

A Lightweight Decentralized Authorization Model for Inter-domain Collaborations

Hannah K. Lee
lee@informatik.uni-hamburg.de

Heiko Luedemann
2luedem@informatik.uni-hamburg.de

Security in Distributed Systems (SVS)
University Hamburg
Vogt-Koelln-Str. 30
D-22527 Hamburg, Germany

ABSTRACT

Inter-domain collaborations comprise of a series of tasks, whose run-time environment stretches over heterogeneous systems governed by different set of policies and where participating organizations desire to preserve control over their resources. One of the major security challenges in modeling those inter-domain collaborations is providing a decentralized authorization solution. At the core of this challenge lie two problems: 1) an authorization decision maker does not know who a principal is and 2) which set of privileges this principal owns if the principal is originated from outside of the decision maker's domain. Currently, a number of different approaches tackle this problem and claim to provide a full-fledged solution. These approaches, however, often require particular use of infrastructures and their own policy languages. In this paper, we propose a lightweight model using the concept of *distributed roles* from the dR-BAC model to bridge different domain boundaries. Based on e-Government collaboration scenarios, we identify a set of requirements of decentralized authorization and propose an extension to the current XACML specification as a realization of our model.

Categories and Subject Descriptors

K.6.5 [Computing Milieux]: Management of Computing and Information Systems—*security and protection, unauthorized access*

General Terms

Security, standardization

Keywords

Authorization policy, e-Government, XACML, dRBAC, inter-domain collaborations

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SWS'07, November 2, 2007, Fairfax, Virginia, USA.

Copyright 2007 ACM 978-1-59593-892-3/07/0011 ...\$5.00.

1. INTRODUCTION

The on-going prosperity of the “e-” trend such as e-Business, e-Government, and e-Learning fosters the ever increasing demand for interactions across organizational boundaries. Along with that comes the urgent need of providing a decentralized authorization service that overcomes the same huddles in an open or a loosely-coupled environment. In fact, numerous research communities have put significant efforts and produced highly elaborated solutions, starting from various extensions to the role-based access control policy [17, 18, 8, 4] and the ones using a meta-policy concept [5] to a number of credential-based delegation models [12, 6, 9, 13, 10]. Many of those models come with their own specification languages, supportive infrastructures, frameworks, and additional configuration steps. We view some of these solutions as overkill for the majority of inter-domain collaborations and intend to come up with a more succinct and easily deployable solution using standard based solutions and preserving locality of authorization controls.

Among current proposals, ones from the role-based trust management model show eloquent solutions for inter-domain collaborations. They incorporate the strengths from both worlds of role-based access control and trust management models. Using the mechanism of delegating authorization decisions to authorities in different domains, they not only provide trust establishment in a multi-domain setting, but some of them also enable a role activation procedure [10]. Nevertheless, deployment of these solutions is not trivial. Often having their own language specifications and infrastructures, compatibility must be checked both at the external and internal interfaces of a given domain. We see a source of complexity of these solutions comes from the premise that delegation is necessary for authorization decision. Hence we have investigated how this can be simplified by using the notion of *distributed roles*. These distributed roles are known to all collaborative organizations and are used to provide an extra layer of abstraction between different domains; they then can be mapped to local roles or users by each domain administrators. Especially for the e-Government scenarios, which we illustrate in Section 2, these distributed roles are analogous to globally known positions and the associated privileges to those positions, and thus do not cause unnecessary “phantom roles” as in SPKI [15].

We have implemented the distributed role model by extending the XACML 2.0 specification, and hence we contribute to provide a decentralized authorization solution that can be utilized with the other Web-service authoriza-

tion building blocks without requiring additional infrastructures or language specifications.

This paper is structured as follows. Section 2 illustrates an e-Government collaboration scenario through Europol and Eurojust collaborations with their 27 EU member states. Section 3 discusses the requirements of decentralized authorization in the light of the given e-Government interactions. Section 4 presents our proposal of extending XACML standard for inter-domain collaborations along with the toolkit for policy administrators. Section 5 states related work followed by the conclusion and the direction of the future work in Section 6.

2. EP & EJ COLLABORATIONS

Europol, the European Police Office (EP) and Eurojust, the European Judicial Cooperation (EJ), are supra-national European agencies that have been set up to facilitate the EU member states in their fight against cross-border organized crimes. To accomplish their primary mission, both Europol and Eurojust carry out specific tasks in the context of joint efforts amongst the police, customs, immigration services, and justice departments of the EU member states for instance [3]. While Figure 1 sums up the bird’s eye view of the generic interactions amongst participants of their collaborations, the following section explains the nature of their interactions, involved resources, roles, and regulations in detail.

Interactions. The nature of the interactions between Europol and Eurojust as well as with the 27 member states are best described as dynamic and ad-hoc. Dependent upon how an investigation unfolds, different member states join or leave in the middle of a collaboration. Thus, it is not always known in advance which path of execution a collaboration will take. While the path of execution is dynamic and unpredictable, the patterns of their interactions are rather static and easily categorized. They have basically two types of interactions: 1) a request/response based information exchange and 2) a direct access to a cross-organizational resource. In the former type of operation, a requester simply inquires a certain set of information to an organization that owns the particular set of information. Upon the approval of such request, a resource provider sends back a requested set of information. The latter type of interaction actually includes the former type of operation in order to receive an approval from a resource provider before executing a direct access across an organizational border. Though a significant amount of their activities are currently paper-based and done off-line, in this paper we consider digital information exchanges only. Some of the examples of their collaborations are forming a joint investigation team, acquiring an European arrest warrant, and requesting for a mutual legal assistance for witness protection during a court proceeding.

Resources. Both Europol and Eurojust utilize a set of resource repositories to store the information related to ongoing investigations. The information they keep are mostly highly confidential and often originally come from the member states. The life-time of the resources being cached in these supra-national agencies is bound to that of the related investigation cases. In both of the organizations, the access control rules with respect to these resource repositories are explicitly specified, requiring highly specific and restricted

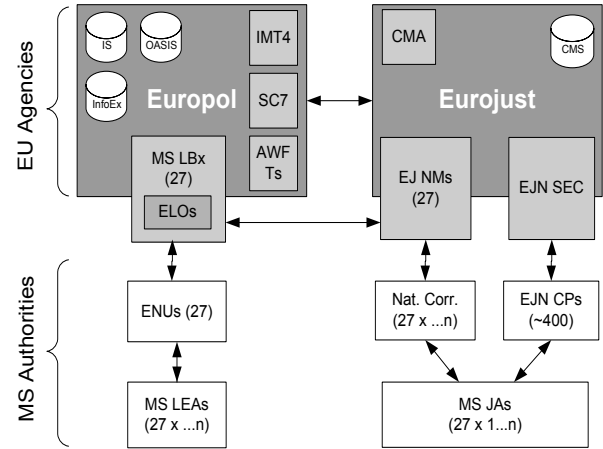


Figure 1: Generic Overview of Europol & Eurojust Collaborations [3]

roles, both internally and externally, to obtain appropriate permissions.

Roles. Each of the 27 member states has an appointed Liaison Officer and a Contact Point to interact with Europol and Eurojust respectively. Within each member state, various judicial and law enforcement authorities collaborate to complete the task that has been assigned from the inter-domain collaboration level. The main roles visible at the cross-organizational level are: Europol’s Information Management Unit (IMT4), Europol’s Liaison Officers (ELOs), Member States’ Liaison Bureaux (MS LBx), Europol National Units (ENUs), Member States’ Law Enforcement Authorities (MS LEAs), Eurojust National Members (EJ NMs), Case Management Analyst (CMA), National Correspondents, Member States’ Judicial Authorities (MS JAs), European Judicial Network Secretariat (EJN SEC), and European Judicial Network Contact Points (EJN CPs). Refer to Figure 1 to view how the roles are related. Through these roles, approximately 5,000 users from the law enforcement side (involving 250-300 Law Enforcement Agencies with 17-20 officials each) and 750 users from the judicial side (involving 400 EJN Contact Points, the 27 National Correspondents on terrorism, the 27 Eurojust National Members, and additional 10 users per Member State) are involved in the Europol and Eurojust collaborations [3].

Rules. Majority of the overarching regulations amongst the involved parties are derived from the European directives and laws. Related regulations cascade from the European Union level to the national level. In addition to the legislative regulations, an agreement exists between Europol and Eurojust regarding security issues of their interactions. For instance, the equivalence principle must be applied upon the data handling code of security level of any information that is being exchanged between them. A higher security code must be observed between a sender and a recipient of data throughout their collaborations [3]. Another note-worthy regulation with respect to the resources stored in these organizations is the notion of ownership. Each set of information stored is associated with its owner, and the owner can specify access control rules regarding his or her own resources.

It is, however, in a way restricted by the general overarching laws and regulations. For example, if the general regulation specifies that Eurojust national members may access analysis work files, and as an analysis work file owner, one can only specify which, out of the 27 Eurojust national members, may have access to his or her resource file. The owner, however, can not make his or her resource accessible to personnel who are not permitted to have access to the type of resources by the overarching rules.

In a way, the Europol and Eurojust scenarios represent typical type of inter-domain collaborations. Here is the summary of the characteristics of their collaborations:

- Collaborative organizations are highly **heterogeneous**.
- The 27 member states want to stay **autonomous** in terms of controlling their resources and executing tasks which they are responsible for.
- The path of execution is **dynamically** determined.
- Related **laws and regulations** provide collaborative partners with the basis of mandatory constraints and pre-conditions they must comply.
- Fairly stable and **globally known roles** do already exist amongst collaborative partners.
- The privileges of these roles are partially defined in the overarching **laws and regulations** that are known to all collaborative partners.

While the first three attributes of the list above make Web-services an extremely attractive means of realizing the EP & EJ collaborations, the other three offer possible optimizations for the needed decentralized authorization service.

3. PRIMARY REQUIREMENTS OF DECENTRALIZED AUTHORIZATION

Based on the analysis of the EP & EJ collaborations discussed at the end of the previous section, we draw the following to be primary requirements of a decentralized authorization model:

Locality of control. Local authorities must be able to determine authorization decisions according to their own criteria, and each collaborative partner must be able to control which parts of their policies to disclose to other collaborative partners.

Compatibility with standard based solutions. Use of open, standard-based building blocks ensures that a solution can be suitable for the majority of inter-domain collaboration scenarios of heterogeneous communities. It also aids a scalable deployment without introducing significant changes.

Flexible choices of enforcement mechanisms. It is necessary that a decentralized authorization model is not bound to a particular access control mechanism but rather be enforceable by as many different types of technologies as possible. This is a similar design principle as how Security Assertion Markup Language (SAML) is built.

Required attributes of authorization policy:

- **Interoperability.** It is unavoidable to eliminate syntactical translations of authorization policies in the context of inter-domain collaborations. However, the number of translation required can be reduced by utilizing standard-based specifications and by avoiding proprietary solutions. This scheme also reduces the risk of having an inconsistent, inaccurate set of policies.
- **Expressiveness.** Notions of roles, capabilities, and temporal constraints must be able to be expressed. Declarative languages have advantages on expressiveness over imperative languages. It is also desirable to be able to express essential organizational controls such as delegation and separation of duty constraints. According to the studies done by Clark and Jones on the maturity level of interoperability, the maturer the level of interoperability, the more shared commands and knowledge are available amongst the participants [7]. This implies the specification of those organizational controls amongst the externally known roles such as separation of duties and delegation is a prerequisite of a mature inter-domain collaboration.
- **Scalability.** Policy specifications must be enforced and managed by policy administrators located in a distributed system. Thus, the deployments of policy specifications must be simple and scalable. Any updates and changes must be propagated to different policy decision points in an efficient manner.
- **Extensibility.** Changes upon policies are anticipated. In the Europol and Eurojust scenarios, a new member state may join their cooperation, introducing new roles and privileges. Thus new rules, entities, and conditions must be easily added or modified without causing major changes on the the structure of pre-existing policies.

A number of currently existing models do satisfy many of the requirements stated above such as allowing locality of control and flexible choices of enforcement mechanisms. However, these solutions rely on their own policy specification languages and additional supportive infrastructures as described in Section 5 in detail. These approaches can be a comprehensive stand-alone solution for a secure and durable collaborative environment such as Virtual Organization, but they are too complex to deploy to more dynamic and rather loosely-coupled collaborative environments as described in Section 2. Therefore, the basic principle of our approach is designing a decentralized authorization model which can be easily integrated into the standard based technologies that enable inter-domain collaborations such as Web-services. XACML has been chosen for the similar basis of the reasoning in addition to the fact that it satisfies most of the other requirements of an decentralized authorization policy listed above.

4. EXTENSION PROPOSAL TO XACML

The kind of inter-domain collaborations we address in this paper are application-level collaborations where collaborative parties have a fair degree of knowledge of one another and where mandatory pre-conditional constraints are known

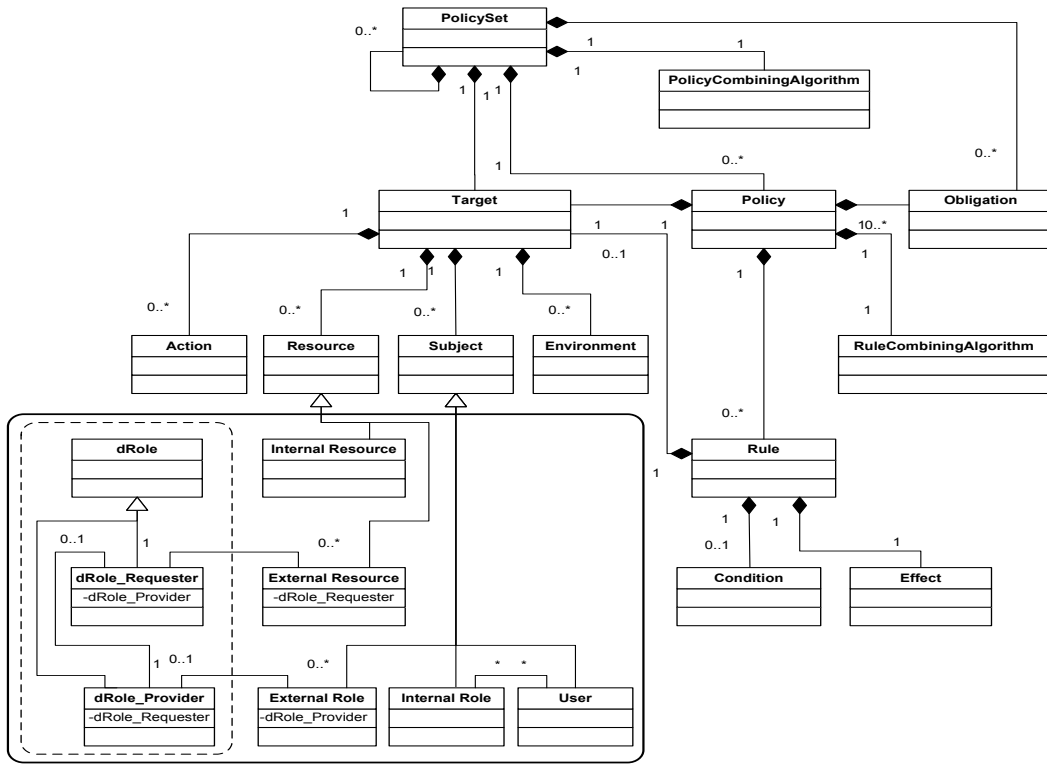


Figure 2: Proposed Extension to XACML Specifications

to all participants. In the e-Government scenarios, they are often the externally known positions and sets of privileges associated with them. Incorporating this piece of knowledge can thus make the necessary decentralized authorization more efficient. To do so, we have adapted the notion of distributed role from dRBAC. It provides another layer of abstraction between two different sets of roles from different organizational domains [10]. Through this indirection, roles can be mapped from the public and private viewpoints of a participating organization without causing changes in the global realm.

4.1 XACML and the RBAC Profile

The OASIS standard, eXtensible Access Control Markup Language (XACML) is an applicable solution to express authorization policies. The overall structure of XACML is depicted as a UML class diagram in Figure 2 [2]. The root element of a XACML policy is called *Policy Set*, which acts as a container for the rest of the other elements in the XACML specification. To allow generalization and specialization of policies, *Policy Sets* themselves can be arranged in a hierarchical order. Due to this possibility of having a nested structure, it is necessary, when evaluating XACML policies, that the results from several policies may need to be combined as one. Therefore, each *Policy Set* has a *Policy Combining Algorithm* element, giving instruction for logical operations on the sub-results of all involved policies, to lead to a single result.

The left subtree of the root node, as shown in Figure 2, is *Target* element. It may have *Policy* elements, which apply *Rule* elements to its sibling *Target* node. Based on the *Condition* elements of a *Rule* element, which gets evaluated

as “true” or “false”, an access control verdict turns out to be either *permit* or *deny*. A *Target* element includes four leaf nodes: *Subject*, *Resource*, *Action*, and *Environment*. *Subject* represents entities that want to get an access permission while *Resource* represents an object for which access request is made. Description of a kind of access is specified in *Action* element. This allows multiple types of actions to be operated upon a resource. Lastly, *Environment* element allows to define constraints about the environment where the access can be performed; this could be, for instance, temporal or spatial constraints. If any of the child elements of *Target* element is missing, that means, any value would be permissible for that omitted element. For example, a missing *Subject* element indicates that any user can use the policy belonging to its *Target* to get access.

The Role-based Access Control (RBAC) profile of XACML enables to specify the concept of role and role hierarchy from the RBAC model [1]. With this profile, an authorization service runs in a chain procedure from privilege to role and role to user, at the same time it is still possible to give privileges directly to users as specified in the core specification. The RBAC profile basically extends *Subject* of a *Target* in a way that roles can be legitimate values of *Subject* field. Roles represent sets of privileges [19]. Users may take a part of multiple number of roles, and multiple number of users can be assigned to a single role. A functionality called “multi-role permissions” [1] is also specified in the profile so that more than one role can be required from a user to get an access to a resource. The profile uses the prefix “&role;” [1] in the attribute field of *Subject* or *Resource* element to indicate that this element is a role and not a user. By utilizing

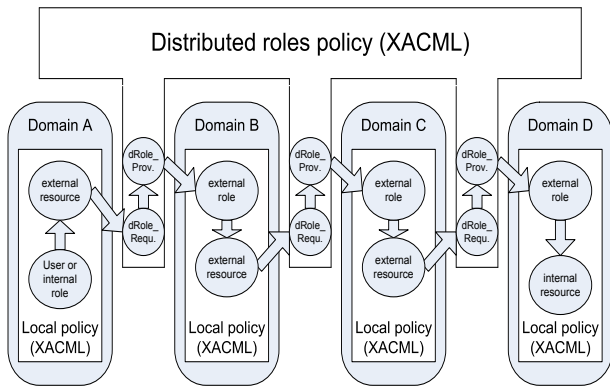


Figure 3: Decentralized Authorization Using the Extended XACML with Delegations

an attribute field, this profile introduces no changes to the original XACML schema file.

4.2 XACML Extension

The extension to the original XACML standard is to realize the concept of distributed roles, which has been adopted from the dRBAC model, in order to link two sets of roles from different domains. Hence, it enables XACML to be capable of specifying authorization policies across organizational boundaries. The extended part is being depicted within the solid box of Figure 2. The elements that are surrounded by a dotted box within the solid box represent a set of distributed roles, which is known to all collaborative partners. Its root element is called *dRole*; further specialization has been made upon this element to distinguish a requester’s side of distributed role (*dRole_Requester*) from a provider’s side distributed role (*dRole_Provider*). This specialization may be useful when there is a clear distinction of requesters and providers amongst distributed roles. In such cases, it can be ensured that an access is only granted from a requester to a provider, but connections between providers and between requesters are prevented.

These specialized distributed roles are associated with the specializations of *Subject* and *Resource* elements, which are called *External Role* and *External Resource* correspondingly. Each of them embodies a connection to a specialized distributed role through a mandatory attribute field containing a link to an appropriate distributed role. As they are just a stub of a specialized distributed role, they do not represent an internal role, to which users or actual internal resources are connected. *External Role* provides an access through a local XACML policy to one or more internal resources that should be made available to other domains. If a local domain acts in the collaboration as a provider, this external role gets connected to *External Role* specialization of a distributed role by having an attribute containing a link to this distributed role. *External Resource* represents the resource of another domain, which is made available to local users, internal roles, or (in case of delegation) even external roles through *dRole_Requester* specialization of *dRole* element.

Connections through distributed roles are bilateral if exactly one domain is connected to each of the instances of a distributed role’s provider and requester parts; however, they can also be multilateral if several external resources or external roles from different domains are connected to

the identical set of distributed roles. A successful access of one user to another domain’s resource will go through the following three steps:

1. Local policy on requester’s domain: A user or one of his local internal roles needs to have the privilege to access the local *External Resource*, which is connected to the appropriate distributed role’s requester part, *dRole_Requester*.
2. A pair of distributed roles: The distributed role has to be associated to the requester’s domain(s) with its requester part and to the provider’s domain(s) with its provider part.
3. Local policy on provider’s domain: *External Role*, which is referenced by the attribute in the distributed role’s provider part, *dRole_Provider*, must have the privileges to access the local resource, which is defined in a local policy. In case of delegation, the local policy can also refer to a local *External Resource*, which would involve another new domain’s resource and requires an additional pair of the specializations of *dRole*. In this case, this step can be recursively followed by step 2. Refer to Figure 3 for a graphic representation of this scenario.

The extension of *dRole*, which is enclosed within the dotted box in Figure 2 can be specified using the existing XACML core schema. *dRole_Requester* part can be represented as a *Subject* element while *dRole_Provider* part, as a value of *Resource*. The necessary links to the local domains can be specified through their attribute fields. All distributed role policies can have a default *Condition* element, that limits their usage to the context of distributed role specifications.

As for the extension of a local part of XACML policies, which is depicted inside of the solid box and outside of the dotted box in Figure 2, new prefixes in the attribute fields of *Subject* (“&externalRole;”) and *Resource* (“&externalResource;”) can be used to indicate the specialization in a similar way the RBAC profile expresses the notion of roles (i.e. “&role;”). The specialized elements will contain additional attribute fields, containing the link to their associated distributed roles, which is mandatory.

The extension of XACML with distributed roles preserves the autonomy of collaborating domains since local policies (compare: step 1 and 3) remain within the scope of local policy administrators. Negotiation between the collaborating domains is necessary only to define common distributed roles (step 2). Entitled local users (directly or through their local roles) and provided resources can be changed without inquiring administrators of other domains. This simplifies administration efforts and allows flexible reactions with respect to changing local demands. Nevertheless, administrators need to have global demands in mind when they change local policies, especially the part that has any connection to distributed roles.

A couple of limitations of this approach can be foreseen. The first one is that each domain can only specify their authorization policies for outside requests in terms of abstract notion of roles. If an identity-based access control is desired, this scheme may be unsuitable. Secondly, composing the extension of XACML may be an error prone process – both

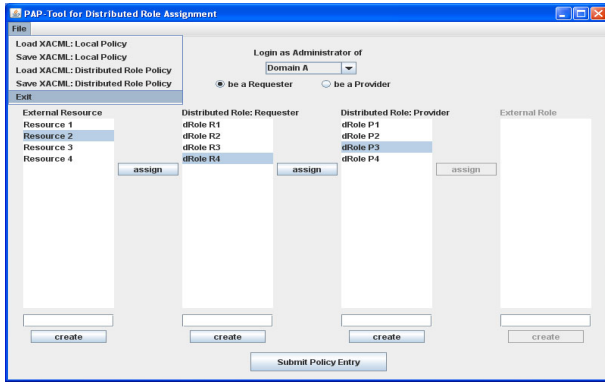


Figure 4: User Interface of PAP Toolkit

syntactically and semantically. In order to ensure the policies are composed correctly, a policy administration toolkit is designed to create the corresponding extension of XACML as modeled above.

4.3 PAP Toolkit

For every successful collaboration, minimum two local policies and one pair of entries in the list of distributed roles are required. To ease the administrative effort, a GUI-based toolkit has been designed for policy administrators. This tool can be used to generate or edit the XACML policy instances that link local roles or users to distributed role sets or to link distributed role sets to access control lists. It also allows to create a distributed role entry and import or export a set of distributed roles across different administrative domains. This tool can also be used for a compliance check for the core XACML specification and the extension introduced in this paper. However, it is intended to support composing authorization policies for inter-domain collaborations only and not meant to replace administrative tools for internal local policies.

Figure 4 shows a screen shot of the Java implementation of the toolkit. The main elements of the tool are four columns; the two in the middle show the contents of the distributed roles while the other two columns show elements from the local policies. The leftmost column represents the elements that can be used as requesters, and the rightmost column lists the ones that can be providers. Dependent upon the selection of the radio buttons on the top of the frame, an administrator can specify requesting side policies or providing side ones. The rightmost column in Figure 4 is disabled because a requesting point of view has been selected. By selecting two elements of two lists next to each other and pressing the “assign” button between them, the connection between the selected local entry and the distributed role entry is defined. An entry can be created for any of the columns described by filling a textfield below a column and pressing “create” button below the text entered textfield. When all mappings between entries of the local and distributed role policies are set, the corresponding XACML instance can be generated by clicking the “submit-policy-entry” button.

5. RELATED WORK

Various models extended from RBAC are intended to accommodate decentralized authorization. Generalized Tem-

poral Role Based Access Control (GTRBAC) [18], Composite RBAC [17], X-RBAC [12], and domain based access control model [8] are some of the examples. Among these, Joshi et al.’s X-RBAC is the first model to use RBAC in the context of inter-domain collaborations [12]. Demchenko et al. propose in [8] an access control model for distributed collaborative applications, which is an extension to the major service-oriented and Grid-based access generic framework such as *Acegi*, Globus Toolkit Authorization framework, and GAAA authorization framework. Their policy specification is an extension of RBAC with domain based security context, and their implementation is using the XACML framework.

Traditional trust management systems such as KeyNote [6] and SDSI/SPKI [9] can be used as a decentralized authorization service as well, but they need to be extended with credential discovery and revocation mechanisms. Role-based trust management (*RT*) combines the strengths of role-based access control and trust management systems and is capable of expressing attribute-based access control [13]. This is not the case with the traditional trust management systems. In addition to more flexible expressiveness of semantics, the advantages of *RT* over the previous trust management solutions include: a declarative, logic-based semantic language specification, strongly-typed credentials and policies, more flexible delegation structures, and more expressive support for separation of duty constraints [14]. While *RT* systems do offer variety of features for a decentralized access control, they require significant support infrastructures and extra installations for their framework to work properly.

dRBAC stands for distributed Role-based Access Control system, and it claims to provide a decentralized access control mechanism through a complete delegation solution that delegates roles to other roles or entities in another local name space where entities may refer to users or resources. Its infrastructure includes *proof monitors*, among others, which verifies delegation chains at each entity [10]. This model takes a similar approach to *RT* in the usage of a delegation chain for establishing trust and deferring authorization decisions. Ma and Woodhead has applied this model to resolve the problems of identity management inherent in a distributed subscription-based resource sharing environment [16]. Using Distributed Role Markup Language (DRML), they specify distributed roles related entities, role constraints, and predicates. This application of dRBAC for subscription-based resource providers does get its mileage out of the abstract layer of external roles as a number of requesters per a given resource increases.

Recently, Jin and Ahn have developed a Role-based Access Management for Resource Sharing (RAMARS) framework, which also enables a distributed policy propagation and supports generic sharing roles amongst collaboration participants [11]. According to RAMARS, *Originator* is the one who defines the roles for collaborations, and *Resource Owner* has a full right to decide which access permission to give to which roles. As far as the extending of the XACML specification is concern, our approach is similar to their implementation.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a lightweight solution of a decentralized authorization model for inter-domain collabo-

rations through adaptation of the notion of distributed roles from d-RBAC model. We have implemented our model by extending the current XACML specification and its RBAC profile to accommodate distributed roles. This solution simplifies decentralized authorization process by replacing the scheme of delegating authorization decision authorities with usage of distributed roles. This appears to be a more efficient and simplified solution than existing solutions, especially for e-Government scenarios where externally known roles are publicly available.

We plan to further develop the notion of distributed roles and introduce a hierarchical structure within this realm of those roles, possibly again extending the XACML standard if necessary; meanwhile, we will continue to focus on keeping it simple and small so as to satisfy the major requirements of easy deployment and compatibility with the related standards for inter-domain collaborations. As for the policy administration point (PAP)'s toolkit, we would like to provide functionalities for PAP's to import and export a global view of an authorization schema of their collaboration without disclosing their local view of authorization policies. This can allow PAP's to collaboratively create distributed roles when pre-defined roles do not exist.

7. ACKNOWLEDGMENTS

This work has been partially funded by the EU Commission under the contract number IST-2004-026650 through the EU integrated project, R4eGov.

8. REFERENCES

- [1] 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, Feb. 2005, (Last visited on 8/23/2007).
- [2] 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, Feb. 2005, (Last visited on 8/23/2007).
- [3] A. D. A. Boujraf and M. Noble. Towards e-Administration in the Large (R4eGov). *Deliverable WP3-D7*, 2007.
- [4] A. Belokosztolszki, D. M. Eysers, and K. Moody. Policy Contexts: Controlling Information Flow in Parameterised RBAC. In *Policy 2003: IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, pages 99–110, 2003.
- [5] A. Belokosztolszki and K. Moody. Meta-Policies for Distributed Role-Based Access Control Systems. In *Policy 2002: IEEE 3rd International Workshop on Policies for Distributed Systems and Networks*, pages 106–115, 2002.
- [6] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. The KeyNote Trust-Management System Version 2. RFC 2704, 1999.
- [7] T. Clark and R. Jones. Organisational Interoperability Maturity Model for C2. *1999 Command and Control Research and Technology Symposium*, June 1999.
- [8] Y. Demchenko, C. de Laat, L. Gommans, and R. van Buuren. Domain Based Access Control Model for Distributed Collaborative Applications. In *E-SCIENCE '06: Proceedings of the Second IEEE International Conference on e-Science and Grid Computing*, page 24, Washington, DC, USA, 2006.
- [9] C. Ellison. SPKI Requirements. RFC 2692, September 1999.
- [10] E. Freudenthal, T. Pesin, L. Port, E. Keenan, and V. Karamcheti. dRBAC: Distributed Role-Based Access Control for Dynamic Coalition Environments, 2002.
- [11] J. Jin and G.-J. Ahn. Role-based Access Management for Ad-hoc Collaborative Sharing. *SACMAT '06: Proceedings of the eleventh ACM symposium on Access control models and technologies*, pages 200–209, 2006.
- [12] J. B. Joshi, R. Bhatti, E. Bertino, and A. Ghafoor. Access-Control Language for Multidomain Environments. *IEEE Internet Computing*, 08(6):40–50, 2004.
- [13] N. Li, J. C. Mitchell, and W. H. Winsborough. Design of a Role-based Trust Management Framework. In *Proc. IEEE Symposium on Security and Privacy, Oakland*, May 2002.
- [14] N. Li, J. C. Mitchell, W. H. Winsborough, K. E. Seamons, M. Halcrow, and J. Jacobson. RTML: A Role-based Trust-management Markup Language. Technical report, Purdue University, 2004.
- [15] J. Lopez, R. Oppliger, and G. Pernul. Authentication and Authorization Infrastructures (AAIs): a Comparative Survey. *Computers & Security*, Volume 23, Issue 7:578–590, 2004.
- [16] M. Ma and S. Woodhead. Constraint-Enabled Distributed RBAC for Subscription-Based Remote Network Services. In *CIT '06: Proceedings of the Sixth IEEE International Conference on Computer and Information Technology (CIT'06)*, page 160, Washington, DC, USA, 2006. IEEE Computer Society.
- [17] J. S. Park, K. P. Costello, T. M. Neven, and J. A. Diosomito. A Composite RBAC Approach for Large, Complex Organizations. In *SACMAT '04: Proceedings of the ninth ACM symposium on Access control models and technologies*, pages 163–172, New York, NY, USA, 2004. ACM Press.
- [18] S. Piromrueen and J. B. D. Joshi. An RBAC Framework for Time Constrained Secure Interoperation in Multi-domain Environment. *IEEE Workshop on Object-oriented Real-time Databases (WORDS-2005)*, 2005.
- [19] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-Based Access Control Models. *IEEE Computer*, 29(2):38–47, 1996.