Securing Access to Sensitive RDF Data

V. Papakonstantinou, G. Flouris, I. Fundulaki, and H. Kondylakis

Institute of Computer Science - Foundation for Research and Technology, Hellas
{papv,fgeo,fundul,kondylak}@ics.forth.gr

Abstract. Given the increasing amount of sensitive RDF data available on the Web, it becomes critical to guarantee secure access to this content. The problem becomes even more challenging in the presence of RDFS inference, where inferred knowledge needs to be protected in the same way as explicit one. State of the art models for RDF access control annotate triples with concrete values that denote whether a triple can be accessed or not. In such approaches, the computation of the corresponding values for the inferred triples is hard-coded; this creates several problems in the presence of updates in the data, or, most importantly, when the access control policies change. We answer the above challenges by proposing an abstract model where the access labels are abstract tokens, and the computation of inferred labels is modelled through abstract operators. We demonstrate our model through the **HACEA** (**H**ealth **A**ccess **C**ontrol **E**nforcement **A**pplication) that provides simple access control/privacy functionalities in the context of a medical use case.

Keywords: Access Control, RDF, Abstract Access Control Models, eHealth

1 Introduction

The potential of the Web of Data is jeopardized by the fact that many of the datasets published by businesses and organizations worldwide may contain sensitive data, and, consequently, owners may be reluctant to reveal this information, unless they can be certain about the proper enforcement of the desired access rights of different accessing entities to (parts of) their data. Thus, the issue of *securing* content and *ensuring the selective exposure of information* to different classes of users is becoming all the more important. This has led to an increased interest in technologies related to *privacy* and *access control* in the context of the Web of Data. Such technologies will allow datasets with potentially sensitive content to be published, thus bringing the Web of Data to its full potential.

Most state of the art approaches for RDF access control [1,2,3,4] are based on the use of *annotation models* where each triple is associated with a *concrete value*, which is an access label designating whether the triple can be accessed or not. These models assign to the inferred triples (i.e., the ones that have been obtained through RDFS inference [5]) a label computed using *pre-specified* semantics. In these annotation models, a change in the assigned access label of an *explicit* triple would require a complete re-computation of the access labels of all triples

obtained through inference, because there is no way of knowing which inferred triples are affected by said change. If this recomputation is not performed, then the dataset is not correctly annotated, and the system might eventually reveal data, that a requestor is not allowed to access [6].

To tackle the above problem we propose an *abstract access control model* [7,8] to provide secure access to *RDF graphs*. The model is defined by a set of *abstract tokens* and *abstract operators*, which are used to compute the access labels of *inferred* RDF triples. Essentially, our model allows us to record *how* the access label of an inferred triple is computed (rather than just the result of the computation). As a result, the proposed model (contrary to state of the art annotation models), does not commit to a specific assignment of values as access labels of triples, or to a predefined semantics for computing the access labels of the inferred triples (this is similar to *how provenance models* [9], proposed for relational data provenance). Note that, in a medical application context, it is often the case that explicit, fine-grained information need not be disclosed, whereas implicit, coarse-grained information can be accessible without jeopardizing the privacy of the data owner (patient). Therefore, we opted for treating implicit data as first-class citizens with respect to access control.

To demonstrate the use of the proposed abstract access control model, we created the **HACEA** (**H**ealth **A**ccess **C**ontrol **E**nforcement **A**pplication) based on a realistic medical scenario. **HACEA** is built on top of our access control system $A_{bs}ACEF$ and provides simple access control/privacy functionalities in the context of a medical use case.

2 Access Control Enforcement using Abstract Models

In this section we give a brief introduction to the proposed abstract access control model; further details can be found in [7,8]. The model is comprised of *abstract tokens* and *abstract operators*. Abstract tokens encode the accessibility information of explicit RDF triples, and are assigned through *authorisations*. Authorisations are comprised of a *query* and an *abstract annotation token* and assign to all triples in the result of the SPARQL `construct` query the annotation token. The only abstract operator considered in our case is the *binary abstract inference accumulator* operator (denoted by $\odot$), which is used to compute the labels of inferred triples.

We represent *annotated* triples as *quadruples* of the form (s, p, o, l) where s, p, o are the RDF triple's *subject*, *property* and *object* and l is an *access label*. An access label is either an *access token* from the set of abstract tokens, or a *complex expression*; the latter is composed of the tokens and operators that describe *how* the access label of said triple is computed. These expressions are computed *once* (i.e., when triples are loaded in the repository) and are recomputed only when updates (either of the data or the authorizations) occur.

To determine whether a triple can be accessed by a requestor we compute the *actual value* of its associated abstract expression, by means of a *concrete policy*. A concrete policy is composed of a set of *mappings* that assign *concrete* values to

the abstract tokens and operators; these values are used to compute the actual (concrete) value of the associated abstract expression. To determine whether a triple is accessible, an *access function* is defined by the policy and is evaluated on the computed concrete value. Note that each concrete policy is associated with a *requestor* and a *purpose*, to determine the triples that are accessible for said requestor for the defined purpose.

Access control enforcement using abstract models is done as follows: first, during the *annotation phase*, the SPARQL queries of the authorizations are evaluated against the dataset in order to annotate each triple with an abstract access token (producing a set of quadruples). Then, the RDFS inference rules, *extended for quadruples*, are applied to compute the *closure* of this RDF dataset, which includes all inferred triples, along with their access labels, which are complex expressions that use the *inference accumulator* operator.

During the *evaluation phase*, when a requestor specifies a query and a purpose for gaining access to a set of RDF triples that pertain to a specific user, the system selects the concrete policy that matches the request, and computes *on the fly* the concrete value of the triples' access labels; the access function determines whether said triples will be accessed by the requestor (or not).

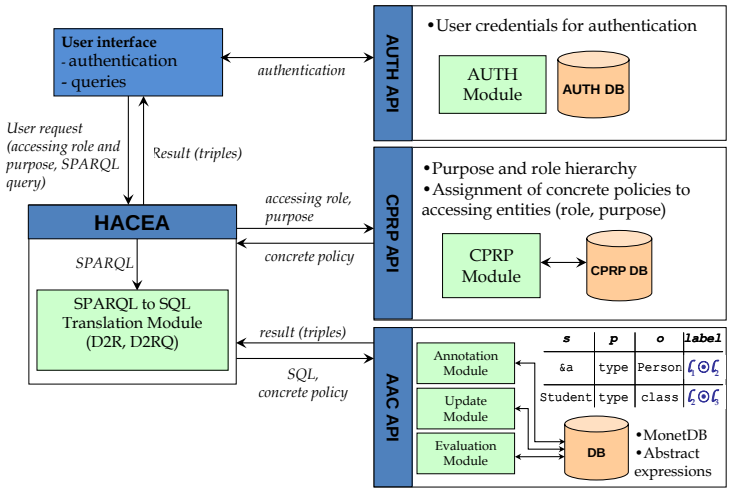

Fig. 1. System Architecture

3 HACEA

Figure 1 shows the architecture of the access control system $A_{bs}ACEF$, on top of which we have build the **HACEA** demo. The system is comprised of the AUTH, CPRP and AAC modules. The AUTH module stores the user credentials and is the module responsible for user authentication. The CPRP module is responsible

for the management of *concrete policies* and is used to associate concrete policies with their corresponding user, requestor and purpose.

Last but not least, the AAC module is the backbone of our system and is comprised of different submodules described below. First, the *Annotation Module*, which is responsible for the *annotation phase* presented in [8]. The *update* module is used for updating annotated quadruples as discussed in [8]. Finally, the *evaluation* module takes as input an SQL query expressed by a requestor, and a concrete policy that matches the request (i.e., requestor, user and purpose which is returned by the CPRP module) and returns the accessible triples. MonetDB[1], a column store RDBMS, is used as the system's backend.

4 Demonstration Scenario

We will demonstrate our access control enforcement approach through **HACEA**. Due to lack of space, we will discuss one representative data access scenario and how it is supported by our demo. A short video[2] and a more detailed description[3] are also available online.

Our demo is based on accessing sensitive patients' information, which is stored in a *Personal Health Record (PHR)*; our demo will be used to allow a patient to authorize a third party (e.g., doctor, nurse, public or private entity) to access to her data through a *consent form*.

Our main example scenario assumes a *public service* (namely, Breast Cancer Action Fund – BCAF) which provides funding to cancer patients. Such a service would require access to the patient's PHR in order to verify that the patient has a malignant tumour indicating breast cancer and provide the benefit. Thus, in order to get a discount, an applying patient (say, Emily Robinson) should allow access to her records by signing the corresponding *consent form*. Such a consent form consists of all the parts of her data, as her menopausal state, her pregnancy state etc., which can be selected for release. However, BCAF is not interested in other information about the patient, such as her pregnancy status, diseases the patient may have or had in the past etc. Moreover, BCAF is not interested in knowing the type of tumour, its stage, its size, the current treatment or other detailed information. Therefore, Emily does not need to disclose fine-grained information on her status (e.g. the exact type of her tumor) but can provide more coarse-grained information (e.g., her tumor malignity) as this is enough for the purposes of the accessing entity; the latter (coarse-grained information) is essentially implicit information, which motivates the need for treating implicit data as first-class citizens with respect to access control. Figures 2 and 3 show the consent form that the applying patient fills in, and the query that BCAF is using in order to obtain access to patient's data.

The dataset that we used for demonstrating **HACEA** consists of a set of 10.000 patients and their corresponding health records, created by the Advanced

[1] `http://www.monetdb.org`
[2] `http://youtu.be/-wYbiWvTfyE`
[3] `http://planet-data.eu/sites/default/files/PD_WhitePaper_HealthCare.pdf`

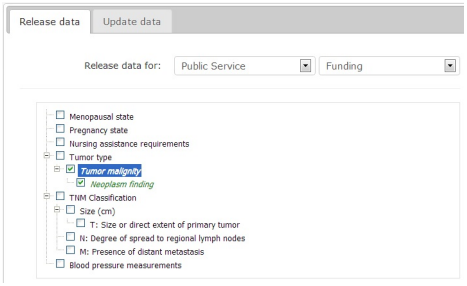

Fig. 2. Releasing data

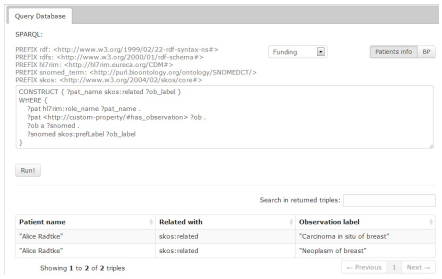

Fig. 3. Querying data

Patient Data Generator (APDG) tool[4] . The APDG is a generator developed in the context of the EU project EURECA which uses clinical and epidemiological background knowledge to generate a set of realistic patient records. The used dataset is represented in RDF format and expressed according to the HL7-RIM[5] schema and SNOMED-CT[6] terms both of which are well-established medical ontologies. This allows the uniform expression of the data using well-defined and commonly accepted terminologies.

Acknowledgments

This work was partially supported by the EU projects PlanetData (FP7:ICT-2009.3.4, #257641) and p-Medicine (FP7-ICT-2009.5.3, #270089).

References

1. Abel, F., Coi, J.L.D., Henze, N., Koesling, A.W., Krause, D., Olmedilla, D.: Enabling advanced and context-dependent access control in RDF stores. In: ISWC. (2007)
2. Dietzold, S., Auer, S.: Access control on RDF triple store from a Semantic Wiki perspective. In: SFSW. (2006)
3. Jain, A., Farkas, C.: Secure resource description framework. In: SACMAT. (2006)
4. Kim, J., Jung, K., Park, S.: An introduction to authorization conflict problem in RDF access control. In: KES. (2008)
5. Brickley, D., Guha, R.: RDF Vocabulary Description Language 1.0: RDF Schema. `www.w3.org/TR/2004/REC-rdf-schema-20040210` (2004)
6. Knechtel, M., Peñaloza, R.: A generic approach for correcting access restrictions to a consequence. In: ESWC. (2010)
7. V. Papakonstantinou, M Michou G. Flouris, I.F., Antoniou, G.: Access control for RDF graphs using abstract models. In: SACMAT. (2012)
8. Papakonstantinou, V.: Controlling access to RDF data using abstract models. Master's thesis, University of Crete, Computer Science Department (2013)
9. Green, T.J., Karvounarakis, G., Tannen, V.: Provenance semirings. In: PODS. (2007)

[4] `wasp.cs.vu.nl/apdg`
[5] `www.hl7.org/implement/standards/rim.cfm`
[6] `www.nlm.nih.gov/research/umls/Snomed/snomed_main.html`