Preventing Information Leakage in Secure Multiple XML Documents Publishing

Yixiang Ding, Tao Peng, Minghua Jiang
College of Computer Science of Wuhan University of Science and Engineering, Wuhan, 430073, P.R.China
dyx@wuse.edu.cn, pt0403@163.com, jmh@wuse.edu.cn

Abstract

The Prevalent use of XML highlights an increasing need that publishing XML documents should meet precise security requirements, without revealing sensitive information to unauthorized users. We consider data-publishing applications where the publisher specifies what information is sensitive and should be protected. Hiding the sensitive information is no enough and the users can use common knowledge (e.g. "all patients in the same ward have the same disease") to infer more data, which can cause leakage of sensitive information. We formulate the process how users can infer data using three types of common XML constraints and several functional dependencies. We develop a novel paradigm for finding a max partial document without causing information leakage when we publish several related XML documents, while allowing publishing as much data as possible. The experiments on real data sets show that effect of inference on data security, and how the paradigm can prevent leaking the sensitive information.

1. Introduction

XML is rapidly emerging as the new standard data representation and exchange on Internet. As large corporations and organizations increasingly exploit the Internet as a means of improving business-transaction efficiency and productivity, it is increasingly common to find operational data and other business information in XML format. Meanwhile, recent database applications see the emerging need to support data sharing and dissemination in peer-based environments, the owner of a data source needs to publish data to others such public users on the Web or collaborative peers. Generally, the data owner may have sensitive information that needs to be protected. As illustrated by the following example, if we publish data carelessly, users can use common knowledge to infer more information from the published data, causing leakage of sensitive information, especially when there are several XML documents to be published. The public users can combine the several XML documents and infer the sensitive information.

A hospital has XML documents about its patients, departments and physicians. One document in Fig 1 is about its patients and physicians, and the other document about its departments. Each patient has name (represented as pname), suffers from a disease (a disease element), and lives in a ward. Each physician has a name (pname), and treats patients identified by their names. For example, Smith is treating patient Tom, who has leukemia and lives in ward 403.

The hospital plans to provide the data showed in Fig1 to anther department at same hospital to conduct related research. Some data are sensitive and should not be published. For example, hospital does not want the department to know the disease of patient Tom (leukemia) for some reason. A way most prone to think of is to remove the leukemia branch

from the XML document. But if read the document carefully, we can get that Tom, John and Mike are in the same ward, and we can infer that Tom has the same disease (leukemia) with John and Mike. You maybe think that removing the branch W 403 1 or W 403 2 and W 403 3 from the document is an effective way to prevent the information leaking, but, if combine the document showed in fig 1 and the document showed in fig 2, we can also infer the Tom's disease by the branch "Physician1/treat/Tom" in Fig1 and "Physician1/simth" in Fig 2. So, it can not prevent the sensitive data leaking to the relations and common knowledge.

A simple solution to this information leaking is to remove the branch Tom, which makes the users do not know the name. But if we must publish the name of the patients, we must do more. One solution is removing the branch leukemia1, W403 1 and Tom in treat1/Tom; the other solution is removing the branch leukemia1, W 403 2, W403 3 and Tom in treat1/Tom. If we can modify two documents, there are some other ways to prevent the sensitive data leaking by removing some branch in Fig 2.

In general, publishing XML document data with security requirements faces multitude challenges when users can infer data using the common knowledge, especially when the number of documents is more than one, users can infer the sensitive data by combine the multitude documents. The relations between two documents or among more than two documents can be used to infer the sensitive data. The common knowledge can be represented as semantic constraints, which specify relationships that must be satisfied by the nodes. For instance, we can infer the disease of Tom from the ward by the function dependency.

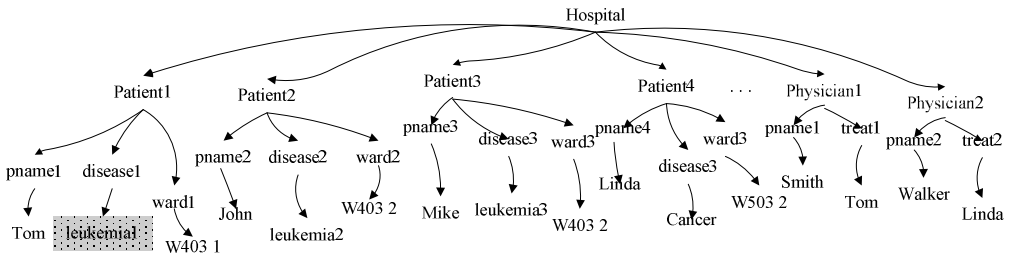

Fig. 1 An XML document of hospital data (the shaded subtree is sensitive data.)

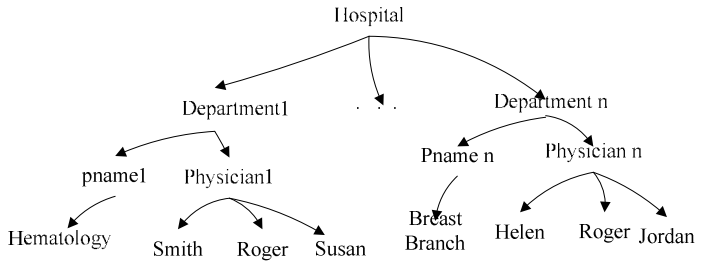

Fig. 2 An XML document of department information

The function dependency is a main problem that we should be considered. The second problem is how we can find out all the relations that the users can be used to infer the sensitive data. At last, we should do is to break the relations are used to infer the data, namely, to compute the partial document to be published without information leakage, even

the users can do inference. There are many partial documents that do not cause the information leakage (such as an empty document), but the purpose of publishing the XML document is to provide information as more as possible. In this paper, we will study these problems and make the follow contributions:

We introduce the process of data inference using the common knowledge represents as semantic XML constraints. When the published documents are more than one, we demonstrate the process of inferring the sensitive data using the relations between two documents.

We purpose an algorithm for computing a partial document to be published without information leaking when the published documents more than one, while to release as much information as possible.

At last, we conduct an experiment using real XML documents to validate the algorithm the paper proposes.

2 Related works

Database security has been studied extensively the past [1, 2, 3], there are also some research about the data publishing [4, 5] and the access control of the database [6].

Because of the prevalence of XML, more and more applications are based on XML; there has been a lot of work on the data security and privacy. Miklau and Suciu [1] show a good diagram (show in Fig 3) to classify different setting for related studies based on trust domains. Consider Fig 3, which classifies scenarios based on trust domains. The trivial case is that of single-user database applications, which have a single trust domain, Fig 3-A, and thus raises no security issues. Most of the applications are based on the scenario B. In scenario B, the data is owned by the server, the server controls the query engineer, but doesn't trust the client asking the queries, and a lot of works focused on how to respond to queries without revealing the sensitive data [7]. Scenario C assumes that the data owner (Client) does not trust the server and a main problem is how to allow the server to answer clients' queries without knowing the exact data.

In the data publishing case, Fig3-D, once the data owner has published the data; she loses the control over it. It can be downloaded, copied, disseminated and redistributed. Both query origination and the execution engine are in different trust domain from the data. So, we must publish the data and hide the sensitive data, at the same time, we must hide the data that can be used to infer the sensitive data with some XML constraints or common knowledge. There have been a lot of works on inference control in relational database [6, 8, 9]. There are also studies on XML security control [5, 10, 11]. Especially, Yang X C [5] studied the common knowledge represented as semantic XML constraints, formulates the process how users can infer data using three types of common XML constraints, and develops algorithms for finding a partial document of a single given XML document without causing information leakage, while allowing publishing a much data as possible. But the schema developed by Yang X C is based on single XML document, and not suit the publishing multiple XML documents.

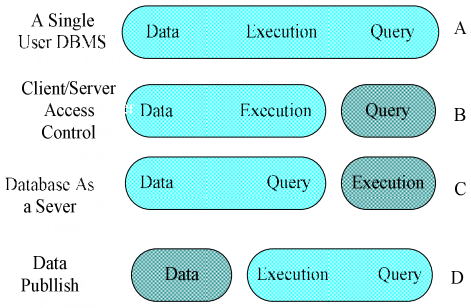

Fig. 3 Different scenarios of database security based on trust domains

3 Data inference based on single document

We view an XML document as tree, when publishing an XML document tree D; some nodes are sensitive and should be protected from common users. In the example showed in Fig 1, the disease name of patient Tom is sensitive and should be protected. The document tree D can be divided into two parts, one is the published part (marked as Dp), the other is the hidden part (marked as Dr). So, we can get the equation Dp=D-Dr namely, the published document is the remain part of the original document D that removing the part should be hidden.

Every node in the document is specified by an XQuery. For simplicity, we represent an XQuery as tree pattern, with two types of edges: (1) a single slash represents an immediate-subelement relationship between a parent and a child (called a "c-child"); (2) a double slash represents a relationship between a node and a descendant (called a "d-child").

We consider the case where users can do data inference using knowledge represented as XML constraints. Such constraints for an XML document specify relationships that must be satisfied by the nodes in the document. XML constraints can be defined using XML schema languages, such as XML DTD, XML Schema. We focus on the following three types of common constraints [5].

A child constraints, represents as $p \to p/p'$, means every node of type p have a child p', marked as C1

A descendant constraints, represented as $p \to p//p'$, means that every node type p must have a descendant of type p'. Marked as C2

A function dependency, represented as $p/p1 \to p/p2$, where p, $p1$ and $p2$ are finite non-empty subsets of paths conforming to the document, it means that for any two substrees $t1$ and $t2$ matching paths $p/p1$, if they have equal values in their $p1$ path, then (1) both of them have non-null, equal substrees that match $p/p2$; or (2) neither of them has a subtree that matches $p/p2$. Marked as C3

With these common constraints, users can infer the results of normal query and get more information, which maybe include the sensitive data. Fig 4 is the example of inference. Fig 4(a) is a normal query about the information of patient John. We can get the information showed in Fig 4(b) from Fig 4(a) with C1 and C2. If users know that Tom lived in W403, they can know the whole information about Tom with C3.

When we publishing the data, the sensitive data should not be leaked, at the same time, we should publish the data as more as possible. So, we must hide the sensitive data and the data that can be used to infer the sensitive data. Yang X C[5] propose an AND/OR Graphs, which used to compute a valid document, the document hide the sensitive data and break all mappings that can be use to infer the sensitive data. The schema Yang X C proposed is a good way to compute the single valid document. But when the published documents are more one, we must consider the relations of all documents.

4 DATA inference based on multiple documents

4.1 process of inference

In section 3, we discuss the data security when publishing a single document. A published document that got by the AND/OR Graphs is considered to be a secure document. But, if there is a document has been published before, and the document has some information related to the secure document, then the secure document maybe become not secure. The users can use the relations the secure document and the pre-published document to infer more information, which can cause the information leaking.

When the schema that Yang X C proposed is applied on the Fig 1, we can get a secure document as Fig 5. We add a constraint that patient's name should not be hidden. In order to remain the structure of the Tree, we use "*****" to replace the nodes that should be hidden.

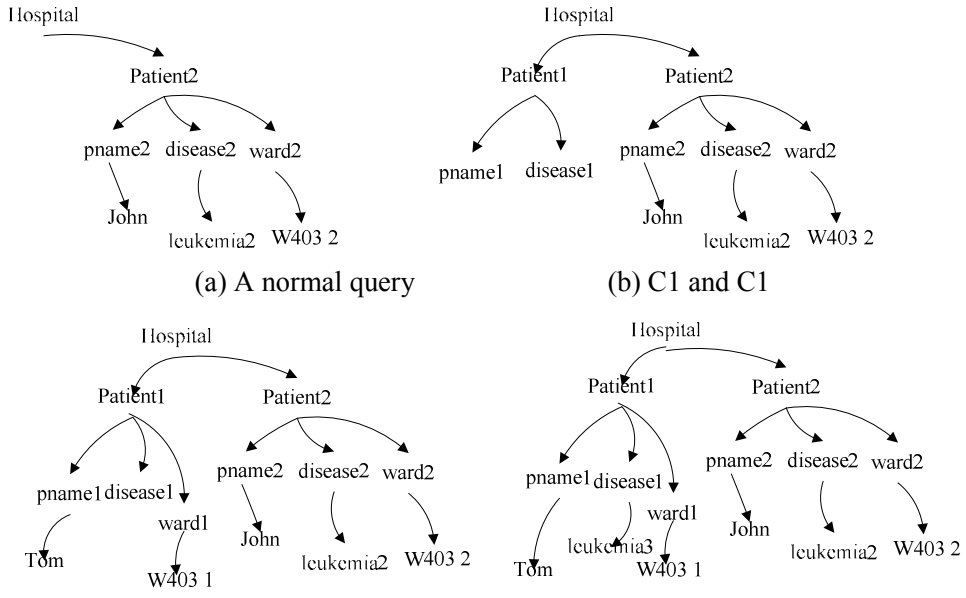

Fig. 4 the process of inference

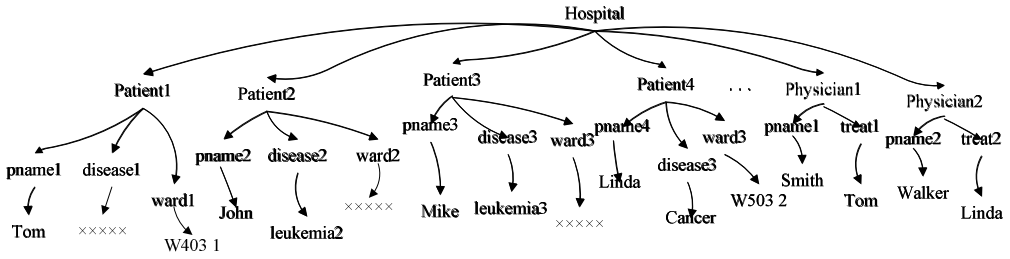

Fig. 5 The secure document computed by AND/OR Graph

To Yang X C's view, the document showed in Fig 5 is a secure XML document, but if the user studies the document carefully, he may get something from hiding such nodes. User will think about the relations among the hidden nodes and get a conclusion that the disease1 must more possible relate to disease2 and disease3. So, there is a flaw in the AND/OR Graph.

In order to protect the sensitive data, a simple and effective way is to hide the information about Tom as much as possible. So, hiding W403 1 is better than hiding W403 2 and W403 3. Fig 6 is the result.

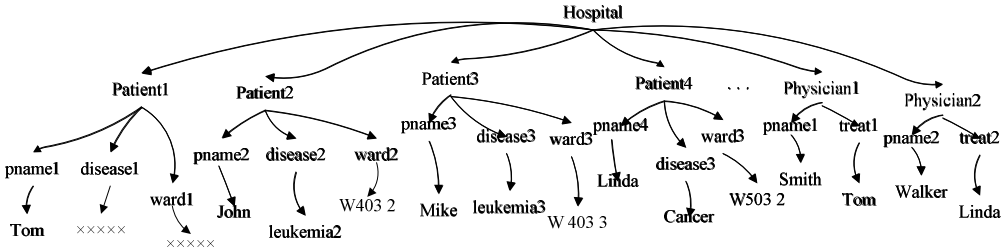

Fig. 6 The improved result

Based on above, we can get a rule:

Rule (1) in order to protect the sensitive information $p//S^*$ (S^* is the sensitive data), the subtree of p that have relations with other nodes should be hidden.

For example, in fig 1, Patient1/disease1/leukemia1 is the sensitive data, and the Patient1/Pname1/Tom or Patient1/ward1/W403 1 should be hidden. Rule (1) means that the inner information is the main source that can be used to infer more information.

With the Rule (1), we can get a single secure XML document. But if the hospital has been published the document showed in Fig 2, user can infer some information with the relation between the document showed in Fig 1 and document showed in Fig 2. User can get that Physical Smith are treating Tom from Fig 1, and Smith belong to department hematology from Fig 2, it is easy to infer that Tom's disease is about hematology and the disease is very serious.

Rule (2) in order to protect the sensitive information $p//S^*$ (S^* is the sensitive data) in a document, the substree in the document out of p relate to the reserved substree of p should be hidden.

For example, if the document D1 showed in Fig 2 has been published, the document D2 showed in Fig 6 is not secure, we must break the relations between the D1 and D2, namely, the node Smith should be hidden.

4.2 Computing a secure document with EIN algorithm

In order to protect the sensitive data, we propose an EIN (Eliminate Inner Nodes) algorithm to compute valid partial document, especially, when the published document is more than one. In section 4.1, there are two rules used to ensure the secure publishing. Rule (1) charge for single document and rule (2) charge for multiple document. The core of EIN algorithm is to hide the inner data as few as possible, which can break the relations with other subtrees in the same document or in published document. We simple the task as Fig 7:

Destination: Protect P//S(S* is the sensitive data)*
Precondition: P//A should be published

Fig. 7 the simple task

The EIN algorithm is to realize the destination with precondition in Fig 7. The algorithm as follows:

(1) Replace the S* with "*****";

(2) If there is an unmarked substrees of the P (marked as P//Ti), exclude the P//A, goto (3), else goto (5)

(3) If there is one or more subtrees, which has the node Ti, in the whole document, replace the Ti in P//Ti with "*****", else, the node Ti should remain, at the same time, marked the subtree P//Ti;

(4) goto (2);

(5) Looking for node A in the document, for every subtree (Qi//A), we should replace the leaf nodes of the other Qi's subtree with "*****";

Step(2). (3) and (4) are used to publish a single secure document, step (5) is used to break the relations with other documents.

With the EIN algorithm, we can get a secure published document is showed in Fig 8.

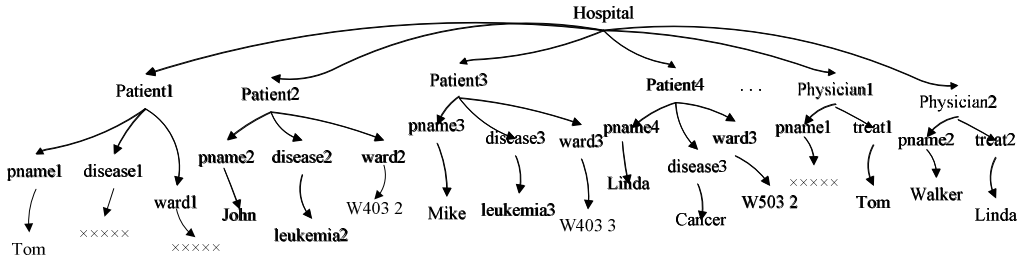

Fig. 8 a secure XML document that can be published

5. Experiments

We conducted experiments to evaluate the effect of data inference on security and the effectiveness of our proposed schema.

Data Sets: there is a real XML data set, which from DBLP. It contained more than 427,000 publications. In this document, the sensitive data is the author who published papers in year

2001, Yang X C [5] analyze the document in detail and apply his algorithm on this document. We give another XML document which includes the Author's information, and the XML document has been published. Comparing to the Yang X C's algorithm, our schema can improve the fault that inferring sensitive data with multiple document.

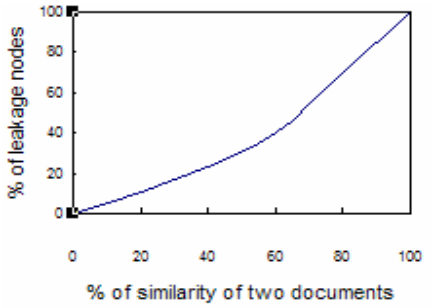

Fig. 9 leakage nodes and the similarity of two documents

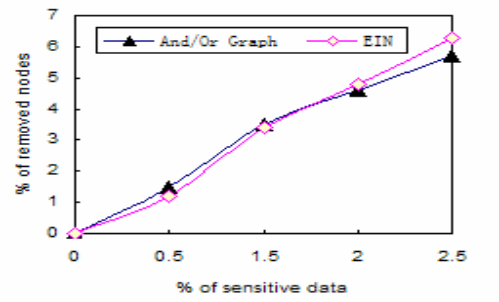

Fig. 10 the compare of two Schemas

From Fig 9, we can get that, with the increasing of similarity of two documents increase, the leakage nodes increase, namely, if two documents have more related context, the possibility of information leakage is increased. From Fig 10, we can know that the removed nodes are increase with the increase of sensitive data, if the sensitive data are small our removed nodes are more then Yang X C's, but when the sensitive data increase,(more than 2%), our removed nodes are more than Yang X C's.

6. Conclusion

In this paper, we studied the effect of data inference using common knowledge (represented as XML constraints) on data security in multiple XML publishing. We analyze the possibility of information leakage when publishing the multiple XML document, and propose a EIN algorithm to protect the sensitive data, which can get a partial document of given XML document without causing information leakage. We gave experiments and compare to the Yang X C's algorithm.

7. Acknowledgments

This paper is supported by Foundation of Wuhan University of Science and Engineering (No: 20073224 and No: 20073205).

8. References

[1] N. R. Adam and J. C. Wortman. Security control methods for statistical databases. ACM computing Surveys, 21(4):515–556, Dec. 1989.

[2] S. Castano, M. G. Fugini, G. Martella, and P. Samarati.Database Security. Addison Wesley & ACM Press, 1995.

[3] D. Denning. Cryptography and Data Security. Addison-Wesley Publishing Co., 1982.

[4] Miklau, G., and Suciu, D. Controlling access to published data using cryptography..In Proceedings of 29th International Conference on Very Large Data Bases (VLDB 2003) (2003), pp. 898-909.

[5] Yang X C,Li C.Secure XML publishing without information leakage in the presence of data inference[C].Proceeding of the 30th VLDB Conference,2004.96-107.

[6] Govind Kabra, Ravishankar Ramamurthy, S.Sudarshan. Redundancy and information Leakage in Fine-Grained Access Control[C]. SIGMOD 2006

[7] J. Biskup and P. Bonatti. Controlled Query Evaluation for Known Policies by Combining Lying and Refusal. In Foundations of Information and Knowledge Systems, pages 49-66, 2002. LNCS 2284.

[8] A. Brodskyand, C. Farkas, and S. Jajodia. Secure Databases: Constraints, Inference channels, and Monitoring Disclosures. TKDE, 12(6):900-919, 2000.

[9] S. Dawson, S. D. C. D. Vimercati, P. Lincoln, and P. Samarati. Minimal Data Upgrading to Prevent Inference and Association Attacks. In PODS, 1999.

[10] E. Bertino, B. Carminati, and E. Ferrari. A Secure Publishing Service for Digital Libraries of XML Documents. In ISC, pages 347-362, 2001.

[11] A.Gabillon and E. Bruno.Regulating Access to XML Documents. In Proceedings of the 15th Annual IFIP WG 11.3 conference on Database Security, July 2001.