

一种基于 Axis2 的 SOAP 安全传输模型的研究

李双江,郝克刚,葛 玮

(西北大学 计算机科学系 软件工程研究所,陕西 西安 710127)

摘 要:目前 Web 服务体系架构存在标准限制且需要支持多种类型的客户端的问题,因而使得 Web 服务的安全极具挑战。SOAP 构成了 Web 服务体系结构中的通信基础,WS-Security 安全规范作为 SOAP 的扩展协议,是 Web 服务环境下最基础的安全协议。利用 Apache Axis2 结合 Rampart 组件实现了一个基于 WS-Security 的 SOAP 消息安全传输模型,从而为 Web 服务提供了一种消息级的安全解决方案。

关键词:Web 服务; SOAP; WS-Security; XML 签名; XML 加密; Axis2

中图分类号:TP393.08

文献标识码:A

文章编号:1673-629X(2008)11-0142-04

Research on a SOAP Security Model Based on Axis2

LI Shuang-jiang, HAO Ke-gang, GE Wei

(Institute of Software Eng., Department of Computer Science, Northwest University, Xi'an 710127, China)

Abstract: The current Web services architecture has standard limits and must supply to the multi-type client ends. How to protect the Web security services become very difficult. SOAP has been the underlying communication basic of Web services, WS-Security security specifications as extension of SOAP protocol, is the most foundation security agreement of Web services. A SOAP security transmission model was presented based on WS-Security security specifications in this article, implementing Apache Axis2 and Rampart, which provided a kind of message level security solution for Web services.

Key words: Web services; SOAP; WS-security; XML signature; XML encryption; Axis2

0 引 言

随着互联网的发展,面向服务的体系架构(SOA)几乎已经成为企业应用架构的主流,Web 服务是 W3C 开发的一种面向服务的体系架构,它代表了现在企业集成的重要解决方案。Web 服务通过网络接受 SOAP 消息请求,并把处理结果封装成 SOAP 消息发送给客户端。SOAP 消息是用扩展标记语言(XML)表示的,可用于任何系统和平台,但 SOAP 消息规范本身并没有直接提供任何机制去解决 Web 服务的访问控制和安全性问题。随着 Web 服务在电子商务等领域应用的不断扩展,SOAP 安全性也越加重要。

1 SOAP 安全性分析

Web 服务主要是通过网络来实现的,因此它面临

着网络所面临的威胁,如:非授权访问、信息遗漏丢失、破坏数据完整性等。目前 Web 服务应用中主要通过 SSL(Secure Socket Layer)和 TLS(Transport Layer Security)来提供传输级别的安全,SSL/TLS 提供的安全特性包括认证、数据完整性、数据机密性。它只限于保证点对点(Point-to-Point)的传输安全。IPSec 是另一个 Web 服务中保证传输层安全的非常重要的网络层标准,IPSec 同 SSL/TLS 一样,也提供了身份认证,保证数据完整性和数据机密性的安全会话^[1],然而仅有这些传输层和网络层的安全机制是远远不够的。

Web 服务的工作过程是将 SOAP 消息发送服务端点,请求特定的 Web 服务,接受 SOAP 响应消息(包括错误提示)。因此模型经常包含有一个具有多个中间跳跃结点的拓扑结构。在这种情况下,中间设备或应用程序是不值得完全信任的,它们随时可以修改传输的信息。因此,SSL/TSL 并不能保证端到端传输的安全性^[2]。

同时,SOAP 消息在传输的时候可以与不同的传输层协议进行捆绑,如 HTTP,SMTP 协议,Web Services 安全通信机制应该能够独立于具体的传输协

收稿日期:2008-02-23

基金项目:国家“863”计划资助项目(2004AA115090)

作者简介:李双江(1982-),男,山东莱芜人,硕士研究生,主要研究方向为中间件、 workflow 技术;郝克刚,教授,博士生导师,主要研究方向为软件工程、分布式计算、 workflow 技术;葛 玮,副教授,硕士生导师,主要研究方向为软件工程、 workflow 技术、中间件。

议^[2]。因此,需要在 SOAP 消息中加入扩展信息,保证传输层的独立性,从而使消息内容与传输层无关。

此外,传统的网络级别防火墙只能保护一个私有网络的安全,缺乏端到端的保护,缺乏不可抵赖性、选择性保护(只加密一部分消息)、灵活的认证机制及消息层的保护。所以要实现 SOAP 消息级别的安全。

2 基于 WS-Security 的一个安全模型

WS-Security^[3]是 OASIS 提出的一个 Web 服务安全规范,它建议了一个 SOAP 扩展集用于建立安全的 Web 服务,实现消息内容的完整性和机密性。其主要目的之一是实现端到端的安全,保证 SOAP 消息通过不安全的中间节点,安全可靠地从请求者到达提供者。WS-Security 规范为 SOAP 消息的交互过程中的三个主要问题提供了解决方案:消息完整性、消息保密性和消息认证。

* 消息完整性:WS-Security 使用 XML 签名对 SOAP 消息进行数字签名,保证 SOAP 消息经过中间结点时不被篡改。

* 消息保密性:WS-Security 使用 XML 加密对 SOAP 消息进行加密,保证 SOAP 消息即使被监听,监听者也无法提取出有效信息。

* 消息认证:WS-Security 引入安全令牌的概念,用其代表消息发送方的身份。通过与多种数字签名结合,消息接收者可以确认 SOAP 消息发送者的合法。

WS-Security 是基于 SOAP 的安全规范,它本身并没有提供完整的安全性解决方案,主要是通过利用现有标准和规范来实现安全性。业界已经解决了许多此类问题,例如 Kerberos 和 X.509 用于身份验证;X.509 还使用现有的 PKI 进行密钥管理;XML 加密和 XML 签名描述了 XML 消息内容的加密和签名方法。WS-Security 在现有规范中添加了一个框架,用于将这些机制嵌入到 SOAP 消息中。除了利用其他现有消息认证、完整性和加密外的标准和规范外,WS-Security 还指定了一个通过用户名令牌(UsernameToken)元素传输简单用户凭据的机制,为了发送用于加密或签名消息的 X.509 证书等,还定义了一个二进制安全令牌^[4]。

WS-Security 将所有安全信息保存在消息的 SOAP 部分中,SOAP 消息到达目的结点后由消息接收者直接验证这些安全信息的真伪并解密相关数据,这是以一种与传输无关的方式完成的。从而为 Web 服务的安全性提供了消息级的、端到端的解决方案,形成了一种基于 WS-Security 规范的 SOAP 消息安全传输模型,如图 1 所示。

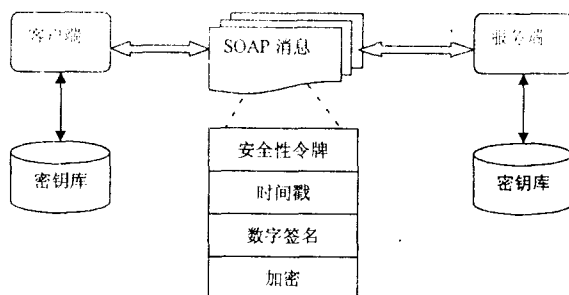


图 1 WS-Security 规范的 SOAP 安全传输模型

(注释:为了实现上图所示 SOAP 安全传输模型,假定服务提供者和客户端已经某种密钥管理系统建立了比较完备的密钥库(KeyStorage)。对于密钥管理、证书认证等内容,文中不再涉及,只关注于 SOAP 的安全传输。)

3 安全模型的实现

3.1 平台说明

Apache Axis2^[5]是 Apache Axis SOAP 项目的后继项目。此项目是 Web 服务核心引擎的重要改进,目标是成为 Web 服务和面向服务的体系结构(Service-Oriented Architecture, SOA)的下一代平台。Axis2 引入了 Web 服务扩展模块的概念;其中模块的主要工作是对核心功能进行扩展。在 Axis 1.x 中,可以通过向处理程序链添加处理程序来实现此目标。与 Axis 1.x 处理程序链相比,使用模块的优势在于,您可以在根本不改变全局配置文件的情况下添加新模块。同时,模块是一个自容器,其中可以包含处理程序、第三方库、模块相关资源和模块配置文件。

Axis2 引擎提供了处理 SOAP 消息的框架,特别是提供了良好的扩展机制,但缺少重要的安全机制和访问控制。针对上述问题,可以利用 Axis2 提供的可扩展处理器机制,结合 SOAP 消息安全框架中的 XML 签名和加密技术,实现上述 SOAP 消息级安全模型。

Rampart 模块^[1,6]是一个基于 Apache WSS4J 的 Axis2 模块。Apache WSS4J 是 WS-Security 标准的一个开源实现。WSS4J 是基于 Java 的库,能够使用 WS-Security 信息对 SOAP 消息进行签名和验证。它提供了用户名令牌验证和传递消息时保证信息的完整性和真实性等一些 Web 服务安全保障。

3.2 SOAP 消息安全传输实现

由于 Rampart 模块对 WSS4J 进行了封装,在 Axis2 中利用 Rampart 模块可以实现 WS-Security 规范为 SOAP 消息传输提供的三种安全保证:完整性、保密性和认证。而要实现上面的要求,需要在服务提供端的 service.xml 和客户端的 axis2.xml 中对输入流安全

(InflowSecurity)和输出流安全(OutflowSecurity)进行相应的参数设置。Rampart 模块定义了两个标签<InflowSecurity>和<OutflowSecurity>:<InflowSecurity>用来设置输入流安全处理程序;<OutflowSecurity>用来设置输出流安全处理程序。每个安全处理程序可以被多次调用,于是引入了<action>标签进行描述;<InflowSecurity>(或<OutflowSecurity>)可以包含多个<action>元素(如表 1 所示)。

表 1 <action>标签的主要元素及描述

参数	描述
items	输入/输出流的安全动作(例:时间戳、签名 SOAP 消息、加密 SOAP 消息等)
user	签名用户名
encryptionUser	加密用户名
signaturePropFile	获取签名/签名认证参数的参数文件
decryptionPropFile	获取加密/解密参数的参数文件

1) 安全性令牌。

对 SOAP 消息进行身份认证的方式很多,不过都是通过 SOAP 消息头中添加一些安全令牌(Security Token)信息来完成的。用于身份认证的最常见方法之一是使用用户名和密码。Rampart 模块提供了 WS-Security 规范中定义的用户名令牌环(UsernameToken)元素传输简单用户凭据。要应用用户名令牌环:首先,在客户端的配置文件的输出流安全中把<items>元素设为“UsernameToken”,<user>元素设为“用户名”;其次,要编写一个密码回调函数(PasswordCallback)。Rampart 模块为了安全不支持把密码写入配置文件,而是通过密码回调函数直接添加到 SOAP 消息中。服务提供端收到消息后,对用户名和密码进行身份认证;如果通过认证,则进行相应的消息处理;否则,抛出异常。

除了用户名令牌,Rampart 也支持二进制令牌(如 X.509 证书)等,具体设置与用户名令牌相似。通过二进制令牌如(X.509 证书)不仅可以实现身份认证,还可以对消息进行数字签名,保证消息的完整性。

2) SOAP 消息签名。

SOAP 消息签名是保证消息完整性和不可否认性的重要手段。Rampart 模块可以通过对<items>、<user>、<signaturePropFile>等多个元素的配置来实现 SOAP 消息签名。另外为了避免重复攻击,数字签名常跟时间戳或者 nonces 等一起使用,将签名的日期和时间加在消息上一起发送。

利用 Rampart 模块进行签名的大致过程为:首先,发送方在配置文件中设定签名用户(<user>)(密钥对的别名),利用回调函数(<passwordCallbackClass>)返回的密码(访问密钥库的密码)在密钥库(<signaturePropFile>)

中获取签名用户的私钥;然后利用签名用户的私钥签名消息的相关部分(<signatureParts>)。当接收端收到消息,它也经过相似流程从密钥库(<signaturePropFile>)中取得签名用户的公钥;然后利用公钥验证签名。

3) SOAP 消息加密。

SOAP 消息加密可以实现消息的保密性。Rampart 模块提供了<items>、<encryptionUser>、<optimizeParts>、<decryptionPropFile>等多个元素用于 SOAP 消息加密的配置,以实现对消息全部或部分的加密。

Rampart 模块支持对称加密和不对称加密。消息发送端通过回调函数(<passwordCallbackClass>)返回的密码(访问密钥库的密码)从密钥库(<signaturePropFile>)获取加密用户(<encryptionUser>)(密钥对的别名)的公钥或共享密钥(<EmbeddedKeyName>)对消息加密部分进行加密;接收方接收到消息后,经过相似的过程获取自己的私钥或共享密钥进行解密。

3.3 应用实例

为说明方案可行性,用一个简单服务的调用过程对其进行测试。在此用例中测试平台为 Tomcat5.5 + Axis2 1.3 + Rampart1.3,签名和加密需要使用到数字证书和密钥对,可以使用 JDK 提供的 KeyTool 工具创建密钥对和数字证书。分别为服务端和客户端创建 RSA 密钥对,并生成各自的 X.509 数字证书(包含公钥和数字签名)。服务端和客户端拥有各自的密钥库 JKS 文件,服务端的密钥库保存服务端的密钥对和客户端的数字证书,而客户端的密钥库保存客户端的密钥对和服务端的数字证书。

首先,在服务端和客户端建立密码回调函数 PWCBHandler.java。为了测试简单,回调函数仅仅是提供用户名密码,一般情况下密码回调函数会通过一个 LDAP 目录或其他的方法把密码和用户名相关联,用以确认用户验证的有效性,当用户名/口令无效时则抛出异常。

其次,在服务端和客户端分别建立属性文件 client.properties 和 service.properties。此参数文件提供关于签名/签名认证(或加密/解密)等操作的参数,其格式大体如下面示例所示:

```
org.apache.ws.security.crypto.merlin.keystore.type=jks(密钥库类型)
```

```
org.apache.ws.security.crypto.merlin.keystore.password=apache(密码)
```

```
org.apache.ws.security.crypto.merlin.file=client.jks(密钥
```

库)

最后,在服务端和客户端进行相应部署。

服务端的部署如下:

```
<parameter name="InflowSecurity">
<action>
<items>Timestamp Encrypt Signature</items>
<passwordCallbackClass> PWCBHandler </passwordCall-
backClass>
<signaturePropFile> service. properties</signaturePropFile>
</action>
</parameter>
<parameter name="OutflowSecurity">
<action>
<items>Timestamp Encrypt Signature</items>
<user>service</user>
<passwordCallbackClass>org. apache. rampart. PWCBHandler
</passwordCallbackClass>
<signaturePropFile> service. properties </signaturePropFile>
</action>
</parameter>
客户端部署如下:
<parameter name="OutflowSecurity">
<items>Timestamp Encrypt Signature</items>
<user>client</user>
<passwordCallbackClass> PWCBHandler </passwordCall-
backClass>
<signaturePropFile> client. properties</signaturePropFile>
<parameter name="InflowSecurity">
<items>Timestamp Encrypt Signature</items>
<passwordCallbackClass>org. apache. rampart. . PWCBHandler
</passwordCallbackClass>
<signaturePropFile> client. properties</signaturePropFile>
</parameter>
```

利用 tcpmon-1.0 捕获的关于客户端和服务提供端传输的 SOAP 消息(如图 2 所示),可看到,SOAP 请求消息已被客户端签名/加密(图 2 上部分);服务提供端经过对 SOAP 消息的解密/认证,调用了相关服务并返回了经签名/加密的 SOAP 消息(图 2 下部分)。

4 结束语

安全问题是 Web 服务中的薄弱环节,文中基于 WS-Security 规范实现了一种简单的 SOAP 消息安全传输模型。随着 Web 服务技术在电子商务中的应用和推广,SOAP 消息的安全性问题越来越引起人们的关注。SOAP 消息的安全性问题所涉及的方面很多,对其加密和签名所采用的方法也有多种选择,在实际

应用中需要结合具体的问题灵活地实现 SOAP 消息的安全性。

```
<ds:SignatureValue>MnyU20yUkZpSik0y1VuUn5Ak1B3
<ds:KeyInfo Id="KeyId-21202114">
<wsse:SecurityTokenReference xmlns:wsu="htt
<wsse:Reference URI="#CertId-148082" Val
</wsse:SecurityTokenReference>
</ds:KeyInfo>
</ds:Signature>
<xenc:EncryptedKey Id="EncKeyId-12115695">
<xenc:EncryptionMethod Algorithm="http://www.w
<ds:KeyInfo xmlns:ds="http://www.w3.org/2000/0
<wsse:SecurityTokenReference>
<wsse:KeyIdentifier EncodingType="http://
</wsse:SecurityTokenReference>
</ds:KeyInfo>
<xenc:CipherData>
<xenc:CipherValue>dRQvEXvKVKgeioY72Rf+ub3P
</xenc:CipherData>

<ds:Reference URI="#Id-3432913">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/20
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/20
<ds:DigestValue>lvSkuptMEifvyaYSrQVXy5GDE8=</d
</ds:Reference>
<ds:Reference URI="#SigConf-12966337">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/20
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/20
<ds:DigestValue>6uF71EBn96LypsAA/rsJg08WIL4=</d
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>M3PaED8xqYysZRGogUR3kVY8rxayFvnbBy
<ds:KeyInfo Id="KeyId-26867942">
```

图 2 tcpmon-1.0 捕获的关于客户端和服务提供端传输的 SOAP 消息

参考文献:

- [1] 金键,张鸿. Web 服务安全性分析[R]. 北京:中科院网络信息中心,2003.
- [2] 杨鲲鹏,李海峰. SOAP 消息安全性分析及其加密、签名的实现[J]. 计算机现代化, 2005(6):123-126.
- [3] 王凡,李勇,朗宝平. 基于 WS-Security 构筑安全 SOAP 消息调用[J]. 计算机应用,2004,24(4):121-123.
- [4] 景建笃. Apache Axis1.1 环境下 WS-Security 的研究与实现[J]. 计算机工程与设计,2005,26(7):1925-1927.
- [5] Apache SoftWare Foundation. Axis2[EB/OL]. 2007. <http://ws.apache.org/axis2/>.
- [6] Apache SoftWare Foundation. WSS4J[EB/OL]. 2007. <http://ws.apache.org/wss4j/>.