Security Interaction of Web Services in Heterogeneous Platforms

Xu Tao[1], Hu Xin[2], Xie Jiwen*[3], Sun Shujuan[4]
[1,3]College of Computer Science and Technology, Civil Aviation University of China, Tianjin, 300300, China
[1,2,3]Information Technology Research Base, Civil Aviation Administration of China, Tianjin, 300300, China
[1,3,4]College of Computer Science, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China
*Corresponding author, e-mail: txu@cauc.edu.cn[1], xhu@cauc.edu.cn[2], xiejiwenpower@163.com[3], sunshujuan87@163.com[4]

Abstract

Currently, there are a large number of heterogeneous platforms. The standards of Web Services in different platforms are different and complex. Therefore, security interaction of Web services based on heterogeneous platform has become increasingly prominent. In order to realize security interaction of heterogeneous platforms, a security interactive model of Web Service based on WebSphere and .NET is proposed in this paper. The model adopts an approach based on predicate logic to integrate the security policies of heterogeneous platforms and uses the integrated policy to sign the SOAP message. The experimental results show that the model can ensure the safety of SOAP message transmission and realize the security session between these two heterogeneous platforms.

Keywords: *heterogeneous platforms, security interaction, SOAP message, web services*

1. Introduction

As the focus of IT industry in recent years, SOA (Service Oriented Architecture) has gradually become the guiding idea of developing IT systems. SOA advocates an idea that system components developed in different platforms and different techniques can be combined rapidly and freely. These components are stand-alone and each component can perform certain functions independently [1, 2].

Generally, traditional integration solution of applications is business-oriented and information-oriented, which is difficult to suit demands changed with the rapid development of business. SOA rebuilds the existing systems and designs a new application system from the view of software architecture. Consequently, it supports to implement the Enterprise Application Integration (EAI) dynamically. In addition, SOA makes the enterprise become more elastic and flexible and can quickly respond to the variations of business requirement, so that the real-time enterprise and dynamic enterprise can be realized.

With the mature standards of Web Service and the popularity of its application, Web Service provides the basis for widespread implementing SOA [3]. It realizes a real sense of platform-independent and language-independent. Yet it brings challenges to secure issues. With the wide use of Web Service, security interaction of Web Services has become increasingly difficult to achieve. Therefore, the specifications on security interaction of Web Services are drawn up and constantly updated [4].

Currently, there are a large number of Web Services Security Specification (WS-Security), including WS-Addressing, WS-Security, WS-Reliable Messaging (WS-RM), WS-Secure Conversation (WS-SC) and so on. These specifications ensure the security of Web Service from different views and ranges. Application Servers providing supports for Web Service are also put more and more attention to interoperability. But different application servers provide different security mechanisms for Web Service [5, 6]. Therefore, how to achieve security interaction between applications running under different environments becomes a difficult problem.

The interaction between J2EE and .NET is an important part of Web Service

Received September 5, 2013; Revised November 4, 2013; Accepted November 21, 2013

Interoperability [7], how to resolve the interaction between them, therefore, has become a serious problem. In this paper, security interaction of heterogeneous platform based on WebSphere and .NET is studied. By analyzing the security mechanisms of heterogeneous platforms, a security interactive model of Web Service in heterogeneous platforms is proposed. This model unifies security polices of heterogeneous platforms by converting the security policy described in XML into assertion represented in predicate. On this basis, with an example of ticket reservation service, the safe handling of SOAP (Simple Object Access Protocol) messages in heterogeneous platforms is achieved. This security model provides theory support for the security interaction of Web Services in heterogeneous platforms. By experimental verification, the model can ensure the security interaction of Web Services effectively.

2. Security Mechanisms of Heterogeneous Platforms

One of the characteristics of SOA is that it allows each service to use its respective technology and platform. In this case, in order to achieve the interaction among services, the definitions of service contract and the communication protocol must comply with the industry standards [8]. As a design idea, SOA does not specify the implementation method to achieve interaction of Web Services. Therefore, software companies have launched their own products for SOA.

2.1. Security Mechanism of .NET

WCF (Windows Communication Foundation V3.0) is a complete technical framework designed by Microsoft for SOA. It supports the industry standards and the core protocols of Web Service [9]. And it unifies a variety of distributed technologies produced by Microsoft [9], including:

1) Web Services and WSE [10].
2) .NET Remoting.
3) .NET Enterprise Services.
4) Microsoft Message Queue (MSMQ).

The architecture of WCF (shown as Figure 1) includes the following aspects such as contract, message, and runtime behavior of service, host and so on.

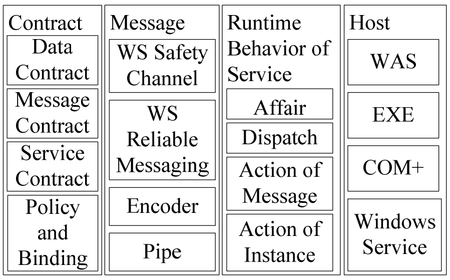

Figure 1. Architecture of WCF

In the architecture of WCF, the specific definition of transmission for SOAP message is defined in policy and binding of the contract. Binding defines a communication mode with the outside, which is consisted of a set of binding elements which are combined to form a communication infrastructure. The binding contains the following aspects:

1) Communication protocols, such as HTTP, TCP, etc.
2) Message encoding mode, such as binary coding, MTOM, etc.
3) Security strategies of message.

The graphics will stay in the “second” column, but you can drag them to the first column. Make the graphic wider to push out any text that may try to fill in next to the graphic.

2.2. Security Mechanism of WebSphere

WebSphere Application Server (WAS) V7.0 supports and extends the WS-Security specification:

1) In addition to the basic types of tokens (Username and X509) [11], WAS also supports customized type of tokens, such as LTPA token.

2) WAS allows to add timestamp in the signature and encryption.

3) WAS supports the certification cache mechanism, and then improves the efficiency.

WAS V7.0 proposes the concept of policy set. When users choose different policy sets for applications, the QoS (Quality of Service) is different. Since WAS V7.0 emphasizes the separation of the Web Service and the security policy, developers only need to consider the business logic without knowing the security-related details. After deploying the application, administrator can configure appropriate policy set for the application according to the needs in the management console or in the manner of Jython, Jacl script. When the security demand changes, administrator just needs to reconfigure the policy set without changing the program code.

3. Security Interactive Model on Heterogeneous Platforms

3.1. Security Interactive Model of Web Service in Heterogeneous Platforms

Security frameworks and configurations for different platforms are quite different, and each platform uses its own security policy and technology to meet the security requirements [12]. Therefore, in order to achieve the security interaction of Web Services in heterogeneous platforms, a middleware or an agent must be added. The middleware can unify security configurations of heterogeneous platforms, and then use the agreed configuration for the security processing of SOAP message. In order to achieve safe handling of SOAP message exchanging between heterogeneous platforms, a security service proxy is added between WebSphere and .NET. This proxy includes a security policy integrated module. Figure 2 shows the security interactive model of Web Service in heterogeneous platforms.

3.2. Security Policy Integrated Module

WCF can use the configuration file to ensure the security of Web Service, that is, through setting appropriate authentication in customized binding. WCF has five authentication modes which are shown as below:

1) UserNameOverTransport;
2) MutualCertificate;
3) UsernameForCertificate;
4) AnonymousForCertificate;
5) Kerberos.

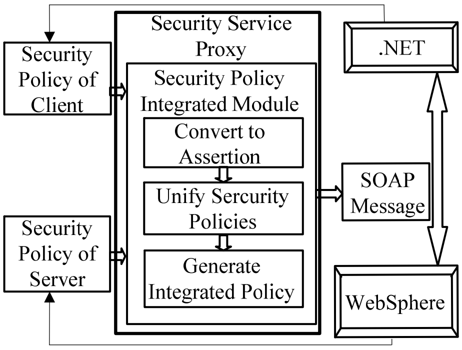
Security Service Proxy
Security Policy of Client
Security Policy Integrated Module
Convert to Assertion
Unify Sercurity Policies
Generate Integrated Policy
Security Policy of Server
.NET
SOAP Message
WebSphere

Figure 2. Security Interactive Model of Web Service in Heterogeneous Platforms

WAS doesn’t have the concept of authentication mode. In order to achieve the security interaction between WCF and WAS, a feasible authentication mode should be chosen from the five modes to generate a solution. In this paper, authentication mode MutualCertificate is selected to achieve the security interaction. This mode supports X.509 authentication and

SOAP extension.

Security policies of WCF and WAS are different not only in the content, but also in the realization form. Security policy of WCF is in the .exe.config document, but security policy of WAS is in policySet.xml.

For achieving security interaction between WCF and WAS, the security policy integrated module is used to analysis the security policies of different platforms and generate an integrated security policy.

In order to achieve the integration of security policies, an approach based on predicate logic to convert security policy from XML description into predicate is adopted. A security policy can be classified into three types of security requirements: signature, encryption and security token. A predicate signature (shown as Figure 3) is used to describe the signature requirement, where sigId is the Id of the signature for a variable in variable list var, and the tokenId is the Id of a security token used for signature. The signature algorithm salgo, the transform algorithm talgo, and the digest algorithm dalgo are used in this approach. Similarly, the predicate encryption is defined for the encryption requirement, where kalgo is the algorithm used for key encryption and dalgo is the algorithm used for data encryption. The predicate token is for the security token requirement, where variable *t* represents the token type, such as X509v3.

```
signature(m:Msg,sigId:String,var:List,tokenId:String,
          salgo:SigM,talgo:TransM,dalgo:DigM)
encryption(m:Msg, encId:String,var:List,tokenId:String,
          kalgo:KeyEncM,dalgo:DataEncM)
token(m:Msg, tokenId:String, t:TokenType)
```

Figure 3. Security Policy Assertion

Both the WCF and WAS use XML to describe the security policy. Therefore, they can be converted into assertion shown as Figure 3. According to the rules of generating integrated security policy, security policies of heterogeneous platforms can be unified, then the integrated security policy is used to sign and encrypt SOAP message. Figure 4 shows the rules for integrating signature requirements. Predicate isIntegrityConsistent is a constraint for consistency of data integrity. It returns true when the variable cVar of the client’s operation cOpp and the variable sVar of the server’s operation sOpp have the same signature requirements. The predicate requestIntegrity returns true if a variable requires integrity.

```
isIntegrityConsistent(cOpp:Operation,cVar:String,
                      sOpp:Operation,sVar:String,salgo:SigM,
                      talgo:TransM,dalgo:DigM,t:TokenType)
requestIntegrity(sOpp:Operation, sVar:String,salgo:SigM,
                 talgo:TransM, dalgo:DigM,t:TokenType)
```

Figure 4. Rules for Integrating Signature Requirements

Similarly, the other requirements (such as encryption and security token) can be defined.

4. Ticket Reservation Service

The application scenario is designed as follows (shown in Figure 5). The user sends a request message to the travel service; the SOAP message contains the basic information (such as username, password, etc.) and authentication token. The user must be authenticated before invoking the airline service. The travel service and the airline service run on different technology platforms, therefore their security policies and technical supports are different.

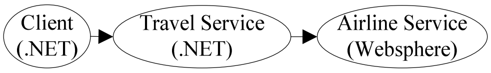

Figure 5. Scenario of Ticket Reservation Service

The basic security demands of the scenario are:

1) The travel service and the user, the travel service and the airline service can validate each other.

2) In the process of transmission, confidentiality and integrity must be ensured.

4.1. Variables of Ticket Reservation Service

Table 1 lists the variables in the process of the ticket reservation service. The userMsg is the information that is submited to travel service by the user. The customerID and password represent the user's name and password respectively. The airMsg is the information that is submited to the airline service by the travel service. The airlineNo represents the flight number of the user.

Table 1. Variables of Ticket Reservation Service

Message	Variables
userMsg	customerID password
airMsg	airlineNo

4.2. Integrate the Security Policies

1) Security Policy of Travel Service

The travel service needs to interactive with both the user and the airline service, so two security policies are required. The travel service and the client run on the same platform, therefore, their security policy assertions are the same. Figure 6 shows the security policy assertion of the travel service, SOAP message that the user provides to travel service includes user's name and password. The signature algorithms rsasha1, the transform algorithm exc14n, the digest algorithm sha1, the encryption algorithm exc14n, and the data encryption algorithm sha1 are used. The SOAP message is signed by username token.

```
signature(userMsg, 'upi:sigID1',['upi:customID', 'api:password'],
           'api:username', rsasha1, exc14n, sha1)
token(userMsg, 'upi:unID', username)
```

Figure 6. Security Policy Assertion between Travel Service and Client

The security policy assertion between the travel service and the airline service is shown in Figure 7.

```
signature(airMsg, 'api:sigID1','api:airlineNo','api:x509ID',
           exc14n, rsasha1, exc14n, sha1)
token(airMsg, 'api:x509ID', x509V3)
```

Figure 7. Security Policy Assertion between Travel Service and Airline Service

2) Security Policy of Travel Service

The security policy of airline service (shown in Figure 8) specifies that all the variables are signed with x509v3, and user's SAML token is needed.

```
signature(airMsg, 'api:sigID1','api:airlineNo','api:x509ID',
            exc14n, rsasha1, exc14n, sha1)
token(airMsg, 'api:x509ID', x509V3)
token(userMsg, 'upi:samlID', saml)
```

Figure 8. Security Policy Assertion of Airline Service

3) Security Policy of Travel Service

There are three variables in the ticket reservation service and each one has a relevant solution. By combining these three solutions, an integrated security policy assertion is generated. And in the process of integrating security policies, it is found that the airline service needs the user’s SAML token through the predicate token, but what the user provides to the travel service is username token. Therefore, the username token needs to be mapped as SAML Token.

4.3. Experimental Results

Figure 9 shows the SOAP message with SAML token captured from the airline service server by using TCP/IP monitor.

Through Figure 9 it can be seen that the SOAP message has been successfully signed. And according with the integrated security policy, the transform algorithm exc14n, the digest algorithm sha1, the signature algorithms rsasha1 and X509 token for signature are used.

```
<soapenv: Envelope xmlns: soapenv="...">
<soapenv:Header>
  urn:oasis:names:tc:SAML:1.0:cm:bearer
</saml:ConfirmationMethod>
</saml:SubjectConfirmation></saml:Subject>
</saml:AuthenticationStatement>
<ds:Signature xmlns:ds="http://www.w3.org/.../xmldsig#"
 Id="uuid7f146542-012a-1e00-9226-ab3d2abc819f">
<ds:SignedInfo>
<ds:SignatureMethod Algorithm="http://...dsig#rsa-sha1">
</ds:SignatureMethod>
<ds:Reference URI="#Assertion-uuid7f14…819f">
<ds:Transforms> <ds:Transform
  Algorithm="http://.../xmldsig#enveloped-signature">
</ds:Transform>
<ds:Transform Algorithm="http://.../xml-exc-c14n#">
<xc14n:InclusiveNamespaces
 xmlns:xc14n="http://www.w3.org/.../xml-exc-c14n#"
 PrefixList="saml"></xc14n:InclusiveNamespaces>
</ds:Transform> </ds:Transforms>
<ds:DigestMethod Algorithm="http://.../xmldsig#sha1">
< /ds:DigestMethod>
<ds:DigestValue>kqp...VieIm5c=</ds:DigestValue>
</ds:Reference> </ds:SignedInfo>
<ds:SignatureValue>
   YzcMUQobkHj9tw0ipc0vF...EnIaCj9X3chtTXg=
</ds:SignatureValue>
<ds:KeyInfo> <ds:X509Data>
<ds:X509Certificate>MIICBzCCAX..xdhikBMZPgdyQ==
</ds:X509Certificate>
</ds:X509Data> </ds:KeyInfo>
</ds:Signature></saml:Assertion>
<p:endpointURL xmlns:p="http://localhost:9081/
   TicketReservation/AirlineReservationService"
   xmlns: ns0="…"  xmlns:xsi="…"
   xsi:type="p:urlAddress"> </p:endpointURL>
</soapenv:Header>
<soapenv:Body>...</soapenv:Body>
</soapenv:Envelope>
```

Figure 9. SOAP Message with SAML Token

5. Conclusion

In this paper, the security issues of Web Service in the SOA architecture are studied. In order to solve these issues, a security interactive model of Web Service for heterogeneous platforms is proposed. The model adopts an approach based on predicate logic to integrate the security policies of heterogeneous platforms, and then uses the integrated policy to sign the SOAP message. By experimental verification, this model can ensure the secure transmission of SOAP message. However, the illustrative example involved in this paper only realizes the signature of message. For encryption and access control, appropriate solution method has not been proposed, which will be a study emphasis in further research.

Acknowledgements

The work is supported by the National Natural Science Foundation of China: "Information Exchange in Civil Aviation Integration Application Based on SOA(No. 60979011)" and Tianjin Research Program of Application Foundation and Advanced Technology: "Security Technology Research on Information Exchange in Service-Oriented Architecture (No. 09JCYBJC02300)".

References

[1] Chai Xiaolu. Web Service Technology, Architecture and Applications. Beijing: Electronic Industry Press (in Chinese). 2003; 6-9.

[2] Sam Weber, Paula Austel, Michael McIntosh. *A Framework for Multi-Platform SOA Security Analyses*. Proceedings of IEEE International Conference on Web Service. 2007; 102-109.

[3] Jin Songchang, Jin Songhe, Yang Shuqiang, et al. Design of a Parallel and Distributed Network Security Simulation Platform. *Telkomnika Indonesian Journal of Electrical Engineering*, 2013, 11(6): 3178-3186.

[4] John Viega. *Why Applying Standards to Web Services is not Enough*. Proceedings of IEEE Security and Privacy. 2006; 25-31.

[5] Ji Hongbin, Zhao Fengyu, Xu Tao. *Security Policy Configuration Analysis for Web Services on Heterogeneous Platforms*. Proceedings of International Conference on Service Science, Management and Engineering. 2010; 182-185.

[6] Gao Yan, Zhang Shaoxin, Zhang Bin. SOA-Based Web Services Composition System. *Journal of Chinese Computer Systems*. 2007; 28(4): 729-733.

[7] Robert Bunge, Sam Chung, Barbara Endicott Popovsky, et al. *An Operational Framework for Service Oriented Architecture Network Security*. Proceedings of the 41st Hawaii International Conference on System Sciences. 2008; 312-320.

[8] Ma Anfeng, Zhao Fengyu. On axis2 Web Service Security Based on Rampart Module. *Computer Applications and Software*. 2009; 26(9): 31-33.

[9] Steve Resnick, Richard Crane, Chris Bowen. Essential Windows Communication Foundation. Beijing: Posts & Telecom Press (in Chinese). 2009; 21-25.

[10] Xu Tao, Yi Chunxiao. SOAP-Based Security Interaction of Web Service in Heterogeneous Platforms. *Journal of Information Security*. 2011; 2(1): 1-7.

[11] Xu Tao, Yi Chunxiao. *Signature and Encryption on Parts of SOAP Message Based on Rampart*. Proceedings of 2nd International Conference on Intelligent Systems and Applications. 2010; 1218-1223.

[12] Muhammad Imran Tariq. Towards Information Security Metrics Framework for Cloud Computing. *International Journal of Cloud Computing and Services Science*. 2012; 1(4): 209-217.