Testing the Inferability of RDF-Data for Secure Partial Encryption

Mark Giereth and Ying Qian
Institute for Intelligent Systems, University of Stuttgart, Germany
giereth@iis.uni-stuttgart.de, qianyg@studi.informatik.uni-stuttgart.de

Abstract

Partial encryption of RDF data allows a fine-grained protection of sensitive information. But by using ontologies and inference mechanisms encrypted data might be deduced. Concerning the security of encrypted data, it is crucial to know whether they are inferable. In this paper, a general approach is presented for testing the inferability of RDF-data to be encrypted.

1 Introduction

Security will play an important role in the emerging Semantic Web. In order to protect sensitive information, there are mainly two approaches. One is to control the data access (as proposed in [Weitzner *et al.*, 2004]), the other is to encrypt the data. The second approach is focused in this paper. Although it is possible to encrypt all the data contained in a Semantic Web document it is more flexible to encrypt only the sensitive parts enabling agents to use the non-encrypted parts. A first approach to partially encrypt Semantic Web documents represented in RDF is described in [Giereth, 2005].

Semantic Web applications typically make use of ontologies. An ontology formulates a strict conceptual scheme about a domain containing the relevant classes, properties, instances, datatypes, cardinalities, etc. Additionally, ontologies can be used to infer implicit knowledge about a domain – but also for inference attacks on encrypted data which might cause security problems.

This paper presents a first study of the role of ontologies in a partial RDF encryption (PRE) process and describes a prototype for checking the inferability of data to be encrypted.

2 Inferability Testing

As a first step, the predefined axioms and entailment rules of RDFS [Brickley and R.V. Guha, 2004] and OWL [McGuinness and Harmelen, 2004] have been evaluated. Table 1 shows the numbers of predefined entailments rules for each category. These predefined rules and axioms can be combined and cascaded in order to derive implicit knowledge contained in an underlying ontology.

relation	category	RDF/ RDFS	OWL Lite	OWL DL	OWL Full
class/ class	hierarchical	2	6	7	7
	equivalent	0	7	9	9
	disjoint	0	0	1	1
class/ instance	type	3	10	11	11
property/ property	hierarchical	2	1	1	1
	equivalent	0	2	2	2
property/ instance	attribute -value	1	11	12	12
instance/ instance	same	0	5	6	6
	different	0	1	2	2
axioms		72	124	140	142

Table 1: Predefined Entailment Rules and Axioms

2.1 PRE-Process Oberview

Figure 1 gives an overview of the PRE-process and shows when inferability testing is be done. The testing module takes an ontology model (including all instances) and an encryption policy as input.

Encryption policies (cf. [Giereth, 2005]) provide a dynamic way to specify *which* data fragments to encrypt and *how* to encrypt them (we only look at the *which* aspect here). Each encryption policy defines a set of queries and corresponding encryption descriptions. The queries are modified SPARQL queries [Prud'hommeaux and Seaborne, 2004] in which the binding mechanism has been changed. Instead of returning a set of variable bindings, ordered triple sequences are returned which are bound to their corresponding query patterns. Each pattern is referenced by its position in the `WHERE`-clause. Based on the query result a corresponding encryption description specifies the data fragments to be encrypted and how to encrypt them. Example:

```
SELECT * FROM ... WHERE
  (?x <cc:cardNumber> ?y)  (?x <rdf:type> <cc:CreditCard>)
USING cc FOR <http://example.de/creditCardOnt#>

ENCRYPT [o,1] [t,2] ...
```

The example shows a query and a simple encryption description (keys and encryption metadata are ignored) for encrypting the card number of all `CreditCard`-instances. The `WHERE`-section defines two patterns which are referenced in

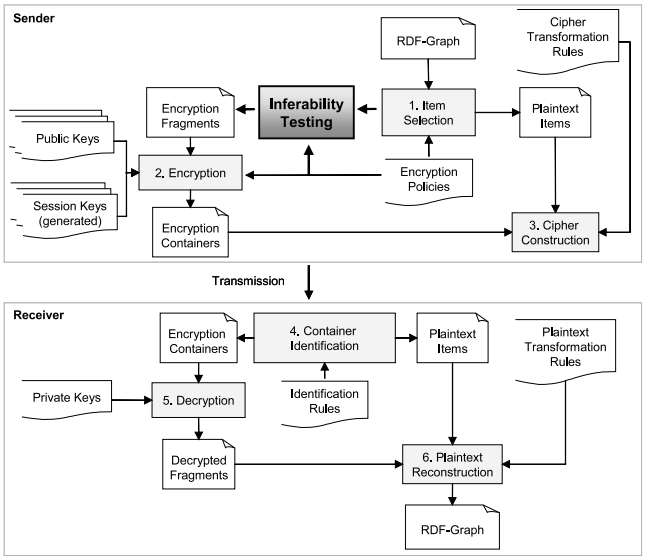

Figure 1: PRE-Process Overview

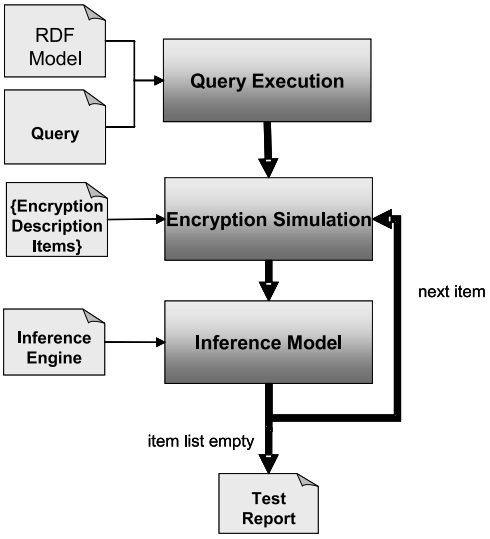

Figure 2: Inferability Testing Module

the `ENCRYPT`-section. `[o,1]` references all triples bound to the first pattern and encrypts the objects of those triples. `[t,2]` references all triples bound to the second pattern and encrypts them. Let the domain of `cardNumber` be `CreditCard`. The encrypted type information can be inferred using the RDFS domain-axiom.

2.2 Testing Module

The inferability testing module takes an RDF-model, a query, an encryption description, and an inference engine as input (figure 2). First, the query is executed on the model. Then for each encryption description item (`[o,1]` and `[t,2]`) the encryption is simulated by removing the corresponding fragments from the model and creating a temporary inference model according to the given inference engine. If the removed fragments can be derived their encryption is not secure. Finally, a report showing the inferability status of the data to be encrypted is generated which can be used for user feedback, etc.

3 Future Work

There are several key challenges to extend this work and making it practical for secure partial encryption of RDF-graphs. First, encryption metadata, such as keys and algorithms to be used, have not been considered by the testing module yet. Each encryption description item is assumed to use a different encryption key and thus is tested independently from the other items. Clustering of encryption description items based on their encryption metadata is expected to improve the performance significantly.

Secondly, the size of the tested models has been small (less than 1000 triples). To use the proposed method for large triple stores further performance improvements are necessary.

Another important aspect, is the usage of information provided by ontologies for known-plaintext attacks which has not been investigated yet.

4 Conclusion

In this paper, a first approach to test the inferability of RDF-data to be encrypted has been presented. An online accessible version has been implemented [Qian and Giereth, 2005] using the Jena framework [HP Labs, 2003] and has been tested with the reasoners for RDFS, OWL Lite, OWL DL, and OWL Full included in the Jena distribution.

References

[Brickley and R.V. Guha, 2004] Dan Brickley and R.V. Guha, editors. *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C Recommendation, http://www.w3.org/TR/rdf-schema/, February 2004.

[Giereth, 2005] M. Giereth. On Partial Encryption of RDF-Graphs. In *Proceedings of the 4th International Semantic Web Conference (ISWC 2005)*, Galway, Ireland, Nov 2005.

[HP Labs, 2003] HP Labs. Jena semantic web framework. 2003. http://jena.sourceforge.net.

[McGuinness and Harmelen, 2004] D. L. McGuinness and F. van Harmelen, editors. *OWL Web Ontology Language Overview*. W3C Recommendation, February 2004. http://www.w3.org/TR/owl-features/.

[Prud'hommeaux and Seaborne, 2004] E. Prud'hommeaux and A. Seaborne, editors. *SPARQL Query Language for RDF*. W3C Working Draft, http://www.w3.org/TR/rdf-sparql-query/, October 2004.

[Qian and Giereth, 2005] Y. Qian and M. Giereth. Inferability Testing Demo, July 2005. http://www.iis.uni-stuttgart.de/pre.

[Weitzner *et al.*, 2004] Daniel J. Weitzner, Jim Hendler, Tim Berners-Lee, and Dan Connolly. Creating a policy-aware web: Discretionary, rule-based access for the world wide web. Hershey, PA (forthcoming), 2004.