

Paper:

# An Efficient Authorization Mechanism for Secure XML Sources on the Web

Sun-Moon Jo\* and Weon-Hee Yoo\*\*

\*Department of Computer Information Technology Education, Paichai University  
439-6 Doma-2Dong, Seo-Gu, Daejeon, Korea  
E-mail: sunmoon@pcu.ac.kr

\*\*Department of Computer Science and Information Engineering, Inha University  
253 Yonghyun-Dong, Nam-Gu, Incheon, Korea  
E-mail: whyoo@inha.ac.kr

[Received October 28, 2005; accepted March 17, 2006]

**XML-based access control technology aims at providing an authorization policy that can be consistently applied to various products for access control services on Internet and different kinds of environment for the products and thus providing interoperability to the existing access control products with diverse kinds of environment and types. The existing access control fails to consider information structure and semantics sufficiently due to the fundamental limitations of HTML. In addition, XML document access control supplies only action read and permits alterations of very limited value for action write. The existing access control has disadvantages that DOM tree should be loaded on memory while all XML documents are parsed to generate DOM tree; that a lot of memory is used in repetitive search for tree to authorize access to all nodes in DOM tree; and that the complex authorization evaluation process may lower system performance. In this paper, we present an authorization mechanism for secure XML sources on the Web.**

**Keywords:** XML document, authorization rule, XML security, subject, object

## 1. Introduction

Over the past five years the eXtensible Markup Language has appeared, offering a framework which promotes the movement of business information across networks. XML is a data format for structured document interchange on the Web which is used to create data structures that can be shared between and among disparate and otherwise incompatible systems [2, 13]. XML can provide a standard data model for exchanging information on a lot of data generated during the operation of corporate database or applied program. For this reason, it is so suitable for documentation management system or component specification requiring definition and description of detailed information and its meaning.

However, as more and more XML-type information is

provided in web environment, a developer or users increased drastically in concerns about security issues for XML documents. As a result, the XML standardization group increasingly perceived the need of functional definition for security and defined standards related to XML security. There are now such fields of XML security technology as XML encryption, XML signature, XKMS, and XACML [7, 10]. As for the existing studies and products for security of XML documents, transmission-layer security protocols including XML signature and encryption and access control to information in web environment were mostly related to HTML documents, which was file-based access control and failed to deal with access control by meanings of partial information, or the principal advantage of XML. So an access control method with the advantages of XML applied became necessary [5].

As for the existing access control, an access control technique becomes complicated for each operation as a new operation is added; labeling and DTD (Document Type Definition) verification processes consume much memory on repetitive DOM tree retrieval and parsing of XML documents, which can reduce the efficiency of the system [5]. The object within DOM enables a developer to read, explore, revise, add, or delete data from documents. It also provides standard functional definition for document navigation and a function to operate contents and structure of HTML and XML documents. However, DOM, which enters the entire document in memory and parses it to document tree, may use very large memory too much and reduce the efficiency of application rapidly. Of course, it can vary with library being used or the internal structure, DOM expression can require memory about ten times as large as the original. With a server to manage a few MBs of document at once, DOM can rapidly cause a bottleneck.

In this paper, we present an authorization mechanism for secure XML sources on the Web. And suggested is a mix of DOM and SAX (Simple API for XML) for parsing of XML documents. What is therefore expected is possibility of maintaining more rapid access control policy security with higher efficiency than that of the existing access control model by managing users and access autho-

rization information management more easily and removing unnecessary parsing and DOM tree retrieval.

The paper is organized as follows: Section 2 examines studies and problems about access control. Section 3 defines the concept of XACML and an action label type group for access authorization system to describe new algorithm and access authorization techniques. Section 4 evaluates the access authorization system suggested and section 5 draws a conclusion and describes the future course of studies.

## 2. Related Works

DOM (Document Object Model) [3] is a platform-independent and language-neutral interface which can express contents and structure of Internet documents including HTML and XML ones as objects and handle them, that is, which can dynamically change contents, structure, and style of documents. As the definition of interface that can be used by operating and accessing objects (tags) and documents, DOM Level 1 can express contents of HTML or XML documents parsed without a loss, support HTML 4.0 and XML 1.0, generate a hierarchical structure of documents, and easily are expanded to use high-level API with ease. Level 2 defines the function of supporting an object model applying style sheet and operating information on the style of documents. DOM higher than Level 2 also includes description of user interface usable in Windows environment. By using this, a user can also define the function and security level of operating DTD of documents. That is, DOM higher than Level 2 serves to design API that enables a user to define, operate, change, and access all the things of documents, including style, events, structure, contents, and security level.

As a standard interface to analyze event-based XML, SAX (Simple API for XML) serves to deal with XML documents using DOM with rapidity and low-level memory. It also analyzes XML documents to extract necessary information and provides API to analyze large-size XML documents more efficiently. The following examines advantages and disadvantages of SAX [11]. Fig.1 compares the styles of dealing with XML documents between DOM and SAX.

### ◆ Advantages of SAX

- SAX API is generally simple, compared with DOM.
- SAX doesn't enter the entire document in memory. So it holds a very small capacity.
- It can form a data structure containing key information which needs less memory than DOM.
- It can conduct operations rapidly because it doesn't read the entire document before beginning the operations.
- It is proficient in filtering and selecting data.

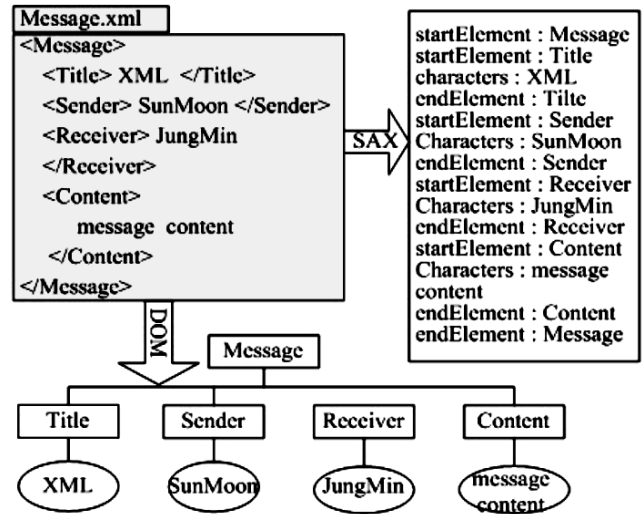


Fig. 1. XML of DOM and SAX a document processing comparison.

### ◆ Disadvantages of SAX

- The sequential model provided by SAX doesn't permit random access to XML documents.
- It cannot make another DTD or revise existing one.
- Since it doesn't have the entire document within memory, it is necessary to prepare modifications for each structure or content. As a user cannot control all elements in documents, it is impossible to make a new document.

An XML DOM tree provides API (Application Program Interface) to access elements of XML documents. The existing access control models [1, 4-6] use such a DOM tree to set access authorization to elements of DTD and XML documents and control users' access to XML data according to information on access authorization set.

According to the process of changes in documents in [6], there is a request for seeing XML documents. As for all XML documents and DTD concerned, information on access authorization is specified in documents called XAS (XML Access Sheet). XML documents are parsed to obtain DOM trees; then, a value of sign is set which means admission (+) or rejection (-) of access to nodes of DOM trees based on XAS of DTD and XML documents. It is called labeling to set authorization to nodes of DOM trees. The nodes with the value of sign set as - are removed from the labeled DOM trees and only those with the value set as + are restored to the user [1, 4-6]. Here, although XML documents with nodes removed from DOM trees can fail to be valid (its solution requires the loosening process, with all elements and attributes set as optional in DTD), they can maintain the existing DTD despite the removal of nodes from DOM trees.

To solve the problem that XML documents with nodes removed from DOM trees can fail to be valid for DTD,