

LockX: A System for Efficiently Querying Secure XML

SungRan Cho
Stevens Institute of Technology
scho@attila.stevens-tech.edu

Laks V. S. Lakshmanan
University of British Columbia
laks@cs.ubc.ca

Sihem Amer-Yahia
AT&T Labs—Research
sihem@research.att.com

Divesh Srivastava
AT&T Labs—Research
divesh@research.att.com

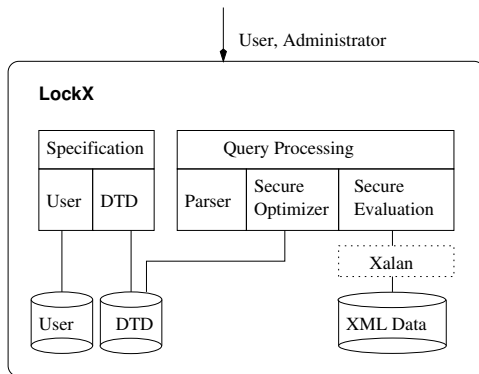


Figure 1: Architecture of LockX

1. MOTIVATION

Companies are using the Web for information dissemination, sparking interest in models and efficient mechanisms for controlled access to information. In this context, securing XML documents is important.

Much of the work on XML access control to date has studied models for the specification of XML access control policies, focusing on issues such as granularity of access and conflict resolution. However, there has been little work on enforcement of access control policies for queries.

A naive two-step solution to secure query evaluation is to first compute query results, and then use access control policies to filter the results. Consider the XML database of an online-seller, which has information on books and customers. Assume that a specific user is allowed access to books and not to customer information. If *only* query results are filtered for accessibility, the XPath query:

```
/seller[./customer/name='smith']//book
```

would allow the user to check the existence of a customer named smith, which is forbidden. Hence, data accessibility needs to be checked *during* query evaluation.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGMOD 2003, June 9-12, 2003, San Diego, CA
Copyright 2003 ACM 1-58113-634-X/03/06 ...\$5.00.

We describe LockX, a system for specifying and efficiently querying secure XML documents. LockX is based on a simple, yet useful, security model that uses a multi-level access control specification. The DTD identifies which elements must, may, or cannot specify their security level. If not specified, elements inherit their security level from their parent element. Each user is assigned a security level and can access elements whose security level is no higher than his own. Given a user query and the security information in the DTD, LockX applies efficient algorithms developed in [1] to guarantee safe query rewriting. To the best of our knowledge, LockX is the first prototype that addresses efficient and secure XML query evaluation.

2. DEMONSTRATION OUTLINE

LockX is a Java-based prototype whose architecture is depicted in Figure 1. LockX uses function calls and can thus be implemented on top of any XQuery engine. Our implementation of LockX uses Xalan (xml.apache.org/xalan-c), and demonstrates the following functionality.

Access Control Specification: Users can select a DTD, browse it using a graphical form and specify a mandatory, forbidden or optional `SecurityLevel` attribute at elements in the DTD. Users are organized in groups where their security level can be displayed and modified.

Access Control Enforcement: Users can write an XPath query. The query is displayed as a tree whose nodes are initially annotated with RC (*recursive check*). The annotation RC may be simplified to LC (*local check*) or NC (*no check*), using the algorithm presented in [1], that determines an optimal set of security check annotations on query nodes, by analyzing the subtle interactions between inheritance of security levels and paths in the DTD graph. This optimization process is shown step-by-step for each query. Once a query is optimized, it is sent to the Xalan system for evaluation.

Access Control Explanation: Users can use the “What if” feature to modify the security level at query nodes, and understand the security implications of their choices.

3. REFERENCES

- [1] S. Cho, S. Amer-Yahia, L. V. S. Lakshmanan, and D. Srivastava. Optimizing the secure evaluation of twig queries. In *Proceedings of the International Conference on Very Large Databases*, 2002.