Using Subject- and Object-specific Attributes for Access Control in Web-based Knowledge Management Systems

Gerald Stermsek, Mark Strembeck, Gustaf Neumann
Department of Information Systems, New Media Lab
Vienna University of Economics and BA, Austria
{firstname.lastname}@wu-wien.ac.at

Abstract

In this paper we present an approach to use subject- and object-specific attributes defined as RDF metadata to specify and enforce access control policies for web-based information systems. We give an overview of the architecture and implementation of our approach.

1 Introduction

The growing amount of electronically managed data is frequently referred to as a motivation to deploy knowledge management systems (KMS) (see, e.g., [10]). Knowledge management systematically supports gathering, organizing and disseminating (structured) information. If a KMS stores sensitive information, proper security management is a major concern. In particular, adequate security measures should prevent unauthorized access to classified data. Web-based knowledge management is an area of emerging interest in the semantic web context (see, e.g., [9]). Well established web technologies as XML [6] or HTTP [13] provide the foundation to deploy web-based KMS, and an implementation of a web-based knowledge management system using well established standards and existing software components constitutes a reduced effort compared to a proprietary "from scratch implementation". Standards for web-based information systems allow for a transparent access to arbitrary documents, meaning that the physical location of documents is irrelevant for their retrieval. For example, the Digital Object Identifier [23] and the Uniform Resource Name [3] mechanisms provide corresponding functionality.

In general, a Web-based architecture is primarily focused on the widespread dissemination and easy access to information sources. Therefore, issues like access control initially played a minor role in web environments. Standards like HTTP authentication [14] only provide simple security measures for systems requiring only a low security level. However, for sensitive information managed via a web-based KMS more sophisticated and fine-grained access control measures need to be established. Traditionally, access control is based on $\langle subject, operation, object\rangle$ triples to decide if a certain subject is allowed to perform a particular operation on a specific object. Thus, it is required that subjects and objects are unambiguously identified to decide on a certain access request.

The Resource Description Framework (RDF) [7, 17] is a central standard in the semantic web context. RDF allows to define arbitrary attributes to describe arbitrary entities. For example, via RDF statements subjects and objects in an information system can be associated with additional attributes aside from unique identifiers. In this paper, we present an approach that uses RDF meta-data describing subjects and objects to render access control decisions.

1.1 Resource Description Framework

The Resource Description Framework (RDF) [17, 7] provides a standard for the description of information resources on the World Wide Web. In particular, RDF statements are metadata about Web resources, for example the title, the author, the size, or the topic area of a certain web-document. RDF statements are human-readable and can be automatically processed by software applications. RDF uses Web identifiers, called *Uniform Resource Identifiers* (URIs) [3] to refer to Web resources and to associate these resources with properties and property values.

A single RDF statement can be written as a $\langle subject, property, value\rangle$ or $\langle subject, predicate, object\rangle$ triple (see also [17]). Moreover, RDF statements can be visualized as graph of nodes and arcs describing a specific resource. In [2] a syntax is defined to express and exchange RDF statements via XML documents. Figure 1 shows a simple example of a subject with the attributes nationality, birthday and project in graphic and XML representation.

The RDF schema standard [7] enables the definition of vocabularies that can be used in RDF statements to describe resources. In particular, an RDF schema defines classes (representing specific resource types) and properties that are asso-

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:attributes="http://nm.wu-wien.ac.at/rdf/attributes#">
  ...
  <rdf:Description rdf:about="http://nm.wu-wien.ac.at/rdf/entities#PublicKey25097">
    <rdf:type rdf:resource="http://nm.wu-wien.ac.at/rdf/attributes#Subject"/>
    <attributes:nationality>Austrian</attributes:nationality>
    <attributes:birthday>1977-07-07</attributes:birthday>
    <attributes:project>Project X</attributes:project>
  </rdf:Description>
  ...
</rdf:RDF>
```

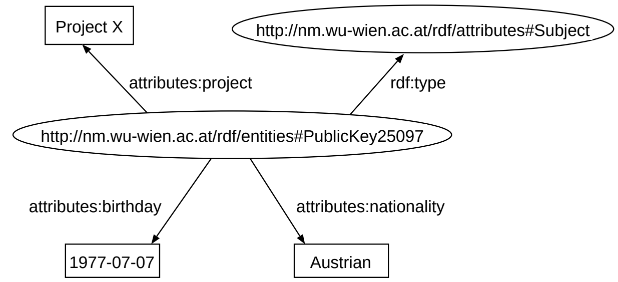

Figure 1: A simple RDF and RDF/XML example

ciated with these classes. Properties are applied to describe concrete instances of the respective classes.

1.2 Approach

A Policy Decision Point (PDP) is a software component which decides if a certain action in an information system conforms to the set of active policy rules. An access control monitor is a specific type of PDP which renders access control decisions. In our approach, the access control monitor uses additional attributes (aside from the traditional $\langle subject, operation, object\rangle$ triple) to render access control decisions. In particular, we use RDF to define these subject- and object-specific attributes. We chose RDF since it is very flexible and can seamlessly be integrated in a semantic web context.

In principle subject and object attributes may be retrieved from different locations, for example from a local database, from a trusted third party, or directly from a client along with a certain access request. Nevertheless, before these attributes can actually be used in access control decisions the respective attribute document needs to validated. Such a validation procedure at least includes an integrity check of the respective document and a validation of the corresponding trust chain associated with an attribute document (to check if the party granting a certain attribute is actually trusted to make this statement, see e.g. [5]).

For validation purposes, attribute documents can be provided with a digital signature, e.g. by using the XML-Signature standard. The XML-Signature specification [11] defines an XML-syntax and processing rules for creating and representing digital signatures. XML-Signature can be used for both signing and verifying attributes and securing data included in arbitrary digital documents. For object attributes the asset owner (the server) can "self-sign" the attributes.

A mechanism has to be provided to specify and check which attributes are allowed to describe particular entities in a specific information system. In our approach, we apply the RDF Schema standard [7] to define attribute vocabularies. These vocabularies need to be defined prior the definition and subsequent analysis of attribute values, of course. A simple example for an access decision based on attributes may be a subject requesting a web page from the intranet of an organization, where the subject has to provide an *employer* attribute to enable access control decisions based an that attribute.

Note that different types of attributes may change in different time intervals. A subject's birthday, for example, may not change at all while a subject's employer could change once in a while. Moreover, object attributes may change even more often, e.g. the most recent editor of a certain document.

As mentioned above, access control decisions are based on a set of authorizations, and traditional authorization rules are represented via $\langle subject, operation, object\rangle$ triples. These triples specify that a certain *subject* is authorized to execute a certain *operation* on a specific *object*. In our approach, additional attributes that can be assigned to the elements of these triples influence access control decisions. We consider two possible scenarios to enforce access control based on subject- and object specific attributes:

- *Access decisions directly based on subject- and object attributes:* In this scenario subject- and object-specific attributes are directly used to render access decisions. In particular, we specify policy rules that define which attributes and attribute values are needed by a subject to access an object (of course, different application domains may require different attribute sets). For example, a policy rule may specify that certain subject and object attribute values must be equal to grant a specific access request, e.g. "$subject_{project}$ == $object_{project}$". Another simple option is to compare attribute values, e.g. "$subject_{balance} > object_{costs}$".
- *Assignment of permissions and/or roles based on subject attributes:* This scenario is essentially based on a classification of subjects with respect to subject-specific attributes. In particular, the attributes of the requesting subject are used to decide which permissions and/or roles are assigned to this subject. The respective roles and permissions are defined in advance. This scenario especially

requires the definition of assignment policies that specify which attributes and attribute values a subject must provide to qualify for a specific role. Subsequently (after the assignment is completed) the PDP renders access decisions for the respective subjects based on these assignments (typically the assignment is valid for exactly one session). In other words: here, subject-specific attributes are primarily used to assign roles and permissions to users rather than directly for access control decisions. This option could especially be applied in more static environments and may result in a better runtime performance of the PDP due to fewer attribute checks.

Object attributes can be directly defined by the corresponding owner. Subject attributes, on the other hand, can be maintained in a specific database or be assigned to the subject via attribute certificates (see, e.g., [12]), for example. In case the subject attributes are stored on the client side (or by a trusted third party), we differentiate three options for providing the server application with the required subject attributes:

- *Client sends attributes:* Using this option, the client sends his subject attributes to the server when requesting a certain service. Depending on the concrete application domain, however, this option may be suboptimal, since the client may not be able to decide which attributes he is willing to disclose to the server. Moreover, the server application might not require all attributes of the client for a particular access control decision.

- *Server asks for particular attributes:* After the initial client request, the server application asks the client to provide particular attributes which are required for the access control decision. This option may require user interaction, i.e. the user may have to resend the request along with the required attributes.

- *Trust negotiation:* Trust between strangers can be established via a trust negotiation protocol (see, e.g., [24]). For example, the different parties exchange digital credentials. Specific policies define what kind of credentials a stranger must disclose in order to gain access to a particular resource. After a particular trust level has been established, access to the requested object may be granted.

Figure 2 shows an example for subject- and object-specific attributes formulated as XML/RDF statements. The subject in our example is identified via a public key and, among others, associated with the *project* attribute. Our example object is identified via an URL and is also associated with a *project* attribute. For example, a specific policy rule may define that subjects may only access objects that are associated with the same project.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:attributes="http://nm.wu-wien.ac.at/rdf/attributes#">
  ...
  <rdf:Description rdf:about="http://nm.wu-wien.ac.at/rdf/entities#PublicKey25097">
    <rdf:type rdf:resource="http://nm.wu-wien.ac.at/rdf/attributes#Subject"/>
    <attributes:nationality>Austrian</attributes:nationality>
    <attributes:birthday>1977-07-07</attributes:birthday>
    <attributes:project>Project X</attributes:project>
    ...
  </rdf:Description>
  ...
</rdf:RDF>

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:attributes="http://nm.wu-wien.ac.at/rdf/attributes#">
  ...
  <rdf:Description rdf:about="http://nm.wu-wien.ac.at/internatl/document1.xml">
    <rdf:type rdf:resource="http://nm.wu-wien.ac.at/rdf/attributes#Object"/>
    <attributes:category>internal</attributes:category>
    <attributes:creator>Jonny Bravo</attributes:creator>
    <attributes:project>Project X</attributes:project>
    ...
  </rdf:Description>
  ...
</rdf:RDF>
```

Figure 2: Sample RDF properties for subjects and objects

Subject-specific attributes are frequently referred to as credentials (see, e.g., [8]). In a credential-based system, permission assignment is (directly or indirectly) based on additional attributes aside from unique identifiers (the credentials), e.g. birthday, nationality, or employer. According to Chaum [8], credentials are “statements based on individual’s relationship with organizations that are, in general, provided to other organizations”. A credential therefore is a digitally signed document that binds attributes (and thereby authorizations) to public keys rather than to individual users. This constitutes a form of anonymity since users are not (necessarily) directly identified and credentials can also be delegated (see [8]). A popular approach is to attached credentials to digital certificates, e.g. X.509 certificates [15].

2 High-level Architecture

This section describes the high-level architecture of our approach (see Figure 3). A corresponding overview of a prototype implementation can be found in Section 3. The architecture consists of five main components:

- *Policy Decision Point:* The Policy Decision Point (PDP) applies a set of policy rules to decide if a certain action is in accordance with these rules and can be granted or must be denied. In our approach, the PDP receives the traditional $\langle subject, operation, object \rangle$ triple as well as subject- and object-specific attributes as input parameters to render access control decisions.

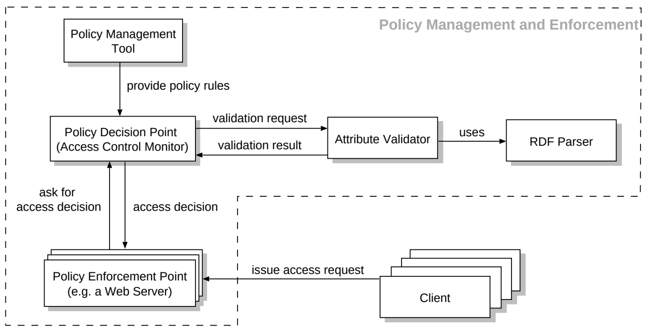

Figure 3: High-level Architecture

- *Policy Enforcement Point*: A Policy Enforcement Point (PEP) interacts with client applications and enforces the decisions of the PDP (e.g. by allowing or denying access to resources).

- *Attribute Validator:* This component ensures the integrity of attribute documents. In case the attributes are presented as XML/RDF statements, the XML-Signature standard [11] can be applied to sign and validate these attributes. After document integrity is assured the component checks if the respective attribute document was issued by an authorized entity (i.e. if the the party issuing a certain attribute document was actually authorized to do so, see also [5]).

- *RDF Parser:* After the validator component assured the integrity and validity of an attribute document, the parser is applied to extract the respective attribute values from the document.

- *Policy Management Tool:* This tool allows for the definition of policy rules and feeds the rules in the PDP.

3 Prototype Implementation

In this section, we give an overview of our prototype implementation. Our proof-of-concept implementation of the architecture depicted in Figure 3 is based on existing and tested components which are tailored to the specific needs of this project. For example, we use the RDF parser, and web server components provided by the ActiWeb framework [21]. In principle, various models and/or technologies can be used to implement attribute-based access control measures. For our implementation we use the xoRBAC component (see [19, 20]) as policy decision point (PDP). In role-based access control (RBAC) permissions are assigned to roles and roles are assigned to subjects. RBAC is policy neutral, and a suitable RBAC-service can be configured to support many different access control models (see, e.g., [22]). To achieve a high flexibility and applicability for real-world application problems, the xoRBAC component also allows for a direct permission-to-subject assignment (see also [19, 20]).

Figure 4 depicts a message sequence chart for the processing of a client request. Note that we only discuss one particular interaction sequence. Depending on the applied interaction scheme different sequences are possible, of course (see also Section 1.2). In Figure 4 the client first sends a `get` request to the PEP (here: an ActiWeb web-server component). Along with the request the client submits the URN of the target object, a certificate containing his public key, and an attribute document containing subject-specific attributes. The PEP first validates the client certificate and then fetches the attribute document associated with the target object (identified via the URN). Subsequently, it invokes the `check` method of the PDP. Along with this invocation it sends the subject-id (e.g. the public-key extracted from the client certificate), the operation (here: `get`), the object-id (the URN), as well as the attribute documents of the respective subject and object. Next, the PDP uses the attribute validator component to validate the attribute documents. After validating the documents, the validator component uses the RDF parser to extract the attribute values from the documents and returns these values to the PDP. Now, the PDP checks its set of policy rules to decide if the request can be allowed and returns its decision back to the PEP. The PEP, in turn, enforces this decision by either returning the requested object or an error message to the client.

4 Related Work

Adam et al. introduce a sophisticated authorization model that was specifically designed to meet access control requirements of digital libraries [1]. In particular their model allows for the consideration of additional user and object attributes aside from unique identifiers. Here, credentials represent attributes that describe certain characteristics and qualifications of users, like age, salary, nationality, or current project involvement for example. Likewise, attributes describing the content of digital library objects are stored (e.g. taxation, civil law, information system research), and digital library objects are divided in different segments, like authors, abstract, sections, bibliography. These information are then used to define fine grained access control policies. That enables the definition of access rights on individual objects (based on their IDs), or on specific parts of a set of objects (like the abstract and author information of all research papers), or on all objects that comprise certain contents, e.g. all objects concerning import taxes. Similarly users may acquire permissions explicitly (through their ID), or implicitly through their characteristics/credentials, e.g. all users with a specific age or nationality.

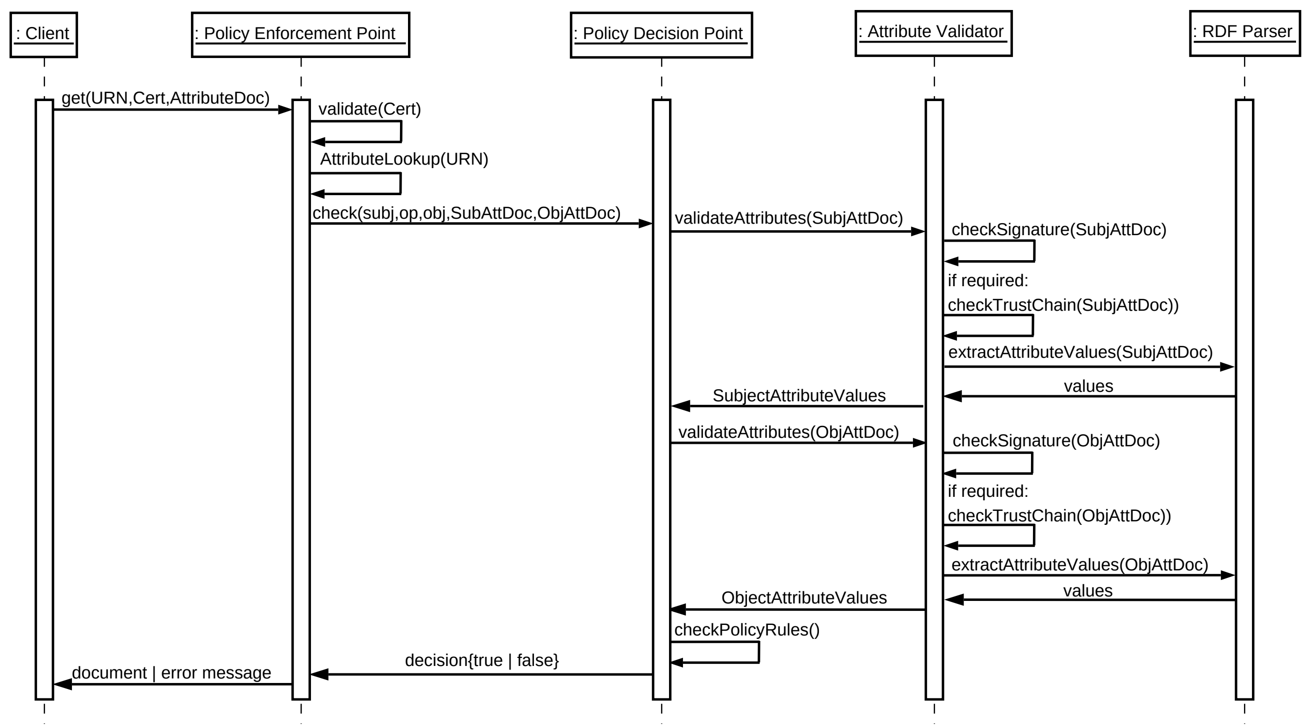

Figure 4: Message sequence chart for a sample client request

Biskup and Wortmann describe a layered approach to access control for distributed and interoperable computing systems [4]. Here, access control policies are declarative statements that define access to particular resources. They use a policy algebra to compose access control policies for the use in distributed systems and describe a credential-based implementation using this policy algebra.

The Security Assertion Markup Language (SAML) [18] defines an XML-based framework for exchanging security information via computer networks. It is based on the SAML protocol which consists of XML-based request and response messages. By this protocol, clients can request assertions from so-called "SAML authorities" (trusted servers). SAML authorities can make three different kinds of assertion statements: authentications, authorization decisions, and attributes. An authentication assertion confirms that a specific subject has been authenticated by a particular means at a particular time. An authorization decision assertion states that a particular access request consisting of a $\langle subject, operation, object \rangle$ triple has been granted by the corresponding SAML authority. Finally, an attribute assertion confirms that a specific subject is associated with a certain set of attributes.

In automated trust negotiation [24] specific policies define the type and order of credentials two parties have to disclose to establish a trust relationship. In [24] Yu and Winslett present an approach to support structured credentials and sensitive policies through interoperable strategies for automated trust negotiation. Kahan [16] describes a distributed authorization model, where node servers are grouped by authorization domains. A server grants access to requested documents based on capabilites presented by the requesting client. Clients acquire capabilites from authorization- or document-servers.

5 Conclusion

In this paper we presented an approach to use subject- and object-specific attributes for access control measures in web-based information systems. In particular, we gave an overview of an architecture and a prototype implementation that uses RDF-based attribute documents. In our future work, we further investigate the definition and enforcement of attribute-based access control policies.

References

[1] N. R. Adam, V. Atluri, E. Bertino, and E. Ferrari. A Content-Based Authorization Model for Digital Li-

braries. *IEEE Transactions on Knowledge and Data Engineering*, 14(2), 2002.

[2] D. Beckett. RDF/XML Syntax Specification. http://www.w3.org/TR/rdf-syntax-grammar/, February 2004. W3 Consortium Recommendation.

[3] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifiers (URI): Generic Syntax. http://ietf.org/rfc/rfc2396.txt, August 1998. IETF, RFC 2396 (Standards Track).

[4] J. Biskup and S. Wortmann. Towards a Credential-Based Implementation of Compound Access Control Policies. In *Proc. of the 9th ACM Symposium on Access Control Models and Technologies*, June 2004.

[5] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized Trust Management. In *Proc. of the IEEE Symposium on Security and Privacy*, May 1996.

[6] T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler, F. Yergeau, and J. Vowan. Extensible Markup Language (XML) 1.1. http://www.w3.org/TR/xml11/, February 2004. W3 Consortium Recommendation.

[7] D. Brickley and R.V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. http://www.w3.org/TR/rdf-schema/, February 2004. W3 Consortium Recommendation.

[8] D. Chaum. Security without Identification: Transaction Systems to make Big Brother Obsolete. *Communications of the ACM*, 28(10), October 1985.

[9] J. Davies, F. van Harmelen, and D. Fensel. *Towards the Semantic Web: Ontology-driven Knowledge Management*. John Wiley & Sons, 2002.

[10] P.F. Drucker. The Coming of the New Organization. In *Harvard Business Review on Knowledge Management*. Harvard Business School Press, 1999.

[11] D. Eastlake, J. Reagle, and D. Solo. XML-Signature Syntax and Processing. http://www.w3.org/TR/xmldsig-core/, February 2002. W3 Consortium Recommendation.

[12] S. Farrell and R. Housley. An Internet Attribute Certificate Profile for Authorization. http://www.ietf.org/rfc/rfc3281.txt, April 2002. IETF, RFC 3281 (Standards Track).

[13] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. http://ietf.org/rfc/rfc2616.txt, June 1999. IETF, RFC 2616 (Standards Track).

[14] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, E. Sink, and L. Stewart. HTTP Authentication: Basic and Digest Access Authentication. http://www.ietf.org/rfc/rfc2617.txt, June 1999. IETF, RFC 2617 (Standards Track).

[15] ITU-T. Information Technology - Open Systems Interconnection - The Directory: Authentication Framework, 1993. ITU-T Recommendation X.509.

[16] J. Kahan. A Capability-Based Authorization Model for the World-Wide Web. In *Proc. of the 3rd International World-Wide Web Conference on Technology, Tools and Applications*, 1995.

[17] G. Klyne and J.J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. http://www.w3.org/TR/rdf-concepts/, February 2004. W3 Consortium Recommendation.

[18] E. Maler, P. Mishra, and R. Philpott. Security Assertion Markup Language (SAML) v1.1. http://www.oasis-open.org, September 2003. OASIS Standard.

[19] G. Neumann and M. Strembeck. Design and Implementation of a Flexible RBAC-Service in an Object-Oriented Scripting Language. In *Proc. of the 8th ACM Conference on Computer and Communications Security*, November 2001.

[20] G. Neumann and M. Strembeck. An Approach to Engineer and Enforce Context Constraints in an RBAC Environment. In *Proc. of the 8th ACM Symposium on Access Control Models and Technologies*, June 2003.

[21] G. Neumann and U. Zdun. Distributed Web Application Development with Active Web Objects. In *Proc. of the 2nd International Conference on Internet Computing*, June 2001.

[22] S. Osborn, R. Sandhu, and Q. Munawer. Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies. *ACM Transactions on Information and System Security*, 3(2), February 2000.

[23] N. Paskin. *DOI Handbook*. International DOI Foundation, April 2004.

[24] T. Yu, M. Winslett, and K.E. Seamons. Supporting Structured Credentials and Sensitive Policies through Interoperable Strategies for Automated Trust Negotiation. *ACM Transactions on Information and System Security*, 6(1), 2003.