Storage and Access Control Policies for XML Documents

George Pallis
Aristotle University of Thessaloniki, Greece

Konstantina Stoupa
Aristotle University of Thessaloniki, Greece

Athena Vakali
Aristotle University of Thessaloniki, Greece

INTRODUCTION

The Internet (and networks overall) are currently the core media for data and knowledge exchange. XML is currently the most popular standardization for Web document representation and is rapidly becoming a standard for data representation and exchange over the Internet. One of the main issues is XML documents and in particular, storage and accessing. Among data management issues, storage and security techniques have a particular importance, since the performance of the overall XML-based Web information system relies on them. Storage issues mainly rely on the usage of typical database management systems (DBMSs), whereas XML documents can also be stored in other storage environments (such as file systems and LDAP directories) (Amer-Yahia & Fernandez, 2002; Kanne & Moerkotte, 2000; Silberschatz, Korth & Sudarshan, 2002). Additionally, in order to guarantee the security of the XML data, which are located in a variety of the above storage topologies, the majority of implementations also provide an appropriate access control. Most storage systems cooperate with access control modules implementing various models (Joshi, Aref, Ghafoor & Spafford, 2001), whereas there are few commercial access control products available. However, there are some standardized XML-based access control languages that can be adopted by most tools.

This article focuses on presenting an overview related to both storing and securing XML documents. Such an overview will contribute to identifying the most important policies for storage and access in Web-based information systems (which extensively use the XML as the data representation format). In addition, the most well-known related implementations are presented. A more integrated survey on these issues is presented in Pallis, Stoupa, and Vakali (2004).

BACKGROUND

Storage Policies

Several solutions for storing XML data have been proposed both in the scientific literature and in commercial products. In particular, storage approaches can be classified with respect to the type of system on which they rely and on the used XML document representation model. In this framework, they can be categorized as follows:

- Relational DBMS: Uses a collection of tables to represent both data and relationships among these data. More specifically, in order to represent XML data by using tables, it is necessary to break down the XML documents into rows and columns. The tree-like structure of XML facilitates both their decomposition and storage in relational tables. However, this process is expected to cause some performance overhead mainly due to the continuous translation of trees to tables (and vice versa). Due to its popularity, several models have been proposed to store XML documents in relational DBMSs (e.g., Bohannon, Freire, Roy & Simeon, 2002; Khan & Rao, 2001; Schmidt, Kersten, Windhouwer & Waas, 2000; Tian, DeWitt, Chen & Zhang, 2000; Zhu & Lu, 2001).
- Object-Oriented (O-O) DBMS: XML documents are stored as collections of object instances, using relationships based on the O-O idea (McHugh, Abiteboul, Goldman, Quass & Widom, 1997). O-O DBMSs have been designed to work well with object programming languages such as C++, C#, and Java. Inheritance and object-identity are their basic characteristics. However, O-O DBMSs cannot easily handle data with a dynamic structure since a new

class definition for a new XML document is needed, and the use of O-O DBMSs for XML document storage is not as efficient and flexible.

- Object-Relational (O-R) DBMS: The XML documents are stored in a nested table, in which each tag name in DTD (or XML schema) corresponds to an attribute name in the nested table. Currently, researchers have showed a steadily increasing interest in O-R DBMSs, since they combine both the benefit of the relational maturity and the richness of O-O modeling (Pardede, Rahayu & Taniar, 2004). In O-R DBMSs, the procedure for storing XML data to relation mapping is modeled by an O-R model. In this context, a number of transformation steps from the XML schema to an O-R model are presented (Widjaya, Taniar & Rahayu, 2004). More specifically, each nested XML element is mapped into an object reference of the appropriate type. Then, several mapping rules are indirectly embedded in the underlying model.
- Native XML DBMS: In this case, the XML document is the fundamental unit of storage (Kanne & Moerkotte, 2000). Therefore, a native XML database defines a (logical) model for an XML document, and stores and retrieves documents according to that model. In particular, two basic steps are involved for storing the XML documents on a native XML DBMS: (1) the data are described by its structure (DTD or XML schema), and (2) a native database XML schema (or a data map) is defined. However, the native XML DBMSs have not yet become very popular, since these systems must be built from scratch.
- LDAP Directories: XML documents are stored in LDAP directories which can be considered as a specialized database (Marron & Lausen, 2001). Therefore, the internal storage model of this database system is defined in terms of LDAP classes and attributes. More details about the architecture of the LDAP model and protocol are discussed by Howes, Smith, and Good (1999). Comparing the LDAP directories with the typical DBMSs, they are more widely distributed, more easily extended, and replicated on a higher scale.
- File Systems: Since an XML document is a file, a typical storage approach is to store it simply as a flat file. In particular, this approach uses a typical file-processing system, supported by a conventional operating system (as a basis for database applications). The wide availability of XML tools for data files results in a relatively easy accessing and querying of XML data (which are stored in files). By using a flat file for XML data, we have faster storing (or retrieving) of whole documents. However, this storage format has many disadvantages, such as difficulty in accessing and updating data, since the only way to accomplish this is to overwrite the whole file (Silberschatz et al., 2002).

Access Control Policies

In order to protect XML files, we need to design authorizations defining which client (subject) can access which protected resource (object) and under which mode. Since XML files are organized according to DTDs or XML schemas, protected resources can be both XML files, DTDs, or schemas, or parts of them, such as a specific element or even attribute.

The basic access control models are:

- Discretionary Access Control (DAC): Every subject and object is enumerated, and there are authorizations connecting each subject and object. The owner is responsible for defining policies in order to protect his/her resources, and (s)he can also define who is going to be assigned some access privileges. It is the most flexible and simple access control model. However, it does not provide high levels of protection, and it cannot be used in case multiple security levels are required.
- Mandatory Access Control (MAC): This is based on the existence of one central administrator responsible for the definition of policies and authorizations. It is expressed with the use of security labels associated with both subjects and objects. Every object has a classification label defining its sensitivity, and each subject is assigned a clearance label defining its trustworthiness. This model is more secure than DAC, but it cannot be used in wide distributed Internet-based environments. Therefore, their usage is gradually decreased.
- Role-Based Access Control (RBAC): The most widely used access control model in modern environments (Sandhu, Coyne & Feinstein, 1996). Subjects are assigned roles that are categorizations of subjects according to their duties in an organization. Thus, every subject is assigned roles that in their part have some authorizations. Therefore, the number of the needed policies is highly decreased. RBAC model is a super set, since it can also express DAC and MAC policies. It is appropriate for distributed heterogeneous networks offering Internet access. Moreover, it is recommended to protect hypertext documents (HTML or XML files). Lately, a generalized RBAC model has been proposed where objects and environmental conditions are assigned roles (Moyer & Ahamad, 2001).

FUTURE TRENDS

Implementations

Some of the most popular database vendors (like IBM, Microsoft, and Oracle) have developed (mainly) database tools for the storage of XML documents, and several storage techniques have been adopted in order to maximize functionality and performance. Furthermore, for reasons of integrity, the majority of the DBMSs are supported by access control mechanisms.

In general, the most well-known software tools that employ XML document storage and access control can be categorized in the following types of databases:

- XML-Enabled DBMSs: These provide various interfaces for extracting structured data from XML documents and then to be stored in DBMSs (according to their model). Table 1 presents the basic products of such an approach.
- Native XML DBMSs: XML documents are stored in XML internally in the database. Table 2 shows some of the most popular DBMSs.

Currently, the most widely adopted technology to enforce the security guarantees of the XML-based databases is the Microsoft (MS) .NET platform. MS .NET has been recently proposed and has been adjusted to support XML Web Services. Its security mechanism is adopted by MS SQL Server 2000 (Rys, 2001), Oracle 9i, and DB2, which have been designed to cooperate with Microsoft .NET technology.

All of the above storage systems can be supported by access control modules. Most of them support the Microsoft .NET access control framework. The idea of protecting XML files is quite new, and therefore most of the implemented systems still belong to the scientific literature and they have not stepped into the market.

The core of every access control mechanism is a language used for expressing all of the various components of access control policies, such as subjects, objects, constraints, and so forth. To date several languages have been proposed. Since XML is a structured language, it can be used in the definition of access control issues, such as roles, policies, and authorizations. Therefore, the most well-known general-purpose XML-based access control languages are:

- XACML (eXtensible Access Control Markup Language): This is an OASIS standard that defines both a policy language and an access control decision request/response language (both written in XML) (www.oasis-open.org/committees/xacml/repository). The policy language is used to describe general access control requirements, and has standard extension points for defining new functions, data types, combining logic, and so forth. The request/response language permits forming a query in order to ask whether a given action should be allowed (or not), and to interpret the result that defines the response.
- XrML (eXtensible rights Markup Language): This is a language for digital rights management (Wang et al.). By using XrML, the owner of a digital resource can specify who can use the resource (principal) and under what circumstances (constraints). The owner thus produces licenses containing such information that are then given to the appropriate principal. Since XrML is mainly used for protecting digital media, it contains schemas defining the metadata of such resources, such as audio or video files. Of course, it can be easily extended for controlling access to any other type of resource.

Table 1. Storage in XML-enabled databases

Product	**DBMS Model**	**Storage**
Oracle 9i *technet.oracle.com*	Object-Relational	XML documents are stored as relational tables.
MS SQL Server 2000 *www.microsoft.com*	Object-Relational	Each XML document is stored as a relational table, and an element is created for each row.
IBM's DB2 *www-4.ibm.com*	Object-Relational	XML documents are stored as relational tables.
XIS *www.exceloncorp.com/xis/*	Object-Oriented	XML documents are stored in a B-tree-like structure.
Lore (McHugh et. al., 1997)	Object-Oriented	XML documents are stored using an Object Exchange Model (OEM).
SHORE (Hess, Schulz & Brossler, 2000)	Object-Oriented	This stores information extracted from XML documents using a variant of R-trees and B-trees structure.

Table 2. Storage in native XML databases

Product	**Storage Model**
NATIX (Kanne & Moerkotte, 2000)	Tree structure: XML documents are split into sub-trees (basic storage and query units) based on certain criteria.
SODA *www.cse.unsw.edu.au/~soda/*	Tree structure: XML documents are stored in a single tree, which preserves all XML information.
Xyleme *www.xyleme.com*	Tree structure: XML documents are stored as trees until a certain depth where byte streams are used.
Ipedo *www.ipedo.com/html/products.html*	Collection structure: XML documents are organized into collections (like directories), which can be typed or un-typed.
eXist *exist.sourceforge.net*	Collection structure: XML documents are stored using a DOM-tree (built from SAX-events) and are organized into hierarchical collections, similar to storing files in a file system.
Tamino (Schoning, 2001)	Collection structure: Each XML document is stored in exactly one collection.
Xindice *www.dbxml.org*	Collection structure: Each XML document is stored in at least one collection (a collection can be created either consisting of documents of the same type, or a single collection can be created to store all documents together).

Table 3. Examples of access control tools

Product	**Type**	**Access Control**
XENA (Tan & Zhao, 2000)	Non XML-Based	Centralized tool protecting XML repositories using relational database technology.
Author-X (Bertino, Castano & Ferrari, 2001)	XML-Based	Java-implemented tool, supporting varying security granularity levels. PULL and PUSH dissemination technology.
ACP (Access Control Processor) (Damiani, De Capitani di Vimercati, Paraboschi & Samarati, 2000)	XML-Based	Decentralized system supporting distributed XML repositories by using both local and global authorizations.
ProvAuth (Kudo & Hada, 2000)	XML-Based	Access control environment where access is allowed with some provisions. It uses XACL.

- ODRL (Open Digital Rights Management): This is a language used for Digital Rights Management (DRM) in open environments (odrl.net/1.1/). It can express rights for protecting any digital content, like electronic publications, digital images, audio and movies, learning objects, computer software, and so forth.

Except for the languages, research has to present several access control tools protecting XML files. Those can be categorized into non XML-based and XML-based. Table 3 contains some of the tools belonging to both categories.[1] The only tool using a standardized access control language is ProvAuth (Provisional Authorizations) (Kudo & Hada, 2000). As its name suggests, access is allowed to users only after (or before) the execution of some provisional actions, like logging the session, encrypting the view of a protected document, and so forth. To support such functionality, a specific-scope XML access control language (XACL) has been introduced which is part of the IBM XML Security Suite. Due to its extensibility and flexibility, the model can be integrated

into a conventional Web application consisting of clients and servers.

CONCLUSION

This article has presented an overview for storage and access control of XML documents. It is important to indicate that no definite guidelines have yet been proposed for selecting an optimal solution when storing and securing XML documents. In order to improve the management of XML documents, some issues should require further research. In particular, the storage of XML documents may be improved by using some data mining techniques (e.g., specific clustering algorithms). Furthermore, the XML management techniques should further extend existing access control policies, in order to improve the security in XML documents accessing. Finally, the commercial DBMSs should be extended to support more sophisticated storage and access control techniques such as integrated methods for locating and accessing dynamic XML documents.

ENDNOTE

[1] The names of the tools, ACP and ProvAuth, are assigned by us for reasons of brevity.

REFERENCES

Amer-Yahia, S. & Fernandez, M. (2002). Techniques for storing XML. *Proceedings of the 18th International Conference on Data Engineering* (ICDE 2002), San Jose, California, USA.

Bertino, E., Castano, S. & Ferrari, E. (2001). Securing XML documents with Author-X. *IEEE Internet Computing*, *5*(3), 21-31.

Bohannon, P., Freire, J., Roy, P. & Simeon, J. (2002). From XML schema to relations: A cost-based approach to XML storage. *Proceedings of the 18th International Conference on Data Engineering* (ICDE 2002), San Jose, California, USA.

Damiani, E., De Capitani di Vimercati, S., Paraboschi, S. & Samarati, P. (2000). Designing and implementation of an access control processor for XML documents. *Proceedings of 9th World Wide Web Conference (WWW9) and Computer Networks*, Amsterdam, Holland.

Hess, A., Schulz, H. & Brossler, P. (2000). *SHORE—a hypertext repository based on XML*. Technical Report, SD&M Corporation, Software Design and Management, Southfield, USA.

Howes, T.A., Smith, M.C. & Good, G..S. (1999). *Understanding and deploying LDAP directory services*. Macmillan Technical Publishing.

Joshi, J.B.D., Aref, W.G., Ghafoor, A. & Spafford, E.H. (2001). Security models for Web-based applications. *Communications of the ACM*, *44*(2), 38-44.

Kanne, C.C. & Moerkotte, G. (2000). Efficient storage of XML data. *Proceedings of the 16th International Conference on Data Engineering* (ICDE 2000), San Diego, California, USA.

Khan, L. & Rao, Y. (2001). A performance evaluation of storing XML data in relational DBMS. *Proceedings of the 3rd ACM CIKM'01 Workshop on Web Information and Data Management* (WIDM 2001) (pp. 31-38), Atlanta, Georgia, USA.

Kudo, M. & Hada, S. (2000). XML document security based on provisional authorization. *Proceedings of the 7th ACM Conference on Computer and Communications Security* (pp. 87-96), Athens, Greece.

Marron, P.J. & Lausen, G.. (2001). On processing XML in LDAP. *Proceedings of the 27th Conference on Very Large Data Bases* (VLDB 2001) (pp. 601-610), Roma, Italy.

McHugh, J., Abiteboul, S., Goldman, R., Quass, D. & Widom, J. (1997). Lore: A database management system for semi-structured data. *ACM SIGMOD Record*, *26*(3), 54-66.

OASIS eXtensible Access Control Markup Language Technical Committee. *eXtensible Access Control Markup Language (XACML) Committee Specification, Version 1.1*. Retrieved February 15, 2004, from www.oasis-open.org/committees/xacml/repository.

Open Digital Rights Language Initiative. *Open Digital Rights Language (ODRL) Specification, Version 1.1*. Retrieved February 15, 2004, from odrl.net/1.1/.

Pallis, G.., Stoupa, K. & Vakali, A. (2004). *Storage and access control issues for XML documents*. In D. Taniar & W. Rahayu (Eds.), *Web information systems* (Chapter 4, pp. 104-140). Hershey, PA: Idea Group Publishing.

Pardede, E., Rahayu, J.W. & Taniar, D. (2004). On using collection for aggregation and association relationships in XML object-relational storage. *Proceedings of the 19th ACM Symposium on Applied Computing* (SAC 2004), Nicosia, Cyprus.

Rys, M. (2001). Bringing the Internet to your database: Using SQL Server 2000 and XML to build loosely coupled

systems. *Proceedings of the 17th International Conference on Data Engineering* (ICDE 2001) (pp. 465-475), Heidelberg, Germany.

Sandhu, R.S., Coyne, E.J. & Feinstein, H.L. (1996). Role-based access control models. *IEEE Computer*, *29*(2), 38-47.

Schoning, H. (2001). Tamino—a DBMS designed for XML. *Proceedings of the 17th International Conference on Data Engineering* (ICDE 2001) (pp. 149-154). Heidelberg, Germany.

Schmidt, A., Kersten, M., Windhouwer, M. & Waas, F. (2000, May). Efficient relational storage and retrieval of XML documents. *Proceedings of the 3rd International Workshop on the Web and Databases* (WebDB 2000) (pp. 137-150), Dallas, Texas.

Silberschatz, A., Korth, H. & Sudarshan, S. (2002). *Database system concepts* (4th Edition). McGraw-Hill.

Tan, K. & Zhao, J. (2000). Effective storage and access control of XML documents with relational database system. Retrieved November 16, 1997, from xena1.ddns.comp.nus.edu.sg:8080/Xena/paper.pdf.

Tian, F., DeWitt, D.J., Chen, J. & Zhang, C. (2002). The design and performance evaluation of alternative XML storage policies. *ACM SIGMOD Record*, *31*(1), 5-10.

Wang, X., Lao, G., DeMartini, T., Reddy, H., Nguyen, M. & Valenzuela, E. XrML-eXtensible Rights Markup Language. *Proceedings of the ACM Workshop on XML Security* (pp. 71-79), Fairfax, Virginia.

Widjaya, N.D., Taniar, D. & Rahayu, J.W. (2004). Transformation of XML schema to object relational database. In D. Taniar & W. Rahayu (Eds.), *Web information systems* (Chapter 5, pp. 141-189). Hershey, PA: Idea Group Publishing.

Zhu, Y. & Lu, K. (2001). An effective data placement strategy for XML documents. *Proceedings of the 18th British National Conference on Databases* (BNCOD 2001) (pp. 43-56), Chilton, UK.

KEY TERMS

Database Management System (DBMS): A collection of interrelated data that is called a database, and a variety of software tools for accessing those data. The three leading commercial DBMSs are: Oracle 9i, IBM DB2, and Microsoft SQL Server.

Discretionary Access Control (DAC): A model where the owner is responsible for defining policies in order to protect his/her resources and (s)he can also define who is going to be assigned some access privileges.

DTD: A set of rules defining the element types that are allowed within an XML document, and specifying the allowed content and attributes of each element type. Also defines all the external entities referenced within the documents and the notations that can be used.

Lightweight Directory Access Protocol (LDAP): A protocol definition for accessing specialized databases called directories. An LDAP directory is organized by a simple "tree" hierarchy.

Mandatory Access Control (MAC): A model expressed with the use of security labels associated with both subjects and objects.

Native XML Database: A database that defines a model for an XML document (as opposed to the data in that document), and stores and retrieves documents according to that model. At a minimum, the model must include elements, attributes, PCDATA, and document order. Examples of such models are the XPath data model, the XML Infoset, and the models implied by the DOM and the events in SAX.

Role-Based Access Control (RBAC): An access control model where subjects are organized into roles defining their duties in an organization or environment.

XML Document: A document consisting of an (optional) XML declaration, followed by either an (optional) DTD or XML schema, and then followed by a document element.

XML Schema: An alternative to DTDs; a schema language that assesses the validity of a well-formed element and attribute information items within an XML document. There are two major schema models: W3C XML Schema and Microsoft Schema.

ENDNOTES

1 XML Documents: Storage and Access Control Policies