ONTOLOGY GUIDED XML SECURITY ENGINE

Andrei Stoica (stoica@cse.sc.edu)
and Csilla Farkas * (farkas@cse.sc.edu)
Information Security Laboratory
Department of Computer Science and Engineerieng
University of South Carolina

Abstract. In this paper we study the security impact of large-scale, semantically enhanced data processing in distributed databases. We present an ontology-supported security model to detect undesired inferences via replicated XML data. Our model is able to detect inconsistent security classifications of replicated data. We propose the Ontology Guided XML Security Engine (Oxsegin) architecture to identify data items exposed to ontology-based inference attacks. The main technical contribution is the development of the Probabilistic Inference Engine used by Oxsegin. The inference engine operates on DTD files, corresponding to XML documents, and detects tags that are ontologically equivalent, i.e., can be abstracted to the same concept in the ontology, but may be different syntactically. Potential illegal inferences occur when two ontologically equivalient tags have contradictory security classifications. These tags are marked with a security violation pointer (SVP). Confidence level coefficients, attached to every security violation pointer, differentiate among the detected SVPs based on the system's confidence in an indicated inference.

Keywords: XML security, ontology based inference attack, data aggregation, multilevel XML security

1. INTRODUCTION

Deployment of the eXtensible Markup Language (XML) (XML, 03) enables large scale information sharing and distribution. XML separates data content from display information, facilitating direct access to data. Web ontologies, like DAML-OIL (DAML-OIL, 01) and OWL (OWL, 03), aim to support the development of machine understandable web content. The envisioned Semantic Web (Berners, 01) is based on these technologies to aid intelligent information processing. However, security implications of these new technologies and corresponding safeguards have not been studied sufficiently. In particular, Web inferences that lead to undesired data disclosure need to be analyzed (Thuraisingham, 2002).

An inference channel is a chain of reasoning that leads to protected information based on intentionally released data and metadata. The

* This material is based upon work supported by the National Science Foundation under Grants IIS-237782 and DUE-0112874

aim of this paper is to develop methods for detecting inference channels that result from inconsistent classification of replicated data. Without automated tools for data processing, the security threat from undesired inferences on public databases was low due to the large amount of information needed to be to processed. However, automated information processing, that is independent of software platforms and could access large amount of public data, increases the risk of undesired inferences. To provide high assurance security on the Web, it is necessary that its technologies, e.g., XML and ontologies, have appropriate safeguards.

There are two main research trends in XML security: 1) XML document instance security and 2) Access Control Models for XML databases. In XML document security the focus is on developing technologies to support XML digital signatures (Devanbu et al., 01) and encryption (W3C, 01). XML signature is used to provide authentication and non-repudiation. XML encryption is used to provide confidentiality. XML Access Control research focuses on developing models to manage access to different segments of the XML instances based on security classifications (Bertino et al., 2000; Gabillon, 2000; Damiani et al., 2000; Dridi, 98; Kudo, 00; XACML, 03; Stoica, 02). These works present methods for authorization propagation over security lattice, positive or negative authorizations, conflict resolution, partial document views, and cover stories.

Unexplored challenges in XML security arise from using ontologies to unify and conceptualize XML tag definitions. An ontology, as defined by Gruber (Gruber, 1993), is a specification of a conceptualization, which includes definition of terms used to describe an area of knowledge. An ontology typically contains the following components (Erdmann, 00): a vocabulary of concepts or classes in a taxonomic structure, relationships between concepts, attributes of concepts, and a set of logical axioms that define the true assumptions about the domain. Ontologies unify the different syntaxes and the structure of the documents and may supply background knowledge for query answering (Erdmann, 1999). Large XML document repositories can be managed efficiently by implementing ontology based query engines. Information is retrieved based on its semantics, requiring from the user minimum knowledge about document structure and syntax.

Ontologies can improve the effectiveness of data search by integrating the information from multiple databases using inference rules and concept definitions. Access to XML data is currently implemented using a number of query languages, such as: Lorel for XML (Abiteboul et al., 1997), XQL (Robie et al., 1998) and Xquery (XQuery, 03). One of the research objectives for developing XML query languages is to use ontologies to retrieve information based on the meaning of the

query rather than the exact syntax. Erdmann proposes a framework that enables semantic queries and relieves the user from the burden of knowing the structure of the XML documents (Erdmann, 00). The query engine uses ontologies to derive additional information using a deductive inference system. The ontology is used to provide the hierarchy of domain concepts, the relationships between the concepts, and derivation rules. Several ontology representations approaches, such as Frame Logic (Kifer et al., 1995) and Description Logic (Horrocks, 02), have been developed. In this paper we adopt Frame Logic to represent the ontology class hierarchy.

The research to incorporate ontologies in XML query engines was mainly driven by extending data availability using accurate data inference about the semantics of the data (Amman et al., 2001). However, the same mechanisms may lead to undesired inferences and data aggregations using large collections of data. These security threats are related to inference attacks in traditional databases, where the ontology corresponds to the domain knowledge. Surveys of the inference problem are presented by Jajodia et al. (Jajodia, 95) and by Farkas et al. (Farkas, 02). However, methods that were developed for traditional databases are not applicable on Web data due to flexible data format and large-scale data availability.

To the authors best knowledge, the security problems created by automated correlation of XML documents using ontologies has not been studied yet. In this paper we address some of the security implications of ontology-supported, large scale processing of distributed XML databases. We show that it is possible to use ontologies to mount specific inference attacks using large collections of XML data. We propose a method to detect replicated data with different security classifications, leading to indirect and undesired data accesses. We present an algorithm to detect inference channels, and develop a technique to measure the system's confidence in the detected inferences. Our method can be used to provide automated support for inference detection. The tool can be used by security officers to verify whether a new release would create an inference channel, or to inform the security officer about an exiting channel. In the first mode, it provides protection of sensitive data, while in the second mode it provides detection of a security breach that may have already occurred. Since not all related data items that constitute an undesired inference are under the control of the security officer, e.g., public data of other institutions, the second mode provides valuable information about an existing security breach.

We propose the Ontology guided XML Security Engine (Oxegin) to detect specific types of undesired inferences in a given domain. The search for ontologically equivalent data is supported by semantic

correlation and structural similarities. The probabilistic engine computes security violation pointers (SVP) with an associated confidence level coefficient (CLC). CLC indicates the likelihood of an undesired inference, i.e., the confidence in the semantic similarity of data units detected at conflicting security levels. Parameters supplied by the security officer allow tailoring the complexity of the algorithm depending on the required accuracy and efficiency. For example, the security officer can define whether to work at DTD schema-level or data-level along with the desired levels of abstractions in the ontology.

The rest of the paper is organized as follows: Section 2 gives an example of ontology-guided attacks for XML data. Section 3 outlines the proposed architecture for the ontology guided XML security engine. Section 4 describes the technical details for the inference process of the security engine. Finally we conclude and propose future research in Section 5.

2. ONTOLOGY-BASED ATTACKS IN XML DATABASES

2.1. Replicated Information under Different Classification

Undesired inferences in multilevel database security have been studied extensively. The inference problem is to detect and remove inference channels leading to disclosure of unauthorized data by combining authorized information and meta-data. In traditional databases we may assume that the security officer has complete control over organizational data. Modifying security classifications and redesigning the database prevent, in most of the cases, the unwanted inferences.

The emergence of distributed networks induced a paradigm change in data processing and security needs. Information is often replicated at different sites and copies are protected by systems that operate under different security requirements. This may lead to undesired disclosure of data. A possible solution is to enforce uniform security requirements. For this, automated tools that are able to handle large amount of data and detect syntactically different copies of a data item must be developed.

To illustrate an undesired inference consider the example in Table I. Similar data items from the same ontological domain have different XML structuring tags and are classified differently. Indeed, File 2 is a copy of File 1 with slight modifications. The first column of Table I shows the design specifications of a cryptographic algorithm as part of a project at CryptoTech, Inc. The second column shows the design

notes of the technical consultant John Smith, an associate professor at the University. [1]

Table I. Replicated Information under Different Format and Classification

a. XML File 1 CryptoTech Database	b. XML File 2 John Smith Computer
<?xml version="1.0"?> <cryptoTools> .... **C** <document> .... **C** <titl> CA1059 <title> <author> John Smith <author> <project> P987HY5 <project> </document> </document>	<?xml version="1.0"?> <academic> ..... **P** <paper> ..... **P** <name> CA1059 <name> <writer> John Smith <writer> </paper> </academic>

For further academic research, John Smith made a set of notes regarding the algorithm and assigned them the security label PUBLIC. Shortly after that, realizing the business potential of the algorithm itself, CryptoTech assigns the algorithm design the security label CLASSIFIED. After the security classification is assigned, CryptoTech security officers perform an automated check to ensure that no public copies of the algorithm design exist. They investigate all files of the corporation and the researchers who worked on the project. However, the security check fails to detect the public copies of the desing on John Smith's computer because he used different tags (different DTD file) to represent the same data. Manual analysis of the two files would certainly detect that <document> and <paper> may correspond to the same real world entity. Further, the corresponding data values would establish the relation of the two files. Without human support, automated tools need to rely on an ontology that unifies <document> and <paper> tags to indicate the possibility of replicated data.

To detect this kind of security problem, automated tools should be aware that <author> and <document> tags have the same meaning as <writer> and <paper>, respectively. This type of correspondence cannot be accommodated directly by the XML database and requires an extension to model external knowledge. Ontologies provide class hierarchy to unify related concepts. The next section proposes the design of a stand-alone component that can be incorporated in a comprehensive XML database security engine to prevent replicated inference attacks within a particular knowledge or semantic domain (such as the

[1] Names of the institutes and people are hypothetical.

one described above). The module employs the aid of ontologies for the inference procedures.

3. ONTOLOGY GUIDED XML SECURITY ENGINE

The **O**ntology Guided **X**ml **SE**curity en**GIN**e (Oxsegin)is designed to aid joint analysis of XML documents from a distributed databases to detect undesired inferences. Oxsegin is a probabilistic tool employing ontoligies to unify semantically similar concepts under different syntactic formats. XML document changes are monitored by Oxsegin to detect replicated data with inconsistent security classifications.

3.1. Oxsegin Architecture

Oxsegin basic functionality is described in Figure 1. The XML database is updated through the Input and Feedback Module (IFM). To prevent undesired inferences, new files are checked in correlation with the existing database for replicated information under different syntactic format and security classification.

IFM forwards the XML file intended to be analyzed to the Probabilistic Inference Module (PIM). We refer to this file as the reference file. PIM analyzes the reference file relative to the documents already in the XML database. We refer to the database documents as the test files. The inference process detects replicated information with inconsistent security requirements. The inference uses the concept hierarchy supplied by the Ontology Module (OM). The inference parameters are supplied by the User Defined Inference Parameters Module (UDIPM). These parameters control the precision as well as the complexity of the analysis.

PIM signals to IFM a set of possible security violation pointers and the confidence in the corresponding security breaches if the new file is inserted in the database. Intuitively, a security violation pointer indicates tags from the reference and test files that might contain the same information under contradicting security classification. Each security violation pointer has an associated confidence level coefficient that reflects the confidence in the security breach. The confidence level coefficient is computed based on structural similarity between the XML sub-trees originating from the tags of the security violation pointer and the ontology abstraction levels of the corresponding concepts.

The design of IFM is outside the scope of this paper. When the analysis of the reference file generates an SVP the security officer might change the security labels of the reference file or the security labels of

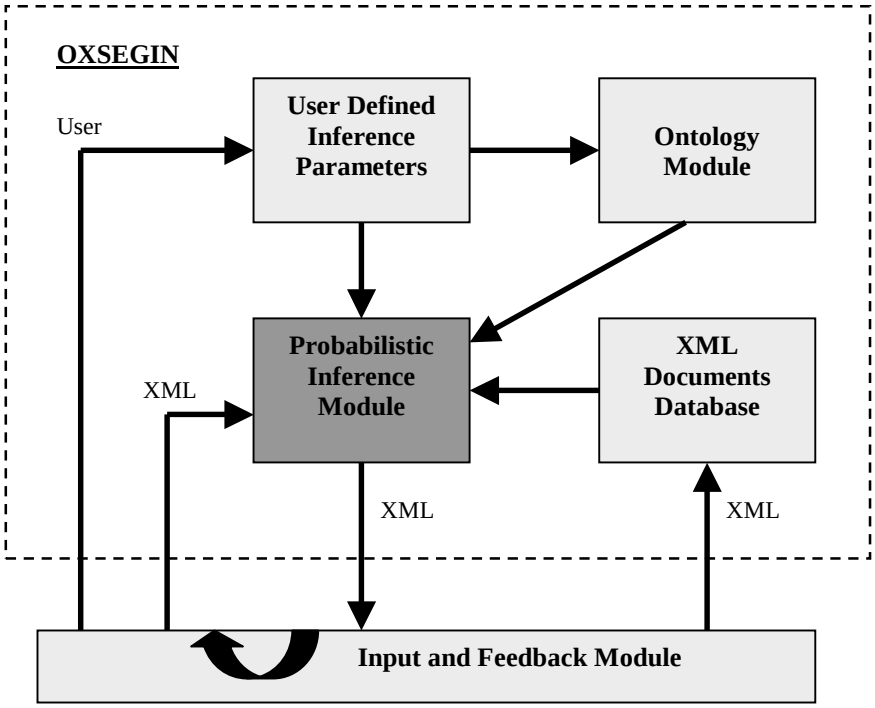

Figure 1. Replicated Information under Different Classification

one of the test files (the files already in the database) to remove the inference channels. Oxsegin can also be used to verify the consistency of these changes. For example, if the security classifications of the reference file are changed, the file is forwarded again to PIM. PIM may detect new inconsistencies based on these modifications. Similarly, if the security classifications of a test file are changed, the file is first removed from the database and then reinserted following the procedures for a reference files (forward to PIM and check security violation). However, changing security classification of a test file may create cascading changes and is not a recommended option.

The XML document database may represent any collection of Web XML files, corresponding to a given Internet domain or a local document repository. Due to its functionality, Oxsegin can also be used to securely publish documents on the web. For this case the reference files are the organizational files selected for publication on the local web site. The test files are the confidential files within the protected, non-public database.

PIM can accommodate different granularity levels for the analysis. The complexity of the analysis is in direct relationship with the accu-

racy of the security violation pointers (reflected by the confidence level coefficients) and also with the processing time.

4. PROBABILISTIC INFERENCE MODULE

The inference engine in the Probabilistic Inference Module (PIM) is based on the Replicated Information Procedure. It assigns security violation pointers to ontologically equivalent data that is under different syntactic form and classified at different security levels. Ontological equivalence of tags established based on the ontology used to guide the inference process.

DEFINITION 1. *(Ontology Class-Hierarchy) An Ontology Class Hierarchy of an ontology O is a directed tree where each node is a concept of O and each edge corresponds to an ISA relation.*

DEFINITION 2. *(Ontological Equivalence) If tags T_i and T_j can be abstracted to the same concept C of on ontology O, we say that T_i and T_j are ontologically equivalent.*

DEFINITION 3. *(Security Violation Pointer) A Security Violation Pointer (SVP) is a pointer to tags T_i and T_j where T_i and T_j are ontologically equivalent and they have different security classifications.*

DEFINITION 4. *(Confidence Level Coefficient) The Confidence Level Coefficient of an SVP over tags T_i and T_j is the confidence of the inference engine in an undesired inference involving the tags T_i and T_j.*

To differentiate between more and less specific concepts in the ontology, the IFM assigns an explicit weight to each concept in the ontology class hierarchy. This is part of the initial setup process. The weights reflect the preference of the inference process towards concepts that are considered more specific than others in the target database. Initial weights assignment is based on how specific the concepts are, e.g. larger weights are assigned to more specific concepts, or based on experimental results. The root of the ontology class-hierarchy has a minimal weight since it is the least specific concept. The lower levels of the ontology, the leaves, usually carry the largest weights. However, there is no direct correspondence between the tree-depth level of a concept in the ontology class hierarchy and the assigned weight. For example, two concepts at the same level in the ontology may have different weights. The weight assignment is domain and task specific. The weights can

be refined based on an iterative process over the quality of the security violation pointers inference.

After IFM assigns the weights for each concept in accordance to the local security policy, the system computes the set of the normalized weights (NWs). When two syntactically different tags are abstracted to a concept C, NW represents the system's confidence that the tags represent the same semantic concept. For example, based on the weight assignment, the confidence in abstracting to concept "person" should be less than the confidence in abstracting to concept "author" since "person" is a less specific concept with a larger domain.

DEFINITION 5. *(Ontological Abstraction Level) Given the concept C of an ontology O, the Ontological Abstraction Level of C, denoted by OAL(C), is the depth of the concept C in the ontology class-hierarchy tree. The root concept RC of the ontology class-hierarchy has OAL(RC) equal 0.*

DEFINITION 6. *(Base Ontological Abstraction Level) The Base Ontological Abstraction Level of an XML tag T, denoted by BOAL(T), is the OAL of the concept C contained within the tag T.*

DEFINITION 7. *(Document Abstraction Level) Given an XML tag T from a DTD tree D, the Document Abstraction Level of T in D, denoted by DAL(T), is the maximum depth of the sub-tree rooted at T. Any leaf tag LT in the DTD tree have DAL(LT) equal 0.*

DEFINITION 8. *(Abstracting a concept N steps) A concept C of an ontology O is abstracted N steps if it is replaced N times by its immediate parent in the corresponding ontology class-hierarchy tree.*

For this formalism, if a file contains the same tag name under different paths from the root, the tags are considered different. One reason is that tags located on different paths from the root might have different DALs resulting in different CLCs for a given SVP. By default, all tags are defined as a pair containing the tag's name and the tag's path information from the root node. For clarity, we omit the path information unless it is needed to differentiate between the tags.

The Replicated Information Procedure uses the ontology class hierarchy tree to abstract XML tags. Tags that can be abstracted to the same concept in the ontology are compared for consistent security classification. An SVP is assigned to every inconsistency found in the security assignment to mark a possible undesired inference. Following the procedures described in detail in the next subsection, each SVP has a CLC. This coefficient is computed based on the normalized weights

of the concepts within the ontology, the relative position of the tags in the XML file structure, and the relative position of the concepts in the ontology class hierarchy tree.

4.1. Replicated Information Procedure

The Replicated Information Procedure abstracts and compares the concepts and their corresponding security labels from two DTD files, using the concepts definitions and hierarchy supplied by the Ontology Module. The reference file is the DTD of the candidate file for XML database update, and the test file is the DTD of a file already in the XML database. Corresponding to XML files in Table I, Table II shows the DTD files and security labels, and Figure 2 shows the associated DTD trees.

Table II. DTD files

A. DTD_r from XML File 1	B. DTD_t from XML File 2
<!ELEMENT cryptoTools (document)$\star$ > **C**	<!ELEMENT academic (paper)$\star$ > **P**
<!ELEMENT document (title, author, project)$\star$ > **C**	<!ELEMENT paper (title, writer)$\star$ > **P**
<!ELEMENT title (#PCDATA)$\star$ > **C**	<!ELEMENT writer (#PCDATA)$\star$ > **P**
<!ELEMENT author (#PCDATA)$\star$ > **C**	<!ELEMENT title (#PCDATA)$\star$ > **P**
<!ELEMENT project (#PCDATA)$\star$ > **C**	

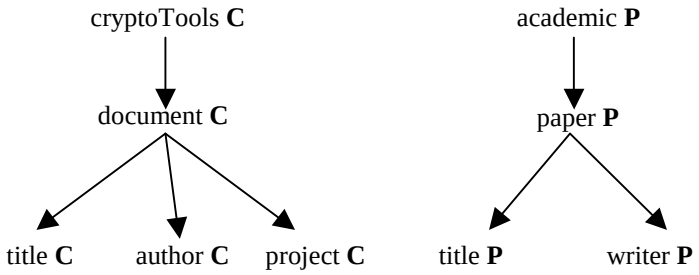

Figure 2. DTD Trees

The Frame Logic statements in Figure 3 represent the ontology associated with the knowledge domain of the DTD files in Table II and Figure 2. Each concept is shown with the associated ontology abstraction level (OAL), weight (WGT), and normalized weights (NW).

To control the number of computational steps, the Replicated Information Procedure abstracts and compares only tags that have DAL less

than a predefined Maximum Document Abstraction Level MDAL. For most of the documents, the relevant data is largely contained at leaves level, allowing MDAL to be small, thus reducing the complexity of the analysis. Also, the concepts are abstracted from the base ontological abstraction level BOAL only a given number of times defined by the Maximum Ontological Abstraction Level (MOAL). MOAL should be small enough to prevent abstracting the tags to highly general concepts, like the root of the ontology, but large enough to allow the system to operate with sufficiently abstract notions.

Object[ ].	OAL=0 WGT=1 NW=1/50
Text :: Object	OAL=1 WGT=5 NW=5/50
Document :: Text	OAL=2 WGT=7 NW=7/50
Paper :: Text	OAL=2 WGT=7 NW=7/50
Person :: Object	OAL=1 WGT=6 NW=6/50
Author :: Person	OAL=2 WGT=8 NW=8/50
Writer :: Person	OAL=2 WGT=9 NW=9/50
Title :: Object	OAL=1 WGT=7 NW=7/50

Figure 3. Domain Ontology

The document analysis has two distinct stages. In stage one the tags from the reference and the test DTD files are abstracted and compared. If there is an ontological equivalence between two tags with different security labels, the corresponding SVP points to the tags in the DTD files. The associated confidence level coefficient CLC is computed based on the DALs of the tags, the OALs, and the NW of the ontology concepts the tags are abstracted to.

In stage two, the CLCs of all SVPs are adjusted based on SVP clusters. Each corresponding CLC of an SVP is relatively adjusted by a composite factor λ in accordance to the local cluster. The λ factor is an average coefficient computed based on the distance to each parent or successor node with an attached SVP and its corresponding CLC. This follows the intuition that there is an increased confidence of a correct ontological equivalence between two tags if their parents or successor nodes in the DTD structure are also ontological equivalent. The factor depends on the distance to these nodes and their corresponding CLC.

Figure 4 gives the formal algorithm for the Replicated Information Procedure. The input of the algorithm consists of the two DTD files (DTDr, DTDt), their corresponding XML instances, and the user defined parameters. The output is a set of (SVP, CLC) pairs.

Identify SVP and calculate corresponding CLC
N = 0
for all tags T_r ∈DTD$_r$ with DAL(T_r)<MDAL **do**
  for all tags T_t ∈DTD$_t$ with DAL(T_t)<MDAL **do**
    for L_r=BOAL(T_r) **downto** (BOAL(T_r)−MOAL) **do**
      Abstract T_r (BOAL(T_r)-L_r) steps
      for L_t=BOAL(T_t) **downto** (BOAL(T_t)−MOAL) **do**
        Abstract T_t (BOAL(T_t)-L_t) steps
        if $T_r \equiv T_t$ and different security labels **then**
          Set SVP$_N$ within DTD$_r$ on T_r and within DTD$_t$ on T_t
          Increase N and compute CLC$_N$
        end if
      end for
    end for
  end for
end for

Adjust CLC for the SVP
for all SVP$_i$ corresponding to T_i ∈DTD$_r$ **do**
  Multiply CLC$_i$ by a composite factor λ_i
end for

Perform data search verification for selected SVP$_s$
if DS **then**
  for SVP$_i$ at T_r ∈DTD$_r$ and T_t ∈DTD$_t$ with CLC$_i$ >DST **do**
    Retrieve all data T_{rk}Data and T_{tl}Data corresponding to tags
    T_r and T_r, from all XML instances corresponding to DTD$_r$, DTD$_t$
    if T_{rk}Data = T_{tl}Data **then**
      CLC$_i$=1.0 (maximum value)
    end if
  end for
end if

Figure 4. Replicated Information Procedure Algorithm

CLC is a function of the tags DALs, and the associated OALs and NWs. Intuitively, if the tags have a similar position in the DTD tree relative to the set of the successor leaves, there is an increase confidence they represent the same abstract concept. Also, equivalence is less likely between the root of a DTD tree and the leaves of another tree. Using the same logic it is more likely that an equivalence originating from tags at the same BOAL, would be semantically correct.

The Replicated Information Procedure computes all possible combination of abstracted tags from the input files. If the number of tags in the reference and test files is n and m respectively, and the ontology depth is k, then the algorithm complexity is $\Theta(n * m)$. The number of computational steps is bounded by $n * m * k^2$.

4.1.1. *Procedure to compute CLC for SVP*
Let SVP point to $T_r \in DTD_r$ and $T_t \in DTD_t$ where T_r and T_t are abstracted to the concepts C. Let NW be the normalized weight of C, DAL(T_r) and DAL(T_t) the document abstraction levels from DTD_r and DTD_t respectively, BOAL(T_r) and BOAL(T_t) the base ontological abstraction levels for the tags T_r and T_t, and OAL(C) the ontology abstraction level of C. We compute CLC of the SVP as:

$$CLC = \frac{1}{max[(BOAL(T_t) - OAL(C)); (BOAL(T_t) - OAL(C))] + 1} * \frac{1}{|BOAL(T_r) - BOAL(T_t)| + 1} * \frac{1}{|DAL(T_r) - DAL(T_t)| + 1} * NW \quad (1)$$

The first factor $\frac{1}{max[(BOAL(T_t)-OAL(C));(BOAL(T_t)-OAL(C)]+1}$ quantifies the number of times the concepts are abstracted until they became ontologically equivalent. The larger the number of abstractions using the ontology, the smaller the confidence that the tags correspond to the same data. The maximum value of 1 is when the concepts are not abstracted using the ontology, in other words the tags contain the same syntactic forms.

The second factor $\frac{1}{|BOAL(T_r)-BOAL(T_t)|+1}$ quantifies the difference between the abstraction levels of the concepts corresponding to the tags of the DTD file. We assume that concepts at different abstraction levels are less likely to lead to correct ontological equivalence. This factor has a maximum value of 1 when the concepts are at the same abstraction level. However, the maximum value for this factor does not necessary implies a correct ontological equivalence.

The third $\frac{1}{|DAL(T_r)-DAL(T_t)|+1}$factor is a rough measurement for the similarity of the sub-trees rooted at the tags T_r and T_t. If the sub-trees rooted at T_r and T_t are similar, this factor has the maximum value of 1. But if T_r and T_t contain different substructures in the corresponding DTD this factor decreases the confidence of a SVP.

4.1.2. *Procedure to compute the composite factor λ for an SVP*
For the adjustment of the CLCs depending of the SVPs clustering in the DTD tree D, the composite factor λ_i for an SVP_i is computed by

averaging the CLCs of the parent and all direct successors SVP of the SVP_i in D. This process is adjusted by a distance factor.

DEFINITION 9. *(Direct Successor SVP_i of SVP_j) Let SVP_i point to the nodes T_i and T_n, SVP_j point to the nodes T_j and T_m in the DTD tree D. SVP_i is a direct successor of SVP_j if T_i is a successor of T_j and there is no SVP_k pointing to the tags T_k and T_l in D such that T_k is a successor of T_j and also an ancestor of T_i.*

For an SVP_i that points to the tag T∈DTD, with SVP_p such that SVP_i is a direct successor of SVP_p, the set $\{\text{SVP}_{k=1\ldots n}\}$ of all direct successors of SVP_i, and d_k the corresponding distance between SVP_i and SVP_k in the DTD tree, we compute the composite factor λ_i as follows:

$$\lambda_i = \frac{\frac{CLC_p}{d_p+1} + \sum_{k=1}^{k=n} \frac{CLC_k}{d_k+1}}{n+1} \tag{2}$$

For an SVP_i that has no direct successor nodes and there is no SVP_p with SVP_i a direct successor of SVP_p, we define the composite factor λ_i, based on the maximum CLC for all SVPs and the depth of the DTD tree, as follows:

$$\lambda_i = \frac{avrgCLC}{depthDTD+1} \tag{3}$$

Composite factors modify the relative difference between the CLCs of all SVPs. The CLCs of the SVPs that are clustered are increased relative to the CLC of an isolated SVP. The closest the SVPs are in a cluster the larger the relative increase factor due to the fact that the distance between the nodes is incorporated in λ. Also the larger the CLCs of the SVPs in a cluster the larger is the relative increase factor because it is more likely to have an ontological equivalence on a node with a parent or a successor indicated by an SVP with a high CLC.

4.1.3. *Data-level Analysis*

The last step of the Replicated Information Procedure is the low-level data granularity search that provides the maximum level of security but also the maximum level of complexity. The IFM controls the balance between complexity and security by either skipping the data search (setting the DS flag), or by adjusting the DST (Data Search Threshold), both within UDIPM. If two data items are the same, the confidence level increases to the maximum value of one. The difference between the low-level data search of the Replicated Information Procedure and a generalized search is that the Oxsegin procedure performs data-level

search only at the security violation pointers and only if the associated confidence level coefficient is over a predefined threshold. The value of the threshold depends on the structure of the ontology used for abstraction as well as the weights assigned to the concepts.

4.1.4. *Example of Replicated Information Procedure*
Figure 5 shows the positions of the SVPs placed by the Replicated Information Procedure (Figure 4) in the DTD files from Table II and Figure 2. The DTD in the left hand side is the reference file (DTD_r) and the DTD in the right hand side is the test file (DTD_t).

If T_r is the <title> tag in DTD_r and T_t is the <title> tag in DTD_t then $\mathrm{T}_r \equiv \mathrm{T}_t$ (the tags are equivalent) with no abstraction in the ontology class-hierarchy. The procedure places SVP_1 on these tags. The tags T_r and T_t are abstracted to the concept "title" (there was no abstraction) with the corresponding $OAL(title) = BOAL(T_r) = BOAL(T_t) = 1$. The normalized weight associated to "title" is $\frac{7}{50} = 0.14$. Within DTD_r and DTD_t the corresponding $DAL(T_r) = DAL(T_t) = 0$. With the above values, $CLC_1 = 0.14$.

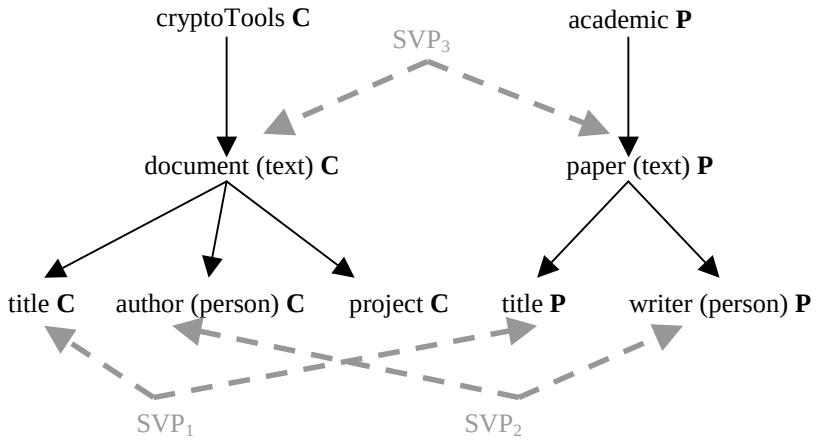

Figure 5. Security Violation Pointers

If T_r=<author> in DTD_r and T_t=<writer> in DTD_t, then $\mathrm{T}_r \equiv \mathrm{T}_t$ if the tags are abstracted to the concept "person". The procedure places on these tags the SVP_2. $OAL(person) = 1, BOAL(T_r) = BOAL(T_t) = 2$, and $DAL(T_r) = DAL(T_t) = 0$. The corresponding CLC_2 is computed as follows:

$$CLC_2 = \frac{1}{max[(2-1);(2-1)]+1} * \frac{1}{|2-2|+1} * \frac{1}{|0-0|+1} * 0.12 = 0.06 \quad (4)$$

After the first phase, the Replicated Information Procedure identifies three security violation pointers with the following attached confidence level coefficients:

$$[SVP_1, CLC_1 = 0.14][SVP_2, CLC_2 = 0.06][SVP_3, CLC_3 = 0.05]$$

The next step is the CLCs adjustments using the composite factors λ. For the <title> and <author> tags in DTD_r indicated by SVP_1 and SVP_2 respectively, there are no direct successor SVP but there is a parent SVP_3 such that SVP_1 and SVP_2 are direct successors of SVP_3. The distance in DTD_r between the tags indicated by SVP_1 and SVP_3, and between the tags indicated by SVP_1 and SVP_3 is 1. As a result, the associated composite factors λ_1 and λ_2 are $\lambda_1 = \lambda_2 = \frac{CLC_3}{1+1} = 0.025$. λ_3 is computed based on CLC_1 and CLC_2 because SVP_1 and SVP_2 are direct successors of SVP_3:

$$\lambda_3 = \frac{\sum k = 1^{k=2} \frac{CLC_k}{d_k+1}}{2} = 0.05$$

After the adjustment phase the relative difference between CLCs is minimized in the cluster represented by the SVP_1, SVP_2, and SVP_3:

$$[SVP_1, CLC_1 = 0.0035][SVP_2, CLC_2 = 0.0015][SVP_3, CLC_3 = 0.0025]$$

The last step of the Replicated Information Procedure, represented by the data level search, is optional. If the data level search is performed with the XML files in Table I, then CLC_1 and CLC_2 are set to 1 (the maximum value) because all tags indicated by the SVP contain similar data. For tags with no data (structuring tags) the CLCs are left unmodified.

$$[SVP_1, CLC_1 = 1.0][SVP_2, CLC_2 = 1.0][SVP_3, CLC_3 = 0.0025]$$

5. CONCLUSIONS AND FUTURE WORK

Investigating the security impact of semantically enhanced XML tools opens a new research area in XML databases security. Ontologies are used extensively in XML-based applications to improve data exchange

in decentralized environments. We showed that these new technologies might lead to undesired data disclosure. Ontologies and semantically enhanced tools can facilitate inference attacks on large, publicly available XML databases.

We proposed an Ontology Guided XML Security Engine architecture to detect specific types of undesired inferences. The inference engine detects inconsistent classification of replicated data. The Replicated Information Procedure uses an ontology aided inference process to identify ontology equivalent information with inconsistent security classification. The probabilistic engine computes security violation pointers and the system's confidence in the corresponding inference. The confidence coefficient measures the structural and ontological similarity between data items with inconsistent security classifications. The system can be used to monitor the information release or to signal possible undesired inferences.

Future work includes refined computation of the confidence level coefficients based on structural similarity between the sub-trees rooted at the nodes in the security violation pointers. We also plan to extend our work to target different types of inferences or data aggregation. For these, we will build a taxonomy of inference threats and develop algorithms to detect such inferences.

Finally, we are building a simulation of the proposed system. We will use this simulation to obtain empirical results and to compare the accuracy of our inferences to the accuracy of a human expert. The simulation will also ensure to fine-tune the parameters used to calculate the confidence coefficient.

Notes

1. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF).

References

Abiteboul, S., Quass, D., McHugh, J., Widom, J., and Wiener, J. The Lorel query language for semi-structured data. *Journal of Digital Libraries. Volume 1, 1997.*

Amann, B., Fundulaki, I., and Scholl, M. Mapping XML Fragments to Community Web Ontologies. In *Proc. Fourth International Workshop on the Web and Databases, 2001.*

Berners-Lee, T., and Hendler, J., and Lassila, O. The Semantic Web. Scientific American, May 2001

Bertino, E., Castano, S., Ferrari, E., and Mesiti, M. Specifying and Enforcing Access Control Policies for XML Document Sources. *WWW Journal, Baltzer Science Publishers, Vol.3, N.3, 2000.*

Thuraisingham, B. XML Databases and the Semantic Web. CRC Press; 1st edition, March 27, 2002

Damiani, E., De Capitani di Vimercati, S., Paraboschi, S., and Samarati, P. XML Access Control Systems: A Component-Based Approach. In *Proc. IFIP WG11.3 Working Conference on Database Security, The Netherlands, August 21-23, 2000.*

DAML-OIL Joint United States / European Union ad hoc Agent Markup Language Committee. http://www.daml.org/2001/03/daml+oil-index.html 2001

Devanbu, P., Gertz M., et al. Flexible authentication of XML documents. In *Proc. of ACM Conference on Computer and Communications Security, 2001.*

Dridi, F., and Neumann, G. Towards access control for logical document structure. In *Proc. of the Ninth International Workshop of Database and Expert Systems Applications, pages 322–327, Vienna, Austria, August 1998.*

Erdmann, M., and Decker S. Ontology-aware XML Queries. http://www.aifb.uni-karlsruhe.de/ mer/Pubs/semantic-xql.webdb00.pdf

Erdmann, M. and Studer, R. Ontologies as Conceptual Model for XML Documents. In *Proc. of the 12-th Workshop for Knowledge, Acquisition, Modeling and Management, Banff, Canada, October 1999.*

Erdmann, M. and Studer, R. How to Structure and Access XML Documents with Ontologies. *Data and Knowledge Engineering, Special Issue on Intelligent Information Integration 2000.*

Farkas, C., and Jajodia, S. The Inference Problem: A Survey. SIGKDD Explorations, Special Issue on Privacy and Security, December 2002 Vol. 4/2, pages 6-12

Gabillon, A., and Bruno, E. Regulating Access to XML Documents. In *Proc. IFIP WG11.3 Working Conference on Database Security, 2001.*

Gruber, T.R. A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition. *Knowledge Acquisition. Vol.6, no.2, 1993. pp199-221.*

Horrocks, I. DAML+OIL: a Description Logic for the Semantic Web. IEEE Data Engineering Bulletin 25(1): 4-9 (2002)

Jajodia. S., and Meadows. C. Inference Problems in multilevel secure database management systems. In Information Security: An ingrated collection of essays, pg. 570-584, IEEE Computer Society Press, Los Alamitos, C.A., 1995.

Kifer, M., Lausen, G., and Wu, J. Logical Foundations of Object Oriented and Frame Based Languages. *Journal of ACM 1995, vol. 42, p. 741-843.*

Kudo, M., and Hada S. XML Document Security based on Provisional Authorizations. In *Proc. of the 7th ACM conference on Computer and Communications Security, Athens Greece, November, 2000.*

OWL Web Ontology Language. http://www.w3.org/TR/owl-ref/ 2003

Robie, J., Lapp, J., and Schach, D. XML Query Language (XQL). In *Proc. of the W3C Query Language Workshop (QL-98), Boston, 1998.*

Stoica, A., and Farkas, C. Secure XML Views. In *Proc. of 16th IFIP WG11.3 Working Conference on Database and Application Security, 2002.*

XACML OASIS eXtensible Access Control Markup Language. http://www.oasis-open.org/committees 2003

XML Extensible Markup Language. http://www.w3.org/XML/ 2003

XQuery XML Query. http://www.w3.org/XML/Query

XML Encryption Requirements. W3C Working Draft, 18 October 2001, http://www.w3.org/TR/2001/WD-xml-encryption-req-20011018.