Signing Individual Fragments of an RDF Graph

Giovanni Tummarello[1,2], Christian Morbidoni[1], Paolo Puliti[1], Francesco Piazza[1]
[1] Università Politecnica delle Marche, Italy, http://semedia.deit.univpm.it [2] ISTI, CNR PISA (Italy)

ABSTRACT
Being able to determine the provenience of statements is a fundamental step in any SW trust modeling. We propose a methodology that allows signing of small groups of RDF statements. Groups of statements signed with this methodology can be safely inserted into any existing triple store without the loss of provenance information since only standard RDF semantics and constructs are used. This methodology has been implemented and is both available as open source library and deployed in a SW P2P project.

Categories and Subject Descriptors
H.2.1 [INFORMATION SYSTEMS]: Logical design

General Terms
Algorithms, Performance, Design, Security, Theory

Keywords
RDF, digital signature, semantic web, trust.

1. The problem
Authorship authentication and signing of RDF graphs is still in its infancy. The most relevant work is certainly [1] by J. Carroll which illustrates a nondeterministic, but relatively simple and efficient, procedure for providing a "canonical serialization" for (entire) RDF graphs. The canonical serialization is needed to digitally sign RDF as the same graph could be serialized in a very large number of model equivalent ways. The same author then argues that a mechanism for trust is "naming" graphs [2]; the two things when combined would work by associating a graph with its signature, which would remain external to the graph. This is similar to what has been informally proposed for signing FOAF files [3].
In this work we present a methodology to attach digital signatures closer to the individual statement and using only the standard RDF semantic [4].
This brings the following advantages:

- triples all lie in the same model (i.e. Computational space) so that they can all be conveniently considered at the same time when performing a query
- no need for special, non standardized, implementations (named graphs, quadruples)
- a graph can safely be split into minimal subsets of statements that nevertheless conserve the ability to verify the digital signature.

2.Definitions and properties
Let's first define what is the minimum "standalone" fragment of an RDF model. As blank nodes are not addressable from outside a graph, they must always be considered together with all surrounding statements, i.e. stored and transfered together. This is of course unless they have an IFP (Inverse Functional Property), which effectively makes them as addressable as URI nodes. We will here give a formal definition of MSG (Minimum Self-contained Graph) and will prove some simple properties laying the base for MSG signing.

WWW 2005, May 10--14, 2005, Chiba, Japan.
ACM 1-59593-051-5/05/0005.

Definition 3. Given an RDF statement s, the Minimum Self-contained Graph (MSG) containing that statement, written MSG (s), is the set of RDF statements comprised of the following:

1. The statement in question;
2. Recursively, for all the blank nodes involved by statements included in the description so far, the MSG of all the statements *involving* such blank nodes;

This definition recursively build the MSG from a particular starting statement; we now show however that the choice of the starting statement is arbitrary and this leads to an unique decomposition of the an RDF graph into MSGs.

Proposition 1. The MSG of a ground statement is the statement itself.

Theorem 1. If *s* and *t* are distinct statements and *t* belong to MSG (*s*), then MSG(*t*) = MSG(*s*).

Proof. If t belong to MSG(*s*), then, by the recursive definition, all statements in MSG(*t*) belong to MSG(*s*), so MSG(*t*) is a subset of MSG(*s*). We will now show that MSG(*s*) is a subset of MSG(*t*), thus proving the theorem. If *t* is a ground statement, MSG(*t*) = *t* != *s*, so *t* is not a ground statement. If *s* involves one of the blank nodes of *t*, then *s* belong to MSG(*t*) and MSG(*s*) is a subset of MSG(*t*). Recursively, if s involves one of the blank nodes of MSG (*t*), MSG(*s*) is a subset of MSG(*t*). If s does not involves any of the blank nodes of MSG(*t*), then MSG(*s*) and MSG(*t*) must be disjoint, which is against the original hypothesis.

Theorem 2. Each statement belong to one and only one MSG.

Proof. Is it straightforward to see that a statement belongs at least to a MSG, as the definition gives also an algorithm to build it. Lets suppose that a statement s belongs both to MSG(*t*) and MSG(*u*), where *t* and *u* are distinct statements. Then MSG(*s*) = MSG(*t*) and MSG(*s*) = MSG(*u*), so MSG(*t*) = MSG(*u*), i.e. they are the actually the same MSG.

Corollary 1. An RDF model has an unique decomposition in MSGs.

This is a consequence of the fact that all the black nodes, in the MSG definition, are "properly surrounded" by actual URIs (or literals). As a consequence, a graph can be properly reconstructed

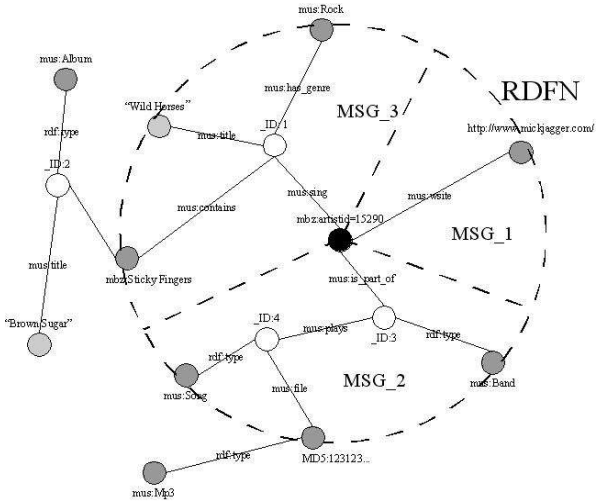

Image 1 The MSGs composing the RDFN involving a URI. Statements outside the circle are not included in the RDFN. Darker nodes are URIs, lighter ones bnodes

between 2 peers by transferring and merging one or more MSG at a time.

Definition 4. The RDF Neighborhood (RDFN) of a resource is the graph composed by all the MSGs involving the resource itself.

It is straightforward to see that a graph can be transferred by moving the RDFN of all the involved URIs. Example MSGs and RDFN involving a resource are illustrated in image 1.

3.Signing MSGs

Given that an RDF statement belongs to one and only one MSG, as previous definitions and properties show, we argue that it is possible to sign an MSG attaching the signature information to a single, arbitrary triple composing it.

The following example shows the signed version of an MSG as produced by our implementation.

A canonic representation of the graph is obtained implementing the algorithm described in [1] and is encrypted with a public key. The the digest is represented in RDF as a literal value. Along with the signature the public key to be used for verification is provided by means of a resolvable URI. This indication is itself covered by the signing procedure.

```
<rdf:Description rdf:about="http://www.musicbrainz.org?artistid=15290">
    <mus:is_part_of rdf:nodeID="3"/>
</rdf:Description>
<rdf:Description rdf:nodeID="3">
    <mus:plays rdf:nodeID="4"/>
    <rdf:type http://dbin.org/music#Band />
</rdf:Description>
<rdf:Description rdf:nodeID="4">
    <rdf:type http://dbin.org/music#Song />
    <mus:file urn:md5:123123....3123 />
</rdf:Description>
<rdf:Description rdf:nodeID="Sign_1">
    <rdf:subject rdf:resource="http://www.musicbrainz.org?artistid=15290" />
    <rdf:predicate rdf:resource="http://dbin.org/music#is_part_of"/>
    <rdf:object rdf:nodeID="3"/>
    <dbin:PGPCertificate rdf:resource:
            "http://public.dbin.org/cont/23878587 2.asc" >
    <dbin:Base64SigValue> McwOPX...A7xcB5w== </dbin:Base64SigValue>
    <rdf:type rdf:resource=
            "http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
</rdf:Description>
```

As shown in the example, by “attach” we mean using a reification procedure. Using the same procedure more signatures can be attached to the same MSG either independently or “layered” thus providing a mechanism for countersigning.

Given the MSG properties, this “information patch” can be merged into any existing RDF graph and the signature properties will be retained, checking the signature on any statement can be performed computing the MSG it belongs to (which will contain no triples from the pre-existing model) and to check if any of the statements carry a MSG signature on it.

4.Supporting information revision in highly replicated P2P environments

Other than authenticating provenience, this methodology has been successfully used to allow remote DB updates in our RDFGrowth P2P semantic web model [5].

In RDFGrowth, peers synchronize the RDFN about URIs they're interested in with those coming from other peer in a fully monotonic model(ever growing knowledge). In this architecture, MSGs produced by some are then passed and replicated by others many times so the only connection between those who produced the information and the consumer is the digital signature attached using the methodology here presented.

This not only supports trust at the client level by individually filtering MSGs from untrusted sources, but also allows a peer to issue "patches" that modify or cancel MSGs that he previously authored. In short, once a MSG has been signed, the hash can be used as a IFP, that is, as a unique way to identify the MSG itself. This in turn can be used in a subsequent MSG to indicate the one that it substitutes. Given that the authorship of this subsequent MSG can be verified to be identical to that of the original one, the client can safely perform the information update, no matter where it received the update patch from.

5.Notes and conclusions

The RDFN definition is similar to the Concise Bound Description (CBD) as used in the URIQUA semantic web agent model [6], albeit more extended than the one that was available at the time when MSGs where first introduced in the RDFGrowth P2P algorithm. Recent modifications of the CBD have also addressed the case where IFP are used on blank nodes and include reifications. The methodology presented here can be extended to encompass all this cases, although details cannot be included here.

Since this methodology uses reifications as a way to attach the signature to the MSGs, it is subject to all the shortcomings of this standard RDF construct. In particular, care should be used when using this proposed method in OWL FULL reasoners as the owl:sameAs property might cause substitutions inside MSGs. Given the digital signatures however, this change would immediately be detected and proper measures could be taken. Reification has also been often accused of being inefficient, that is, of causing “Triple bloat”. While this method does in fact see a consistent increase of triples when applied to very small MSGs (as in the previous example), this side effect becomes negligible as the MSG size grows, as only one statement needs reification.

This methodology has been implemented and is available as OS Java library. This library is also deployed in the SW P2P application Dbin (www.dbin.org) where it provides the foundations for a provenience based trust model as well as the knowledge update mechanisms as specified above.

6.Acknowledgments

Our gratitude goes to Mauro Mazzieri for the theorem formalization and to Fabio Panaioli for the implementation. We also thank Johan Johansson and Oreste Signore for the general support.

7.References

[1] J. Carroll, "Signing RDF graphs", HP technical report 2003

[2] J. Carroll, C. Bizer, P. Hayes, P. Stickler, "Named Graphs, Provenance and Trust ", HP technical report 2004

[3] E. Dumbill, "Signign FOAF files”, http://usefulinc.com/foaf/signingFoafFiles , personal communication

[4] RDF Semantics, W3C Recommendation, 2004

[5] G. Tummarello, C. Morbidoni, J. Petersson, P. Puliti, F. Piazza, "RDFGrowth, a P2P annotation exchange algorithm for scalable Semantic Web applications", P2PKM , Boston 2004

[6] P.Stickler URIQA The URI Query Agent Model, NOKIA 2003