Scientific Research

SOAP-Based Security Interaction of Web Service in Heterogeneous Platforms[*]

Tao Xu, Chunxiao Yi

College of Computer Science and Technology, Civil Aviation University of China, Tianjin, China
E-mail: txu@cauc.edu.cn, yibin128@126.com

Abstract

With the development and application of SOA technology, security issues of Web services based on heterogeneous platform have become increasingly prominent. The security of SOAP message is of great importance to Web service security. In order to solve the security issue of heterogeneous platforms, a security processing model named SIMSA (Security Interactive Model based on SOAP and Authentication) based on SOAP and authentication is proposed in this paper. By experimental verification, the model ensures the safety of SOAP message transmission and enhances the security of Web service in heterogeneous platforms.

Keywords: SOAP, Heterogeneous, Web Service, SIMSA, Security Interaction

1. Introduction

With the growth of the Service-oriented Architecture (SOA) application scale, there are hundreds of services in a large company. Different services may be deployed to platforms from different vendors, and different services installed in different locations have different access rights and security policy (encryption, signature, prevention of attacks, etc.). How to guarantee the security of services has become hot spot in foreign research institutions and scholars. IBM Tokyo Research Institute (Fumiko Satoh *et al.*) puts forward the best practice models and support tools for the specific service safety profile construction for the IBM Websphere Server according to security policy using mapping rules [1]. Microsoft Research in University of Cambridge (Karthikeyan Bhargavan *et al.*) [2] publishes a security policy configuration guidance to help developers construct the security policies of Web service according to security requirements. IBM Research Division in New York (Sam Weber *et al.*) [3] points out that there are a large number of heterogeneous platforms and different platforms have many Web Service standards and complex technologies. Even if there is a variety of "best security practices mode", it is still very difficult to ensure how to achieve the proper se-

curity.

Although SOA has solved the Web services called in heterogeneous platforms, and there are relevant security standards (WS-Security) of security information exchange between different platforms, WS-Security only gives an abstract framework to achieve security goals, including XML signatures, encryption, authentication and authorization. As for how to use them to achieve the goal of SOA security, it presents a challenge both in theoretical and technical practices [4,5].

Authentication policies and SOAP message-based security interactive study of Web services in heterogeneous platforms have been proposed in this paper. First the security feature of heterogeneous platforms is analyzed, and then the details of the security interaction model of heterogeneous platform named SIMSA is given. Combined with concrete application examples, user authentication during a Web service call as well as the safe handling of SOAP messages in heterogeneous platforms is achieved. The security model provides theoretical support for the security interacts of Web services in heterogeneous platforms and is verified by experiments. This model ensures the security interactions of Web service effectively.

2. Security Features of Heterogeneous Platform

SOA needs a wide range of interoperability between ser-

vices. In the development of Web service's logic functionality, if the security features are designed, Web services will become extremely complex and service performance and scalability will be greatly reduced. In terms of the analysis on the security needs and consideration of a variety of security measures, it can not be sure whether the security components are appropriately organized and whether the system is more secure [6,7]. Security issues of Web services in SOA should be out of service functions, and the security requirements of Web services can be achieved through appropriate security configuration and mechanism. In this way, it can not only ensure the simplicity of design and call of Web services, but also can achieve the security of SOAP messaging [8,9].

An application system is usually based on a platform such as Microsoft. NET or Apache Axis. The service platform has its own security solution such as Microsoft's WSE (Web Service Enhancement) and Axis's of Rampart, etc. For the same security policy such as using certificates to sign the message, it can be achieved in the same platform for service requester to sign the message and service provider to do signature verification. If the service requester and provider are in different platforms, the security interoperability can not be guaranteed.

Each application platform has its own security mechanisms and security API. When using SOA to integrate enterprise application, services of different applications may be deployed in different application platforms and security requirements may be achieved by different security policy and platform technology. Then it needs an agent mechanism dealing with service security to package the specific realization of platform security from the logic, so that the security SOAP information of heterogeneous platform can be consistently understood and treatment.

Security processing mechanisms of Web services in heterogeneous platform provide security policy configuration and security implementation method of SOAP message [4]. Security service agents use WS-Security and other specifications to achieve the following three aspects of security requirements of SOAP message [10,11]:
- Message integrity

WS-Security takes XML signature to do digital signature for SOAP message to ensure that SOAP message passes through intermediate nodes without being tampered.
- Message confidentiality

WS-Security uses XML Encryption to encrypt the SOAP message, so that the message sender can ensure that the contents of SOAP message can be achieved by the intended recipient uniquely. In this way, even if SOAP messages are listened, listeners can not extract confiden-

tial information from the messages.
- Message authenticity

WS-Security introduces the concept of security tokens, which can represent the identity of the message sender. Combined with digital signatures, the message recipient can confirm the legitimacy and authenticity of the SOAP message sender.

3. Security Model for Heterogeneous Web Services SIMSA

Security framework and configuration strategies for heterogeneous platforms are quite different. Therefore, in order to achieve the security interaction of Web service in heterogeneous platforms, a third-party certification agency must be added. It can complete the relevant certification according to the request of the client. After the verification, client could send a request to call the Web service. To ensure the safety of service call process, both the client and Web server set the security service agent module to conduct safe handling to SOAP messages in the service interaction, including the signature and encryption of the SOAP message. The authentication module of client user is added to Web server, and only after the verification can client call the Web service. In this way, the security interactions of Web services in heterogeneous platforms can be achieved.

Combined with WS-Security specification, a security model of Web services in heterogeneous platform named SIMSA is constructed in this paper (shown in **Figure 1**). SIMSA model is mainly based on the extension of SOAP header including signature, encryption and authentication. The various components and functions of the security model are described as follows.

3.1. SIMSA Model Composition

- UDDI Server

Its main function is to store service descriptions by category. It can be a private registry, such as Capeconnect's UDDI Registry server, or it can also be a public registry, such as IBM Corporation and Microsoft's UDDI registry.
- Third-Party Certification Agency

It is used to verify the client's identity information, and only the users who pass through the authentication can send service requests to the Web server.
- WSDL Builder

It is used to describe how to use SOAP to invoke the Web service, and its function is to generate the corresponding WSDL document.
- Security Service Agent

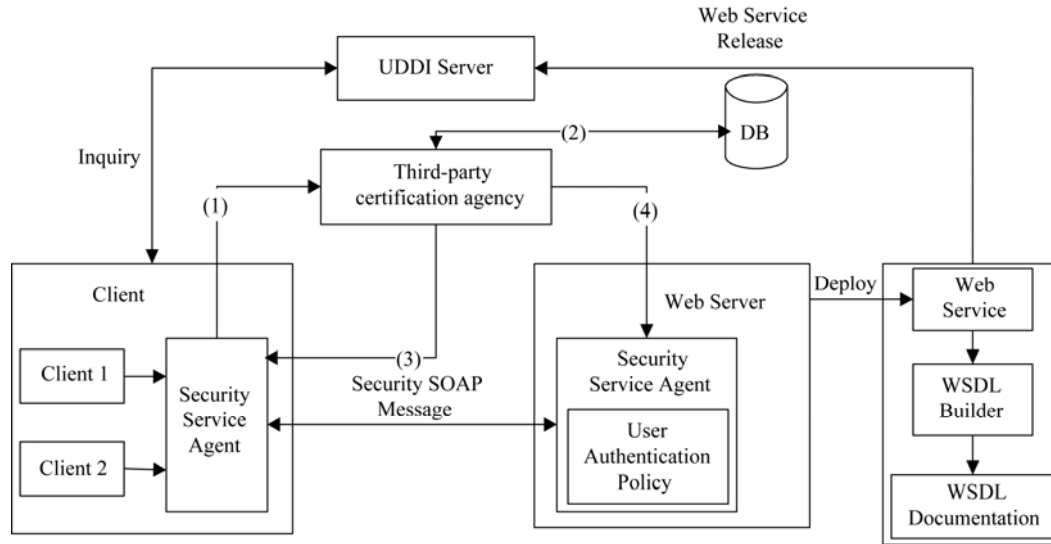It is the core module of the model. It is responsible for

Figure 1. SIMSA Model.

the security of Web service during transmission and achieves the security requirements of the model includeing signature and encryption of the SOAP message.

- User Authentication Policy

It is also the core module of the model. It is responseble for the request verification of client's identity, and only authenticated users can call the appropriate Web service.

3.2. Third-Party Certification Agency of SIMSA Model

In order to provide reliable authentication information for the Web server, SIMSA adds a third-party certification agency. Certification agency will compare the requestor's information such as usernames, passwords and permissions and other information with that stored in the certification database (DB in **Figure 1**). It can provide users with the information needed to verify to invoke Web services. This information is encapsulated in an encrypted message, and this message will be sent to the Web server with the user's SOAP message, waiting for the server's validation. In order to formally describe the process of third-party certification agency, **Table 1** defines the parameters and function description.

The formal description of certification process in third-party certification agency is shown as follows (the following numbers correspond to that in SIMSA):

1) Client's security service agent sends user information to the third-party certification agency:

$$CSP \rightarrow AC : SOAP(IDc, PWDc, IDs) \qquad (1)$$

2) Third-party certification agency gets the user information (including user name, password, etc.) and

Table 1. Parameter description of certification.

Abbreviation	Content
AC	Third-party certification agency
CSP	Client's security service agent
SSP	Web server's security service agent
Message	The encrypted message that third-party certification agency return to the client's security service agent
Key	The encryption key that third-party certification agency uses to encrypt the Message
IDc	Client ID
IDs	Web server ID
PWDc	Client password
SOAP(Head, Body)	SOAP message
COMPARE( )	Certification agency compares requestor's information with that in the database

compares them with that in the database:

$$AC : COMPARE(IDc, PWDc) \qquad (2)$$

3) After the comparison, if properly the third-party certification agency will return a message encrypted with the *Key*, and otherwise it will reject the user's authenticcation request.

$$AC \rightarrow CSP : SOAP(Message) \qquad (3)$$

4) The third-party certification agency sends the *Key* used to encrypt the *Message* to the Web server:

$$AC \rightarrow SSP : SOAP(Key) \qquad (4)$$

4 T. XU *ET AL.*

3.3. Web Server's User Authentication of SIMSA Model

When the client's security service agent receives the en-crypted Message that the third-party certification agency returns, the client can use the Message to invoke the Web service in the server. The SOAP message that client's security service agent sends to the Web server's security service agent includes the client ID and Message, and its formal description is:

$$CSP \rightarrow SSP : SOAP\big(Head\big(IDc, Message\big), Body\big) \quad (5)$$

When the SOAP message including the client ID and the Message reaches the Web server's security service agent, the security service agent must verify the SOAP message whether the client has the permission to call the Web service. In the process it will verify whether the ID included in the Message is the same as the client claims, if they are the same client could invoke the Web service. A user authentication policy is designed in the server's security service agent for the authentication of the client ID. The user authentication policy is shown in **Figure 2**.

The process of Web server's user authentication is:

1) Extract the *Key* used to encrypt *Message* from the SOAP message that third-party certification agency sends.

2) Extract the client ID and *Message* from the SOAP message that client sends.

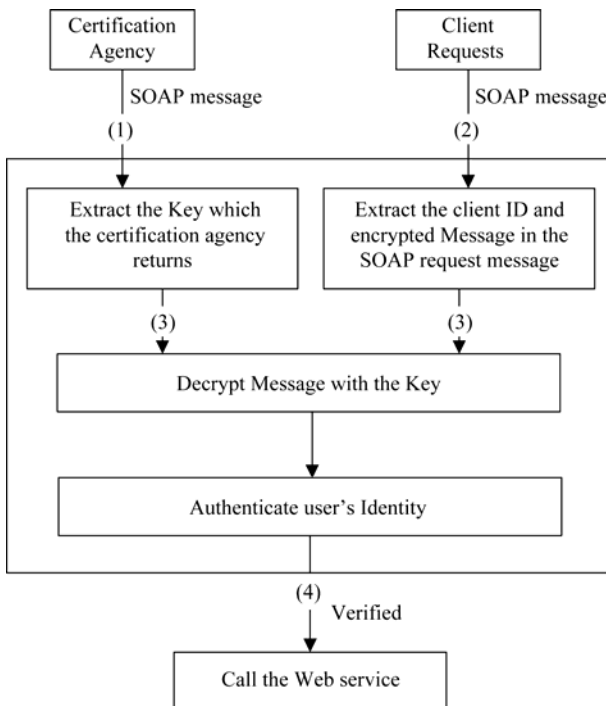3) Use the *Key* to decrypt the *Message* and extract the client ID that the *Message* contains.

Figure 2. User Authentication Policy.

4) Verify whether the ID that client claims is the same as that in the *Message*, if they are the same, the client will be allowed to call the Web service, otherwise au-thentication will fail.

3.4. Security Service Agent of SIMSA Model

After the user's request passes through the identity vali-dation in the Web server, a connection is established be-tween the Web server and client, and customers can call the Web service. In order to ensure the security of Web services exchanged between heterogeneous platforms, the SOAP messages transmitted between the server and cli-ent must be handled safely. The security service agent module of SIMSA implements these security require-ments, and it achieves security interaction of end to end in message-level mainly through the security extension of SOAP message. This module is to realize the signa-ture and encryption of SOAP message. **Figure 3** shows the security service agent module of SIMSA.

To formally descript the security interaction process of SOAP messages between heterogeneous platforms, **Ta-ble 2** defines the relevant parameters and their functions.

The simplest form of security interaction of SOAP message is that a signed and encrypted Web service re-quest M1 is sent to the server security service agent from the client security service agent and corresponding to that the server security service agent will return a re-sponse message M2 which has been handled safely to the client security service agent. The security interaction pro-

Table 2. Parameter description of security interaction in heterogeneous platform.

| Abbreviation	Content
S,C	Web server and client built on different platforms
Cc	Client certificate
Cs	Server certificate
Pu(cert)	Public key of cert
Pr(cert)	Private key of cert
Mes	SOAP Message
S(Pr(cert), Mes)	Use the private key of cert to sign the Mes
Dm	The digital signature of Mes
VS(Mes, Pu(cert), Dm)	Use the public key of cert to validate the digital signature of Mes
E(Pu(cert), Mes)	Use the public key of cert to encrypt the Mes
DE(Pr(cert), Mes)	Use the private key of cert to decrypt the Mes

Copyright © 2011 SciRes. *JIS*
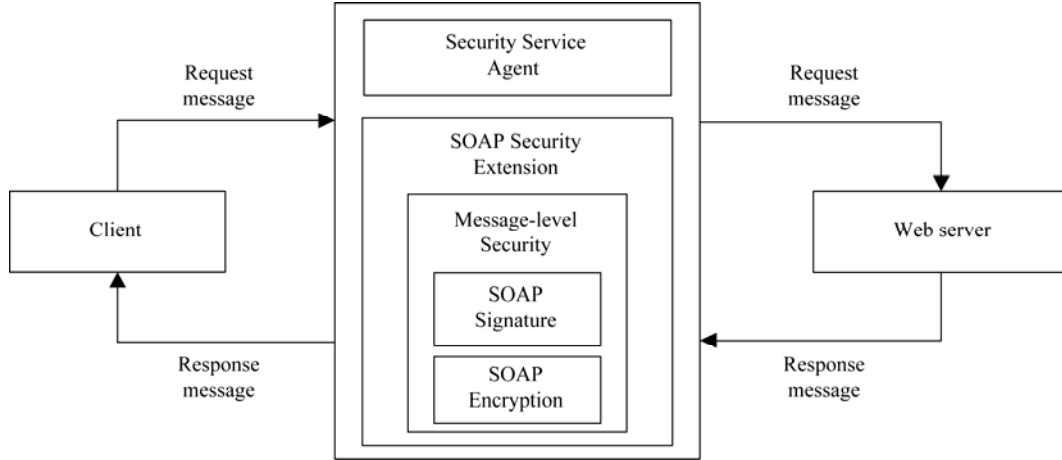
Figure 3. Security Service Agent Module.

cess of SOAP message can be formally described as:

1) Client request: encrypt and sign the request message.

$$C \rightarrow S : E\big(Pu(Cs), M_1\big), S\big(\mathrm{Pr}(Cc), M_1\big) \qquad (6)$$

2) Server: verify the signature and decrypt the request message.

$$S : VS\big(M_1, Pu(Cc), Dm_1\big), DE\big(\mathrm{Pr}(Cs), M_1\big) \qquad (7)$$

3) Server response: encrypt and sign the response message.

$$S \rightarrow C : E\big(Pu(Cc), M_2\big), S\big(\mathrm{Pr}(Cs), M_2\big) \qquad (8)$$

4) Client: verify the signature and decrypt the response message.

$$C : VS\big(M_2, Pu(Cs), Dm_2\big), DE\big(\mathrm{Pr}(Cc), M_2\big) \qquad (9)$$

4. Security Interactive Example

The user authentication module and security service agent module of the SIMSA are realized in this paper. They implement the user authentication in heterogeneous platforms and security extension of SOAP message during the process of Web service interaction.

4.1. User Authentication Implementation

The client sends the request SOAP message to the third-party certification agency. The message's format is shown in **Figure 4**, in which <authentication> shows the information required certification agency to verify; <clientID> specifies the client ID; <password> specifies client password and <serverID> specifies server ID.

The third-party certification agency queries the database to verify the client request SOAP message and after that it returns an encrypted SOAP message including the

client ID and the encrypted *Message* to the client. The message's format is shown in **Figure 5**, in which <Message> contains the encrypted information.

The third-party certification agency sends the encryption key which is used to encrypt the *Message* to the Web server. The encryption key can be used to decrypt the *Message* in the user authentication security strategy when the client calls the Web service. The SOAP message's format is shown in **Figure 6**.

```
<soapenv:Header xmlns:wsa="http://www.w3.org/2005/
    08addressing">
    <authentication>
        <clientID>admin</clientID>
        <password>admin123</password>
        <serverID>server</serverID>
    </ authentication>
</soapenv:Header>
```

Figure 4. Client Request SOAP Message.

```
<soapenv:Header xmlns:wsa="http://www.w3.org/2005/
    08addressing">
    <ToClient>
        <clientID>admin</clientID>
        <Message>encrypted message…</Message>
    </ ToClient>
</soapenv:Header>
```

Figure 5. SOAP message that certification agency returned to the client.

```
<soapenv:Header xmlns:wsa="http://www.w3.org/2005/
    08addressing">
    <ToServer>
        <key>Key</key>
    </ ToServer>
</soapenv:Header>
```

Figure 6. SOAP message that certification agency returned to the server.

Copyright © 2011 SciRes.   *JIS*

```
<?xml version='1.0' encoding='utf-8'?>
<soapenv:Envelopexmlns:soapenv="http://www.w3.org/2003/
05/soap-envelope">
<soaenv:Header xmlns:wsa="http://www.w3.org/2005/
08/addressing">
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org
  /wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
  " soapenv:mustUnderstand="true">
  <ds:Signature>
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://
        www.w3.org/2001/10/xml-exc-c14n#" />
      <ds:SignatureMethod Algorithm="http://www.w3.org
        /2000/09/xmldsig#rsa-sha1" />
      <ds:Reference URI="#id-8303462">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org
          /2001/10/xml-exc-c14n#" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org
        /2000/09/xmldsig#sha1" />
        <ds:DigestValue>q0ut1qK9WER7MXSuX
        4vV4wYS3oQ=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>
      SKPKP5ICsX/lcZzCdYxk0cAsQV6Gbyau0bBJvpq
      NKL/kSyh9KvUMJIJ7i96gT46tCdexHne+LzE2CO
      1xUkBLDv8+zX049Klk++BdqiuZLF6PB/X79dqyd
      RlWYOMYuN2nMvP5Qdo3MzYOvvi2K7w3gcbiy
      euDwmWglkeR8iCqHvk=
    </ds:SignatureValue>
    <ds:KeyInfo Id="KeyId-11463270">
      <wsse:SecurityTokenReference xmlns:wsu="http://
      docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
      wssecurity-utility-1.0.xsd" wsu:Id="STRId-367156">
      <ds:X509Data>
        <ds:X509IssuerSerial>
          <ds:X509IssuerName>CN=Sample Client,OU=
            Rampart,O=Apache,L=Colombo,ST=Western,
            C=LK</ds:X509IssuerName>
          <ds:X509SerialNumber>1187603652
          </ds:X509SerialNumber>
        </ds:X509IssuerSerial>
      </ds:X509Data>
      </wsse:SecurityTokenReference>
    </ds:KeyInfo>
  </ds:Signature>
  </wsse:Security>
  <wsa:To>http://10.6.233.177:8080/axis2/services/Signiture.
  signitureHttpSoap12Endpoint/</wsa:To>
    <wsa:MessageID>urn:uuid:22EC2A05BDC41A98491289
    284864538</wsa:MessageID>
  <wsa:Action>urn:echo</wsa:Action>
 </soapenv:Header>
<soapenv:Body>
  <xenc:EncryptedData>
  ......
  </xenc:EncryptedData>
 </soapenv:Body>
</soapenv:Envelope>
```

Figure 7. Security SOAP Message.

4.2. SOAP Message Signed and Encrypted by Security Service Agent

The SOAP message output from the server security ser-

vice agent is signed and encrypted. The signed part of the message is shown in **Figure 7**. <ds:Signature> mainly consists of three parts: <ds:SignedInfo>, <ds:Signature Value> and <ds:KeyInfo>. <ds:SignedInfo> also contains three parts including <ds:SignatureMethod>, <ds:Digest Method> and <ds:DigestValue>. <ds:SignatureMethod> indicates the algorithm that signature used; <ds:Digest Method> indicates the algorithm need to be used to generate the abstract data; <ds:DigestValue> specifies the abstract data. <ds: SignatureValue > points out the signature value. <ds:KeyInfo> shows the information of the certificate which signature uses, including the data of the X.509 certificate and the information of the certificate publisher. <soapenv:Body> contains only one element named <xenc:EncryptedData> which indicates the encrypted information. As the encrypted information is too large, the part of the information is omitted.

From **Figure 7** it can be seen that the SOAP message has been successfully signed and encrypted which ensures the security of SOAP messages transmitted between different platforms.

5. Conclusions

SOA promotes the application and integration of information technology, but the security of application and integration is much more complex. In connection with SOA application and integration practice, the security issues of Web service in the SOA architecture have been proposed. In order to solve these issues, a security interactive model of heterogeneous platform named SIMSA is designed. This model realizes security requirements during the process of calling Web services in heterogeneous platform. By making client authentication, signing and encrypting SOAP message in the process of Web service interaction in heterogeneous platform, it achieves the security interaction of Web service in heterogeneous platform, which greatly enhances Web service's security features.

6. References

[1] F. Satoh, *et al.*, "Adding Authentication to Model Driven Security," *IEEE International Conference on Web Services* (*ICWS*), Chicago, 2006, pp. 585-594. doi:10.1109/ICWS.2006.25

[2] K. Bhargavan, C. Fournet, *et al.*, "An Advisor for Web Services Security Policies," *Proceedings of the* 2005 *workshop on Secure web services*, New York, 2005, pp. 1-9. doi:10.1145/1103022.1103024

[3] S. Weber, P. Austel and M. McIntosh, "A Framework for Multi-Platform SOA Security Analyses," *IEEE International Conference on Web Service*, Salt Lake City, 2007, pp. 102-109.

[4] J. Viega, "Why Applying Standards to Web Services is

not Enough," *IEEE Security and Privacy*, Vol. 4, No. 4, 2006, pp. 25-31. doi:10.1109/MSP.2006.110

[5] Z. P. Liu, D. D. Zhou, L. Y. Xue, X. M. Chang and X. J. Song, "A Security Model of Web Service Based on SOAP," *Journal of Wuhan University* in Chinese, Vol. 52, No. 5, 2006, pp. 570-573.

[6] L. Y. Tang and S. H. Qing, "Administration of Multiple Roles in the Hybrid RBAC-DTE Policy," *Chinese Journal of Computers*, in Chinese, Vol. 29, No. 8, 2006, pp. 1419-1425.

[7] X. M. Wang and Z. T Zhao, "Role-Based Access Control Model of Temporal Object," *Acta Electronica Sinica*, in Chinese, Vol. 33, No. 9, 2005, pp. 1634-1638.

[8] W. F. Zheng, T. Xu and Q. F Gu, "Design and Implementation of Core Service in Civil Aviation Integrated Information Platform," *Computer Engineering*, In Chinese, Vol. 34, No. 21, 2008, pp. 267-269.

[9] R. Bunge, S. Chung, B. Endicott-Popovsky and D. McLane, "An Operational Framework for Service Oriented Architecture Network Security," *Proceedings of the 41st Hawaii International Conference on System Sciences*, Waikoloa, 2008, pp. 312-320.

[10] N. Bieberstein, S. Bose, M. Fiammante, K. Jones, R. Shah and Z. Ning, "Service-Oriented Architecture Guide," in Chinese, Posts & Telecom Press, Beijing, 2008, pp. 160-166.

[11] Z. P. Liu, X. M. Chang, D. D. Zhou and X. J. Song, "A Safe ID Authentication Policy in Web Service," *Journal of Computer Research and Development*, in Chinese, Vol. 43, 2006, pp. 551-555.