ShareEnabler: Policy-Driven Access Management for Ad-hoc Collaborative Sharing

Jing Jin[1], Gail-Joon Ahn[1], and Mukesh Singhal[2]

[1] Department of Software and Information Systems
University of North Carolina at Charlotte
Charlotte, NC 28223, USA
{jjin,gahn}@uncc.edu
[2] Department of Computer Science
University of Kentucky
Lexington, KY 40506, USA
singhal@cs.uky.edu

Abstract. The rise of the Internet has introduced dramatic changes in managing and sharing digital resources among widely dispersed groups. This paper presents a policy-driven access management approach for ad-hoc collaboration to enable secure information sharing in heterogeneous network environments. In particular, we attempt to incorporate the features of distributed role-based access control, delegation and dissemination control to meet the fundamental access control requirements associated with resource originators. These features are realized in a set of XACML-based Role-based Originator Authorization policies (ROA). We propose a security architecture, called ShareEnabler, to achieve effective authorization and enforcement mechanisms in the context of Peer-to-Peer (P2P) networking oriented file sharing. We briefly discuss our proof-of-concept prototype implementation based on an existing P2P file sharing toolkit developed by Lawrence Berkeley National Laboratory.

1 Introduction

The rise of Internet has led collaborators to face dramatic changes in managing and sharing their resources. Subsequently, it has extremely influenced to the traditional information sharing fashion. Firstly, collaborative information sharing has increasingly turned outward to connect distributed participants across enterprises and research institutes. By removing the barriers of the time and geographical distance from research collaborations, people are able to work together regardless of their locations. And new terms such as *virtual organization*, *virtual laboratory*, and *collaboratorium* have been introduced consequently. Also, the heterogeneous network environments demand more open and flexible infrastructures as well as system architectures to enable collaborative sharing. In addition, there is a need for ad-hoc collaborative sharing systems to support autonomous and spontaneous collaboration among diverse participants, minimizing administrative complexity.

Traditionally, collaborative information sharing heavily relies on client-server based approach or email systems. By recognizing the inherent deficiencies such as a central point of failure and scalability issue, several alternatives have been proposed to support collaborative sharing of resources, including Grid computing [1] and Peer-to-Peer (P2P) networking [2]. While Grid suits for highly structured collaborations with centralized infrastructures, P2P works well on heterogeneous network environments and promises to be more flexible and reliable for smaller ad-hoc collaborative interactions [3]. Especially, with the decentralized structure and load balancing feature, P2P based file sharing system offers better scalability and robustness. As demonstrated in the newly proposed SciShare system [3, 4] from Lawrence Berkeley National Laboratory (LBNL), P2P file sharing has great potentials to support collaborative sharing. However, most P2P technologies mainly focus on sharing services such as availability and scalability. Ad-hoc collaborative sharing requires the resource sharing be highly controlled and the confidentiality and integrity be properly protected during sharing sessions. On one hand, systematic techniques such as secure group communication protocols are needed to protect the communication traffic for each sharing session. On the other hand, access control mechanisms should be in place to allow resource owners, also called originators, to define and enforce access control policies for participating peers. Although some researchers have investigated secure group communication protocols and technologies [5–7], there are few attempts in exploring practical access control models and mechanisms for such environments. Our immediate motivation of this paper is to provide effective and practical policy-driven access management mechanisms for fulfilling access control requirements associated with ad-hoc collaborative sharing environments. Our approach emphasizes the originator as the principal source of policy to determine the collaboration control space and delegate fine-grained access capabilities to collaborators. The policy framework incorporates the features of distributed role-based access control, delegation and dissemination control. The policy enforcement system is then proposed to guarantee the policies being propagated and enforced appropriately.

The rest of the paper is organized as follows. In Section 2, we give an overview of motivation and background technologies. Section 3 introduces our access management framework, including the originator-initiated approach and role-based management framework followed by the underlying policy specification framework. We then realize the proposed policy framework in a concrete collaborative sharing example and show the detailed policy evaluation procedures. Our proposed ShareEnabler system and implementation issues are also discussed in this section. Section 4 concludes the paper.

2 Problem Statements and Background Technologies

To better understand the ad-hoc collaborative sharing environments, we proceed with a typical example of collaborative sharing [8], from which we identify

the key concepts involved in the environment and derive generic access control requirements for our approach:

NIH sponsored large-scale biomedical science collaborations involve a consortium of universities and research groups participating in several testbed projects related to the brain imaging of human neurological disease and associated animal models. Researchers from any of the groups may contribute their research results and data to be shared by other members in the collaboration group. Suppose Regional Medical Center (RMC), jointly initiated with Bioinformatics Department at University of XYZ, administers a local magnetic resonance imaging (MRI) data repository and would like to share the data with other collaborators for testing new hypotheses on human neurological diseases.

RMC needs to protect and control the data access and dissemination during the collaborative sharing. Due to the large group of collaborators, RMC would like to have a flexible and easy way to define the sharing collaborators as well as the access privileges for them. For faster and more convenient sharing, instead of contacting all the researchers in collaborating labs, RMC may need to notify the director or the coordinator of each collaborator's lab. The data are then shared with all other lab members through them. Furthermore, to protect the patentable data, any dissemination of the data should be under RMC's agreement. Meanwhile, Bioinformatics Department at University of XYZ as a co-owner of the data also would like to have the control on the data.

From the example above, we first identify several key concepts in an ad-hoc collaborative sharing environment that are used through the rest of this paper:

- **Originator**: In collaborative sharing environments, we refer the *resource owner* or the *initial information provider* as an originator. An originator plays a critical role in providing the resource to be shared and in controlling how the resource is shared among collaboration participants. The originator could be an individual principal or an organizational entity. For a particular resource, there may be one or multiple joint originators. In our scenario, RMC and XYZ University, both as organizational entities, act as joint originators for the MRI data repository.
- **Collaborative sharing space**: In general, collaborative sharing space refers to the *control domain* of the collaborative sharing. An originator needs to define her collaborative sharing space by including a collection of, mostly distributed, people or organizations and granting fine-grained access privileges to them. In our example, the whole NIH sponsored biomedical science consortium or a subgroup of consortium could be considered as the collaboration space. This should be determined at the originator's discretion.
- **Collaborator**: Each entity that is included inside the collaborative sharing space is referred as a collaborator. These collaborators are the actual *recipients* or *consumers* of the shared resource(s). Similar to the originator, a collaborator could be an individual principal such as independent researcher or an organizational research lab.

– **Disseminator**: We define two types of disseminators, namely, the *root disseminator* and the *designated disseminator*. The root disseminator refers to the originator since the originator triggers the initial sharing process with other ad-hoc collaborators. Designated disseminator, on the other hand, refers to a group of collaborators, with the consent of originator, to further distribute the resource. This can be achieved through the notion of delegation. Indeed, designated disseminator is a subgroup of ad-hoc collaborator. In our case, the directors/coordinators of collaboration laboratories are the designated disseminators.

2.1 Access Control Requirements

From the above-mentioned example, we derive several generic access management requirements for ad-hoc collaborative sharing:

– **Flexible and manageable access control**: Collaborative sharing may involve a large amount of distributed collaborators across domains. The diversity and unpredictability of the involved participants determine that the authorization cannot be established on per-individual basis like the way ACL does. The access control system needs to provide appropriate abstraction of collaborators and privileges to achieve the flexibility and reduce the complexity of security administration.
– **Flexible delegation/revocation**: The nature of distributed resource sharing requires delegation in place to allow the access privileges as well as administrative responsibilities of an originator to be distributed among different collaboration parties. Especially, it should also allow an originator to delegate not only all of the privileges, but also partial privileges. In addition, revocation as the counterpart of delegation should be supported as well.
– **Effective originator-controlled dissemination**: As the shared information leaves the originator's domain, it is hard for the originator to have control on such information. With the originator-initiated control, originators should be able to control and track down the re-dissemination of their resources to make sure the dissemination happens within the collaborative sharing space, and only the legitimate collaborators could share the resources.

2.2 Background Technologies

Role-based Access Control (RBAC): RBAC is a proven technology for managing and enforcing security in large-scale and enterprise-wide systems [9, 10]. The essential idea of RBAC is that permissions are associated with roles, and users acquire permissions by being members of appropriate roles. With the abstraction between users and permissions, RBAC could tremendously reduce the complexities of security management for system administrators. Meanwhile, many role-based delegation models [11–13] have been proposed as a complementary to RBAC in leveraging an effective way of propagating authorities as well as

responsibilities among various distributed entities. Our framework is built on existing role-based delegation models by applying decentralized user assignments.

ORCON, UCON and DCON: Originator control (ORCON) is a special access control policy defined by a resource originator to control the dissemination of restricted resources [14, 15]. ORCON policy requires that resource recipients obtain an originator's permission to re-disseminate protected resources to users who are not originally designated as authorized recipients by the originator. Traditional ORCON solutions used a non-discretionary access control list, which limits the ability to enforce ORCON policies in a closed centralized control environment [16]. The concept of Usage Control (UCON) is introduced in [16, 17] for controlling access and usage of digital information objects. The re-dissemination control in ORCON is also one of the key concerns in UCON. By introducing license and ticket [16], UCON has the potential to support and enforce ORCON policies in more versatile and flexible ways for distributed environments. Most recently, the notion of dissemination control (DCON) has been proposed in [18]. DCON involves a much richer and broader concept than ORCON and UCON concerning with controlling information during the dissemination activities.

SciShare File Sharing Infrastructure: Traditional P2P sharing applications, such as Gnutella [19], allow end users to search and download information from other peers, and make their own information available to other peers. The search component often broadcasts a query to all known peers, while sending response and downloading information are unicast communications. LBNL's framework, called Scishare [4], is a security enhanced version of P2P file sharing system. SciShare leverages X.509 public key certificate as the central security component. The certificate can be either self-signed or signed by a trusted organizational certification authority (CA). To facilitate new peers joining the community quickly, the system allows the new peers (called pseudo user) to create self-signed X.509 certificates. However, the pseudo user cannot gain higher level of trust or privileges in the system. The instantiation of secure and reliable multicast communication is provided by Secure Group Layer (SGL) [5], while TLS [20] is used to achieve confidentiality and integrity in unicast communication when peers play traditional role of client in some cases and the traditional role of a server in others. SciShare also supports access control that is primitive and limited to group-based discretionary access control approach.

3 Policy-Driven Access Management Framework

In this section, we describe a policy-driven access management framework to provide a means of comprehensive access management model beyond SciShare. The framework emphasizes originator-initiated role-based access control and delegation. Originators dynamically create and include roles in their collaborative sharing space while delegating fine-grained access and dissemination capabilities to the roles. Distributed role-assignment is achieved through *Delegation of*

Delegation Authority. These features are expressed in a set of Role-based Originator Authorization policies (ROA). ROA policies serve as the foundation of our framework and are further evaluated and enforced in our proposed security architecture for P2P based file sharing.

3.1 Supporting Originator Control and Role-based Approach

An originator, as the resource owner, is responsible for initiating the controls to secure her respective resources over sharing and dissemination activities among other peer collaborators. To accommodate the originator-initiated control approach, it is essential for an originator to define her collaborative sharing space in a set of access management policies and delegate fine-grained privileges through these policies. The specified policies should be propagated and enforced properly by the underlying security system during the resource re-dissemination.

RBAC provides an effective way to abstract privileges using roles. Instead of including every individual ad-hoc collaborator, the originator could simply define the collaborative sharing space in a collection of specific roles, such as "engineer" and "investigator". And each peer collaborator is dynamically included in the sharing space to gain access privileges by claiming their role. Therefore, bringing "role" in our framework becomes a natural choice to achieve the manageability in the ad-hoc collaboration environments. In addition, we introduce role-based delegation as another layer of privilege and authority decentralization to accommodate the needs of distributed role assignment and fine-grained privilege propagation in collaborative sharing environments. In particular, our framework incorporates the following types of delegation relationships for ad-hoc collaboration:

- **Delegation of access capabilities**: The permission-role assignment in traditional RBAC usually deals with the abstraction of privileges in a closed organizational domain. In a distributed collaborative sharing environment, an originator delegates fine-grained access capabilities to certain roles in the collaboration space so that the privileges are propagated and distributed across various participating entities through these roles.
- **Constrained dissemination delegation**: To achieve better resource availability and continuous resource dissemination, besides the normal access privileges, the resource dissemination privilege can be delegated by an originator to a certain set of roles, so that the collaborators who are assigned to these roles are allowed to further disseminate the pre-obtained resources on the originator's behalf. These collaborators, in another words, are the designated disseminators. As "constrained" delegation, the scope of delegation should be within the originators and the designated disseminators.
- **Delegation of delegation authority**: This is a special form of administrative delegation that enables an originator to partially delegate the role assignment privilege to trusted third parties. In our example, the originator defines a set of roles in her collaborative sharing space and delegates the role assignment authority to the directors/coordinators of each collaboration

group so that these directors/coordinators may assign roles to their members on the originator's behalf.

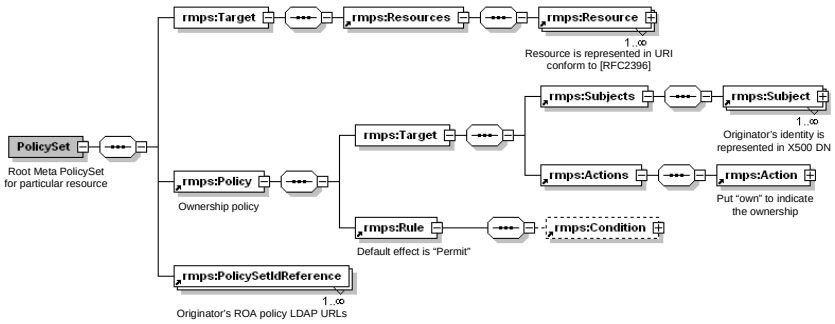
3.2 Designing Policies

In our policy framework, an originator defines her access management policies in a set of authorization policies using XACML (eXtensible Access Control Markup Language) [21]. We introduce two major types of policies, Root Meta PolicySet (RMPS) and Role-based Originator Authorization PolicySet (ROA).

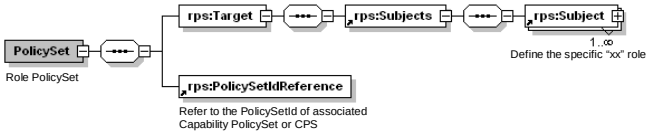
Root Meta Policy Set (RMPS) is the starting point of the originator's authorization policies for the shared resources. Since the shared resources may have single or multiple distributed originators, their authorization policies may be maintained in multiple administrative domains. We need a policy to identify these originators and locate their policies so that the underlying enforcement system could retrieve and enforce these distributed policies. RMPS is designed for this purpose. In RMPS policy schema, the *Target* element specifies the resource to which ROA authorization policies are applied. The resource is represented as a URI that conforms to RFC2396 standard format [22]. The *PolicySet* contains one ownership *Policy* and one or more *PolicySetIdReference* elements to specify ROA policy locations in the format of LDAP URLs. In the ownership *Policy*, originators are identified as *Subject* attributes in their X.500 DNs. The ownership is specified as "*own*" in *Action* element. Figure 1(a) illustrates the schema of RMPS.

ROA policy sets are the real role-based authorization policies where an originator defines her collaborative sharing space and delegates fine-grained capabilities. We extend OASIS RBAC profile [23] to support the delegation and distributed role assignment in our framework. ROA contains four major sub-components: role specification policy (RPS), capability specification/role-capability assignment policy (CPS), user-role assignment policy (RAPS), and delegation of delegation authority policy (DoDPS).

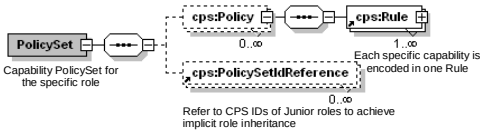
- Role *PolicySet* (RPS) is a role specification *PolicySet*. The originator defines the collaborative sharing space in a set of RPSs, and associates each role RPS with a Capability *PolicySet* (CPS) that actually contains capabilities of the given role. The role is specified as a *Subject* attribute, the corresponding CPS is referenced through *PolicySetReference*. Figure 1(b) shows the schema of RPS.
- Capability *PolicySet* (CPS) specifies the actual capabilities assigned to the given role. CPS contains *Policy* and *Rule* elements that describe the delegated capabilities as the resources and actions. By granting the "*disseminate*" action to a specific role, the originator delegates her dissemination privilege to the role. The collaborator who is assigned to that role then becomes a designated disseminator to re-disseminate the resources. The CPS may also contain references to the CPSs associated with other roles that are junior to the given role, thereby achieving the role hierarchies through the capability aggregation. Figure 1(c) shows the schema of CPS.



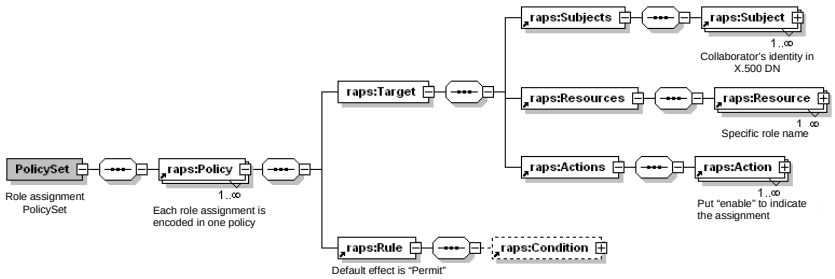
(a) RMPS Policy Schema



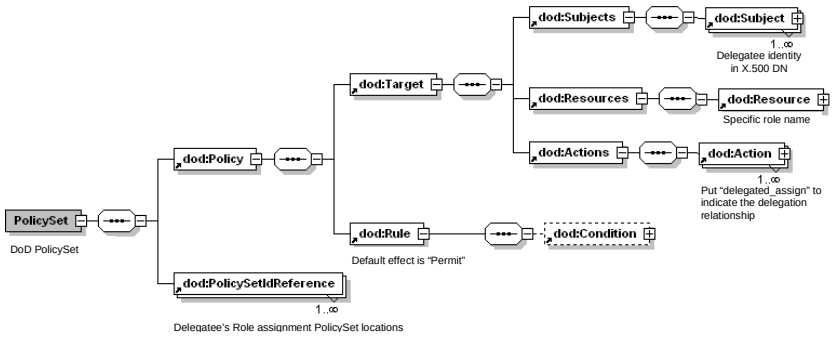
(b) RPS Policy Schema



(c) CPS Policy Schema



(d) RAPS Policy Schema



(e) DoDPS Policy Schema

– Role Assignment *PolicySet* (RAPS) is specified by an originator or a delegated third party authority to define which roles are assigned to which collaborators. In RAPS, the principals are specified in their X.500 DNs as *Subject* attributes. The assigned role is specified as *Resource* attribute. And the term"*enable*" is used as *Action* attribute to indicate the assignment relationship. Figure 1(d) shows the schema of RAPS.
– Delegations of Delegation Authority *PolicySet* (DoDPS) reflects the type of "*delegation of delegation authority*" with originators specifying which role assignments are delegated to which specific trusted authorities. The construction of DoDPS is similar to RAPS, except that the *Subjects* are the trusted delegatees and the delegation relationship is indicated as "*delegated_assign*" in *Action*. Figure 1(e) shows the schema of DoDPS.

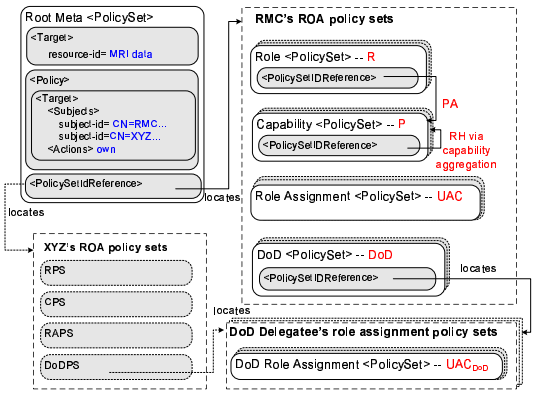
3.3 Policy Framework Realization and Policy Evaluation

In this section, we extend the earlier discussed collaborative sharing scenario into a concrete example and proceed implementing a set of access management policies to realize our proposed policy framework. We then show how the authorization system evaluate these policies and make decisions.

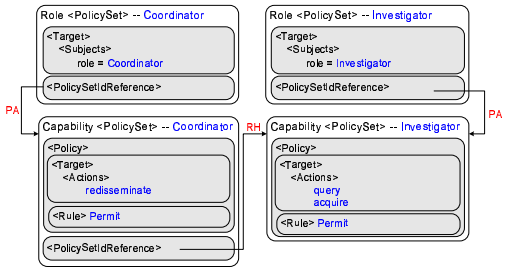
Inside the NIH biomedical science research community, a team of biologists from LIISP research lab, with John as the team leader and Dave as one of the team members, is conducting research tasks related to animal modeling comparisons and analysis. John's team needs to collaborate with RMC and use RMC's data to verify a new hypothesis drawn from their research.

As discussed earlier, both *RMC* and *XYZ University* are joint originators for the *MRI data* resource. For simplicity, we focus on how RMC develops the ROA policies and omit the control from the XYZ University. End each individual member in *LIISP* lab, *John* and *Dave*, is considered as an ad-hoc collaborator that needs to be authorized individually in *RMC*'s collaborative sharing space. To authorize accesses to the members in *LIISP* lab, *RMC* defines two roles such as *Coordinator* role and *Investigator* role, where the *Coordinator* role is senior to the *Investigator* role. *RMC* delegates the capabilities of "*query*" and "*acquire*" to the *Investigator* role, and further delegates the capability of "*re-disseminate*" to the *Coordinator* role. *RMC* then assigns the team leader *John* to the *Coordinator* role and delegates *John* to perform the user-*Investigator* role assignment through the delegation of delegation authority, so that *John* is able to assign his other team members (i.e. *Dave*) to the *Investigator* role and re-disseminate the resource to them as a designated disseminator.

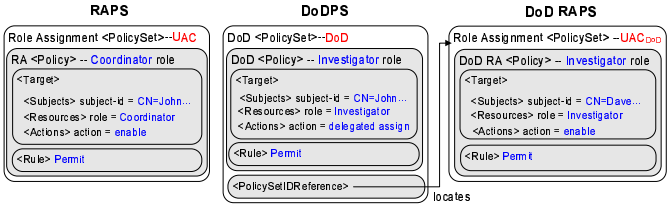
Figure 2(a) shows the overall structure of our policy framework and the relationships among the individual policy components. In particular, RMPS specifies the resource with the originator(s) who "*own*" the resource, and locates the originators' ROA policy sets. In the example scenario, *RMC* and *XYZ University*, both represented as *Subject* attributes in their X.509 DNs, are joint originators of the "*MRI data*" resource. Since we focus on the control of *RMC*, only the URL



(a) Overall Policy Framework



(b) RPS-CPS Details



(c) DoDPS-RAPS Details

Fig. 2. Policy Framework Realization

location of *RMC*'s ROA policy sets is referenced through the *PolicySetIdReference* element. In RMC's ROA policies, there is a set of RPSs and CPSs, a RAPS and a DoDPS. As shown in Figure 2(b), RPSs define two roles in RMC's collaborative sharing space, namely the *Coordinator* role and the *Investigator* role.

CPSs specify the corresponding capabilities associated with these two roles. The reference link between each pair of RPS and CPS reflects the permission-role assignment relation where the originator delegates the fine-grained access and dissemination capabilities to the role. By referencing to the CPS of the *Investigator* role, the *Coordinator* role inherits all the capabilities that are assigned to the *Investigator* role. In this context, the role hierarchy is achieved indirectly through capabilities aggregation. As shown in Figure 2(c), the RAPS specifies the user-role assignment relation that RMC assigns *John* to the *Coordinator* role. By being assigned to the role, *John* is included in RMC's collaborative sharing space, and thus obtains the delegated capabilities. DoDPS realizes the *delegation of delegation authority* where RMC delegates the user-*Investigator* role assignment to *John*. And *John*'s RAPS policy is finally referenced in the DoDPS where *John* assigns his team member *Dave* to the *Investigator* role.

As our policy framework conforms to the XACML standard, the policy evaluation and authorization decision making can be done as specified in [21]. The typical setup is that the Policy Enforcement Point (PEP) forms an access request based on the requester's attributes (X.509 identities, roles, etc.), the resource in question, and the action towards the resource. The request is sent to a Policy Decision Point (PDP) for policy retrieval and policy evaluation. Basically, the PDP first finds the top-level policy elements that the *Target* elements match the attributes specified in the access requests, and then evaluates the boolean expressions included in each *Rule* elements and finally combines the results using the specified policy combination algorithms. A response with an access *Decision* element of value "*Permit*", "*Deny*", "*Indeterminate*" or "*NotApplicable*" will be made and returned to the PEP for further authorization enforcement. In our system, we introduce the Context Handler as a subcomponent of the PDP to conduct a series of query-generation and decision-making process for a single access query sent by the PEP. In this section, we focus on how the PEP, Context Handler and PDP interact with each other and how the PDP evaluates an access request against the originator's ROA policies. The detailed system design and implementation will be discussed shortly in next section.

Figure 3 shows the detailed sequence diagram of the policy retrieval and policy evaluation. The PEP formulates an access request with the requester's X.509 identity and the action towards the requested resource. For instance, the PEP may generate an access request for the PDP to evaluate whether a requester Dave (*CN=Dave...*) is allowed to "*acquire*" the "*MRI data*" resource. Along with the associated RMPS for the "*MRI data*", the request is sent to the PDP. The Context Handler parses the RMPS and locates *RMC*'s policy directory. The role attributes that are assigned to the user's identity are retrieved from the originator's policy repository. In our case, the *Investigator* role is assigned to *Dave* by *John*. Since the attribute is assigned by an entity other than the originator, the Context Handler will prompt to formulate a DoD request for the PDP to evaluate whether the role attribute issuer (*CN=John...*) is a legitimate delegated authority to conduct the user-*Investigator* role assignment. The PDP Engine conducts the ***DoD Evaluation*** based on DoDPS and confirms the dele-

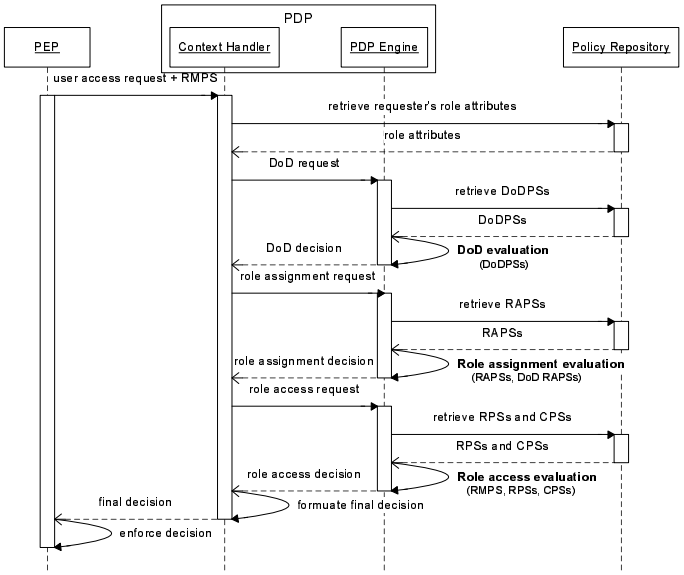


Fig. 3. Policy Retrieval and Evaluation

gation of delegation authority relationship. The Context Handler then formulates the role assignment request for the PDP Engine to check whether the requester (*CN=Dave...*) is "*enabled*" with the *Investigator* role attribute that is retrieved earlier. The PDP Engine conducts the ***Role Assignment Evaluation*** against the retrieved RAPSs defined by the originator and/or the DoD RAPSs defined by the DoD delegatee (in our case, only the DoD RAPS is evaluated). Finally, the Context Handler formulates the role access request for the PDP Engine to check whether the assigned *Investigator* role is able to perform the "*acquire*" action towards the "*MRI data*" resource as specified in the PEP's access request. The PDP Engine conducts the ***Role Access Evaluation*** against the RMPS, RPSs and CPSs. Based on the authorization decisions of these three evaluations, the Context Handler generates the final user access decision and sends back to the PEP for further decision enforcement process.

3.4 ShareEnabler System Architecture and Discussions

In this section, we give an overview of our system architecture, called ShareEnabler. ShareEnabler casts our proposed framework as detailed authorization services and mechanisms which are bound to specific communication infrastructure from LBNL's SciShare toolkit [4].

In our collaborative sharing system, each participant is represented by a ShareEnabler (SE) agent that executes sharing services on the collaborator's behalf. Similar to most of existing P2P file sharing systems, the resource discovery involves broadcasting a query to all known peers. As shown in Figure 4, Sha-

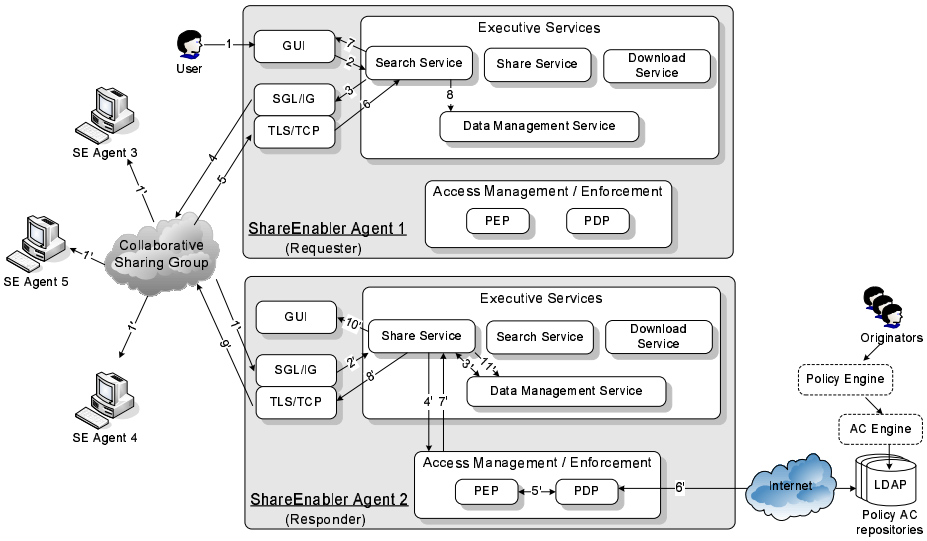


Fig. 4. ShareEnabler System Architecture

reEnabler Agent 1 sends a broadcasting query message to all known peers in the collaborative sharing group. Upon receiving the query message, SE Agents 2 - 5 look up their own posted contents. SE Agent 2 finds the matched content(s), evaluates the originator's ROA policies and sends a unicast query response with the metadata of the authorized content(s) to SE Agent 1, while SE Agents 3 - 5 are not necessary to respond to the requester. We call this process as metadata sharing. SE Agent 1 then can send out the download request, while the SE Agent 2 will further check with the originator's ROA policies and initiate the data transferring process if the requester is authorized to download to resource.

Figure 4 also shows the detailed components inside the ShareEnabler Agent and their interactions in the process of metadata sharing between the SE Agent 1 (as the requester) and the SE Agent 2 (as the responder). Each ShareEnabler agent is composed of five components: Graphical User Interface (GUI), Executive Services, Access Management/Enforcement, SGL/IG and TLS/TCP. GUI is the interface through which the user operates and executes the sharing services. Executive Services are the real services required by collaborative sharing behaviors, which include Search, Download and Share Services. All these services are based on the underlying Data Management Service, which provides data storage and cache functionalities. The Data Management Service also serves as the background database in the system. The Access management/enforcement is the central component for the core access and dissemination control. The PEP is responsible for the request processing and access decision enforcement. The PDP, which consists of the Context Handler and the PDP Engine, is designed for the policy retrieval and authorization decision making. Secure Group Layer (SGL) and the underlying InterGroup protocol provide the secure group communica-

tion services. Similarly, Transport Layer Security (TLS) and the underlying TCP protocol provide the secure communication between two ShareEnabler agents, which in the category of unicast communication. The functionalities for both SGL/IG and TLS/TCP are adopted from SciShare [4].

In the context of metadata sharing, on the requester agent side (ShareEnabler Agent 1), a user interacts with the GUI to specify the keywords and search criteria (step 1). GUI invokes the Search Service to formulate the query message and broadcast to all known peers in the collaborative sharing group through the SGL/IG protocol (step 2 - 4). Upon receiving responses from other peers, the TLS/TCP module notices the Search Service with the response messages (step 5 - 6), and these responses are parsed and then shown in the GUI (step 7), through which the user may further interact to download the data resource. The search results are finally cached through the Data Management Service (step 8).

On the responder agent side (ShareEnabler Agent 2), the SGL/IG module notices the Sharing Service (step 1' - 2') upon receiving the request. The Sharing Service then invokes the Data Management Service to find matched resources against the query (step 3'). When a list of matched resources is returned back to the Sharing Service, the Access Management/Enforcement component is invoked for access checking (step 4' - 6'). The PEP enforces the decision by removing unauthorized resources from the list and returns the updated list back to the Sharing Service (step 7'). Finally, the Sharing Service formulates the response message with the metadata of a list of matched and authorized resources, and sends back to the requester through the TLS/TCP module (step 8' - 9'). The metadata sharing result is shown in the GUI and cached in the Data Management Service (step 10' - 11').

As also shown in Figure 4, ROA policies are deployed separately from the major ShareEnabler application and its enforcement components. These ROA policies will be retrieved and enforced at run time whenever the ShareEnabler agent needs to respond to other peer's requests. In doing so, an originator can easily maintain and change the policies without requiring changes to sharing service systems. We decide 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) [24] 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 Policy Administration Facility application to provide the utility modules for originator to create and maintain ROA policies. Especially, originators use the *Policy Engine* to create their ROA policy sets. *Attribute Certificate Engine* is then invoked to generate the ROA policy ACs and store them in distributed LDAP policy repositories.

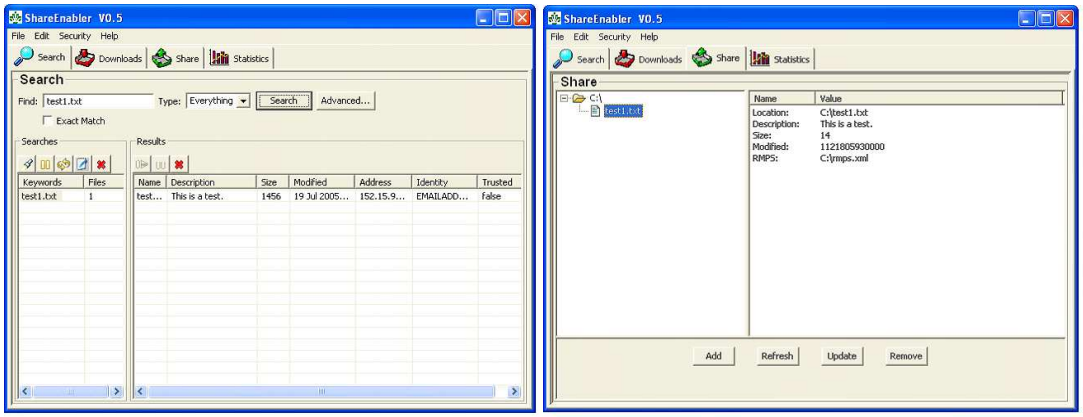
The goal of access and dissemination control of ShareEnabler is to guarantee the resource is shared within the collaborative sharing space defined by ROA policies. Our system applies a distributed policy propagation and enforcement scheme with decentralized, self-enforcing, and self-monitoring features at each ShareEnabler agent level. Especially, each disseminator ShareEnabler agent

should ensure that ROA policies are enforced locally by the Access Management/Enforcement component, and these ROA policies are propagated to other ShareEnabler agents while those agents may further act as disseminators to respond to other peers' requests. Since the Root Meta Policy Set (RMPS) plays an important role for the ShareEnabler Agent to locate and enforce originator's policies. It is essential to make sure the RMPS is propagated along with the data dissemination and the confidentiality and integrity are properly protected . In achieving these goals, we design a new data structure that strongly encapsulates the data resource together with the associated RMPS policy. As the originator initiates the sharing process, instead of sending out the original data resource, originator's ShareEnabler agent disseminates the encapsulated data structure to other agents, which can only be decrypted at runtime by the ShareEnabler Agent. By doing this, we leave the ShareEnabler Agent with full enforcement power and make it extensible for more advanced dissemination tracking mechanisms.

In our prototype, we use JDK1.4 core packages as well as other necessary 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 extend Sun's XACML implementation to accommodate the functionalities in PDP and PEP. IAIK's java crypto library is used to implement the major components of cryptography and attribute certificate. And 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 5(a) shows a user interface of an SE Agent for searching for specific file resource and displaying search results based on the responses from other peers. Figure 5(b) shows an originator posts new resource to be shared with other collaborator peers. The Figure 5(c) shows the user interface of the policy creation that allows an originator to create new roles in her collaborative sharing space and delegate fine-grained capabilities to the roles. The ROA policies will then be generated automatically based on the originator's input. Finally, Figure 5(d) shows the interface of policy attribute certificate generation with the originator specifying the validity period of the attribute certificate and using her private key (encapsulated in an X.509 Personal Information Exchange Certificate [25]) to sign the attribute certificate.

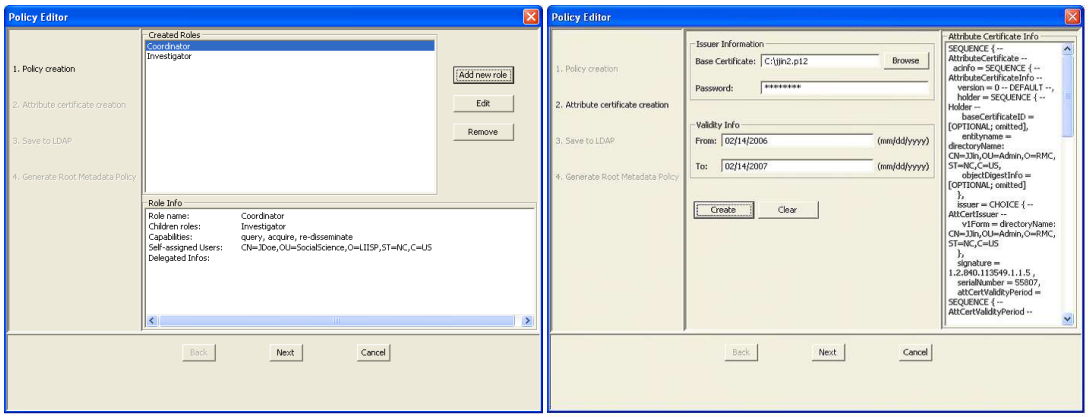
4 Conclusion

In this paper, we have presented a policy-driven access control framework for ad-hoc collaborative sharing. Especially, we articulated distinctive access control requirements in ad-hoc collaborative sharing and proposed a family of XACML-based policy schemas that are comprehensive and flexible enough to meet the identified requirements. In addition, we briefly described the enforcement mechanisms as well as a proof-of-concept prototype of P2P based file sharing system, called ShareEnabler. An important contribution of this work includes special fea-



(a) New Search and Search Results

(b) Post New Resource



(c) Policy Creation

(d) Attribute Certificate Generation

Fig. 5. ShareEnabler User Interfaces

tures of originator control, delegation and dissemination control. Our approach allows originators to authorize distributed collaborators and control over the resources being shared. The delegation of delegation authority was introduced to systematically achieve user-role assignments in distributed environments.

Our future works are geared towards several directions. We would investigate and apply more advanced system-level dissemination control enforcement mechanisms. In collaborative sharing environment, the resources are stored and updated in distributed places. This causes another control issue of how to maintain the originator-initiated control of data usage and modification after the dissemi-

nation, which in turn, relates to the enforcement mechanisms. Furthermore, the inconsistency of data representation and instances needs to be dealt with while the resources are shared and updated. Developing an integrated infrastructure would be another research direction as well.

5 Acknowledgments

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

References

1. Baker, M., Buyya, R., Laforenza, D.: The Grid: International efforts in global computing. International Journal of Software Practice and Experience. (2002)
2. Oram, A. (ed.): Peer-to-peer: Harnessing the power of disruptive technologies. O'Reilly. (2001)
3. Berket, K., Agarwal, D.: Enabling secure ad-hoc collaboration. In: Proceedings of the Workshop on Advanced Collaborative Environments. (2003)
4. Berket, K., Essiari, A., Muratas, A.: PKI-based security for peer-to-peer information sharing. In: Proceedings of the Fourth IEEE International Conference on Peer-to-Peer Computing. (2004)
5. Agarwal, D., Chevassut, O., Thompson, M.R., Tsudik, G.: An integrated solution for secure group communication in wide-area networks. In: Proceedings of the 6th IEEE Symposium on Computers and Communications. (2001) 22–28
6. Kihlstrom, K.P., Moser, L.E., Melliar-Smith, P.M.: The securering protocols for securing group communication. In: Proceedings of 31st IEEE HICSS. (1998) 317–326
7. Reiter, M.K.: Secure group membership protocol. In: Proceedings of IEEE Symposium on Research in Security and Privacy. (1994)
8. NIH: NIH data sharing workbook. http://grants.nih.gov/grants/policy/data_sharing/data_sharing_workbook.pdf (2004)
9. Sandhu, R., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role based access control models. IEEE Computer **29** (1996)
10. Ferraiolo, D., Sandhu, R., Gavrila, S., R. Kuhn, R.: Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security (TISSEC) **4** (2001) 224–274
11. Zhang, L., Ahn, G.J., Chu, B.T.: A rule-based framework for role-based delegation and revocation. ACM Transactions on Information and System Security (TISSEC) **6** (2003) 404–441
12. Ahn, G.J., Mohan, B.: Secure information sharing using role-based delegation. Journal of Network and Computer Applications **2** (2005)
13. Barka, E., Sandhu, R.: Framework for role-based delegation models. In: Proceedings of the 16th Annual Computer Security Applications Conference (ACSAC), IEEE Computer Society (2000) 168

14. Abrams, M.D., Heaney, J., King, O., LaPadula, L.J., Lazear, M., Ol, I.M.: Generalized framework for access control: Towards prototyping the orgcon policy. In: Proceedings of the 14th National Computing Security Conference. (1991) 257–266
15. McCollum, C.J., Messing, J.R., Notargiacomo, L.: Beyond the pale of MAC and DAC — defining new forms of access control. In: Proceedings of IEEE Symposium on Security and Privacy. (1990) 190–200
16. Park, J., Sandhu, R.: Originator control in usage control. In: Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY'02). (2002)
17. Park, J., Sandhu, R.: Towards usage control models: beyond traditional access control. In: Proceedings of the 7th ACM Symposium on Access Control Models and Technologies (SACMAT 2002). (2002) 57–64
18. Thomas, R., Sandhu, R.: Towards a multi-dimensional characterization of dissemination control. In: Proceedings of the 5th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY04). (2004)
19. Gnutella. http://www.gnutella.com/
20. RFC2246: The TLS protocol version 1.0. http://www.ietf.org/frc/rfc2246.txt (1999)
21. 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 (2005)
22. RFC2396: Uniform resource identifiers (URI): Generic syntax. http://rfc.net/rfc2396.html (1998)
23. 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 (2005)
24. ITU-T: The directory: Public-key and attribute certificate frameworks. ISO/IEC 9594-8:2001 (2001)
25. RSA: PKCS #12: Personal information exchange syntax standard. ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-12/pkcs-12v1.pdf (1999)