A Secure Mediator for Integrating Multiple Level Access Control Policies[⋆]

Isabel F. Cruz   Rigel Gjomemo   Mirko Orsini[⋆⋆]

ADVIS Lab – Department of Computer Science – University of Illinois at Chicago
{ifc|rgjomemo|orsinim}@cs.uic.edu

Abstract. We present a method for mapping security levels among the components of a distributed system where data in the local sources are represented in XML. Distributed data is integrated using a semantic-based approach that maps each XML schema into an RDF schema and subsequently integrates those schemas into a global RDF schema using a global as view (GAV) approach. We transform the security levels defined on the XML schema elements of each local source into security levels on the triples of the local RDF schemas, which form a lattice. We show how the merged data in the global schema can be classified in different security classes belonging to the global partially ordered security graph.

1 Introduction

Data interoperation systems integrate information from different local sources to enable communication and exchange of data between them. A common model for these systems involves a global representation of the local data, which acts as a mediator for translating queries and conveying data to and from these sources using the global-as-view (GAV) approach [13]. Semantic Web languages such as RDF Schema (or RDFS) [4] and OWL [16] are particularly suited to represent the global information and to abstract from the particular data formats (relational, XML, etc.) or from the different schemas within the same format, thus addressing respectively problems of syntactic heterogeneity [6] and of structural (or schematic) heterogeneity [17].

We describe a possible scenario in which we consider two healthcare organizations, for instance a health insurance company and a hospital that want to integrate some of their patient data. The data are stored in XML. Figures 1.1 and 1.2 show respectively portions of the XML schemas of the hospital and of the health insurance company. Although data pertain to the same domain, the XML schemas display structural heterogeneities—the element *patient* is contained (nested) in the element *hospital* in one schema, while in the other schema the element *hospital* is contained (nested) in the element *customer*. In reality, the relationship between patients (or customers) and hospitals is "many-to-many" but due to the hierarchical nature of XML such relationships need to be represented using containment.

In addition to structural heterogeneity, our example also illustrates a case of semantic heterogeneity in that two elements that refer to the same concept have different

[⋆] Work partially supported by NSF Awards ITR IIS-0326284 and IIS-0513553.

[⋆⋆] Primary affiliation: Dipartimento di Ingegneria dell'Informazione, Università di Modena e Reggio Emilia, Italy. Work partially supported by MUR FIRB Networked Peers for Business project (http://www.dbgroup.unimo.it/nep4b).

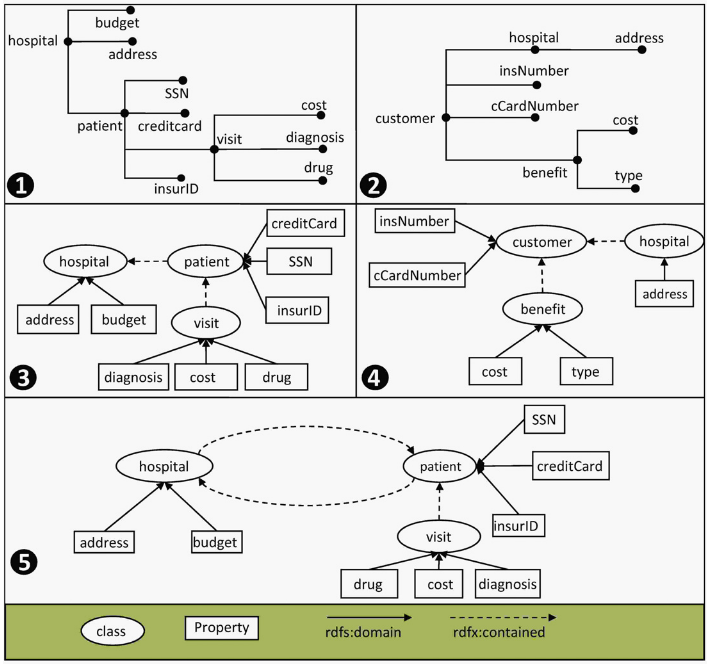

Fig. 1. 1. Hospital XML schema 2. Insurance XML schema 3. Hospital RDF schema 4. Insurance RDF schema 5. Global RDF schema.

names: *patient* and *customer*. In order to overcome syntactic, structural, and semantic heterogeneities, schemas can be integrated at a semantic level. For example, the problem of structural heterogeneities has been addressed in a previous approach [17], where a two-step integration framework is proposed. In the first step, the XML schemas are transformed into RDF schemas. RDF is a language built on top of XML, which can be used to describe relationships between entities. These relationships can be expressed in terms of *triples* of the form (s, p, o). The first element, s, is the *subject* of the triple, the second element, p, is the *predicate* or *property*, and the third element, o, is the *object* or *value* of the property. The subject of the triple is also called *domain* of the property and the object is called *range* of the property. We define a mapping function μ next.

Definition 1. *The mapping function μ maps an XML schema element to an RDF schema element. If v is a complex XML schema element, then $\mu(v)$ belongs to the set of RDFS classes. If v is a simple XML schema element or an attribute, then $\mu(v)$ belongs to the set of RDF properties.*

As shown in Figures 1.1 and 1.3, the complex XML schema element *patient* is mapped to the RDFS class *patient*, whereas the simple XML schema element *creditcard* is mapped to the RDF property *creditCard*. As can be seen in Figures 1.3 and 1.4., the two structurally heterogeneous elements are now mapped to two different classes. A property called *rdfx:contained* is used to record the parent-child relationship between complex XML elements. The second step is that of merging the local RDF schemas into a global schema and it consists of: (1) merging of equivalent RDFS classes and RDF properties from the local sources into a single class or property on the global schema; (2) copying a class or property into the global RDF schema if an equivalent class or property does not exist. A possible global RDF schema is shown in Figure 1.5. Here the local classes *patient* and *customer* have been mapped to the global class *patient*.

The problem that we address in this paper is the security of the interoperation model described above. In particular, if the local schemas are integrated in the global schema, how can the security policy of the global schema be specified taking into account the local security policies?

We adopt a model in which each local organization enforces a multiple level access control model on its schemas [3]. In this model, data are categorized into security levels and users are assigned security clearances. We define a *partial order* or *lattice* $\preceq$ on the set of security levels as follows: given two security levels s_i and s_j, data classified at level s_i can be accessed by anyone with security clearance s_j, such that $s_i \preceq s_j$. The partial order can be represented by a directed acyclic graph. A chain in the graph represents a total order among the security levels along the chain.

The paper is organized as follows. Section 2 presents our security framework, including the *autonomy*, *confidentiality*, and *availability* requirements, the local security lattices and the process in which they are merged to form a global security lattice; we introduce definitions and a theorem that states that the security mappings that need to be established between two local schemas and between a local and a global schema satisfy the requirements. The last two sections, Sections 3 and 4, give a brief overview of related work, of our main contributions, and point to future work.

2 Security Framework

In this section we discuss the process of mapping security levels associated with the elements of the local XML schemas to the global RDF schema triples. The local security policies are represented as local security lattices associated with both the XML and the RDF schema levels. Local security lattices are merged into a global security lattice representing the global security levels associated with the global RDF schema. We assume that the only action that is permitted on the local sources is the *read* action. The results can be extended also to the *write* action, but we assume that users can only write and change the values of the local sources they are associated with. The security of the interoperation systems must satisfy the following requirements:

- *Autonomy*. The local security policies must not be affected by the security policy of the global level.
- *Confidentiality*. Given a security clearance, if a schema element is not accessible locally before the integration, then it must not be accessible after integration.
- *Availability*. Given a security clearance, if a local schema element is accessible before integration, then it must continue to be accessible after integration.

We also make the following assumptions and observations on the local XML and RDF schemas: very sensitive portions of the local XML schemas might not be shared at all; the global level contains the RDF schema, but not the instances (which reside locally). The security levels on the local XML schema elements are used to restrict access to the corresponding XML instance elements.

Definition 2. *A* security specification *on the XML schema tree is a pair* $[v, s]$ *where* v *is a node of the local XML schema and* s *is the security level associated with* v*. We denote the* set of security specifications *by* S_X.

We modify a previously proposed model to specify the security levels globally, which assigns security levels to RDFS triples based on RDF patterns [11]. Instead, we assign security levels to RDFS triples based on XML schema elements.

Definition 3. *A* security object *is a pair* $[t, s]$*, where* t *is an RDFS triple and* s *is the security level associated with* t*. We denote the* set of security objects of a local RDF schema *by* S_L.

We consider two kinds of RDF schema triples: *subject triples* and *subject-object triples*.

Definition 4. *A* subject triple *is an RDFS triple (s, p, o) where the subject s is a mapping* $\mu(v)$ *of an XML schema element* v*, and the predicate p and object o belong to the RDFS vocabulary. A* subject-object triple *is an RDFS triple (s, p, o) where the subject s and object o are two mappings* $\mu(u)$ *and* $\mu(v)$ *of two XML schema elements* u *and* v *which are in a parent-child or containment relationship, and the predicate p is either* rdfs:domain *or* rdfx:contained*.*

For example, *(hospital, rdf:type, rdfs:Class)* is a *subject triple* where only the subject *hospital* is mapped from an XML schema element. The triple *(creditCard, rdfs:domain, patient)* is a *subject-object triple* where the subject *creditCard* and the object *patient* are mapped from XML schema elements. Security levels assigned to *subject triples* will restrict access to information on single entities of the original XML schemas whereas in *subject-object triples* the two elements of the local XML schema may have different security levels. Accordingly, we define two security mappings that associate security specifications on the local XML schemas to security objects on the local RDF schemas.

Definition 5. *A* subject security mapping σ *maps a security specification in* S_X *of the form* $[v, s]$ *to a set of security objects in* S_L*, of the form* $[t, s]$*, such that (1)* t *is a subject triple; (2)* s *is the same security level for all security objects. There are, therefore, as many security objects as there are triples* t *that correspond to XML schema element* v*. A triple* t *can either correspond directly to an element* v *or can be classified by inference using RDFS entailment [11].*

For instance, consider the security specification [*SSN*, *adm*] in Figure 2.1. The subject security mapping σ maps that security specification to security object [*(SSN, rdf:type, rdfs:Class), adm*] in Figure 2.3 and to security object [*(SSN, rdf:type, rdfs:Resource), adm*] containing the entailed triple (because due to inheritance every *class* is also a *resource* in the RDFS model).

Definition 6. *A* subject-object security mapping κ *maps a pair of security specifications* $[v_1, s_1]$ *and* $[v_2, s_2]$ *in* S_X *to a security object* $[t, s]$ *in* S_L*, where* t *is a subject-object triple and the security level* s *is the least upper bound (LUB) of the security specifications levels* s_1 *and* s_2.

Every *subject-object triple* is assigned to the least upper bound (LUB) of the security levels of the corresponding XML schema elements. Instead, the *subject triples* are assigned to the security level of the corresponding XML schema element. For instance,

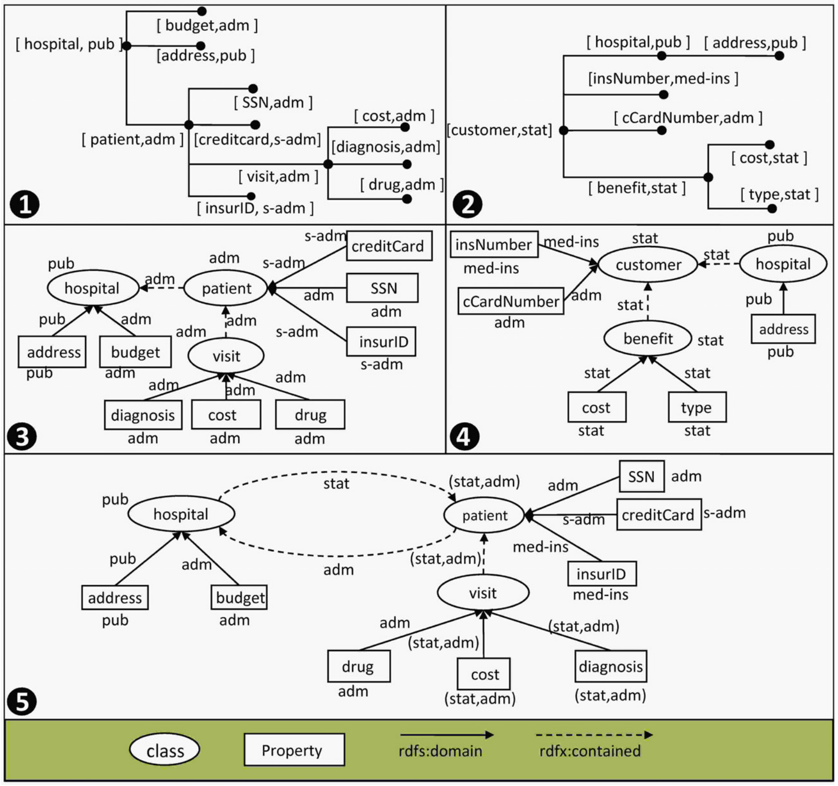

Fig. 2. Security levels and mappings: 1. Hospital XML schema 2. Insurance XML schema 3. Hospital RDF schema 4. Insurance RDF schema 5. Global RDF schema.

consider the security specifications [*hospital, pub*] and [*budget, adm*] in Figure 2.1. where *hospital* and *budget* are in a parent-child relationship. The subject-object security mapping κ maps them to the security object [*(budget, rdfs:domain, hospital), adm*], if $LUB(pub, adm) = adm$. Figure 2 shows the mappings of the security specifications on the XML schemas to the security objects on the local RDFS triples.

Next, we discuss the process of merging the local security lattices into a global security lattice representing the global security levels associated with the global RDF schema, and the classification of the global RDFS triples. The merging process can be carried out by an agreement among the security administrators of the local sources. Some local security levels from different sources may be merged in the global security lattice, while others may be just copied into it. Constraints on the orderings among security levels at the different local sources are used to define the global order. One requirement of the merging is that there are no cycles in the resulting partial order [9, 14]. The partial order $\preceq$ in the local sources must also be preserved in the global security lattice. Therefore, one or more local security levels can be merged into a global security level.

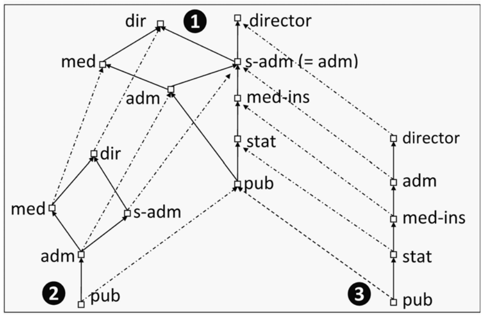

Fig. 3. 1. Global security lattice 2. Hospital security lattice 3. Insurance security lattice.

Definition 7. *The mapping function θ maps a local security level to a global security level. The mapping function Θ maps a set of local security levels, L_i, to a set of global security levels $\Theta(L_i) = \{\theta(l) | \, l \in L_i\}$.*

We show an example in Figure 3 in which the dotted lines represent the mappings defined by θ. The local levels *s-adm (secure administration)* and *adm (administration)* are merged into the global level *s-adm* and $\preceq$ is preserved globally. The classification of the global triples is performed by exploiting the mappings between the triples of the local and of the global RDF schemas and the mappings between the local security levels and the global ones after the merging. A global triple will be assigned a security level by taking into account the security levels of the corresponding triples in the local sources. In the most general case, the local triples mapped to the same global triple will have local security levels mapped to different global security levels. Therefore, there can be more than one candidate security level for a global triple.

Definition 8. *Let S be a subset of the global security levels. The source of S,* source(S)*, is the subset of S such that for each element s_i in $source(S)$ there is no element s_j in S such that $s_j \preceq s_i$ in the graph induced by S. Each element s_i is called* minimal.

In Figure 3.1, $source(\{dir, med,$ *med-ins*$\})$ is the set $\{med,$ *med-ins*$\}$.

Definition 9. *Let S_G be the set of security objects of the global RDF schema and S_{Li} be a set of local security objects in S_L, where the triples in each security object in S_{Li} are mapped to the same global triple t_{g_i}. Let L_i be the set of local security levels of S_{Li}. The* global security mapping *γ maps each S_{Li} to a subset S_{Gi} of S_G, whose elements share that same triple but have as security level one of the security levels in $source(S)$, where $S = \Theta(L_i)$. The cardinality of the set S_{Gi} is the same as the cardinality of $source(S)$.*

For instance, consider two local triples *(cost, rdfs:domain, visit)* in Figure 2.3 and *(cost, rdfs:domain, benefit)* in Figure 2.4. that are mapped to the same global triple t_{g1} = *(cost, rdfs:domain, visit)* in Figure 2.5. The global security mapping γ maps the set S_1 formed by the two local security objects [*(cost, rdfs:domain, visit), adm*] and [*(cost, rdfs:domain, benefit), stat*] to the set S_{G1} formed by the global security objects [*(cost, rdfs:domain, visit), stat*] and [*(cost, rdfs:domain, visit), adm*], because $source(S_1) = \{adm, stat\}$.

Theorem *Assuming security autonomy after source integration, the local security mappings σ and κ and the global security mapping γ preserve data confidentiality and availability.*

Proof Sketch By means of the local security mappings σ and κ, the local security levels are mapped either to themselves (in the case of a subject triple) or to their least upper bound (in the case of a subject-object triple). Given two local security levels l_1 and l_2, we have $l_1, l_2 \preceq LUB(l_1, l_2)$. The global security mapping γ maps a set of local security objects to a set of global security objects where the global security levels are minimal. It may be that a global triple is associated with a global security level $g \preceq LUB(l_1, l_2)$, but due to the security autonomy of the local sources the local triple will remain classified at level $LUB(l_1, l_2)$. Therefore, if an XML schema element cannot be accessed before integration, it will continue to be inaccessible afterwards, thus guaranteeing the confidentiality of the data.

Through the subject security mapping σ, the local security level remains the same, therefore the XML schema element remains available. Subject-object security mapping κ maps two security specifications to a security object, therefore the security level obtained may be more restrictive. This type of mapping deals with the security of the relationship between the subject and the object elements. Even if the relationship is restricted, they can always be accessed individually at the corresponding single triples' security levels. The global security levels obtained by the global security mapping γ are minimal because some local triples are classified at those minimal security levels. Therefore, the minimal global security levels guarantee the availability of the data.

3 Related work

XML Access Control Models XML access control models have been the focus of recent research, including approaches in which the access control model is expressed in terms of tuples that specify who can access which schema element, what type of access is allowed, and how the access rights propagate on the XML tree [2, 7, 8].

RDF/S Access Control Models A method for transforming RDF graphs into trees so as to hide subtrees of a given node has been proposed [12]. Related work includes the work by Farkas and Jain [11] that has been mentioned in Section 2.

Secure Interoperation Models The approach by Pan *et al.* uses a mediator among database systems in an RBAC access control model and mappings between roles in different local sources [15]. In another approach, Candan *et al.* propose a secure interoperation model where a global mediator can enforce global access control rules, or just be a conveyer of the information exchanged between the local sources [5]. Bonatti *et al.* propose the merging of sets of ordered security levels using a logic programming approach [3]. In other work, Dawson *et al.* propose a framework for secure interoperation between local applications mediated by a global application [9]. The work of Farkas *et al.* is the closest to ours [10]. However, they use a "top-down" approach in which they start from the RDF global schema, whereas we start from the XML sources. Another difference is that they use discretionary access rights, whereas we use multiple level security lattices.

4 Conclusions and Future Work

We have proposed a translation model for security levels from local XML schema sources to a global RDF schema. We follow a bottom-up approach and respect the principle of local autonomy in that local security policies continue to be valid. In the future, we will consider the implications of having specifications of security levels not only on the XML schema elements, but also on their instances. We will expand our approach to full XML schemas, including for example IDREF tags. We will also investigate how this approach can be generalized to other data representation models. Furthermore, we plan to incorporate our model into the MOMIS system [1].

References

1. D. Beneventano, S. Bergamaschi, M. Vincini, M. Orsini, and R. C. N. Mbinkeu. Getting through the THALIA benchmark with MOMIS. In *International Workshop on Database Interoperability (InterDB) co-located with VLDB*, 2007.
2. E. Bertino, S. Castano, E. Ferrari, and M. Mesiti. Protection and administration of XML data sources. *Data and Knowledge Engineering*, 43(3):237–260, 2002.
3. P. A. Bonatti, M. L. Sapino, and V. S. Subrahmanian. Merging heterogeneous security orderings. *Journal of Computer Security*, 5(1):3–29, 1997.
4. D. Brickley and R. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. http://www.w3.org/TR/rdf-schema, W3C Working Draft, February 2004.
5. K. S. Candan, S. Jajodia, and V. S. Subrahmanian. Secure mediated databases. In *IEEE International Conference on Data Engineering (ICDE)*, pages 28–37, 1996.
6. I. F. Cruz and H. Xiao. Using a Layered Approach for Interoperability on the Semantic Web. In *Int. Conf. Web Information Systems Engineering (WISE)*, pages 221–232, 2003.
7. E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. A fine-grained access control system for XML documents. *ACM Trans. on Information and System Security*, 5(2):169–202, 2002.
8. E. Damiani, P. Samarati, S. De Capitani di Vimercati, and S. Paraboschi. Controlling access to XML documents. *IEEE Internet Computing*, 5(6):18–28, 2001.
9. S. Dawson, S. Qian, and P. Samarati. Providing security and interoperation of heterogeneous systems. *Distributed and Parallel Databases*, 8(1):119–145, 2000.
10. C. Farkas, A. Jain, D. Wijesekera, A. Singhal, and B. Thuraisingham. Semantic-aware data protection in web services. In *IEEE Workshop on Web Service Security*, 2006.
11. A. Jain and C. Farkas. Secure resource description framework: an access control model. In *ACM Symp. on Access Control Models and Technologies (SACMAT)*, pages 121–129, 2006.
12. S. Kaushik, D. Wijesekera, and P. Ammann. Policy-based dissemination of partial web-ontologies. In *Workshop on Secure Web Services (SWS)*, pages 43–52, 2005.
13. M. Lenzerini. Data integration: a theoretical perspective. In *ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems (PODS)*, pages 233–246, 2002.
14. M. Oliva and F. Saltor. Integrating security policies in federated database systems. In *Annual Working Conf. on Database Security (DBSec)*, pages 135–148, 2000.
15. C.-C. Pan, P. Mitra, and P. Liu. Semantic access control for information interoperation. In *ACM Symp. on Access Control Models and Technologies (SACMAT)*, pages 237–246, 2006.
16. M. K. Smith, C. Welty, and D. L. McGuinness. OWL web ontology language guide. http://www.w3.org/TR/owl-guide/, February 2004.
17. H. Xiao and I. F. Cruz. Integrating and exchanging XML data using ontologies. In *Journal on Data Semantics VI*, volume 4090 of *Lecture Notes in Computer Science*, pages 67–89. Springer, 2006.