Secure Information Sharing*

Ganesh Godavari, and Edward Chow
Department of Computer Science, University of Colorado, 1420 Austin Bluffs Parkway
Colorado Springs, Colorado 80917 USA
gkgodava@cs.uccs.edu, chow@cs.uccs.edu

1 Introduction

As the WWW is fast becoming a place for sharing of information, piracy, and misuse of information are fast becoming a threat. Security and Authorization are becoming necessary. This situation not only provides excellent business opportunities but also research challenges. One of the most challenging problems is managing large information sharing systems is the complexity of security administration, particularly digital certificate and role based access control.

Role Based Access Control simplifies access control administration and provides better manageability in enterprise environments by allowing permissions to be managed in terms of user job roles [8]. RBAC maps user job roles to application permissions so that access control administration can be accomplished in terms of users job role. This means that administrators will have to set up and assign clients with roles, such as employee, manager and administrator, with out having to change the access permission on each object.

Many of the e-commerce applications require authentication services, in addition to the basic services provided by Public Key Infrastructures (PKI), to allow users to do what they are allowed to do. Authentication means that that the sender of a message or transaction is verified to be who they claim to be, while Authorization means that someone who has the authority to do, so he/she can initiate or progress a transaction, process, or activity. In simple terms, Authentication is what is required to gain access e.g., a passport, driving license or in computing terms suchas password, strong digital certificate authentication. Authorization deals with what you are permitted to do, once you are authenticated. Public key certificate (PKC) strongly binds a public key to its subject (country, location, organization unit etc.) helping to identify the holder of the certificate. Attribute certificates have been proposed as a solution for the authorization services. The Attribute certificates are designed to convey (potentially short-lived) attributes about a given subject to facilitate flexible and scalable privilege management. The attribute certificate may point to a public-key certificate that can be used to authenticate the identity of the attribute certificate holder.

Some research and development efforts have been done in this area [2, 15, 6], but these efforts are still in primary phase, and no authorization mechanism is widely accepted. We were motivated by the need of using PKI, PMI and RBAC concepts to construct an authorization mechanism which uses the PERMIS [1, 2] model of storing the user's roles in ACs. Access control decisions are driven by an authorization policy, and the authorization policy is also stored in an AC.

The organization of this paper is as follows. Section 2 provides an overview of the related research. Section 3 describes the design and implementation of the secure information sharing (sis) system architecture. In Section 4 we present the sis prototype and analysis of its performance results. In Section 5, and 6 we discuss future directions of research and conclusions.

*This research work was supported in part by a NISSC AFOSR Grant award under agreement number F49620-03-1-0207.

2 Related Research Technologies

2.1 Role Based Access Control

Role-based access control [4, 14, 12, 16, 3] has gained attention as a proven alternative to traditional discretionary and mandatory access control mechanisms. RBAC helps specify organization's security policies reflecting its organizational structure. In the RBAC, a user can be assigned one or more roles, and a role can be assigned to one or more users. roles are based on the user's job responsibilities in the organization. This provides for flexibility and finer granularity during assignment of access permissions to roles and users to roles. In role-based model the role hierarchy partially determines which roles and permissions are available to users via various inheritance . For example, a senior role can inherit permissions from junior roles. A user establishes a session during which he activates some subset of roles that he is a member of. RBAC provides static seperation of duty relations to prevent conflict of interests that arise when a user gains permissions associated with conflicting roles, and Dynamic Seperation of Duty relations to place constraints on roles that can be activated in a users session.

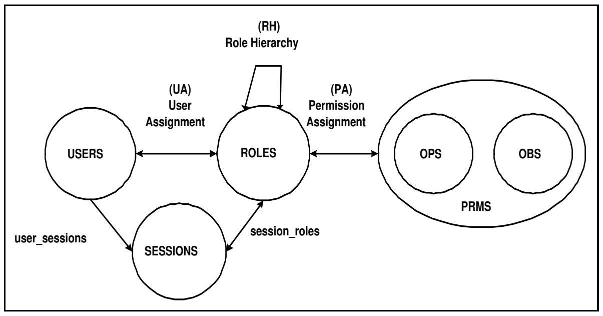

Figure 1: RBAC model[8]

The RBAC in Figure 1 consists of 1) a set of users (USERS) where a user is an intelligent autonomous agent, 2) a set of roles (ROLES) where a role is a job function, 3) a set of objects (OBS) where an object is an entity that contains or receives information, 4) a set of operations (OPS) where an operation is an executable image of a program, and 5) a set of permissions (PRMS) where a permission is an approval to perform an operation on objects. The cardinalities of the relationships are indicated by the absence (denoting one) or presence of arrows (denoting many) on the corresponding associations. For example, the association of user to session is one-to-many. All other associations shown in the figure are many-to-many. The association labeled Role Hierarchy defines the inheritance relationship among roles.

Further information about RBAC is available at [8].

2.2 Privilege Management Infrastructure (PMI)

PMI is the information security infrastructure that assigns privilege attribute information such as privilege, capability, and role, etc., to users, and issues and manages it using the X.509 Attribute Certificate. Attribute Certificates (ACs) were initially introduced in Recommendation X.509 Ũ 97, but they were fully covered in Recommendation X.509 - 2000 published in year 2001. The PKIX WG from the IETF has endorsed a profile of attribute certificates in April 2002 with the RFC 3281. The PMI supports access control service using the user's privilege management in application services. The function of the PMI is to specify the policy for the attribute certificate issuance and management. Then, the PMI carries out the AC-related management functions such as issuing, updating, and revoking an attribute certificate based on a specified policy.

In PMI the ACs issuer is called Attribute Authority (AA). ACs are digitally signed by the AA, so they are tamper-resistant. The trusted root is called source of authority (SOA). When a user's authorization permissions need to be revoked, AA will issue an attribute certificate revocation list (ACRL) containing the list of ACs no long to be trusted. There are two primary models for distribution of attribute certificates: the 'push' or 'pull' model. In the push model the client needs to present its AC to the server, so the client needs to contains AC. The 'push' model is suitable in application where the client's permissions should be authenticated/validated in the client's 'home' domain. In the 'pull' model the client presents its identity to the server, the server retrieves the client's AC from a repository. Therefore the client doesnot contain AC. The 'pull' model is suitable when the client's privelages should be authenticated in the inter-domain. In the 'pull' model a change in the organizations role permissions only requires an update in the repository, while in the push model clients with the same role need to update their AC.

Figure 2 below shows the difference between PKC's, and AC's. PKC binds a subject(DN) to a public key while AC's have no Public Key but binds permission (attributes) to an entity.

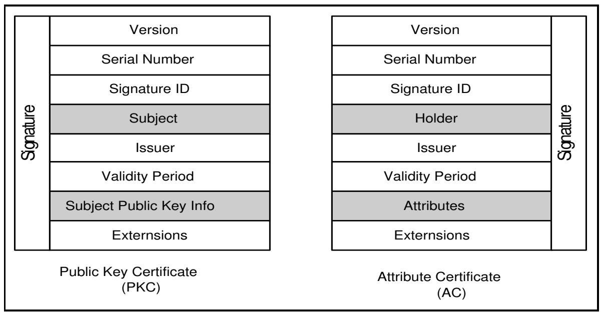

Figure 2: PKC and Attribute certificates

Further information about AC is available at [13]

3 Design and Implementation of SIS System

3.1 Design Considerations

Although the concept of role-based access control (RBAC) began 25 years ago, It gained wide spread interest in 90's. A study by NIST [12] on 28 organizations revealed that RBAC addresses many needs of the commercial and government sectors. In this study of 28 organizations it was found that many organizations based access control decisions on the roles that individual users take on as part of the organization and also found that permissions assigned to roles tend to change relatively slowly compared to changes in user membership of roles. With RBAC it is possible to predefine role-permission relationships, which makes it simple to assign users to the predefined roles.

Since access control mechanism is crucial in enforcing and tracking secure information distribution and traditional discretionary and mandatory access control are too restricted, we have investigated RBAC, which provides flexibility and allows dynamic update. National Institute of Standards and Technology (NIST) has recently ratified RBAC draft into a standard. RABC is currently being used in various database management systems like Sybase and J2EE/Java servers.

The central notion of RBAC is that permissions are associated with roles, and users are assigned to appropriate roles. This greatly simplifies management of permissions. Roles are created for the various job functions in an organization and users are assigned roles based on their responsibilities and qualifications. Users can be easily reassigned from one role to another. Roles can be granted new permissions as new applications and systems are incorporated, and permissions can be revoked from roles as needed.

RBAC model can be used for interaction between organizations are planning to coordinate and share information in entirety or partwise. Some of the challenges faced in ensuring cooperation are

- Confidentiality: Information available at the organizations are confidential and should not be shared with people outside the organization(s). Access to such information has to be restricted to a selected group of people with in the organizations that are involved in the cooperation. This problem becomes hard if the roles of the people outside the organization are not defined properly.
- Non-Repudiation: The shared information may changed by people belonging to various organization. Changes made must be monitored to ensure reliability of the information, along with the ability to provide non-repudiation service for changes made to the shared information.
- Decentralized maintenance and control: Information shared by each organizational should be managed and maintained by that organization. This helps not only to remove the disputes raised by questions like "who is responsibile for what ?", but also simplifies the maintainance of information. If USERA of Organization-A wants to access information from organization-B, then organization-B is responsible for providing a certificate to USERA for authentication and autherization. These certificates are stored at Organization-A along with other information about USERA in the LDAP server. This might cause the user to be overwelmed by number of certificates he needs to maintain; one for each organization involved in the coordination.

3.2 Establishing PKI for SIS

When a task force is formed with multiple agencies or organizations, the first taks is to issue a valid certificates to the members of the task force. User certificate mainatanence problem can be avoided, if all the organizations participating in the information sharing service have the same rootCA. The following steps outline the procedure for setting up a PKI at his/her site.

1. The coordinator of the task force of this multiple agencies set up a rootCA-MA (root CA for Multiple Agencies)
2. Each organization requests a certificate to be signed by rootCA-MA
3. Each organization issues a new PKC to each user in its organization involved in the taskforce
4. At each server providing secure information sharing service for this task force, add the rootCA-MA information into CABundle (file containing list of valid CA's)
5. Each client/user installs the certificate in the local browser or application(s)

With a large task force, it takes a long time for the PKI to be established. we developed a tool for creating certificates. We compare the performance of this automated tool with that of manual signing process. The automated tool uses expect library and openssl to sign the certificates. Automated tool can sign 100 certificates in 2 min and 13 sec on a pentium-III machine with 512 MHz, 512 mb RAM configuration. Based on my own person experience generating one certificate manually takes about 2 min 35 sec. In a joint task force with 100 organizations and 100 participants from each organization coordinating, in the worst case the time taken in the manual process is 516 min and 40 sec, while time taken in an automated process = 2 min and 14.33 sec.

We assume that security incidents adding new users, breach of confidentiality etc are resolved either through a third-party or by talks between the various organizations. This is outside the scope of this paper.

3.2.1 Organizational Information Sharing System Overview

Our access control system is designed to support RBAC using X.509 PKIs and ACs. The authentication is implemented by PKI, and the authorization is implemented by AC. Role information is stored in User

Role Specification AC's (see section 'Administration tool'). All the access control decisions are made based on authorization policies. They are written in XML and stored in ACs. ACs and their corresponding PKIs are all stored in LDAP servers [9]. In our current prototype implementation we have a simple RBAC policy specification file as shown in Table 1.

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<!-======= SIS rbac parsing example =======->
<SIS>
    <userRoleSpecification>
        <Role>Teller</Role>
        <Group>Info Share</Group>
        <OU>banking</OU>
    </userRoleSpecification>
</SIS>
```

Table 1: Sample RBAC File Format

SIS system consists of the following components

- *Administration Tool:* is used for creating key pair, PKIs, User Role Specification ACs.
- *RBAC Policy file:* specify the roles and what privileges the role can have on the resources. Access control decision are made based on these privileges. This information is stored in AC generated using administrative tool.
- *Ldap Server:* stores the user's information along with User Role Specification ACs and Delegated Role Specification AC's.
- *Access Control Decision and Enforcement:* executes the function of authorization and informs the target if the user has the privileges or not.
- *Resources:* they may be web servers, database servers, or any other format of resources.

Figure 3 shows the interaction of the various SIS components. The administrative tool generates X509 certificates, and User Role Specification AC for the users participating in the secure information sharing service. The RBAC policy file passed to the administrative tool is used for embedding the policy information in the AC. The AC generated by the administrative tool are stored in the LDAP server along with other information about the user. The user X509 certificates are installed in the his/her client application. Access Control Decision and Enforcement (ACDE) engine makes sure that the user has the required authorization privelages to access the resource. The user needs to submit his/her X509 certificate to the ACDE in order to access a resource. The ACDE verifies the user X509 certificate, and queries the LDAP server for the user AC. If the user has the required privilages to access the resource, access permission is granted.

3.3 Mapping Role Hierarchy to permissions

Mapping of Role Hierarchy of the organization to permissions for directory access is critical for enabling information sharing and providing access restrictions. Figure 6 shows the mapping of user roles to directory access Permissions. USER3 has access permissions to USER3, USER2 and USER1 directories. The organizational role hierarchy information is contained in the configuration file of the apache. The module uses the

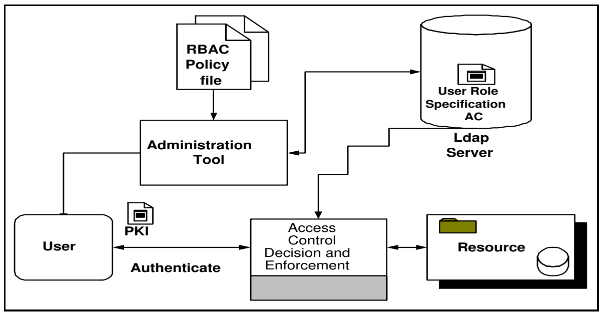

Figure 3: Interaction between various SIS componenets

Role information in the user AC along with role hierarchy information, to determine the access permissions to the requested web document. Figure 4 shows the mapping of role hierarchy to access permissions.

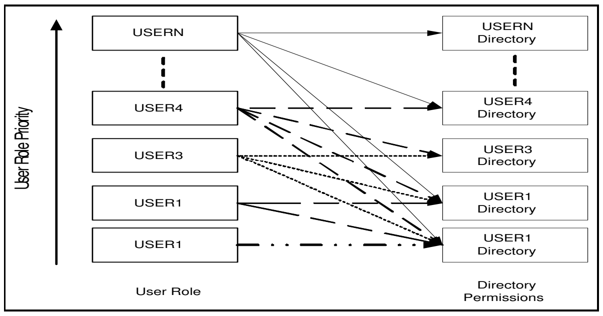

Figure 4: Mapping Role Hierarchy to Permissions

3.4 Administration tool

OpenSSL [10] provides strong open source cryptographic library for X509, and SSL&TLS. Currently OpenSSL does not have any support for AC except for RFC 3281 AC's ASN.1 object definitions. We wrote the code for AC generation using the crypto library and ASN.1 object definitions in OpenSSL. There are two types of Attribute Certificates in our proposed architecture system:

1. 'User Role Specification' attribute certificate which tells what privilege(s) a user has. It is used by the decision making service to make a decision to determine whether a user has access information or resources available in application services.
2. 'Delegated Role Specification' attribute certificate which tells what privileges are given for a resource(s) by a user of higher authority.

In the User Role Specification AC the issuer, and signature values belong to that of Attribute Authority. In Delegated Role Specification AC the issuer and signature value belongs to the user who delegated the authority. To issue User Role Specification AC, the tool needs the AA's certificate and key, and user's certificate along with RBAC policy file. To issue Delegated Role Specification AC, the tool needs the delegating user's certificate and key, and user's certificate along with RBAC policy file specifying the authority delegated.

In our prototype we adopt AC ‘Pull’ model, so the role ACs are not given to users. The ‘User Role specification’ and ‘Delegated Role Specification’ ACs are all stored in LDAP servers.

3.5 Access Permissions specification Format

Currently there are no XML parsers that can parse XML data fast. This can create a bottleneck if the XML file are large. We can optimize parsing by parsing XML tags that are of interest to us. In the XML document shown below, some of the tags are repeated, e.g., Role, Group, OU. Hence, a rule syntax is needed to allow for selecting a particular set of tags in the rule set. Here is an example of a scheme that addresses this problem. To specify a rule based on Group value present in the second item tag within the first userRoleSpecification tag, the rule will be specified as ’sis:1.userRoleSpecification:2.OU’. As another example, ’sis:1.userRoleSpecification:1:Group’ specifies a rule based on the Group tag present within the first UserRoleSpecification tag in the first sis tag. we use this method of representation for specifying access permissions.

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<!–======= SIS rbac parsing example =======–>
<SIS>
    <userRoleSpecification>
        <Role>Manager</Role>
        <Group>communication</Group>
        <OU>Info Share</OU>
    </userRoleSpecification>

    <userRoleSpecification>
        <Role>TeamLeader</Role>
        <Group>communication</Group>
        <OU>banking</OU>
    </userRoleSpecification>
</SIS>
```

Table 2: Access Permissions specification Format

3.6 Information Sharing among Multiple Agencies

We implemented the Access Control Decision and Enforcement (ACDE) engine as an Apache [5] module. It is responsible for providing the authorization service for web requests between user and the requested target file(s). This framework seperates authentication service, provided by ModSSL [7] from authorization service. Our prototype consists of following components: Initiator (e.g. a browser), Target (web server), Access Control Decision and Enforcement (ACDE) provided by the apache_sis_module. Figure 5 shows the control flow in ACDE engine. Figure 6 shows the message flow between these components.

The initiator submits secure web access request to the web server. Through SSL protocol the client and the web server are mutually authenticated through the exchange of their digital certificates. From the uri of the submitted HTTP request, the web server identifies that the corresponding directory contains special access control as specified in the httpd.conf configuration file. The corresponding <directory>section of

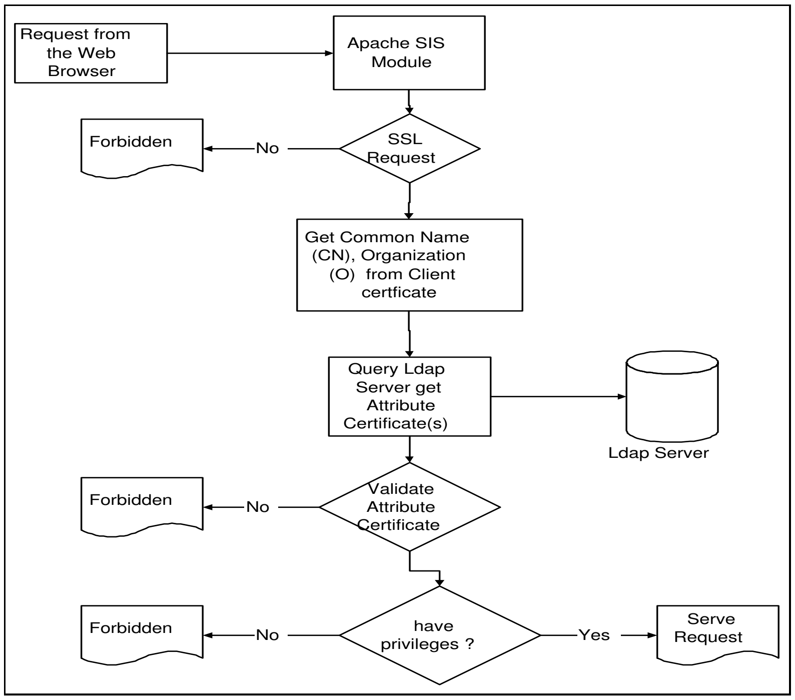
Request from the Web Browser
Apache SIS Module
Forbidden
No
SSL Request
Get Common Name (CN), Organization (O) from Client certficate
Query Ldap Server get Attribute Certificate(s)
Ldap Server
Forbidden
No
Validate Attribute Certificate
Forbidden
No
have privileges ?
Yes
Serve Request

Figure 5: Control Flow in Access Control Decision and Enforcement engine

the httpd.conf file contain the information port number, bind password of the LDAP server where the web server queries for the authorized permission on behalf of the client. Besides the LDAP server location, it also includes the specification of the access right associated with the role which will be contained in the LDAP query result.

The web server extracts the client information from the subject field of the client certificate and submit them in the LDAP query request to the LDAP sever. The LDAP server retrieves the corresponding client's attribute certificate and returns it to the web server. The web server translates the role containing in the attribute certificate into the related access right. If the command contains in the HTTP request does not match with the resulting accessing right, the request will be rejected. For example, if this is a POST command while the role of the client only has read access right, the request will be rejected.

4 Experimental results

In this section, we present the details of our prototype and its performance.

4.1 Prototype implementation

We developed a Secure Information Sharing system prototype/testbed for multiple agencies using web based approach. Authentication was provided for Apache (v 1.3.31) web server using third party module Mod_SSL (v 2.8.18-1.3.31), which uses OpenSSL (v 0.9.7d) package for providing SSL & TLS. The Web server is configured to validate the clients, by requesting for client certificates. LDAP module [apacheldap] for Apache was enhanced to provide ACDE functionality. Attribute Certificate's attribute definitions was added to inetorgperson.schema in OpenLDAP (v 2.0.27-8) [11]. attributeCertificateAttribute attribute is added to

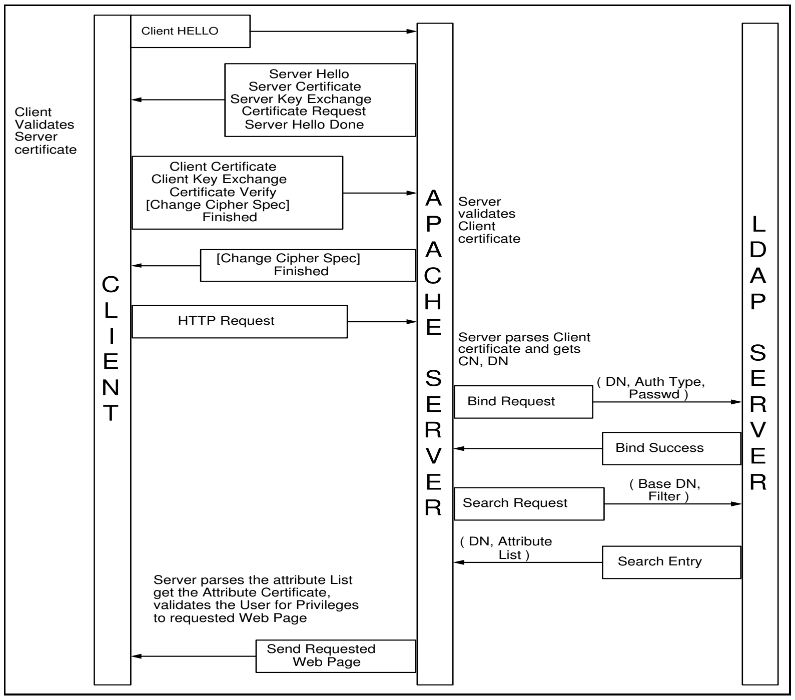

Figure 6: Message Flow between the components

inetOrgPerson objectclass in inetorgperson.schema file. Table 3 below shows the AC attribute definition. OpenSSL libraries were also used for generating X509 certificates.

```
# attribute certificate attribute definition
attributetype (2.5.4.58 NAME 'attributeCertificateAttribute'
   DESC 'A binary attribute certificate'
   EQUALITY octetStringMatch
   SYNTAX 1.3.6.1.4.1.1466.115.121.1.8 )
```

Table 3: attributeCertificateAttribute definition [11]

For example, We use the following command to add the attribute certificate to the enhanced LDAP server.

```
ldapadd -xv -D "cn=manager,dc=sis-nissc,dc=edu" -W -f entries.ldif -h hostDNSname
#ldapsearch -C -b "cn=alpha-sis-nissc,ou=Research,ou=coordinationExercise,dc=sis-nissc,dc=edu" -x "(ob-
jectclass=*)"
```

The entries.ldif is a file that contains the LDAP entry in Lightweight Directory Inter-exchange Format. -D option specifies the root DN of the LDAP server. -x specifies the use of simple authentication instead of SASL. -W is used to prompt the password. -v for the verbose mode. The commented ldapsearch command can be used to verify if the entry has been added to the LDAP server.

Below we show part of the entries.ldif content. It includes the description of the attribute certificate of a participant named alpha-sis-niss. The attribueCertificateAttribute specifies the AC file created by Attribute Authority or its delegates using SIS administration tool described in Section 3.4.

```
dn: cn=alpha-sis-nissc, ou=Research, ou=coordinationExercise, dc=sis-nissc, dc=edu
ou: Research
ou: coordinationExercise
cn: alpha-sis-nissc
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson mail: alpha-sis-nissc@sis-nissc.csnet.edu
givenname: Alpha
sn: alpha
homePostalAddress: ENG 142
l: Colorado Springs
st: CO
postalcode: 80920
telephoneNumber: (800)777-1212
homePhone: 800-555-3334
mobile: 800-555-1348
title: Team Leader
facsimileTelephoneNumber: 800-555-3344
uid: alpha
userPassword: help
attributeCertificateAttribute;binary:<file:///root/alpha-sis-nissc.pem
```

4.2 Experimental setup

We set up a testbed to simulate coordination between 4 different agencies. The four different agencies shared a similar Directory Information Tree (DIT) shown in Figure 7. Every SIS node has OpenLDAP and Apache webserver with sismodule running on them. The operating systems are Linux Redhat 8.0, 9.0. Netscape and Internet Explorer browsers were used as clients. Each SIS node is a HP Kayak machine (PII 233MHz, 96MB RAM, 10 Mb Ethernet connection).

Figure 8 shows the multi-agency testbed. The following steps take place for a user (alpha-sis-canada) in organization sis-canada.csnet.edu to retrieve a web document from a web server located in sis-nissc.csnet.edu

1. alpha-sis-canada sends a secure web request to sis-nissc.csnet.edu. sis-nissc.csnet.edu validates alpha-sis-canada's certificate for authentication.
2. If the certificate is valid, sis-nissc.csnet.edu uses the subject (DN) in the certificate, to establish connection with the LDAP server on sis-canada, and retrieves his AC.
3. sis-nissc.csnet.edu validates alpha-sis-canada's AC and checks if alpha-sis-canada has the right privilages. The coordinationExercise group is allowed to retrieve data from sis-nissc.csnet.edu
4. sis-nissc.csnet.edu returns web documents back to alpha-sis-canada using the secure communication channel.

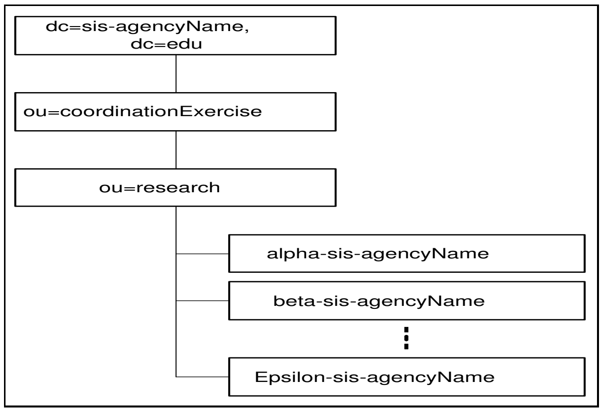

Figure 7: LDAP DIT Format at each agency

4.3 Performance Results

Table 4 show the performance results of sis-module multiple agency scenario.

client-server	Total time taken for LDAP access (ms)	Total Time taken for Attribute certificate retrieval and validation (ms)
sis-canada-sis-nissc.csnet.edu	54.623001	96.885002
sis-canada-sis-connecticut.csnet.edu	51.845001	93.778999
sis-canada-sis-newjersy.csnet.edu	51.191002	93.310997

Table 4: Performance Results in a multiple agency scenario

5 Future Directions

We have a design and implementation of multi-agency information sharing system using web based approach. Policy specification is critical in a information sharing environment. We need to evaluate if and how a policy specification language like eXtensible Access Control Markup Language, can be used for specifying RBAC policies. We also need to look into how these policies can be defined so that no one policy governs all organizations involved in information sharing service. We also need to look into how Digital Rights and Attribute Certificates can work together for monitoring malicious activities on files that can be downloaded by the client.

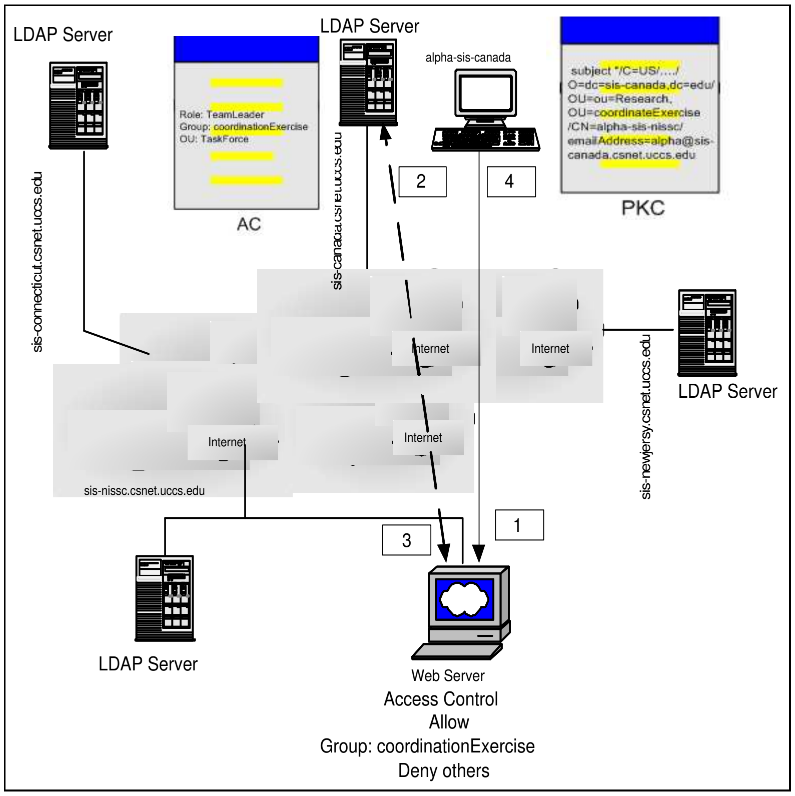

Figure 8: multi-agency prototype testbed of SIS

References

[1] Chadwick, D. W., and Otenko, A. Rbac policies in xml for x.509 based privilege management. In *Int. Conf. On Information Security* (2002), pp. 39–53.

[2] Chadwick, D. W., and Otenko, A. The permis x.509 role based privilege management infrastructure. *Future Gener. Comput. Syst. 19*, 2 (2003), 277–289.

[3] D. Ferraiolo, J. C., and Kuhn, D. R. Role-based access control: Features and motivations. *Computer Security Applications Conference* (1995), 241.248.

[4] Ferraiolo, D., and Kuhn., D. R. Role-based access control. *15th NIST-NCSC National Computer Security Conference* (1992), 554–563.

[5] Jakarta. http://www.apache.org/, 2004.

[6] Javier Lopez, Antonio Mana, J. J. O. J. M. T., and Yague, M. I. Integrating pmi services in corba applications. *Comput. Stand. Interfaces 25*, 4 (2003), 391–409.

[7] ModSSL. http://www.modssl.org/, 2004.

[8] NIST. Role-based access control, 2004.

[9] OpenLDAP. The open source lightweight directory access protocol (ldap), 2004.

[10] OpenSSL. The open source toolkit for ssl/tls., 2004.

[11] PERMIS. http://www.permis.org/, 2004.

[12] R. Sandhu, E. J. Coyne, H. L. F., and Youman., C. E. Role-based access control models. *IEEE Computer 29* (1996), 38–47.

[13] S. Farrell, R. H. An internet attribute certificate profile for authorization, 2002.

[14] Sterne., D. F. A tcb subset for integrity and role-based access control. *15th NIST-NCSC National Computer Security Conference NIST/NSA* (1992).

[15] Thompson, M. R., Essiari, A., and Mudumbai, S. Certificate-based authorization policy in a pki environment. *ACM Trans. Inf. Syst. Secur. 6*, 4 (2003), 566–588.

[16] von Solms, S. H., and van der Merwe, I. The management of computer security profiles using a role-oriented approach. *Comput. Secur. 13*, 9 (1994), 673–680.