SECURE XML VIEWS

Andrei G. Stoica and Csilla Farkas

Abstract   Recently more and more data is stored in XML format. While XML increases flexibility, it also raises new security challenges such as access control for multilevel security. This paper considers the problem of generating secure and free of semantic conflicts partial views from XML documents. In the context of DTD-based multilevel security classification, we develop techniques to generate single-level DTDs for partial views. For this purpose, we define and manipulate two graphs, a Minimum Semantic Conflict Graph (MSCG) and a Multi-Plane DTD Graph (MPG). MSCG contains all semantic relationships among the XML tags that must be preserved within any partial view. Intuitively, MSCG ensures the generated views will be free of semantic conflict. MPG captures the structural relationships among tags and their security classifications. We show that secure views can be generated from the first reduced form MPG_0 (i.e., an MPG that does not have edges outside the targeted security space), by ignoring unauthorized security planes. We define a set of procedures to restructure a general MPG into an MPG_0 according to the corresponding MSCG.

Keywords: Multilevel XML security, view-based access control, secure partial views, semantic correctness, structural cover stories, semantic conflict

1.   INTRODUCTION

Semi-structured databases and corresponding query languages have been studied extensively during the last few years [1, 6, 2]. The need for syntactic and semantic interoperation led to the development of more standardized languages such as the eXtensible Markup Language (XML) [10, 5, 11]. Applications that use XML to store data are being wildly used. But some of the XML data is sensitive, requiring as such the development of models and tools to express access control requirements for multilevel security XML documents. Access control models have been proposed by several researchers [4, 3, 7, 9, 8]. The main focus of these works is to assign access permissions (e.g., security classification labels) to XML documents and tags. Currently existing techniques to enforce these requirements, however, are limited and may result in reduce data availability, violations of the document's semantic consistency or may permit illegal inferences.

```
<medicalFiles>
 <countyRec>
  <patient>
   <name>John Smith</name>
   <phone>111-222-3333
   </phone>
  </patient>
  <physician>Jim Dale
  </physician>
 </countyRec>
 <countyRec>
  <patient>
   <name>Mary Gray</name>
   <phone>222-333-4444
   </phone>
  </patient>
  <physician>Joe White
  </physician>
 </countyRec>
 <milBaseRec>
  <patient>
   <name>Harry Green</name>
   <phone>333-444-5555
   </phone>
  </patient>
  <physician>Joe White
  </physician>
  <milTag>MT78</milTag>
 </milBaseRec>
</medicalFiles>
```

```
<medicalFiles>
 <countyRec>
  <patient>
   <name>John Smith
   </name>
  </patient>
  <physician>Jim Dale
  </physician>
 </countyRec>
 <countyRec>
  <patient>
   <name>Mary Gray
   </name>
  </patient>
  <physician>Joe White
  </physician>
 </countyRec>
 <milBaseRec>
  <patient>
   <name>Harry Green
   </name>
  </patient>
  <physician>Joe White
  </physician>
 </milBaseRec>
</medicalFiles>
```

```
<medicalFiles>
   <name>John Smith
   </name>
  <physician>Jim Dale
  </physician>
   <name>Mary Gray
   </name>
  <physician>Joe White
  </physician>
   <name>Harry Green
   </name>
  <physician>Joe White
  </physician>
</medicalFiles>
```

Figure 1. Medical Database XML file(left), Unclassified views(middle and right)

To illustrate the limitations of the previous models in enforcing the access control, consider the XML file and its corresponding security classifications in Figure 1 (left). The file is part of a local hospital database and includes records of the county patients along with records of the nearby military base patients. The hospital policy is to release the name of the patients' physicians for emergency purposes.

Single security level views can be generated from this file by suppressing all nodes outside the permitted security area [9, 7, 3]. Figure 1(middle) shows the Unclassified partial view of the XML document in Figure 1(left). This approach, however, reveals the existence and structural location of data classified at incomparable or higher security levels than the level of the partial view. A different approach may be to collapse the XML structure by connecting the permitted nodes to their nearest and permitted ancestor. While this method conceals the existence of data the user is not permitted to access, it may create semantic conflicts. In Figure 1 (right) the Unclassified partial view of the Medical Files document does not preserve the associations between patients and their physicians

None of the above techniques provide secure and semantically correct decomposition of multilevel security XML documents into single-level views. Methodologies that do not consider the semantic correlation between tags or the illegal inferences lack flexibility and cannot account for all possible scenarios when building partial views. In this paper we propose a new approach, using cover stories and modification of data structure based on semantic conflict analysis, to produce single-level views of the multilevel XML document. We introduce the concepts of Minimum Semantic Conflict Graph MSCG and Multi-Plane DTD Graph MPG. MSCG contains the minimum set of semantic relationships among XML tags that need to be preserved within partial views. Intuitively, it ensures that the generated views will be free of semantic conflicts that can arise with the modification of the data structure (see example in Figure 1 (right)). MPG captures the structural relationships among tags and their security classifications, and is directly derived from the associated DTD. Further, we define the security space as the set of security planes within a partial view. For an MPG without edges outside the targeted security space (MPG_0), we show that it is possible to build a partial view that is free of semantic conflicts. We propose techniques to build MSCG and MPG, and a set of procedures to transform a general MPG into an MPG_0 while preserving the semantic constraints defined in MSCG.

The organization of this paper is as follows. In Section 2 we give the definition for the fundamental concepts of our model. Section 3 contains detailed descriptions of procedures that transform a general MPG into a MPG0. Finally we conclude and recommend future research in Section 4.

2. MINIMUM SEMANTIC CONFLICT GRAPH, MULTI-PLANE DTD GRAPH

This section defines the foundations of the proposed model: the DTD graph (auxiliary to building the multi-plane DTD graph), the Minimum Semantic Conflict Graph (MSCG), the Multi-Plane DTD Graph (MPG) and the Security Space. We assume a DTD-level security granularity paradigm, similar to the one in [4]. In this paradigm, any XML instantiations must follow the corresponding DTD security classifications. The security labels are assigned to the elements and attributes in the DTD. Same tags located under different paths from the root may be classified at different security levels. Figure 2 (left) shows the corresponding DTD tree for the Medical Files database (in Figure 1) with the associated security labels. From the DTD tree we generate the DTD graph as an intermediate step in building the final structure to create partial views. The DTD graph, that is used as a base for generating a multi-plane DTD graph, is build from the DTD tree by adding the tags domain and eliminating redundant information. Figure 2 (right) shows the DTD graph for the DTD
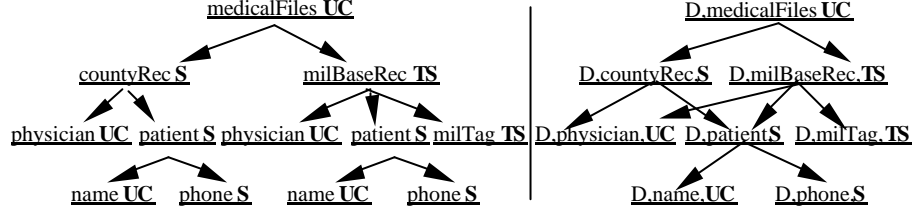
Figure 2. DTD tree (left) and graph (right) for Medical Files Database

tree in Figure 2 (left). Note the tuples (D, patient, S) and (D, physician, UC) are listed once, collapsing the original DTD tree into a graph. The DTD graph incorporates domain information to prevent possible semantic conflicts while manipulating the DTD structure. In this case, we have only a single domain D=Hospital_Patients_Domain. For multiple-domain XML documents, the domain information is used to differentiate tags with the same name but different meaning based on the implicit information of the tag location (a $< name >$ tag under a $< person >$ tag represents a person's name while a $< name >$ tag under a $< pet >$ tag represents the pet's name).

To avoid semantic conflicts in building partial views we introduce the Minimum Semantic Conflict Graph (MSCG). MSCG captures the minimum set of semantic relationships between tags, that needs to be preserved to maintain semantic consistency when building single level views. Intuitively, partial views need to preserve the logical association between data items corresponding to nodes and edges in MSCG. For example, in the Medical Files database, the association between the patients' names and their physicians must be preserved in every partial view to maintain semantic consistency. Therefore, the MSCG must include the corresponding nodes and edge between $< name >$ and $< physician >$ tags.

Definition 1 *(Minimum Semantic Conflicts Graph - MSCG) Let G = (N, E) be a DTD graph, where N is the set of nodes and E is the set of edges. A Minimum Semantic Conflict Graph G' = (N', E') corresponding to G is a connected, undirected graph created as follows: N'$\subseteq$N and given two nodes N_i, N_j $\in$N', there is an undirected edge between N_i and N_j if and only if every instance of the corresponding N_i and N_j tags must be associated with each other under any given partial XML view.*

Each edge in MSCG is assigned a context dependent tag name that is later used for DTD structure modification. This label is employed as a cover story to prevent users from identifying system-generated tags under partial views. If the system needs to generate a parent tag for a pair of nodes to preserve

a semantic correlation (destroyed during security tags restructure) it will use the label of the edge between the nodes instead of generating a new tag. The semantic association between tags is a binary relation that is not transitive. To build an association between three or more tags, all possible pairs of tags in the group need to have a corresponding edge in the MSCG. For example, let (tag_A, tag_B) represent the edge between tag_A and tag_B within the MSCG. To build a semantic association between tag_A, tag_B and tag_C we need three edges: (tag_A, tag_B), (tag_A, tag_C) and (tag_B, tag_C).

MSCG only considers the semantics of the data tags and doing so, the associated security labels have no relevance. Note that from a given DTD graph several different semantic conflcts graphs can be derived. Choosing or ignoring semantic correlation between tags is often domain dependent and is out of the scope of this paper. Building the MSCG translates into deciding the semantic links that need or need not to be preserved, and is a task for a data domain expert or the security officer. To build the MSCG, the security officer decides if there is a semantic conflct between any two tags in the XML files and if there is, includes the nodes and the corresponding edge in MSCG. There is also a tradeoff involved in building an MSCG. The goal is to build a graph that captures all significant semantic relations while maintaining a small size to avoid inducing complex structural changes in the final partial view DTD (note that MSCG is used to guide possible structural security changes). To help constructing MSCG we separate the tags in two categories: data tags and container tags. According to their definition, the container tags are mainly used to structure the XML documents. Since they do not contain direct information, we can consider any semantic correlation involving container tags a weak semantic associations. Using this assumption, MSCG will contain mostly data tags.

Definition 2 *(Container and Data tags) A container tag is an XML tag that holds only structured information in the form of other XML tags and has no tags attributes. A data tags is an XML tag that contains at least one unit of information. A data tag may contain data and container tags.*

Figure 3(left) shows the MSCG for the Medical Files database (Figures 1(left) and Figure 2). Tags $< medicalFiles >$, $< countyRec >$, and $< milBaseRec >$ are container tags, and listing them in any order or combination will not create semantic conflct. Therefore, they will not be represented in MSCG. The patient's $< name >$ and $< phone >$ tags are correlated and need to be represented in MSCG. Intuitively, this means, that if we release a patient's name and phone number for a particular partial view, we also need to preserve the associations between them. A physician is associated to a patient, but $< patient >$ is just a container tag. The semantic association between physician and the patient's phone number is considered only a weak
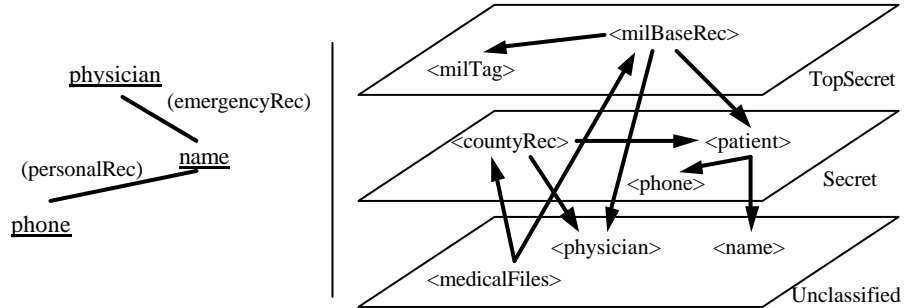
Figure 3. MSCG (left) and MPG (right) for the Medical Files Database

connection. Intuitively, this means, that if we release a physician's name and a patient's phone number, we don't need to preserve the associations between them. As a result, $< physician >$ tag is semantically associated only with the patient's $< name >$ tag.

While MSCG is used to avoid semantic conflicts, the DTD for the partial view is derived from the Multi-Plane DTD graph (MPG). The MPG is the DTD graph restructured on multiple security planes (each corresponding to a security label). Figure 3(right) shows the MPG corresponding to the Medical Files Database DTD in Figure 2. (the domain is ignored since it applies to all tags). Clearly if the MPG is a single-plane graph, that is, the entire set of data is classified at the same security level, providing security views represents a straight-forward problem since the partial view spans the entire document. However, in the presence of cross-plane edges, building secure views may require in some cases structural changes in the DTD structure. Cross-plane edges will be later correlated with the corresponding MSCG edges to build semantically consistent partial views.

Definition 3 *(Security Space) A security space SP associated with a security plane P is the set of all planes dominated by plane P, i.e., SP = { P_i | $P{\geq}P_i$ }.*

Definition 4 *(MPG first reduced form - MPG$_0$) An MPG is in the first reduced form MPG$_0$ for a given security plane P$\in$MPG and the associated security space SP, if and only if there are no directed edges AB or BA such that A$\in P_i$ $\in$SP and B$\in P_j$ $\notin$SP*

Definition 5 *(MPG second reduced form - MPG$_1$) An MPG is in the second reduced form MPG$_1$ if and only if for all directed edges AB such that A$\in P_i$ and B$\in P_j$ then $P_i \leq P_j$*

The partial view corresponding to a given security label SL contains all tags with security labels either SL or dominated by SL. In the context of the multi-plane graph, the set of planes holding the partial view tags defines the security

space corresponding to the label SL. The multi-plane graphs have two reduced forms. The first reduced form MPG_0 defines an isolated security space for a corresponding security label SL, i.e. a set of tags with security labels either SL or dominated by SL, that neither contain or are contained by tags with different security labels. The second reduced form MPG_1 defines a multi-plane graph where all edges are either single-plane edges (contained within a plane) or from a low level to a high level plane in the security hierarchy (the hierarchy relation is relative to the corresponding security labels hierarchy). In both cases, building a partial view is strait-forward and implies just ignoring the nodes and edges outside the targeted security space. For the second reduced form MPG_1, we also ignore the edges outbound of the security space.

Lemma 6 *(Secure Views from MPG_0) Given an MPG_0 for a given security plane $P \in MPG_0$ and the associated security space SP, a single security level view that is secure and free of semantic conflicts can be generated for the security space SP by ignoring all nodes in the planes $P_j \notin SP$ and their associated edges.*

Lemma 7 *(Secure Views from MPG_1) Given an MPG_1, a secure and free of semantic conflicts single security level view can be generated for any security space SP corresponding to a plane $P \in MPG_1$, by ignoring all nodes and edges in the planes $P_j \notin SP$.*

Proof sketch: Cross-plane edges in MPG1 represent edges from a dominated to a dominating security plane and have a descending direction in the DTD tree. They correspond to the case when security labels increase as traversing the DTD structure downwards. Intuitively, ignoring the higher security labels nodes at the lower levels of the DTD creates secure partial views by completely shielding the sensitive information from disclosure and illegal inference.

3. BUILDING SECURE VIEWS

3.1. REDUCING MPG TO THE FIRST REDUCED FORM

For most XML documents, the associated MPG is neither in the first nor in the second reduced form. However, relative to a given security space, it is possible to transform a general MPG into the first reduced form MPG_0 and then create secure single security level views with no semantic conflicts (conform to Lemma 6). Transforming MPG into MPG_0 may require structural changes in the DTD. These changes are made to prevent disclosure as well as inference about data at higher security levels than the level of the current partial view. In some instances new tags are created to provide appropriate cover stories.

The transformation from a general MPG to the first reduced form is based on a set of eight iterative procedures. Algorithm 1 reduces MPG to MPG_0 using the associated MSCG for any security spaces SP_i. Procedure 1 is the only global transformation on MSCG, i.e., it is not relative to a security space. After Procedure 1 and for each SP_i, the algorithm creates a temporary copy of MPG and MSCG. Procedures 2 to 8 are specific to a given SP_i, and applied in sequence to the temporary copies they reduce MPG to MPG_0. Note that, for each security label the reduction to MPG_0 starts from the original MPG and MSCG.

```
Algorithm 1: Reducing an MPG to MPG₀
Input: MPG, MSCG
Output: MPG₀ for all security spaces
BEGIN
  Procedure₁ (MSCG)
  FOR all security labels lᵢ and the associated spaces SPᵢ DO
  BEGIN
    Create MSCG_tmp = MSCG
    Create MPG_tmp = MPG
    FOR j = 2 to 8  DO
    BEGIN
      Repeat
        Procedureⱼ (MSCG_tmp, MPG_tmp, SPᵢ)
      Until no more changes occur
      IF MPG_tmp is in MPG₀ form THEN
        break loop (FOR j = 2 to 8)
    END
    Generate and output MPG₀ by removing from MPG_tmp
    all nodes and edges outside the security space SPᵢ
  END
END
```

3.2.   REDUCTION PROCEDURES

Figure 4 gives brief examples for the preconditions of each procedure along with the proposed solution. The left side represents the Multi-Plane Graph while the right side represents the corresponding Minimum Semantic Conflct Graph.

Procedure 1: global MSCG pruning
Description: removes from MSCG common edges between MSCG and MPG if the edges lie in a single security plane in MPG (Figure 4.a)
Precondition: $\forall P \in MPG$, $\forall A$ and $B \in P$, $AB \in MPG$ and $AB \in MSCG$
Action: remove AB and all single disconnected nodes from MSCG
Proof: if edge e=AB∈MSCG has a corresponding directed edge e'=AB∈MPG such that nodes A and B are in the same security plane, then in any partial view, the tags A and B will be either both included or both excluded from the view according to their (common) security label. If the tags are included, their semantic association is preserved (edge e'∈MPG means tag B will be listed under the tag A) making the edge e∈MSCG redundant.

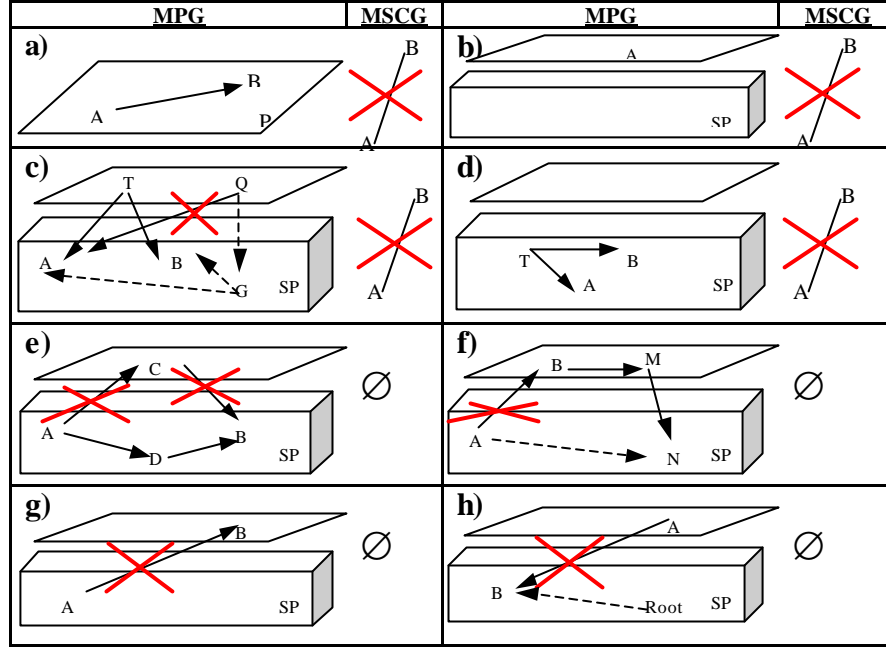**Procedure 2:** MSCG pruning - removes edges relative to the security space

Figure 4. Descriptions of Procedures 1-8

Description: removes from MSCG the corresponding nodes that are outside the given security space in the MPG (Figure 4.b)

Precondition: SP∈MPG, $\forall P_0 \notin$ SP and $\forall A \in P_0$, A∈MSCG

Action: remove A, all edges that contain A, and all single disconnected nodes from MSCG

Proof: the nodes outside the SP are not included in the view; therefore, semantic connection among them or between them and the nodes inside the SP need not be considered. Semantic associations have relevance only for nodes included in the view.

Procedure 3: MPG restructure - generates node

Description: creates in MPG a new parent node for two semantically related nodes that are inside the security space SP to replace a parent node that is outside the SP (Figure 4.c)

Precondition: SP∈MPG, ∃AB∈MSCG, A and B∈SP, AB and BA∉MPG, and for ∀ T parent of A and B, ∃ MN on the path from T to either A or to B such that M∉SP

Action:

1. Generate the tag G=label(AB), edges GA and GB in P, where P=max(P_i, P_j), A∈P_i, B∈P_j

2. Remove edge AB and single disconnected nodes from MSCG

3. $\forall$ Q where QA or QB$\in$MPG replace QA respectively QB with QG

Proof: if tags A and B are semantically associated (AB$\in$MSCG), then the corresponding nodes must have a common parent T within the MPG. While building the partial view, the association between A an B may be destroyed by excluding from the view either T or a node M on the path from T to either A or B. Therefore we generate a new tag in the highest security plane between tag's A and B planes. The generated tag will bare the name of the edge's AB label in the MSCG, providing this way the appropriate cover story. The links to both A and B are also changed to point to the generated tag. This may create redundant data in the view but it prevents possible semantic conflcts if A or B have multiple parents.

Procedure 4: MSCG pruning - removes redundant edges

Description: removes from MSCG edges between nodes that have a common parent T within SP and there is at least one path from T to the target nodes contained in SP (Figure 4.d)

Precondition: SP$\in$MPG, AB$\in$MSCG, $\exists$ T$\in$SP$\in$MPG and $\exists$ a path P_{T-A} from T to A, P_{T-A} $\in$SP and a path P_{T-B} from T to B, P_{T-B} $\in$SP

Action: remove edge AB and single disconnected nodes from MSCG

Proof: if tag A and B have a common parent T inside SP, and there exist at least one path within SP from T to A and from T to B, the semantic association between A and B is preserved in the partial view for the given SP. In this case the edge AB$\in$MSCG becomes redundant and it can be removed without affecting the partial view semantic consistency.

Procedure 5: MPG pruning - removes redundant paths

Description: removes from MPG the paths outside the SP that have a corresponding path inside the SP (Figure 4.e)

Precondition: MSCG=Φ, SP$\in$MPG, $\exists$ A and B$\in$SP, $\exists P_1=\{M_{i=[0...m]}\}$ and $P_2=\{N_{j=[0...n]}\}$ paths from A to B such that $M_0 = N_0 = A$, $M_m = N_n = B$, $\forall$i=[0...m] $M_i\in$SP, $\forall$j=[1...n-1] $N_j\notin$SP, $\forall$i, j tagDomain(M_i) = tagDomain(N_j)

Action: remove all edges from the path $N_{j=[0...n]}$

Proof: if there are two paths from A to B, paths P_1 inside SP and P_2 outside SP, then the DTD contains at least two tags, one inside and one outside SP, with the same type of data (represented by tag B). The partial view will present data available from the outside SP tag structured under the inside SP tag. These structural changes assume all involved tags (on both paths) to be part of a single semantic domain and no semantic restrictions (MSCG = Φ).

Procedure 6: MPG pruning - shortcut paths outside SP

Description: shortcuts in MPG paths outside the security space SP that have

the first and the last nodes within SP, by directly connecting these nodes (Figure 4.f)

Precondition: MSCG=Φ, $\exists$ A$\in$SP, $\exists$ B$\notin$SP, AB$\in$MPG, $\exists$ M_i N_j node sets, $\forall$i M_i $\notin$SP, $\forall$j N_j $\in$SP, M_iN_j $\in$MPG, $\forall$i B parent of M_i , and $\forall$i, j removing M_iN_j disconnects A and N_j

Action: $\forall$ i, j replace M_iN_j with AN_j and remove AB from MPG

Proof: a path P that begins and ends in a given security space SP, with all intermediary nodes outside of SP, corresponds to a tag in SP with only a part of its successors inside the SP. If P is the only path between the parent and the successor nodes and there are no semantic constraints (MSCG=Φ), the nodes along P outside SP can be ignored by pulling the successor nodes directly under their first parent within SP.

Procedure 7: MPG pruning - removes non-cyclic outgoing paths

Description: removes in MPG outgoing edges from a given security space SP if they are not part of a path that begins and ends in SP (Figure 4.g)

Precondition: MSCG=Φ, SP$\in$MPG, $\forall$ A$\in$SP, $\forall$ B$\notin$SP, AB$\in$MPG, and for $\forall$ paths $\{M_{i=[0...n]}\}$ such that M_0=A and M_n $\in$SP then M_i $\in$SP for $\forall$ i

Action: remove edge AB from MPG

Proof: a directed edge AB$\in$MPG with A$\in$SP and B$\notin$SP represents a tag containing information classified at incomparable or higher security levels than the corresponding SP level. If all successors of node A are outside SP, and there are no semantic constraints (MSCG=Φ), removing AB conceals data not included in the partial view (desired effect).

Procedure 8: MPG pruning - removes incoming edges

Description: removes in MPG incoming edges to a given a security space SP, if they are not part of a path that begins and ends in SP (Figure 4.h).

Precondition: MSCG=Φ, SP$\in$MPG, $\forall$ A$\notin$SP, $\forall$ B$\in$SP, AB$\in$MPG, for $\forall$ path $\{M_{i=[0...n]}\}$, if M_n=B then M_i $\notin$SP for $\forall$ i

Action: remove edge AB from MPG and for $\forall$ T$\in$MPG, TB$\notin$MPG create a directed edge from the SP root to B

Proof: an edge from a node A outside the security space SP to a node B inside SP, that is not part of any path that started in SP corresponds to a tag (tag B) containing information allowed in the view but hierarchically structured in the DTD under tags (such as tag A) not included in the view. Tag B will be included in the view under a parent within SP (if such parent exists) or directly under the SP root if B becomes a disconnected node. The SP root is the closest node to the DTD root out of all nodes inside the SP.

3.3.　　MEDICAL FILES PARTIAL VIEWS

In the following, Algorithm 1 is used to build partial views from the Medical Files database. In Figure 3 we have shown the Multi-Plane Graph MPG and respectively, the associated Minimum Semantic Conflict graph MSCG for the Medical Files XML document in Figure 1(left). Algorithm 1 creates a secure and free of semantic conflicts corresponding DTD, which mapped to the XML file, creates the actual view for the end user. For a user with Unclassified clearance, the corresponding security space SP contains only the Unclassified security plane. Procedure 1 generates no changes in MSCG since MSCG and MPG don't share edges between the same nodes. Relative to the given SP, Procedure 2 deletes the $< phone >$ node from MSCG along with the $< phone >$-$< name >$ edge because $< phone >$ tag is classified Secret and the Secret plane is not contained in SP.

Procedure 3 (see Figure 5) generates the first structural change in MPG along with the first cover story. Tags $< name >$ and $< physician >$ are semantically associated but all their common parent nodes are either outside the SP (such as $< countyRec >$) or inside the SP but with all connecting paths containing outside SP edges (such as $< medicalFiles >$). The new tag name (i.e. $< emergencyRec >$) created in the Unclassified plane is not computer generated (users are able to identify computer generated tag names) but instead is taken from the label of the MSCG edge between $< name >$ and $< physician >$. All edges towards $< name >$ (such as from $< patient >$) and $< physician >$ are modified to point towards newly created $< emergencyRec >$ node. After removing the edge between $< name >$ and $< physician >$ from MSCG, there are no semantic associations that may create conflicts in the partial view (MSCG=Φ).

Procedure 6 (see Figure 5) shortcuts the paths that begin and end within SP. $< medicalFiles >$ is a parent node for $< emergencyRec >$ within SP but the connecting paths have outside SP edges. Procedure 6 directly connects $< medicalFiles >$ with $< emergencyRec >$ and removes the edges from the original paths. After executing Procedure 6, MPG is in the first reduced form MPG_0 relative to the security space corresponding to an Unclassified user security clearance.

Figure 5 shows the MPG in the first reduced form for an Unclassified partial views. According to Lemma 6, excluding all nodes outside the security space generates the DTD's for the partial view. The corresponding DTD is mapped to the actual XML data to generate the partial view. Note that in some instances, the DTDs don't match the XML files because of the structural security changes. The partial view is an instantiation of the first reduced form corresponding DTD with data from the XML file.
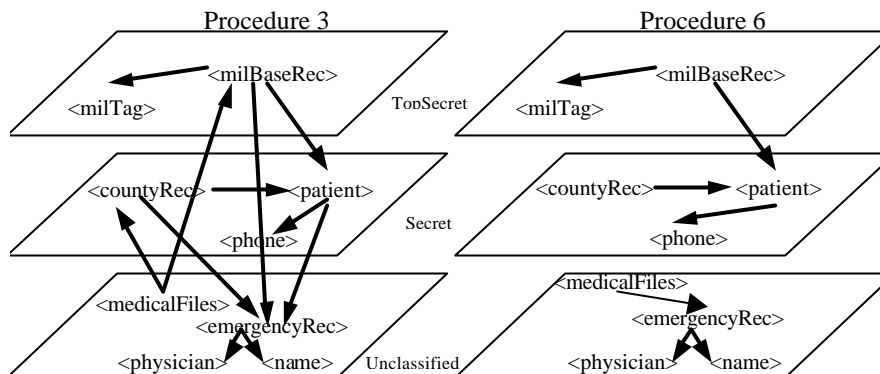
Figure 5. MPG to MPG_0 for Unclassified Clearance

4. CONCLUSIONS AND FUTURE WORK

In this paper we propose a new technique to generate secure and free of semantic conflicts views for multilevel security XML data. We introduce the concepts of Minimum Semantic Conflict Graph (MSCG) and Multi-Plane DTD Graph (MPG). We have defined the MPG_0 and MPG_1 reduced forms to create secure views, and proposed a set of procedures to manipulate a general MPG structure into an MPG_0, while preserving semantic associations defined in the MSCG.

In future research we plan to define and incorporate a Minimum Semantic Constraints Graph (MSCTG) to capture the semantic constraints, rather than the semantic conflicts. MSCTG will include the set of tags that can only be released in conjunction with each other . We will also extend our model for data-level granularity. In this case, the input may be an XML file containing similar tags, on the same path from the root node (in the corresponding DTD), but with different security classifications. MSCG will accommodate and differentiate these type of nodes bonded within a single semantic conflict. We also intend to take advantage of the specific XML constraints, such as tag cardinality, to aid building the MSCG.

14

References

[1] S. Abiteboul. Querying semistructured data. In *Proc. ICDT*, 1997.

[2] C. Beeri and T. Milo. Schemas. Schemas for integration and translation of structured and semistructured data. In *Int'l. Conf. On Database Theory*, 1999.

[3] E. Bertino, M. Braun, S. Castano, E. Ferrari, and M. Mesiti. Author-x: A java-based system for xml data protection. In *Proc. IFIP WG11.3 Working Conference on Database Security*, The Netherlands, August 2000.

[4] E. Bertino, S. Castano, E. Ferrari, and M.Mesiti. Specifying and enforcing access control policies for xml document sources. In *WWW Journal*, volume 3. Baltzer Science Publishers, 2000.

[5] Jon Bosak and Tim Bray. Xml and the second-generation web. *Scientifi c American*, May 1991.

[6] P. Buneman, S. Davidson, G. Hillebrand, and D. Suciu. A query language and optimization techniques for unstructured data. In *Proc. ACM SIGMOD*, 1996.

[7] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. Xml access control systems: A component-based approach. In *Proc. IFIP WG11.3 Working Conference on Database Security*, The Netherlands, August 2000.

[8] A. Gabillon and E. Bruno. Regulating access to xml documents. In *Proc. IFIP WG11.3 Working Conference on Database Security*, 2001.

[9] M. Kudo and S. Hada. Xml document security based on provisional authorizations. In *Proc. of the 7th ACM conference on Computer and Communications Security*, Athens Greece, November 2000.

[10] IEEE Computer Society. Bulletin of the technical committee on data engineering. *Special Issue on XML*, September 1999.

[11] W3C Recommendation. *Extensible Markup Language Language 1.0 specifi cation, http://www.w3.org/TR/2000/REC-xml-20001006*, October 2000.