

基于 Web Service 技术的 MOM 安全性模型

梁 明, 徐宏炳, 毛 赞

(东南大学 计算机科学与工程学院, 江苏 南京 210096)

摘 要: Web Service 作为一门新兴的技术, 在中间件技术中有广阔的应用前景。但是, 现有的 Web Service 安全传输方案, 不适用于应用层的消息安全保护。因此, 这种结合方式将会为消息中间件系统带来安全隐患, 无法保证消息中间件的安全消息传递。文中提出了一个基于 Web Service 技术的合理的中间件模型。并结合消息中间件的特点, 在分析 Web 服务安全技术的基础上, 进一步提出了消息中间件的安全传输模型, 同时阐述了保证 SOAP 消息安全传输的基本方法。设计了平台无关的实现方式以体现出应用的透明性。

关键词: Web Service; 消息中间件; 安全

中图分类号: TP393.08

文献标识码: A

文章编号: 1673-629X(2008)09-0119-04

MOM Security Model Based on Web Service

LIANG Ming, XU Hong-bing, MAO Yun

(School of Computer Science & Engineering, Southeast University, Nanjing 210096, China)

Abstract: Web Service as a new technology has a good view in the field of middleware. But existing transport solutions for Web Service are not suitable for protecting message security at application layer. So, obviously, this way for MOM leads to more security issues and can not meet the requirement for MOM security. In the need of network-safety, a sound model of MOM based on Web Service is put forward. Furthermore, combining the features of MOM and the analysis of Web Service security, a security model is proposed. And a new approach for secure transmission of SOAP messages is presented later based on this model. Finally, a platform independency realization is devised to achieve application transparent.

Key words: Web service; MOM; security

0 引 言

消息中间件(MOM, Message-Oriented Middleware)是中间件中非常重要的一种。它屏蔽了网络硬件平台的差异性和操作系统与网络协议的异构性, 能在不同平台之间通信, 实现分布式系统中可靠的、高效的、实时的跨平台数据传输, 使应用软件能够平滑地运行于不同平台上, 实现互连互通。

传统的消息中间件专注于应用在企业局域网环境中, 在扩展到 Internet 上, 特别是 Web 环境中时就显得有些不足。适用于 Internet 环境的新型的消息中间件是解决 Web 服务中异步、可靠消息交换的有效方案。

1 基于 Web Service 技术的消息中间件

1.1 Web Service 技术的特点

(1) 互操作性: Web Service 之间可以自由进行交互, 同时, 由于 SOAP 协议是所有主要供应商都支持的新标准协议, 因而避免了在 CORBA、DCOM 和其他协议之间转换的麻烦。另外, Web Service 不受编程语言的限制, 开发者无须更改其开发环境, 就可生产和使用 Web Service。

(2) 真正的跨平台: 可以使用任何语言来编写 Web 服务, 可以运行在任何操作系统之上。

(3) 简单性: Web 服务使用 HTTP 和 XML 进行通信, 实现松散耦合。

(4) 行业支持: 所有主要的供应商都支持 SOAP 和 Web Service。如微软的 .NET 平台就是基于 Web Service 的。

1.2 基于 Web Service 技术的消息中间件框架

如图 1 所示, 消息中间件的服务器部分部署在 HTTP 服务器中, 它分两部分组成: 安全代理和消息代

收稿日期: 2007-12-27

基金项目: 江苏省十五高技术计划(BG200434)

作者简介: 梁 明(1979-), 男, 甘肃酒泉人, 硕士研究生, 研究方向为数据库设计与应用、数据集成、系统结构; 徐宏炳, 教授, 研究方向为数据库设计与应用、数据挖掘技术、企业数据集成。

理,其他部分还有 Client 和 DB Server,主要部分的功能如下:

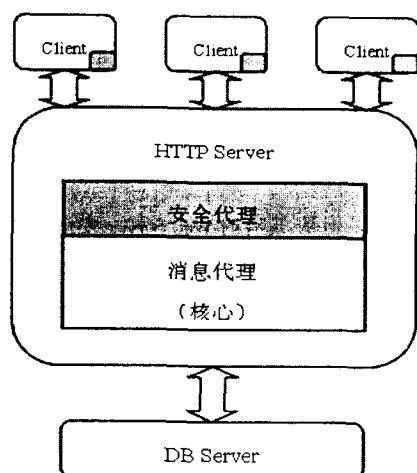


图 1 基于 Web Service 技术的消息中间件总体模型

(1) Client 部分是发布给客户的部分,为客户提供一个轻量级的接口,API 结构采用会话模式。客户通过合法性身份验证后,发起一个连接并获得系统返回的唯一会话标识。客户利用这个连接创建发布者或订阅者对象,并使用唯一的会话标识发布或订阅消息,同时完成 SOAP 消息的加密和解密。在集群环境中,它还负责发送获取负载最轻的服务器的请求。

(2) 代理部分。安全代理,包括了时间戳处理、签名处理和加密处理等。通过在 SOAP 头中加入签名、证书、加密的 Session Key,在 SOAP Body 中加入经过加密的明文 SOAP 来构造安全的 SOAP。消息代理,管理和控制进出中间件的消息并进行相应的逻辑处理,最后将消息返回给 Client。在集群环境中,消息代理还负责采集其他被管理服务器的负载信息,以及根据负载信息分发任务。消息代理和安全代理可以做插件形式,方便服务器端扩展。

(3) DB Server 负责消息的持久化,在服务器的负载过重时,将过多的消息从内存中存储在 DB Server 中,当服务器的负载较轻时,将 DB Server 中的消息再放回内存中。此外,当服务器关闭时,将内存中的消息存储到 DB Server 中,开启服务器时再从 DB Server 中读回内存。

1.3 基于 Web Service 技术的消息中间件优越性

这种消息中间件用 SOAP 来封装消息,具有编程语言无关性、平台无关性;利用 HTTP 协议传递数据,能穿越防火墙,这使得消息能在不同的网络环境中传输;Web Service 协议都是开放的标准,这使得设计基于 Web Service 技术消息中间件有良好的兼容性和扩展性;Web Service 的安全技术、Web 服务器的负载均衡等技术通过改造就可以用于消息中间件中。

2 安全性分析

2.1 网络安全的五个基本要求

(1) 机密性(Confidentiality):保证没有经过授权的用户、实体或进程无法窃取信息。在一个开放的网络环境里,维护信息机密是全面推广应用的重要保障。因此,要预防非法的信息存取和信息在传输过程中被非法窃取。

(2) 授权(Authorization):授权是确定允许用户做什么的过程。可将不同的特权给予不同类型的用户。例如,每个人都能阅读公共图书馆的联机卡片目录,甚至不必是该系统的认证用户。换句话说,所有用户都被授权可阅读目录。但系统可能会将借书的权限仅限于已认证用户,这里已认证是指持有此图书馆的有效借书卡。取决于认证机制的复杂程度,系统可能根据所持的卡来限制用户的特权。例如,可能授权某些用户可以借阅的书不限数量,而限制其他用户只能借阅一定数量的书籍。

(3) 数据完整性(Data integrity):保证没有经过授权的用户不能改变或者删除信息,从而信息在传送的过程中不会被偶然或故意破坏,保持信息的完整和统一。因此,要预防对信息的随意生成、修改和删除,同时要防止数据传送过程中信息的丢失和重复。

(4) 原始性证明(Proof of Origin):对信息或数据的发送者进行标示。保证信息被经过标示的发送者所传送,从而避免以前的数据包被重复发送。

(5) 防止抵赖(Non repudiation):保证信息的发送者不能抵赖或否认对信息的发送。当然信息发送前需要对发送者进行安全认证,为信息的传输过程中为参与交易的个人、企业或国家提供可靠的标识。

2.2 Web Service 的安全技术

Web Service 是基于 SOAP 来传递消息的而 SOAP 就是 XML 文档,因此 Web Service 存在先天的不足——安全性,SOAP 消息都是以明文传送的,缺乏必要的认证加密手段。原有的 Web Service 安全机制主要依赖于传输层的安全协议如 SSL, TLS 等,如图 2 所示。这一方案的不足是:当中介转发者接收、处理和转发消息时,超出了传输层的点到点安全范围,SOAP 消息在经过中间节点处理时,SOAP 消息对中间节点是可见的,只能提供点到点(peer-to-peer)的安全性,造成安全漏洞。

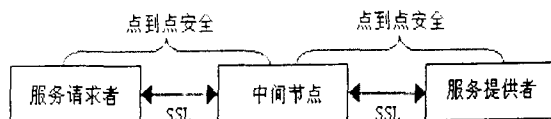


图 2 点到点安全模型

WS-Security^[1]是一个基于 XML 的安全性元数据容器的规范,它通过在现有规范中添加一个架构,用于将现有安全机制嵌入到 SOAP 消息中。WS-Security 的主要目的之一是实现端到端(end-to-end)的安全(如图 3 所示),保证 SOAP 消息通过不安全的中间节点,安全可靠地从服务请求者到达服务提供者。为了实现这一目的,WS-Security 不再依赖传输层安全机制,而是直接在 SOAP 信息头中嵌入安全信息(如数字签名,X.509 证书等),并对需要保密的数据进行加密。SOAP 节点到达目的节点后,由服务提供者直接验证这些安全信息的真伪并解密相关数据。这样就避免了对中间节点的依赖,从而实现了端到端的安全性。

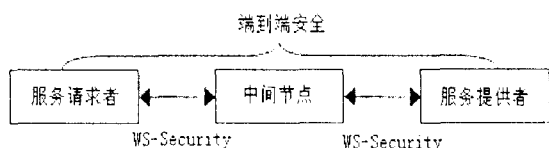


图 3 端到端安全模型

WS-Security 规范为 Web Service 应用的安全提供了保证。简单说来,WS-Security 使用 XML Signature^[2]对 SOAP 消息进行数字签名,保证 SOAP 消息在经过中间节点时不被篡改,从而保证消息完整性;WS-Security 使用 XML-Encryption^[3]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;WS-Security 引入安全令牌(Security Token)的概念,安全令牌代表 Web 服务请求者的身份,通过和数字签名技术结合,服务提供者可以确认 SOAP 消息由合法的服务请求者产生,完成身份验证。更重要的是,WS-Security 提供了一层足够灵活的基础安全机制,基于这一机制,可以根据具体的 Web Service 应用环境,构筑更完善的安全模型,例如基于 WS-Security 开发的 Web 服务信托模型 WS-Trust^[4]等。

虽然 WS-Security 提供了一套完整的关于 Web Service 的身份认证、权限管理、消息加密和消息完整性的方案^[5],但有些技术不适合用在消息中间件中,如需要第三方提供证书或者必须有第三方参与等。下面结合 WS-Security 技术和消息中间件的特性^[6,7],提出基于 Web Service 技术的消息中间件的安全设计思路:

(1) 按需要对 SOAP 包封装的信息(Body)或其关键元素进行加密。首先要选用合适的加密算法进行加密处理。为了兼顾速度和安全性,可以采用 DES 标准对称分组密码算法,加密强度(位数)可根据实际需要选定(如 128 位)。

(2) 整个 SOAP 包进行完整性监测,可以采用 MD5 单向 HASH 签名密码算法对 SOAP 包取 HASH

值,并插入 SOAP 包传递。此过程用来保证消息在传输的过程中不会被篡改。当接收方收到消息后,会利用已经接收到的消息内容和已知散列算法重新计算散列值,并与收到的散列值进行比较。

(3) 保证用户的合法性,通过向用户提供私钥,自己保留公钥,通过公私密钥的匹配来验证用户的合法身份。在将客户端程序交给用户时,生成一对公钥和私钥,公钥存放在服务器端,私钥交给用户。私钥是用户合法身份的凭证。公钥和私钥同时用于加密客户机与服务器端的通信 DES 加密的密码和数字签名。

(4) 权限方面,将用户的权限分为发布消息、订阅消息、创建消息、删除消息等,这些权限可以组合。

3 基于 Web Service 技术的消息中间件的安全代理模型

根据上述消息中间件的设计,几乎所有的消息都通过 SOAP 来传递。因此,保证 SOAP 消息创建、发送、传输和接收全过程的安全性是消息中间件安全传输方案的设计基础,而这一基础又以 SOAP 头部安全扩展和信息加密为核心。

图 4 是此消息中间件的安全代理模型^[8]。

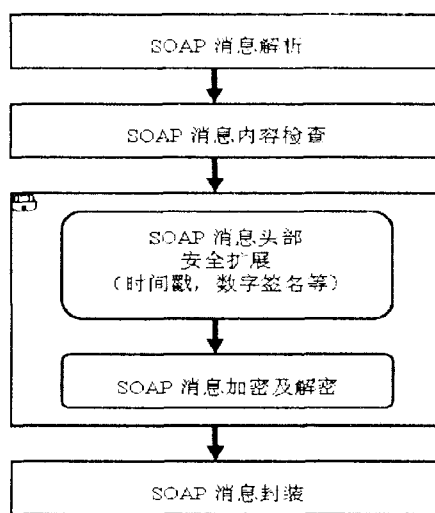


图 4 基于 Web Service 技术的消息中间件的安全代理模型

主要功能:

1) 时间戳模块。创建一个用于确定消息发送者即 SOAP 签名者的<Timestamp>模板,包括以下几个元素:<Createtime>,<Received>,<Lifetime>和<Receiver>。时间戳服务在收到时间戳请求后,读取系统时钟创建<Received>和<Created>元素的标记值,在给每个元素进行赋值后,把这些元素值封装在<Timestamp>标签中作为时间戳服务请求的计算结果返回给请求者。

2) 数字签名模块。XML 签名生成过程为: 创建 <Reference> 元素, 应用 <DigestMethod> 子元素指定的算法对数据对象计算摘要值, 应用 <DigestValue> 子元素收集摘要值; 创建 <SignedInfo> 元素, 它可以包括一个或多个 <Reference> 元素, 其中 <SignatureMethod> 子元素用来指定签名算法标识符, <SignatureValue> 子元素存放生成的签名值; 创建 <Signature> 根元素, 包括 <SignedInfo> 和 <SignatureValue> 子元素, 加上可选的 <KeyInfo> 和 <Object> 子元素, 形成完整的 XML 签名。

签名验证用来验证从 <SignedInfo> 元素计算得来的加密签名, 签名验证从获取验证钥开始, 验证密钥从 <KeyInfo> 元素或者应用程序的密钥源得到, 然后使用此密钥和规范化的 <SignatureMethod> 元素中指定的算法对规范化形式的 <SignedInfo> 元素计算签名值, 将此签名值和 <SignatureValue> 元素内的值做比较, 若两者匹配, 则验证成功; 否则, 验证失败。

3) 数字加密模块。主要功能为: 客户端请求消息的加密和解密、服务器端响应消息的加密和解密。加密时, 首先选定加密方法, 同时生成一个初始化密钥, 采用密钥信息元素记录密钥。然后对 SOAP body 对象利用初始化密钥进行加密, 被加密部分由 <EncryptedData> 元素取代, 其中包括加密算法、加密结果及密钥信息。

密钥信息采用对方公钥进行再次加密, 结果包含于 <EncryptedKey> 元素中, 这样只有拥有相应私有密钥的接收方才能够获得对消息加密的对称密钥。解密时, 先进行完整性验证, 由 <EncryptedData> 提供的解密信息和公钥还原出对称密钥, 最后用对称密钥解密相应的加密消息。

这个模型较好地解决了安全性问题, 针对 Web 服务领域中的安全需求, 该模型提供了相应的解决方法:

(1) 保密性: 对消息进行加密处理, 使得只有合法的消息接收者可以获得消息的内容。

(2) 授权: 通过公私密钥的匹配来验证用户的合法身份。

(3) 消息完整性: 使用数字签名验证消息的完整性。

(4) 消息确认: 使用时间戳和数字签名保证消息来自已知来源并且不是重复消息。

(5) 不可否认性: 使用数字签名和公钥加密保证消息创建者不能否认发出的消息。消息的发送方和接收方在消息的交换过程中, 必须对于自己的行为不可

否认, 以保证不可抵赖。

4 平台无关性实现

Microsoft .NET 与 Sun J2EE 是目前企业 Web Services 平台市场上两个最重要的应用框架, 其中 .NET 对于 SOAP 扩展进行了比较好的设计, 允许在 SOAP 序列化操作时按需要加入 SOAP 扩展。当然, 可以采用一致的实现方法, 屏蔽平台的差异性, 这也是消息中间件追求的目标。在实现时, 客户端采用 HTTP 代理可以屏蔽浏览器、应用程序的差异性。服务器端采用消息插件机制, 允许在客户端的 HTTP 请求送到应用程序前处理 HTTP 数据包, 这样, 就使得消息插件能够完成加密解密、验证、处理 SOAP 消息等工作。采用这样的方案, 可以实现平台无关, 同时对应用程序也做到透明。而且, 当需要升级 Web Service 时, 通过配置就能完成。

5 结束语

提出了基于 Web Service 技术的消息中间件的实现模型, 并根据这种模型给出了相应安全传输的设计, 有较好的应用前景。当然, 随着 Web Service 安全技术的发展, 以及新的中间件模型的提出, 将会有更多更好的安全传输方案出现。

参考文献:

- [1] 柴晓路. Web 服务架构与开放互操作技术[M]. 北京: 清华大学出版社, 2002.
- [2] W3C Working Draft. XML Encryption Syntax and Processing [EB/OL]. 2002-12-10. <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210>.
- [3] W3C Proposed Recommendation. XML Signature Syntax and Processing [EB/OL]. 2001-08-20. <http://www.w3.org/TR/2001/PR-xmlsig-core-20010820>.
- [4] Security in a Web Services World: A Proposed Architecture and Roadmap [EB/OL]. 2002-04-01. <http://www-106.ibm.com/developerworks/i2library/ws-secmap>.
- [5] 严毅, 宁葵, 唐天兵. Web 服务的安全技术[J]. 微机发展, 2005, 15(9): 65-67.
- [6] 陈明, 潘家铭, 阎保平. 消息中间件的设计与实现[J]. 微电子学与计算机, 2005(4): 4-7.
- [7] 刘宇晓, 吴宇红. 有关消息中间件的安全问题分析[J]. 电子科技, 2004(8): 32-36.
- [8] 付登科, 郝克刚, 葛玮. 一个 Web 服务消息级别的安全模型研究[J]. 微机发展, 2005, 15(11): 30-33.