Role-based Access Management for Ad-hoc Collaborative Sharing

Jing Jin and Gail-Joon Ahn
The University of North Carolina at Charlotte
{jjin,gahn}@uncc.edu

ABSTRACT

Under scientific collaborations, resource sharing tends to be highly dynamic and often ad hoc. The dynamic characteristics and sharing patterns of ad-hoc collaborative sharing impose a need for comprehensive and flexible approaches to reflect and cope with the unique access control requirements associated with the ad-hoc collaboration. In this paper, we propose a role-based access management framework to enable secure resource sharing, especially focusing on the digital information sharing in the heterogeneous scientific collaboration environments.

Our framework incorporates role-based approach to address distributed access control, delegation and dissemination control involved in the resource sharing within such environments. A set of XACML-based policy schemas is proposed to specify policies on our framework. To demonstrate the feasibility of our framework, we design and implement a proof-of-concept prototype system called ShareEnabler, which is based on a peer-to-peer information sharing toolkit developed by Lawrence Berkeley National Laboratory.

Categories and Subject Descriptors: D.4.6 [Operating Systems]: Security and Protection—Access controls; K.6.5 [Management of Computing and Information Systems]: Security and Protection—Unauthorized access.

General Terms: Security.

Keywords: access control, ad-hoc collaboration, information sharing, XACML-based policy framework.

1. INTRODUCTION

The rise of Internet and Web technologies has enabled traditional scientific collaboration to turn outward and connect distributed participants across enterprises and research institutes. By removing the geographical distance barriers, scientists and engineers from different organizations are able to establish collaboration relationships and share information collaboratively. Under many circumstances, the collaboration relationship is established based on spontaneous interactions and use patterns in an ad-hoc fashion. For example, groups of universities, laboratories, and industrial companies may collaborate and mutually share research results on a particular human disease; different educational agencies collaboratively implement, disseminate, and institutionalize effective practices for supporting and promoting people from underrepresented groups in Computer Science; and a crisis management team collaborates with special agents responding to a chemical spill accident. As many of these examples show, the establishment of collaboration relationship is highly dynamic and may vary tremendously in terms of purpose, scope, size, duration, and the number of involved participants. We define this type of collaboration as *ad-hoc collaboration*, and the resource sharing involved in such collaborations as *ad-hoc collaborative sharing*.

In scientific collaborative communities, ad-hoc collaboration allows individual participants who belong to many different organizations to spontaneously establish collaboration relationships, dynamically contribute data resources to be shared, and share data offered by others within the collaboration group. Compared to the well-structured collaboration that is addressed in the Grid [7], the formulation of ad-hoc collaboration interactions is essentially more transient and there is no pre-established global consensus of trustworthiness among all participating parties. As a result, it requires a more light-weighted infrastructure without pre-configured environments or central management authorities to support the ad-hoc collaborative sharing.

Nevertheless, given all the diverse contexts of collaborative sharing, achieving the effective access control is a critical requirement. The data sharing is necessarily to be highly controlled, with resource providers carefully defining what and how is to be shared, who is allowed to share, and under which condition the sharing occurs. Recently, some approaches have been proposed to support secure collaborative sharing [4, 5, 15, 19]. These approaches, however, focus more on authorization mechanisms rather than the generic models. Our preliminary study clearly indicates that there is a need to design a comprehensive access control framework that is general and flexible enough to reflect and cope with the special access control requirements associated with the ad-hoc collaboration. In this paper, we make one step towards this direction. Particularly, among the many forms of collaborative sharing, we focus on the digital information sharing.

The rest of the paper is organized as follows. In section 2, we define our problem domain starting with a typical collaborative sharing scenario, from which we identify generic

SACMAT'06, June 7–9, 2006, Lake Tahoe, California, USA.
Copyright 2006 ACM 1-59593-354-9/06/0006 ...$5.00.

access control requirements and data sharing patterns associated with the ad-hoc collaboration. We introduce our role-based access management framework in section 3. In section 4, we show our policy specification framework using XACML. The integrated system design and prototype implementation is described in section 5. In section 6, we review other related works that dealt with authorization issues in collaborative environments. Section 7 concludes the paper with future research directions.

2. PROBLEM DOMAIN ANALYSIS

In this section we proceed with a typical scientific collaborative sharing scenario [11], from which we identify the generic access control requirements associated with the adhoc collaboration. We then analyze the patterns of collaborative data sharing and dissemination.

2.1 Ad-hoc Collaborative Sharing Scenario and Access Control Requirements

The tobacco control and addiction research relies on a transdisciplinary collaboration involving a number of universities and research groups to mutually share research results and explore the integrated knowledge across disciplines.

Suppose Regional Medical Center (RMC) investigates on genetical factors of tobacco addiction by analyzing patients' genotypes and their family medical histories maintained in a local database. The data has the potential to be used in other types of research, for instance, to verify a new sociological hypothesis, to conduct animal model comparisons, or to be used in pharmaceutical studies. In the context of ad-hoc collaboration, any of interested research organizations, especially each individual member within these organizations, could directly join the collaboration relationship with RMC to share and distribute the data. However, RMC's collected data from many sources may be sensitive and sharing of the data can be restricted by particular regulations. The resource owner (RMC) obviously needs to apply appropriate conditions on the dissemination of and access to the data.

From the scenario above, we first differentiate the concepts of *collaborating organizations* and *collaborating participants*. In this paper, we assume the resource owner has the limited trust on some collaborating organizations based on pre-established relationships. The *collaborating organizations*, however, do not directly share the data. The individual members in these organizations are the actual data consumers where the access control should be applied on. We name these individuals as *collaborating participants*. The distributed and spontaneous natures of collaborating participants determine unique access control requirements.

Firstly, compared to the ever-changing collaboration relationships and participants in the ad-hoc collaboration, the involved resource and the ownership of the resource are relatively static. It is essential to allow the resource owner (also called the *originator*) to define her collaborative sharing domain by performing naming and authentication of the collaborating participants (also called *collaborators*) in order to authorize and delegate fine-grained access capabilities. And the definition of the collaborative sharing domain should be defined on per-resource basis for different sharing purposes.

Secondly, since the ad-hoc collaboration may involve a large amount of participants across domains, it is impossible to enumerate all potential participants for a given resource. The naming of collaborating participants should be classified by the resource to be shared and privilege abstractions are needed to achieve the flexibility and reduce the administrative complication.

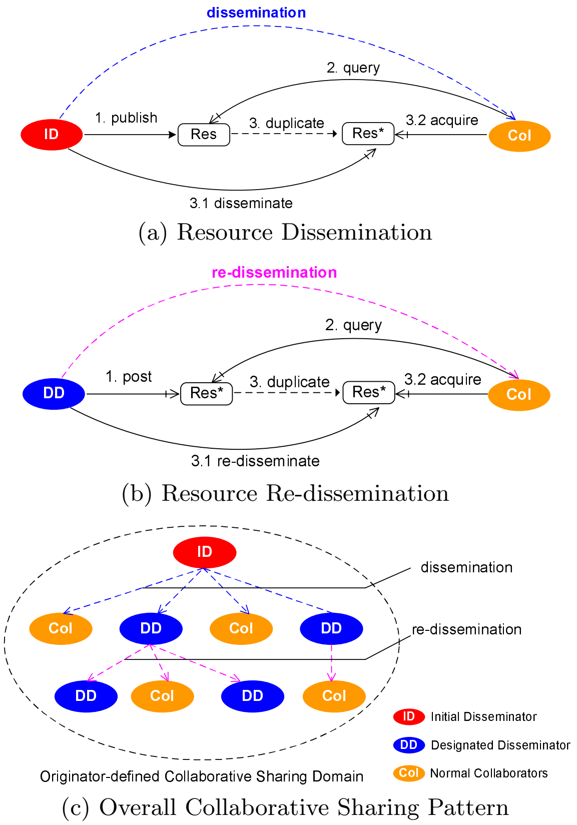
Thirdly, resources may be called upon to be shared with little prior knowledge of collaborating participants. The requester's identity alone is not good enough for being meaningful to the resource originator. Therefore, delegation is necessary to leverage an effective way of propagating authorities as well as constructing trustworthiness among distributed collaborating parties. The trustworthiness on the individual participant should be based on the trustworthiness of the delegated authorities.

Finally, since the collaboration relationship is loosely established, there is no central administrative point or global agreement of trust in the ad-hoc collaboration. Any involved parties cannot rely on centralized administrative authorities for crucial security services, such as membership management, access and usage control. Originators, therefore, are responsible to define their own trust relationships, formulate and enforce access control policies to protect their resources from unwanted hostile ones. As collaborative sharing may involve digital information transmission among participants, a distributed policy propagation and enforcement scheme with decentralized, self-enforcing, and self-monitoring features is required to fulfill functional and assurance requirements desired by originators in order to efficiently share their resources.

2.2 Collaborative Sharing Patterns and Dissemination Requirements

Generally, collaborative sharing involves a set of generic behaviors such as *resource discovery*, *resource acquisition* and *resource distribution* (including the *resource dissemination* and *re-dissemination*). In the particular case of digital information sharing, the sharing starts with an originator publishing the original resource in the collaborative community. A collaborator gets aware of the availability of the resource through *resource discovery*. The collaborator then can further request to share the resource through *resource acquisition*. The originator sends a copy of the digital resource to the requester and fulfills the initial *resource dissemination*. With the consent of the originator, the resource recipient may further *re-disseminate* the pre-obtained resource copy to others. We consider the originator as the *initial disseminator* (*ID*) since she triggers the initial resource distribution. The initial disseminator disseminates copies of the original resource (*Res*). And we call collaborators who further disseminate resource copies (*Res**) as the *designated disseminator* (*DD*). Figure 1 shows the use patterns as stepwise procedures of resource dissemination and resource re-dissemination, where effective access control mechanisms should be applied on each procedure.

In a collaborative sharing environment, all participants and their capabilities should be clearly defined, and all sharing behaviors should be highly regulated. Especially, it is required that only the originator as the *initial disseminator* (*ID*) and/or *designated disseminators* (*DDs*) could distribute the resource to other legitimate collaborators within the originator's collaborative sharing domain. Figure 1(c) shows an ideal pattern of the well-defined and highly regulated resource dissemination and re-dissemination in a collaborative sharing environment. Basically the root of the



(a) Resource Dissemination

(b) Resource Re-dissemination

(c) Overall Collaborative Sharing Pattern

Figure 1: Use Patterns of Resource Dissemination and Re-dissemination

sharing tree should be an *ID* and the sharing flows coming out from the *ID* are types of resource dissemination. Similarly, the intermediate nodes should be *DD*'s and the sharing flows from the *DD*'s are types of resource re-dissemination.

3. ROLE-BASED ACCESS MANAGEMENT FOR RESOURCE SHARING (RAMARS)

3.1 General Principles

As identified in the previous section, the unpredictable user participation poses great challenges for an originator to define her collaborative sharing domain and the resource authorization. Instead of enumerating all potential participants, the originator could use roles, such as *data analyst* or *lab coordinator*, to formulate classes of participants and define her collaborative sharing domain(s). Through being assigned to these roles, participants are automatically included in the originator's collaborative sharing domain(s) and thus obtain various access capabilities on the resource. Being revoked from the roles, participants are then excluded from the originator's domain(s) and lose the privileges. Therefore, bringing "role" in our framework becomes a natural choice to achieve the manageability in the ad-hoc collaboration environment.

Our role-based approach, however, distinguishes from traditional RBAC [18, 6] in various aspects. On one hand, existing RBAC models tend to rely on a single organizational policy to define the role constructs. We see roles as more flexible and more widely applicable to be defined independently across multiple administrative domains in a distributed environment. With such views, we design a model that supports generic sharing roles to capture the identified resource sharing patterns. Also, we design originator related roles and collaborator roles that are resource sharing specific, in order to reflect the special characteristics of an originator as well as to accommodate an originator's collaborative sharing domain on the given resources. On the other hand, as an extension to the traditional RBAC permission assignment, our framework allows the originator to delegate usage and dissemination capabilities through the capability-role assignment.

In addition, simply introducing roles reduces the management complexity, yet the user-role assignment remains as an issue to the originator due to the unknown participants. As we assume an originator has put limited trust on her collaborating organizations based on pre-established relationships, we introduce a special type of administrative delegation, called *Delegation of Delegation Authority* (*DoD*), as another layer of authority decentralization to achieve distributed role assignment. In particular, *DoD* enables an originator to partially delegate the role assignment authority to trusted collaborating organizations or trusted individuals. For example, suppose an originator attempts to share the resource with all *students* in the University, the originator may delegate the user-*student* assignment authority to the University registrar and trust the assignment tasks conducted by the University registrar.

3.2 RAMARS in Details

We propose a framework to support Role-based Access Management for Ad-hoc Resource Sharing (RAMARS). We define a collection of basic elements and relations that are involved in the ad-hoc collaboration. This covers the core set of features to be encompassed in collaborative resource sharing systems. The basic concept of our framework is built based on traditional RBAC concepts:

> There are roles and role hierarchy constructs (R,RH); participants (PAR)[1] are assigned to roles (UA); capabilities (CAP)[2], as operations (OP) towards resources (RES)[3], are assigned to roles (PA); and participants acquire capabilities by being members of roles.

Our framework introduces new elements and functions. Firstly, we introduce a new element, *Organization*, to identify the collaborating participants. Accordingly, a new function *belongs_to* is introduced to associate the participants (PAR) with the organization (ORG). Each participant belongs to at least one organization, from which the participant obtains his/her credentials (e.g. X.509 certificates) for further authentication and authorization in collaboration

[1] We use *participants* instead of *users* to distinguish distributed collaborating participants from traditional RBAC users in a closed organizational environment.

[2] We use *capabilities* to address the extension of *permission* delegation.

[3] As the collaborative sharing involves resource duplication, we need to clearly specify the capability as an operation towards either an original resource or a copy of the original resource.

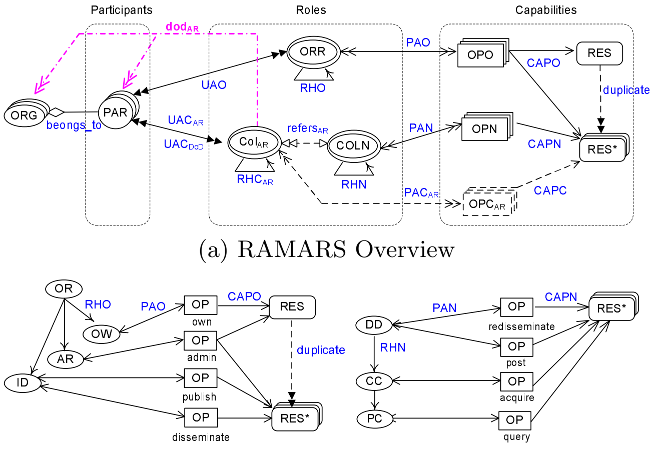
services. The trustworthiness on the participant is based on the trustworthiness on the organization which he/she belongs to. Even though we use "organization" here, the association is not necessarily at the organizational level, any sub-division of an organization along the trust chain could be the trusted entity to identify participants.

Secondly, based on different relationships to the shared resource, roles are partitioned into originator related roles (ORR) and collaborator related roles. Originator related roles address special characteristics and functionalities of an originator. In particular, the *originator* role (OR) abstracts a set of specific roles such as the resource *owner* (OW) who owns the resource; the resource *administrator* (AR) who is responsible to define the access management policies for the resource; and the resource *initial disseminator* (ID) who triggers the initial resource dissemination process. Collaborator related roles (Col_{AR}), on the other hand, are the resource-sharing specific roles defined at an originator's discretion to reflect her sharing purposes and to delegate fine-grained capabilities ($CAPC_{AR}$). In doing so, our proposed framework is flexible and scalable to fulfill diverse collaborative sharing requirements. The partition of roles induces a parallel partition of UA and PA. UA derives two additional relations, participant-originator role assignment (UAO) and participant-collaborator role assignment (UAC). Similarly, capability-originator role assignment (PAO) and capability-collaborator role assignment (PAC) are derived from PA.

Thirdly, we generalize a set of normative collaborative sharing operations (OPN) to reflect the identified generic sharing behaviors. The capabilities associated with these normative operations are global capabilities ($CAPN$) in the ad-hoc collaborative sharing environment that are independent of any administrative domains. In accommodating the role-based approach, we design a set of normative collaboration roles ($COLN$) as *Designated Disseminator* role (DD), *Common Collaborator* role (CC) and *Potential Collaborator* role (PC) to abstract these capabilities. Resource discovery (*query*) is the most fundamental capability associated with the PC role. Resource acquisition (*acquire*) is an advanced capability associated with the CC role with the capability of *query* as a prerequisite. And resource distribution (*post* and *redisseminate*) are the most advanced capabilities associated with the DD role. Besides, DD should inherit all capabilities associated with PC and CC. In order for the collaborator roles to achieve the generic sharing capabilities, each collaborator role is mapped to one of the normative roles through a function $refers\_to_{AR}$.

In addition, to accommodate the distributed role assignment, the delegation of delegation authority is reflected as a relation (dod_{AR}) mapping a collaborator role (Col_{AR}) to a set of participants (PAR) and/or a set of organizations (ORG). The participant-collaborator role assignment is done either by the originator alone (UAC_{AR}), and/or by other authorities defined in the *delegation of delegation authority* relation (UAC_{DoD}).

Finally, there exists resource duplication when the resource is distributed (*duplicate*). Instead of the original resource, a copy of the resource is disseminated at each sharing event. Figure 2 overviews our framework. Detailed design of originator related roles and normative roles are shown in Figure 2(b) and Figure 2(c) respectively. We finally summarize our framework as follows:



(a) RAMARS Overview

(b) ORR Details

(c) COLN Details

Figure 2: RAMARS Model

- Sets of PAR, ORR, $COLN$, Col_{AR}, OPN, OPO, OPC_{AR}, RES, RES^* and ORG (denoting participants, originator related roles, normative collaboration roles, customized collaborator roles, normative collaboration operations, operations for originator roles, customized operations for collaborator roles, resources, resource copies and organizations respectively).
  $ORR = \{OR, ID, AR, OW\}$
  $COLN = \{DD, CC, PC\}$
  $OPN = \{query, acquire, post, redisseminate\}$
  $OPO = \{publish, disseminate, own, admin^4\}$
- $R = ORR \cup COLN \cup Col_{AR}$
- $CAPN = 2^{OPN \times RES^*}$, a set of normative capabilities.
  $CAPO = 2^{OPO \times (RES \cup RES^*)}$, a set of capabilities for originator roles.
  $CAPC_{AR} = 2^{OPC_{AR} \times RES^*}$, a set of customized capabilities for collaborator roles.
  $CAP = CAPN \cup CAPO \cup CAPC_{AR}$
- $PAN \subseteq CAPN \times COLN$, a many to many capability-normative role assignment relation.
  $PAN = \{(query, PC), (query, CC), (acquire, CC),$
  $(query, DD), (acquire, DD), (post, DD),$
  $(redisseminate, DD)\}$
  $PAO \subseteq CAPO \times ORR$, a many to many capability-originator role assignment relation.
  $PAO = \{(own, OW), (admin, AR), (publish, ID),$
  $(disseminate, ID), (own, OR), (admin, OR), (publish, OR),$
  $(disseminate, OR)\}$
  $PAC_{AR} \subseteq CAPC_{AR} \times Col_{AR}$, a many to many capability-collaborator role assignment relation.
  $PA = PAN \cup PAO \cup PAC_{AR}$
- $UAO \subseteq PAR \times ORR$, a many-to-many participant-originator role assignment relation.
  $UAC_{AR} \subseteq PAR \times Col_{AR}$, a many-to-many participant-collaborator role assignment relation defined by an originator as AR role.
  $UAC_{DoD} \subseteq PAR \times Col_{AR}$, a many-to-many participant-collaborator role assignment relation defined by a DoD delegatee.

[4] A simplified operation that abstracts a set of administrative operations in defining the elements, relations and functions. We use the subscript $_{AR}$ to denote *admin* operations.

$UAC = UAC_{AR} \cup UAC_{DoD}$
$UA = UAO \cup UAC$

- $RHN \subseteq COLN \times COLN$, is a partial order on $COLN$. $RHO \subseteq ORR \times ORR$, is a partial order on ORR. $RHC_{AR} \subseteq Col_{AR} \times Col_{AR}$, is a partial order on Col_{AR}. $RH = RHN \cup RHO \cup RHC_{AR}$.
- $capabilities(r : R) \rightarrow 2^{CAP}$, the mapping of role r onto a set of capabilities in the presence of a role hierarchy. $capabilities(r) = \{cap \in CAP | r' \succeq r, (cap, r') \in PA\}$.
- $belongs\_to(par : PAR) \rightarrow ORG$, a function mapping each participant par to a single organization which he/she belongs to.
- $refers\_to_{AR}(col : Col_{AR}) \rightarrow COLN$, a function mapping each collaborator role to a single $COLN$ role.
- $dod_{AR}(col : Col_{AR}) \rightarrow 2^{ORG} \cup 2^{PAR}$, a function mapping one Col_{AR} role to a set of organizations and/or participants as the trusted third party authorities to conduct the participant-collaborator role assignment.
- $duplicate(res : RES) \rightarrow 2^{RES^*}$, a function mapping an original resource to a set of duplicated resource copies.

3.3 Framework Realization in an Example

To evaluate our proposed framework, we extend the initial collaborative sharing scenario to realize our framework as follows.

Inside the tobacco research community, a team of sociobiology scientists from LIISP research lab, with John as the team leader and Dave as one of the team members, is conducting research tasks related to tobacco use based on cultural and family modeling analysis. John's team needs to collaborate with RMC and use RMC's data to verify a new hypothesis drawn from their survey.

In the above example, *RMC* is the *originator* and each individual member in *LIISP* lab, *John* and *Dave*, is considered as a *collaborating participant* that needs to be authorized individually in the collaborative sharing domain. To authorize accesses of the members in LIISP lab, RMC defines two collaborator roles as ***Coordinator*** role and ***Investigator*** role, where ***Coordinator*** is a senior role mapped to ***DD***, and ***Investigator*** is a junior role mapped to ***CC***. By knowing *John* as the team leader through previous relationships, RMC assigns ***John*** to the ***Coordinator*** role so that *John* is able to re-disseminate the data. RMC also delegates *John* to perform the participant-***Investigator*** role assignment through delegation of delegation authority. Under $dod_{AR}(Investigator)$, *John* is able to assign his other team member ***Dave*** to the ***Investigator*** role. Figure 3 shows how this example is realized in the framework.

4. POLICY SPECIFICATION

We design a set of XACML-based policies to support our RAMARS framework. We assume the readers are familiar with the OASIS standard XACML [14] specification language for expressing access control policies. In supporting RBAC, OASIS has recommended a specification for RBAC policies [13] (we name "OASIS specification" for simplicity). Our policy modules extend the OASIS specification in expressing role and role hierarchy constructs, capability-role assignments and participant-role assignments. We further contribute to define the role mapping between a collaborator role and a normative collaboration role as well as to define the "delegation of delegation authority" relations. We emphasize the *Issuer* attributes in policies to differentiate distributed authorities and trust relationships. Our policy framework starts with defining Role-based Originator Authorization policy sets (**ROA**). Besides, we design a Root Meta Policy Set (**RMPS**) as a means to accommodate distributed deployment of **ROA** policies and to achieve policy reusability and portability. The components of our policy framework are explained as follows:

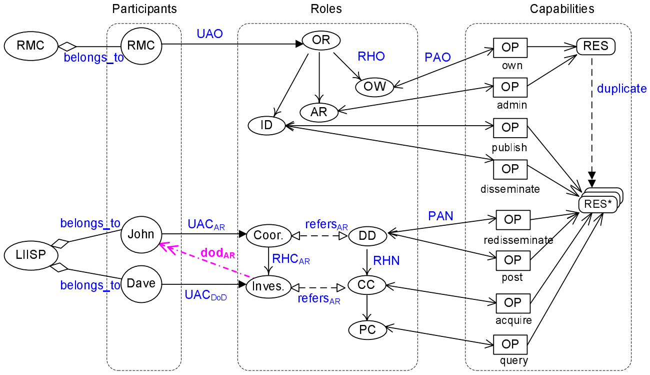


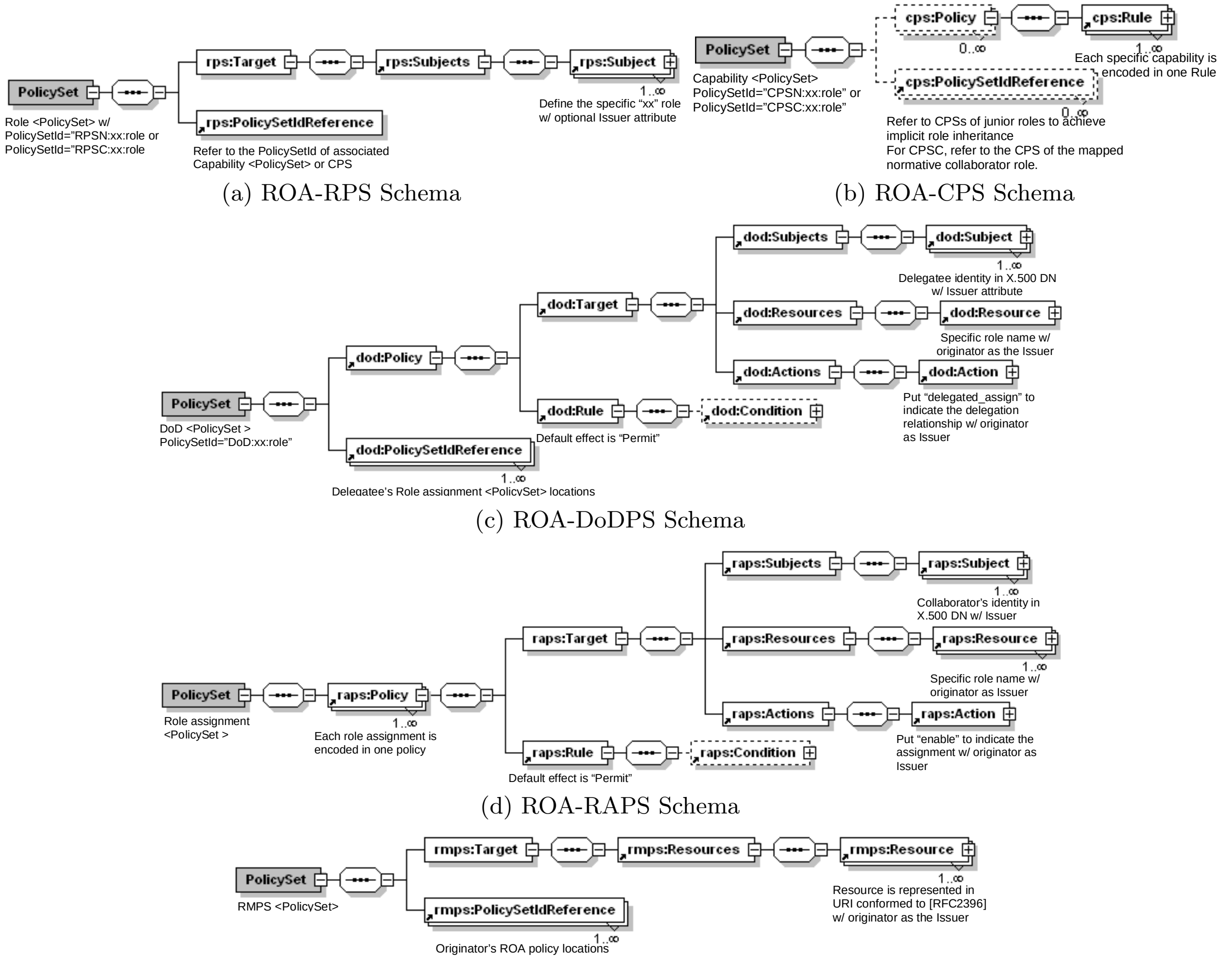
Figure 3: Collaborative Sharing Example

Role Policy Set (RPS) is a role specification policy set. An originator defines her collaborative sharing domain in a set of RPS's. In achieving the role-capability assignment, each RPS is associated with a *Capability PolicySet* (*CPS*) that actually contains capabilities of the given role. The role is specified as a *Subject* attribute, and the corresponding *CPS* is referenced through a *PolicySetReference* element. To differentiate specifications of normative roles from originator-defined collaborator roles, we use *PolicySetId*'s starting with RPSN or RPSC [5]. In addition, the originator is required to add the *Issuer* attribute in the specified role *Subject* to claim her role specification authority. Figure 4(a) shows the schema of *RPS*.

Capability Policy Set (CPS) specifies the actual capabilities assigned to the given role. The CPS contains *Policy* and *Rule* elements that describe capabilities as *Resources* and *Actions*. The CPS may also contain references to other CPSs associated with other roles that are junior to the given role, thereby achieving the role hierarchies through the capability aggregation. In order to map a customized collaborator role to a normative collaboration role, the CPS of the collaborator role should include a reference to the CPS of the corresponding normative collaboration role. Figure 4(b) shows the schema of *CPS*.

Delegation of Delegation Authority Policy Set (DoDPS) reflects the "delegation of delegation authority" with the originator specifying which role assignments are delegated to which trusted authorities. The third party delegatee(s) are represented as their X.509 DN(s) in *Subjects*, and the role of delegated assignment is specified as the *Resource*. The term "*delegated_assign*" is used in *Action* to explicitly indicate the delegation relationship with the originator as

[5] "N" stands for *normative* and "C" stands for *customized*.


PolicySet
Role <PolicySet> w/
PolicySetId="RPSN:xx:role or
PolicySetId="RPSC:xx:role
rps:Target
rps:Subjects
rps:Subject
1..∞
Define the specific "xx" role
w/ optional Issuer attribute
rps:PolicySetIdReference
Refer to the PolicySetId of associated
Capability <PolicySet> or CPS
PolicySet
Capability <PolicySet>
PolicySetId="CPSN:xx:role" or
PolicySetId="CPSC:xx:role"
cps:Policy
0..∞
cps:Rule
1..∞
Each specific capability is
encoded in one Rule
cps:PolicySetIdReference
0..∞
Refer to CPSs of junior roles to achieve
implicit role inheritance
For CPSC, refer to the CPS of the mapped
normative collaborator role.
PolicySet
DoD <PolicySet >
PolicySetId="DoD:xx:role"
dod:Policy
dod:Target
dod:Subjects
dod:Subject
1..∞
Delegatee identity in X.500 DN
w/ Issuer attribute
dod:Resources
dod:Resource
Specific role name w/
originator as the Issuer
dod:Actions
dod:Action
Put "delegated_assign" to
indicate the delegation
relationship w/ originator
as Issuer
dod:Rule
dod:Condition
Default effect is "Permit"
dod:PolicySetIdReference
1..∞
Delegatee's Role assignment <PolicySet> locations
PolicySet
Role assignment
<PolicySet >
raps:Policy
1..∞
Each role assignment is
encoded in one policy
raps:Target
raps:Subjects
raps:Subject
1..∞
Collaborator's identity in
X.500 DN w/ Issuer
raps:Resources
raps:Resource
1..∞
Specific role name w/
originator as Issuer
raps:Actions
raps:Action
Put "enable" to indicate the
assignment w/ originator as
Issuer
raps:Rule
raps:Condition
Default effect is "Permit"
PolicySet
RMPS <PolicySet>
rmps:Target
rmps:Resources
rmps:Resource
1..∞
Resource is represented in
URI conformed to [RFC2396]
w/ originator as the Issuer
rmps:PolicySetIdReference
1..∞
Originator's ROA policy locations

(a) ROA-RPS Schema

(b) ROA-CPS Schema

(c) ROA-DoDPS Schema

(d) ROA-RAPS Schema

(e) RMPS Schema

Figure 4: Policy Schemas

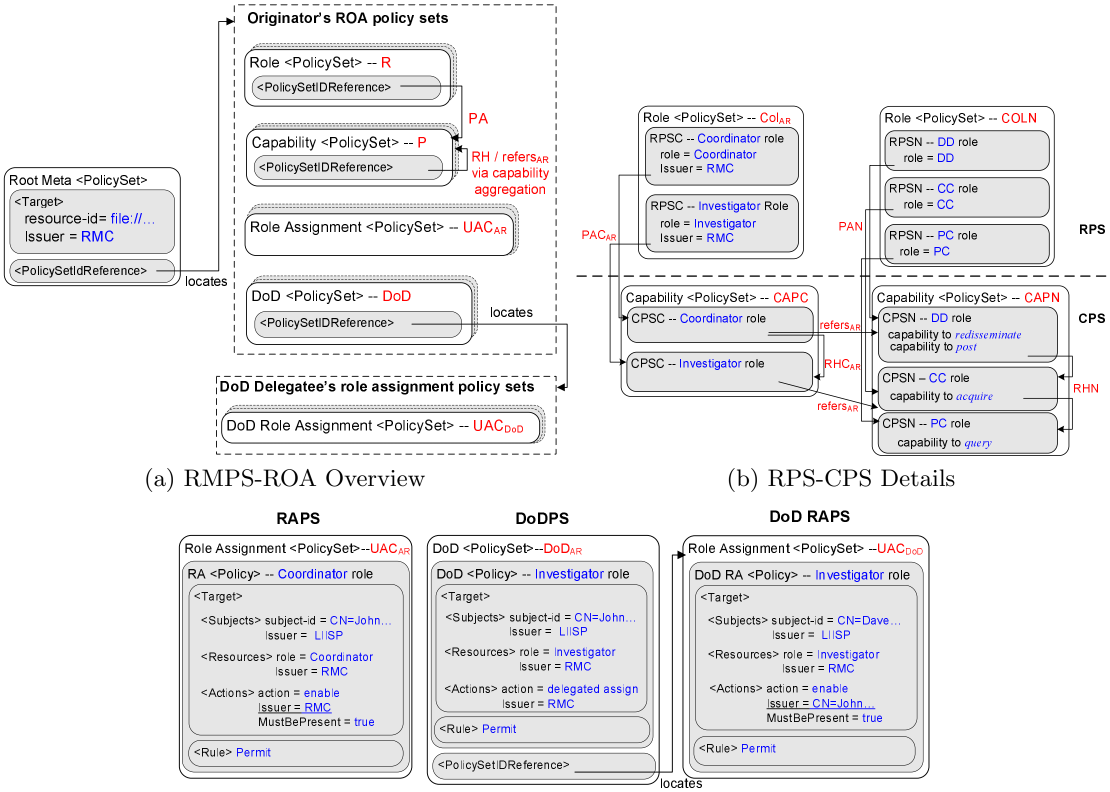
the *Issuer*. In addition, the delegatees' role assignment policies are referenced through *PolicySetIdReference* elements. Figure 4(c) shows the schema of *DoDPS*.

Role Assignment Policy Set (RAPS) is specified by an originator or a DoD delegatee to define which roles are assigned to which participants. Collaborating participants are specified in their X.500 DNs as *Subjects*. The assigned role is specified as the *Resource*. And the term"*enable*" is used in *Action* to indicate the assignment relationship. *Issuer* attributes are applied both in the *Resource* and the *Action* to distinct authorities of the role specification and role assignment. In particular, if the role assignment issuer is the same as the role issuer, it means the role assignment is performed by the originator through UAC_{AR}. Otherwise, the role assignment is done by a DoD delegatee through UAC_{DoD} and the dod_{AR} relation should be checked as well. Figure 4(d) shows the schema of *RAPS*.

Root Meta Policy Set (RMPS) is designed to achieve the portability and reusability of ROA policies. As the ROA policies discussed above are independent from the applied resource and can be deployed in distributed originators' domains, RMPS is designed to associate ROA policies with the specific resource and enables the policy enforcement system to locate the ROA polices. In RMPS, the resource is represented as a URI [16], with the originator's X.509 DN as the *Issuer*. The originator's ROA policy is referenced through *PolicySetIdReference* elements. Figure 4(e) illustrates the schema of *RMPS*.

4.1 Policy Examples and Policy Evaluation

In this section, we show how the proposed policy framework can be realized and how the policies are evaluated. Using the same example, Figure 5(a) shows an overview of the whole policy framework structure and the relationships among the policies. In general, RMPS specifies the resource with *RMC* as the originator and locates the originator's ROA policy sets. *RMC*'s ROA policies contain a set of RPSs, CPSs, RAPSs and DoDPSs. RPSs define two roles of *Coordinator* and *Investigator* in the collaborative sharing domain and normative collaboration roles of *DD*, *CC* and *PC*. CPSs specify the corresponding capabilities associated with these roles. The role hierarchy and role mapping are indirectly achieved through capabilities aggregation, representing that the *Coordinator* role is senior to the *Investigator* role, the CPS of *Coordinator* role references to the CPS of *Investigator* role so that the *Coordinator* role could inherit all capabilities that are assigned to the *Investigator*



(a) RMPS-ROA Overview

(b) RPS-CPS Details

(c) RAPS-DoDPS Details

Figure 5: Policy Examples

role. The role mapping is done in similar ways where the *Coordinator* role is mapped to *DD* role and the *Investigator* role is mapped to *CC* role. The DoDPS specifies the $dod_{AR}(Investigator)$ relation between the *Investigator* role and *John*. John's role assignment policy is referenced via the *PolicySetIdReference* element. Figure 5(b) and 5(c) show the detailed definitions in the policy sets.

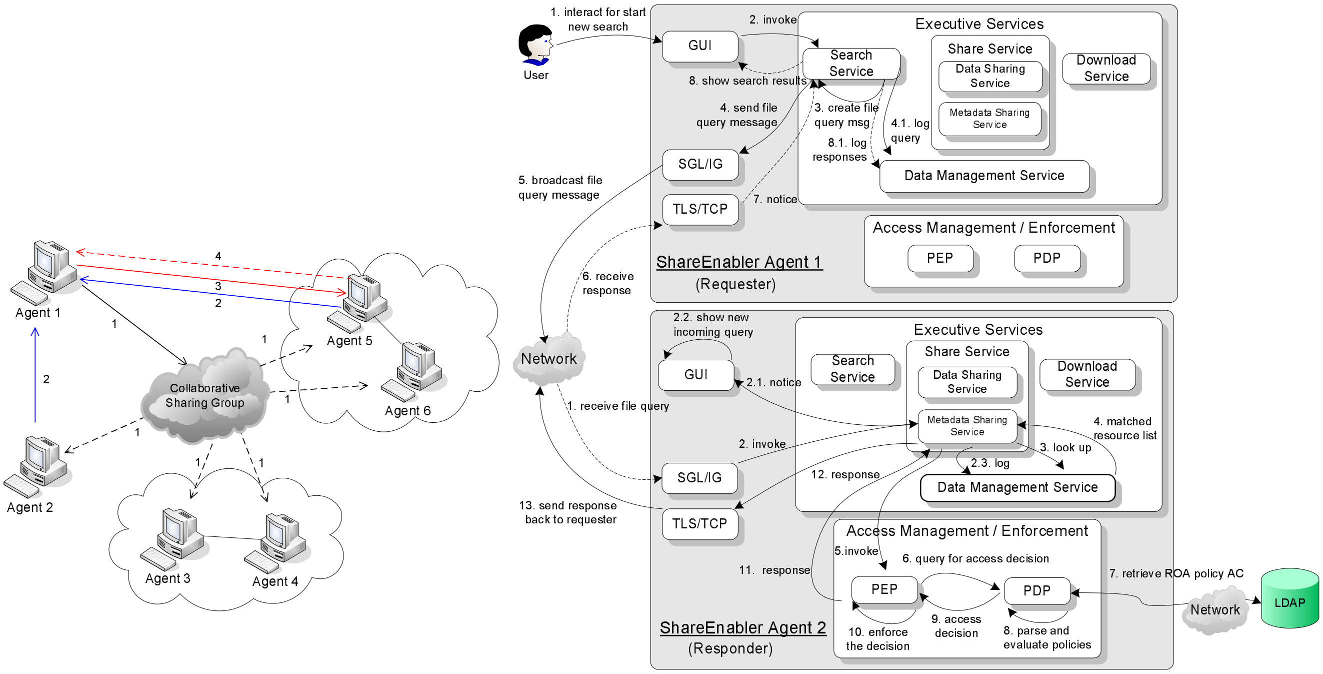
As our policy framework conforms to the XACML standard, the policy evaluation and decision-making can be done as specified in [14]. The typical setup is that PEP forms a standard access request based on the requester's attributes, the resource in question, the action, and other information pertaining to the request. The PEP will then send this request to PDP and wait for the PDP to evaluate the request against applicable policies and come up with a response with one *Decision* element of value *Permit*, *Deny*, *Indeterminate* or *NotApplicable*.

Suppose *Dave* from *LIISP*, is allowed to *acquire* the data file. PEP constructs the access request including *Dave*'s X.509 identity, the requested file resource and the specific action (*acquire*). The PDP first retrieves the roles that are assigned to the requester's X.509 identity (*CN=Dave...*), the role issuer and the role-assignment issuer. In our case, the PDP retrieves that *Dave* is assigned to the *Investigator* role by *John* (*CN=John...*). Since the role-assignment issuer (*John*) is different from the originator (*RMC*), it means a *delegation of delegation authority* should be involved. The PDP conducts the DoD evaluation by examining the *DoDPS* to check whether the role-assignment issuer *John* is a legitimate DoD delegatee. The PDP then conducts the role-assignment evaluation by evaluating John's DoD *RAPS* to examine whether the requester (*Dave*) is assigned to the *Investigator* role. Finally, the PDP conducts the role-access evaluation by evaluating *RMPS*, *RPS* and *CPS* to examine whether the *Investigator* role is allowed to conduct the "*acquire*" action on the file resource. The final decision is sent back to the PEP for further decision enforcement.

5. SYSTEM DESIGN AND PROTOTYPE IMPLEMENTATION

As part of our on-going research efforts, we have designed and implemented a prototype system, *ShareEnabler*, to demonstrate how the proposed access management framework and policy specification can be deployed as detailed authorization services and mechanisms within the context of collaborative sharing applications.

In particular, ShareEnabler adopts a specific communication infrastructure from a P2P based information sharing toolkit SciShare [4] developed by Lawrence Berkeley National Laboratory (LBNL). In our collaborative sharing system, each participant is represented by a ShareEnabler agent that executes sharing services on the participant's behalf. Similar to most of existing P2P file sharing systems, the resource discovery involves broadcasting a query to all known peers, while sending responses and resource dissemination are bound to unicasting communications. Figure 6(a) shows an overview of the system infrastructure. Suppose the collaborative sharing group consists of six peer participants and



(a) Collaborative Sharing Infrastructure Overview

(b) Operational Architecture of ShareEnabler Agent

Figure 6: ShareEnabler System Infrastructure and Architecture

each participant represents as a ShareEnabler agent. Agent 1 sends a broadcasting query message to all other peers (step 1). Upon receiving the query message, Peer Agents 2 - 6 look up their own posted contents. Agent 2 and 5 find the matched content, and both send a unicast query response with the metadata of the matched content to Agent 1 (step 2), while Agents 3, 4, and 6 are not necessary to respond to the requester. We call this process as metadata sharing. Agent 1 then sends a download request to Agent 5, and downloads the content from Agent 5 (step 3 - 4). We call this process as data sharing.

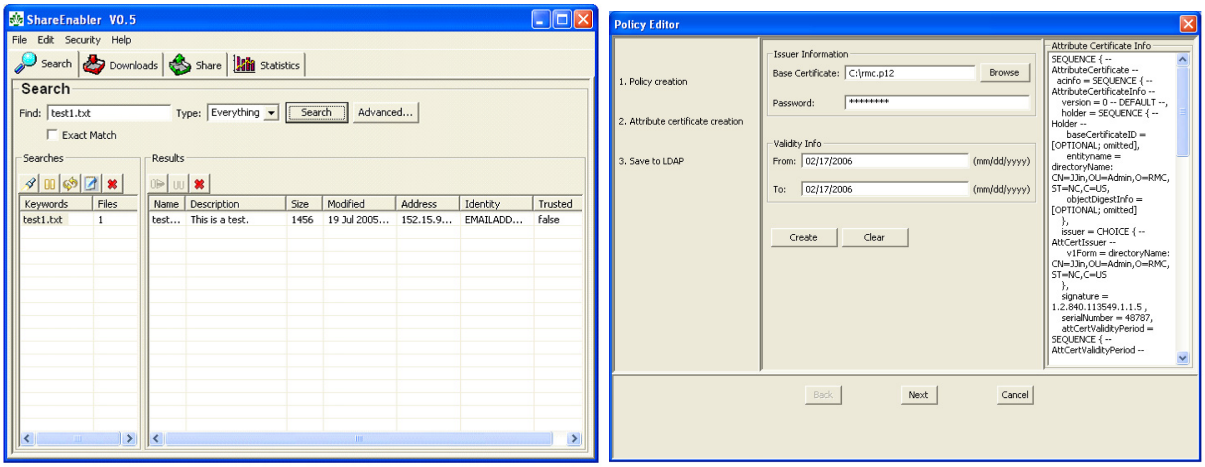
For security settings, X.509 certificate forms a major credential for each peer agent to authenticate itself to other agents in the collaborative sharing group. The certificate can be either self-signed or signed by a trusted organizational authority that the participant belongs to. The self-signed certificate is used by a new peer participant (called pseudo user) to join the community quickly. However, the pseudo user cannot gain higher level of trust or privileges in the system. The secure and reliable multicast communication is achieved by the combination of the InterGroup protocol [9] and the Secure Group Layer (SGL) [2], while the unicast communication security is achieved by TLS [8] when peers play the traditional role of client in some cases and the traditional role of a server in others.

Each ShareEnabler agent is composed of five components: graphical user interface (GUI), executive services, access management/enforcement, SGL/IG and TLS/ TCP. Figure 6(b) shows the interactions among these components in the context of metadata sharing on a pair of ShareEnabler agents as requester (ShareEnabler Agent 1) and responder (ShareEnabler Agent 2) respectively.

On the requester agent side, a user interacts with the GUI to specify the query keywords and set up the search criteria (step 1). GUI invokes the Search Service to formulate the query message and broadcast to all peers in the collaborative sharing group through SGL/IG (step 2 - 5). Meanwhile, the Search Service backs up the new query via Data Management Service (step 4.1). Upon receiving responses from other peers, TLS/TCP notices the Search Service with the response messages (step 6 - 7), and these responses are parsed and then shown in the GUI (step 8). The search results are backed up through Data Management as well (step 8.1).

On the responder agent side, the SGL/IG module notices the Metadata Sharing Service (step 1 - 2) upon receiving the file request. The Metadata Sharing Service invokes the Data Management Service to find matched resources against the query (step 3). The Data Management Service returns a list of matched resources to the Metadata Sharing Service and the PEP is invoked for access checking and enforcement (step 4 - 5). The PEP generates a standard XACML access request and sends it to the PDP for the access decision (step 6). The PDP retrieves relative ROA policies from the originator's LDAP directory and examine whether the requester is allowed to *query* the resource (step 7 - 9). Upon receiving the access decision from the PDP, PEP enforces the decision by removing unauthorized resources from the list and returns the new resource list to the Metadata Sharing Service (step 10 - 11). Finally, the Metadata Sharing Service formulates the response message and sends it back to the requester through the TLS/TCP protocol (step 12 - 13).

In our implementation, ROA policies are deployed separately from the major application and enforcement components. Therefore, an originator can easily maintain and change the policies without requiring changes to sharing ser-



Find: test1.txt

Type: Everything

☐ Exact Match

Keywords	Files
test1.txt	1

Name	Description	Size	Modified	Address	Identity	Trusted
test...	This is a test.	1456	19 Jul 2005...	152.15.9...	EMAILADD...	false

Base Certificate: C:\rmc.p12

Password: ********

From: 02/17/2006 (mm/dd/yyyy)

To: 02/17/2006 (mm/dd/yyyy)

```
SEQUENCE { --
AttributeCertificate --
  acinfo = SEQUENCE { --
AttributeCertificateInfo --
    version = 0 -- DEFAULT --,
    holder = SEQUENCE { --
Holder --
      baseCertificateID =
[OPTIONAL; omitted],
      entityname =
directoryName:
CN=JJin,OU=Admin,O=RMC,
ST=NC,C=US,
      objectDigestInfo =
[OPTIONAL; omitted]
    },
    issuer = CHOICE { --
AttCertIssuer --
      v1Form = directoryName:
CN=JJin,OU=Admin,O=RMC,
ST=NC,C=US
    },
    signature =
1.2.840.113549.1.1.5 ,
    serialNumber = 48787,
    attCertValidityPeriod =
SEQUENCE { --
AttCertValidityPeriod --
```

(a) New search and search results

(b) Attribute certificate creation

Figure 7: User Interfaces of ShareEnabler Agent and Administrative Policy Editor

vice systems. We decided to apply X.509 attribute certificates to encapsulate access management policies. X.509 attribute certificate (AC) is a basic data structure in Privilege Management Infrastructure (PMI) [1] to bind a set of attributes to its holder. With its portability and flexibility, AC is considered as an ideal container of subject attributes as well as authorization policies in ShareEnabler. We also developed a separate facility application, called Administrative Policy Editor, for an originator to create her ROA policies, generate policy attribute certificates, and store the ACs in LDAP directory.

In our prototype, we use JDK1.4 core packages as well as other necessary Java libraries to develop the components specified in the system architecture. Especially, we adopt SciShare's Reliable and Secure Group Communication (RSGC) package for the implementation of SGL/TLS communication protocol as well as the basic authentication mechanisms. We extended Sun's XACML implementation to accommodate the functionalities in the PDP. IAIK's java crypto library was used to implement major components of cryptography and attribute certificate. And the IPlanet Directory Server serves as the back-end LDAP policy repository. The beta version of ShareEnabler system implementation has been completed for further testing and evaluation. Figure 7 shows two user interfaces of ShareEnabler and Administrative Policy Editor for searching resources and attribute certificate generation, respectively.

6. RELATED WORKS

A number of authorization systems [15, 3, 19, 10, 4] have been developed to provide access control to shared resources in distributed environments. These systems diverse in various aspects and hardly address the high level access control models.

The Community Authorization Service (CAS) [15] framework is proposed to support group authorization in Grid communities. Every Grid community instantiates a CAS server representing the community and is controlled by a community administrator. The community administrator manages fine-grained authorization permissions among community users based on the community-specific trust relationships. Community members can access the available resources by obtaining individual credentials in the form of X.509 proxy certificates [20]. Another similar community-based authorization framework is realized in Virtual Organization Membership Service (VOMS) [3]. The VOMS-based system differs from the CAS framework in its representation of the community privileges. Both CAS and VOMS are designed for Grid communities, where the control strongly relies on the central CAS and VOMS servers, respectively. Their dependence on a pre-established community administrator prohibits them from supporting the structure and control-independent requirements of ad-hoc collaborative sharing.

The Akenti [19] system enforces access control on resources based on policies expressed by multiple distributed stakeholders. Akenti makes extensive use of X.509 public key certificates as the authorization token for encoding both user attributes as well as usage conditions. However, Akenti mainly supports discretionary access control (DAC) using ACLs and group based approach without considering hierarchical relationship between each group. PRIMA [10] privilege management framework is conceptually similar to Akenti and it allows multiple authorities to delegate access privileges. Resource privileges are expressed and distributed as privilege attributes to the users. The access to a resource enforced by PRIMA is based on the aggregated set of privilege attributes presented by the user.

SciShare [4] is a P2P-based sharing system and leverages X.509 certificates for authentication and authorization. However, SciShare only supports a limited features of Akenti and its DAC based approach cannot meet the unique requirements posed in the ad-hoc collaborative sharing environment. Our prototype adopts the relevant practices of Scishare introducing more flexible access management mechanisms.

7. CONCLUSION AND FUTURE WORK

In this paper, we have proposed an access management framework for ad-hoc collaboration including XACML-based policy framework. Our access management framework applies a role-based approach to incorporate special features of originator control, delegation and dissemination control. A prototype file sharing system, ShareEnabler, has been presented as a proof-of-concept implementation. The system applies a fully distributed approach both on authorization policies and policy enforcement mechanisms, while the standardized XACML policy specification and request/response messages achieve the consistent policy interpretation and decision making. The XACML based policy module is fully independent from the implementation and can be easily deployed or interoperated with other applications using the standard structure. The policy can also be bound to other standard transport protocols or mechanisms such as SAML [17, 12]. The ShareEnabler system serves as the client-side reference monitor that enables the platform-to-platform policy enforcement and propagation in distributed collaboration environments.

Our future work would be exploring access management solutions for more diverse collaborative resource sharing environments, supporting clusters, storage systems, and scientific instruments such as haptic devices and motion capturing tools. We believe our current work provides a solid foundation towards this direction.

8. ACKNOWLEDGMENTS

The work was partially supported by the grants from National Science Foundation (NSF-IIS- 0242393) and Department of Energy Early Career Principal Investigator Award (DE-FG02-03ER25565).

9. REFERENCES

[1] ITU-T Rec. X.509 ISO/IEC 9594-8. The directory: Public-key and attribute certificate frameworks, May 2001.

[2] D. Agarwal, O. Chevassut, M. R. Thompson, and G. Tsudik. An integrated solution for secure group communication in wide-area networks. In *Proc. of the 6th IEEE Symposium on Computers and Communications*, pages 22–28, July 2001.

[3] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, A. Gianoli, F. Spataro, F. Bonnassieux, P. Broadfoot, G. Lowe, L. Cornwall, J. Jensen, D. Kelsey, A. Frohner, D. Groep, W. S. de Cerff, M. Steenbakkers, G. Venekamp, D. Kouril, A. McNab, O. Mulmo, M. Silander, J. Hahkala, and K. Lhorentey. Managing dynamic user communities in a grid of autonomous resources. In *Proc. of Computing in High Energy and Nuclear Physics (CHEP03)*, 2003.

[4] K. Berket, A. Essiari, and A. Muratas. PKI-based security for peer-to-peer information sharing. In *Proc. of the Fourth IEEE International Conference on Peer-to-Peer Computing*, August 2004.

[5] D. Chadwick, T. Dimitrakos, K. K.-V. Dam, D. M. Randal, B. Matthews, and A. Otenko. Multilayer privilege management for dynamic collaborative scientific communities. In *Proc. of the Workshop on Grid Security Practice and Experience*, pages 7–14, July 2004.

[6] D. Ferraiolo, R. Sandhu, S. Gavrila, and R. R. Kuhn. Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security (TISSEC)*, 4:224–274, August 2001.

[7] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. *Lecture Notes in Computer Science*, 2150, 2001.

[8] The TLS protocol version 1.0. http://www.ietf.org/frc/rfc2246.txt.

[9] K. Berket, D. Agarwal, and O. Chevassut. A practical approach to the intergroup protocols. *Future Generation Computer Systems*, 18(5):709–719, 2002.

[10] M. Lorch and D. G. Kafura. The PRIMA grid authorization system. *Journal of Grid Computing*, 2(3):279–298, 2004.

[11] NIH guide: Transdisciplinary tobacco use research centers (rfa-ca-04-012). http://grants.nih.gov/grants/guide/rfa-files/RFA-CA-04-012.html, 2003.

[12] OASIS. XACML profile for SAML 2.0. Working draft 04, http://www.oasis-open.org/committees/download.php/8831/oasis-xacml-profile-saml-wd-04.pdf, August 2004.

[13] OASIS. Core and hierarchical role based access control (RBAC) profile of XACML v2.0. http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-rbac-profile1-spec-os.pdf, February 2005.

[14] OASIS. XACML 2.0 core: extensible access control markup language (XACML) version 2.0. http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf, February 2005.

[15] L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke. A community authorization service for group collaboration. In *Proc. of the 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY'02)*, June 2002.

[16] Uniform resource identifiers (URI): Generic syntax. http://rfc.net/rfc2396.html.

[17] OASIS. Security assertion markup language. www.oasis-open.org/committees/security/.

[18] R. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role based access control models. *IEEE Computer*, 29, February 1996.

[19] M. Thompson, A. Essiari, and S. Mudumbai. Certificate-based authorization policy in a PKI environment. *ACM Transaction on Information and System Security (TISSEC)*, 6(4):566–588, 2003.

[20] S. Tuecke, V. Welch, D. Engert, L. Pearlman, and M. Thompson. Internet x.509 public key infrastructure (PKI) proxy certificate profile. RFC 3820, 2004.