A Study of RDF Security Concerns in Semantic Web

Ubaidullah, Fizza Abbas, Rasheed Hussain, Junggab Son and Heekuck Oh
Dept. of Computer Science and Engineering, Hanyang University, South Korea
ubaidullah@hanyang.ac.kr

Abstract

The Semantic Web is leading us to a world of information sharing by enabling distributed knowledge aggregation and creation. RDF is the foundations of the Semantic Web. For secure Semantic web we need to secure RDF as well. Unauthorized access to an RDF document can change or damage its semantics or manipulate the relations between resources. This article includes the study of RDF security issues and analysis of the existing solutions. After finding limitations of existing solution, a hybrid approach has been proposed.

Keywords- Semantic Web, RDF, Security, Encrypted Data.

I. INTRODUCTION

Semantic web is an initiative taken by World Wide Web Consortium (W3C) and is aimed to convert the current web, which is currently a web of structured and semi structured documents, into a web of data. The idea was to make machines able to directly manipulate the data by “understanding” it, i.e. it makes documents on the web self-explanatory so that machines can understand them in a similar way as humans do. By this understanding we can imagine semantic web as a sophisticated and intelligent web where many of the tasks which are carried by humans nowadays, can be carried out without or with little help of them. Tasks such as integration of different types of information systems, collaboration and coordination of various activities, complex scheduling as well as finding a particular piece of information on web are considered as to be done by machines by exchanging and understanding well-structured documents [2].

There are many important components of semantic web such as XML, Resource Description Framework (RDF) and Ontologies. eXtensible Markup Language (XML) is designed to transport and store data on web. It is a set of rules which are used to define web documents in a way which is readable by both human and machines. XML uses tags to describe its structure and data. These tags are not pre-defined like HTML but they are defined by the author of an XML.

RDF on the other hand is a framework which describes a web resource. For example an RDF can describe the author, title, contents, modification date and copyright information about a web page [1]. RDF documents are written by using XML. For identifying resources on the web RDF uses Web identifiers (URIs).

Ontologies are used to define the concepts and relationships which are used to describe and represent a specific area of interest. For example British medical association defines their own ontologies for describing and representing knowledge about symptoms of a disease and its treatment while American medical association describes their own ontologies or maybe they both define their combined set of ontologies.

These components that constitute the semantic web have to be secure. In addition, the components have to be integrated securely. In subsequent sections we turn our focus preliminary on RDF and related security issues.

II. BACKGROUND

As we have discussed above, the web is evolving from human understandable form to a more machine/human understandable form. This semantic web vision enables to use web as a source of information sharing by creation and aggregation of knowledge distributed across the web. It is, therefore, necessary to manage the large amount of data require by most of semantic web applications. This data is stored in knowledge stores. [3]

The basic components of RDF data model are resources, properties and statements. These components can be represented by a simple graph model. They present the concept of a resource that can be uniquely identified by a URI. This resource can be anything on the web, thus implies that it is not necessary that a resource can only be a web
A secure semantic web is of utmost priority [10]. Consider the scenario when machines are communicating and collaborating, resources are being identified and statements are being analyzed. Many of RDF documents’ security are critical for example exchange of RDF documents about some military projects or a bank’s e-transactions. Now to make this semantic web secure we need to make RDF secure. To secure RDF we need to address many issues. For example the security of RDF resources, properties and statements, security of container models i.e. bag, sequence and alternatives [5]. We need to see that should we secure only security of container models i.e. bag, sequence and RDF resources with their associated properties and the value of the properties. When RDF statements are used to describe a similar scenario then it is referred to as RDF description [4]. RDF containers are a more advance concept and it has three types of container object. There is concept of Bag, Sequence and Alternatives. A Bag comprises of an unordered list of resources. It states that a property contains more than one values but the order of those is not important. On the other hand a Sequence is an ordered list hence the order is important. Alternatives represent a list of resources and it represents multiple alternatives for the value a property contains [1].

III. MOTIVATION

Tim Berners Lee has specified various layers for the semantic web. Security and privacy cut across all layers as shown in Fig 1.

The dependability of semantic web includes security, privacy, trust, Real-time processing, fault tolerance all referred as “Trustworthiness”. Confidentiality is to prevent the release of unauthorized information considered sensitive. Privacy is concerned with individuals and trust is the confidence one has about an individual that he provides his/her correct information.

A secure semantic web is of utmost priority [10]. Consider the scenario when machines are communicating and collaborating, resources are being identified and statements are being analyzed. Many of RDF documents’ security are critical for example exchange of RDF documents about some military projects or a bank’s e-transactions. Now to make this semantic web secure we need to make RDF secure. To secure RDF we need to address many issues. For example the security of RDF resources, properties and statements, security of container models i.e. bag, sequence and alternatives [5]. We need to see that should we secure only some particular parts of an RDF document or complete document. The other thing is what should be the security mechanism i.e. how security should be enforced. These are the current issues and the security of RDF documents is still evolving. In the following section we present review of literature so that we could come to know that what is happening related to RDF security and what limitations we face in current research [2].

IV. LITERATURE REVIEW

There are different strategies dealing with RDF security. Some authors propose frameworks for enforcing RDF as security access control enforcement, some focus on RDF store with integrated maintenance capabilities and access control using user defined policies and other follow the strategy that specification relies on XML specification. XML security is essential for securing RDF documents. Following is the discussion about features of the exiting solutions and the identification of their limitations.

A. RDF Document Securities with Cryptography and Digital Signature

As per Internet Engineering Task Force (IETF) document RFC 3870, RDF data can be secured for integrity, authenticity and confidentiality using any of the mechanisms available. This includes XML data, including XML signature, XML encryption. RDF document is written using XML syntax. There are many approaches for RDF document security. Here two approaches are mentioned. First is “W3C XML Encryption Syntax and Processing” that is release in 10 December 2002 (1.0). Its Latest version has released on 18 Oct 2012 (1.1). The other one is IBM developersWorks’ XML Security.

1) XML Encryption Syntax and Processing

This approach is given and a recommendation by W3C, specifies a process for encrypting data and representing the result in XML/RDF. The data may be arbitrary data (including an XML/RDF document), an XML/RDF element, or XML/RDF element content. Result of encrypting data is an XML Encryption “EncryptedData” element which contains (via one of its children's content) or identifies (via a URI reference) the cipher data. When encrypting an XML element or element content the EncryptedData element replaces the element or content (respectively) in the encrypted version of the XML document. [1]

2) IBM developersWorks: XML Security

IBM developerWorks XML Security implements the RDF document security by introducing security features such as digital signature, element-wise encryption, and access control. Digital signature implementation is also based on "XML-Signature Syntax and Processing" by W3C/IETF. [6].

3) Limitations

a) Relationship to XML Digital Signatures

The first problem arises when we encrypt and digitally signed portions of XML/RDF documents. On decryption or when verifying signature it is then necessary to know that if the signature was computed on encrypted elements or unencrypted elements [5].

Another important consideration is the induction of cryptographic vulnerabilities when encryption and digital signature are applied on an XML element. Famous cryptography expert Hal Finney suggests that when you only
encrypt digitally signed data but not the digital signature it may lead to plain text attack [5]. Although W3C suggest that the use of secure hashes and nonces may help to reduce these kinds of attacks.

b) Information Revealed

The use of symmetric key is another issue in this recommendation of W3C. When the symmetric key is shared among users, it should not be used for messages which are not for all the users. Even if one recipient is not directed to information, intended (exclusively) for another in the same symmetric key, the information might be discovered and decrypted [6]. W3C presents no solution in this type of scenario. [5]

c) Nonce and IV (Initialization Value or Vector)

Many of the encryption algorithms possess an undesirable property that when the same plain text is encrypted with the same key it produces same ciphertext. Though it is not surprising but it invites for plaintext attacks. This can be mitigated by attaching a non-repeating data under a given key with the plain text. In encryption chaining modes this data is the first to be encrypted and is consequently called the IV (initialization value or vector). [5]

d) Denial of Service

A denial of service (DOS) attack is also possible. Suppose a scenario where EncryptedKey “A” requires to be decrypted with EncryptedKey “B” and EncryptedKey “B” in turn requires to be encrypted with EncryptedKey “A”. It is also possible that an attacker sends some encrypted data whose decryption requires lots of network resources. In this regard the suggestion from W3C is to restrict network resources to be allocated when a request comes. [5]

e) Unsafe Content

An attacker can send malicious data in encrypted form to a firewall or virus detector. Hence the firewall or virus detectors must treat any encrypted item similarly as a threat as it considers a normal data item. [5]

W3C only give some suggestions with respect to these security weaknesses and does not provide any concrete solution of above mentioned limitations in W3C XML Encryption Syntax and Processing. An RDF/XML document does not itself contain any method to prevent these kinds of security threats. Hence other mechanism should be enforced to avoid a security threat.

B. RDF Metadata for Access Control

The approach given by Gowadia et al. [7] utilizes RDF metadata i.e. DTD and XML schema as an access control method. The architecture is composed of three components which are 1. Query Engine 2. Access Control and 3. User’s history. Query engine is responsible for generation of responses to requests by users. Access control is responsible for evaluating the authorization of a user’s request based on the security policy and the history of the user. When a query comes and no security violation is detected then the query’s answer is returned and user’s history is updated.

1) Limitations

This scheme does not provide any security when a query answer is returned. It simply returns the result as plain text and thus a potential target for a sniffer or eaves dropper as he/she may listen to and/or alter the result of the returned query. Also if the metadata is needed to be changed then security policy may also be needed to write again.

C. Policy Specification and Enforcement by using RDF

Here we discuss an approach which is given by B. Carminati et al. [4]. They present the idea that access control policies should be defined and enforced by using RDF document itself. The policies are defined by analysis and the Security Administrator or owner of the data defines them. It is worth mentioning here that these policies are enforced to a similar domain with similar security requirements. The paper suggests that the policy is a set of authorizations and is implemented with set of tuples (s,o,m) i.e. subject s can access object o under policy m. The architecture is implemented by a high level policy generator which is used to generate a domain specific policy after receiving an RDF document. This is done by Security Administrator (SA). The policy is then enforced by an authorizations entailment component. Finally access is granted by a reference monitor.

Another similar approach has been discussed in V. Palakonstantinou et al. [8], which enforce access control for RDF by using RDF quadruples (s, p, o, l) where s, p and o are subject, property and object while l is an abstract access control expression. The model is comprised of abstract tokens and abstract operators. Tokens are assigned to RDF triples through authorization rules, while the abstract operators describe the computation of access labels for implied triples. The scheme uses SPARQL queries to assign tokens to triples.
1) Limitations
The security of above mentioned approaches depends on the security of underlying databases where the RDF documents are stored. Also if restructuring of XML/RDF document is required then it will be a tedious task because the document structure is itself containing the security policy.

D. Oracle’s RDF triple store
An RDF statement is expressed as a triple i.e. (Subject or Recourse, Predicate or Property, Object or Value). Oracle database provides one universal repository for storing RDF triples [9]. The data is stored in the Oracle’s Spatial Database. An RDF triple is treated as one database object. The Oracle triple store implements security by giving different privileges for creating, updating and deleting rule bases or objects. Similarly it gives privileged access to perform DML operations to a user. Rule bases are the criteria which are defined to access an RDF triple.

1) Limitations
The main limitation is seems to be the security implementation on a complete RDF triple i.e. user cannot implement security policy to the individual elements of a triple. For example if a user wishes that a particular subject must not access some particular object then this privilege is not possible but a complete triple can be securely accessed.

V. PROPOSED SOLUTION
By looking at the various approaches presented for RDF security, it is evident that they lack a concrete solution. Every approach has its own limitations. On one side W3C implements security on RDF document with the help of symmetric key / public key cryptography but faces issues related to cryptography security. On the other side when XML/RDF schema and triples are used for security policy description and enforcement then the restructuring of schema and security of underlying database becomes a hindrance. After analyzing all these approaches and their limitations it is felt that the need for a concrete solution is still required.

As we have seen that all of above solutions have some limitations when providing a security mechanism for the secure access to the RDF document. Despite these limitations, each of the solution also contributes significantly towards RDF security. By looking at this aspect we propose that a hybrid solution can be made by combining different advantages of various approaches.

A. Use of a robust and well secured Database
The architecture must utilize a robust database like Oracle RDBMS for storing RDF documents or triples. This not only provides security features such as authentication, integrity and authorization (in case when a request comes) but also when a response is generated, it is in encrypted format and an attacker cannot sniff it.

B. Utilizing metadata of RDF Document
The proposed architecture utilizes RDF metadata for analyzing that a user has the privilege to access the particular document. On a positive reply it will retrieve the RDF document form the database. The database will return that document in an encrypted format and that will be returned to the user. On a security violation attempt by a user, the user will be added to the malicious user’s history record.

D. User History
A user history is the history which keeps the record of a malicious user who attempts to breach the security of the system.

This proposed solution, on one hand, is very generic and needs to be more detailed. On the other hand it provides the basis for a concrete solution for implementation of the RDF security.

VI. CONCLUSION
RDF has significant role in developing the semantic web. Researchers address various solutions for RDF security concerns. Some researchers emphasis on XML Signature, Syntax and processing, XML-Encryption Syntax and Processing and XML Key Management to secure RDF document, other focus on frameworks to enforce RDF as security policy & access control and RDF triple stores. In this paper the limitations of these approaches are identified. Finally by looking at various aspects of these solutions, a combined approach has been presented which aims to provide better results in concern of several security measures. There is still further room for the betterment in the security issues align with efficiency of RDF document accessing.

References