentication server every certain interval. After reception of this segment, authentication server knows that the terminal proxy procedure sent this segment is alive.

7. Module of alerting illegal operation can raise an alarm to terminal proxy procedure to inform there exists illegal operations. As soon as terminal proxy receives an alert from authentication server, it analyzes the alert and carry out corresponding process: to locate corresponding terminal and lock it, then raise an alert to users, also can provide authentication server a supervising video channel.

8. Module of timely supervising terminal can monitor terminal's screen, which is a multiple-thread process to turn on supervising some terminal's screen. If it checks out some suspicious behavior, it can send out an alert, lock and interrupt corresponding terminals.

2 TERMINAL PROXY SYSTEMS:

1. IC card interface decode IC card's PIN code, lock IC card, query state and prevent user from pulsing and pulling card, etc. As terminal user want to login in, they are required to plus IC card into IC card interface and input PIN code of their IC cards. If users input right PIN code, terminal system read out operators' identity. If user continuously input wrong PIN code there times, the IC card will be lock. During operation, IC card should not be pull out, else terminal system will lock terminal's operating system.

2. Operator identity authentication is to authenticate terminal operators' identity. It is through communicating between authentication server and terminal proxy system. To identify terminal operators. Identity information of terminal operators is read from IC card prepared by ECA system for operators. During authentication, it also need help from the certificate interface of ECA system and related authenticating protocols and simultaneity query the state of terminal users' certification (through sending an event of requesting query state of operator's certification). As for the users who cannot pass the authentication, terminal system will lock terminal's operating system and not allow them to login in.

3. Requesting query state of certification aims to send an event to authentication server to activate the service of certification query, which will finish the query behavior.

4. Requesting operation privilege query is designed for querying operator's rights. As for terminal's operators, they are granting some rights to operate. Before they operate, they should pass operation privilege query first. If they have no corresponding rights, they are not allowed to do related operation. This module communicates with authentication server to carry out this operation privilege query function.

5. Module of locking and unlocking terminal system is a module to lock or unlock terminal's operating system. After receives an alert, this module can lock terminal operating system in the following cases:

(1) Users don't plus IC card.
(2) During operating, users pull out IC card.
(3) Users input wrong PIN code of IC card.
(4) Users cannot pass identity authentication.
(5) Illegal certification.
(6) Illegal operation.

During login, if users pass identity authentication and their certification is legal, terminal system will unlock the operating system.

6. Module of capturing users' operation information is a module to capture users' behavior such as keyboard input, opening a window as well as mouse input. These messages can form a part of terminal operation log.

7. Module of requesting to write log record is a module to deliver login log or operation log to authentication server which will store these records into its log database server.

8. Module of timely processing supervision picture is a module to turn on timely supervising the terminal's screen and transform picture of screen into video stream. It also has transport function. It cooperates with authentication server's timely supervising module to monitor terminal operator's behavior.

VII. CONCLUSION

This system proposes a new user authentication scheme with anonymity for grid computing environment networks.

The proposed scheme is not only simple but also secure. Besides, the scheme only uses low-cost functions; therefore, it can be executed very efficient and implemented easily on a mobile device in wireless environments.

This is more adaptive to the demand of accessing and controlling shared resource in grid computing environment.

VIII. REFERENCES

[1] I Foster, C Kesselman, G Tsudik, and S Tuecke, A Security Architecture for Computational Grids, in Proc 5th ACM Conference on Computer and Communications Security. 1998. p. 83-92.

[2] Howard Chivers, John A. Clark, and Susan Stepney, Smart Devices and Software Agents; the Basic of Good Behaviour, in Proceedings of the

first International Conference on Security in Pervasive Computing. 2003, Springer-Verlag: Boppard, Germany.

[3] L Pearlman, et al., A Community Authorisation Service for Group Collaboration, in Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks. 2002, IEEE.

[4] M Erdos and S Cantor, Shibboleth Architecture, Internet2, 8 October, 2001. http://middleware.internet2.edu/shibboleth/

[5]F. Azzedin and M. Maheswaran, "A Trust Brokering System and Its Application to Resource Management in Public Resource Grids", in Proceedings of IPDPS 2004.

[6]C. Lin, V. Varadharajan, Y. Wang and V. Pruthi, "Enhancing Grid Security with Trust Management", in Proceedings of Services Computing 2004 (SCC 2004).

[7]T.B. Quillinan, B.C. Clayton and S.N. Foley, "GridAdmin: Decentralising Grid administration Using Trust Management",in Proceedings of the ISPDC/HeteroPar'04, pp. 184–192.

[9]S. Tuecke, "Grid Security Infrastructure (GSI) Roadmap",Internet Draft, October 2000, http://www.gridforum.org/security/ggf1_2001-03/drafts/draft-ggf-gsi-roadmap-02.pdf.