XFire 和 WSS4J 下的 Web Service 组合安全策略①

Combination Security Policy of Web Service in XFire and WSS4J

邓子云　(湖南现代物流职业技术学院 物流信息系 湖南 长沙 410131)
黄　婧　(中南大学 商学院 湖南 长沙 410083)
罗　涛　(湖南现代物流职业技术学院 物流信息系 湖南 长沙 410131)

摘　要： 结合运用 XFire 和 WSS4J，由 XFire 提供 Web Service 容器，由 WSS4J 提供安全机制，给出了“数字签名+报文加密+身份认证”的组合安全策略，并在现代物流实验中心运用 Tuscany 作为 SCA 中间件，运用这种组合安全策略作出了应用实验。

关键词： XFire WSS4J Web Service 组合安全策略

1　引言

在 SOA(Service Oriented Architecture，面向服务的架构)软件架构的进一步深入应用过程中，Web Service 技术也得到推广使用。然而 Web Service 具有分布式、无状态的特性，处理 Web Service 的安全过程比处理其它领域应用的安全性问题更为复杂[1]。

没有进行安全技术处理的 Web Service 暴露在 Internet 中，调用接口对外是开放的，任何一个可以连接到 Internet 的客户端，都可以来访问这个 Web Service，虽然达到了共享的目的，但安全性的问题也十分突出。

2　Web Service的安全机制

Web Service 的安全性涉及身份验证和授权，数据加密解密等多个方面，其根本目的就是解决完整性、不可抵赖性、保密性的问题[2]。

SSL 在 Web Service 的安全性上显得无能为力。SSL 是端对端的通信，脱离传输层就无法保证安全性，且消息必须全部加密、签名，而不能针对消息的某部分，没有考虑 XML 处理上的需求。SSL 对应着传输层，而 Web Service 采用的是基于 HTTP 协议之上的 SOAP 协议，这与传输层无关的[1]。

2.1 WSS4J 与 WS-Security

WS-Security 是一种提供在 Web 服务上应用安全的方法的网络传输协议。2004 年 4 月 19 日，OASIS 组织发布了 WS-Security 标准的 1.0 版本。 2006 年 2 月 17 日，发布了 1.1 版本。WS-Security 是最初 IBM、微软、VeriSign 和 Forum Systems 开发的，现在协议由 Oasis-Open 下的一个委员会开发，官方名称为 WSS。

WS-Security 安全规范包括安全凭证、XML 签名和加密、消息附件等问题，其中有 WS-Security 核心规范、用户权标规范、X.509 权标规范、Kerberos 权标规范、SAML(Security Assertion Markup Language，安全断言标记语言)权标规范、REL 权标规范、带附件的 SOAP(Simple Object Access Protocol，简单对象访问协议)规范和模式等子内容。针对不同领域的细分问题，OASIS 在 WS-Security 的基础上又继续制定了 WS-Secure Conversation、WS-Federation、WS-

① 基金项目:湖南省科技计划(2007GK3059)
收稿时间:2008-12-23

Authorization、WS-Policy、WS-Trust、WS-Privacy 等规范,形成了一个庞大的 Web Service 安全性协议家族,如图 1 所示。

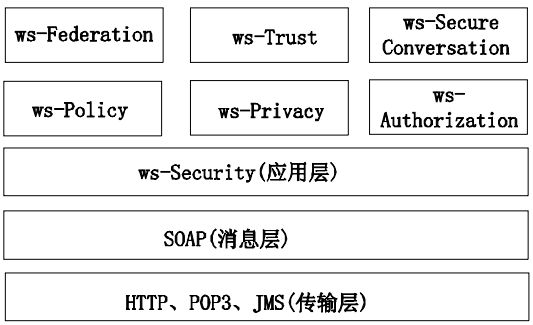

图 1　Web Service 安全性协议家族

WSS4J 是对 WS-Security 的开源实现，它通过对 SOAP 中 WS-Security 相关的信息对 SOAP 报文进行验证和签名，这样就可以通过 WSS4J 来实现 Web Service 应用中的安全性、机密性和身份验证等需求。

2.2 XFire 与 WSS4J

XFire 是 codeHaus 组织提供的一个开源框架，它构建了 POJO 和 SOA 之间的桥梁，主要特性就是支持将 POJO 通过非常简单的方式发布成 Web 服务，这种处理方式不仅充分发挥了 POJO 的作用，简化了 Java 应用转化为 Web 服务的步骤和过程,也直接降低了 SOA 的实现难度,为企业转向 SOA 架构提供了一种简单可行的方式。

XFire 内建在 STAX(Streaming API for XML)基础之上,提供了支持 Web Service 的框架与各项协议，能与 Spring 无缝集成，在 Spring 中应用 XFire 开发 Web Service 相当便利[2]。XFire 完全基于流数据处理进行工作，以管道方式接收 SOAP 流数据，从管道中接收一个 SOAP 请求到返回一个 SOAP 响应，会经历多个阶段，如图 2 所示。在管道调用的任何一个阶段，XFire 都可以添加一些额外的 Handler。Handler 用于定义 SOAP 发送和接收之前的各种处理逻辑。

在 SOAP 请求消息对 Web Service 调用之前会经过传输、预转发、转发、策略实施、用户信息处理、预调用、服务调用等阶段。当 Web Service 调用后，XFire 生成响应 SOAP 消息并通过管道发送给客户端请求者，这一过程会先后经历调用后、用户信息处理、策略实施、传输 4 个阶段。每个阶段都是一个可控点，通过开发一些相应的 Handler 即可实施安全方面的处理逻辑，如审计、SOAP 消息加密、签名等。

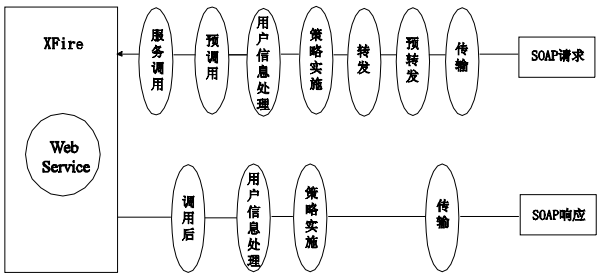

图 2　XFire Web Service 请求和响应的过程

3　XFire和WSS4J下的Web Service组合安全策略

3.1 策略总体方案

XFire 通过 Handler 即可实施 WSS4J[3]，当发送 SOAP 报文时可注册一系列的 OutHandler 对 SOAP 报文作加密处理、签名、添加用户身份信息等后置处理操作，而在接收 SOAP 报文时则通过注册一系列的 InHandler 对 SOAP 报文进行解密、验证签名、用户身份认证等前置操作。策略总体情况如图 3 所示，即“数字签名+报文加密+身份认证”的组合安全策略。

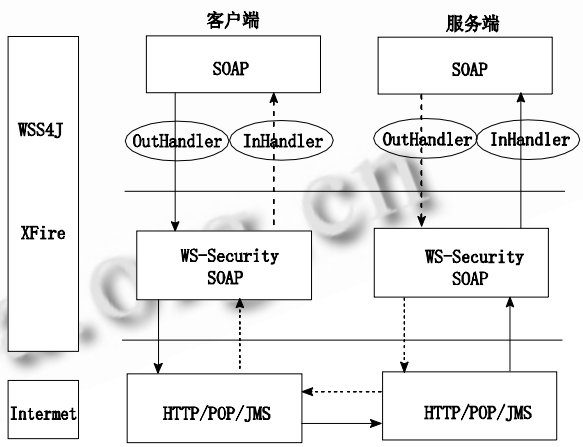

图 3　XFire 和 WSS4J 下的 Web Service 安全策略总体情况

签名和加密需要使用到数字证书和密钥对，如创建 RSA 密钥对、X509 数字证书。服务端和客户端都会有各自的密钥库 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，而客户端的密钥库则保存客户端的密钥对和服务器端的数字证书。

这里需要注意两点，一是 Java 策略文件用于控制加密的强度和算法，包括 local_policy.jar 和 US_export_policy.jar 文件，没有直接包含在 JDK 中，需

要额外增加；二是 WSS4J 使用了 BouncyCastle 的 SecurityProvider，需要事先在 java.security 文件夹中进行配置。

3.2 身份认证的实现

SOAP 报文进行身份认证的方式都是通过在 SOAP 报文头中添加安全凭证信息来实现的，比如“用户名/密码”、“X.509 证书”、“Kerberos 票据和认证者”、“SIM 卡的移动设备安全性凭证”等，其中“用户名/密码”是简单直观的实施方式。

这种方式需要对 SOAP 作前置处理，因此必须将 STAX 流模型的 SOAP 转换为 DOM 模型，再作为一个 DOMInHandler 来完成，与其对应的是 DOMOutHandler 来对 SOAP 报文作后置处理。如果集成在 Spring 中则定义 Action 来作认证的操作。

3.3 对 SOAP 报文作数字签名的方法

身份认证可以实现授权访问，但并不能保证数据在传输过程中不被篡改，从而影响报文的完整性。为保证交易的不可抵赖性，可采用数字签名的方式来实现[4]。

客户端通过私钥对 SOAP 报文进行数字签名，由于私钥只为个人所有，不可抵赖性得到了保证，用于对报文的摘要进行加密，只有报文在传输过程中不被篡改，接收端在作数字签名验证时才能成功，完整性又得到了保证。

3.4 对 SOAP 报文体作加密处理

有了身份认证和数字后，报文体还是以明文方式进行发送，报文内容有可能被监视，保密性较弱。报文中如果有敏感性的内容就得以加密的方式进行传输。客户端使用服务端的公钥对请求 SOAP 报文进行加密，服务端公钥包含在服务端的数字证书中。服务端则使用私钥进行解密，私钥包含在密钥对中。XFire 中通过注册相应的 Handler 即可作加密解密处理。

从请求和服务的角度上来看，Web Service 的交互两端实施组合安全策略时是对等的，如果客户端请求 SOAP 使用了 WS-Security，需要注册并配置 OutHandler，服务端则相应地注册并配置 InHandler。

如果是在 Spring 中集成应用，由于一般的 SOA 应用系统并非两点交互的系统，大多拥有多个客户端，由于服务端需要使用到客户端的数字证书，而数字证书是动态设置的，需要发送给哪个客户端就使用哪个客户端的数字证书，故不适合在 Spring 配置文件中定义响应的 SOAP 报文处理信息。

4 策略应用实验

在现代物流实验中心作出了本文中所述 Web Service 组合安全策略的应用实验。在物流数据交换平台中需要与多个系统作集成应用，如图 4 所示。

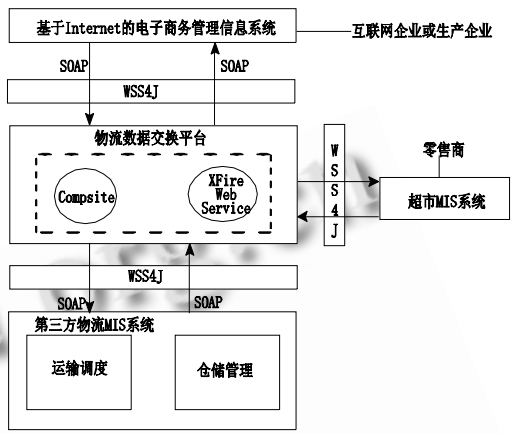

图 4　物流数据交换平台 Web Service 的组合安全策略

考虑到实验条件的限制以及本实验的需要，直接使用了 KeyTool 来创建交易各方的密钥对和数字证书，在全社会推广应用的系统中，应当使用可信任的第三方 CA 颁发的数字证书。

比如在超市 MIS 系统向第三方物流 MIS 系统下达运输订单时，通过注册和配置的 OutHandler 先在 SOAP 报文头中加入身份认证信息，再将数据交换平台的公钥对报文体加密，用数据交换平台的数字证书作签名处理。数据交换平台的 XFire 在收到 SOAP 报文后，通过 InHandler，首先在数字签名验签，如果验签成功再作身份验证，身份验证成功后，用私钥解密 SOAP 报文体，取出订单业务数据，调用订单业务数据处理 Web Service，将处理数据放入到数据库中。第三方物流 MIS 系统每隔一段时间发送 SOAP 报文，处理过程同以上过程，调用获取订单的 Web Service 获取订单数据。订单的处理结果最终要通过电子商务系统对外展现。

在数据交换平台内部，使用了 Tuscany 作为 SCA 的中间件软件，这样可以方便地集成 Web Service 及采用其它各种接口技术的应用系统[5]。获取订单数据 Web Service 和接收订单数据 Web Service 的外在表象为一个 composite，内含有多个 component[6]，

(下转第 64 页)

(上接第 56 页)

完成处理前后的各种业务逻辑，诸如日志处理，数据字段加工等。

实验结果表明，XFire 和 WSS4J 下的“数字签名+报文加密+身份认证”组合安全策略方案可行，良好地支持了 SOA 架构下的系统集成。

5 结语

单一的安全机制都不足以解决 Web Service 的安全性问题，由于 XFire、WSS4J、Spring 集成的方便，以及 SOA 应用的需要，采取“数字签名+报文加密+身份认证”式的组合安全策略让 Web Service 通过 SOA 应用中间件提供给各方使用，实验证明方案也是可行的，这对推动服务库技术的研究与进展，SOA 架构在企业级应用集成上的进一步应用都将具有重要的现实意义。

参考文献

1 陈雄华.精通 Spring2.x 企业应用开发详解.北京:电子工业出版社, 2008:530－531.

2 汪丽才.基于 Axis 和 WSS4J 实现 SOAP 消息安全. 西南科技大学学报, 2006,15(4):32－35.

3 谢明明.XML Web 服务的安全模型的应用研究[硕士学位论文].上海:华东师范大学, 2007:6－8.

4 续亚锋.一种 Web 服务安全模型的研究与应用[硕士学位论文].郑州:河南大学, 2008:8－12.

5 Component Architecture(SCA)-focus on policies. 2007-10.http://www.osoa.org/download/attachments/250/SCA_OASIS_Tutorial_part2.pdf version=1.

6 邓子云.贯通 Java Web 轻量级应用开发:JSP+Struts+Hibernate+Spring 实例精解.北京:电子工业出版社. 2008,7:503－505.