CS437 XML Webdata Management

Semester Project: Secure XML Publishing

Phase 1 Report

Aaron Powers

Tom Guenther

Matt Nuckolls

Abstract

This paper presents an overview of our progress on our semester project thus far. Our project is the study of Secure XML Publishing. This paper is organized as follows: First we introduce the problem in our Problem Statement section. Next we explore the problem further in the Analysis of the Problem section by a brief overview of XML and its shortcomings in terms of security. We then propose the remaining focus of our efforts for the rest of the semester in the Proposed Solution section. Lastly, we present a brief overview of the papers we have reviewed thus far on the topic of Secure XML Publishing.

Problem Statement

XML provides a versatile way to store data. Data stored in an XML format can be published on the web for distribution purposes. This allows for a convenient method of distributing data to other individuals interested in the data. Naturally, some of this data may be sensitive or may only be intended for a certain individual to read. A secure method of publishing XML data is needed to prevent unauthorized access. The method of secure publishing should support the ability to verify a receiving individual’s allowance access as well as preventing the individual from seeing portions of the document they are not intended to see. Additional features that would be beneficial would be for the publisher to have the ability to publish only changes to an existing document instead of republishing the entire document as well as the receiving individual having the ability to only receive changes to the document relevant to the portions they are allowed to access instead of re-downloading the entirety of the document.

Analysis of the Problem

XML is a flexible markup language. It is intended to be both machine readable as well as human readable. Though it has a specific structure defined by tags, elements, attributes, etc. it has no specific format nor requirement for tag sets. This flexibility makes it versatile but also vulnerable.

Since XML is simply a markup language, it has no intrinsic security mechanisms. Therefore, XML data storage and transmission security must be handled externally to the XML content. XML document encryption is necessary for secure storage and transmission. Furthermore, XML document access must also be handled externally. Again, XML has no intrinsic access controls. Since access and transmission control must be handled externally to the XML document (implicitly: the XML parser), a “3rd party” application may be required to authorize access over a secure transmission protocol such as SSL (secure socket layer).

Possible Solutions

One possible solution is to implement a publisher/subscriber middle layer that handles all transactions between the publisher (entity containing the data) and the subscriber. The middle layer would serve as a gateway to provide secure access to the stored XML data. At some level it will be assumed that the content provider who originates the data to be published will take reasonable precautions to ensure that the data is presented to the publisher in a secure way. The content provider will ensure that the middle layer is informed of any document access restrictions and constraints under which the publisher must act. At this point we don't have a clear picture of what this middle layer will look like. More effort will be spent on fleshing out the details. At the present time this is the solution we are exploring.

In addition, we will be seeking to build on the work of Mohammad Ashiqur Rahaman, Yves Roudier, and Andreas Schaad presented in their paper entitled “A Publish/Subscribe Model for Secure Content Driven XML Dissemination” by extending their work to include incremental updates. We will explore possible extensions to the publish/subscribe model presented by Rahaman, Roudier, and Schaad to include work by Mohamed Nabeel and Elisa Bertino, presented in their paper “Secure Delta-Publishing of XML Content”. Nabeel and Bertino present secure XML publishing that supports small incremental updates. We will explore this method further and study its implications for use in extension of Rahaman, Roudier, and Schaad's work.

Lastly, our project will focus on analysis of existing solutions to the problem of secure XML publishing. The next section of this report summarizes several papers already published on the topic. Each paper presents a solution to the problem and discusses some limitations to the proposed solution. We will explore these solutions further and attempt to lend our novel ideas to extending and enhancing the proposed models. Likely we will focus on a single model as there are many and our time is limited.

Review of Related Papers

Currently, there is a large amount of work being done in the area of Secure XML Publishing. Mohamed Nabeel and Elisa Bertino published a paper titled “Secure Delta-Publishing of XML Content”. In this paper the authors propose a method of secure XML publishing that supports small incremental changes without the publisher needing to republish the entire paper. Their method also allows that the third party publishers do not need to be trusted due to encryption used by the original publisher.

Erwin Leonardi, Sourav S. Bhowmick, and Mizuho Iwaihara published a paper titled “Efficient Database-Driven Evaluation of Security Clearance for Federated Access Control of Dynamic XML Documents”. The authors of this paper propose a method of Secure XML distribution that utilizes a “policy enforcer” and “data provider”. Essentially, the data provider stores the XML documents as well as the access policy of those xml documents. A user can query the policy enforcer for data from the XML documents which the policy enforcer passes on to the data provider. The data provider runs the query then returns to the policy enforcer the documents found in which the user is allowed to access.

A sophisticated approach to the problem of secure xml publishing was presented in a paper by Mohammad Ashiqur Rahaman, Yves Roudier, and Andreas Schaad titled “A Publish/Subscribe Model for Secure Content Driven XML Dissemination”. This paper presents a very detailed example of a publish/subscribe model and how it could be used in a real world scenario between different organizations. This method ensures that the XML schema as well as the data itself are both confidential, ensures the integrity of transmitted data, and removes the coupling of publishers with subscribers by inducing a dissemination layer between the publishers and subscribers. The method also ensures the XML data cannot be read by the dissemination layer. In this model a publisher first establishes authorization policies that are enforced by the dissemination network then a user sends a subscription request along with his credentials to the dissemination network. When a publisher publishes his encrypted XML document he annotates it with information regarding the “concept” of the document and the dissemination layer forwards the document to the subscribed users. In this model, users only get portions of the document they are authorized for; they can get them automatically by distribution from the dissemination layer; there is confidence in the integrity of the data; and, the data is kept hidden from anyone that is not authorized to see it. This model seems to solve many of the problems surrounding secure XML publishing as well as providing efficiency within the dissemination network layer. However, it does not appear that this model supports incremental updates from publishers.

Another method has been proposed by B. Carminati, E. Ferrari, and E. Bertino in their paper “Secure Third Party Distribution of XML Data”. This is yet another method that provides confidentiality, integrity, and authenticity in secure XML data publishing; but, it also provides completeness. Completeness is provided by ensuring that the users are receiving every portion of the document they are allowed to receive based on their access control policy. Like the method proposed by Rahaman, Roudier, and Schaad, this method does not require that the third party publishers be trusted. However, data still remains confidential. This is achieved by the user receiving keys directly from the data publisher.