A Method for Accessing Trusted Services Based on Service-Oriented Architecture

Xin Yu
College of Automation
Beijing Union University
Beijing, China
yx_paper@126.com

Yu Bai
English Department
Beijing Union University
Beijing, China
Zdh_buu@126.com

Abstract—Service-oriented architecture provides flexible methods for systems development and integration. It is widely adopted in web services. On the basis of studying service-oriented architecture and web service security standards, this paper proposes a method for accessing trusted services based on service-oriented architecture and gives out a system implementation model.

Keywords—service; SOA; security; XML; SAML

I. INTRODUCTION

Service-oriented architecture (SOA) provides an effective way for constructing loose coupled web services composition system. It is divided into separate business functions, the so-called services, which users can combine and reuse them in the production of business applications [1]. As applications are opened up, how we can combine these services securely becomes an issue. However, traditional security mechanism makes chiefly use of HTTPS and SSL to secure the data transmission. But it is not sufficient to solve all the application scenarios of security. Some emerging technologies and standards address different aspects of the problem of security in SOA, such as WS-Security, WS-Trust and WS-Security Policy address the security problem for SOA implementations that use web services. This paper gives out a method for accessing a trusted service by service SAML assertion in order to accomplish services calling each other securely.

II. RELATED KNOWLEDGE

A. SOA Standards and Logical Architecture

SOAP, WSDL and UDDI are core standards of SOA. As in Fig.1, service provider describes its services with WSDL and publishes the description to service registry which conforms to UDDI standard. The address of service provider and interface information of service return to service requester after service requester has sent a message of looking for service to service registry, the process of communicating using SOAP messages.

Figure 1. SOA standards.

We focus on analysis of UDDI now. UDDI (Universal Description, Discovery and Integration) describes a registry of web services and service interfaces for publishing, retrieving, and managing information. It is a platform-independent, XML-based registry to publish and discover service dynamically. But traditional UDDI does not provide a fine-grained security, which restricts access to certain service entity information based on their role [2]. This paper gives out an improved service registry instead of traditional UDDI registry.

In order to describe the process of accessing trusted services based on SOA, we firstly illustrate a SOA logical architecture [3], as shown in Fig.2.

Figure 2. A SOA logical architecture.

The whole business process based on web services conforms to WS-BPEL that is an executable language for specifying interactions with web services. The ESB provides loosely coupled interconnectivity between the service
requester and the service provider, such as transforming message formats, routing requests and converting transport protocol.

B. SOA Security

Due to the loose coupling of services and their operation across organizational boundaries, security issue becomes even more critical for implementations structure according to SOA principles. It touches on a wide range of technologies. Chief security technology and standards involved in this paper are listed as follows:

1) Network and transport layer security: These layers focus on secure communication at the network and transport layers. It has serious weaknesses:
   a) The inflexibility of security: for example, it can not encrypt for parts of a message.
   b) Not end-to-end security: Each intermediary may have to handle information, so SOA require end-to-end security. However, usual transport layer technology only provides point-to-point security.
   c) Only transmitting security: It only guarantees to security of information transmitting. Once messages have arrived at the end point, they are no longer safe.

2) XML security: WS-Security is a communications protocol providing a means for applying security to web services. It specifies the XML signature and XML encryption and defines how digital signatures, message digests, and encrypted data are wrapped in a SOAP envelope.
   a) XML signature: XML signature specifies the processing rules and syntax to wrap message integrity, message authentication, and authentication items inside XML syntax. Typical structure of XML signature is shown as Fig.3[4].

   b) XML encryption: XML encryption specifies the data confidentiality using encryption techniques. It provides to encrypt an entire XML syntax or items inside XML syntax.

   Typical Structure of XML encryption and its process are shown as Fig.4[5].

   ![XML encryption structure and encryption process.](image)

   3) SOAP security: SOAP is a simple, flexible, and extendable mechanism for exchanging structured data, and expressed in XML. SOAP envelope enables us to add a variety of information to the message, such as some security information. The SOAP message is broken up into two portions: the SOAP header and the SOAP body. The SOAP header provides a flexible mechanism for extending a SOAP message. Although the SOAP header is the best place to add security features to messages, the SOAP specification itself does not specify such header elements [6]. The SOAP body is used to hold the basic data contents that go along with the message.

   4) SAML assertion: Security assertion markup language (SAML) is an XML-based standard for exchanging authentication and authorization data between different security domains. A SAML assertion consists of the following elements:

   a) Conditions: It describes the assertion expiry date through attribute NotBefore and NotOnOrAfter.
   b) AuthenticationStatement: It consists of the attribute AuthenticationMethod, the attribute AuthenticationInstan and the element Subject. This part is important to our design, so we list its components and functions as follows:

   ![Security assertion markup language (SAML) structure.](image)
TABLE I. COMPONENTS AND FUNCTIONS OF AUTHENTICATIONSTATEMENT

<table>
<thead>
<tr>
<th>Elements</th>
<th>functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuthenticationMethod</td>
<td>identifiers authentication methods</td>
</tr>
<tr>
<td>AuthenticationInstant</td>
<td>specifies the instant of time when the authentication took place</td>
</tr>
<tr>
<td>Subject</td>
<td></td>
</tr>
<tr>
<td>NameIdentifier</td>
<td>specifies the name of the subject</td>
</tr>
<tr>
<td>SubjectConfirmation</td>
<td>specifies the relationship between the subject of an assertion and the author of the message that contains the assertion</td>
</tr>
<tr>
<td>c) Signature: This element is the signature of the assertion authority. The signature of the authority allows the recipient of the assertion to verify that the assertion is true. In this paper, some peculiar security information is embedded in the SOAP envelope in order to achieve information encryption, signature, authentication, and generation the share key.</td>
<td></td>
</tr>
</tbody>
</table>

III. A METHOD FOR ACCESSING TRUSTED SERVICES BASED ON SERVICE-ORIENTED ARCHITECTURE

A. Design Objectives

The design objectives of system are to solve security problems as follows:
- Allow users to complete authentication in the local security domain, but it cannot gain its SAML assertion. The SAML assertion is shown by a management center in the verification process so as to avoid user’s illegal use of its SAML assertion. For example, users has left local domain.
- Achieve single sign-on so that users cannot reenter the security certificate.
- Provide the methods that can dynamically generate share key and generation process can defeat replay attack so that the vast amount of data can be quickly encrypted and transmitted over http between entities.
- Data exchange via XML within a SOAP envelope is safe.

B. Solutions

Aiming at the design objectives mentioned above, this paper gives out the scheme, as following Fig.5.

The chief processes of system solutions are listed as follows:

Step 1: SP sends registration information to SRAM. The information is wrapped in encrypted SOAP envelope and its structure is shown in Fig. 6.

![Figure 6](image-url) A structure of request registration.

The big prime number is used in the generation of share key though Diffie-Hellman key agreement protocol when authentication has been verified between service provider and service consume, but it is not essential.

Step 2: SRAM generates a SAML assertion on the basis of information of request registration from service provider. A basic structure of the SAML assertion is shown in Fig. 7.

![Figure 7](image-url) A basic structure of the SAML assertion.

There are two additional subelements embedded in the element SubjectConfirmation: BigPrime and Sstamp, where BigPrime is a parameter for Diffie-Hellman algorithm generating from service registrant. The Sstamp is a hash value of the request time and return to service registrant. It is used in defeating replay attack in the course of generation share key by Diffie-Hellman algorithm. The meanings of other elements refer to table 1.

Step 3: SRAM sends the SAML assertion to SIS.
Step 4: SA sends service request to SIS.
Step 5: SA receives service SAML assertion from SRAM if authentication has been verified.
Step 6: SA verifies the service’s authenticity through element signature within SAML assertion.

In the steps mentioned above, information transmission and exchange is via the encrypted SOAP message and the precondition is that service applicant must trust the SRAM belonging to security domain as well as service provider’s.
C. Performance Analysis

Due to adopting XML encryption, it ensures message end-to-end confidentiality; SOAP signature ensures message integrity and using SAML achieves authentication between different entities. Furthermore, utilizing SAML assertion achieves single sign-on [7] and can generate a share key by Diffie-Hellman algorithm as occasion requires. The system security performance depends on XML encryption algorithm.

IV. CONCLUSIONS

This paper gives out a method for accessing trusted services based on service-oriented architecture. It can achieve information confidentiality, integrity, authentication, Single Sign-on and generation of dynamic share key based on Diffie-Hellman key agreement protocol. What we should do next is to design a proxy firewall based on XML context to prevent such attacks as denial of service and buffer overflow.

REFERENCES