Research on the Model of Secure Transmission of SOAP Messages

Haixia Zhao[1], Yaowei Li[2], Mingchuan Zhang[1], Ruijuan Zheng[1], Qingtao Wu[1]

[1] **Electronic & Information Engineering college, Henan University of Science and Technology, LuoYang, 471003, P.R. China**

[2]**LuoYang Electronic Information Equipment Testing Center. China, LuoYang, 471003, P.R. China**

Abstract

SOAP as the basis application of Web Services, and, SOAP messages are closely related to the heterogeneous Web services. Secure transmission of SOAP messages play a vital role for the applicability of Web Services. The main challenges to the secure transmission of SOAP messages includes: confidentiality, authentication, integrity, both-party nonrepudiation, and single sign-on. We analyzed and took advantage of the existing technologies and solutions related to SOAP and Web Services, and proposed a model of secure transmission of SOAP messages, which adopting technologies like XML Signature, XML Encryption, and X.509 Certificate. The analysis in this paper indicates that for the basic requirements towards secure transmission of SOAP messages our model are fulfilled and for the high-level security and efficiency our model are acquired.

Keywords: *SOAP messages, Web Services, secure transmission model, both-party nonrepudiation, single sign-on.*

1. Introduction

Web Services are already a reality for many organizations and are just around the corner for most of the rest of us. One of the core specifications on which Web Services rely heavily is SOAP (Simple Object Access Protocol). In terms of a services-oriented architecture, SOAP is used to send data from one application to another. Web Services make use of SOAP (of course, together with other technologies) to tie heterogeneous business systems together, and as a result, companies can now create and deploy distributed applications without regard to the hardware platform, OS, programming language, or network topology of either party wishing to communicate with the chosen Web Services application. Just like all other network technologies, security is the bedrock for Web Services to enjoy widespread deployment. Without a convincing security model, the Web Services framework would be next to useless[10].

SOAP is an XML-based, simple information exchange protocol applied in dispersed or distributed environment. SOAP's main advantage is loosely coupled[1]. Seen in terms of a service-oriented architecture, SOAP allows for applications to bind to other applications in order to make use of their functionality. SOAP can either be used for messaging between applications (called "Document-based SOAP") or for Remote Procedure Calls (called "RPC SOAP"). Both of messaging and RPCs are the important aspects of SOAP, but in most cases, messaging is preferable to RPC, since it means that applications do not have to share an object model, or rely on a synchronous always-on connection[3]. SOAP is defined as an enveloping protocol, so it is sometimes seen as a messaging protocol as well as a means of using functionality that is published by a remote application.

One of the goals of SOAP designing is simplicity, so security was not taken into account by the SOAP specification. SOAP messaging security relies on the established security concepts and technologies, such as , encryption, digital signature, authentication, and data integrity. This paper is to study the secure transmission of SOAP messages.

2. Related Work

SOAP, which is a messaging protocol based on XML, is about sending messages, meaning that it specifies a way to send XML-based messages from one process to another, usually from one machine to another[8]. More specifically, SOAP is a protocol that specifies an enveloping mechanism for sending data (via XML). Furthermore, it specifies how to send these messages to a final destination, and the processing model that applies if that message goes through several intermediaries. And, it specifies how to do this over HTTP.

The SOAP specification describes four major components: formatting conventions for encapsulating data and routing directions in the form of an envelope, a transport or protocol binding, encoding rules, and an RPC mechanism. The envelope defines a convention for describing the contents of a message, which in turn has implications on how it gets processed. A protocol binding provides a generic mechanism for sending a SOAP envelope via a lowerlevel protocol such as HTTP. Encoding rules provide a convention for mapping various application datatypes into an XML tag-based representation. Finally, the RPC mechanism provides a way to represent remote procedure calls and their return values. As to the structure, a SOAP message consists of an envelope containing an optional header and a required body, as shown in Figure 1. Envelope, the topmost container, comprises the SOAP message; Header contains additional blocks of information about how the body payload is to be processed; and Body contains the actual message to be processed. Each element contained by the Header is called a header block. The purpose of a header block is to communicate contextual information relevant to how the message is to be processed. This includes routing and delivery settings, authentication or authorization assertions, and transaction contexts. XML elements and attributes for the purpose of SOAP security are just placed inside the SOAP header. The body contains the actual message to be delivered and processed. Anything that can be expressed in XML syntax can go in the body of a message.

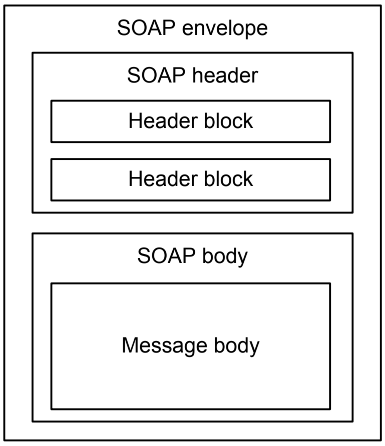

Fig. 1 SOAP message structure.

A SOAP message can be anything: a purchase order, a request for a current stock price, a query for a search engine, a listing of available flights, or any number of other pieces of information that may be relevant to a particular application.

While a SOAP message is fundamentally a one-way transmission of an envelope from a sender to a receiver, that message may pass through various intermediate processors that each in turn do something with the message. The set of intermediaries that the message travels through is called the message path. Every intermediary along that path is known as an actor. SOAP dose specify a mechanism of identifying which parts of the SOAP message are intended for processing by specific actors in its message path. This mechanism is known as “targeting”. Targeting can only be used in relation to header blocks, and the body of the SOAP envelope cannot be explicitly targeted at a particular node. The value of the actor attribute is the unique identifier of the intermediary being targeted. Intermediaries that do not match the actor attribute must ignore the header block[11].

The construction of a message path (the definition of which nodes a message passes through) is not covered by the SOAP specification. Various extensions to SOAP, such as Microsoft's SOAP Routing Protocol (WS-Routing) have emerged to fill that gap. WS-Routing defines a standard SOAP header block for expressing routing information. Its role is to define the exact sequence of intermediaries through which a message is to pass.

3. Proposed Model of Secure Transmission of SOAP Messages

In an enterprise application scenario, along with the involvement of purchase order, services providing, and payment, the information integration among enterprises extends security boundary from intranet to internet. Naturally, the risk of security increases evidently.

3.1 Security Analysis of SOAP messages transmission

The division of information security into logical components makes it easier to understand, and therefore easier to deploy[10]. These logical components, each of which maps a challenge to the security of SOAP messages transmission, are confidentiality, authentication, integrity, and nonrepudiation.

Confidentiality is used to refer to the requirement for data in transit between two communicating parties not to be available to third parties that may try to snoop on the communication. And, confidential information in a SOAP message should remain confidential over the course of a number of SOAP hops[4].

Authentication is an identity-authenticating process. In the web services world, answering the following questions is vitally important:

Who am I?

How do I prove who I am?

Why should you trust me when I tell you who I am?

Who are you?

How can I prove that you are who you say you are?

Why should I trust you when you tell me who you are?

Authentication is just a standard method to ask and answer these questions. And in multiple hops message transit, so called single sign-on is necessary. “Single sign-on”, also called “federated trust”, means the challenge of providing such functionality: enabling a user to sign on once, and then, without having to sign on again, access different domains that would normally be outside the scope of the primary sign-on domain[12].

Integrity has a special meaning in the field of information security. It does not mean that information cannot be tampered with. It means that if information is tampered with, this tampering can be detected. In an untrusted network, it may be impossible to ensure that the data is tamper-proof when it is in transit to its destination. So, knowledge about the fact that tampering has occurred is the next best thing[9].

Nonrepudiation literally means that the originator of a message cannot claim not to have sent a given message[7]. Nonrepudiation, which promises that malicious message sender cannot deny the fact he has sent the message, and so promises that the constructor and sender of the message is same, is vitally important to B2B applications. Furthermore, the nonrepudiation is a both-party concept in the messaging in B2B applications. Besides the attacks launched by the sender and the malicious third-party, malicious receiver attack is to be protected to fulfil both-party nonrepudiation.

3.2 Technologies and Solutions that Address the Security of SOAP Messages Transmission

SOAP does not yet have a standard binding for reliable messaging. The security provided by HTTPS cannot satisfy the more and more complicated requirement of SOAP message security. A number of technologies and solutions have been developed for the security of SOAP message transit. Several vendors offer reliable messaging solutions[6].

XML Encryption provides not only a way of encrypting portions of XML documents, but also a means of encrypting any data and rendering the encrypted data in XML format. XML Encryption is ideal for confidentiality. The ability to selectively encrypt XML data makes XML Encryption very useful for Web Services. By selectively encrypting data in the SOAP message, certain information may be hidden from SOAP intermediaries as it travels from the originator to the destination Web Services[12].

XML Signature explains how to express the digital signature of any data as XML, as well as explaining how to digitally sign portions of an XML document. The power of XML Signature for Web Services is the ability to selectively sign XML data. For example, if a single SOAP parameter needs to be signed but the SOAP message’s header needs to be changed during routing, an XML Signature can be used that only signs the parameter in question and excludes other parts of the SOAP message. If the SOAP request passes through intermediaries en route to the destination Web Service, XML Signature ensures end-to-end integrity[12].

Security Assertions Markup Language (SAML) provides a means of expressing information about authentication and authorization, as well as attributes of an end user in XML format. SAML does not provide authentication, but can express information about an authentication event that has occurred in the past. By authenticating once, being authorized, and effectively reusing that authorization for subsequent Web Services, single sign-on for Web Services can be achieved. If an entity is authorized based on the fact that they were previously authorized by another system, this is called “portable trust[10]”.

The XML Key Management specification (XKMS) enables PKI services such as trustworthily registering, locating, and validating keys through XML-encoded messages. PKI is a system that allows public keys to be trusted by providing key signing and key validation services. Although accepted as an important, even vital, technology, PKI has a reputation for being notoriously difficult to implement. By leveraging the benefits of XML and by learning from past experiences with pre-XML PKI architectures, XKMS makes PKI practical for common use[10].

Microsoft’s Passport technology takes a different approach to single sign-on. The user authenticates to the passport infrastructure, either directly through www.passport.com or through an affiliate site that makes use of functionality provided by passport.com. Once the user is authenticated and authorized by Passport, their authentication status is also available to other Web Services that use Passport[10].

Another industry proposal for the single sign-on on the Web is the Liberty Alliance Project, championed by Sun. The Liberty Alliance Project aims to enable a non-centralized approach to single sign-on, termed a "federated network identity." It appears the Passport proposal by Microsoft may be taking a similar tack to the Liberty Alliance Project[10].

WS-Security, which has emerged as the de facto method of inserting security data into SOAP messages, is primarily for securing SOAP messages. WS-Security explains how technologies such as XML Signature, XML Encryption, and SAML are used for Web Services security in particular. WS-Security defines placeholders in the SOAP header in order to insert security data, how to add encryption and digital signatures to SOAP messages, how security tokens are contained in SOAP messages, and how XML Security specifications are used to encrypt and sign these tokens. In practice, this means defining the XML elements and attributes that are used to enclose tokens into SOAP messages, and the means to enclose XML Signature and XML Encryption into SOAP[5].

3.3 The Architecture and Mechanism of the Secure Transmission Model

A model of secure transmission of SOAP message is developed here to fulfill the security requirement. The building blocks of the model includes: confidentiality, authentication, integrity, both-party nonrepudiation, and single sign-on. Security of the model is achieved through inserting security blocks into SOAP header, as well as adopting technologies such as XML Encryption and XML Signature. Figure 2 is the architecture of the model.

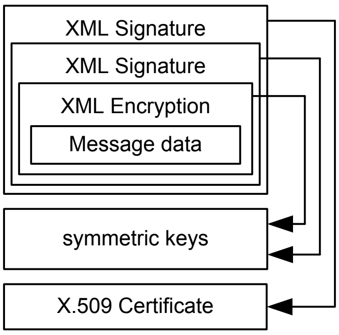

Fig. 2 Architecture of the model

The arrowed lines in Figure 2 represent the reference to the keys or token.

The basic idea is: encrypting the body data using the symmetric keys, signing the encrypted data using the symmetric keys again, and then signing again the signed data, making use of the private key provided by the X.509 certificate of the recipient.

Firstly, XML Encryption is implemented upon the message data, to realize the confidentiality of message data. The result of the encryption to a resource forms EncryptedData, which will replace the original resource being encrypted. How many resources are there to be encrypted, as many EncryptedDatas will be generated. Here, encryption to message data adopts symmetric keys, which are produced randomly every time.

Secondly, XML Signature is implemented to realize the integrity of message data. It includes three steps to construct an XML Signature: to make digest of the object to be signed, to sign the digest using the signature method, and to encrypt the digest, still using symmetric keys. Through decryption, digest verification and signature verification, the recipient can verify the integrity of message.

Finally, another XML Signature is implemented upon the result of first signature, adopting the private key provided by the digital certificate defined in the security block. This additional XML Signature using the digital certificate of the recipient is the key of the model to implement sender's nonrepudiation and single sign-on. Authentication is realized inside the process of single sign-on as part of the latter. The recipient's signing the response message using the digital certificate of the sender is the key of the model to implement recipient's nonrepudiation.

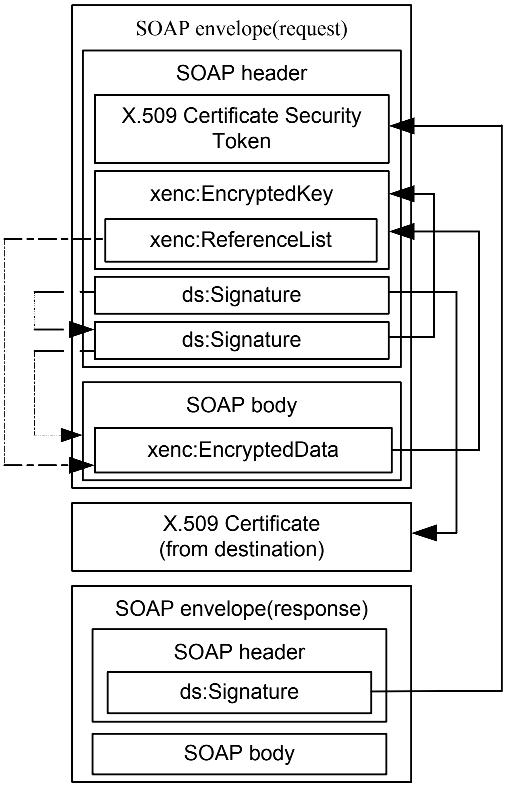

Fig. 3 Mechanism of the model

After being received, the second XML Signature is decrypted and verified. If the integrity is available, the keys, signature method, digest method, and encryption key can be obtained credibly. Thus, the original SOAP information is securely transmitted from the sender to the recipient. The client X.509 certificate and server X.509 certificate supply the asymmetric keys which are necessary in secure transit of the symmetric keys used in XML Encryption and XML Signature.

Figure 3 shows how the security mechanism of the model is established. In Figure 3, the arrowed solid lines represent the reference to the keys or token, well the arrowed dashed lines represent the secure operation.

An example of the implementation of the preceding process is listed as following.

```
<?xml version="1.0" encoding="utf-8"?>
<SOAP-ENV:Envelope xmlns:
SOAP-ENV="http://www.w3.org/2001/12/soap-
envelope"
 xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
 xmlns:xenc="http://www.w3.org/2001/04/xmlenc"

xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/ut
ility">
 <SOAP-ENV:Header>
  <wsse:Security

xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/sece
xt">
   <wsse:BinarySecurityToken wsu:Id="X509token"
    ValueType="#X509v3"
    EncodingType="#Base64Binary">
    ……
   </wsse:BinarySecurityToken>
   <xenc:EncryptedKey wsu:id="userSysmetricKey">
    <xenc:EncryptionMethod
     Algorithm="……"/>
    <ds:KeyInfo>
     <wsse:SecurityTokenReference>
      <wsse:Reference URI="#userSysmetricKey"
      ValueType="......"/>
     </wsse:SecurityTokenReference>
    </ds:KeyInfo>
    <xenc:CipherData>
     <xenc:CipherValue>…...</xenc:CipherValue>
    </xenc:CipherData>
    <xenc:ReferenceList>
     <xenc:DataReference   URI="#DataBeEncrypted
"/>
    </xenc:ReferenceList>
   </xenc:EncryptedKey>
   <ds:Signature wsu:id="originSignature">
    <ds:SignedInfo>
     <ds:CanonicalizationMethod
Algorithm="……"/>
     <ds:SignatureMethod Algorithm="……"/>
     <ds:Reference URI="#BodyData ">
      <ds:DigestMethod Algorithm="……"/>
      <ds:DigestValue>…...</ds:DigestValue>
     </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>…...</ds:SignatureValue>
    <ds:KeyInfo>
     <wsse:SecurityTokenReference>
      <wsse:ReferenceURI="#userSysmetricKey"
      ValueType="......"/>
     </wsse:SecurityTokenReference>
    </ds:KeyInfo>
   </ds:Signature>
   <ds:Signature>
    <ds:SignedInfo>
     <ds:Reference URI="#originSignature">
      <ds:DigestMethod Algorithm="……"/>
      <ds:DigestValue>…...</ds:DigestValue>
     </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>…...</ds:SignatureValue>
    <ds:KeyInfo>
     <wsse:SecurityTokenReference>
      <wsse:Reference URI="X509token"
     </wsse:SecurityTokenReference>
    </ds:KeyInfo>
   </ds:Signature>
  </wsse:Security>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body wsu:Id="BodyData">
<xenc:EncryptedData
   wsu:Id="DataBeEncrypted"
   type="……">
   <xenc:EncryptionMethod Algorithm="……"/>
   <CipherData>
    <CipherValue>…...</CipherValue>
   </CipherData>
  </xenc:EncryptedData>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4. Security Analysis of the Secure Transmission Model

XML Encryption to the SOAP message body realizes the confidentiality of the data, and XML Signature to the encrypted data realizes the integrity of the data. Evidently, the encrypting(decrypting) speed of symmetric keys is much faster than the encrypting(decrypting) speed of asymmetric ones. The practice that the symmetric keys used in XML Encryption and XML Signature are produced randomly is securer than the symmetric keys produced using hashing method, because as for the latter, once the keys were captured, succedent transmission would lose security. Through introducing the both-party X.509

certificate (those of client and server), which contain both the asymmetric keys and the identity information of the entity, the preceding security transmission model of SOAP messages acquires a high-level transmission security, as well as enjoys the benefit of high efficiency. The solution to single sign-on is to include information about the end user in the SOAP message itself. Furthermore, by making use of the identity information of the entity, the transmission mechanism realizes both-party nonrepudiation and single sign-on. The model itself is simple and light, but its running requires the support of certificate release of requester and responser. That is, the main load of the whole work is borne by certificate infrastructure. This maybe represents the shortcoming of it.

5. Conclusion and Expectation

Aiming at the challenges that SOAP messages transmission faces in Web Services applications among enterprises, a simple and light transmission model is developed, based on existing technologies. Analysis indicates that the model fulfils the security requirement of SOAP messages transmission: confidentiality, authentication, integrity, both-party nonrepudiation, and single sign-on, enjoying well advantage and efficiency. The cost of this kind of advantage and efficiency is the deployment of X.509 digital certificate on applications communicating via SOAP messages.

It is important to keep the entire security context of the Web Service in mind. This includes properly configured firewalls, the use of patched and locked-down Web servers, and (especially if digital certificates are used) the use of an adequate security policy document. It would be foolish to address just the new security challenges posed by Web Services and leave a system open to attack through more traditional channels.

There is a lot of work to do to strive for higher security of SOAP messages transmission, or even Web Services. To heighten the security and efficiency of the model, a particular block can be inserted into the SOAP header. Add a mustUnderstand="true" attribute to the header block, and require that the recipient must understand it. If this flag is present, and the recipient does not understand the block to which it is attached, the recipient must reject the entire message. In addition, the model developed in this paper should be strengthened to avoid the risk of reply attack.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (NSFC) under Grant No. U1204614, No. 61003035 and No. 61142002, and in part by the Plan for Scientific Innovation Talent of Henan Province under Grant No. 124100510006, and in part by the Science and Technology Development Programs of Henan Province under Grant No. 112102210187, and in part by the Youth Foundation of Henan University of Science and Technology under Grant No. 2011QN51.

References

[1]David Chappell,Tyler Jewell, Java Web Services, O'Reilly, March 2002, 28-50.
[2]Dongxi Zheng, Shaohua Tang,Shaofa Li, "XML Web Services Security Technology Overview", Computer Engineering and Application, 2004.7, 38-41.
[3]Doug Tidwell, James Snell, Pavel Kulchenko, Programming Web Services with SOAP, O'Reilly, December 2001, 39-61.
[4]IBM developerWorks. http://www.ibm.com
[5]International Business Machines Corporation, Microsoft Corporation,VeriSign, Inc., Web Services Security (WS-Security) Version 1.0, April, 2010.
[6]Jian Jin, Hong Zhang, Jiahua Liang, Hualin Qian, "Analysis of Web Services Security", Micro-electronics and Computer, 2004.3, No3, Vol 21, 19-24.
[7]Jimei Wang, Lianfu Jin, "RESEARCH AND RESOLUTION ON WEB SERVICE SECURITY", Computer Applications and Software, February, 2004, No 12, Vol 21, 91-93.
[8]Keith Ballinger, .NET Web Services: Architecture and Implementation, Addison Wesley, February, 2003, chapter 9.
[9]Luciano Baresi, Elisabetta Di Nitto, Test and Analysis of Web Services, Springer, March 2011, 395-440.
[10]Mark O'Neill et. Web Services Security, McGraw-Hill/Osborne , 2003, chapter 3,4,5,9.
[11]Xiaoning Xu, "Security Study on transport of SOAP messages on Web Services", Information Security, 2011, No 11-3, Vol 22, 115-117.
[12]ZDNetChina community. http://www.zdnetchina.com

Haixia Zhao received her B.S. degree from South West Normal University in 1998 and M.S degree from National University of Defense Technology in 2005. She works as a Lecturer in Henan University of Science and Technology from 1998 to now. In particular, her research interests include wireless sensor networks, Internet of Things, cognitive network, database theory and technology etc.

Yaowei Li received his B.S. degree from National University of Defense Technology in 1998 and M.S degree from National University of Defense Technology in 2004. He works as a Engineer in LuoYang Electronic Information Equipment Testing Center from 1998 to now. In particular, his research interests include Information security, Internet of Things, cognitive network etc.

Mingchuan Zhang received his B.S. degree from Luoyang Institute of Technology in 2000 and M.S degree from Harbin Engineering University in 2005. He works as a Lecturer in Henan University of Science and Technology from 2005 to now. In particular, his research interests include ad hoc network, Internet of Things, cognitive network and future Internet technology.

Ruijuan Zheng received her B.S. degree from Henan University in 2003, studied in Harbin Engineering University from 2003 to 2008, and received Ph.D. degree. She works as an Associate Professor in Henan University of Science and Technology from 2008 to now. In particular, her research interests include bio-inspired networks, Internet of Things, future Internet and computer security.

Qingtao Wu received his Ph.D. degree from East China University of Science and Technology. He works as an Associate Professor in Henan University of Science and Technology from Mar 2006 to now. His research interests include component technology and future Internet security.