A Lightweight Mutual Authentication Based on Proxy Certificate Trust List

Li Xin and Mizuhito Ogawa

Japan Advanced Institute of Science and Technology
1-1 Asahidai Tatsunokuchi Nomi Ishikawa, 923-1292 Japan
{li-xin, mizuhito}@jaist.ac.jp

Abstract. We propose Proxy Certificate Trust List (PCTL) to efficiently record delegation traces for grid computing. Our security solution based on PCTL provides functions as follows: (1) On-demand inquiries about real time delegation information of grid computing underway; (2) Lightweight mutual authentication that is beneficial for proxy nodes with limited computation power as wireless devices in mobile computing; (3) A kind of revocation mechanism for proxy certificates to improve the security and availability of grid computing.

Keywords: Grid Computing, Proxy Certificate, Mutual Authentication

1 Introduction

Proxy certificate (PC) is used in grid computing for securing private keys, delegation and single-sign-on[1] [2]. PC is issued by either grid users or grid proxys with limited life span. However there are some open problems. First, Grid Security Infrastructure (GSI) provides weak control on agents without a revocation mechanism for PCs. For instance, a PC may become invalid while computing is still underway due to network latency, underestimate, etc. Next, proxy certificate path verification is mechanically repeated in each mutual authentication, placing a heavy burden on agents to keep and exchange a long proxy certificate chain. Last but not least, grid participants often need to know the current delegation information, but GSI provides no means to do this.

In this paper, Proxy Certificate Trust List (PCTL) is proposed to partially solve these problems by providing on-demand delegation trace inquiries, lightweight mutual authentication, and a proxy certificate revocation mechanism.

2 Core Strategy for System Based on PCTL

2.1 Certificate Register Authority (CRA)

Our system is based on the existing Certificate Authority (CA)[3]. The additional and independent module is a trusted third party named the Certificate Register Authority (CRA). The main functions of the CRA are: (1) Maintain the trust

relations for PCs. (2) Respond to on-demand inquiries for detailed information about PCs and delegation traces. (3) Generate PCTL. (4) Revoke compromised or expired PCs. The data structure PNode is for recording PC information, as shown in Table 1. The information of End Entity Certificate (EEC), a standard X.509 certificate, is supplied directly from CA/LDAP servers. The IP address plus port number of the agent are contained in the PC's Relative Distinguished Name (RDN) to ensure unique names as well as active service from the CRA.

Table 1. Data Structure PNode for Proxy Certificate in CRA

Entry	Value
Index	Relative Distinguished Name + Certificate Serial Number
Delegation Depth	Permitted length of the delegation trace
Certificate Identifier_1	Hash code of proxy certificate
Certificate Identifier_2	Hash code of public key
Certificate Status	"Valid","Wait for Update","Invalid"
Validity	Life span of proxy certificate
Public Key	Public key of proxy certificate
Parent Pointer	Pointers to the issuer
Child Pointer	Pointers to all the issued grid proxys

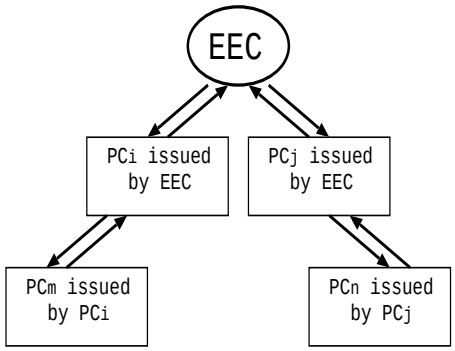
2.2 Definition of PCTL

PCTL records trusted delegation traces for grid computing. An n-ary dual-linked tree, TrustLogicTree, is constructed based on PNode to maintain delegation relations (Figure 1). PCTL records PC information on some trusted delegation trace with short life span, issuer information, security context, etc and is signed by CRA. The format for each PCTL entry differs at various security levels. An example of PCTL with a high security level is shown in Figure 2.

- High Level: An entry is a triplet (Index, Issuer, Certificate Identifier_1), each item of which corresponds to the definition in PNode. The hash of PC ensures the integrity of the whole certificate information.
- Middle Level: An entry is a triplet (Index, Issuer, Certificate Identifier_2). The hash of the PC's public key ensures the binding of the proxy name and its public key. Thus none can pretend to be another proxy by issuing PC with the same RDN.
- Low Level: An entry is a pair (Index, Issuer). It runs with the highest efficiency and benefits mobile computing with limited computation power.

2.3 Basic Algorithms

Register When delegation is needed, the issuer is required to register the new PC to the CRA after signing. The CRA will find the entry for the issuer by




```
Proxy Certificate Trust List (PCTL):
Signature Algorithm: md5WithRSAEncryption
Issuer: /C=Japan/ST=Ishikawa/O=Jaist/OU=School
        of Informaion Science/CN=CRA
Security Level: High
Last Update: June 1 16:20:30  2004 GMT
Next Update: June 2 00:20:30  2004 GMT

Subject: /C=Japan/ST=Ishikawa/O=Jaist/OU=School
        of Informaion Science/CN=LiXin
Serial Number: 18
Issuer: /C=Japan/ST=Ishikawa/O=Jaist/OU=School
        of Informaion Science/CN=CA

Subject: /C=Japan/ST=Ishikawa/O=Jaist/OU=School
        of Informaion Science/CN=LiXin/CN=Proxyi_150.65.200.100:80
Serial Number: 1
Issuer: /C=Japan/ST=Ishikawa/O=Jaist/OU=School
        of Informaion Science/CN=LiXin
Certificate Identifier(sha-1):
5D 4D 82 86 E1 02 AC CB 4F 07 8B 4D B3 3A BC 05 98 6D B4 04

Subject: /C=Japan/ST=Ishikawa/O=Jaist/OU=School
        of Information Science/CN=LiXin/CN=Proxyi_150.65.200.100:80
        /CN=Proxym_150.65.200.67:80
Serial Number: 2
Issuer: /C=Japan/ST=Ishikawa/O=Jaist/OU=School
        of Informaion Science/CN=LiXin/CN=Proxyi_150.65.200.100:80
Revocation Time: June 1 19:13:25  2004 GMT
Certificate Identifier(sha-1):
AD 47 19 A4 0D 7C BB 2D 93 33 64 B0 46 AF 69 22 6A 30 FB 85
......
Signature:
......
```

Fig. 1. N-ary dual-linked tree TrustLogic

Fig. 2. Example for PCTL in High Level

Index and verify the PC to be issued. If verification succeeds, CRA will create a corresponding PNode for the new PC and add it into the TrustLogicTree.

PCTL Acquisition Figure 3 shows a synchronous manner to get a PCTL when delegation and register are bounded together and the sequence order is preserved. Sequence (1)-(3) in Figure 4 shows an asynchronous manner where delegation and register are independent. It is a more lightweight handshake, but may require a timeout and retry if mutual authentication proceeds right after the delegation, that is, if an update of TrustLogicTree is later than a correlative PCTL use. Sequence (4)-(5) shows an on-demand inquiry for PCTL.

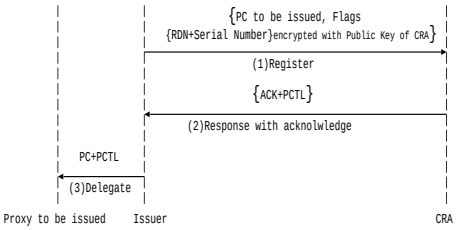


Fig. 3. Synchronous Message Sequence

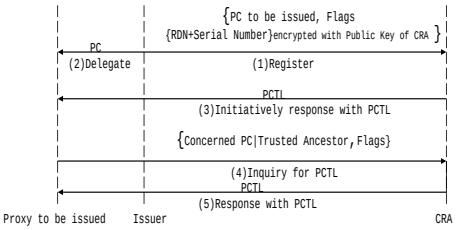


Fig. 4. Asynchronous Message Sequence

PCTL Generation The algorithm to generate PCTL is governed by "Flags" (Figures 3 and 4). If the concerned PC exists and is valid, CRA will generate PCTL. To improve availability, PCs with status "Wait for update" are also recorded in PCTL with revocation times, as in Figure 2.

– Flags=0 (Only the concerned PC is known): Find PNode for the concerned PC in CRA, and record all nodes whose status is "Valid" or "Wait for update" on the path between EEC and the concerned PC into PCTL.

– Flags=1 (Only the trusted ancestor of the concerned PC is known): Traverse the subtree rooted with the trusted ancestor by Depth-First-Search, and ignore the subtree rooted with PNode whose status is "Invalid". Then record all the nodes in the subtree into the PCTL.
– Flags=2 (Both the concerned PC and its trusted ancestor are known): Traverse the subtree rooted with the trusted ancestor by Depth-First-Search, then record all the nodes whose status is "Valid" or "Wait for update" on the path between the trusted ancestor and the concerned PC into PCTL.

Proxy Certificate Revocation (1) When some private key leaks, CRA will be notified to disable all the sub-trees rooted with the attacked PC by resetting all nodes' status from "Valid" to "Invalid". (2)When some PC expires, CRA does similarly to (1). The difference is only the expired PC will be disabled by resetting the status to "Wait for update" to improve availability.
Free Once an end entity finishes its task, CRA will release the subtree rooted with its EEC.

2.4 A Lightweight Mutual Authentication with PCTL

Let Proxy A and Proxy B be under a mutual authentication. Let PC_B be the PC of Proxy B. Let $PCTL\_B$=(Index, Issuer, CI) be the PCTL of Proxy B. Certificate verification with PCTL for Proxy A is shown briefly as follows: First, A decrypts $PCTL\_B$ with CRA's public key and check its validity. If it expired, A updates $PCTL\_B$ from CRA or asks B to provide a fresh one. After that,

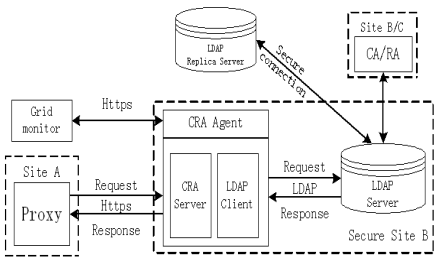
– High Level: Proxy A finds the entry for B by Index in $PCTL\_B$, and then computes the hash of PC_B and compares it with CI.
– Middle Level: Proxy A finds the entry for B by Index in $PCTL\_B$, and then computes the hash of B's public key and compares it with CI.
– Low Level: If there is an entry for Proxy B in $PCTL\_B$, Proxy B can be trusted without any computation.

3 Compatibility with GSI

Figure 5 shows the relationship between CRA and the current GSI system. To support PCTL, the required modification is kept to a minimum: (1) Additional negotiation is needed for SSL/TLS protocol when PCTL is enabled. (2) A new Object Class pkiProxyLDAP is needed for LDAP Schema [4] [5] (Figure 6).

4 Conclusions and Future Work

Our solution provides a "One-Time-Verification" on behalf of grid agents. A delegation tracing method was proposed in [6] by suggesting use of a ProxyCertInfo extension field. However this method can not reflet dynamic delegation changes. With the introduction of CRA, bottle-neck and single-point failure problems




```
pkiProxy OBJECT-CLASS ::={
SUBCLASS OF   {top}
KIND          auxiliary
MAY CONTAIN   {proxyIndex|proxyIdentifier_1|
              proxyIdentifier_2|proxyDepth|
              proxyStatus|proxyValidy|proxyIssuer|
              proxyParent|proxyChildren|proxyPubKey}
ID joint-iso-ccitt(2)ds(5)attributeType(4)pkiProxy(...)}

ProxyIndex        ATTRIBUTE ::={...}
proxyIdentifier_1 ATTRIBUTE ::={...}
proxyIdentifier_2 ATTRIBUTE ::={...}
proxyStatus       ATTRIBUTE ::={...}
proxyValidy       ATTRIBTUE ::={...}
......
```

Fig. 5. CRA implementation based on PKI **Fig. 6.** LDAP schema to support PC

need to be considered. Fault-tolerant techniques similar to those applied to CA can be used in a real implementation. Since in the current system agents might access CA for a Certificate Revocation List (CRL), the only additional overhead is the handshake with CRA in the register phrase, which doesn't take the time of computing in asynchronous manner. In our solution, certificate chain exchange can be avoided by exchanging a much smaller PCTL or by getting the PCTL itself. Certificate chain verification can also be avoided by simple hash manipulation. Assume L be the delegation depth and W the delegation width, the rough time cost of mutual authentication for the current system is $O(LW)$. So the advantages of our solution loom large when delegation is deep and frequent.

5 Acknowledgments

The authors thank Professor Kefei Chen (Shanghai Jiao Tong University) that work in this paper begin with his guidance, Li Qiang (Shanghai Jiao Tong University) for providing an openssl platform to run examples, and Professor Yasushi Inoguchi (JAIST) for his helpful comments. This research is supported by Special Coordination Funds for Promoting Science and Technology by Ministry of Education, Culture, Sports, Science and Technology.

References

1. I. Foster, et al. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Supercomputer Applications, 15(3), 2001.
2. V. Welch, et al. Security for Grid Services. *Twelfth International Symposium on High Performance Distributed Computing (HPDC-12)*, PP.48-57, 2003.
3. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. RFC 2459, 1999.
4. Internet X.509 Public Key Infrastructure LDAPv2 Schema. RFC 2587, 1999.
5. Internet X.509 Public Key Infrastructure LDAP Schema and Syntaxes for PKIs. draft-ietf-pkix-ldap-pki-schema-00.txt, 2002.
6. V. Welch, et al. X.509 Proxy Certificates for Dynamic Delegation. *3rd Annual PKI R&D Workshop*, 2004.