

Research and Design Issues in Access Control for Network Services on the Web

Reiner Kraft
IBM Almaden Research Center
San Jose, CA, U.S.A.

Abstract—The service oriented architecture (SOA) is gaining more momentum with the advent of network services on the Web. A programmable and machine accessible Web is the vision of many, and might represent a step towards the semantic Web. However, security is a crucial requirement for the serious usage and adoption of Web services technology. This paper reviews existing work related to Web service security and access control. Different access control models for Web services are enumerated, and research and design issues related to access control that arise in a world of network services on the Web are discussed. The paper's main goal is to stimulate more research in the area of access control and security for Web services.

Keywords— Web, Web services, security, access control

I. INTRODUCTION

WEB services promise to promote the vision of a machine accessible Web, which can be used as a platform to conduct e-commerce and provide interoperability across organizations leveraging from a global Web infrastructure. Recent informal polls [1] showed that security was the top issue among those considering Web services. When decision makers were asked what are the biggest obstacles to implement Web Services, 45.5% pointed out security and authentication issues. A common fear among many IT decision makers is that it is not clear what level of exposure opening up an organization's Intranet to deploy Web services will have. There is clearly a difference between a Web site and a Web service: A Web site can be set up in a secure environment (e.g., behind a company's firewall) serving static pages, or dynamically created ones based on user input from forms. This is a common scenario and people know how to make it relatively secure. However, a Web service may expose a company's secure back office and business logic for transactions to the public, potentially opening up a large security hole for hackers. Despite all standardization attempts on a common messaging framework (e.g., SOAP [2]), Web service interfaces (e.g., WSDL [3]), work flow (e.g., WSFL [4]), and service discovery (e.g., UDDI [5], WSIL [6]) the Web service architecture will not be broadly adopted as long as there are no standard ways on securing Web services.

Security efforts in the area of Web services are mostly related to data integrity, non-repudiation, and encryption. Web services have to be reliable. There must be assurances regarding the identity of the systems and principals that interact (authentication), messages are delivered once and only once, and all business processes are completed.

For instance, *XML Key Management (XKMS)* [7] de-

fines protocols for distributing and registering public keys. There are established standards on how to encrypt documents (e.g., SSL[8]) and digitally sign those. *XML Encryption* [9] and *XML Digital Signature* [10] handle more complex situations, where different parts of the same document need different treatment (e.g., parts of a document need to be signed, perhaps by different people, and this may need to be done in conjunction with selective encryption). The *SOAP Security Extensions* [11] propose a standard way to use the XML Digital Signature syntax [10] to sign SOAP [2] messages. Furthermore, the *Organization for the Advancement of Structured Information Standards (OASIS)*[12] proposed a *Business Transaction Protocol (BTP)* [13] that is designed to allow transactional coordination of participants, which are part of services offered by multiple autonomous organizations (as well as within a single organization) in a Web Service environment. In addition, Microsoft is working on a *Global XML Web Services Architecture (GXA)* [14], which includes a proposal called the Web service security language (WS-Security) [15].

The paper first presents a comparison of different access control models (e.g., mandatory, role-based, chain of trusts) and discusses how they are suited for Web services. Then it identifies research issues on authorizations for Web services. Towards the end the paper critically reviews recent work related to security with a focus on access control in Internet information systems. The paper's main contribution is to provide an overview of existing work and to propose ideas that are intended to stimulate more research in the area of authorization issues and access control in a distributed Web service environment.

The conclusion is that there is no existing integrated solution and architecture available yet that addresses access control for Web services adequately. Such a solution is required as a foundation towards a broad deployment and usage of Web services to materialize the vision of a programmable Web.

II. ACCESS CONTROL MODELS FOR WEB SERVICES

This section will first review existing access control models on how they can be applied to Web services. Sandu and Samarati [16] provide an overview of access control principles and practice. They discuss security goals, the traditional access control matrix, along with some implementation approaches (e.g., access control lists, capabilities, authorization relations). Merkl et al. [17] present a detailed

discussion of different security models (e.g., discretionary, mandatory, role-based) in regards to their applicability in hypertext systems.

There are similarities between hypertext systems and Web services. The Web represents the world biggest hypertext system: It comprises Web servers that communicate to Web browsers or other clients using the HTTP [18] protocol. A Web server receives a request, computes a response for that request (e.g., compiles a dynamic HTML page), and returns that response (e.g., in form of HTML). Therefore a Web server provides a document viewing or representation service. It has a well-defined interface based on the standardized HTTP protocol. From a conceptual standpoint we can argue that a Web server is a special (limited) type of Web service. A more careful study on how hypertext systems relate to Web services would help to gain more insights about Web services security. Nevertheless, it seems to be a reasonable approach to adopt some established security models related to hypertext systems [19] as a basis and extend these appropriately to accommodate the goals and requirements for Web service security.

All presented access control models share in common the assumption that identification, authentication, and auditing of actions on behalf of the users is being properly taken care of (e.g., by the underlying operating system or Web server), such that the described models do not have to be concerned with these types of security checks.

A. The discretionary access control model

The discretionary access control model enjoys a high popularity among the research community [19], [16], [17] and is widely commercially used (e.g., in operating systems, Web servers, or RDBMS). The central idea in this model is assigning privileges on objects to subjects. An access control decision is based on the user's identity, an access mode, an object, and some predicates (to represent content based access rules). The discretionary model is widely used mostly because of its flexibility. It works fine in cooperative autonomous environments, but has drawbacks in companies and organizational structures. This is because of its delegation of rights concept. An owner may grant privileges to another subject. However, in a company a user might not be the owner of an information piece, although he/she is the author. More problems arise with cascading authorization (cascading revocation chains), and interactions between positive and negative authorizations can become complicated. Damiani et al. [20] argue they are necessary to make the security model flexible enough, and allow exceptions. Further, the discretionary model provides no assurance on information flow, and dissemination is not controlled. Nevertheless, many recent Internet information systems use this model or a variation of it, because access control lists (ACLs) are intuitive to understand (e.g., WebDAV [21], LDAPv3 [22]) and work well.

There are semantic differences between securing document objects or Web services that are dynamic and interactive. For instance, the document content of a specific document version is typically static. In contrast, the con-

tent of a Web service may depend on input parameters and is therefore highly dynamic. Furthermore, the notion of authorship for content does not necessarily apply to Web services. Therefore the protection of documents deals more with intellectual and copyright issues, as well as preventing access to particular information, whereas the protection of Web service objects focuses on computational resources and transactions. It needs to be further investigated how these semantic differences affect the usage of the discretionary model.

B. The mandatory access control model

The mandatory model's main goal besides enforcing integrity is to keep secrets. It is based on classification of subjects and objects. A security level is assigned to each subject (clearance) and each object (classification). These levels build a partial ordered set (lattice of security levels). For instance, in military environments this could be *top secret*, *secret*, *confidential*, and *unclassified*. The concept of *read down* assures that a person can only read classified information, which is lower than or equal to her/his clearance. Similarly, a subject is prevented from writing information into an object with a lower security level than the subject's clearance (*write up*). This model works well in rigid environments (e.g., military), since it assumes the ownership of information by a central authority. It suffers from the inability to properly represent the variety and nature of social roles of users in the lattice structure of security labels. Furthermore, it may lead to poly-instantiation (simultaneous existence of multiple information instances referring to the same real world concept, but differing by their classification level and by their contents). Due to its rigidity it seems that this model is not well suited for a Web service security model.

C. The role-based access control model

Many practical requirements (e.g., administration of authorizations in dynamic user environments) are not covered well enough in the discretionary and mandatory models, which motivates to further investigate alternative models. The role-based access control model (RBAC) [23] addresses many of these issues, and there are proposals on how RBAC can be integrated into the WWW environment [24], [25], [26].

In the role-based model we first need to identify roles in a system. Subjects are then mapped to these roles. Users can exercise several roles at the same time or they can be forced to assume only one role. Overall, roles are based on activities the user executes in a system. Roles can form a hierarchy of roles, and inheritance can be used to propagate properties of roles. The role-based access control has gained popularity in the commercial world, since it simplifies security management (e.g., if a subject's responsibility changes, we simply assign a new role to it). The model works well in dynamic environments. It looks promising to incorporate parts of this model into a general security model for Web services.

One of the challenges is the mapping of authentication

credentials to roles. There are many different scenarios where this mapping occurs. For instance, in a centralized approach a trusted third party (e.g., certification authority) could maintain a role database and subject mappings to those roles. This certification authority (CA) would then issue a signed certificate that states that a subject indeed assumes a particular role.

In a more decentralized approach a trusted subject would make statements about a role another subject assumes. For example, A knows B , and certifies that B assumes the role r at time t ($A \perp_r^t B$). C has a trust relationship to A . When B makes a request to C , and provides the certificate of $A \perp_r^t B$, C may grant the request. Another approach is that a Web service provider has to maintain its own user and role database, which introduces an administrative burden of keeping subject-role mappings up to date.

D. Capability based access control

In this access control model a subject is associated with a list that describes the objects for which a subject is authorized. This list is called the *capability list*. Sandhu et al. describe how this model relates to access control lists [16]. The possession of a capability allows access to resources, similar to a key that unlocks a secured door. Once a capability is distributed, it is difficult to revoke it. Analogously, if someone loses a key for a front door, the lock has to be replaced to prevent misuse by someone who finds the key.

Although the model did not prove to be commercially successful, it has many advantages in a distributed environment like the Web. For instance, it facilitates single sign-on for Web services, since repeated authentication for a subject is not required. Once a Web service is authenticated and has obtained a capability, it can present this capability to other Web services to gain access to particular resources. Furthermore, capabilities can be combined with ACLs to provide finer-grained control to resources.

Overall the model seems to have desirable properties that can be exploited for Web services (e.g., single-sign on). Kahan [27] presents a capability-based authorization model for the Web. It needs to be examined in more detail on how this model can be further extended and integrated into the Web services architecture.

E. Chains of trust

Chains of trust have several desirable properties (e.g., low administrative efforts) [28] that makes them suitable for authorizations of Web services. The basic idea is if a subject A trusts B on subject t ($A \perp_t B$), and B trusts C on subject t ($B \perp_t C$), then A may decide to also trust C on subject t based on transitivity of trust ($A \perp_t C$). However, if $B \perp_t C$, then $A \perp_t C$, only if C is providing some evidence that shows that indeed $B \perp_t C$.

The term “trust” is very broad, and it is a good practice to narrow the semantics of trust to be very specific. What is it exactly that A trusts B on? For instance, a Web service A does not trust B to keep confidential information secret, but A trusts B to be good for transactions less than \$10. In this case A would not have to run a credit check on B .

The trust relationship t here is “be good for a transaction up to \$x”.

The main advantage of this model is that it reduces administration of authorizations considerably and therefore fits nicely in the decentralized Web services model. However, there’s no standardized trust specification language. Research and ideas in this area are just emerging (e.g., SAML [29]). It needs to be analyzed on how trust relationships can be used to authorize access to Web services. An access control design for Web services should be flexible enough to allow an integration of such a model at a later time.

F. Other access control models

More security models have been proposed [17], for instance, the Clark and Wilson model and the personal knowledge approach. Both seem to be difficult to apply to Web services.

A reasonable security model for Web services might be a combination of the presented models, possibly with some additions based on further research: The discretionary model combined with the role-based model with support for chains of trust. Or, a capability based one using chains of trust to share authorizations across different domains. The discussion and prior work can be seen as a starting point that may help to define that most appropriate security model.

III. RESEARCH ISSUES IN DESIGNING AN ACCESS CONTROL MODEL FOR WEB SERVICES

This section investigates what are possible subjects, objects, and actions in the Web services architecture, and explores research issues related to the specification of these.

A. Specification of subjects

Subjects or principals are entities [16] that initiate an activity in form of actions or operations on objects. Typically subjects or principals are users, organizations, or programs acting on a user’s behalf. In some cases subjects can be objects themselves. This is based on the observation that the initiator of an operation can be the target of another.

The existing proposals for securing Web services [30] or related work in the area of WebDAV [21], [31] are mostly based on some form of the discretionary model, which requires the specification of subjects for each authorization. What are subjects in the Web services context? How can subjects be grouped to facilitate administration? What are flexible notations to group subjects? There are many things to consider, and we do not even look at the problem of proper authentication, which addresses the question whether a subject is indeed the subject it claims to be.

A.1 Requirements for different levels of authentication

There are many examples when we need to think about how subjects can be specified and what level of authentication is needed for a transaction. Groups and roles are a way to specify many subjects efficiently. There are cases when a Web service provider may want to restrict access

to a Web service object to some specific retailers (based on identity). Or, access should be limited to customers connecting from a trusted domain (based on originating network location). Sometimes a subject wants to remain anonymous and doesn't want to reveal its identity or current network location. In general, there might be a conflict between the authentication level a Web service provider requires, and the authentication level a Web service client wants to reveal for a particular transaction. If the two parties cannot agree there is no transaction. If money is involved in a transaction it will be important for the both parties to have an agreement.

The level of authentication introduces also privacy issues. Once a subject's identity is revealed, a Web service provider could use this information to track activities and compile a profile. This problem becomes severe in a world where software agents represent users and programmatically access a large amount of Web services simultaneously to perform specific tasks.

A.2 Management and administration of groups and roles

How can group or role memberships be efficiently managed and administered? Web service providers want to keep administrative efforts low, thus administrative issues are important. Using roles or groups to specify authorizations will help to achieve this. Still, a Web service provider might not want to manage the subject/group relationships and may choose to rely on assertions about role or group memberships from a different entity to avoid managing a user database. A subject might be able to provide credentials that shows evidence (e.g., a signed certificate from a trusted third party) that it belongs to a particular group (e.g., a non-profit organization).

Users who pay for computation could be allowed to gain access to a Web service method with different functionality compared to a non-paying user (different types of access based on role of subject). This relates to the group administration issue, but involves accounting.

A.3 Choice of subject identification

There are more problems in the specifications of subjects. First, what user id should be used for a particular user? Damiani et al. [20] identify users by a user id, an IP address, or a symbolic network address. These can be combined and complemented with wildcards. Typically a user uses many virtual identities. To gain access to a network or Web site each requires some form of subject identification and authentication. This results in a huge amount of user ids and passwords a typical Internet user has to manage. If such a user is running a Web service client that interacts with other Web service objects, what is the identity for that Web service client? We can easily see that IP addresses are somewhat useless. The reason for this is that in the current Internet infrastructure with firewalls and network address translation (NAT), thousands of different Web services sharing the same firewall could have the same IP address to the outside. Similarly, network names are not too useful for the same reason. Those might be useful on a

lower level to restrict some network traffic. However, even at this level this is not a reasonable solution. Blocking an IP address from a large company, because some machine causes network traffic problems could result in denial of availability to possibly thousands of other users. Bertino et al. [19] are proposing to use the user id of the host, where the user is currently connected to. In the previous scenario this would be the local user id at the user's home machine, which is meaningless in a global environment.

A.4 Naming schemes for subject identification

There's is a need for a naming scheme that can be used globally to identify subjects (e.g., using URNs). Other possible solutions include the usage of a central authority on identities (e.g., MS Passport [32]). In this case MS Passport takes care of proper user authentication, and a valid global user id. Alternatively, each site uses its own registration database. This resembles the current practice of having a user id per site, which is generally used on the Web and puts a the burden of managing a fast growing amount of user ids and passwords to the user. In addition, all the sites that are maintaining user ids and passwords need to have a secure infrastructure to manage these. It can be seen that this decentralized approach introduces many interoperability problems. However, the case of misuse of a single site user id might not be as severe as the misuse of a centralized passport id. For instance, an attacker that gains access to a passport id can use that id to impersonate the user of that id to a variety of sites that accept the passport. Whereas the damage of a single site user id might be limited.

B. Specification of objects

According to Damiani [30] an object characterizes an entity on which access is being requested and consequently authorizations define whether access to the entity should be granted or not. We can represent Web services using a hierarchical containment model that comprises Web service methods at the lowest level, Web service objects, and Web service collections at the top. A Web service method represents an operation that can be invoked on a Web service object (similar to object oriented programming languages). Related Web service objects can be grouped into a Web service collection. Each object in this Web service model is a resource (e.g., Web service collection, Web service object, Web service method). Therefore the resource is the entity we need to protect from unauthorized access.

B.1 Fine-grained access control for Web services

Besides the obvious security objects such as Web service objects and methods, there are many not so obvious ones. As an example, a service provider may want to limit access to a range of values for a specific parameter of a particular method, i.e. Alice is allowed to invoke `foo()` with all positive integer values, whereas Bob is only allowed to invoke `foo()` with integers less than a value of 10. It might be of interest for complex functions to limit computational time for particular subjects. There are more scenarios where

access control can be useful to limit the set of possible values to be passed as arguments to methods. Therefore it might be required to have a more fine-grained level of access control to specify authorizations on method parameters, or even on value ranges for a specific parameter. The specification of objects can be extended to properties of resources. Metadata that describes these properties in a standardized way (e.g., RDF schemas [33]) may be used to specify authorization objects. Damiani et al. [34] proposed to use SOAP messages as authorization objects to provide fine-grained access control for Web services. One of the problems with this approach is that it changes SOAP requests, which then may not reflect the original intention of the sender.

B.2 Access control for associated data of Web services

The Web services architecture allows interaction of participating components through a well defined interface. However, objects may have an internal state (e.g., in form of program variables), which may change upon invocation of some method. Furthermore, data that resides in a database system, a file system or elsewhere may have changed after an invocation of a Web service method. Therefore the internal data that is associated with a Web service object should also represent an object in the access control matrix. Conceptually we can treat internal program variables and externally associated data as an internal state that belongs to a Web service object. Some clients may be allowed to have full access to all data members, others may only be allowed to access a subset of the data, others may have read-only access. What makes it difficult is that some of this data typically is managed by a different system, that doesn't necessarily allow the exchange of authorizations easily. The SAML [29] approach is a step in this direction towards an exchange of authorization information.

C. Other considerations for access control

There might be other plausible scenarios for access control. We could envision a form of provisional access control as proposed by Kudo and Hada [35]. In this case a user has to satisfy a particular condition before access to an object is authorized (or after). A condition in this case is associated with a (subject, object) pair.

For instance, a policy could enforce that the user has to digitally sign the request, and that the access has to be logged. Taking this a step further, a Web service may require proof of some properties of a subject. As an example, we assume that the subject is authenticated and agrees that the access will be logged, and the message is encrypted and digitally signed. However, a Web service provider may need some form of proof that the subject is over 18 years old. We could require that the subject shows proof that it assumes the role of "persons over 18 years old".

The subject may not be able to provide a proof for various reasons. In this case the Web service provider could consult a subject-role verification service before access is granted. Such a service would given a subject id, and a

desired role or property, confirm or deny that the subject assumes that role or has that property. For example, the government as a trusted authority already has access to personal information. Therefore it would be technically feasible to provide such a subject-role verification service. However, this raises privacy issues and concerns. If everyone would be able to retrieve certified personal profile information of subjects there is a high probability of misuse that can be done with that data (e.g., a marketing company uses the information to send out spam email).

IV. RELATED WORK

Sandhu and Samarti [16] survey access control principles and models, and provide a solid background on the topic of access control. In addition, Sandhu et al. [23] present an overview of the role-based access control model. Oppliger [36] reviews security mechanisms on the Internet and confirms the lack of security on the Web. In addition, Oppliger provides a comprehensive overview of methods for securing applications on top of the Hypertext Transfer Protocol (HTTP). The original HTTP/1.0 [37] specification provided a simple password-based basic authentication. Web servers are using this to provide basic access control (e.g., to protect files or directories). Authentication information is sent using Base64 encoding. Nothing is being done to prevent passive eavesdropping to collect username and passwords. Therefore this authentication scheme is considered weak similar to other TCP/IP applications, such as TELNET or FTP).

The HTTP/1.1 [18] specification introduced an improved authentication scheme called "*Digest Authentication*", which uses a more elaborate security protocol that no longer transmits passwords in clear. However, this authentication scheme was not widely adopted by commercial browsers and is still considered weak compared to other technologies (e.g., SSL).

To secure communication on the Internet in general, SSL (Secure Sockets Layer) was invented. Its primary goal was to provide a secure communication channel and to authenticate a Web server (optionally the client). SSL operates between any two applications that do not necessarily need to be on the same (secure) network. After 1994, SSL was widely adopted in commercial browsers. There were many problems with earlier versions of SSL [8]. SSL itself is quite complicated, and comprises many technical details (e.g., versioning, firewall traversal, error handling). It also requires public key certificates. This introduced new problems on how to manage these certificates securely, which helped certification authorities (CA) and public key infrastructures (PKI) to gain momentum.

IPSec was introduced [38] to create a secure network of computers over insecure channels by providing security for low-level network packets. The main difference between SSL and *IPSec* is that SSL secures two applications, whereas *IPSec* secures an entire network.

In distributed authoring scenarios Web resources may be accessible by multiple principals (e.g., WebDAV [21], [31]). To control how these principals can access and alter a re-

source, a system of access control is needed. It needs to be defined what actions a particular principal is allowed to exercise on a particular resource. Access control in WebDAV is based on the discretionary model. It supports different notions of principals: user, client software, servers, groups. HTTP scheme URLs are used to identify principals and HTTP digest is used for authentication. No fine grained access control is supported (only on resource level, not on properties). Furthermore, no role based security model is supported yet (a role is dynamically defined collection of principals).

Efforts that begin to solve problems regarding authorization over cross-organizational boundaries are just in its beginnings. The *Extensible Access Control Markup Language (XACML)* [39] and *Security Assertions Markup Language (SAML)* [29] are two of these visible efforts. SAML represents an XML-based security standard for exchanging authentication and authorization information, whereas XACML is an XML specification for expressing policies for information access over the Internet. It can be argued whether XACML and SAML are robust enough or may overlap, and cause interoperability or integration problems. AuthXML [40] and Security Services Markup Language (S2ML) [41] were two approaches to the problem of adding authentication features to XML. Both of these are subsumed into SAML. At present however there's no comprehensive and integrated design of an access control or authorization architecture that addresses many problems that arise in a world of network services on the Web.

There are two major directions to where efforts are leading towards a security model for Web services. The first one is trying to leverage from existing security solutions in the area of Internet technologies and information systems. An overview on how this can be accomplished has been written by Kirtland [42]. The author proposes to use authentication features found in the HTTP protocol, Web servers, and in operating systems. However, the problem with this approach is that it might only work together with system level security in a corporate homogeneous Intranet scenario. This is in contradiction with the notion of Web services, that are supposed to foster collaboration across network boundaries in heterogeneous and distributed environments.

The second direction towards a security model for Web service is to build a new infrastructure and framework that works in distributed and heterogeneous environments based on new emerging standards (e.g., XML, RDF) and open Internet technologies (e.g., HTTP, SOAP). For instance, one of the motivating factors behind SAML is to enable interoperability between different systems that provide security services. Traditionally, security has been implemented within a single organization. As a goal it would be desirable to have transactions initiated at one site to be completed at a different site. This requires security information to be shared among the various Web sites involved in a transaction. The overall direction where SAML is heading may be promising, and it needs to be investigated further on how this can be integrated in a security model

for Web services.

Building new technologies based on open standards towards a security framework for Web services seems to have many benefits compared to trying to patch existing infrastructure. A combination of existing, emerging, and new technologies might be a reasonable choice: For instance, leveraging from existing HTTP protocol mechanisms and its extensibility, the usage of emerging open standard (e.g., XML, RDF Schemas), and the design and development of new protocols and specifications that address shortcomings of the previous two.

V. CONCLUSION

The paper concludes that there are many open research issues related to authorizations for Web services: First, what is an preferred security model for access control for Web services that addresses appropriately the research issues described in the previous sections? Second, the management of subjects, groups, and roles poses many challenges. How can these be managed efficiently? Third, trust management including chains of trust promise to solve many administrative problems and promote a decentralized approach for access control. How do we manage trust, so that it can be used efficiently and reliably as a basis for an access control model?

Furthermore, privacy issues are gaining more importance. A subject should be able to decide about the dissemination and access to its personal data. What implications does this have to Web services, and how can an access control model for Web services incorporate privacy friendly policies?

Finally, if we treat Web services as a sub set of the Web, an access control model should be general, comprehensive, and extensible enough to also include the world of hypermedia and hypertext. There exists a plethora of proprietary access control mechanisms, systems, designs, and proposals (some of them are mentioned in this paper). Still, many recent Internet information systems (e.g., LDAPv3 [22], WebDAV [21], [31]) are incorporating their own access control design, instead of leveraging from an existing access control model and framework. More research in these areas may lead gradually to a more integrated solution for access control on the Web.

The paper showed that security presents the foundation for the wide spread deployment of Web services. The discretionary model combined with the role-based model can be used as a plausible starting point. Also, using capabilities and integrating trust relationships, plus exchanging trust assertions seems to be important to facilitate interoperability and reduce administrative efforts in a decentralized environment.

Furthermore, the paper reviewed related work in the area of security and access control for Web services and XML, and pointed out research issues for a suitable security model and for the specification of authorization subjects and objects.

The paper concludes that no integrated access control solution is available yet and further research in the area of

access control and security for Web services to develop new security infrastructure based on open Internet standards and protocols represents a critical requirement for a broad acceptance of the technology.

VI. ACKNOWLEDGEMENTS

I am grateful to Jim Whitehead for his valuable comments and suggestions.

REFERENCES

- [1] John Fontana, "Top Web services worry: Security," Network World, <http://www.nwfusion.com/news/2002/0121webservices.html?docid=7747>, January 2002.
- [2] Martin Gudgin, Marc Hadley, Jean-Jacques Moreau, and Henrik Frystyk Nielsen, "SOAP Version 1.2 Part 1: Messaging Framework," <http://www.w3.org/TR/soap12-part1/>, December 2001.
- [3] Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana, "Web Services Description Language (WSDL) 1.1," <http://www.w3.org/TR/wsdl>, last accessed: 3/5/2002.
- [4] Frank Leymann, "Web Services Flow Language (WSFL 1.0)," <http://www-4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf>, May 2001.
- [5] UDDI, "UDDI homepage," <http://www.uddi.org/>, last accessed: 3/5/2002.
- [6] Tarak Modi, "WSIL: Do we need another Web services specification?," <http://www.webservicesarchitect.com/content/articles/modi01.asp>, January 2002.
- [7] Warwick Ford, Phillip Hallam-Baker, Barbara Fox, Blair Dillaway, Brian LaMacchia, Jeremy Epstein, and Joe Lapp, "XML Key Management Specification (XKMS)," <http://www.w3.org/TR/xkms/>, March 2001.
- [8] D. Wagner and B. Schneier, "Analysis of the SSL 3.0 protocol," in *Proceeding of 2nd USENIX Workshop on Electronic Commerce*, November 1996.
- [9] W3C, "XML encryption WG," <http://www.w3c.org/Encryption/2001/>.
- [10] XML Signature WG, "XML Digital Signatures," <http://www.w3.org/Signature/>, last accessed: 4/15/2002.
- [11] Allen Brown, Barbara Fox, Satoshi Hada, Brian LaMacchia, and Hiroshi Maruyama, "SOAP Security Extensions: Digital Signature," <http://www.w3.org/TR/SOAP-dsig>, last accessed: 3/5/2002.
- [12] Organization for the Advancement of Structured Information Standards (OASIS), "OASIS homepage," <http://www.oasis-open.org/>.
- [13] Alex Ceponkus, Peter Furniss, and Alastair Green, "Business Transaction Protocol," <http://www.oasis-open.org/committees/business-transactions/draft.0.9.pdf>, October 2001.
- [14] Microsoft Developer Network (MSDN), "An Introduction to GXA: Global XML Web Services Architecture," <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dngxa/html/gloxmlws500.asp>, February 2002.
- [15] Bob Atkinson et al., "Web Services Security Language (WS-Security)," <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/ws-security.asp>, January 2002.
- [16] R. Sandhu and P. Samarati, "Access control: Principles and practice," *IEEE Communications*, pp. 40–48, September 1994.
- [17] D. Merkl and G. Pernul, "Security for next generation of hypertext systems," *Hypermedia*, vol. 6, no. 1, pp. 1–18, 1994.
- [18] R. Fielding, J. Gettys, J. Mogul, and H. Frystyk, "Hypertext transfer protocol - HTTP/1.1," *Network Writing Group, Request for Comments*, no. 2068, January 1997.
- [19] E. Bertino and P. Samarati, "Research issues in authorization models for hypertext systems," in *ACM SIGSAC New Security Paradigms Workshop*, La Jolla, CA, August 1995, pp. 22–27.
- [20] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati, "Design and implementation of an access control processor for XML documents," *Computer Networks*, vol. 33, no. 1–6, pp. 59–75, June 2000.
- [21] Lisa Lipper, "WebDAV Access control goals (Internet Draft)," <http://www.webdav.org/acl/goals/draft-ietf-webdav-acl-reqts-00.txt>.
- [22] E. Stokes, B. Blakley, R. Byrne, R. Huber, and D. Rinkevich, "Access control model for LDAPv3," <http://www.ietf.org/internet-drafts/draft-ietf-ldapext-acl-model-08.txt>.
- [23] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman, "Role-based access control models," *IEEE Computer*, vol. 20, no. 2, pp. 38–47, 1996.
- [24] Gustaf Neumann and Stefan Nusser, "A framework and prototyping environment for a W3 security architecture," *Proceedings of Communications and Multimedia Security, Joint Working Conference IFIP TC-6 and TC-11*, September 1997.
- [25] D. Ferraiolo S. Gavrilla J. Barkley, A. Cincotta and D. Kuhn, "Role-based access control for the World Wide Web," *20th National Computer Security Conference*, 1997.
- [26] D. Ferraiolo S. Gavrilla Lynne S. Rosenthal Mark W. Skall J. Barkley, A. Cincotta and D. Kuhn, "Role-based access control for web," *20th National Computer Security Conference*, 1998.
- [27] J. Kahan, "A capability-based authorization model for the world wide web," *Third World-Wide Web Conference*, pp. 1055–1064, 1995.
- [28] Rohit Khare and Adam Rifkin, "Weaving a Web of Trust," *World Wide Web Journal*, vol. 2, no. 3, pp. 77–112, summer 1997.
- [29] SAML, "Security Assertion Markup Language (SAML)," <http://www.oasis-open.org/committees/security/docs/draft-ssstc-saml-01.pdf>, last accessed: April 2002.
- [30] Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati, "Fine grained access control for soap e-services," *WWW10*, May 2001.
- [31] Geoffrey Clemm, Anne Hopkins, Eric Sedlar, and Jim Whitehead, "WebDAV Access Control Protocol," <http://www.webdav.org/acl/>.
- [32] Microsoft Passport, "Microsoft Passport homepage," <http://passport.microsoft.com>, last accessed: April 2002.
- [33] Dan Brickley and R.V. Guha, "Resource description framework (RDF) schema specification 1.0," <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>, last accessed: 3/7/2002.
- [34] Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati, "Fine grained access control for SOAP E-Services," in *Proceedings of WWW10*, Hong Kong, May 2001.
- [35] Michiharu Kudo and Satoshi Hada, "XML document security based on provisional authorization," *CCS'00*, 2000, IBM Tokyo Research Laboratory.
- [36] R. Oppliger, "Methods of securing applications for the world wide web (WWW)," *Computer Security Journal*, vol. 15, no. 1, pp. 1–9, Winter 1999.
- [37] T. Berners-Lee, R. Fielding, and H. Frystyk, "Hypertext transfer protocol - HTTP/1.0," *Network Writing Group, Request for Comments*, May 1996.
- [38] Naganand Doraswamy and Dan Harkins, "IPSEC: The new security standard for the Internet, intranets, and virtual private networks," Prentice Hall, 1999.
- [39] OASIS, "eXtensible Access Control Markup Language (XACML)," <http://www.oasis-open.org/committees/xacml/index.shtml>.
- [40] Robin Cover, "AuthXML Standard for Web Security," <http://xml.coverpages.org/authxml.html>, last accessed: 3/7/2002.
- [41] Robin Cover, "Security Services Markup Language (S2ML)," <http://xml.coverpages.org/s2ml.html>, last accessed: 3/7/2002.
- [42] Mary Kirtland, "Authentication and Authorization," <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dn.voices-webservice/html/service02282001.asp>.