Investigation on Implementation Method of Web Services Security Proxy

Yi Zhuang[1], Xuechen Ji[1], and Lixi Wang[1]

[1] Dept. of Computer Science, Nanjing University of Aeronautics and Astronautics, Postal Code 210016
Nanjing, China
zhuangyi@263.net

Abstract. Security has been an essential problem in the development of Web Services, and traditional technology cannot satisfy the needs in this area any more. Through an analysis on the existing schemes, the Web service security implementation frame SP-WSS based on SOAP security proxy is put forward, which has such advantages as being transparent, flexible and easily implemented. Finally, United ID authentication will be also discussed in the paper.
Key Words. Web Services, SOAP, Security, Proxy

1 Preface

Web Services is a technical platform for building up the distributed application that is mutually operable. With Web Services, various kinds of applications can be enclosed and allocated to Internet, and can realize cross-platform seamless mutual-operation through dynamic findings and integrated mechanism under network environment. Its applications in airfreight network etc. show a remarkable advantage in particular. Web Services is a study hotspot for online service and integrated application fields in the Internet today.

Security is one of the hottest concerns in the Internet application field, and a key problem needed to be solved for Web Services' becoming into mainstream Internet application technology. The Web-based Internet application has to meet data security requirements such as confidentiality, completeness and authorization validation etc. The Security of Web Services is an important tache for the whole Web Services technical platform.

However, the existing security technologies are not always applicable in Web Services, for example, package filtration technology and transmission layer security technology. Firstly, since Web Services are realized mainly by dint of the SOAP message[1] on HTTP(S), the firewall will lose its role as long as SOAP/HTTP messages are allowed to pass through the security boundary. Secondly, since SOAP message has to go through multiple intermediate stations before arriving at the destination, the existing transmission layer security technology (for example SSL) normally provides only point-to-point security protection rather than the end-to-end security

protection[2]. Traditional security technology cannot satisfy Web Services any more and seeking for a new security solution suitable for Web Services has become an urgent issue.

The paper is organized as follows. In Section 2, the security criteria of Web services will be introduced. In Section 3, we will discuss the implementation scheme for SOAP message security. And in Section 4, a SOAP proxy-based Web Services security model will be proposed and its efficiency will be analyzed then. In section 5, a solution for SAML-based United ID authentication will be described in details. Finally, Section 6 is a conclusion.

2 Security criteria of Web Services

At present, Web Services is realized by using HTTP as transmission layer, but HTTP and SOAP message layers don't provide any security mechanism. Though SSL/TLS can ensure the confidentiality and completeness of data during the transmission, the security mechanism provided by SSL/TLS aims only at two parties in direct communication, and cannot meet the requirement of security by SOAP message model because it is entirely possible that there are several intermediate coordination stations between the message sender and the receiver. In this case, neither the message sender nor the receiver can ensure the security of message during the transmission. For instance, the intermediate station may repeatedly send the same message for many times, which, in spite of whether it is goodwill or malice, will bring about a threat on the application security. There is the same problem with the Web Services that relies on other transmission layer protocols.

Since the transmission layer under SOAP layer cannot provide an end-to-end security protection for messages, an end-to-end security should be implemented in application layer or between application layer and SOAP. The existing solution is to introduce the security mechanism into SOAP message layer. In order to ensure the mutual operationality of SOAP message security, IBM and Microsoft (two Web Services initiators) and Verisign Co. published WS-Security Criterion[3] in April 2002. As a message-based SOAP security model, this criterion ensures the security of a message even though it passes through security boundary. Through a security expansion on the head of the SOAP message, this criterion enables each message to contain all necessary security information. But the criterion itself doesn't propose any new encryption algorithm or security model. Researchers have to combine Web Services protocol and application layer protocol with various encryption technologies and security models to achieve the completeness, confidentiality and validation of messages under Web Services.

WS-Security supports three kinds of security functions: security token transmission, message completeness and message confidentiality:

(1) WS-Security provides a universal mechanism for associating security token with messages. WS-Security doesn't require any specific security token. It is designed to be expandable, including X509 certificate and Kerberos bill etc.

(2) WS-Security uses XML signature and security token to provide the message completeness, support multiple signatures and support multiple signature formats in an expandable way. The signature can cite a security token.

(3) WS-Security uses XML encryption and security token to encrypt the content of SOAP message and ensure its confidentiality. Encryption mechanism supports the operation of multiple encryption technologies and processes together with multiple participants. Encryption can also cite a security token.

In SOAP, it introduces XML signature and XML encryption standard. A series of SOAP message heads are also defined to contain safety messages such as digital signature, encryption information and security token etc. Figure 1 illuminates how WS-Security criterion works combining with SOAP, HTTP and SSL and its relationship with XML signature and XML encryption protocol. Bottom layer is the HTTP connection between message sender and receiver; above the bottom layer is SOAP message layer; top layer provides end-to-end security connection by expanding SOAP message head and in combination with the relevant protocols of XML signature and XML encryption etc.

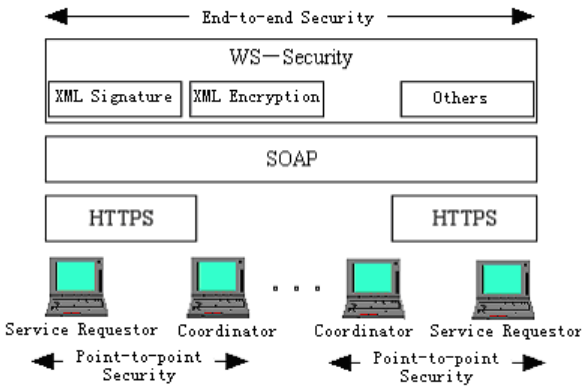

Fig. 1. End to End Security

3 Analysis on the implementation scheme of Web Services

Web Services security criterion itself is not involved in practical implementation. At present, the common way is to provide the relevant class libraries to the researchers, who will program to implement security control by using these libraries. For example, Microsoft's WSE is an expansion to Visual Studio .NET programming environment. However, such kind of scheme will bring more difficulties to program research and require Web Services researchers to have a deep understanding on the relevant protocols and criteria.

The way of calling class libraries could be combined with the security service provided by the service terminal's running environment and service terminal administrator can set the relevant security strategies to enhance service security. In order to access the service, the client terminal must provide security credit information according to the requirement of the service terminal. The difficulty for practical implementation lies in how to use the related tool set at client terminal to create SAML statement containing authentication information and to insert the statement in SOAP message head.

Another scheme that can be easily implemented is Web Services security implementation scheme based on SOAP security proxy [4]. SOAP security proxy can be transparently embedded in the existing Web Services, without changing the existing application. Working on the application layer, SOAP security proxy can check security on the incoming and outgoing SOAP messages, for example, checking the validity of XML mode and filtering the message contents, which is impossible to effect on transmission layer. In addition, SOAP proxy can also provide the security function of application layer such as role-based access control and audit etc. By using the published standards such as SAML, mutual operation with other types of security products can be realized as well.

SOAP security proxy cannot replace the existing package filtration mechanism but a kind of supplement for it. The messages form outside must go through the package filter to get to the SOAP security proxy, where they are then analyzed and identified for security according to the security strategy on application layer level.

Compared with the scheme based on class libraries, the disadvantages of SOAP security proxy scheme is the increased amount of security management and the extra demand for a security administrator to set the relevant strategies. To avoid these, security strategies can be centralized into a SOAP security proxy, which, similar to the gateway, simplifies the practical implementation of security management and has a remarkable advantage especially when the service platform is not heterogeneous.

As different service providers may require different security information, client terminals are required to maintain their individual credentials in the traditional security implementation scheme. While in the SOAP security proxy based frame, a SOAP security proxy can also be equipped, which, after the corresponding credentials of each outgoing message are found through mapping, will be inserted to the message head to add security information such as SAML statement and digital signatures for all outgoing messages within credited field.

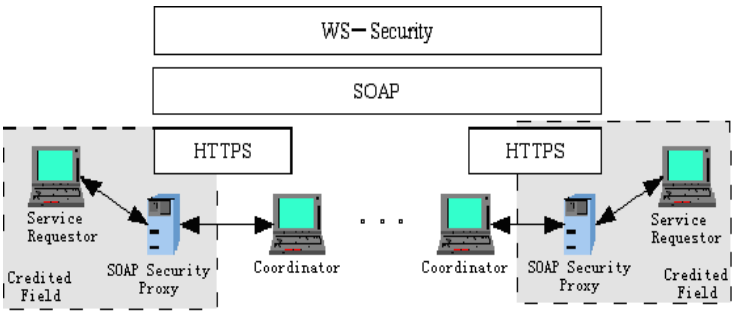

Fig. 2. Scene of SOAP Security Proxy

During the application of this scheme in airfreight network and B2B E-commerce, both sides on transactions can adopt multiple information platforms based on different technologies. To make sure a safe exchange in between, all client terminal software must be modified, which is no doubt a tremendous cost. If a centralized SOAP security proxy is adopted, it can attest and map the credentials for client terminal, and add the information into all outgoing messages. And there is no need to modify client terminal software and distribute the credentials etc, as a result it will success in not only saving money but also provide convenient for implementation.

4 Material SOAP proxy-based Web Services security implementation model SP-WSS

According to the analysis above, a Web Services security application model SP-WSS based on SOAP can be set up. This model is mainly composed of seven modules (ID authentication, credentials mapping, insertion security statement, mode analysis, authorization, exception processing and control console), security strategies library and audit log library, as shown in Figure 3. Each individual function is described as below:

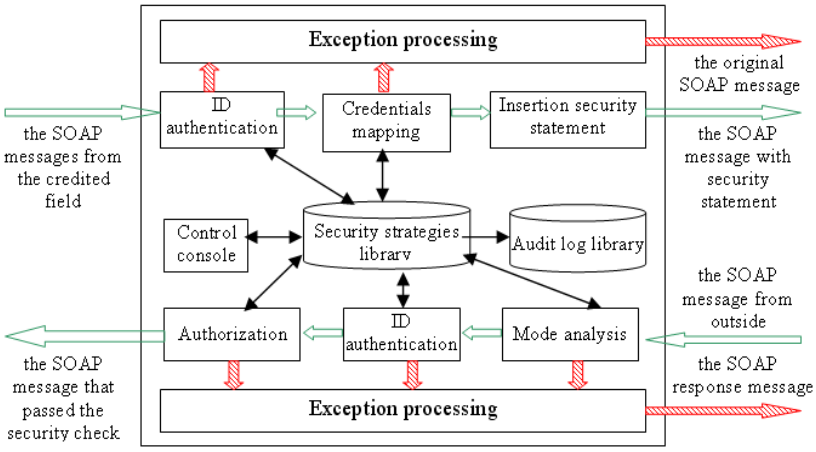

Fig. 3. Web Services Security Implement Model: SP-WSS

(1) ID authentication: to make ID authentication on the SOAP messages from client terminal within the credited field and from outside so as to confirm the identity of message sender. It can support the following authentication mechanisms:

- Anonymous access,
- ID authentication based on IP address,
- Basic ID authentication (ID authentication based on user name and password),
- XML-based security token (SAML statement).

(2) Credentials mapping: to make credentials mapping for SOAP messages from various client terminals so as to insert the corresponding credentials in message head;
(3) Insertion security statement: to insert credentials into outgoing SOAP message head;
(4) Mode analysis: to check the XML structure of incoming SOAP messages so as to ensure the message has the expected contents; to validate the XML digital signatures contained in message head so as to ensure the completeness of every message.
(5) Authorization: to send a security strategy request to the security strategies library according to the received URL or method name upon SOAP request, so as to decide on the resources that SOAP request may access to.
(6) Exception processing: in case outgoing messages fail in passing ID authentication or in credentials mapping, the original SOAP message will be forwarded outside; in case incoming messages fail in passing security check, they will be forbidden to pass through and a failure response will be sent to the message sender accordingly;
(7) Control console: to function as an interface for security management. By the control console, system administrator can equip the security frame and strategies for the whole system in a convenient and flexible way;

(8) Security strategies library: to store the parts of security strategies, to make responses to security strategy requests and to keep records for security events as needed for security audit;
(9) Audit log library: to store the parts of system security audit logs.

5 United ID authentication

Since in virtual enterprise and large corporations, multi-application systems are always based on distinct platforms, and have their own module of identity verification. When users switch different systems, they have to login frequently, or in order to reduce the login times, usernames and passwords should be set in the program codes. Given the inconvenience and the reduced efficiency it caused, it is not an ideal solution. However, united identity verification (ID authentication) can solve this problem.

SAML, a Standard based on the XML security, is published by OASIS. It is used to exchange security information like identity verification and authorization proof in the different internet credited field. SAML inherits some XML features- the data description feature crossing platforms. One of the main purposes for design is the Single Sign-on，SSO, which is used to describe the users' Assertion transmitting among different sites and so that we can overcome the limits of platforms, languages and computer architecture. The SAML Standard describes the main four parts: Assertion, Protocol, Binding and Profile. Figure 4 illustrates the location of SAML protocol in SOAP messages.

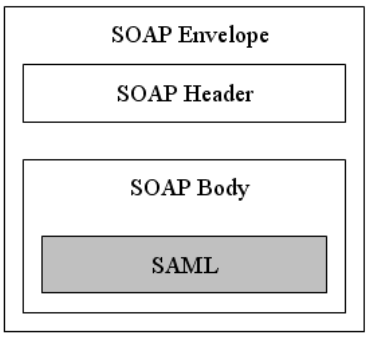

Fig. 4. SAML in the SOAP message

The merits of SAML include the following:

- Enable different types of security service systems to interact with each other,
- Provide the SSO identity verification function. It can significantly cut down the demands for copy security between sits

– Do not depend on any system it interacts with. Every system can set up its own Security strategies for the verification of user identity and authorization,
– Provide the attributes-based verification of identity function. It has the superiority over the identity verification based on XML digital signature.

As the features described above, we propose a method of SAML-based united identity verification, that is WS-UID, on the basis of analyzing Pull and Push pattern.

Pull pattern In this pattern, the SAML assertions are saved at the source sites. When the destination sites need, it will be "pulled" back from the source. Thus, the request sent by user to the destination site contains only one citation to SAML, which is denoted by Artifact, a string with certain length to mark the SAML assertion.

Push pattern Contrast to the Pull pattern, in Push pattern, the user automatically submits the SAML assertion to the destination site. During the course, the user "pushes" the assertion to the destination by HTML table <FORM>. In Push pattern, the Web destination site generates an authorization token, while the source will use it to redirection to the destination sites.

Third-Party Pattern The third pattern is for the third party, the security service, to execute the verification operation to the entity, while the entity access a series of Web destination sites and adopt the Pull pattern at the first sites and the Push at the second. Such kind of pattern superiors to the former two patterns above only in the case that the organization depends on the external security services, for example the security services provided by Microsoft Passport.
If the organization uses Liberty Alliance and then creates the center service, rather than certain member of the organization provides it, such pattern will be probably become a feasible solution, because the Web source and destination sites may not be able to take charge of the extra work which result from the management of those tokens. In this case, it is more reasonable to start the center special service.

WS-UID United ID authentication In Pull pattern, the Web source site verifies identity to the entity and sends a token for destination use. In Push pattern, the source site requests for a token from the destination site, and then the destination authorize one which will be used by entity when access. The third-party pattern have to create the centralized security services, so it can easily cause efficiency bottleneck and single-sign efficiency-loss problems, for instance, the single-sign on mode of Web service put forward in [6] belongs to the third-party pattern. In addition, Microsoft .Net Passport use similar pattern as well. However, in Pull pattern, the SAML assertions for identity verification do not transmit with the entity together and it has higher security compared to Push. Therefore, in WS-UID, we make use of the idea of Pull pattern to create the distributed system of united identity verification.

The precondition for implementing united identity verification is that the distinct credited fields will trust in each other. If it is the different credited fields within the same corporation, such kind of trust can be put to effect by means of setting the configuration of system management; if within different corporations, the relationship of trust probably should be constrained by related protocols.

Every credited field has a unique identity among the trust alliance. (Tagged by the X.509 certificate.) Every field maintains a tag set of local user identity and both the local user identity and the identity of credited field make up of the global unique user identity.

Among the alliance, the status of the united identity verification system of every credited field is equivalent, there is no centralized united verification service, that is not completely depend on a centralized server for verifying identity that all corporations among alliance trust in. In this way, any malfunction occurred in any verification service from credited field of the alliance will not have influence on other field verification. After every application system that supports the united identity verification is added into the credited field, it should login in UDDI center.

When a user successfully signs on in the credited field SD1, if he needs to access another credited field SD2 of the untied trust alliance, he requests for a assertion to SAML authorization organization, that is the SD1 identity assurance service. It will generate an identity assurance note for the user so that he can access SD2 then. SD2 checks the note after receiving the user request, and then send a request to SD1 for verifying the validation of the note. Once SD1 confirm the response, the global user identity will be mapped to a local one, and then the signing on succeed.

The verification service has two major functions: verify the sign request of both local users and the global users. For the local case, it is similar to the user sign-on in single credited field that the local verification strategies are used during the sign-on process, the simplest case is the verification of the username and password, or more complex case, the one of the certificate, which lies in the security strategies in local credited field. For the global case, it is more complicated. As Figure 5 shows.

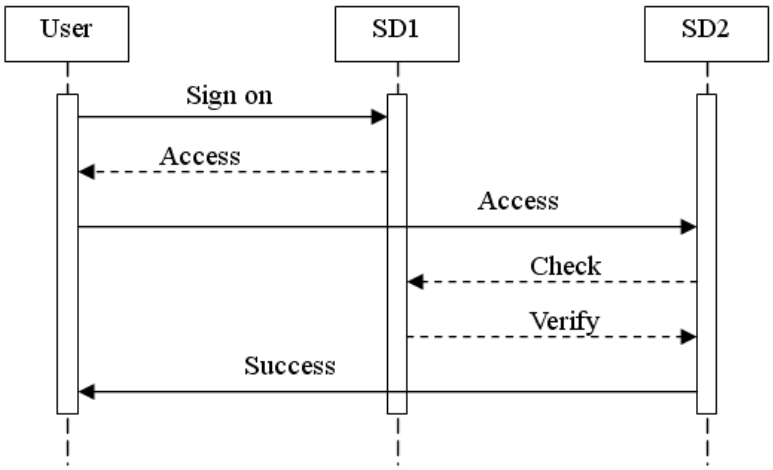

Fig. 5. The process of united ID authentication

6 Conclusions

Compared with other security implementation schemes of Web Services, SP-WSS has the following advantages:

Transparency: it can be transparently allocated between Web Services requester and provider. Its existence is imperceptible to both service requester and provider.

Flexibility: it enables Web Services researchers to step out of security obsession and to concentrate on the design of operation flow of Web Services.

It can be easily implemented: it is unnecessary to make the second security study individually on the existing Web service instead, it is only needed to make a simple security allocation on SOAP security proxy to obtain the same effect. Therefore it can not only protect the existing investment but also reduce the risk in implementation.

Of course, the solution based on SOAP security proxy has its deficiencies. For example, it needs additional security management configuration; when there is large amount of communications, SOAP proxy will become a bottleneck for the performance; and the existence of SOAP security proxy will exert an influence on the speed of service response. These issues are expected to be improved in later researches.

References

[1] W3C Recommendation, "SOAP Version 1.2 Part 0: Primer," http://www.w3.org/TR/soap12-part0/, 24 Jun. 2003.

[2] P Kearney, J Chapman, N Edwards, M Gifford and L He, "An overview of Web Services security," BT Technology Journal, Vol. 22 No. 1, Jan. 2004.

[3] IBM, Microsoft, VeriSign, "Web Services Security Language"[OL], http://msdn.microsoft.com/ws/2002/04/Security/, Apr. 2002.

[4] Brose, Gerald, "Securing Web Services with SOAP Security Proxies,"[A] Proceedings of the International Conference on Web Services, p 231-234, 2003.

[5] IBM, Microsoft, " Security in a Web Services World: A Proposed Architecture and Roadmap " [OL], http://www-900.ibm.com/developerWorks/cn/webservices/ws-secmap/index.shtml , V1.0, Apr. 2002.

[6] Handong Mao, Weiming Zhang, "A Single Sign-On System Based on Web Service," Computer Engineering and Applications, Vol. 40 No. 24, Jan. 2004.