Volume 6, Issue 1, January 2019 ISSN: 2348-8565 (Online)

International Journal of Modern Engineering and Research Technology

Website: http://www.ijmert.org Email: editor.ijmert@gmail.com

Implementation of Hybrid Security Scheme for the Application of Web Service

Shinkita Shrivastava
M.Tech. Research Scholar
Shri Ram Group of Institutions
Jabalpur (M.P.), [INDIA]
Email: shinkitashri@gmail.com

Sapna Choudhary
Head of the Department
Department of Computer Science & Engineering
Shri Ram Group of Institutions
Jabalpur (M.P.), [INDIA]
Email: choudharysapnajain@gmail.com

ABSTRACT

XML and Web services are widely used in current distributed systems. The security of the XML based communication, and the Web services themselves, is of great importance to the overall security of these systems. Furthermore, in order to facilitate interoperability, the security mechanisms should preferably be based on established standards. In this paper we provide a tutorial on current security standards for XML and Web services. The discussed standards include XML Signature, XML Encryption, the XML Key Management Specification (XKMS), WS-Security, WS-Trust, WS-Secure Conversation, Web Services Policy, WS-Security Policy, the eXtensible Access Control Markup Language (XACML), and the Security Assertion Markup Language (SAML).

This paper analyzes the limitation of conventional digital multi-signature. By utilizing the structure predominance of the XML documents and the security of conventional digital signature, we propose a multi-signature scheme of XML documents based on RSA. In our scheme, we use the Xpath to transform an XML document into subdocument, and each participant signer signs the sub-document that they are responsible for. Consequently, improve the efficiency and the flexibility of signs.

Keywords:— XML, Multi-signature, RSA, Xpath

I. INTRODUCTION

A WEB service is defined as a software system designed to support interoperable machine-to-machine interaction over a network [1]. Put in another way, Web services provide a framework for system integration, independent of programming language and operating system. Web services are widely deployed in current distributed systems and have become the technology of choice for implementing service-oriented architectures (SOA). In such architectures, loosely coupled services may be located across organizational domains.

The suitability of Web services for integrating heterogeneous systems is largely facilitated through its extensive use of the Extensible Markup Language (XML). The interface of a Web service is for instance described using the XML based Web Services Description Language (WSDL). Furthermore, communication is performed using XML based SOAP messages.

Thus, the security of a Web services based system depends not only on the security of the services themselves, but also on the confidentiality and integrity of the XML based SOAP messages used for communication.

The Organization for the Advancement of Structured In-formation Standards (OASIS) and the World Wide Web Consortium (W3C) have over the last years standardized several specifications related to security in Web services and XML. This paper provides an overview of these security standards. Using these established standards when creating Web services, instead of custom solutions, clearly has the advantages of facilitating both system interoperability and reusability.

XML has become a standard format for data interchange in Internet. Due to its extensibility, XML is widely adopted in various fields such as Web services, EDI (Electronic Data Interchange), E-government and E-commerce and so on. With the widespread application of XML, people pay more attention on the problem of XML data security. XML digital signature not only guarantees the integrity, the authenticity and the undeniable of the information, but also provides the status distinction. When processing the signature of XML document, it displays much incomparable technical superiority with the conventional digital signature, thus it meets more special signature.

In the E-government or E-commerce's work flow, many documents such as the contract, the agreement, the request form, it not only need the independent signature personally, but also need many person's signatures. For example, when a company enters in the accounts, it need manager, the treasurer, and the teller's signature. Extending to the digital signature domain, which requests lots of users to do the digital signature for the same information, it is called digital multi-signatures [1].

An intuitive approach for constructing a conventional multi-signature for a document consists of three steps: firstly, the same copy of document is sent to all participant signers. Secondly, each participant signer generates his signature by using his private key based on a well-known digital signature scheme, such as RSA's or ElGamal's digital signature algorithm. Finally, the personal signatures are collected to form a multi-signature for the document. The conventional multi-signature has the disadvantages that the size of the multi-signature grows in proportion to the number of signers and the time required by the multi-signature verification is equal to the time required by the multi - signature generation. Thus it makes the generations and the verifications of multi-signatures become inefficient. To overcome the problem, Wu et al. proposed a delegated multi-signature scheme so that participant signers sign only on the subdocuments that they are responsible for[2]. However, Wu's scheme is unable to guarantee the sub-documents decomposed meaningful, and it doesn't consider the logical structure of XML, and it doesn't appropriate for XML documents.

In this paper, we proposed a XML multi-signature scheme base on RSA broadcast digital multi-signature, and utilized the logical structure of XML document. According to the rule of visiting node about XPath language of XML, establishing corresponding relationships of each sub-document and the different signer, and participant signers sign only on the subdocuments that they are responsible for. We designed a frame of a multi-signature system about XML.

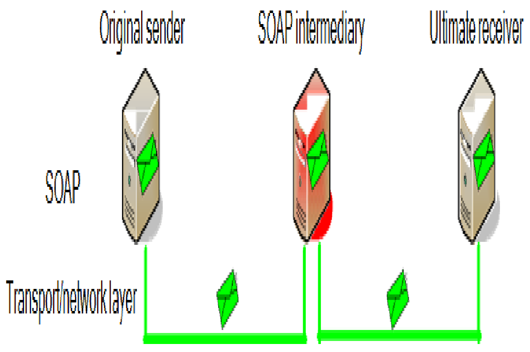

Figure 1: The Transport/Network Layer Security (e.g., SSL/TLS or IPSec) is Broken at the Intermediary SOAP node. By applying security at the SOAP/XML level, on the other hand, end-to-end security can provided.

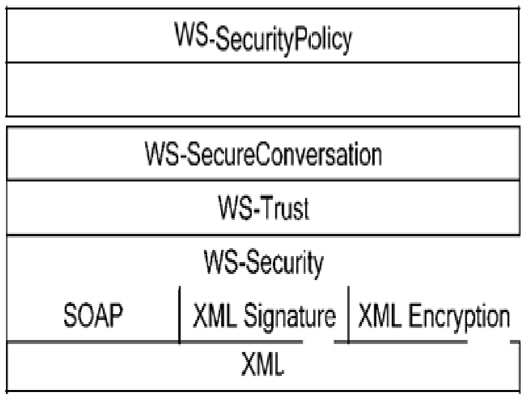

Figure 2: The conceptual relationship between the XML and Web services security standards.

Be aware that Web Services Policy can also be used (independently) for other purposes than security. Also recall that XKMS may provide key management for use with XML Encryption and XML Signature, although this is not shown in the figure.

Encryption. XKMS basically defines simple Web services interfaces for key management, thereby hiding the complex-ity of traditional public key infrastructures (PKIs) from the clients. XML Signature, XML Encryption, and XKMS are all discussed in more detail in Section III.

As noted previously, SOAP is an XML based messaging for-mat. Thus, XML Signature and XML Encryption are obvious candidates for being reused to provide SOAP security as well. As illustrated in Figure 2, this is exactly what WS-Security does. WS-Security specifies how to apply XML Signature and XML Encryption to SOAP messages, effectively providing integrity and confidentiality to SOAP messages (or parts of SOAP messages). As multiple encryptions can be used within the same SOAP message, the different parts of a SOAP message may be encrypted for different receivers (SOAP intermediaries). Likewise, a SOAP intermediary may add an additional signature to a SOAP message, thereby providing integrity protection for a newly added header or supporting separation-of-duty through co-signatures.

In addition to providing confidentiality and integrity for SOAP messages, WS-Security also provides a mechanism to avoid replay attacks (i.e., timestamps) and a way to include security tokens in SOAP messages. Security tokens are typically used to provide authentication and authorization.

WS-Security has no notion of a communication session, that is, it is only concerned with securing a single SOAP message or a single SOAP request/response exchange. In cases where multiple message exchanges are expected, WS-Secure Conversation may be used to establish and maintain an authenticated context. The authenticated context is represented by a URI in a context token and consists of a shared secret that can be used for key derivation. WS-Secure Conversation relies on WS-Trust to establish the security context.

WS-Trust basically defines a framework for obtaining security tokens (including the context tokens used in WS-Secure Conversation) and brokering of trust. With a range of Web services standards, interoperability becomes very difficult unless the communicating parties know what standards to use and how these standards are to be used. Web Services Policy provides the means by which service providers and clients can specify their interoperability requirements and capabilities. WS-Security Policy can be viewed as an extension to Web Services Policy, defining how Web Services Policy can be used to specify requirements and capabilities regarding the use of WS-Security, WS-Trust, and WS-Secure Conversation. For instance, a service provider may specify using WS-Policy/WS-Security Policy that it requires certain message parts to be encrypted.

The last two standards covered in this paper are the Security Assertion Markup Language (SAML) and the eXtensible Access Control Markup Language (XACML). SAML may be used to communicate authentication, attribute, and authorization information in a trusted way. SAML is based on XML and although its original motivation was single sign-on for Web browsing, it is also well suited for use in Web services. XACML on the other hand is used to define access control policies in XML, and may be used to define access control policies for any type of resource. Because SAML and XACML are not targeted exclusively at Web services, SAML and XACML were not included in Figure 2. However, this does not imply that there is no interaction between these standards. A XACML implementation may for instance rely on the security tokens of WS-Security for authentication. As to security tokens, there is also a SAML based security token in WS-Security.

II. An Overview of XML and Web Services Security

XML based SOAP messages form the basis for exchanging information between entities in Web services systems. The information contained within these SOAP messages may be subject to both confidentiality and integrity requirements. Although mechanisms at lower layers may provide end-to-end security, these lower layer mechanisms are often insufficient. This is due to the fact that a SOAP message may be subject to processing and even modification (e.g., removal/insertion of a SOAP header) at intermediary nodes. The result being that the end-to-end security provided by lower layer mechanisms (e.g., SSL/TLS) is broken, as illustrated in Figure 1. Relying on lower layers for end-to-end security may also cause problems if a message is to pass through various networks utilizing different transport protocols. Furthermore, security at the XML level has the advantage of enabling confidentiality and source integrity to be maintained also during storage at the receiving node(s).

XML Signature and XML Encryption are used to provide integrity and confidentiality respectively. Although these two standards are based on digital signatures and encryption, none of them define any new cryptographic algorithms. Instead, XML Signature and XML Encryption define how to apply well established digital signature/ encryption algorithms to XML. This includes:

A standardized way to represent signatures, encrypted data, and information about the associated key(s) in XML, independent of whether the signed/encrypted re-source is an XML resource or not.

The possibility to sign and/or encrypt selected parts of an XML document.

The means to transform two logically equivalent XML documents, but with syntactic differences, into the same physical representation. This is referred to as canonicalization. In order to be able to verify the signature of an XML resource that has had its representation changed, but still has the same logical meaning, it is essential that canonicalization is performed as part of the XML signature creation and verification processes.

As both XML Signature and XML Encryption rely on the use of cryptographic keys, key management is a prerequisite for their effective use on a larger scale. Therefore, the XML Key Management Specification (XKMS) was created to be suitable for use in combination with XML Signature and XML

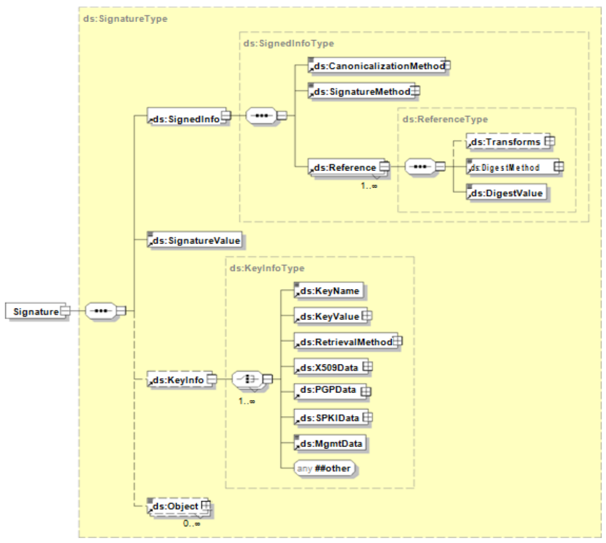

Figure 3: The Signature element. (XML attributes are not shown.)

III. An XML Multi-Signature Scheme

A frame of multi-signature scheme

Generally speaking [1], the digital multi-signature scheme's participant has the news sender (issuer), the news signer (signer), the signature verifier (verifier) and the signature gatherer (collector).

We designed a frame of the multi-signature scheme. There are four modules involved in this scheme: a single signature module (SS), a system authorization module(SA), the documents dispatcher module(DD), and signature collection module(SC). SA provides services such as the initialization of system parameters and issuing the certificate to each signer. DD is responsible for XML documents decomposition and subdocument delegation. These subdocuments assign for every signer by a delegation algorithm. SS produces each signer's signature. SC collects and verifies personal signatures generated by each signer and constructs a multi-signature for the group. Each module's relation is shown in Figure 1[3]. Let $G=\{U_1,U_2,U_3\ldots,U_n\}$ be the registered group, and U_i is one signer of G. Let $M=\{ M_1,M_2,M_3\ldots M_n\}$ be the document to be signed, and M_i be the sub-document of M delegated to U_i by DD.

The whole process for generating a multi-signature in our scheme

The procedure for generating a multi-signature consists of four stages: SA sends certificate to each signer stage, documents division stage, the multi-signature generation stage, and the multi-signature verification stage. Each stage is described as follows.

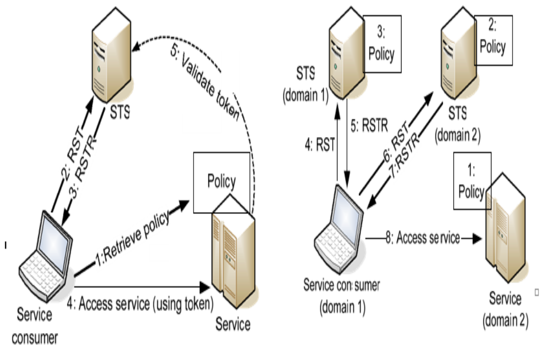

Figure 4: Access Service using token

The policy of the service (which is expressed using Web Services Policy/WS-Security Policy) specifies a security token required to access the service. After retrieving the policy (e.g., from the WSDL file of the service), the service consumer uses the information in the policy to obtain the correct security token from the security token service (STS). The communication with the STS consists of a request security token (RST) element/message and a request security token response (RSTR) element/

message. When receiving the security token from the service consumer, the service may have the token validated by the STS or validate the token itself.

The policy of the service specifies that a security token from the STS in domain 2 is required. The policy of the STS in domain 2 requires a security token from the STS in domain 1 (i.e., one of its policy alternatives accepts such a security token). The STS in domain 1, for which the service consumer has a valid username/password, requires a username token for authentication. After retrieving the policies in the given order (1, 2, and 3), the service consumer obtains a security token, from the STS in domain 1 (4 -5), which is then used to obtain a security token from the STS in domain 2 (6-7). This last security token is then used to access the service (8).

1) SA sends certificate to every signer First, SA generates the private/public key for itself based on RSA algorithm. Each stage of generating key is described as follows.

Step 1. SA chooses *two large prime number p and q ;*

Step 2. *SA calculates n= p*q , n is a modulus of RSA;*

Step 3. *SA calculates Φ(n),Φ(n)= (P-1)(q-1);*

Step 4. *SA chooses e randomly, and e should satisfy 1< e <Φ(n) and gcd (e,Φ(n))=1 . Thus（e, n）is the public key of SA ;*

Step 5. SA *calculates d by ed=1 mod Φ(n).（d, n）is the private key of SA ;*

Let H() is a one-way hash function such as SHA, MD5. In order to distribute certificate to every Ui , SA selects a one-way hash *function and calculates the following formula by using it's private key .*

hi=H(IDi)
si= hi -d(mod n)
Finally, SA sends (IDi,si)as certification *to every Ui∈G*

via a secure channel. *Then every signer calculates hi = si -e(*mod *n) to verify (IDi,si) [4] .*

2) Documents division

Xpath, the XML Path Language, is used to locate information inside an XML document. DD utilizes Xpath to locate a subdocument of an XML document, and decomposes the document M into a set of subdocuments M={M1,M2,…,Mn}. Mi is delegated to Ui .

Suppose that the following XML document is needed to sign.

<?xml version="1.0" encoding="gb2312"?> <order>

<bookinfo>

<title> data structure </title> <author>YAN Wei-min</author>

<publisher>Tsinghua University publishing house </publisher>

<ISBN>0-764-58007-8</ISBN> <price>40.00 </price>

</bookinfo>
<creditcard>
<name>XingShan</name>
<number>123456789</number>
<expiry>7/20/2008</expiry>
</creditcard>
</order>

We can use *Xpath transformation show as <Transform*

Algorithm=http://*www.*w3.org/TR/1999/ REC-xpath-199911 16>

<XPath> descendant-or-self::creditcard </ XPath> </Transform>

The result of the Xpath transformation is shown as <creditcard>

<name>XingShan</name>

<number>123456789</number>

<expiry>7/20/2008</expiry>

</creditcard>

3)The process for generating and verifying a multi**-**signature

In our scheme, let M be the XML document to be cooperative signed by the signers in G. Let T ={t_1 , t_2,…, t_m} be a set of rules that is Xpath expressions. Let T_i and M_i be an element of T and an element of M delegated to user U_i respectively. Through a simple Xpath processor C with the rule T_i, one can easily obtain a subdocument M_i=Ct_i (M).

Step l. Each U_i ∈G chooses a number r_i randomly, and calculates R_i = r_i^e （mod n, then sends R_i to SC.

Step 2. SC calculates R= R_1 *…* R_n（mod n, and broadcasts R.

Step 3. DD sends {H(M),M_i, T_i} and {H (T), H(M)}to U_i and SC, respectively.

Step 4. Each signer U_i verifies M_i which has been received by calculating Ct_i (M). If M_i=Ct_i(M), it indicates that M_i is the sub -document which U_i need to sign. All U_i ∈ G cooperatively check the integrity of M by verifying H(M)=H(M_1||M_2|| … ||M_n)where "||"is the concatenation symbol**.**

Step 5.Each U_i calculates the following formula**.**

$$m_i=H（R, M_i）$$

$$D_i=r_i\, s_i^{\,m}（\mathrm{mod}\ n）$$

Each U_i sends his own signature result D_i to SC(the signature collection module).

Step 6. SC calculates D = $D_1D_2…D_n$ and m= m_1 m_2… m_n , and publishes（D, R, m）as the multi-signature result of M for G.

Signature collection module SC acts as the verifier and SC needs to carry on the following operation.

Step 1. SC calculates h_1 = H（ID_1）,…, h_n = H（ID_n）.

Step 2. SC calculates both $T^*=D^e（h_1h_2…h_n）^m$ m*=H（M, T*）

Step 3. SC checks the result of multi-signature by verifying m = m*. If m = m* is tenable**,** explained the multi -signature correctly.

The security analysis of this scheme

Because of the public ID_i of each signer, anybody can calculate H（ID_i, but can not calculate s_i= h_i^{-d}(mod n) . Because two big prime numbers p, q are unknown, and Φ(n) is also unknown. So nobody can get d from e. The difficulty for solving d is equal to decomposing a great integer.

Suppose that the attacker wants to sign M (the document needs to sign) as a signer U_i, from above signature process, the attacker can choose a number ri’, and calculate Ri'= (r_i’)（^{e}mod n）and m_i’=H（R_i’,M_i）to counterfeit R_i and m_i. But he can not counterfeit D_i, because he can't get s_i, the difficulty for solving the SI is equal to decomposing a great integer[1,4] .

IV CONCLUSION

The conventional digital signature technology can not sufficiently consider the characteristics of XML such as structure and description. Moreover, it can not meet the new XML requirements of more fine-grained encryption and signature, and multiple signatures. This XML documents multi-signature scheme proposed in this paper has fully considered the advantages of XML documents structure, which is based on the RSA broadcasting multi-signature. This scheme uses the Xpath transform rule of the XML correlation technique to compartmentalize the documents, and the signer needs to sign for their own sub-document. This scheme improves the efficiency of the signature and correspondence, and it has certain extension and flexibility.

REFERENCES:

[1] Hao Ze-Mao. theory of digital signature. Beijing of china: science press, 2007.

[2] Tzong-Chen Wu, Shih-Chan Huang, D. J. Guan. delegated multi-signature scheme with document decomposition. journal of systems and software, 2001, 55(3):321-328.

[3] Meng Jiang, Cao Li-Ming design and realization on Multi Signature scheme of xml-based electronic medical record. computer engineering,2006,32 (19):264:266

[4] Zhang Jian-Hong, Wei Yong-Zhuang, Wang Yu-min. digital multi signature scheme based on RSA. journal of china institute of communications, 2003,24(8):150-154

[5] Zhu Sheng-Lin, Xiao De-Qin, Lin Pi-Yuan research and implementation on multi signature of electronic official documents based on xml. journal of south china agricultural university, 2005, 26(1):115-118

[6] Hu Yingsong, Liu Zhipeng. a multi signature scheme with xml document decomposition, net security technologies and application,2006, (7):87-89

[7] Hakim Khali, Ahcene Farah, dsa and ecdsa-based multi-signature schemes. ijcsns international journal of computer science and network security, 2007,7(7) :11-18

* * * * *