A Novel Adaptive Proxy Certificates Management Scheme in Military Grid Environment*

Ying Liu, Jingbo Xia, and Jing Dai

Telecommunication Engineering Institute, Air Force Engineering University,
Xi'an, shanxi, 710077, P.R. China
yying_liu @126.com

Abstract. Proxy Certificates (PCs) is one of key mechanisms in Grid Security Infrastructure (GSI). Users need PCs to access grid service. But there is no effective mechanism to manage the PCs in GSI. In order to apply GSI in Military Grid, a novel adaptive Proxy Certificates management scheme is brought forward based on the hierarchical one-way hash chains. The hierarchical one-way chain consists of two or more levels of chains, where values of a first-level chain act as roots of a set of second-level chains and each PC is protected by a hash value, so the PCs' available time can be controlled adaptively and safely. The experimental results indicate that the scheme more adapt to the Military Grid environments.

Keywords: Military Grid, GSI, proxy certificate, hash value, hierarchical one-way chains.

1 Introduction

The Military Grid(MG) is a huge and complex undertaking that is intended to integrate virtually all of military information systems, services, and applications into one seamless, reliable, and secure network. MG's overall concept is to enable data access for a variety of systems and users in the network no matter which military service owns a weapon system or where a user might be located around the world. MG is designed to form the basis of a network-centric or "netcentric" way of fighting wars and to create a decisive advantage over adversaries. With such a large number of nodes unproven, MG require enhanced security mechanisms[1].

The Grid Security Infrastructure (GSI) [2] that is the portion of the Globus Toolkit [3] has been developed to support Grid environments, and widely used in Grid deployments worldwide. Our MG security Infrastructure is based on GSI. GSI uses public key cryptography (also known as asymmetric cryptography) as the basis for its functionality. GSI provides a delegation capability by using a proxy certificate that is derived from, and signed by, a normal X.509 Public Key [4] End Entity Certificate or by another proxy certificate for the purpose of providing restricted proxy and delegation within a PKI based authentication system. If a Grid computation

* This research is supported by Shaanxi Provincial Natural Science Foundation of China under Grant No. 2004F14.

K. Li et al. (Eds.): NPC 2007, LNCS 4672, pp. 23–30, 2007.
© IFIP International Federation for Information Processing 2007

requires that several Grid resources be used (each requiring mutual authentication), or if there is a need to have agents (local or remote) requesting services on behalf of a user, the need to re-enter the user's password can be avoided by creating a proxy. But there is no effective mechanism to manage the proxy certificates if GSI is implemented in MG. So, a new adaptive proxy certificates management scheme is brought forward in this paper. We focus on the problem to manage proxy certificates [5] available time and improve the success rate of Grid tasks.

The rest of this paper is organized as follows: Section 2 brief analyzes MG security platform and the existing proxy certificates mechanism. In Section 3 we present an adaptive proxy certificates management scheme based on the hierarchical one-way chains and describes it in detail. Experimental results and conclusions of the scheme are discussed in Section 4.

2 MG Security Platform Based on GSI

A central concept in GSI authentication is the certificates. Every user and service on the Grid is identified via a certificate, which contains information to identifying and authenticating the user or service. The GSI uses public key cryptography as the basis for its functionality. So the MG security platform is based on digital certificates. Six main components are described in Fig.1: Security Authenticate Sever Security Control Server, Security Management Server, Security Manager, User Register Database and User Access Database. The platform has the speciality of logical centralized management and physical distributed control and can ensure MG resources security. The communications in the security platform between severs and databases or users are transmitted by the encrypted safe channels.

This paper addresses the security problems of proxy certificates in MG. In GSI, a proxy consists of a new certificate and a private key. The key pair that is used for the

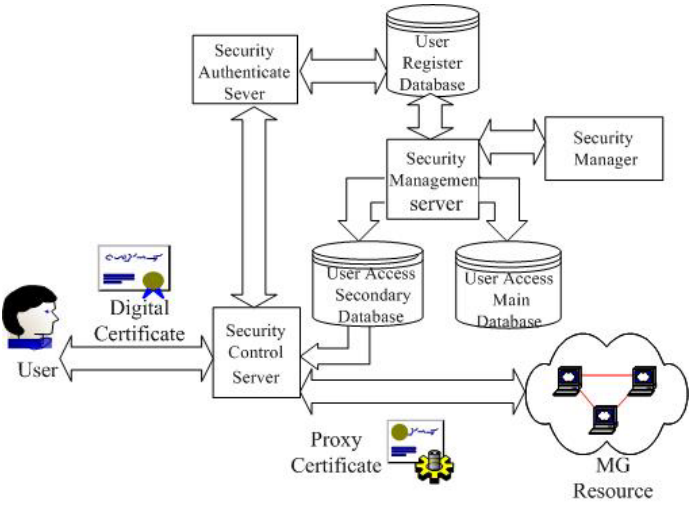

Fig. 1. MG security platform

proxy, i.e. the public key embedded in the certificate and the private key, may either be regenerated for each proxy or obtained by other means. The new certificate called proxy certificate contains the owner's identity, modified slightly to indicate that it is a proxy. The new certificate is signed by the owner, rather than a CA. The proxy certificate also includes a time notation after which the proxy should no longer be accepted by others. Proxy certificate have limited lifetimes.

When proxies are used, the mutual authentication process differs slightly. The remote party receives not only the proxy's certificate (signed by the owner), but also the owner's certificate. During mutual authentication, the owner's public key (obtained from his certificate) is used to validate the signature on the proxy certificate. The CA's public key is then used to validate the signature on the owner's certificate. This establishes a chain of trust from the CA to the proxy through the owner.

Existing proxy certificates scheme can achieve the requirement of dynamic delegation, dynamic entities and repeated authentication. But this approach has two drawbacks. First of all, in order to enable single sign-on, the proxy certificate's private key is stored in a local storage system without being encrypted. If an attacker can get the file by some ways, he may imitate the operation of the user. It may bring some security leaks. Secondly, GSI provides weak control on proxy certificates without a revocation mechanism. Available time of proxy certificates can not be dynamically changed by the owners of certificates or managers of the Grid. If over long available time of a proxy certificate is defined, it might bring some security problems because of the naked personal key of the proxy certificate. If over short available time is defined, it may be disabled while the task have not finished yet. As we know, it is difficult to forecast the completion time of a task correctly in a large-scale and dynamic MG environment.

3 Proxy Certificates Management Scheme

In this section we present a new adaptive proxy certificates framework in which the maximum lifetime of a certificate is divided into short periods and the certificates could expire at the end of any period under the control of the Grid user. Our scheme is based on Hierarchical one-way hash chains we proposed. One-way chains are an important cryptographic primitive in many security applications. As one-way chains are very efficient to verify, they recently became increasingly popular for designing security protocols. And the constructions for one-way hash chains is simple, a low-powered processors can compute a one-way function within milliseconds.

3.1 One-Way Hash Chains

Hash chains are based a public function $\boldsymbol{H}$ that is easy to compute but computationally infeasible to invert, for suitable definitions of “easy” and “infeasible”. Such functions are called one-way functions (OWF) and were first employed for use in login procedures by Needham [6]. If the output of a one-way function is of fixed length, it

is called a one-way hash function (OWHF). The definition of OWHF is given in [7] Definition:

A function $\boldsymbol{H}$ that maps bit strings, either of an arbitrary length or a predetermined length, to strings of a fixed length is an OWHF if it satisfies three additional properties

Given x, it is easy to compute $\boldsymbol{h}(x)$

Given $\boldsymbol{h}(x)$, it is hard to compute x

It is hard to find two values x and y such that $\boldsymbol{h}(x) = \boldsymbol{h}(y)$, but $x \neq y$.

A hash chain of length N is constructed by applying a one-way hash function $\boldsymbol{H}$ ( ) recursively to an initial seed value s.

$\boldsymbol{H}_N(s) = \boldsymbol{H}(\boldsymbol{H}(\boldsymbol{H}(...\boldsymbol{H}(s)...)))$ (N times)

The last element $\boldsymbol{H}_N(s)$ also called the tip T of the hash chain resembles the public key in public key cryptography i.e., by knowing $\boldsymbol{H}_N(s)$, $\boldsymbol{H}_{N-1}(s)$ can not be generated by those who do not know the value s, however given $\boldsymbol{H}_{N-1}(s)$, its correctness can be verified using $\boldsymbol{H}_N(s)$, This property of hash chains has directly evolved from the property of one-way hash functions.

3.2 Hierarchical One-Way Hash Chains

Here we bring forward hierarchical one-way Chains. A hierarchical one-way chain consists of two or more levels of chains, where values of a first-level chain act as roots of a set of second-level chains. The secondary chain rooted in the i^{th} value of the primary chain as the i^{th} secondary chain. Here, all the values of the i^{th} secondary chain are released before any of the values of the $i + 1^{st}$ chain is released; the primary chain value H_i is released in between. As Fig. 2 shows, to set up the hierarchical chain, the generator picks H_0 at random and computes the primary chain H_1, $H_1 \ldots H_n$, then choose $H_0, H_1, \ldots, H_n$ as the seed to generate n hash chains respectively.

3.3 Proposed Adaptive Proxy Certificates Scheme

We discuss the generation of proxy certificates from two aspects as follows: one condition is that Grid nodes have much better computing and communication capability,

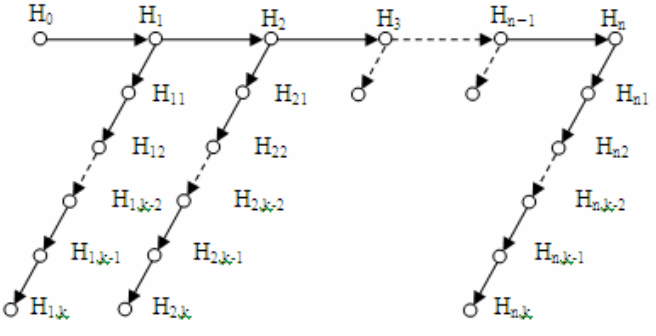

Fig. 2. Hierarchical one-way chains

such as personal computer, workstation and so on. The other is that the nodes have worse computation and communication capability, and can not afford the overhead of proxy certificate generation and management, Such as hand-held devices, wireless devices and so on.

In actual Grid environments, such classification is reasonable and indispensable. About the first condition, all operations of proxy certificates run on local nodes. A proxy certificates with an extensible expiry date could be generated in the following way:

(1) Forecast prospective proxy certificate path of length n, and generate first-level chains $H_0(r)\ldots H_n(r)$, $H_0(r)=r$ is a random value which is known to the user only.

(2) Generate the first proxy certificate.

(a) User U generate a pair of keys:

SK_P – private key

PK_P – public key

(b) Define the proxy certificate parameters:

T – maximum lifetime (the most common available time period of proxy certificates is 12 hours in traditional schemes, but we can revoke the proxy certificate at any time in our schemes, so proxy certificate's maximum lifetime can be long enough according to Grid application)

D – starting valid date

L – time period for refreshing validity of the proxy certificate

Suppose $k = T/L$ is an integer. The refreshing points are denoted as $D_1 = D+L$, $D_2 = D+2*L\ldots D_k = D+k*L$.

(c) Generate the first proxy certificate = $SIGN_U(U, PK_P, D, H_{n,k}(r), k, L)$ of n^{th} chain, then User signs this proxy certificate with his private key.

(d) Generate other proxy certificates of this chain. Like the first proxy certificate, generation process of others need not the user's the signature but the previous proxy's.

(e) Issue of the hash value. At the starting valid date of j^{th} proxy certificate that belong to this chain, U release $H_{j,k-1}(r)$ to initialize the validity of proxy certificate, which then has an expiry date $D = D_j + L$. Suppose the next refreshing point of proxy certificate is D_e. User will release $H_{ji}(r)$, where $i = k - (D_e - D)/L$. The value of T, D and L in each proxy certificate may be same, or different according to requirements. In our scheme, suppose they are same. Then, at each refreshing point, corresponding hash value should be released.

(3) Generate the other proxy certificate chains according to the Grid Tasks. This process like(2). After the entire Grid task have finished, a confirmation is return and all the proxy certificates' available time end automatically.

(4) Verification to proxy certificate. When a resource or another user V wants to verify this proxy certificate, it or he can take the following steps to check its status.

(a) V verifies the user's signature on $(U, PK_P, D, H_{jk}(r), k, L)$. If true, V is sure that proxy certificate's public key is PK_P. The starting valid date is D, the maximum lifetime is $T = k*L$, the refreshing time period is L, and the last hash value for j^{th} proxy certificate in the one-way hash chain is $H_{jk}(r)$.

(b) V checks that $0 \leq i < k$ and $H_{j,k-i}(H_{ji}(r)) = H_{jk}(r)$. If true, V believes that $H_{ji}(r)$ is a valid hash value in the one-way hash chain ended with $H_{jk}(r)$.

(c) V checks that $D \leq D + (k-i)*L$. If true, V concludes that proxy certificate is valid now, and remains valid until $D = D + (k-i)*L$. In such a way, U can control the validity of proxy certificate by releasing the corresponding $H_{ji}(r)$.

In the second condition, proxy certificates servers must be set up in Grids as Fig. 3.

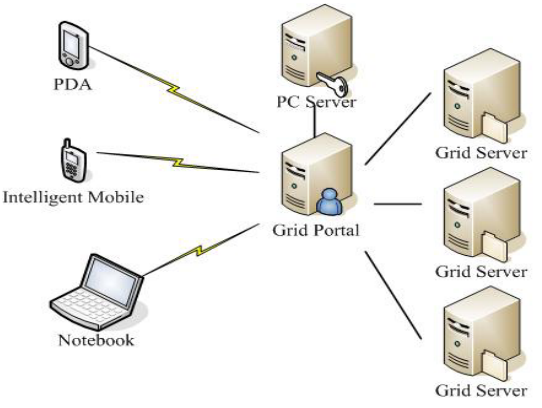

Fig. 3. Proxy certificates servers

The proxy certificates server may accept Grid nodes' request and replace all operations of the proxy certificates, include generation of the proxy certificates and the hash chain and issue of the hash value.

3.4 Protection of Hash Chain Root

For the first condition, the proxy certificate owner U relies on the hash chain root r to control the expiry date of the proxy certificate. There is an advantage on the use of a separate secret r to protect the private key SK_p. The system remains secure. The proxy certificates should also include an identifier of the hash function used to generate and verify the hash chain, as long as either r or SK_p is not compromised. If SK_p is compromised, U could destroy r then proxy certificates will expire shortly at the next refreshing point. Similarly, if r is compromised, U could destroy SK_p and stop using it. It might be at the same risk, however, if r and SK_p are stored in the same computer system. If the system is broken, both r and SK_p will be compromised. Then, a hacker holding r and SK_p can always generate valid signatures by refreshing the validity of the proxy certificate until its maximum lifetime T. Therefore we need to protect them separately or destroy r after memorize it.

For the condition that there are the proxy certificates servers, because all operations of the proxy certificates are completed by the proxy certificates server, the security of the server seems very important. So the servers must be managed by a trusted third party and has some security contracts with its users.

4 Experiments and Conclusions

The experiments were conducted in the MG platform of which each node provides the different types information in five military departments. The hardware configuration and operating systems used in the nodes are given in the Table 1. Grid middleware Globus Toolkit 4.0.3 is installed on each Grid node and the 100M switch and 100M transmission line are used in the network.

Table 1. The Small Grid environment

Name	Number	CPU	Memory	Operating system
LenovoE3 PC	$p_1 \ldots p_{100}$	Intel P4 2.8G Processor	512M	Win XP
IBM X455 Server	I_1, I_2	Intel Itanium 2 Processor 1.5G*4	56G	Win2003 Server
LenovoT630 Server	L_1, L_2, L_3	Intel Xeon Processor 2.0G *2	10G	Win2003 Server

In order to monitor the network flows, network flow monitoring software OnlineEye Pro V1.6.0 are installed on each server and ten personal computers. The information of five different military departments is saved on five servers respectively. A MG task is to inquire about the information simultaneously on one computer as a MG node in the certain department and record the every time. Let the maximum length of proxy certificates be 100, the largest life-cycle be one hour and the updating cycle be one minute. Our experiments select two situations as example: one is visiting I_1, I_2, L_1, L_2, L_3 directly by a random workstation, the other is visiting indirectly $p_2, \ldots p_{100}$ and I_1, I_2, L_1, L_2, L_3 by the agent on p_1.

The studies indicated that in the first situation when visiting the database directly, our improved the system could operate normally and the flows would not obviously changed, the running time only 0.01s; in the second one our system also could operate normally, but the flows increased a little but not evident, the running time 1.93s. Considerating the dynamic join of other nodes and network delay, the burden of computation and communication in our scheme can be ignored.

The paper addresses the security problems of GSI proxy certificates applied in MG. A novel adaptive proxy certificates management scheme is brought forward based on the hierarchical one-way hash chains. Every certificate is protected by a hash value, so the proxy certificate's available time can be controlled adaptively and safely. According to theory analysis and experiments result, it can improve the security of proxy certificates and the success percentage of MG tasks, and don't debase the performance of the system obviously. Our future work is to implement and test our schemes in large scale MG environments.

References

1. Buda, G., Choi, D., Graveman, R.F., Kubic, C.: Security standards for the global information Grid. In: Proceeeding of Military Communications Conference 2001. Communications for Network-Centric Operations: Creating the Information Force, vol. 1, pp. 617–621. IEEE Computer Society, Los Alamitos (2001)

2. Welch, V., Siebenlist, F., Foster, I.: Security for Grid Services. In: Twelfth International Symposium on High Performance Distributed Computing (HPDC-12), IEEE Press, Los Alamitos (2003)
3. Foster, I., Kesselman, C.: Globus: A Metacomputing Infrastructure Toolkit. Int. J. Supercomputer Applications, 115–129 (1997)
4. Kocher, P.: On Certificate Revocation and Validation. In: Hirschfeld, R. (ed.) FC 1998. LNCS, vol. 1465, pp. 172–177. Springer, Heidelberg (1998)
5. Tuecke, S., Welch, V., Engert, D.: Internet.509 Public Key Infrastructure Proxy Certificate Profile. In: IETF RFC 3280 (2004)
6. Wilkes, M.V.: Time-Sharing Computer Systems. Elsevier, New York (1972)
7. Berson, T.A., Gong, L., Lomas, T.M.A.: Secure, Keyed, and Collisionful Hash Functions. Technical Report. SRI-CSL-94-08. SRI International (1994)