Secure Information Sharing[*]
A Final Report submitted to
Network Information and Space Security Center (NISSC)
for Spring 2004 Sponsored Project

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 Project Goal

The objective of the proposal is to examine the existing security models for supporting information sharing among multiple agencies and suggest extension for addressing the new requirement. Procedures for enforcing and tracking secure information distribution will be investigated. Existing group key management will be enhanced to address the scaling and tracking issues. To validate the revised security model and the secure information distribution procedures, a secure information distribution prototype based on SGFR secure instant messaging software we have developed in NISSC sponsored project will be developed. It will help identify the critical features and system design issues. Field trials with Northcom will be attempted in the follow-up project.

Extensive literature survey related to secure information sharing was performed. We examined existing security models and found that role based access control (RBAC) provide flexibility and extensibility for supporting the information sharing among multiple agencies [9]. To have finer granularity of sharing, existing public key certificate can be enhanced with the attribute certificate mechanism [14]. Attribute certificate binds users with specific resource access permission. we have established a role based access control infrastructure and developed a prototype that uses X.509 public key certificates (PKCs) and attribute certificates (ACs). We explore the use of Attribute Certificates with RBAC for supporting large scale secure information sharing. We use Ldap servers for storing ACs, PKC to provide authentication and authorization services for Web services. The scheme proposed has many advantages that satisfy the needs of inter-organization information sharing using attribute certification.

In the paper [12] by Charles phillips et al, information sharing and security in dynamic coalitions is a complex task, which manifests itself throughout the lifetime of the coalition. The critical issues that arise during a coalitionŠs formation, and in support of its day-to-day management and usage, include, but are not limited to the following:

- Federate groups of users quickly and dynamically in response to a crisis.
- Bring together resources (e.g., COTS, databases,legacy systems, etc.) without modification for usage in support of the crisis.
- Dynamically realize and manage a security policy during simultaneous crises.

[*] Draft Report 0.1

- Identify users by their roles to finely tune their access in support of a crisis.
- Authorize, authenticate, and enforce a scalable security policy that can be managed and changed in response to the needs of the coalition.
- Provide a distributed security solution in support of DCP that is portable, extensible, and redundant for survivability.
- Offer robust security policy definition, management, and introspection capabilities that are able to track and monitor system behavior and activities of users.

2 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 access control using

Role Based Access Control simplifies access control administration‘and provides better manageability in enterprize environments by allowing permissions to be managed in terms of user job roles [9]. 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 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 eg, a passport, driving license or in computing terms password, strong authentication. Authorization is details of where you are permitted to go, 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, 16, 7], 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 architecture of the software system implemented. In Section 4 we analyze the results. Finally, in Section 5, we discuss future directions for research.

3 Related Research Technologies

3.1 Role Based Access Control

Role-based access control [5, 15, 13, 17, 3] has gained attention as a proven alternative to traditional discretionary (DAC) and mandatory access control (MAC) mechanisms. RBAC helps to specify organization's security policies reflecting its organizational structure. In the core 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 (SSD) 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.

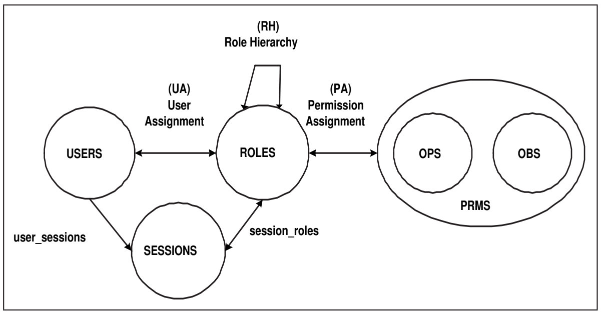

Figure 1: RBAC model

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 more information about RBAC is available at [9].

3.2 Privilege Management Infrastructure

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 Ű 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. The 'push' model is suitable when the client's permissions should be authenticated/validated in the client's 'home' domain, whereas the 'pull' model is suitable when the client's privelages should be authenticated in the inter-domain.

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.

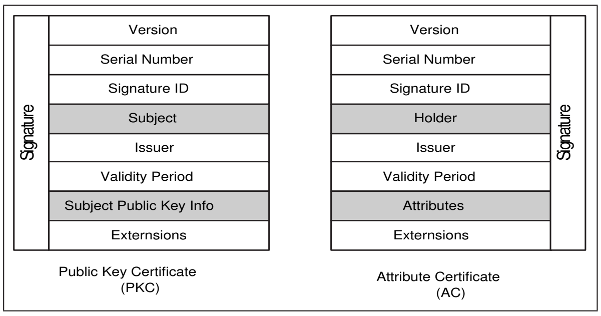

Figure 2: PKC and Attribute certificates

Further more information about AC is available at [14]

4 Design and Implementation

4.1 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 [13] 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.

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.

User certificate mainatanence problem can be avoided, if all the organizations participating in the information sharing service have the same rootCA. Key issues that need to addressed in order to have a shared rootCA among various organizations are

1. Task force of this multiple agencies set up a rootCA-MA (root CA for Multiple Agencies)
2. Each organization requests a certificated signed by rootCA-MA
3. each organization issues a new PKC to all users in its organization involved in the taskforce.
4. At each server providing secure information sharing service for this taskforce, add the rootCA-MA information into CABundle (file containing list of valid CA's)
5. each client/user needs to install the certificate in his/her browser

Automated process fo 100 certificates took 2 min and 13 sec. Based on my own person experience generating 1 certificate takes about 2 min 35 sec. In a coordination effort where we have 100 organizations and 100 people in each organization coordinating, Time taken in the manual process = 260 min and 55 sec 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.

4.1.1 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 [10]. In our current prototype implementation we have a simple policy specification file. (we need to evaluate policy specification using a) eXtensible Access Control Markup Language (SAML) [4] b) PERMIS X.500 PMI RBAC policy [1]).

Our current RBAC policy specification file is shown below

- *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.
- *Administration Tool:* is used for creating key pair, PKIs, User Role Specification ACs.

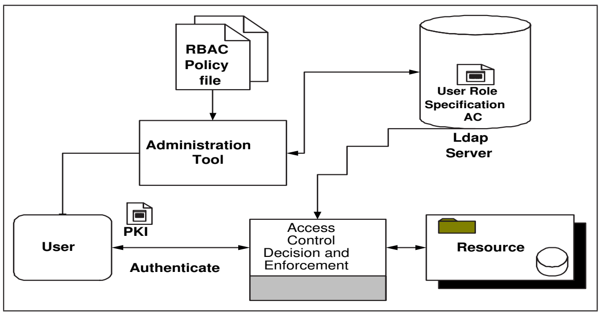

Figure 3: Overview of the SIS Access Control System

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

    <userRoleSpecification>
        <Role>Teller</Role>
        <Group>Info Share</Group>
        <OU>banking</OU>
    </userRoleSpecification>
</SIS>
```

Table 1: Sample RBAC File Format

- *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.

4.1.2 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 supplied to the Apache module. The module uses the Role information in the user Attribute Certificate along with Role Hierarchy information, to determine the access permissions to the requested web document.

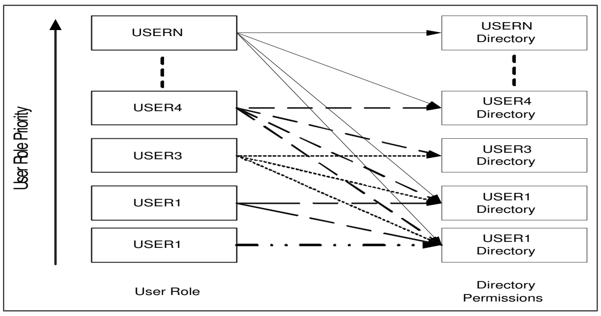

Figure 4: Mapping Role Hierarchy to Permissions

4.1.3 Administration tool

OpenSSL [11] provides strong open source cryptographic library for X509, and SSL&TLS. Currently OpenSSL does not have any support for Attribute Certificates except for RFC 3281 AC's ASN.1 object definitions. Attribute Certificates generation code was written using the 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 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. We currently donot have support for ACRL (Attribute Certificate Revocation List) needed by the 'push model'. We plan to provide this capability soon. This handicap can be circumvented by deleting the entry from the LdapServer.

4.2 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.

4.3 Information Sharing among Multiple Agencies

The Access Control Decision and Enforcement (ACDE) engine is implemented as an Apache [6] module. It is responsible for authorization service for web requests between user and the requested target file(s). This framework seperates Authentication service, provided by ModSSL [8] from authorization service. Our

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

    <userRoleSpecification>
        <Role>Teller</Role>
        <Group>Info Share</Group>
        <OU>banking</OU>
    </userRoleSpecification>
</SIS>
```

Table 2: Access Permissions specification Format

prototype consists of following components: Initiator (e.g. a browser), Target (web file request), Access Control Decision and Enforcement (ACDE) provided by the apache_sis_module.

The initiator submits web access request to the target, The ACDE gets the initiators details from the submitted public key and queries the LDAP server for validation. The ACDE uses information like organization available in the Subject Distinguished Name of the initiator's certificate. Once the user is found ACDE retrieves the User's Attribute Certificates and validates his/her User Role specification AC. Once the user is validated it makes a decision whether the user has the required privilage to access the information. Only users with Valid privilages are allowed to have access to the target by the ACDE.

5 Experimental results

In this section, we present some experimental results of Secure Information Sharing.

5.1 Prototype implementation

We implemented the secure Information Sharing for multiple agencies using web services. 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 [reference] for Apache was enhanced to provide ACDE functionality. Attribute Certificate object and attribute definitions were added to OpenLDAP (v 2.0.27-8). inetOrgPerson object class was modified to contain attribute certificate value(s). OpenSSL libraries were also used for generating X509 certificates.

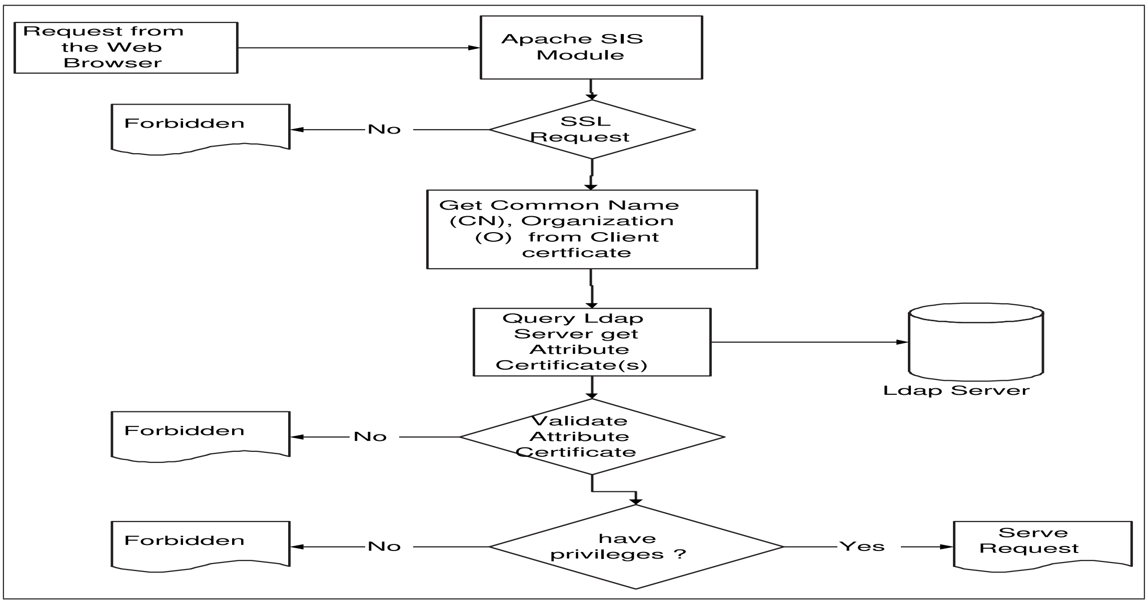
Request from the Web Browser
Apache SIS Module
Forbidden
No
SSL Request
Get Common Name (CN), Organization (O) from Client certficate
Query Ldap Server get Attribute Certificate(s)
Ldap Server
Forbidden
No
Validate Attribute Certificate
Forbidden
No
have privileges ?
Yes
Serve Request

Figure 5: Control Flow in Access Control Engine

5.2 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 organizational agency 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.

5.3 Analysis of Results

Table 2 and Table 3 show the performance results of sis-module in a single agency and multiple agency scenario. There is no much overhead in a multiple agency when compared with single agency scenario.

	Total time taken for LDAP access (ms)	Total Time taken for Attribute certificate retrieval and validation (ms)
1	46.820999	81.358002
2	52.733002	88.288002
3	55.066002	90.517998
Avg.	51.540001	86.721334

Table 3: Performance Results on a single agency

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.

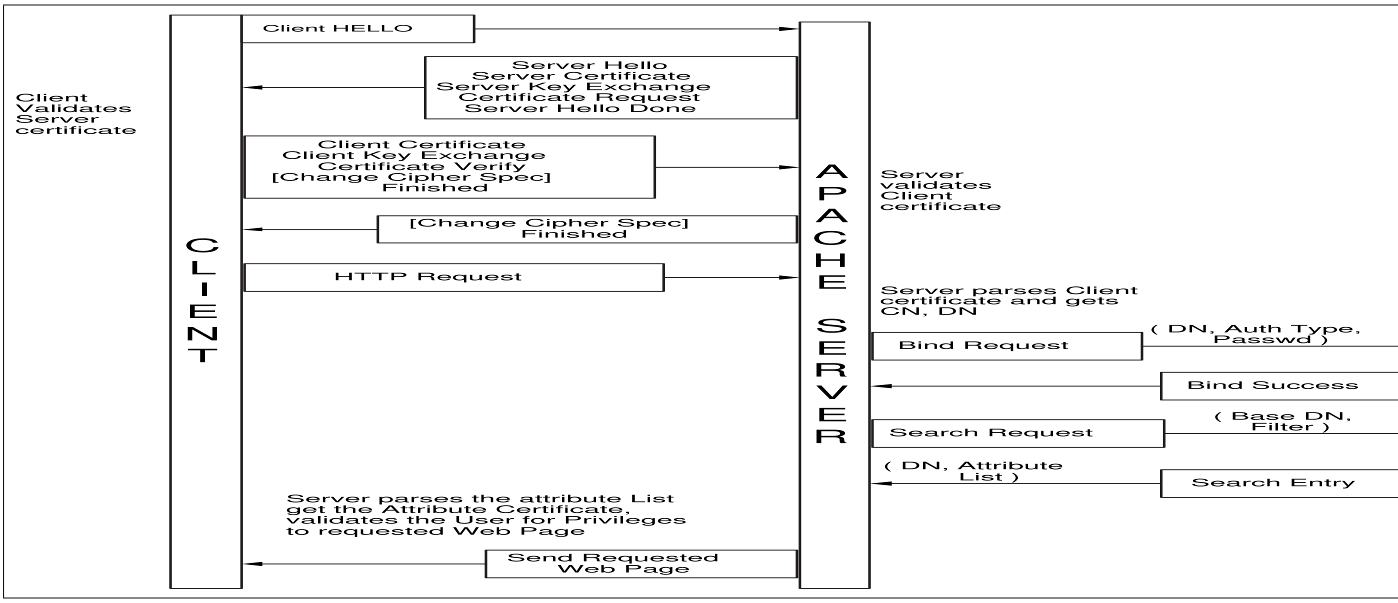

Figure 6: Message Flow between the components

	Total time taken for LDAP access (ms)	Total Time taken for Attribute certificate retrieval and validation (ms)
1	54.623001	96.885002
2	51.845001	93.778999
3	51.191002	93.310997
Avg.	52.55300133	94.65833267

Table 4: Performance Results in a multiple agency scenario

[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] eXtensible Access Control Markup Language (XACML). authorization policies specification in xml, 2004.

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

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

[7] 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.

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

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

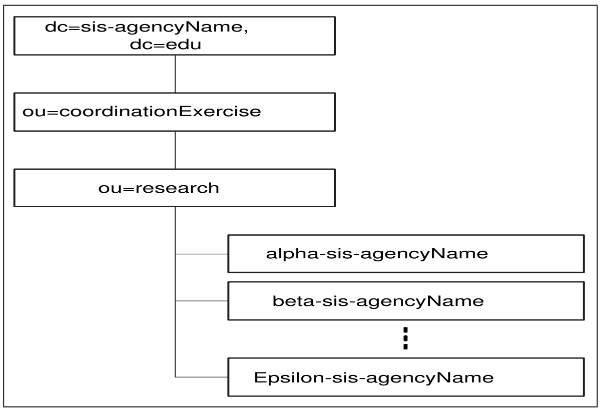

Figure 7: LDAP DIT Format at each agency

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

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

[12] Phillips, Jr., C. E., Ting, T., and Demurjian, S. A. Information sharing and security in dynamic coalitions. In *Proceedings of the seventh ACM symposium on Access control models and technologies* (2002), ACM Press, pp. 87–96.

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

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

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

[16] 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.

[17] 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.