International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

A Control Mechanism for Enforcing Inference Usability of XML Documents

Smita Chaudhari[1], Rakesh Rajani[2]

[1] PG Student, Department of Computer Engineering, Alard College of Engineering and Management, Pune India

[2] Professor, Department of Computer Engineering, Alard College of Engineering and Management, Savitribai Phule Pune University, Pune

Abstract: ***Handling of the important information from the XML Document file is the crucial part in the current competitive world as there are more secure transactions in the lot of organization. The most important feature of XML formatting is, it allows for adding schema declarations with integrity constraints to instance data, and to compose individual pieces of data in a tree-like fashion, where a link from a parent node to a sub tree carries some ontological information about the relationship between individual pieces of data. We are proposing an algorithm for weakening of the XML document by eliminating the confidential information, inference capabilities, modifying the schema of XML. The weakened XML document, modified schema conforms to the inference-proof viewed of the generated document to the client.***

Keywords: XML document, XML schema, Inference control, Inference rule, XML data security

1. Introduction

XML is a markup language which is good to store the relevant data and information regarding the projects data or as per the need of the clients. These data may contain the confidential data as well as some secure information regarding some organization which will be very delicate information for that particular organization. While passing this information from one organization to other, there must not be leakage of such information or inference of the data associated with the document.

We have to control a client's options to infer information from observing XML documents, either accessing them directly or indirectly by receiving answers to queries [1]. If a node is sensitive it must be eliminated from the answers to queries. Consider one example XML document generated by a hospital to store the name, diseases, treatments, age, and the names of the doctors for each patient. The hospital wants to share this document with some organizations for some reasons.

So, our main aim is to weaken such XML document file and to make a new inference view of the document to prevent such a confidential data being leaked or bridged by the third party. These are the important parts of our proposed system architecture where it consists of mainly generation of new inference proof view of the XML document.

2. Related Work

2.1 XML Document

With the development of the Web and more usage of the same thing, new data models have started to play a more important role. In particular, XML (eXtensible Markup Language) has emerged as the standard data model for storing and interchanging data on the Web. As more companies adopt XML as the primary data model for storing information, the problem of designing XML databases is becoming more relevant as their needs to be consider the overall structure of the database.

An XML document is a finite, unranked, unordered and labeled tree T = (N_E, A, n_r, child, descendant, f_l; f_s) such that the following conventions and properties hold:

1) NE is the finite set of elements and $n_r \in N_E$ is a distinguished root element. f_l: $N_E \rightarrow \Sigma$ is a function assigning a label to an element.
2) A is the finite set of functions from $N_E \rightarrow \gamma$, where each function a ∈ A has a distinct attribute name from Σ, by the injective function fs: A → Σ Given an element n ∈ NE and a function a ∈ A, if n is associated with with attribute $f_s(a)$, then a(n) is the attribute value, otherwise ∈ I.
3) Child C (N_E x N_E) is a binary relation between a ("father") element and its "children" in the document.

XML Documents and Definitions:

XML Schema

XML Schema specifies the structure to the XML document typically expressed in terms of constraints on structure and content of documents. It defines legal building blocks to the XML document .These constraints are expressed using combination of grammatical rules governing the order of elements. It provides allowable contents and checks for correctness of data. The use of XML schema over DTD is the use of namespace. Namespaces are a mechanism use for breaking up your schemas. Primary defining an XML namespace is to avoid naming conflicts

Path Instance:

Given an XML document T, a path instance φ of T is a sequence of T: child-connected elements in T, i.e., more formally, $\varphi = n_1/n_2/\ldots./n_m$ satisfies the following properties: for all i ∈ [1, m]: $n_i \in T{:}N_E$, and for all i ∈ [1, m-1] : (n_i,n_i+1) ∈ T. child.

In this definition, the functions P and Q serve to confine the possible sub elements and attributes, respectively, of an

element in a conforming document.

Path:
A path is a structure given by ψ = (PN_E, PN_A, child, descendant, pnr, pnl, fe, BackBone) such that the following properties hold:

1. PNE, PNA are finite and disjoint sets of path nodes, $pn_r \in PN_E$ is a distinguished root path node, $pn_l \in (PN_E \cup PN_A)$ is a path node.
2. Child PN_E x (PN_E U PN_A) is a binary relation between a path node and its children, representing a tree.
3. Descendant is the transitive closure of child (PN_E X PN_E).
4. f_e is a function returning the string expression of a path node and satisfying the following constraints:
   a. For each $p_n \in PN_E$, we have $f_e(p_n) = \sigma$, where $\sigma \in \Sigma$.
   b. For each p_n ˜ P N_A, we have $f_e(p_n)$=@ σ # v ,where and σ ∈ Σ and v γ. A path is a structure given by ψ = (PN_E, PN_A, child, descendant, pnr, pnl, fe, BackBone) such that the following properties hold:
5. PNE, PNA are finite and disjoint sets of path nodes, $pn_r \in PN_E$ is a distinguished root path node, $pn_l \in (PN_E \cup PN_A)$ is a path node.
6. Child PN_E x (PN_E U PN_A) is a binary relation between a path node and its children, representing a tree.
7. Descendant is the transitive closure of child (PN_E X PN_E).
8. f_e is a function returning the string expression of a path node and satisfying the following constraints:
   c. For each $p_n \in PN_E$, we have $f_e(p_n) = \sigma$, where $\sigma \in \Sigma$.
   b. For each p_n ˜ P N_A, we have $f_e(p_n)$=@ σ # v ,where and σ ∈ Σ and v γ.

3. Literature Survey

3.1 Inference Control

The main aim of inference control is to prevent an unauthorized client from inferring sensitive information, whether directly or indirectly. It is necessary to control the inference channels in the information set of the system [2]. XML documents have complicated structures than relational database. The purpose of inference control for XML document is to protect the given information. Consider given an XML document with some information assumed to be a prior to a client and a declaration of pieces of information assumed to be very sensitive and thus to be kept confidential, the client can never conclude that the document contains any sensitive information through any inference channel.

3.2 Completeness and Incompleteness:

The first feature distinguishes how a client relates the information set managed by an information system to the actual scenario in the "real world." Assuming fulfillers of the client considers the information set to contain all the information that is correct in the real world, and accordingly holds this information not contained in the information set to be false. However, in many scenarios, a client will restrict himself to assume incompleteness of the document. The information set arranged by the information system contains information considered to be true, while nothing is predicted about information that is not contained in the information set. For instance, the information set given by the XML documents type arranged by an organization, like the one e.g., the company may not know all employee's postal address. In this section, we focus on inference control for XML documents with incompleteness of information, or incompleteness of XML documents.

3.3 Confidentiality Policies

Confidentiality policies tell what is protected by inference control. We protect the fact that an XML document contains some particular pieces of information, and we call those pieces as potential secrets.

3.4 Inference Rules

Inference capabilities are identified using functional dependency. Using this functional dependency the inferring elements are identified and when sending the data to the client potential secret and the inferring elements are hidden. Thus the inference problem is resolved.
For example:
• LHS → RHS
1. {@pid, treatment}®{@dname}
2. {@pid, symptom}®{@dname}
3. {@pid, doctor}®{@dname, treatment}
4. {@pid, address}®{@pname}

Where 'dname' is disease name, 'pname' is patient name and 'pid' is patient id.LHS are inferring field and RHS are potential secret.

4. Proposed System

We propose a formal notion of an inference-proof view of an XML document, to generate the requirements of our goal of effective inference control under the inference capabilities of a client. We enhance the algorithm for generating an inference proof view by using the XML Schema over Document Type Definition. The inputs include an XML document and schema, potential secrets, and the inference capabilities. We also introduce variables into the inference capabilities to increase their high complexity of the algorithm because of the variables.

Advantages of XML schema over DTD
1) The major advantage of schemas is their ability to more strongly type the data in XML documents. Schemas are described using XML instead of the archaic form used by DTDs. Schemas also use to provide a richer approach to describing complex XML types.
2) Document structure can be described more accurately with xml schemas as well, using some features such as minOccurs and maxOccurs to specify the number of times an element can occur within a particular context.
3) The Introduction of Algorithm Complexity Analysis to increase their expressiveness and analyze the high complexity of the algorithm

5. Alternative XML Document

Although the actual stored XML document may contain some potential secrets, the inference control should make the client believe in the possibility that there is an alternative XML document that does not contain any potential secret. In particular, the client should not be able to distinguish the actual document from the alternative.

6. Information Weakening

If the alternative XML document not containing any potential secret is an inference-proof view on the actual XML document, then the answers to the queries do not need to be distorted when these queries are evaluated over the alternative XML document. We use weakening to construct the alternative XML document, that is, all the information contained in the alternative XML document must be contained in the actual XML document, but some pieces of information of the actual XML document might not be part of the alternative XML document

7. Generating an Inference Proof View

In our algorithm for generating an inference-proof view, we will eliminate all the confidential information and some information that can be used to infer some confidential information from an XML document by weakening the document step by step. The inputs of our algorithm include an XML document, the XML schema of the XML document, a finite set of potential secrets, and a finite set of inference rules on the XML document. The outputs of our algorithm include an inference-proof view of the input XML document w.r.t. the two input sets and Xml schema to which both the inference-proof view and the input document conform. The returned XML schema might be different from the input XML schema.

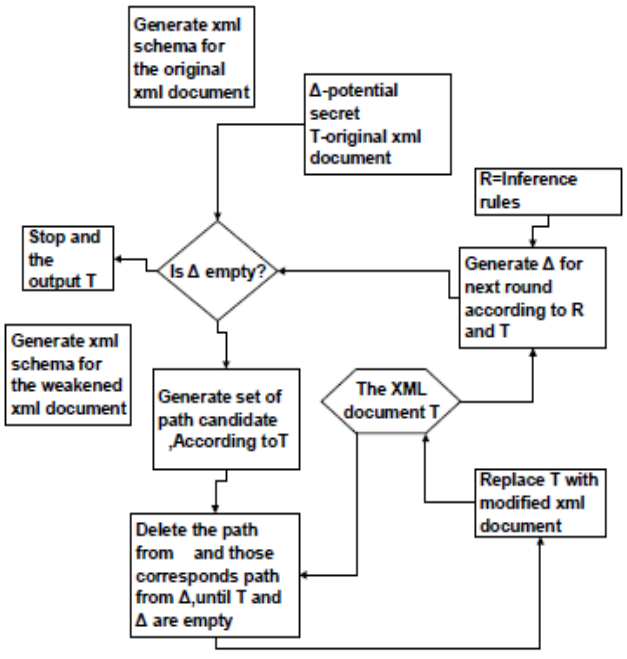

Figure 1: Generation of inference proof view

1. Firstly we are going to work on pure XML document
2. After studying and the finding of the important data (Potential secrets) then goes to step 3
3. Generate the new inference view of the XML document to eliminate the important data.
4. New generated data is nothing but a new XML document (weakened document) without the confidential data.
5. Repeat procedure until no confidential data is available.
6. These new XML are available to the end user clients.
7. This concludes the algorithmic steps where we protect the XML document.

Algorithm:

Procedure Generationofview ():

Input: the set R of inference rules which are going to be checked for the new generation of the XML document, the set S of potential secrets which are being stored in the confidential format, the inference-closed XML document T w.r.t. R, the XML Schema.

Output: the weakened XML document not containing the information which is confidential, the modified XML Schema for the weakened XML document. Ω (notation for the new document).

In our proposed architecture and the algorithm for generating an inference-proof view, we will eliminate all the important information and some information part that can be used to infer means (guess) the some confidential information from an XML document by weakening the document step by step.

1: Δ ← S
2: loop
3: X ← φ
4: for each path set δ ∈ do
5: p←func_obtaincandidate(δ){The generation of path candidates from δ}
6: for each path p ∈ P do
7: proc_insertpath (∈,p) {inserting into fI}
8: end for
9: end for
10: for each path set γ ∈ f do
11: proc_refinpathset (γ)
{
refine γ that should avoid containing more than one path candidate in path set
}
12: end for
13: loop
14: if == φ then
15: break
16: end if
17: Select a path set γ from f with largest cardinality in f I
18: proc_weakendocument (T, D,γ)
{
weaken T for all path in γ and modify D for weaken document
}
19: f←f–{ γ }

20: for each path p ∈ γ do
21: δ←func_findpath (Δ, p) {Finding the path set from which p is generated in }
22: Δ ←1: Δ – {δ}
23: for each path candidate p' (from δ except p do
24: proc_delpath(f,p'){delete p' from I }
25: end for
26: end for
27: end loop
28: for each rule r ∈ R do
29: R' ← fun_enumassignment(r){R' is the set of rules after grounding r }
30: for each r' ∈ R' do
31: if func_violated (T,r')is true then
32: Δ ← Δ U{r'.conditionpart}
33: end if
34: end for
35: end for
36: if == φ then
37: break
38: end if
39: end loop

Modifying XML schema for inference proof XML:

The idea of generating the new XML Schema is to modify some expressions of the original XML schema after every new kind of modification to the content of the XML document. Suppose the algorithm needs to weaken the XML document T for a path and the algorithm will modify some expressions of the original XML Schema D according to the type of ψ.pnl.

8. Conclusion

As per the above context and the overall description regarding theory and algorithm used we have did analysis to generate the new inference view. We are concluding our document with the bottom line as we proposed a very new and the novel technique to improve the transfer of the information between the two organizations where the end user client also cannot bridge the confidential information of the organization from where that information was sent to that particular client.

References

[1] E. Bertino, S. Castano, E. Ferrari, and M. Mesiti, "Specifying and Enforcing Access Control Policies for XML Document Sources," World Wide Web, vol. 3, no. 3, pp. 139-151, 2000.
[2] E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati, "A Fine-Grained Access Control System for XML Documents,"ACM Trans. Information Systems Security, vol. 5, no. 2, pp. 169-202,2002.
[3] L. Li, X. Jiang, and J. Li, "Enforce Mandatory Access Control Policy on XML Documents," Proc. Seventh Int'l Conf. Information and Comm. Security (ICICS), S. Qing, W. Mao, J. Lopez, and G. Wang, eds., pp. 336-349, 2005.
[4] B. Finance, S. Medjdoub, and P. Pucheral, "The Case for Access Control on XML Relationships," Proc. 14th ACM Int'l Conf. Information and Knowledge Management (CIKM), O. Herzog, H.-J. Schek, N. Fuhr, A. Chowdhury, and W. Teiken, eds., pp. 107-114, 2005.
[6] W. Fan, C.Y. Chan, and M.N. Garofalakis, "Secure XML Querying with Security Views," Proc. ACM SIGMOD Int'l Conf. Management of Data, G. Weikum, A.C. Ko¨nig, and S. Deßloch, eds., pp. 587-598,2004.
[7] B. Groz, S. Staworko, A.-C. Caron, Y. Roos, and S. Tison, "XML Security Views Revisited," Proc. 12th Int'l Symp. Database Programming Languages (DBPL), P. Gardner and F. Geerts, eds.,pp. 52-67, 2009.