Abstract

Scientific repositories promote information sharing at a global scale. However, users will share sensitive resources in such repositories only if they are trustable. In addition, they must provide intuitive mechanisms to manage who can access to resource collections. Current approaches, which rely on a central administration, are not flexible and do not scale to cope with large numbers of users. Moreover, discretionary access control is a legal standard for health-related resources. This work proposes a group-based discretionary access control model with decentralized permission and group management. The model also differentiates permissions for data and meta-data, enabling the sharing of meta-data while protecting sensitive data. The access control model is introduced and formalized using Barker’s Unifying Metamodel. We also describe how the model has been implemented in the Epidemic Marketplace, an open software information platform for epidemic studies, designed to foster cooperative behavior and data sharing.

Keywords: Access Control; Discretionary; Group-Based; Information Sharing

* Corresponding author.
E-mail address: epiwork@lasige.di.fc.ul.pt.
1. Introduction

The recent explosion in produced data in every field of science has resulted in the emergence of highly data driven sciences. These sciences, in turn, also generate new data resources adding to the increasing complexity of storing, managing and sharing this data. Therefore, in order to cope with this "data tsunami", adequate information platforms are necessary that can deal with the increased volume, scale and complexity related to managing, preserving and sharing these data.

In 2010, the Riding the Wave report identified the need for an e-infrastructure for data management enabling seamless access, re-use and trust of data [1]. The authors present challenges to improve scientific discovery and support collaboration across disciplinary and geographical boundaries. These challenges include: data preservation and curation; linking people and data; describing data through adequate metadata and semantics for data discovery; enabling interoperability and data exchange across scientific domains; and establishing trust through adequate authentication and authorization platforms.

The Epidemic Marketplace (EM) is an information platform that addresses the above challenges in the epidemiology domain [2]. It stores and manages health-related resources, including sensitive data, such as epidemic incidence datasets.

An additional challenge in epidemiology is that the generated data is often sensitive in nature and must therefore be under adequate access control mechanisms. While some previous approaches hint towards giving resource owners more control over their resources, when managing sensitive health-related resources it is the standard in practice to give this responsibility to the resource owner, and in some countries a legal requirement [3,4].

The EM, which has been publicly available since September 2010, is built on open source technologies. The back end is based on a Fedora Commons repository [5] for storing and managing resources and a Lightweight Directory Access Protocol (LDAP) server [6] for user management. The EM also includes a Drupal-based front end [7] and web services to access repository content, both by the front end and external applications.

In this paper, we explore the implementation of access control mechanisms from these technologies enabling users to share their created resources while protecting sensitive data. We identify the access control requirements of an epidemiological resource repository and propose a group-based approach to access control over repository resources, which could also be used in similar scientific environments.

We adopt a decentralized and discretionary approach over permission assignment as well as in the management of user groups. Additionally, our access control model includes an object structure that separates data from meta-data, enabling the search of resources without exposing sensitive data. We evaluate the applicability of the proposed model on the EM, built from open source technologies showing it is a feasible solution for information platforms.

The remainder of the paper is organized as follows: Section 2 provides insight into previous approaches to access control; Section 3 is an assessment of conceptual requirements for an epidemiological information platform; Section 4 presents the proposed group-based access control model; Section 5 describes its implementation and evaluation in the Epidemic Marketplace; Section 6 presents the conclusions as well as challenges that remain to be explored.

2. Access Control

The access control model is defined through the use of relations between the several entities. We use Barker's meta-model to meet the access control requirements of the Epidemic Marketplace through a group-based discretionary access control model [8]. This model was proposed to express access control primitives common to access control models. The meta-model can aid in expressing access control models in similar terms and extend them to express further and even broader access control needs. The broader, abstract, entities
used in Barker's model make it possible to describe a wider variety of access control models, rather than just
describing an existing problem. Barker considers sets of categories ($C$), principals ($P$), actions ($A$), situational
identifiers ($S$), events ($E$) and resources ($R$), which other access control models typically refer to as objects.

One of the main entities used by Barker is the category. This abstract entity can describe multiple things
such as types, roles or groups of users. However, it is not restricted to users and can be used to categorize other
entities.

In Discretionary Access Control (DAC) models, object owners define the permissions for their objects [9].
Access control is defined by a set of access permissions granted to each user to perform operations on a
specific object. Of particular note are Ownership DAC models, where each object has an owner who can set
permissions, and Decentralized DAC models, where the owner can delegate the ability to set permissions to
other users.

Assigning policies for an object to individual users is complex and inappropriate for large scale systems
because it requires a high degree of maintenance to keep policies updated. One approach to overcome this
problem was to introduce user groups to DAC and assigning permissions to these groups instead of individual
users [10,11,12]. However, in most implementations, these groups are administrator controlled and therefore
may not be sufficiently flexible to portray the sharing needs of a scientific repository where contacts between
contributing users are not restricted to groups easily identifiable by system administrators.

Role-Based Access Control (RBAC) is a wide-spread and accepted approach to access control [13]. While
its implementations vary greatly, most are based on the same principles and concepts. These can be extended
globally to the implementation's needs. The main distinction of RBAC from other access control models is
the introduction of the Role concept. Users in RBAC are associated with roles to which permissions are
assigned. Users receive permissions indirectly through their associated roles. Roles can represent tasks,
responsibilities and qualifications within an organization, e.g. doctor, researcher or student [14]. A single user
can be assigned to many roles and a single role can have multiple users.

Scientific online repositories and other collaborative environments have adapted several access control
models to suit their needs. Some Fedora Commons based repositories, namely those extended by the Muradora
front-end [15] currently use RBAC with static roles. However, static approaches like these are not flexible
enough to encompass the flexible user needs for information sharing within a scientific community.

A Mediawiki extention to access control [12] has taken a step in a different direction, enabling granting of
permissions at the resource level to a group of users. Groups are typically seen as collections of users but not
collections of permissions, unlike roles in RBAC [16]. An advantage of the Mediawiki approach to access
control is that content creators may assign specific permissions to their resources based on group membership.
However, Mediawiki still requires administrative roles to define group membership and group creation.

Overall, these approaches are either variations of RBAC that require administrative management of roles, or
still require administrative support for group management. In the context of data-sharing and open-repositories,
it is relevant to give the user, as creator of resources, management over permissions assigned to his/her
resources and the user-entities used in this permission assignment, tailoring them to specific needs. To achieve
it, we propose a decentralized discretionary approach to scientific repository access control where owners
define permissions over their objects. Furthermore, we use owner-managed groups that enable resource owners
to identify the users with whom to share information with, independently from administrator created groups.

Group-Centric Secure Information Sharing (g-SIS) models approach the need to "share but protect" in
collaborative environments by bringing together users and objects into groups [17]. Unlike traditional models,
where policies are attached to objects, these "Group-Centric" models define access control based on the
membership of users and objects to a group, giving users access to an object when they are both members of
the same group. However, this model becomes confusing when assigning different access rights to an object for
different groups.
3. Requirements for Epidemiological Access Control

We identify the issues with the sharing of epidemiology data and discuss how these impact the proposed access control model.

3.1. Administration and Repository Trust

Trust in an information repository is directly related to the ability of the resource creators to maintain control over their created objects. In order to promote the uploading and resource sharing with the community, it is necessary that resource creators retain control over resources.

The current practice in epidemiology, and in some cases the legal standard [3,4], is to give the creators of sensitive resources responsibility for their resources. Therefore, they must have rights to assign permissions over their resources. This decentralization of permission management has the advantages of reducing administrative overhead as well as helping to foster trust in the repository.

3.2. User Groups

The EM has the goal of fostering collaboration within the epidemiology community. In order for access control to be adequate, collaborations should be embedded within the access control model itself. Groups of users can be used to identify these collaborations. Because collaborations are dynamic in nature and are not confined to organizational roles, users should be able to create and manage their own groups which can be created to fit a user's need to share his/her resources.

Furthermore, resource owners may want to share a resource with several groups at a given time, i.e. an epidemiologist may want to share a field data resource with epidemics modelers, public health officials and a group of project members at the same time, possibly with different permissions depending on the nature of the collaboration. Therefore, it is clear that g-SIS models, with groups containing both users and objects are not adequate to this situation, as it becomes confusing to distinguish the access control permissions given to each group.

3.3. Object Structure and Protecting Sensitive Data

The Epidemic Marketplace deals with heterogeneous resources, such as datasets, web resources, events and software. Because epidemiological studies often use and generate more than one resource, owners must be able to organize their resources in container entities and, to facilitate sharing, be able to assign permissions over resources in a container collectively, without assigning individual permissions to those resources.

![UML Class diagram of Repository Objects.](image-url)
The folder-like object structure of the Epidemic Marketplace, represented in Figure 1, is similar to that seen in operating systems and became familiar to end-users. Resources are data content objects and also contain meta-data. Collections can contain resources, meta-data and other collections. Resources and collections are described by the contained meta-data.

Since data in epidemiology can be sensitive, users may want to reveal to the public that a certain resource exists containing data about a sensitive subject without exposing the data. Additionally they may want to share the sensitive data only with a specific set of users. For this reason, owners must be able to assign different permissions over metadata and the data itself. To access the protected data, a user would request the owner of the resource for additional permissions.

By using container entities, users can assign permissions at different levels of object granularity. Assigning policies to a collection should implicitly assign the same policies to its children objects. This enables assigning permissions to groups of objects, and therefore reduce the complexity of policy management.

4. Access Control Model

In this model, users assign permissions over their resources to groups of users which are themselves user managed. We use the entities and relations from Barker's unifying meta-model to express some of the specific details of our access control model, such as object containment and implicit relations.

4.1. Permission Assignment

In the EM access control model, resource owners can assign permissions to groups. Permissions denote the capability of performing an action over an object.

Principals are all the users of the Epidemic Marketplace. The principal set is \( P = \{ p_i | i = 1...n \} \) where \( p_i \) is used to identify a principal.

Categories, in Barker's meta-model are used to classify and create subsets of entities. Here, Groups of users are described as categories of users, such as the group set \( G = \{ g_i | i = 1...n \} \), where \( g_i \) is an arbitrary group identifier. We use the relation \( pga(p,g) \) to denote that a principal \( p \) is assigned to a group \( g \).

Giving users the ability to define groups through \( pga() \) enables them to portray collaborative dynamics, which may represent several different things such as a research team in an institute or a group of authors interested in the same topics who wish to share resources among them.

Access control is defined by assigning permissions that enable principals to perform actions on an object.

To make use of groups for access control, enabling users to use collaboration as a base for data sharing, users must be able to define these permissions to groups and not only individual principals. This can be accomplished by the relation \( arca(a,r,g) \), which expresses that action \( a \) can be performed on object \( r \) by the principal in the group \( g \).

Objects are all the entities in the repository a user may or not have access to perform actions on, including collections, resources and their meta-data. The Object set is denoted as \( R = \{ r_i | i = 1...n \} \) where \( r_i \) denotes an arbitrary object.

Expressing the relationships and constraints in Figure 1 can be achieved using containment. Using the relation \( contains(r,r') \) we express that object \( r \) contains object \( r' \). We express the assignment of objects to categories as \( rca(r,c) \), denoting that object \( r \) is assigned to category \( c \), and define constraints in terms of object categories.
contains(r,r') ← assignedContainment(r,r'), rca(r, collection), rca(r', X),
X = collection ∨ resource ∨ metadata

(1)

contains(r',r'') ← assignedContainment(r',r''), rca(r', resource), rca(r'', X),
X = metadata

The $assignedContainment(r,r')$ expresses user-assigned containment, such as placing a resource inside a collection. This way, collections can only contain resources and metadata, while resources can only contain metadata. Additionally, this containment enables users to set permissions at a collection level which should mean implicit permissions for all contained resources and metadata. Similarly permissions set at a resource level should be implicit for the contained metadata. This can be expressed through:

\[
\text{arca}(a, r, g), \text{contains}(r, r') \rightarrow \text{arca}(a, r', g)
\]

(2)

In this definition, if principals assigned to group \(g\) are permitted an action \(a\) on object \(r\) and object \(r\) contains object \(r'\), then principals in group \(g\) are also permitted the action \(a\) for object \(r'\). Furthermore, since we have this object structure, permissions can be set at a higher level for metadata than for the containing object. This enables users to make their resources visible in public searches, through metadata, while protecting the data in the resource with stricter permissions.

**Actions** are the types of access that users can perform on objects.

The action set is denoted as \(A = \{a_i\mid i = 1...n\}\), where \(a_i\) denotes an arbitrary action identifier.

We consider the actions: \textit{read}, \textit{write}, \textit{delete} and \textit{insert}. Object owners are granted all actions for their owned objects.

The \textit{insert} action only exists for collections and allows users to create new resources inside collections. This enables collection owners to set permissions to create resources inside collections without giving them access to edit the collection or other contained resources. The \textit{delete} action allows users to remove an object. In addition, contained objects are also removed when the container object is deleted.

4.2. Administration

Administration in the EM access control model is defined by who can create \(pga(p,g)\), and \(arca(a,r,g)\) relations. As mentioned, to establish trust and to follow the standard policy of giving resource owners responsibility over their data, it is necessary to enable discretionary permission assignment. This is achieved by giving owners control over \(pga(p,g)\), and \(arca(a,r,g)\) for their objects and groups. Additionally, in our access control model, resource owners have all permissions for their resources. This is done by assigning permissions to all actions to the owner through the relation \(par(p,a,r)\), which assigns a principal \(p\) access to perform action \(a\) on an object \(r\), when an object is created.

Furthermore, granting an \textit{admin} action to a group of users gives that group rights over the \(arca(a,r,g)\) relations for that object, granting it the ability to administrate permissions for that object. Because owners have all permissions over their objects, they also have visibility over all permissions granted over their objects, enabling them to revoke permissions assigned by other groups with admin rights, if necessary.

4.3. Group Management

Group management is decentralized and discretionary. Group owners have permissions to edit and delete his/her created groups. Furthermore, we define that groups have three types of visibility: \textit{Private}, where only
the group creator is able to see the group; *Group Visible*, where all the members of a group can see the group and grant permissions to that group; and *Public*, where everyone in the repository sees the group and can grant permissions to that group for their objects.

5. Implementation and Evaluation

The EM back end consists of a Fedora Commons Repository and an LDAP server. Fedora Commons uses a digital object model that aggregates one or more content items into a single digital object. This digital object is identified by a unique identifier (PID) and is composed of object properties and datastreams with object content.

Using these digital objects we create the object structure described in section 3. Resources and Collections are mapped directly to digital objects, each with a unique PID. Metadata is stored within an *EM* datastream in its respective container object. Resource data is stored in content datastreams. Collections are digital objects composed only of metadata and linked to resources through a datastream reserved for object relations. These relations create the hierarchy described in the EM's object model, which will enable users to manage access control at a collection, resource or metadata level.

5.1. Authentication and Authorization

The *Fedora Security Layer* (FeSL) manages authentication and authorization. In our implementation we configured FeSL to authenticate users against LDAP credentials.

Authorization is based on XACML, the OASIS eXtensible Access Control Markup Language [18], following an architecture similar to that proposed by the XACML standard with a Policy Enforcement Point (PEP) and a Policy Decision Point (PDP) (see Figure 2). When a user requests a resource from Fedora Commons, the PEP intercepts the request and forwards it to the PDP. The PDP gather the required policies and user attributes and informs the PEP on whether the user should or not be granted access. If the user has access, the PEP provides the resource.

---

![FeSL XACML authorization architecture](image-url)

Fig. 2. FeSL XACML authorization architecture
To fulfill our user group requirement, access control policies must be evaluated based on group membership. To accomplish this, we extended FeSL’s PDP to retrieve group membership information from the LDAP server.

Because we propose decentralized management of user groups it is possible for users to generate and be members of a large number of groups. However, LDAP is optimized to retrieve attributes of a known entity and not all entities containing an attribute. Retrieving all groups a user is member of would mean retrieving all groups in LDAP and comparing the id of the members of the group to the user id. This does not scale as the number of groups increases. The solution involves creating LDAP indexes over group members, enabling the server to efficiently retrieve the groups that a user is member of, resulting in close to constant response times as the number of groups increases.

Figure 3 shows a plot comparing LDAP performance with and without indexes. The test was performed on an openLDAP 2.4 server with 51 users, 4 threads with 10 requests per thread for the varying number of groups a user is member of in a simulation where this is always a quarter of the total existing groups.

In our implementation, XACML policies are themselves stored in the repository. Because permission management is discretionary users can grant a large number of permissions to their groups and therefore adequate policy storage is necessary. We tested different strategies to creating and storing policies within the repository:

- Storing each permission assigned to a group on a separate digital object.
- Storing all permissions for an object in a datastream inside the object with separate rules for each permission.
- Storing all permissions for an object in a datastream and aggregate permissions into rules. One rule granting access to one action for multiple groups.

We found that the first solution, while compartmentalizing permissions, creates unnecessary workload on the repository, delaying permission management as well as the authorization process, and therefore access to resources. The first solution is therefore inadequate. When comparing the remaining two options, they performed similarly in terms of repository workload, and both had better performance than first solution. However, the third solution improved the handling of XML, minimizing the changes that have to be made to an XACML document to grant or revoke access to a group.

Since one Epidemic Marketplace requirement is that resource owners maintain control over their resources, a generic owner policy grants all possible actions to resource owners. The owner is identified in a digital object property. Then group policies are created and edited by resource owners or by users with admin rights, granted by the owner, over the resource. All permissions for a resource are visible and manageable at all times by the resource owner.
5.2. Interface

From an interface standpoint, user interaction with access control mechanisms is divided into two sections: managing permissions and managing user groups. Managing permissions is similar to performing other actions on a resource. In the resource page of the information platform, a user can find a link to the Manage Permissions action, which points to the section where he can set the intended permissions to user groups (Figure 4).

The defined access control model enables the creation of separate permissions for metadata and data for all the actions in the repository. However, because it might be complex to an end-user to know which permissions to assign, in our interface we opted to present the metadata read permission as the “visibility of the resource”, and is managed separately from other permissions. This reduced clutter and still enables users to make metadata visible to the public while protecting potentially sensitive resource data. In the interface the users can also set permissions for users, which in the presented model are considered to be groups with only one member.

User group management is made through specific menu which directs users to an area where they can create new groups or edit existing groups.

6. Conclusions and Future Work

Throughout this paper we explored the access control needs of scientific repositories and proposed an access control model aimed at promoting information sharing while protecting it. The proposed access control model was implemented in the Epidemic Marketplace information platform, available from http://www.epimarketplace.net. Based on the experience of designing and implementing the Epidemic Marketplace, we explored how the access control requirements for scientific repositories are attainable through a group-based discretionary access control model and how it can be implemented using an open-source repository software common in existing digital libraries.

In our proposed model, resource owners assign permissions to user groups, which can be used to easily identify cooperating users. Groups are created and managed by users so that they can be flexibly tailored to a
resource creator’s information sharing policies. Furthermore, our object structure enables assignment of permissions at the collection level, reducing the complexity of assigning individual permissions to resources. Assigning permissions at a meta-data object enables resource owners to expose meta-data to the public, making it searchable, while protecting resource data which can then be shared to a more restrictive group of users. Furthermore, this makes it possible for user to request access to a resource which has exposed metadata but private data. Once an agreement has been reached, the resource owner can add the requesting user to a group with adequate access permissions.

A challenge that can be further explored is the incorporation of this group-based access model in a social network environment, where users expose explicit relations with each other to better portray the collaborative nature of the repository.

Acknowledgements

The authors want to thank the European Commission for the financial support of the EPIWORK project under the Seventh Framework Programme (Grant # 231807), the EPIWORK project partners, FCT (Portuguese research funding agency) for its INESC-ID and LaSIGE Multi-annual support.

References


[4] European Comission et al., Proposal for a regulation of the european parliament and of the council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (general data protection regulation), 2012.


