WUJNS
Wuhan University Journal of Natural Sciences
Vol.9 No.5 2004 755-759

Article ID: 1007-1202(2004)05-0755-05

Design and Implementation of Web Services Security Based on Message Layer

WANG Cui-ru, XU Zheng-wei, YUAN He-jin, MA Hui-min

School of Computer Science and Technology, North China Electric Power University, Baoding 071003, Hebei, China

Abstract: Along with the development of Internet, Web Services technology is a new branch of Web application program, and it has become a hotspot in computer science. However, it has not made great progress in research on Web Services security. Traditional security solutions cannot satisfy the Web Services security require of selective protection, end-to-end security and application layer security. Web Services technology needs a solution integrated in Web Services framework to realize end-to-end security. Based on cryptography and Web Services technology and according to W3C, XML encryption specification, XML digital Signature specification and WS-Security, which proposed by IBM and Microsoft, a new Web services security model based on message layer is put forward in this paper. The message layer is composed of message handlers. It is inserted into the message processing sequence and provides transparent security services for Web Services. To verify the model, a Web Services security system is realized on .net platform. The implementation version of the model can provide various security services, and has advantages such as security, scalability, security controllability and end-to-end security in message level.

Key words: Web services; Web services security; message layer

CLC number: TP 393.08

Received date: 2004-06-15

Biography: WANG Cui-ru (1954-), female, Professor, research direction: database and information management system. E-mail: Wangcuirul@eyou.com

0 Introduction

Web Services technology is a new branch of Web application program with the properties of self-containing, self-description and modularization and can be released, located and called by Web. It is composed of a series of protocols and criterions, such as XML, SOAP, WSDL and UDDI. It is an innovatory technology based on XML, and its naissance will have great influence on traditional software design. A new generation of Web application will upgrade to Web Services-oriented design pattern. And besides, due to XML's flexibility and platform independent function, Web Services will inevitably get broad application in many fields, such as E-commerce, system integration and platform-independent operation etc[1].

However, Web Services security is just like western region waiting for exploration and development. The current circumstance is that software development based on Web Services is very simple, but it is very difficult to achieve Web Services on the safe side. So, the urgent affairs are to design and achieve secure Web Services system.

1 Deficiencies of Existing Security Technologies on Protecting Web Services

The existing security technologies are quite mature on protecting Web information, and these technologies can be adopted directly to protect Web Services security. The primary technologies include SSL (Secure Sockets Layer), VPN (Virtual Private Network), Firewall technology and HTTP authentication. However, as a new technology, Web Services

technology has its special security requirement, so there are certain deficiencies of existing security technologies on protecting Web Services:

1) SSL (Secure Sockets Layer)

At present, SSL (Secure Sockets Layer) is essentially a security communication standard on Web. But it has certain deficiencies mainly in performance, intermediate nodes and selective protection etc[2].

2) VPN(Virtual Private Network) technology

VPN technology can provide good protection for Web Services, but its application is limited in a finite scope. To those large enterprises that have built VPN, Web Services can be protected without increasing investment. But when we want to release services to the Web, we aren't able to demand everyone use VPN. Furthermore, to construct VPN needs to purchase corresponding software and hardware, which would increase system expenditure. Therefore, VPN can only be used as an auxiliary method and it cannot realize universal security[3].

3) Firewall technology

By using firewall, IP addresses trying to access illegally can be shielded off. If an organization uses proxy server to get Web Services, all requirements from it would show a same IP address. If this IP address were shielded off, those legal users' access would also be restricted. And besides, other security services, such as digital signature and encryption, could not be realized yet. So, firewall technology is also regarded as an auxiliary method to protect Web Services[4].

4) Authentication technology

There are many kinds of security authentication mechanisms at present, and different authentication mechanisms usually are in different protocol layers. Applying these mechanisms often needs complicated equipments and installations, and considering the different software environments it will become more complicated. The security of Web Services demands an integrated extendible authentication mechanism. So the existing authentication mechanisms cannot satisfy this require ment [5].

On all accounts, new security requirements presented by Web Services technology are: selective encryption, end-to-end security, security based on application layer and security integrated in Web Services system. Apparently, the present security technologies cannot meet all these requirements.

2 Web Services Security Model Based on Message Layer

2.1 Presentation of the Model

Aiming at the characteristics of Web Services security requirements and combining the existing security models, this paper puts forward a Web Services security system model based on message layer. The architecture of the model is given in Fig. 1.

Fig. 1 shows that this model adds a message processing layer at both Web server side and client side. And some relative components to enhance security, such as digital certification library and user's database, are added too. Each layer's function is introduced in detail as follows [6-10].

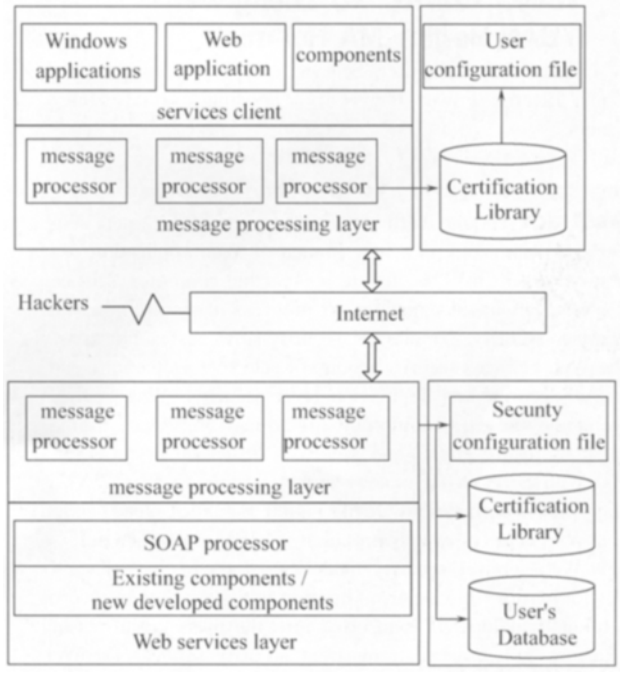

Fig. 1 Web Services security model based on message layer

1) Services client layer

This layer is Web Services' client layer, and its primary functions are shown in two aspects. On the one hand, when client requests services, it serializes the request information to SOAP message of XML format and sends the message to the next layer. On the other hand, when it receives message, it parallelizes the message to corresponding objects and passes them to client to call methods and deal with data. Different from traditional application programs, client mentioned here is multi-

form, besides traditional C/S and B/S application programs, and it also can be components of enterprise application system or other Web Services.

2) Client message processing layer

This layer's responsibility is to deal with corresponding message and request information. It is composed of a series of message processors including encryption/decryption processor, signature/verification processor etc. And each group of processors can realize certain function. Of course, this layer provides a special message processing mechanism and users can define message processor conveniently, thus the custom-built commercial logic can be realized. In order to implement security processing, the layer must acquire some additional messages and among them the most important message is user's cipher key.

3) Server message processing layer

Similar to client message processing layer, this layer's responsibility is to perform security processing on SOAP message but the operations here is more complicated. Not only encryption/decryption processing and signature/verification processing are needed by security, but also user's databases are required to authenticate users, implement access control policy, register accessing users and form log etc.

4) Web services layer

Web Services layer is mainly composed of SOAP processor and transaction components. SOAP processor is in charge of formatting the request information to object-calling information, and then calling corresponding transaction components. Afterwards, the transaction components get calling results and transform them to the message with XML format and create response message, then through security processing layer send the message to services client.

From Fig. 1 we know that security processing has been implemented before sending message to client. Even if Web Services hackers captured the message and fabricated or destroyed it, message-processing layer could easily discover whether the message were true or false and modified.

2.2 Message Processing Workflow

In Fig. 2 we give a client message processing workflow from sending out requirement to receiving response information.

Steps of message processing are as follows:

1) Client sends out the original object-calling information and serializes it to XML document by SOAP formatting program.

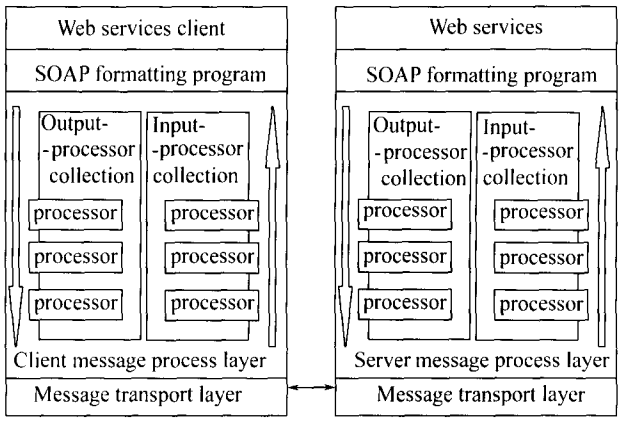

Fig. 2 Web service message processing workflow

2) When request information passes through message processing layer, the corresponding processor will pick up system parameters and process the message. The typical operation is that adding user's authentication information, putting the digital signature to the message and encrypting them. Finally, some compressing work can be done too. After these processes the message will be transferred to the transport layer.

3) Request information which has been preprocessed at upwards steps is sent to client through some transport layer protocols, such as HTTP etc.

4) When Web server receives the request information, it first decompresses the message with decompression processor, then authenticates users, and finally, decrypts the message and validates the digital signature.

5) The message processed by a set of input processors is to be translated into original request information and then becomes corresponding object-calling information through SOAP formatting program.

6) The response message repeats the upwards steps and is transmitted to the client.

The preceding part is a typical request/response message model, from which we can know that message processing layer captures the message and deals with them and makes them have a secure format both before message sending and after receiving. Consequently, the message is secure during the course of message sending.

3 Realization of Web Services Security Model

3.1 Environment of the System

We adopt Micorosoft. NET as system realizing envi-

ronment and Windows 2000 Server as operating system. Our development tool is Visual Studio. NET 2003.

NET adapts CryptoAPI into System. Security. Cryptography namespace and makes cryptography services get rid of mystery of SDK platform and become simple using of . net namespace. Owing to sharing with the whole frame components, it is much easier to realize cryptography services. Classes in System. Security. Cryptography namespace can realize almost all the kinds of algorithms. If users want to realize their algorithms themselves, the basic framework is also supplied. And users can easily realize relative algorithms by extending corresponding basic classes[1].

3.2 Realization of Web Services Security Component Library

3.2.1 Implementation scheme

Implementation of processors and XML security criterion in the model are encapsulated in a component library ultimately, and the physical form is shown as a file named MonicaSoft. WebServices. dll. Users can use this file to redevelop secure Web Services.

The root namespace of this component library is MonicaSoft. WebServices. Basic classes and interfaces are defined in this namespace to support secondary development, and the main parts of which include security component deployment interface class, message processor model class, message processing layer class, security message integrated processor and access control processor which are implemented ultimately and so on. Interfaces to process class of XML elements are also defined in the namespace.

In the root namespace, there is a sub-namespace, MonicaSoft. WebServices. Security namespace. In this namespace, there are another three namespaces: Encryption, Signature and Token, and they represent a set of components that are used to realize XML encryption, XML digital signature and security token. This part is kernel code of the whole Web Services security component library and it realizes three criterions about XML and Web Services concretely, and they are XML encryption criterion, XML digital signature criterion and WS-Security criterion.

The implementation of the component library mainly includes implementation of security token, XML data encryption criterion, XML digital signature, message processing layer and access control policy. Limit to the length of the article, we only introduce the implementation of message processing layer.

3.2.2 Implementation of message processing layer

The basic unit of message processing layer is message processor. It is the least function unit to process message but very important. There are two types of message processor: InputMessageProcessor and OutputMessageProcessor. Accordingly, two abstract classes, SoapInputFilter and SoapOutputFilter, are defined. We have also defined abstract method of message processing, ProcessMessage, in both two classes. The practical task of message processor is to realize self-defined message processing mechanism in that method. For example, user can define an EncryptionOutputFilter processor and in its ProcessMessage method to get encryption algorithms and cipher key by accessing user's database and set other properties, and to call Encrypt method of EncryptedData object. Thus, encryption processing is done. And message processor defines a Collection class to manage several kinds of message processors. This class is integrated in CollectionBase class to realize ICloneable interface.

Message inputting processing layer includes two objects: output-message-processor collection and input-message-processor collection. There are two methods to deal with input message and output message: ProcessInputMessage() and ProcessOutputMessage ().

During user calling Web Services methods with SOAP protocol, method-calling is first serialized to SOAP message and then sent to the server, finally server anti-serializes SOAP message to method-calling and executes corresponding operation. The process that client gets results from Web Services is similar to that mentioned upwards. In order to extend SOAP message, net provides a SoapExtension class. Users can add their own processing to change SOAP message before or after message serialization and anti-serialization by inheriting the class, and they can even do it to both client side and server side at the same time. By this way, we can solve the interface problem between message processing layer and Web Services.

Fig. 3 is a sketch map of Web Services life-span. According to it, by the means of capturing SOAP message in the different stages of life-span, message processing layer can gain the goal of processing message transparently.

On the basis of the definition of SoapExtension class, SecurityExtension class is realized. And we use message layer objects to process message in it. Class dia-

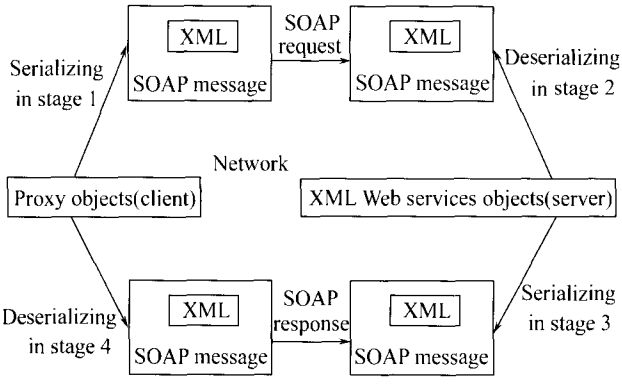

Fig. 3 Life-span of Web Services message

gram of SecurityExtension class is given in Fig. 4.

Each Web Service has a file, Web. config, which is a XML document. By setting the document's corresponding XML configuration nodes, SecurityExtension class can be inserted into message processing sequence, and thereby, message layer can process message transparently.

SecurityExtension
_baseStream: Stream _newStream: MemoryStream _pipeline: Pipeline
_GetInitializer(in methodInfo: LogicalMethodInfo, in attribute:SoapExtensionAttribute): object _GetInitializer(in type: Type): object _Initialize(in initializer:object) _InitializePipeline() _ProcessMessage(in message: SoapMessage) _ChainStream(in stream: Stream): Stream _AfterDeserialize(in message: SoapMessage) _AfterSerialize(in message: SoapMessage) _AfterSerializeClient(in message: SoapMessage) _AfterSerializeServer(in message: SoapMessage) _BeforeSerialize(in message: SoapMessage) _BeforeDeSerialize(in message: SoapMessage) _BeforeDeSerializeServer(in message: SoapMessage) _BeforeDeSerializeClient(in message: SoapMessage)

Fig. 4 Class diagram of Web Services interface classes

4 Conclusion

According to W3C, XML encryption criterion proposed by Microsoft and IBM, XML digital signature criterion, XML security criterion (WS-Security etc) and security technology such as access control technology, this paper presents a Web Services security model based on message layer and realizes a Web Services security system in . net platform. After analyzing and validating, this system has shown a good many advantages in lots of aspects, such as security, extensibility, service transparency, security controllability and end-to-end security in message level and it is proved with important significance to Web Services security.

References

[1] Banerjee A, Corera A. Net Remoting Architecture. *C# Web Services—Building Web Services with. NET Remoting and ASP. NET*. Beijing: Tsinghua Press, 2002. 114-159(Ch).

[2] Shi Wei-peng, Yang Xiao-hu. SOAP-Based Fundamental Security Specification of Web Service (WS-Security). *Computer Application Research*, 2003,**20**(2): 100-102, 105 (Ch).

[3] Zhang Shuang, Shi Hao-shan. Study on The VPN System Realization Technique. *Computer Engineering*, 2002,**28**(8): 276-278(Ch).

[4] Chen Wei-wei, Wang Qing-xian. FireWall Technology and Vulnerability Analysis. *Computer Applications*, 2003, **23** (10): 46-48(Ch).

[5] Neuman B, Ts'o T Kerberos. An Authentication Service for Computer Networks. *IEEE Communication Magazine*, 1994, **32**(9): 33-38.

[6] Zhu Yu, Deng Xiao-yan, Shao Pei-nan. A Solution Based on XML to Application Security. *Computer Engineering*, 2003, **29**(2): 180-181,203(Ch).

[7] Microsoft. Security in a Web Services World: A Proposed Architecture and Roadmap. *http://www-900. ibm. com/ developerWorks/cn/Webservices/ws-secmap/index_eng. shtml*, 2002.

[8] Bartel M, Boyer J. XML-Signature Syntax and Processing. *http://www. w3. org/TR/2001/PR-xmldsig- core-20010820/*, 2001.

[9] Atkinson B, Della-Libera G. Web Services Security(WS-Security). *http://www-900. ibm. com/ developerWorks/cn/ Webservices/ws-security/index-eng. shtml*, 2002.

[10] Imamura T, Dillaway B. XML Encryption Syntax and Processing. *http://www. w3. org/TR/xmlenc-core/*, 2001.

□