A Standards-Based Approach for Supporting Dynamic Access Policies for a Federated Digital Library

K. Bhoopalam, K. Maly, F. McCown, R. Mukkamala, and M. Zubair

Department of Computer Science,
Old Dominion University, Norfolk, Virginia 23529 USA
Voice: 1+757+683+3915
{kbhoopal, maly, fmccown, mukka, zubair}@cs.odu.edu

Abstract. With the increasing acceptability of interoperability standards like Open Archives Initiative protocol for metadata harvesting, it is becoming feasible to build federated discovery services which aggregate metadata from different digital libraries (data providers) and provide a unified search interface to users. Content-based access control is one of the primary requirements of data providers. While this concept has been predominant in the research realm, practical systems incorporating this concept are rare. In this paper, we propose a framework that supports and enforces content-based access policies using existing COTS components. We have prototyped the framework by building a system using XACML, and a XACML policy engine. The system can also be generalized to environments other than digital libraries.

Keywords: metadata, access control, content management.

1 Introduction

With the increasing acceptability of interoperability standards like Open Archives Initiative protocol for metadata harvesting, it is becoming feasible to build federated discovery services [9, 10]. These services aggregate metadata from different digital libraries (data providers) and provide a unified search interface to users. One of the primary obstacles that keep data providers from joining the federation is the lack of an infrastructure to support content-based access policies. A data provider is more willing to share its metadata with a service provider if it can provide content-based access control, in addition to the traditional access control (e.g., role-based [14]).

Our earlier works [2, 3] addressed the basic issues in managing access to a federation service that is being accessed by many communities (e.g., educational institutions), each having different contracts with different commercial data providers (e.g., American Physical Society) to the federation and content-based restriction using XACML [12]. In this paper, we propose a framework that supports and enforces content-based restrictions and provisional actions defined by data providers for a federated digital library.

Content-based restrictions restrict access to full text or metadata containing specific phrases. For example, we can restrict any material containing word *nuclear* from being accessed by a specific user group. In addition, we provide another

E.A. Fox et al. (Eds.): ICADL 2005, LNCS 3815, pp. 242 – 252, 2005.

important feature relevant for both government and commercial agencies: *provisional actions*. Provisional actions [7] are directives such as auditing of information access prior to the granting of access privileges to a user. For example, an administrator may require a digital library to send an e-mail prior to providing access to a user from a certain organization. It is possible to combine content-based restrictions with provisional actions such as "send an e-mail to the data provider if a specific user community accesses any of its material containing the word *nuclear*."

This paper elaborates our framework and a prototype implementation to incorporate the above two features into a general access management system. While the framework is flexible in terms of its modularity and ability to incorporate COTS components, the prototype implementation of the framework illustrates how the available technologies such as Shibboleth and XACML can actually be employed to achieve the goals. The paper is organized as follows. Section 2 summarizes previous work in this area. Section 3 describes the proposed framework. In section 4, we provide details of our prototype implementation. Section 5 discusses the frameworks and the prototype implementation challenges. Finally, section 6 summarizes our contributions and discusses future work. In particular, we describe our goals for the framework and the flexible framework for access control models in digital libraries to include provisional actions and content-based restrictions.

2 Previous Work

In this section, we provide background information on Archon a federated digital library on which much of our work is based and also discuss previous work in this area.

2.1 Archon: A Distributed Access Management System for Federated Digital Library

In a federated digital library, the aggregator enforces a custodial contract governing the relationship between contributors and subscribers using an access manager. Archon [11] is an Open Archives Initiative [9] compliant federated digital library with an emphasis on physics for the National Science Digital Library (NSDL) [13]. In our earlier work [3], we developed an authentication and access control architecture for Archon. Archon uses the Dublin Core Metadata standard [5] to store metadata. We have used vocabulary from the Dublin Core standard for representing resources and vocabulary from EduPerson [6] for representing subject attributes in our preliminary '<subject , resource, action>' Access Policies, thereby providing a uniform naming convention for resources and subject attributes. It also provides a technological demonstration of secure federated digital libraries to support authentication at authoritative sources. In [2] we have shown how COTS based policy languages can be used to represent content-based access control.

2.2 Content Labeling vs. Content Restrictions as Obligations

Some of the earlier work on content based authorization models [1] proposes the association of "concepts" with digital library. "Concepts" as the name implies are a set of phrases that accurately capture the relevance of a digital object. The association of concepts is akin to content-based authorization based on a label value. We believe that their approach adds considerable administrative overhead whenever a new digital object is introduced. Additionally, it is not favorable for content-based access on pre-existing digital archives that have not introduced this labeling mechanism. Our approach does not require additional labels, and allows for the specification of content-based restrictions on the values of any of the meta-data fields or labels.

2.3 Embedding Access Control Information Within Digital Objects vs. External Representation

Some systems such as [8] wrap a digital object (e.g., multimedia objects) with authorization information. Although such an association allows for fine-grained access to parts of a digital object, the association remains static. Hence, it is not possible to include content-based access control using this method. Additionally, the model also does not permit provisional actions. In our work, we employ external representation of authorization information to facilitate more flexible authorization as well as accommodate content based restrictions.

3 Proposed Access Enforcement Framework

As mentioned in the introduction, the primary contribution of this paper is a flexible framework for access management in federated digital libraries. In particular, the objectives for the framework are as follows.

1. Provide a modular framework for the enforcement of content-based access control with provisional actions.
2. Facilitate content based access control in digital libraries without any fundamental changes to the submission, dissemination and preservation process of digital collections.

In our framework (figure 1), we enforce content based access restrictions and provisional actions without additional infrastructure or tools, beyond what may be used for access control on metadata fields. We have incorporated content-based restrictions and the provisional actions at two points: the Policy Decision Point (PDP) and Policy Enforcement modules (PEP). The framework is described in terms of interactions (1-15) among different components. The access requestor receives the user's request via the gateway in any (or among a set of) domain dependent formats (1). Upon receiving the request, the access requestor fetches the policies required for access evaluation and the necessary information required for request construction from the resource and access policy directory (2). Then, the access requestor submits the relevant policies and the requests to the PDP (3). The PDP evaluates the requests against the policies, and provides responses to the access decision handler (4). The access decision handler constructs an access token to store the compendium of the access decisions and invokes the pre-query provisional action fulfiller, the query builder, and the post query provisional action fulfiller to implement content independent provisional actions, fetch content from the digital library and implement content based provisional actions (5-13). Finally the access decision handler passes the fetched content and the access token to a user interface filter that renders the content based on the access decisions in the access token (14,15).

Our framework modularizes the Policy Enforcement Point and establishes clear interfaces with the PDP, the resource directories, the information repository (database) and other tools required for provisional actions. This modular architecture allows for changes in the modules and also the exclusion of modules as and when user requirements change with minimum impact. For example, if user's attributes are received in a different format, the access requestor is the only module that needs to be changed. Similarly, if the domain of the resources and the permitted actions change, the Access-Requestor needs only to interact with a different Resource and Access policy Directory. The Pre and Post-Query Provisional Action fulfiller can be excluded if the access control system does not require provisional actions. The Post-Query Provisional action fulfiller can be excluded if there aren't any provisional actions that depend upon the contents of the information fetched from the database. A separate Query Builder isolates data-base connection handling and access in a separate module, hence, allowing for queries to be optimized for various databases (the system uses JDBC to connect to an oracle database, and hence is considered database independent as long the database understand the SQL standard,). Additionally a separate User-Interface handler provides for the separation between access-evaluation, storage mechanism and information presentation.

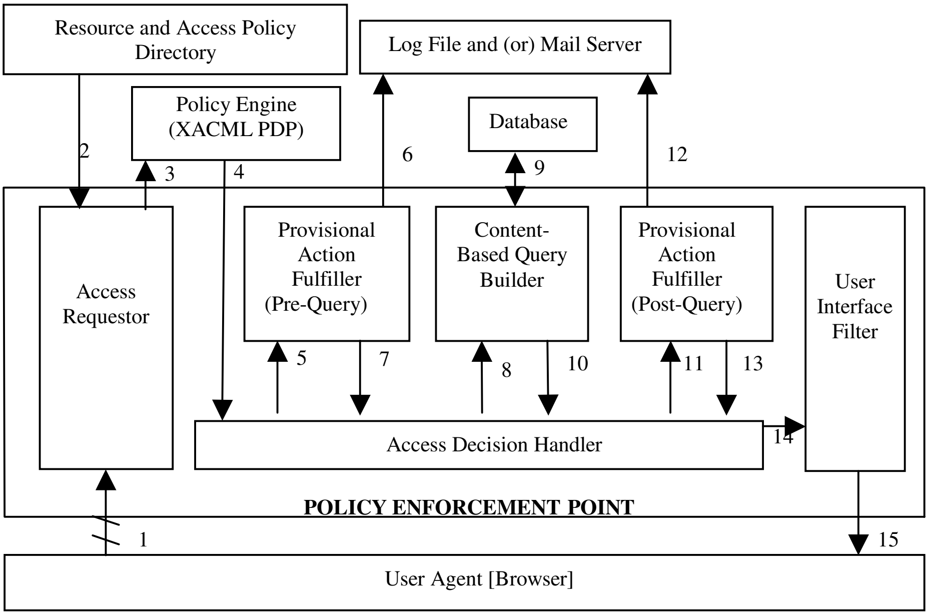

Resource and Access Policy Directory
Log File and (or) Mail Server
Policy Engine (XACML PDP)
Database
2
3
4
6
9
12
Access Requestor
Provisional Action Fulfiller (Pre-Query)
Content-Based Query Builder
Provisional Action Fulfiller (Post-Query)
User Interface Filter
5
7
8
10
11
13
Access Decision Handler
14
POLICY ENFORCEMENT POINT
1
15
User Agent [Browser]

Fig. 1. Access Enforcement Framework

4 Implementation

The current implementation of the proposed framework is primarily based on three components: (a) OAI-PMH based Archon [12] (b) the Shibboleth framework [4] to provide secure remote authentication and transport of authenticated information, and (c) XACML to specify access control rules. In the rest of the section we elaborate on the specification and enforcement of content restrictions and provisional actions (using XACML), the implementation of the access enforcer, and how the adoption of a standards-based tool for specification (e.g., XACML) has influenced our design of the access enforcer.

4.1 Implementing Content-Based Restrictions and Provisional Actions

We have used XACML's *obligation* feature (element) for specifying content restrictions and provisional actions. In the current implementation, we have limited content-based restrictions to metadata fields only. For example, students from a particular subscribing institution are restricted from viewing records with (i) a "description" metadata field containing phrases "nuclear weapon" and/or "anthrax" and (ii) a "subject" field containing the phrase "WMD."

In XACML, obligations are used to provide directives to the enforcer. Each obligation in a *Policy* element can be associated with a "Permit" or "Deny" decision. If an obligation accompanies a "Permit" decision for a particular access request, then the XACML semantics state that all obligations *must* be fulfilled prior to the enforcement of the "Permit" decision (by the *Policy Enforcement Point* or PEP, referred here as the enforcer). We use this feature for expressing content restrictions. Hence, if obligations specify content restrictions for a "Permit" decision, then it must be enforced by the PEP. In this context, obligations provide a content filter that is used by the PEP to restrict data to users.

We can also use the *obligation* feature to express provisional actions. For example, if a contributor (or digital data owner) wishes to know the request pattern for a certain subset of its subscribers, it would not be possible for a web server's native log mechanism to provide such fine grained conditional logging. Expressing such provisional actions through access policies eliminates the need for rewriting the enforcement code at the data aggregator (e.g., Archon) in a procedural language. This flexibility reduces the necessity of frequent changes to the source code and redeployment.

4.2 Content-Based Restrictions in XACML: An Illustration

Figure 2 shows an obligation element that encodes content restrictions. Here we assume the scenario of a subscribing academic institution with *faculty* and *student* roles. The obligation element is placed in a *Policy* element that encodes rules for the student from the chosen subscribing institution.

The responsibility is on the enforcer to interpret the contents of the obligation element and take the appropriate actions. As the vocabulary of the `ObligationId` and `AttributeAssignment` attribute values are not standards based, it is important that people responsible for coding the policy enforcer and the policy editor

```
<Policy>
  ….
 <Obligations>
   <Obligation ObligationId="content_restrictions"
   FulfillOn="Permit">
   <AttributeAssignment AttributeId="description"
             DataType="http://www.w3.org/2001/XMLSchema#string">
             nuclear:anthrax</AttributeAssignment></Obligation>
   </Obligations>
 </Policy>
```

Fig. 2. Content Restrictions in XACML

agree upon the syntax and the semantics of the various directives listed in the obligations. In the absence of such cooperation, the enforcer would not understand the obligation(s) and hence would not be able to provide any access privileges (based on XACML guidelines) to the user. It should be noted that this example can lead to "false positives," for example, a content restriction string If multiple phrases need to be specified for content restrictions in each metadata field, they are separated by a colon (e.g., nuclear**:**anthrax). As this obligation is encoded to be fulfilled on a permit decision, the policy engine returns the entire obligation *as is* to the enforcer (PEP). This obligation is translated into a SQL statement below to ensure that only the required information is fetched from the database. XACML snippets are agreed upon by the enforcer and the policy specifier. In the example, the obligation with obligationId "content-restrictions" states the following: "Whenever a (student role) user's request is permitted (e.g., permission to read metadata), the user may not see records that contain *nuclear* or a*nthrax* in the description field.

```
SELECT <permitted column list indicating metadata names>
FROM <database table>
WHERE (description NOT LIKE (%nuclear%)
OR description NOT LIKE (%anthrax%))
```

However, this method may have unintended consequences of excluding valid material. If the phrase "nuclear" was intended to hide digital objects that contain "nuclear bomb" or "nuclear device" may also hide digital objects that contain phrases like "nuclear family". We are currently working towards a method that allows for the specification of regular expressions, so that a trained security administrator, using a visual editor, can accurately define content-restriction phrases thereby reducing the occurrence of such false positives.

4.3 Provisional Actions: An Example

In the XACML implementation for our system, all content based provisional actions are characterized as either "pre-query" or "post-query" using the `obligationId`.

The snippet in Figure 3 encodes a pre-query provisional "pre_query_audit", which mandates that the policy enforcer log the *time* and the *role,* or *identity* if available of the requesting user, and a "post_query_email" which mandates that an e-mail be sent to dlib-admin@cs.odu.edu if the "description" metadata field of contents fetched from the database have the phrase “particle physics” in them. The policy evaluator responds with the obligations only if the request is permitted. This requirement is stated in the FulfillOn attribute of the obligation.

```
<Policy>
  ….
 <Obligations>
    <Obligation ObligationId="pre_query_audit" FulfillOn="Permit">
   <AttributeAssignment AttributeId="time"
                  DataType="http://www.w3.org/2001/XMLSchema#string">

                  CURRENT_TIMESTAMP</AttributeAssignment>
   <AttributeAssignment AttributeId="subject"
                  DataType="http://www.w3.org/2001/XMLSchema#string">
                  role:identity</AttributeAssignment></Obligation>
   <Obligation ObligationId="post_query_email" FulfillOn="Permit">
   <AttributeAssignment AttributeId="content_description"
                  DataType="http://www.w3.org/2001/XMLSchema#string">
                  particle physics</AttributeAssignment>
   <AttributeAssignment AttributeId="emailto"
                  DataType="http://www.w3.org/2001/XMLSchema#string">
                  dlib-admin@cs.odu.edu</AttributeAssignment>
   <AttributeAssignment AttributeId="static body"
                  DataType="http://www.w3.org/2001/XMLSchema#string">
                  Accessing flagged records.</AttributeAssignment>
              </Obligation>
   </Obligations>
 </Policy>
```

Fig. 3. Content-based Provisional actions in XACML

4.4 Formal Specification of Content-Based Access Control with Provisional Actions

Content restrictions have the effect of hiding the entire digital object for which the content restriction rule satisfies. Label or metadata based restrictions have the effect of hiding only the label or metadata, and applies to all digital objects being retrieved.

Content based access control: (credentials, labels$_1$, privilege, +) ^ Σ(credentials, label$_2$, restriction-phrase, -)

A user with attributes ‘credentials’ is granted the privilege (currently a read permission) on the labels$_1$ of those digital objects which do not have the phrases specified as ‘restriction-phase’ in label$_2$. The ‘Σ’ indicates that content-restrictions can be specified on different labels of a digital object.

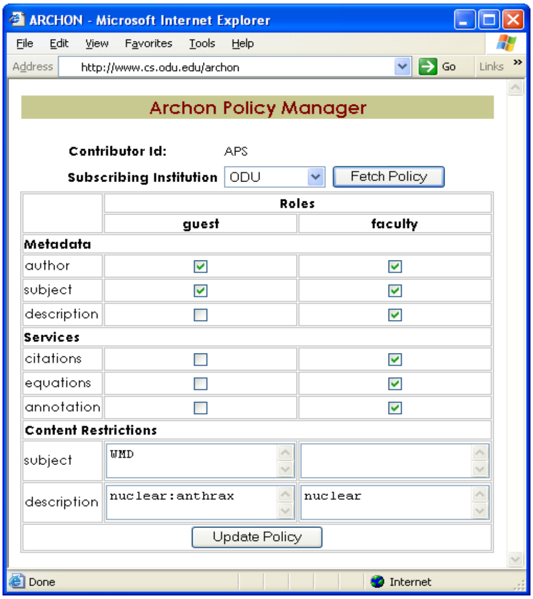
Content based provisional actions: (credentials, labels$_1$, privilege, [+ or -]) ^ Σ(credentials, label$_2$, restriction-phrase, pa)

A user with attributes ‘credentials’ is granted or denied the privilege (currently a read permission) on the labels$_1$ of digital objects, and if the digital objects contain the phrases specified in ‘restriction-phase’ for label$_2$, the provisional actions ‘pa’ must be implemented. The ‘Σ’ indicates that content-restrictions can be specified on different labels of a digital object. Although most previous instances of the “credentials” in this paper refer to user's role in his home (also subscribing) organization, the XACML specification allows any number of attributes of the user including user ID, age, time or location of access, etc. Using XACML provides our implementation with the capability to easily extend the complexity of the access rules. In our system, the XACML policy specifies the role-privilege mapping. The user-role mapping is performed at the home organization of the user and the mapping is honored by the policy enforcer at the aggregator.

4.5 Management of Access Policies

As specified earlier, a contributor has a contractual agreement with subscribers, thereby enabling selected personnel from the subscribing institutions to have access to the contributor's content (hosted by Archon). Archon provides the contributor with a "Policy Editor" tool to manage access policies for end-user roles of its subscribing institutions. The matrix format [15] shown in figure 4 is among the most widely used access models and visual representation method to specify access control as it allows for only consistent rules to be specified. XACML per se does allow inconsistent rules to be formulated. As we have demonstrated already in [3], this form of policy specification can be automatically translated into XACML and is extremely easy to use by non-technical people.

Figure 4 is a scaled-down version (contains fewer services and metadata, and end user roles than the test bed) of the policy editor we have developed for our system. Figure 4 shows the access policy of the contributor APS for faculty of the subscribing institution ODU. A selected check box indicates the metadata or the services is permitted for the specific end user role. The checked items are listed as resources in


ARCHON - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address
Go
Links
Fetch Policy
Update Policy
Done
Internet

http://www.cs.odu.edu/archon

Archon Policy Manager

Contributor Id: APS

Subscribing Institution ODU

	Roles	
	guest	faculty
Metadata		
author	☑	☑
subject	☑	☑
description	☐	☑
Services		
citations	☐	☑
equations	☐	☑
annotation	☐	☑
Content Restrictions		
subject	WMD	
description	nuclear:anthrax	nuclear

Fig. 4. Access Policy Editor

an XACML rule and each permitted resource is accessed from the database by including the corresponding column name in the SQL query shown earlier. The content restrictions are applied to the description and subject metadata fields and are specified as colon separated phrases. The content administrator manages XACML access policies through this simple point-and-click editor that enforces the specification of consistent access policies.

5 Discussion

As we have adopted a standards-based approach to implement access control, we have been limited by several constraints of the standards. This section describes some of the complexities of our implementation due to our choice of XACML for access specification. XACML specifies a schema for specifying requests to the policy decision point and for responses when the policy decision point responds with its decision. The schema of the request format constrains each XACML request to encapsulate only one `resource` element and only one `action` element. The schema of the response context contains a single `decision` element that specifies whether the request was permitted or denied. This request specification format introduced several limitations in our system.

5.1 Number of Interactions Between Policy Enforcer and Policy Decision Point

End users can have different sets of permissions on the resources provided by different contributors; hence, multiple requests are required to compose the compendium of access privileges. If the number of resources provided by a contributor is O(K), the number of requests, evaluations and responses that are required to construct a compendium of the end-users access privilege on resources of one contributor is O(K). The number of requests would be larger if there were more than one permissible action on the resource. This would have introduced another multiplicative factor in the number of requests (and hence evaluations). We believe that this constraint of XACML would induce substantial delays in high-transaction digital libraries.

5.2 Changing Formats

Due to the current XACML request format, it was necessary for the policy enforcer to translate user assertions available as HTTP request parameters into XACML context requests. To compose each XACML request, the policy enforcer embeds the user credentials and a resource identifier, which is mutable, within the immutable constructs required for an XACML request context. Although simple in design because of the assumptions that were made, the composition of XACML requests and subsequent processing of XACML responses is a computational overhead.

6 Conclusions and Future Work

Access control for digital libraries currently is rather primitive due to the fact that most large digital libraries are solitary, proprietary systems that do not interoperate and are only available to the user community managed by each digital library. To provide a seamless access to many digital libraries simultaneously, a more sophisticated security model is needed. We have provided in this paper a framework that we have implemented that provides a sophisticated access paradigm to distributed user groups for distributed digital libraries at no noticeable cost to the user in terms of response time. By using declarative languages such as XACML we can make changes in policies effective immediately and minimize the cost of changing enforcement code at the resource (typically a federation service). Enforcement actions that need to be written into the source code of the resource are restricted to two places: the presentation layer and the query construction modules. All decision making as to access permissions are made by a standard XACML policy engine.

Currently, we are investigating the possibility of incorporating the role-based access control on hierarchical roles and subjects using declarative languages like XACML. We are also investigating the usage of a canonical set of subject attributes in government and commercial organizations to broaden the usage of our work.

Acknowledgements

The work has been partially funded by IBM University Research Fellowship Program.

References

1. Adam N.R., Atluri V., Bertino E., and E. Ferrari. A content-based authorization model for digital libraries. IEEE Trans. on Knowledge and Data Engineering, 14(2):296–315, March 2002.
2. Bhoopalam, K., Maly, K., Mukkamala, R., Zubair, M. A Flexible Framework for Content Based Access Management for Federated Digital Libraries. Proceedings of IADIS, Madrid, October 6-9, 2004.
3. Bhoopalam, K., Maly, K., Mukkamala, R., Zubair, M. Access Management in Federated Digital Libraries. Proceedings of IADIS, Madrid, October 6-9, 2004.
4. Cantor, S., Erdos, M. Shibboleth-Architecture DRAFT v05, http://shibboleth.internet2.edu/docs/draft-internet2-shibboleth-arch-v05.pdf (21 April 2004).
5. DCMI Metadata Terms, Dublin Core Metadata Initiative, http://dublincore.org/documents/dcmi-terms/
6. EduPerson Specification, http://www.nmi-edit.org/eduPerson/internet2-mace-dir-eduperson200312.html
7. Hada, S. and Kudo, M. XML Access Control Language: Provisional Authorization for XML Documents. (Tokyo Research Laboratory, IBM Research). October 16, 2000.
8. Kodali N., Farkas C., Wijesekera D. An Authorization Model for Multimedia Digital Libraries. The Int. Journal of Digital Libraries, Vol 4, 139 -155., 2004.
9. Lagoze, C.,and H, Sompel, V., Nelson, M., Warner, S. The Open Archives Initiative Protocol for Metadata Harvesting, Open Archives Initiative. http://www.openarchives.org/OAI/openarchivesprotocol.htm (21 October 2004).
10. Liu, X., Maly, K., Zubair, M., and Nelson, M. 2001. Arc -- An OAI Service Provider for Cross Archiving Searching. Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, Roanoke, VA, June 24-28, 2001, pp. 65-66.

11. Maly, K., Anan, H., Tang, J., Nelson, M., Zubair, M. and Yang, Z. Challenges in Building Federation Services over Harvested Metadata. Proceedings of ICADL2003. pp.602-614, Kuala Lumpur, Malaysia, Dec 2003.
12. Moses, T. (eds.). OASIS eXtensible Access Control Management Language (XACML). Version 2.0, OASIS Standard, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf (1 February 2005).
13. National Science Digital Library. http://www.nsdl.org/, (05 November 2004).
14. Sandhu, R., et al. Role-Based Access Control Models. IEEE Computer 29(2): 38-47, IEEE Press, 1996.
15. Sandhu, R. The typed access matrix model. In Proc. of the 11th IEEE Symp. on Security and Privacy, pp. 122-136, 1992.