# Towards secure XML document with usage control

Jinli Cao[1], Lili Sun[2], and Hua Wang[2]

[1] Department of Computer Science & Computer Engineering
La Trobe University, Melbourne, VIC 3086, Australia
jinli@cs.latrobe.edu.au

[2] Department of Maths & Computing, University of Southern Queensland
Toowoomba QLD 4350 Australia
(sun, wang)@usq.edu.au

**Abstract.** XML promoted by the World Wide Web Consortium (W3C) is a de facto standard language for document representation and exchange on the Internet. XML documents may contain private information that cannot be shared by all user communities. Several approaches are designed to protect information in a website. However, these approaches typically are used at file system level, rather than for the data in XML documents that have to be protected from unauthorized access. Usage control has been considered as the next generation access control model with distinguishing properties of decision continuity.

In this paper, we present a usage control model to protect information distributed on the web, which allows the access restrictions directly on structures and documents. The model not only supports complex constraints for XML components, such as elements, attributes and datatypes but also provides a mechanism to build rich reuse relationships between models and documents. Finally, comparisons with related works are analysed.

## 1 Introduction

XML Web Service is a platform-independent Web application that accepts requests from different systems on the Internet and can involve a range of web technologies such as XML [7], SOAP [6], WSDL[8] and HTTP[14]. XML is used to store and exchange data in the Internet that may include private message of customers. For example, the following XML document displays customer's information.

```
<?xml version=“1.0” encoding=“UTF-8”? >
  <customerInfo xmlns=“http://www.hotel.com/CustomerInfo” gender=”Male”>
    <ssn>123-45-6789</ssn >
    <name>
      <firstName> Tony </firstName >
      <lastName> Wang ></lastName >
    </name >
    <creditCardInfo >
        <type >Master card </type >
        <cardNo >8888888888888888 </cardNo >
        <expireDate >12/05 </expireDate >
        <nameOnCard > Tony Wang </nameOnCard >
    </creditCardInfo >
</customerInfo >
```

Table 1: XML Document Example

XML document not only shows the contents of data but also the constraints and relationships between data. In Table 1, the element *customerInfo* includes *ssn, name* and *creditCardInfo* sub-elements. The sub-element *ssn* is a simple type while sub-elements *name* and *creditCardInfo* are combined with their own sub-elements. An XML document may be generated from various resources with varying security requirements due to its ability to express complex relationship between data. Alternatively, a user may like to limit access to particular parts of an XML document. In the example above, Tony objects that everyone can read all information on his Mastercard. Another example may happen in an University, when an XML document can consist of information from applications among several faculties and multiple databases. When an internal or external user accesses this document, his/her access permission has to be monitored according to security policies in all these faculties and databases. These examples show that secure XML document is an significant topic.

Several approaches are designed for the security of XML document [12, 13, 22]. But all those approaches have some limitations. Encryption and decryption skills [12] are used in protection of communications between servers and clients rather than dissemination from clients, since these skills focus on the protection of file level but not on a systematic level. Additionally they only manage access requirements from a server side. Both Secure Sockets Layer (SSL) [13] and firewall [22], the recent techniques used on Intranet assume that the computers know XML web services, and firewall can accept only connections coming from those computers which IP addresses are known already. However, IP addresses of users are unknown to services before they connect on the Internet. SSL is applied to encrypt and decrypt messages exchanged between clients and servers. It can protect messages from unauthorized reading while message are in transition and also verifies that incoming messages actually come from the correct sender. However, SSL is not satisfactory in Web service environment, specially in association with XML message used in SOAP - based communications [6]. This is because the SOAP is a specification for performing methods request and was conceived as a message format not bound to any single protocol [10].

There are several security issues that need to be addressed in the application of XML documents. In particular, the following two problems are critical to developing a secure and flexible access control architecture.

*Problem 1.* Restricting access to XML documents to authorized users;

*Problem 2.* Protecting XML documents from malicious dissemination.

To address these problems, this paper presents authorization models which adopt usage control to manage access to XML documents and secure architectures to protect against malicious dissemination. Traditional access control has analyzed authorization decisions on a subject's access to target resources. "Obligations" are requirements that have to be followed by the subject for allowing access resources. "Conditions" are subject and object independent requirements that have to be passed. In today highly dynamic, distributed environment, obligations and conditions of new hosts are decision factors for the management of XML documents.

The remainder of this paper is organized as follows: Section 2 presents the background of XML and usage control. Three decision factors *Authorization, Obligation, Conditions* and Continuity properties *pre* and *ongoing* are introduced in this section. Section 3 shows our proposed authorization models for usage control. It includes *pre-Authorizations, ongoing-Authorizations, pre-Obligations, ongoing-Obligations*, *pre-Conditions* and *ongoing-Conditions.* Section 4 discusses how to build secure architectures for XML documents by using reference monitors in details. Section 5 compares our work with the previous work on XML document security. The difference between this work from others is presented. Section 6 concludes the paper and outlines our future work.

## 2 Related technologies

### 2.1 XML

XML [7] is a markup language for describing semi-structured information. The XML document is composed of nested elements, each delimited by a pair of start and end tags (e.g. <name> and </name>) or by an empty tag. XML document can be classified into two categories: well-formed and valid. Well-formalization requires XML document to follow some syntax, such as, there is exactly one element that completely contains all other elements, elements may nest but not overlap, etc. Validation requires XML instance to contain specified elements and attributes, following specified datatypes and relationships.

### 2.2 Usage control

There are eight components: subjects, subject attributes, objects, object attributes, rights, authorizations, obligations, and conditions in usage control model [20] (see Figure 1). Subjects and objects are familiar concepts from the past thirty years of access control, and are used in their familiar sense in this paper. A right represents access of a subject to an object, such as read or write. The existence of the right is determined when the access is attempted by the subject. The usage decision functions indicated in Figure 1 make this determination based on subject attributes, object attributes, authorizations, obligations and conditions at the time of usage requests.

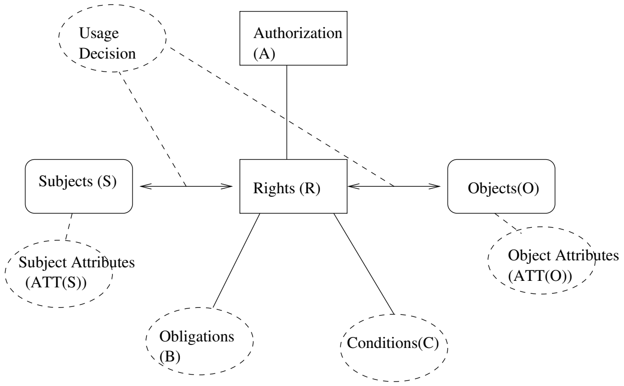

**Fig. 1.** Components of Model

Subject and object attributes can be used during the access decision process. Examples of subject attributes are identities, group names, roles, memberships, credits, etc. Examples of object attributes are security labels, ownerships, classes, access control lists, etc. In an on-line shop a price could be an object attribute, for instance, the book Harry Potter is priced at $20 for a read right and priced at $1000 price for a resell right.

Authorizations, obligations and conditions are decision factors used by decision functions to determine whether a subject should be allowed to access an object. Authorizations are based on subject and object attributes and the specific right. Authorization is usually required prior to the access, but in addition it is possible to require ongoing authorization during the access, e.g., a certificate revocation list (CRL) may be periodically checked while the access is in progress. An access is immediately revoked if the relevant certificate appears on the CRL. Authorizations may require updates on subject and object attributes. These updates can be either ‘pre’, ‘ongoing’, or ‘post’ that are called continuity properties shown in Figure 2.

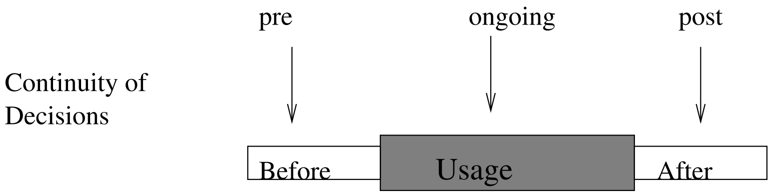

**Fig. 2.** Continuity Properties

Conditions are decision factors that depend on environmental and system-oriented requirements. For example, IEEE member can access full papers in the

IEEE digital library. They can also include the security status of the system, such as low level, normal, high alert, etc.

As discussed above, continuity is another decision factor as shown in Figure 2. In traditional access control, authorization is assumed to be done before access is allowed (pre). However, it is quite reasonable to extend this for continuous enforcement by evaluating usage requirements throughout usages (ongoing).

## 3 Authorization models

We now discuss authorization models for XML documents adopting usage control in this section. Based on three decision factors: authorizations, obligations, and conditions, we develope a family of core models for usage control. By core models, we mean that they focus on the enforcement process and do not include administrative issues. We assume there exists a usage request on an XML target object. Decision-making can be done either before (pre) or during (ongoing) exercise of the requested right. Decision-making after the usage has no influence on the decision of current usage. Based on these criteria, we have 6 possible cases spaces as a core model for usage control: pre-Authorizations, ongoing-Authorizations, pre-Obligations, ongoing-Obligations, pre-Conditions and ongoing-Conditions. Depending on the access requirements on XML documents in real world, it is possible to utilize more than one case.

For simplicity we consider only the pure cases consisting of *Authorizations, Obligations* or *Conditions* alone with *pre* or *ongoing* decisions only. We focus on developing comprehensive usage control models for XML documents. Next we present usage control models (UCM) with different pure cases.

**A.** $UCM_{preA}$**:pre-Authorizations Model**
In an $UCM_{preA}$ model, the decision process is performed before access is allowed. The $UCM_{preA}$ model has the following components:

1. $S, XD, XDL, R, R_1, ATT(S), ATT(XD), ATT(XDL)$ and usage decision Boolean functions $preA, preA_1$ on $XD, XDL$, respectively,
   where $S, XD, XDL, R, R_1$ represent Subject, XML document, element in XML document and Rights required on XML document and on element (e.g. read, write) respectively. $ATT(S), ATT(XD), ATT(XDL)$ represent attributes of subjects, XML documents and elements in XML document respectively. $preA$ and $preA_1$ are predicates about authorization functions. Fox example, when users log in IEEE website, $preA$, or $preA_1$ is a function on users' account and password (subject attributes), IEEE XML documents and rights (read, write or resell) that is used to check whether users can access the documents with the right or not,
2. $allowed(s, xd, r) \Rightarrow preA(ATT(s), ATT(xd), r)$,
   Where $A \Rightarrow B$ means $B$ is a necessary condition for $A$. This predicate indicates that if subject $s$ is allowed to access XML document $xd$ with right $r$ then the indicated condition $preA$ must be true.

3. $allowed(s, xdl, r_1) \Rightarrow preA_1(ATT(s), ATT(xdl), r_1)$.
   The $allowed(s, xdl, r_1)$ predicate indicates that if subject $s$ is allowed to access XML document $xdl$ with right $r_1$ then the decision function $preA_1$ is true.

The $UCM_{preA}$ model provides an authorization method on whether a subject can access XML document or not. The $allowed(s, xdl, r_1)$ predicate shows that subject $s$ can access some part of information in XML document. At this stage, private information in XML document is restricted.

**B. $UCM_{onA}$:ongoing-Authorizations Model**

An $UCM_{onA}$ model is used to check ongoing authorizations during access processes. The $UCM_{onA}$ model has the following components:

1. $S, XD, XDL, R, R_1, ATT(S), ATT(XD), ATT(XDL)$ as before, and ongoing usage decision functions $onA$ on $XD$ (XML document) and $onA_1$ on $XDL$ (XML document elements),
   $onA$ and $onA_1$ are used to check whether $S$ can continue to access or not.
2. $allowed(s, xd, r) \Rightarrow true$,
   This is a prerequisite for ongoing authorization on $xd$.
3. $allowed(s, xdl, r_1) \Rightarrow true$,
   This is a prerequisite for ongoing authorization on $xdl$.
4. $stopped(s, xd, r) \Leftarrow \neg onA(ATT(s), ATT(xd), r)$,
   The access of subject $s$ to $xd$ is terminated if the ongoing authorization $onA$ is failed.
5. $stopped(s, xdl, r_1) \Leftarrow \neg onA_1(ATT(s), ATT(xdl), r_1)$.
   The access of subject $s$ to $xdl$ is terminated if the ongoing authorization $onA_1$ is failed.

$UCM_{onA}$ introduces the $onA$, $onA_1$ predicate instead of $preA$, $preA_1$. $allowed(s, xd, r)$ and $allowed(s, xdl, r_1)$ are required to be *true*, otherwise ongoing authorization should not be initiated. Ongoing authorization is active throughout the usage of the requested right, and the $onA$ and $onA_1$ predicates are repeatedly checked for continuation access. These checks are performed periodically based on time or event. The model does not specify exactly how this should be done. When attributes are changed and requirements are no longer satisfied, *stopped* procedures are performed. We use $stopped(s, xd, r)$ and $stopped(s, xdl, r_1)$ to indicate that rights $r$ and $r_1$ of subject $s$ on object XML and XML document elements are revoked and the ongoing access terminated. For example, suppose only one user can access customer Mastercard information in an object XML simultaneously. If another user requests access and passed the pre-authorization, the user with the earlier time access is terminated. While this is a case of ongoing authorizations, it is important that the certificate should be evaluated in a *pre* decision.

Based on the length of the paper, other authorization models are omitted. The following algorithm is based on these models and introduces how to manage an XML document access control when a user (subject) applies to access an XML document (target.xml) with right $r$.

**XML-based Algorithm**:
**Input**: Access request: (u, r, target.xml)
**Output**: target'.xml
**Method**:

```
// Verify UCM_preA:
if $preA(ATT(s), ATT(xd), r) \bigcup preA(ATT(s), ATT(xdl), r_1) = false$
    // The process in pre-Authorization is not successful
    ACCESS denied;
endif

// Verify UCM_onA:
if $preA(ATT(s), ATT(xd), r) \bigcup preA(ATT(s), ATT(xdl), r_1) = false$
    // The process in pre-Authorization is failed, donot need further verification.
    Application denied;
endif
$onA(ATT(s), ATT(xd), r) \bigcup onA(ATT(s), ATT(xdl), r_1) = false$
    // The process in ongoing-Authorization is not successful
    ACCESS stopped;

// Verify UCM_preB:
if $preObfill(s, xd, r) = false$
    // Obligations are not fulfilled and pre-Obligation is not passed.
    ACCESS denied;
endif

//Verify UCM_onB:
if $allowed(s, xd, r) = false$
    Stop verification.
endif
if $onObfill(s, xd, r) = false$
    // Obligations are not continually fulfilled and on-Obligation is not passed.
    ACCESS is stopped;
endif

// Verify UCM_preC:
if $preC(s, xd, r) = false$
    // Conditions are not satisfied and pre-Condition verification is not passed.
    ACCESS denied;
endif

//Verify UCM_onC:
if $allowed(s, xd, r) = false$
    Stop verification.
endif
if $onC(s, xd, r) = false$
    // Conditions are not continually satisfied and on-Condition is not passed.
```

```
ACCESS is stopped;
endif
ACCESS target.xml is permitted;
Output target'.xml;
```

Table 3: Algorithm of XML Access Control

We obtain an authorization method for XML document and its elements by checking users' (subjects') authorizations, obligations and conditions with continuity properties. The algorithm provides a solution of the problem 1. We analyse security architectures in the next section for the problem 2 in which both client and server sides is required to be monitored.

## 4 XML security architecture

In this section, we discuss architecture solutions for XML control based on reference monitors. Reference monitors have been discussed extensively in access control community. Subjects can access XML objects only through the reference monitor since it provides control mechanisms on access XML document and its elements.

### 4.1 Structure of Reference Monitor

ISO has published a standard for access control framework by using reference monitors [16]. Based on the standard, XML reference monitor consists of Usage Decision Facility (UDF) and Usage Enforcement Facility (UEF) as shown in Figure 3. Each facility includes several functional modules.

UEF includes *Customization, Monitor and Update modules* and UDF includes *authorization, conditions and obligations decision modules.* When a subject sends an access request through *Customization module* to *Authorization module*, *Authorization module* verifies authorization process and checks whether the request is allowed or not. It may return yes or no or metadata information of the authorization result. This metadata information can be used for approved access on XML objects by *Customization module* in UEF. *Condition module* is used to make a decision for whether the conditional requirements are satisfied or not. *Obligation module* is applied to verify whether obligations have been performed or not before or during the requested usage. When any obligation is changed, it must be monitored by *monitor module* and the result has to be resolved by *Update module* in UEF. Applications of these modules rely on object systems requirements.

### 4.2 Architectures

There are two kinds of reference monitors: Server-side Reference Monitor (SRM), and Client-side Reference Monitor (CRM). Servers provide XML document and

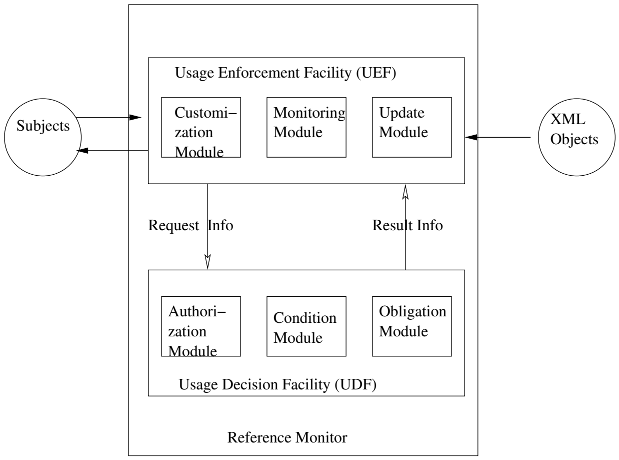

**Fig. 3.** XML Reference Monitor

clients require access to the XML object. Like a traditional reference monitor, an SRM works in server system environment and manages access to XML objects in the server. On the other hand, a CRM works in the client environment and controls access to XML objects when it works as a server for other clients. For example, the client acts as a server when the XML document is disseminated to other users. SRM and CRM can coexist within a system. For real implementations, both CRM and SRM should be used for better security. We analyse architectures according to reference monitors on server side only (SRM-only), on client side only (CRM-only) and on both server and client sides (SRM & CRM).

**SRM-Only Architecture**

A system with SRM-only facilitates works on server side only to control subjects access XML objects. In this case an XML object may or may not be stored in client-side. If the XML object is allowed to reside in client-side, it means the saved client copy of the XML object is no longer valid and doesn't have to be controlled. It can be used and changed freely at client-side. For example, an XML on-line bank statement can be saved at a client's local machine for his records and the server (bank) doesn't care how the copy will be used by the client since the bank keeps original account information safe. However if the XML document or some parts of the document has to be protected and controlled centrally, the XML must remain at server-side storage and is not allowed to be stored in client-side. This is the main topics of traditional access control and trust management system.

**CRM-Only Architecture**

No reference monitor exists on the server-side in a system with CRM-only environment. Rather, a reference monitor exists at the client system for controlling usage of disseminated XML documents. In this environment XML objects can be stored either centrally or locally. The usage of XML saved at the client-side on behalf of a server is still under the control of CRM. Distributed XML document are associated with certain usage rules and users may need to prove they have sufficient credentials to access the document.

**SRM & CRM Architecture**
With both SRM to CRM, this architecture can provide a comprehensive access control. SRM may be used for distribution related control while CRM can be used for XML document dissemination. For instance, in SRM, XML objects can be pre-customized for distribution. The pre-customized XML objects can be further controlled and customized by CRM. As a result, server can restrict or eliminate unnecessary exposure of XML objects that do not have to be distributed. If a user requests certain XML document that includes some secret information, SRM can pre-customize the requested objects before distribution such that the distributed version of the objects doesn't include any secret information. If the document cannot be disseminated, the CRM at client side can do this work.

The SRM & CRM architecture provides a solution for restricting access to XML documents and protecting XML documents from malicious dissemination.

## 5 Comparisons

Related work has been done on secure and selective dissemination of XML documents [2] and securing XML Web services [9].

Elisa and Elena [2] proposed an access control system supporting selective distribution of XML document among possible large user communities by using a range of key distribution methods. They demonstrate a formal model of access control policies for XML documents. Policies defined in the model take into account both user profiles, and document contents and structures. An approach based on cryptograph is designed to allow sending the same document to all users, and to enforce the stated access control policies. The approach consists of encrypting different portions of the same document according to different encryption keys, and selectively distributing these keys to the various users. This proposal is different from ours in two aspects. First, it focuses on key distribution methods to protect XML document. Therefore, it only discussed the management in server side and without any management about how to control the XML document when users get keys. By contrast, our work provides a rich variety of options that can deal with XML documents in both server and user sides. Second, users can access XML documents with their keys at any time, even as their properties are updated. It means there is no ongoing authorization for users. In our scheme, users have to satisfy pre-Authorizations, pre-Obligations and pre-Conditions as well as ongoing-Authorizations, ongoing-Obligations, and ongoing-Conditions.

Securing XML Web services is described by Damiani, Vimercati and Samarati in 2002 [9]. Two experiments are discussed. One is that restricting access to an XML Web service to authorized users. Another one is that protecting the integrity and confidentiality of XML messages exchanged in a Web service environment. The authors introduce SOAP highlights, how to use SOAP headers for credential transfer and access control. The main difference between our scheme and the work in [9] is that we focus on a systematic level for XML document by using usage control model and consider a solution for different kinds of authorizations, whereas the latter is a discussion of providing a secure infrastructure to XML Web services.

## 6 Conclusions and future work

This paper has discussed access models and architectures for XML documents by using usage control. We have analysed not only decision factors in usage control such as authorizations, obligations and conditions, but also the continuity. Different kinds of models are built for XML document. To protect XML documents from malicious dissemination, we have analysed reference monitors on both server and client sides and obtained several secure architecture solutions. The work in this paper has significantly extended previous work in several aspects, for example, the ongoing continuity for authorizations, obligations and conditions. These methods can be used to control XML document in a dynamic environment since they provide a robust access control for XML document and can protect sensitive messages from dissemination. It also begins a new application with usage control.

The future work includes develop algorithms based on the models and architectures proposed in this paper and application of the algorithms in real implementation.

## References

1. Arenas M. and Libkin L.: A normal form for XML documents. ACM Transaction on Database System. Vol. 29 (2004) 195–232
2. Bertino E. and Ferrari E.: Secure and selective dissemination of XML documents. ACM Transaction on Information System Security. Vol. 5 (2002) 290–331
3. Bertino E., Castano S., Ferrari E. and Mesiti M.: Specifying and enforcing access control policies for XML document sources. World Wide Web, 3. Baltzer Science Publishers BV (2000) 139–151
4. Bertino E., Castano S. and Ferrari E.: Securing XML documents: the author-X project demonstration. Proceedings of the 2001 ACM SIGMOD international conference on Management of data. Santa Barbara, California, United States (2001) 605
5. Bertino E.,Castano S., Ferrari E. and Mesiti M.: Controlled access and dissemination of XML documents. Proceedings of the second international workshop on Web information and data management. Kansas City, Missouri, United States (1999) 22–27
6. Box D.: Simple Object Access Protocol (SOAP) 1.1. World Wide Web Consortium (W3C). Cambridge, MA, USA (2000) http://www.w3.org/TR/soap

7. Bray T., Paoli J., Sperberg M. and Maler E.: Extensible Markup Language (XML) 1.1 (Second Edition). World Wide Web Consortium (W3C). Cambridge, MA, USA (2000) http://www.w3.org/TR/REC-xml
8. Chinnici R., Gudgin M., Moreau J. and Weerawarana S.: Web Services Description Language (WSDL) 1.2. World Wide Web Consortium (W3C). Cambridge, MA, USA (2002) http://www.w3.org/TR/wsdl12
9. Damiani E., Capitani S. and Samarati P.: Towards Securing XML Web Services. Proc. of the 2002 ACM Workshop on XML Security. Washington, DC, USA (2002)
10. Damiani E., Sabrina D., Paraboschi S. and Samarati P.: Fine grained access control for SOAP E-services. Proceedings of the tenth international conference on World Wide Web. Hong Kong, China (2001) 504–513
11. Damiani E., Vimercati S., Paraboschi S., and Samarati P.: Securing XML Documents. Lecture Notes in Computer Science. Vol. 1777 (2000) 121 – 135
12. Ford W. and Baum M. S.: Secure electronic commerce: Building the Infrastructure for Digital Signatures & Encryption Prentice Hall PTR (1997)
13. Freier A., Karlton P., and Kocher. P.: The SSL Protocol - Version 3.0. http://ftp.nectec.or.th/CIE/Topics/ssldraft/INDEX.HTM (1996)
14. Gettys J., Mogul J., Frystyk H., Masinter L., Leach P., and Berners-Lee T.: Hypertext Transfer Protocol - HTTP/1.1. (1999)
15
15. Goldschlag D., Reed M., and Syverson P.: Onion routing for anonymous and private Internet connections. Communications of the ACM. Vol. 24 (1999) 39 – 41
16. ISO: Security frameworks for open systems: Access control framework. ISO/IEC 10181-3 (1996)
17. Jajodia S., Samarati P., Subrahmanian V. and Bertino E.: A unified framework for enforcing multiple access control policies. Proceedings of the 1997 ACM SIGMOD international conference on Management of data. Tucson, Arizona, United States (1997) 474–485
18. Kudo M. and Hada S.: XML document security based on provisional authorization. Proceedings of the 7th ACM conference on Computer and communications security. Athens, Greece (2000) 87–96
19. Li Q. and Atluri V.: Concept-level access control for the Semantic Web. Proceedings of the 2003 ACM workshop on XML security. Fairfax, Virginia (2003) 94–103
20. Park J. and Sandhu R.: Towards usage control models: beyond traditional access control. Proceedings of the seventh ACM symposium on Access control models and technologies. Monterey, California, USA (2002) 57–64
21. Sabrina D.: An authorization model for temporal XML documents. Proceedings of the 2002 ACM symposium on Applied computing. Madrid, Spain (2002) 1088–1093.
22. Todd B.: Auditing Firewalls: A Practical Guide. http://www.itsecurity.com/papers/p5.htm (2004)
23. Wang H., Cao J. and Zhang Y.: Formal authorization allocation approaches for permission-role assignments using relational algebra operations. Proceedings of the 14th Australasian Database Conference, Adelaide, Australia (2003) 125–134
24. Wang H., Zhang Y., Cao J., Varadharajan V.: Achieving secure and flexible M-services through tickets. In Benatallah B. and Maamar Z. editor: IEEE Transactions on Systems, Man, and Cybernetics, Part A, Special issue on M-Services. Vol. 33 (2003) 697-708
25. Zhang X., Park J. and Sandhu R.: Schema based XML Security: RBAC Approach. Proceedings of the IFIP WG (2003)