Preventing Private Information in Secure Dissemination

Hye-Kyeong Ko

Division of Computer Engineering, Sungkyul University, Anyang-city, South Korea
hkko@sungkyul.ac.kr

Abstract. Secure dissemination of XML document is becoming a crucial requirement for many Web-based applications. By secure dissemination, we mean that the delivery of information to users must obey the access control policies. In this paper, we present an approach for the secure dissemination of web contents for users. Our approach uses a labeling scheme that protects the private information of web contents for users. Our experiment results show that the proposed approach is efficient protecting the private information.

Keywords: Security, Data Encryption, Access method, Secure dissemination.

1 Introduction

The success of the Web as a platform for EC and information dissemination has brought an increasing awareness of the fact that document exchange on the Internet should meet precise security requirements such as fine-grained authenticity, secrecy, and access control involving data units at the level of granularity stipulated by the communicating parties [1], [4]. In the Web environment, eXtensible Markup Language (XML) is rapidly becoming the standard for data representation and exchange. Today, more and more applications that utilize XML as the primary data format are being deployed. Generally, the content may have private information that needs protection. For example, if the content is disseminated carelessly, users can infer more information from the disseminated data in terms of inference [2], [8].

The rest of the paper is organized as follows. Section 2 surveys related work. Section 3 presents the principal techniques for the proposed secure dissemination service. Section 4 presents the results of our experiments, and we conclude in Section 5.

2 Related Works

To secure of an XML document, the XML Encryption Working Group of W3C develops a process for encrypting/decrypting XML documents and XML syntax used to represent the encrypted information that enable an intended user to decrypt it [13]. W3C XML Encryption [13] is only capable of encrypting full subtrees, while XML access control can remove sensitive material from the middle of the tree. In W3C XML Encryption, if the contents overlap, the same portions of the XML document could be re-encrypted for multiple users (called "super-encryption"). Access control concerns with who can access which information under what circumstances. XML

James J. (Jong Hyuk) Park et al. (eds.), *Advanced Multimedia and Ubiquitous Engineering*,
Lecture Notes in Electrical Engineering 352, DOI: 10.1007/978-3-662-47487-7_11

access control refers to the practice of restricting access to parts of XML data to only authorized subjects [1]. XML pool encryption approach [5] is able to hide the size and the existence of encrypted contents. In XML pool encryption, nodes containing sensitive information are selected, and are moved into a pool and encrypted. A number of security models have been proposed for XML [2], [3]. Under information push, the system periodically broadcasts documents to users, rather than sending them upon request [3]. The security views approach has recently been proposed in [4], [7]. Sensitive data is effectively protected from access and potential inferences by unauthorized subjects, and authorized subjects and provided with necessary schema information.

3 Proposed Technique

3.1 Proposed Approach

The system will be capable of encrypting arbitrary parts of the content. The work reported in this paper builds on the access control mechanism for content by Christian Geuer-Pollmann in [5]. The idea behind XML pool encryption is to remove private nodes from the tree and encrypt each private node individually. These encrypted nodes are stored in encrypted node sets. After decryption, each node can find its way back to its appropriate location in the content and be reconstructed correctly.

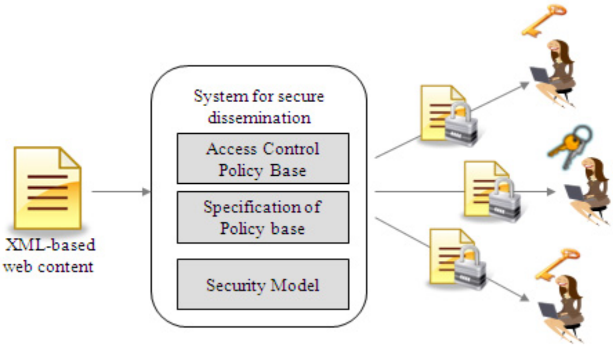
XML-based web content
System for secure dissemination
Access Control Policy Base
Specification of Policy base
Security Model

Fig. 1. Example of secure dissemination framework

The overall framework is depicted in Fig. 1. In the subscription phase, a user can be assigned rights, the service returns specific information to the user, which is required to decrypt parts of the XML-based web content source according to the user access rights.

Example 1. News Company provides content to subscribers. Two kinds of content are disseminated: paid content and free content. The paying subscribers can access paid content. The payment records identifying paying subscriber are managed by the company.

The game content in sports, and the story and the music content in culture are all paid content. Subscriber A paying for sports content can access both free and sport content.

Example 1 presents an example of encrypted dissemination of XML-based content. A pool of encrypted nodes (paid content) and unselected nodes (free content) are disseminated to multiple subscribers. Each subscriber can decrypt the content, depending on the access rights.

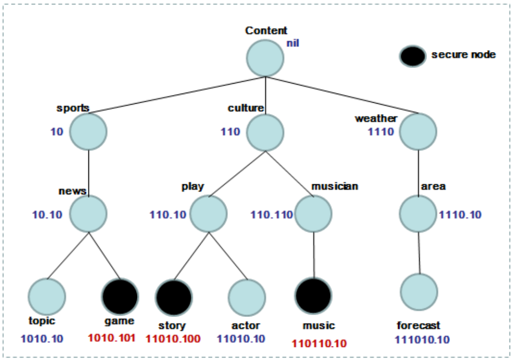

Fig. 2. Example of web content with labeling

3.2 XML Node Labeling

To decrypt the encrypted node, the system requires the location information of the decrypted node; i. e., where the node was in the original content. An efficient labeling scheme is very much required for identifying the location of a node. In this paper, we exploit an IBSL [9], which takes advantage of the lexicographical order of binary strings. The labeling procedure separates pubic and secure nodes. When the encrypted nodes are removed from the original tree, an adversary can guess the encrypted node location through labeling information and make good assumptions about the structure of the plaintext content. Algorithm 1 gives the details of the operation to label of public nodes with IBSL. In Algorithm 1, if the node is the root node, nil is assigned at the node (Lines 2 to 3 in Algorithm 1). For the child node of the root node, 10 is assigned at a first child for the second child (Lines 4 to 8 in Algorithm 1). Finally, the label of each sibling node is the *n.self label* concatenated with the *parent.label* (Line 10 in Algorithm 1). This method is applied until the full tree is traversed. After the traversal, each node in the tree has a unique value, i.e., a full label of public node. The delimiter "." is employed to assist subjects in figuring out the relationship (i.e., parent-child relationship) between nodes. For example, by looking at node 110.100, one realizes that it is a child of node 110. Algorithm 2 can help to label the secure node efficiently. Algorithm 2 can always label a new binary string between lexicographically ordered binary strings. The label length increases by one bit for the labeling by Algorithm 2. After labeling the plaintext document, the pruning procedure removes secure nodes from the labeled plaintext document. The pruned document is called public document. In fig. 2, the three terms (game, story and music) all refers to the same node, while it is in different states of the pool encryption procedure and the pool decryption procedure. Besides the basic functionality of being able to reconstruct the document, the system should prevent information leakage to the subject as good as possible. After decryption of the document, the subject has access to the labels of all public nodes and decrypted nodes.

Algorithm 1. Assign label of public node

```
Input: each public node n in the XML document
Output: label (N)

begin
for (i = 0 i < childnum i++) do
if (n is the root node) then
label(N) = nil;
else if (n is a child of root node) then
parent = root.childnum;
n.self label [1] = 10
n.self label [i] = 1 $\oplus$ n.self label [i-1];
label (N) = n.self label [i];
else parent.childnum++;
label(N) = parent.label $\oplus$ delimiter $\oplus$ n.self label [i]
end if
end for
return label (N)
end
```

Algorithm 2. Assign label of secure node

```
Input: N_left, N_right
Output: label (Nc)

begin
if (N_left is empty, but N_right is not empty) then
Nc = N_right $\oplus$ 0;
else if (N_left and N_right are not empty) then
if (len(N_left) $\leq$ len(N_right) then
Nc = N_right $\oplus$ 0;
else if (len(N_left) > len(N_right) then
Nc = N_left $\oplus$ 1;
else (N_left is not empty, but N_right is empty)
Nc = N_left $\oplus$ 1;
end if
return label (Nc)
end
```

The node reconstruction procedure takes the labeled public document and the decrypted nodes as input and inserts the decrypted nodes into the appropriate locations. In order to reconstruct the decrypted nodes, the reconstruction procedure has to identify which node inside the decrypted document is the parent node of the decrypted node and if that parent node already has child nodes, and determine whether some of the decrypted node's siblings are not siblings, and must be turned into children of the decrypted node.

4 Performance Evaluation

We conducted experiments to evaluate and compare the performances of the three schemes, namely NoLabel (baseline), XISS [10], and M-IBSL, used in XML pool encryption [5]. The experiment is against a DOM representation of the baseline without a labeling, namely NoLabel. M-IBSL scheme is implemented using Java 2 and XML Security Suite. Experiments were carried out on a 3.40GHz Pentium processor with 4GB of RAM running Windows 7. In selecting nodes to be encrypted, XPath was used. We conducted experiments 20 times to obtain small confidence intervals.

Table 1 present the XPath expression used to represent nodes to be encrypted.

Table 1. Locations of node to be encrypted

XMark
P1 //item/mailbox/mail/date
P2 /africa/item/*
P3 parlist/listitem/text
DBLP
P4 title//*/sub
P5 sub/sup/tt/ref

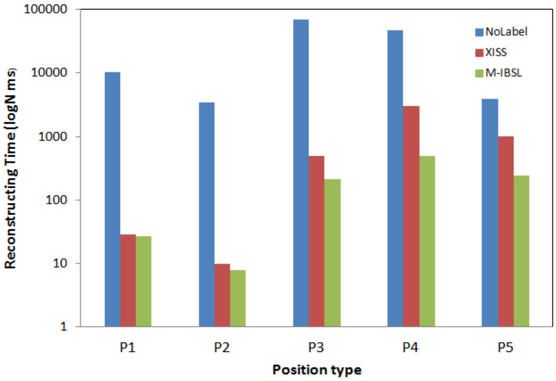

Fig. 3. Reconstructing time of decrypted nodes (0.1MB)

In Fig. 3, M-IBSL outperformed XISS, on all location types for an XML document with 0.1MB of the XMark and DBLP datasets. In DBLP dataset, the number of sibling node increases when the width of the XML document is increased.

The results demonstrate that the number of nodes to be encrypted is related to the number of compared nodes in reconstructing location, and this affects the location reconstructing time. In M-IBSL, a label is not compared with labels of other nodes because a child node is labeled by extending the parent's label to represent the structural information of the XML document. This labeling scheme supports the easy identification of relationships among nodes. In comparing location reconstructing times for the three encryption schemes, the number of encrypted nodes used in searching for a location in the XML document was determined and the location reconstructing time recorded.

5 Conclusion

In this paper, a labeling scheme for the secure dissemination of web contents for users is proposed. When the encrypted nodes are removed from the tree, an adversary can guess the location of an encrypted node from the labeling information. To solve this problem, we presented an enhanced labeling scheme, named M-IBSL, which separates public and private nodes. The results of the experimental study are presented for evaluating the performance of the M-IBSL and XISS. The M-IBSL is superior to XISS in terms of the number of nodes used to reconstruct for the proper location of a node, and the number of nodes used to search for the proper location of a node, and the location reconstructing time.

In the future, we will investigate how to reduce the label size and process the internal node encryption.

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2014R1A1A3051552), South Korea.

References

[1] Bertino, E., Carminati, B., Ferrari, E.: Securing xml documents with author-x. IEEE Internet Computing 5(3), 21–31 (2001)

[2] Bouganim, L., Ngoc, F.D., Pucheral, P.: Client-based access control management for xml. In: Proceedings of the Very Large Data Bases, pp. 84–95 (2004)

[3] Damiani, E., De Capitani di Vimercati, S., Paraboschi, S., Samarati, P.: Securing XML documents. In: Zaniolo, C., Grust, T., Scholl, M.H., Lockemann, P.C. (eds.) EDBT 2000. LNCS, vol. 1777, pp. 121–135. Springer, Heidelberg (2000)

[4] Fan, W., Chan, C.-Y., Garafalakis, M.: Xml querying with security views. In: Proceedings of the ACM SIGMOD, pp. 587–598 (2004)

[5] Geuer-Pollmann, C.: Xml pool encryption. In: Proceedings of the 2002 ACM Workshop on XML Security, pp. 1–9 (2002)

[6] Kunda, A., Bertino, E.: An model for secure dissemination of xml content. IEEE Transactions on Systems Mans and Cybernetics Part C: Applications and Reviews 38(3), 292–301 (2008)

[7] Kuper, G., Massacci, F., Rassadko, N.: Generalized xml security views. In: Proceedings of ACM SIGMOD, pp. 77–84 (2005)

[8] Ko, H.-K., Kim, M.-J., Lee, S.: On the Efficiency of Secure XML Broadcasting. Information Sciences 177(24), 5505–5521 (2007)

[9] Ko, H.-K., Lee, S.: A binary string approach for updates in dynamic ordered xml data. IEEE Transactions on Knowledge and Data Engineering 22(4), 602–607 (2010)

[10] Li, Q., Moon, B.: Indexing and querying xml data for regular path expressions. In: Proceedings of the ICDE, pp. 361–370 (2001)

[11] Steele, R., Min, K.: HealthPass: Fine-grained Access Control to Portable Personal Health Records. In: Proceedings of the AINA, pp. 1012–1019 (2010)

[12] Schmidt, A., Wass, F., Busse, R.: XMark: A benchmark for xml data management. In: Proceedings of the Very Large Data Bases, pp. 974–985 (2002)

[13] Imamura, T., et al.: W3C. Xml encryption and processing (2002)