

Exploiting XML Dom for Restricted Access of Information

B.Naga malleswara Rao¹, N.Samba Siva Rao², V.Khanaa³

Department of Computer Science and Engineering

Bharath University, Chennai -600 073, India.

Email ¹: bnmallik@yahoo.com ,

Email ²: {snandam@hotmail.com, drvkannan62@yahoo.com}

Abstract - Time bound access (TBA) has been a major challenge for online information distribution services. A secure and constrained information transmission over the public network is the key aspect of such services wherein a genuine user who is privileged should not be denied of service under any circumstances.

As it is quite obvious that a document stored in a particular file format at the source side, when requested for, needs the right application at the destination for an exact interpretation of the same and this forces the end users to have all the available proprietary applications in the systems' abode. This overhead can be avoided by choosing a format that has well compliance portability independency, throughout the process of transmission and XML provides with all these features.

A clue to the afore-mentioned problem has thus seen light and we mean to say that, the intelligence when encapsulated in XML format, can successfully override the hurdles caused by the existing scenario and also provides with a framework for online information distribution services for achieving the desired goals of constrained and secure information transmission. Finally, we describe cryptographic mechanisms for enforcing the protection model on published data XML files.

Index Terms - XML DOM, DES, CA. Cryptographic Key assignment. XML signature.

I. INTRODUCTION

The Extensible Markup Language (XML) is widely used for data presentation, integration, and management because of its rich data structure. In applications such as business transactions and medical records, sensitive data may be scattered throughout an XML document and access control on node-level (element or attribute) is required to ensure that sensitive data can only be accessed by authorized users. Access control must be expressive and be able to support rules that select data based on the location and values of the data. In practical applications, such as electronic libraries, and credit card companies, the number of access control rules is on the order of millions, which is a product of the number of document types (in 1000's) and the number of user roles (in 100's). It is obvious that such applications call for high scalability and performance from the underlying access control system.

Restricted or time-bound access (TBA) of information for an authenticated and authorized end user can be provided by deriving time bound cryptographic keys [8] that are assumed to be different for different time

periods thus limiting the users access when the period expires. Data Encryption standards [3] allow the service providers to impinge security and constraints on the information. The Time Bound keys are generated for each class and they are used for encryption. Deriving the same keys that are used for encryption can decrypt the information [1]. Keys are derived corresponding to the users input of subscription period that is given as input to the algorithms that derive the key.

In the existing scenario information is stored in one format in the source and it remains forever. It failed to provide a flexible transaction in the Internet, since while accessing information the format may differ in different parts of the network. The document may need to be transferred to different applications and it will not be handled efficiently. In the proposed Time Bound system this drawback is rectified by representing the class information in the form of Document Object Model tree structure, where the information can be transferred anywhere without considering the format prevailing in the recipient machines.

In this paper, we argue that such a possibility is becoming very concrete with the advent of new technological standards, such as XML [4], and XML DOM and with the parallel development of DOM interface based parsers [12], Furthermore, the Internet and web communities are repeatedly proposing the use of XML in network protocols applications-XML-RPC [11], DOM [10], are only a few examples.

Our proposed approach falls under the generic framework of security services such as information is stored in one format in the source and it remains forever. It failed to provide a flexible transaction in the Internet, since while accessing information the format may differ in different parts of the network.

The rule that we propose in this paper is currently supported by XML; however, the standard bodies, and particularly the W3C, are taking appropriate steps in order to make the implementation of such rule rather simple. In particular, XML rules capitalize on the existence of events in DOM and of a standard XML query language, named Xquery, which has recently been proposed by W3C XML Query working Group.

We have presented two specific instantiations of access control-using XML. The information is to be grouped into a hierarchical structure according to the specification given by the source. It is to be encrypted

using cryptographic keys and they are sent to users. And the user can be able to decrypt them using the same keys, which are used for encryption. The user subscribed for a higher class can be able to decrypt all the lower class information.

The information is stored in XML DOM structure and since DOM structure contains the elements in their nodes, insertion and deletion of node or element is more flexible. Taking into account the class relation and time period keys are derived for the class information. They are generated by mathematical computations [5], [6]. The information is encrypted using the keys and they are sent to the user. The user using information from the source derives the keys. The merit of the existing system over the proposed system is flexibility of transferring information.

This paper organized as follows. After an overview of related work in Section 2, Section 3 briefly presents the syntax and semantics of XML DOM. Section 4 gives the architecture of Source Key generation and Encryption and describes the application scenario for rule access control using XML. Finally, Section 5 draws the conclusions.

II. AN OVERVIEW OF XML DOCUMENTS

An XML document consists of three parts: an XML declarations, a DTD (Document Type Definition) or XML Schema, and an XML instance (XML document data). An XML declaration and Schemas are not mandatory for an XML document. An XML declaration specifies the version and the encoding of XML being used. A DTD or XML Schema is schema that constrains the structure of XML instances, and corresponds to an extended context free grammar. An XML instance is a tagged document. We omit concrete description of an XML declaration and a DTD.

An XML instance is a hierarchy of elements the boundaries of which are either delimited by start-tags and end-tags, or, for empty elements, by empty-element tags. Character data between start-tags and end-tags are the content of the element. Figure 2 shows an example of an XML instance. A start-tag is the token that encloses an element type with \langle / \rangle and $\langle \rangle$. Elements can nest properly within each other, the nesting represents logical structure. Within start-tags attribute names and attribute values can be specified.

XML Documents have two levels of conformance: Valid and well-formed. A well-formed XML document follows tagging rules prescribed in XML. An XML document is valid if it is well-formed and if the document complies with the constraints expressed in an associated schema.

A. Implementing Protection with XML Encryption:

The recent W3C Recommendation on XML Encryption Syntax and processing [15] provides a standard schema for representing encrypted data in XML form, along with conventions for representing cryptographic keys and specifying encryption algorithms.

The basic object is an XML element Encrypted Data containing four relevant sub elements: EncryptionMethod describes the algorithm and parameters used for encryption/decryption; KeyInfo describes the key used for encryption/decryption (but not contain the values); CipherData contains the output of the encryption function, represented as base64-encoded text; EncryptionProperties contains optional user-defined descriptive data. The cipher text included in the CipherData element is the encryption of an XML element or element content. When the encrypted content is itself an EncryptedData element, it is called nested encryption.

III. THE DOM MODEL FOR XML DOCUMENTS

The Document Object Model is an object model to represent XML documents and an interface to work with that model. It gives a tree overview of all XML elements and how they relative to one another. It is a good model for handling XML documents because it takes the tree like model of XML as its core idea and makes no presumptions about the structure of the document. This presents a good starting point for creating mappings to other tree based structures.

IV. RELATED WORK

The approach is supposed to lead to the design of a system that encapsulates all the features and functions necessary for a time bound information access scheme. Hierarchically structured information would pose access control related challenges that can be solved by assessing different key at different levels, For example, the data in a government are usually classified into four classes: “unclassified”, “confidential”, “secret” and “top-secret” such that only top officers can access top secret data. The classes are arranged as “unclassified” \leq “Confidential” \leq “secret” \leq “top-secret”. If the top secret data is encrypted with K_4 , the secret data with K_3 , the confidential data with K_2 , and the unclassified data with K_1 by a key assignment scheme. The key K_i is used to derive K_j for $j \leq i$. If the top officer is given the key K_4 , he can gain information in all classes. If an employee gets key K_2 , he can only read the information in classes “unclassified” and “confidential” only. This type of key allocation can control the information flow. An elaborate work on time bound access control scheme is presented in [7].

Similarly in the proposed system, the information items are classified into security classes C_i ($1 \leq i \leq m$), which are partially ordered by a binary relation such that they form a partial order hierarchy. They are stored as DOM tree hierarchy structure. In the partial order hierarchy $C_j \leq C_i$ denotes that the security level of class C_j is lower than that of class C_i and $C_j \leq C_i$ denotes additionally that $C_j = C_i$ is possible. One function of Central Authority is to assign a cryptographic key K_i to each class C_i such that K_i is used to derive K_j if and only if $C_j \leq C_i$.

The key K_i will be used to encrypt the data in class C_i . The user subscribes for an account with his details such as authentication information, duration of subscription; level of his class in the hierarchy etc will be stored in the database. Each user in the system is assigned

a class according to his security clearance. A user with higher security clearance will be assigned a higher class and a user with lower security clearance will be assigned a lower class.

The problem concentrates on time bound key assignment scheme in which a user may be in a class for only a period of time. Each class C_i has many class keys $K_{i,t}$, where $K_{i,t}$ is the class key of C_i at time t . The time is not necessarily a real time. Actually the time is divided into time periods, starting at 0. When the user enters in the system his starting (t_1) and ending period (t_2) which are available in the database are compared with current time (t). If t lies between t_1 and t_2 , then he is a valid user. The maximum number of time periods is $z + 1$, where z is an integer. The constraint on maximum number of time periods will not be the limitation of the system.

The information in each node of a class will be encrypted using the generated key of that class. The publicly known Hash Function and DES are used for encryption. The user who subscribed for the class C_i can get the information content $I(i, t_1, t_2)$, $1 \leq i \leq m$, $t_1 \leq t \leq t_2$, from the source. With that information $I(i, t_1, t_2)$, the user can compute the class key $K_{j,t}$ of C_j at time t if and only if $C_j \leq C_i$ and $t_1 \leq t \leq t_2$. And by using that key information which same as the key used for encryption in the source the user can decrypt the data stored in class C_j at time t . The following figures show the source key generation and database validations.

Many TBA approaches for enforcing XML access control have proposed. Some of them [14] support full XPath [8] expression to provide expressiveness by creating the projection of the policy on DOM tree. However these approaches incur massive cost when handling large policy or a deeply layered XML document. The mechanisms proposed [10] perform more efficiently but also encounter the same problem since the node-level access control on a DOM-based view can be expensive when processing large numbers of XML documents.

A different approach with document-level optimizations is also proposed by Yu et al. [16]. Their scheme enforces efficient access control with an accessibility map which is generated by compressing neighboring accessibility rules to improve cost efficiency. However, since the maps are generated on a document-level, document updates or policy updates may trigger expensive re-computations especially for a large XML database.

V. SYSTEM ARCHITECTURE

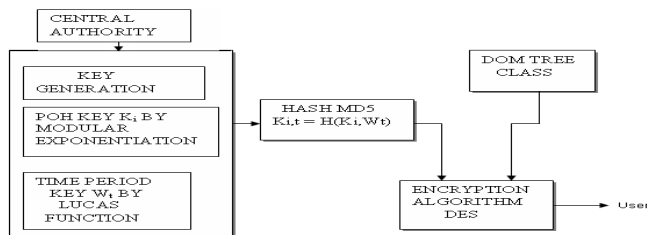


Figure: 1 Source Key Generation and Encryption

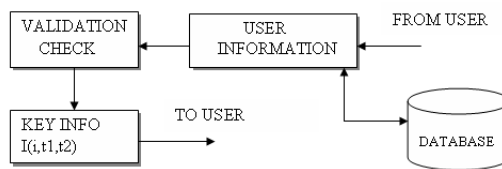


Figure 2: Database Validation

The Figures shown above gives the Source Key Generation and Encryption and Database Validation.

So far we discussed the application of XML DOM to implement secure portable in dependent applications. We infer that XML DOM can be used for successful implementations of security applications as well as document maintenance. In the present work, XML DOM has been chosen to generate the keys for classes.

The information is grouped into security level classes and it is represented in XML DOM tree structure in the source. A class can have many different keys, which differ in time period, but time period of the key should not exceed the total time period of the system. The keys will be generated by mathematical computations, such as Modular exponentiation [8] (for partial-order hierarchy) and Lucas Function [9] (for time period). Both will be combined and given as input to the hashing function to produce a single hash code, which is to be used as the key for encryption. That hash code and the plaintext from the DOM tree are taken and given as input to the DES algorithm for encryption.

The approach is based on a cryptography mechanism and adopts a hash function to ensure integrity. DOM_HASH is the first algorithm proposed by Maruyama to calculate a has value for XML data. In this algorithm MD5 and SHAI were adopted to generate has values with four different node types related to xml data. The four typed include element, attribute, processing instruction (PIs), and text. This algorithm is limited to the contents of the XML data and therefore, does not provide for authentication of the internal or external subset of the DTD. Inspired by DOM HASH, the XHASH algorithms has been proposed Brown [21]. The XHASH makes use of two parameters the first digit function such as SHA1; the second which is optional, can be used to determine how non-significant space characters will be handled by default. However, possible values for this attribute are limited to 'default' and 'preserved'. Thus, there is no known way to explicitly specify that non-significant space characters should be discarded. W3C published XML signature specifications in 2000. This specification provides the format for data integrity expression in XML signatures, and gives the optional algorithm to generate digest value, such as SHA-1, SHA-256. However, signed XML data can be copied to another document but still keep signature valid. Devanbu adopted DOM-HASH and the Merkle has function to maintain the integrity of

XML data queries [20]. Bertino also adopted the Merkle hash tree to handle XML documents [21]. Based on cryptography, this kind of approach has a higher security level than first approach. But, this kind of approach still has some problems described above.

VI. XML DOM REPRESENTATION OF CLASS INFORMATION

In this section we show the guidelines that can be used for the implementation of Time bound cryptography for access using XML control. The class information is represented in the form of DOM tree structure. Each node in the tree corresponds to a textual data. DOM parser processes the node values. The hierarchy is maintained by arranging the sequence of the nodes in the tree. If a particular node is selected only the information that falls under that node as well as child nodes of the particular nodes can be accessed. Sample code [2] for the system

```
<?xml version="1.0"?>
<Xpress>
<all>
<news>
<general>
<gen>generalnews1</gen>
<gen>generalnews2</gen>
<gen>generalnews3</gen>
</general>
<worldnews>
<wn>worldnews1</wn>
<wn>worldnews2</wn>
<wn>worldnews3</wn>
</worldnews>
<sportsnews>
<sp>sportsnews1</sp>
<sp>sportsnews2</sp>
<sp>sportsnews3</sp>
</sportsnews>
</news>
<entertainment>
<movie>
<mo>movienews1</mo>
<mo>movienews2</mo>
<mo>movienews3</mo>
</movie>
</entertainment>
</all>
```

The system concerns with the implementation of DOM tree, mathematical background which are used to generate keys by the CA, encryption of class information, sending the key information to authorized client and decryption of information and finally.

```
com.ibm.xml.parser.*;
import org.w3c.dom.*;
The child nodes are accessed by using the java
property,
NodeList list = n.getChildNodes();
```

The information can be added further by using the property

```
Document d = new TXDocument();
Element e = d.createElement("astro");
e.appendChild( d.createTextNode("LEO"));
d.appendChild(e);
```

VII. DISCUSSION AND ANALYSIS

A. Compatibility with XML signature specification

The "XML" signature Syntax and Processing" recommendation is an internet standard which defines syntax and processing model of special format for digital signatures. Standard contents describe clear statement of the regulations on XML signature to maximize the security and the extent of the standardized contents, integrity, message and user authentication and no repudiation. These signatures are represented in an XML format and can sign arbitrary resources, including XML and parts thereof

The structure and processing of XML signatures introduces some interesting concepts which will be explained briefly. The primary elements of XML signatures are digital signature information. Signature elements consist of "SignedInfo" with digital signature information, "SignatureValue" with actual digital signature value and "KeyInfo" with digital signature key information. In particular, "SignedInfo" describes how signature information is standardized, the algorithm for the signature and subordinate algorithm. "Reference" consists "DigestMethod", the algorithm summarizing signature data, and the element "DigestValue" showing the signature data, and the element "DigestValue" showing the result "KeyInfo" described in XML security is used to illustrate key information in XML digital signature.

```
<Signature>
<SignedInfo>
<CanonicalizationMethod/>
<SignatureMethod/>
<Reference>
<DigestMethod/>
<DigestValue/>
</Reference>
</SignedInfo>
<SignatureValue/>
<KeyInfo/>
</Signature>
```

As described in our proposed scheme, the result time bound access is a has value, thus it can be described in element "DigestValue" and this approach falls under frame work of security services such as information is stored in one format in the source and it remains forever.

VIII. EXPERIMENTS

In this section we describe our experiments to evaluate the security of our XML format based access control mechanism for XML documents. All the

experiments we conducted on a machine with 2.0GHz Dual core CPU, 2 GB of main memory and JDK 1.5.

To demonstrate the security of the XML file format, we examine the XML format policy is loaded into destination side and access control processing time when a large XML document is processed with secure format.

IX. RESULTS

The main purpose of this experiment is to see whether a clue to the afore-mentioned problem has thus seen light and we mean to say that, the intelligent when encapsulated in XML format, can successfully override the hurdles caused by the existing scenario and also provided with framework online information distribution services for achieved the desire goals of constrained secure transmission with cryptographic mechanisms.

X. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed the use of a Time Bound access control Scheme. We have shown that such Scheme satisfy the needs of many important security applications. Below, we list some of the possible obstacles that could limit the applicability of our solution, and explain why each of them is not critical.

The first observation is that in order to write an efficient service it may be necessary to know the schema of XML DOM resources. Where the scheme for a set of classes represented in DOM structure is more flexible for transactions in the Internet and the changeover of documents from one application to other. Since XML is used for representing class information, the information can be given as input to another application, which uses different format of data representation.

The system can be enhanced further by designing a key assignment scheme for a partial-order hierarchy such that the number of public parameters is independent of the total number of classes in the hierarchy. This can be accomplished by using XML path language, which helps in specifying the path to the element relative to the current element being processed. Relative path easily locates elements nested within other element. Implementing Advanced Encryption Standards can also enhance the level of encryption.

REFERENCES

- [1] Avishai Wool (2000), 'Key Management for Encrypted Broadcast', ACM Transactions on Information and System Security, Vol. 3, No.2, pp. 107-134.
- [2] Brett McLaughlin, 'Java and XML', first edition, O'Reilly publications.
- [3] Peter Gemmill (2000), 'A Survey of Basic Cryptography', CS 491/591-10.
- [4] T.Bray, J. paoli, and C.M. Sperberg-McQueen(Eds.) "Extensible Markup Language (XML) 1.0-2nd Edition", Oct.2000.
- [5] A.J.Meneses, P.C. van Oorschot, and S.A. Vanstone, Handbook of Applied Cryptography. Boca Raton: CRC Press, 1996.
- [6] B.Schneier, Applied Cryptography: Protocol, Algorithms, and Source Code in C, second edi., New York: John Wiley and Sons,1996.
- [7] S.G.Akl and P.D Taylor (1983), 'Cryptographic Solution to a problem of Access Control in a Hierarchy', ACM Trans. Computer Systems, Vol. 1, No. 3, pp. 239-248.
- [8] Wen-Guey Tzeng (2002), 'A Time Bound Cryptographic Key Assignment Scheme For Access Control in a Hierarchy', IEEE Transactions on Knowledge and Data Engineering, Vol. 14, No. 1, pp.182-188.
- [9] William Stallings, 'Cryptography and Network Security Principles and Practice', second edition, Published by Pearson Education-Book.
- [10] Angela Bonifati Politecnico di Milano Piazz Leonardo Da Vomci, Stefano Paraboschi "Pushing Reative Services to XML Repositories using Active Rules".
- [11] XML-RPC Spcification(userland)" Updated 16 October 1999. <http://www.xmlrpc.com/spe>.
- [12] Document Object Model (DOM) Level 2 Core Specification Version 1.0 W3C Proposed Recommendation", September 2000. <http://www.w3.org/TR/2000/PR-DOM-Level-2-Cpre-20000927/>.
- [14] A. Gabillon and E. Bruno: Regulating Access to XML Documents. Working Conference on Data and Application Security (2001) pp-219-314.
- [15] D. Eastlake and J.Reagle. Xml encryption syntax and processing. <http://www.w3.org/TR/xmlenc-core>, 3 October 2002. W3C Proposed Recommendation.
- [16]. T.Yu, D. Srivastava, L.V.S Lakshmanan, and H.V Jagadish. Compressed Accessibility Map: Efficient Access Control for XML VLDB (2002) pp.478-489.
- [17] H.Maruyama, K.Tamura, N.Uramoto, Digest Values for DOM (DOM-HASH), RFC2903. Available at: <http://www.landfield.com/rfcs/rfc2803.html> (Accessed 13 November 2008)
- [18] WWW Consortium Document Object Model 2.0, 200. <http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113/>. W3C Recommendation 13 November, 2000.