XML and Web Services Security

Lili Sun, Yan Li
Department of Mathematics and Computing, University of Southern Queensland
Toowoomba, QLD 4350, Australia
{sun, liyan}@usq.edu.au

Abstract

With an increasing amount of semi-structured data XML has become important. XML documents may contain private information that cannot be shared by all user communities. Therefore, securing XML data is becoming important. Several specifications progressed toward providing a comprehensive standards framework for securing XML-based application have been presented. These applications can be effective to protect information in a website. In this paper, we present XML and Web service security main standards and most specifications for these standards. Each standard which connects with protecting XML based documents is discussed, especially we present XML undeniable signature as an application with XML digital signature. We also briefly describe the relations with these standards based on existing technologies. Finally, comparisons with related works are analyzed.

Keywords: XML, XML documents, Security technologies, Web service security.

1. Introduction

Over the past several years, there has been a tremendous surge of interest in XML as an universal, queryable representation for data. XML web service is a platform-independent Web application that accepts requests from different systems on the Internet. XML is a fundamental component in many XML web services and it is used to store and exchange data in the Internet environment that may include private message. It overcomes the complexity of Standard Generalized Markup Language (SGML) and the user can define document structures, removing the limit of the fixed tags in Hypertext Markup Language (HTML).

Security technologies provide security algorithms and technologies that can be used in XML security, but for many of them the actual formats used to implement security requirements are inappropriate for most applications. Usually these standards are not designed for use with XML and do not support common XML technical approaches for managing contents, such as specifying contents with uniform resource identifier strings (URIs) or using other XML standard definitions for locating portions of XML contents (like XML Path Language[XPath])[15]. Meanwhile these standards use binary formats that require specialized software for interpretation and use them, even for extracting portions of the security information. In addition, some current existing security technologies, such as Secure Socket Layer (SSL), Transport Layer Security (TLS) and HTTPS [12] provide several specifications for web services to enable security. But there are some issues with these schemes. SSL, TLS both provide transport level security, not message level security. They are point-to-point security only and do not handle end-to-end multi-hopped messaging security. Security only when data is in transition, does not secure data off transition. HTTPS do not support non-repudiation. In 2002, several specifications were proposed for securing XML-based applications and web services. These standards support to integrate security functionalities into their XML based applications.

Figure1 shows some of the most important specifications for XML and Web service security. This provides a standard framework for XML based applications. Usually Simple Object Access Protocol (SOAP) is used for message transport. XML digital signature and XML encryption are used for data confidentiality and integrity. Security Assertion Markup Language (SAML) focuses on authentication assertions. XML Access Control Markup Language (XACML) is for information access control. XML Key Management Specification (XKMS) is used to manage Public key infrastructure and Web service security brings standards together.

The remainder of this paper is organized as follows: Section 2 illustrates the background of XML. Section 3, 4 and 5 present a brief overview on the following core XML Security standards [14]:

- Integrity and signatures - XML Digital Signature
- Confidentiality - XML Encryption
- Key Management - XML Key Management Specification
- Authentication and Authorization Assertions – Security Assertion Markup Language
- Authorization Rules - XML Access Control Markup Language
- Web Services Security - WS-Security

Finally, Section 6 concludes the paper.

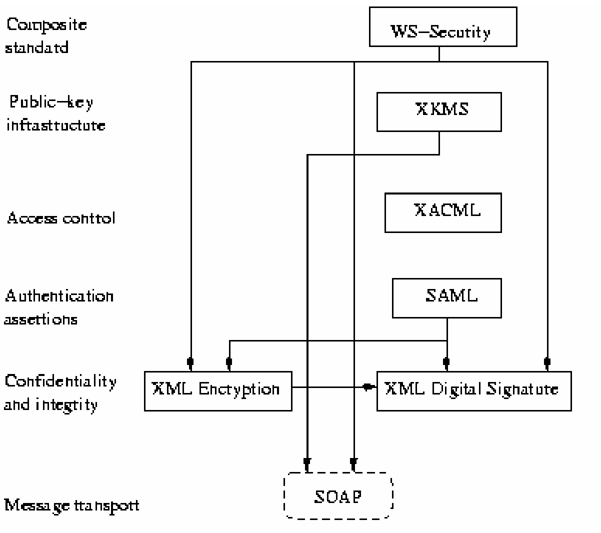

Figure1: XML security standards

2. XML

Extensible Markup Language (XML) [8, 10] is a markup language for describing semi-structured information. XML documents can be classified into two categories: well-formed and valid. A document can be well-formed if it follows the grammar rules of XML, such as there is exactly one element that completely contains all other elements, elements may nest but not overlapped, etc. A well-formed document is valid only if it contains a proper Document Type Definition (DTD) in the source and if the document obeys the constraints of that declaration. Validation requires XML instance to contain specified elements and attributes, the following example shows an XML document of a student card information specified datatypes and relationships. XML differs from HTML since it allows users to define their own vocabularies of opening and closing tags.

```
<?xml version= “1.0” encoding= “UTF-8”? >
   <studentInfo xmlns=
    “http://www.school.com/StudentInfo”>
     <ID> 123-45-6789 </ID>
        <name>
           <firstname> Mike </firstname>
           <lastName> Harman </lastName>
        </name>
     <studentCardInfo>
        <type> Student card </type>
        <cardNo> 8888888888888 </cardNo>
        <expireDate> 10/08 </expireDtae>
        <nameOnCard> Mike Harman
        </nameOnCard>
     </studentCardInfo>
   </studentInfo>
```

Table 1: XML Document Example

XML document not only shows the contents of data but also gives the constraints and relationships among data. In Table 1, the element *studentInfo* includes *ID*, *name* and *studentCardInfo* sub-elements. The sub-element *ID* is a simple type while sub-elements *name* and *studentCardInfo* are combined with their own sub-elements. Since an XML document can express complex relationship between data, it may be generated from various resources with varying security requirements. Also such advantages a user may like to access particular parts of an XML document. In the above example, for the *studentInfo* objects everyone can read all the information. On the other hand, when an internal or external user accesses this document, his/her access permission has to be limited according to security policies in all databases. These examples show that securing XML document is a significant topic.

3. XML digital signature

Digital signatures are an important element in electronic security because they can be used to ensure the integrity, authenticity, and non-repudiability of data [16]. XML digital signatures are designed for use in XML document transactions. XML signatures include authentication, data integrity, and support for non-repudiation to the data that they sign. It has established an approach for data exchange as well as some vocabulary shared by the other standards. The XML digital signature specification on the website, www.w3.org/TR/2002/REC-xmldsig-core-20020212[2], is a final draft which provides guidelines in this area. It specifies the basic structures of defining a digital signature using XML. XML Signatures can be applied to any digital contents, including XML documents. Table 2 shows the structure of XML signature and its key elements.

```
<Signature ID>
      <SignedInfo>
          <CanonicalizationMethod/>
          <SignatureMethod/>
           <SignatureValue>
             (<Reference URI>
                 <DigestMethod>
                   <DigestValue>
             </Reference>)
       </SignedInfo>
       (<KeyInfo>)
</Signature>
```

Table 2: XML Signature Structure

An XML Signature can sign more than one type of resources [11]. There are three types of XML digital signature structures, namely enveloped signature, enveloping signature and detached signature. For enveloped and enveloping signatures, the signed XML documents and its securing signatures are within the same files. For the detached signatures, the XML signatures are in a separated document.

An XML digital signature differs from other protocols for message signing, such as PGP, since it supports for signing only specific portions of the XML tree rather than the complete documents.

3.1. Xml undeniable signature

The central role of digital signatures in the commercial and legal aspects of the evolving electronic commerce world is well recognized [17]. Advanced signature schemes include group signatures, blind signatures, undeniable signatures and proxy signatures. Undeniable signatures were firstly introduced by Chaum and Van Antulerpen [9]. They secure that signatures cannot be easily verified. Undeniable signature schemes are used in places where the co-operation of a signer is required in the verification. XML undeniable signature approach builds a bridge between the existing XML technologies and data security theories. XML undeniable signature scheme is based on RSA undeniable signature algorithm for XML documents.

We apply a undeniable signature algorithm for <SignatureMethod> in the XML signature structure. It consists of generating signature, confirmation protocol and deniable protocol. The application of the undeniable signature algorithm in XML signature is the fundamental and important step for the new XML undeniable signature. The XML undeniable signature approach is a new way to secure sensitive information in XML document transitions and signers can not deny their signatures. It can provide a secure framework to XML web services.

4. Xml encryption and key management

4.1. Xml encryption

The XML Encryption Syntax and Processing specification defines an XML vocabulary and processing rules for protecting confidentiality of XML document [3]. It may work in whole or in part of XML documents and non-XML data as well. In contrast to other commonly used technologies for confidentiality such as SSL, XML encryption also applies to document parts and documents in persistent storages.

Encryption is generally using symmetric key encryption. But this may cause a problem as sending confidential information to a receiver, the sender and the recipient must also share the symmetric key without anyone else. This can be difficult without person to person contact. To avoid this problem and make it easier to share confidential contents with a number of people, asymmetric or public-key cryptography was designed. Public key cryptography uses a matched pair of keys, one for encryption and one for decryption. In this encryption process the sender encrypts using the recipient's public key that can be shared widely. The recipient decrypt using private key that known only to themselves. This helps to take over the difficulty of establishing confidential communication. But when public key cryptography and symmetric cryptography are used together, they will become more efficient. The symmetric key is used to encrypt the content, and then the symmetric key is encrypted using public key cryptography. Both the encrypted content and encrypted symmetric key are then sent to the recipient. In the Encryption syntax, <EncryptedData> element is core element. It also contains: <EncryptionMethord>, <KeyInfo>, <CipherData> and <EbcryptionProperties> sub-elements. As same as above example, it shows a *customerInf* in XML document. It only encrypts the elements of <CreditCard> element.

```
<customerInfo
   xmlns= “http://www.hotel.com/CustomerInfo”>
     < ID > 123-45-6789 </ID >
     <name >
       <firstName > Tony </firstName>
       <lastName > Zhang </lastName>
     </name >
     <creditCardInfo>
        <EncryptedData
       xmlns= “http://www.w3.org/2001/04/xmlenc#”
         Type=“http://www.w3.org/
                2001/04xmlenc#Content” >
         <CipherData>
             <CipherValue>A12B34C657
             </CipherValue>
         </CipherData>
       </EncryptedData>
     </creditCardInfo>
  </customerInfo>
```

Table 3: XML Encryption

4.2. Xml key management specification

Public key technology is an essential part of XML Digital Signature, XML Encryption and other security applications. XML Key Management Specification (XKMS) defines protocols between XKMS client and server for performing public-key infrastructure (PKI) operations [1]. XKMS defines XML message formats to support requests and responses for public key management, including public key registration, public key validation, public key discovery and public key revocation. Therefore, XKMS is designed to use along with XML digital signature and XML encryption, it helps to manage the public key enabling signature verification and encrypting for recipients. PKI plays an important role in the Web services and E-commerce. Since PKI operations are too expensive to small devices, using XKMS may reduce the processing burden by moving it to an XKMS server. On the other hand, PKI operations are too complex to many applications, using XKMS can ease the integration of PKI by moving the complexity of PKI operations to an XKMS sever.

The following example shows that XKML works with a document signature. A client receives a signed XML document, the client sends the <ds:Keyinfo> element to the location service requesting that the <KeyName> and <KeyValue> elements be returned. In the <ds:Keyinfo> element it specifies a <ds:RetrievalMethod> for an X.509 certificate that contains the public key. The location service resolves the <ds:RetrievalMethod> to obtain an X.509v3 certificate. The certificate is parsed to obtain the public key value that is returned to the client. The <KeyName> returned is obtained from the certificate.

```
  Request:
    <Locate>
        <Query>
             <ds:KeyInfo>
                 <ds:RetrievalMethod
                 URI= “http://www.PKeyDir.test/
                 Certificates/01293122”
                 Type= “http://www.w3.org/2000/09/
                  xmldsig#X509Data”/>
              </ds:KeyInfo>
        </Query>
      <Respond>
           <string>KeyName</string>
           <string>KeyValue</string>
      </Respond>
    </Locate>
Response:
    <LocateResult>
       <Result>Success</Result>
            <Answer>
                <ds:KeyInfo>
                   <ds:KeyName>
                    O=XMLTrustCernter.orgOU= “Crypto”
                    CN= “Alice”
                   </ds:KeyName>
                   <ds:KeyValue>...</ds:KeyValue>
                </ds:KeyInfo>
            </Answer>
      </LocateResult>
```

Table 4: XKML Validation Request and Respond

5. Language and web services security

5.1. Extensible access control markup language

Extensible Access Control Markup Language(XACML) is an XML specification for expressing fine-grained information access policies in XML documents or any other electronic resources [5]. XACML expresses or communicates by using the rules and policies. An access control mechanism uses to derive an access decision for a set of subjects and attributes. Access control lists in XACML are 4-tuples: subjects, target objects, permitted action, provision. For example, in the access request, “Allow the school manager to create files in the student folder on the school server”, the subject is the “school manager”, the target resource is the “student folder on the school server”, and the action is “create files”. Using XACML can standardize the access control language in XML. It can make lower costs when people use it since no writing policy in several languages and administrators only need to understand one language.

Considering the following rule taken from the XACML, this example will grant a read access to patient medical record only by their primary doctors. A patient has his/her patient records which includes mental problem notes. The patient grants an access right to mental problem notes only to their primary care doctors. The primary care doctor then grants an access to patient record to associate doctor with access restriction so that associate doctor has no access to mental problem notes.

```
<content>
   <entry>
       <name> Alice</name>
       <record> mental problem </record>
   </entry>
</content>
<policy>
    <xacl>
         <object href= “/contents”/>
         <rule>
             <subject>
                 <uid primary care doctor />
             </subject>
             <action >
                  name= “read” permission= “grant”
             </action>
         </rule>
    </xacl>
  </police>
```

Table 5: XACML Using Example

5.2. Security assertion markup language

Security Assertion Markup Language (SAML) defines an XML framework for exchanging authentication and authorization information [6]. By comparing with XACML, they both share a lot of concepts and a domain – the domain of authentication, authorization, and access control. However, the problems they address in the same domain are different. SAML addresses authentication and provides a mechanism for transferring authentication and authorization decisions between cooperating entities, while XACML focuses on the mechanism for arriving at those authorization decisions. SAML can be used to share security information in Single Sign-on (SSO) with different systems and platforms. When using multiple networked systems a general requirement is "single sign-on", that means authenticating once and then sharing the result of authentication with multiple systems to avoid repeated authentication. For example, Logged-in (authenticated) users of Smith.com are allowed to access to their sister site Johns.com without relogin.

SAML is not concerned with confidentiality, integrity, or nonrepudiability of assertions in transit. The following simplified authentication assertion example states that Smith (subject) was authenticated by "password" at certain time towards Single Sign-on using.

```
<Assertion>
    <AuthenticationStatement
      AuthenticationMethod= "password"
      AuthenticationInstant= "2003-12-05T10:00:00Z">
        <subject>
          <NameIdentifier
            SecurityDomain= "Johns.com"
            Name= "Smith"/>
            <ConfirmationMethod>
            http://...core-25/sender-vouches
            </ConfirmationMethod>
        </subject>
      </AuthenticationStatement>
 </Assertion>
```

Table 6: SAML Assertion Example

5.3. Web services security

In April 2002, IBM and Microsoft have issued a Web Services security architecture and roadmap (www.106.ibm.com/developerworks/webservices/library/ws-secmap/) that state a strategy and specifications to bring different security technologies together [4]. The WS-Security specification provides how XML Digital Signatures and XML Encryption may be used with SOAP [7] messages. Specifically, WS-Security describes enhancements to the existing SOAP message. It provides quality of protection for SOAP messages through applications in message integrity, confidentiality, and single message authentication. WS-Security provides technologies to build a wide variety of security models. For example, a client might provide a proof of identity and a signed claim to the bank that he/she has a fix time-deposit certification. In the Web service when receiving such a message could then determine what kind of trust he/she places in the claim. XML Signature may provide message integrity and security tokens can ensure that messages have originated from an appropriate sender and were not modified in transit. Similarly, XML Encryption may provide message confidentiality and security tokens can keep portions of a SOAP message confidential.

6. Conclusions

XML is today widely used in a large variety of applications and industry products as it has become the standard for describing data and documents circulated across the web. We need to secure XML messages to form the basis of business transactions. In this paper we present a brief introduction to XML and Web services security standards and how they work together. The XML Security standards define XML languages and processing rules for meeting common security requirements. These standards incorporate with the use of the other XML Security standards, especially the core XML Digital Signature and XML Encryption standards. SAML and XACML are used for the sharing policy statements. Obviously, XML security standards will be essential as XML technologies are adopted for Web services. The existing standards have established a good framework for developers who need to integrate security functionality into their XML-based applications.

Reference

[1] XML Key Management Specification 2.0(XKMS). 10 March 2002, http://www.w3.org/TR/xkms2/.

[2] XML-Signature Syntax and Processing. February 2002, http://www.w3.org/TR/xmldsig-core/.

[3] XML Encryption Syntax and Processing. 10, December 2002, http://www.w3.org/TR/xmlenc-core/.

[4] Web Services Security: SOAP Message Security 1.0 (WS-Security 2004). March 2004, http://docs.oasis-open.org/wss.

[5] eXtensible Access Control Markup Language (XACML). 1 Feb 2005, http://docs.oasis-open.org/xacml/2.0/.

[6] Security Assertion Markup Language (SAML) 2.0 Technical Overview. 20 Feburary 2005, http://www.oasis-open.org/committees/security.

[7] Box D. Simple Object Access Protocol (SOAP) 1.1. World Wide Web Consortium (W3C), Cambridge, MA, USA. http://www.w3.org/TR/soap, 2000.

[8] Bray T., Paoli J., Sperberg M. and Maler E. Extensible Markup Language (XML) 1.1 (Second Edition). *World Wide Web Consortium (W3C)*, Cambridge, MA, USA. http://www.w3.org/TR/REC-xml, 2000.

[9] Chaum D. and Van Antwerpen H. Undeniable signatures. *Advances in Cryptology--Crypto89 volume 435 of Lectures Notes in Computer Science*, pages 212—216, Springer-Verlag, 1990.

[10] Damiani E., Capitani S. and Samarati P. Towards securing xml web services. *Proc. of the 2002 ACM Workshop on XML Security*, Washington, DC, USA, November 2002.

[11] Ford W. and Baum M. S. Secure electronic commerce: Building the Infrastructure for Digital Signatures & Encryption. Prentice Hall PTR, 1997.

[12] Freier A., Karlton P., and Kocher. P. The ssl protocol - version 3.0. http://ftp.nectec.or.th/CIE/Topics/ssldraft/.

[13] Jothy R. and David R. Securing Web Services with WS-Security: Demystifying WS-Security, WS-Policy, SAML, XML Signature, and XML Encryption, Sams, 2004.

[14] Naedele M. Standards for XML and Web Services Security, *ABB Corporate Research*, 2003.

[15] Rutgers Security Team. WWW Security. 1999, http://www-ns.rutgers.edu/www-security/.

[16] Simon E., Madsen P. and Adams C. An Introduction to XML Digital Signature. 08 August 2001, http://www.xml.com/pub/a/2001/08/08/xmldsig.html.

[17] Sun L. and Li Y. Xml undeniable signature. *Proceedings of International Conference on Computational Intelligence for Modeling, Control and Automation*, pages 981--985. IEEE, 2005.