用 WSE 构建安全可靠的 Web Services

续亚锋,陈志国,李 涵
(河南大学 计算机与信息工程学院,河南 开封 475004)

摘 要:在 Web Services 应用中,保证传输消息的安全性和完整性非常重要,而当前技术下的 Web Services 却在安全性方面存在着一些不足。采用 WSE 扩展框架来增强 Web Services 的安全性,已成为新的研究热点。文中通过 X.509 证书实现了 Web Services 消息签名和加密,阐述了利用 WSE 构建安全 Web Services 的方法,并结合高招志愿采集系统中的 Web Services 安全传输方案,给出了部分关键代码。

关键词:Web 服务;WSE;X.509 证书;策略
中图分类号:TP393 **文献标识码**:A **文章编号**:1673-629X(2008)08-0155-04

Construction of a Safe and Reliable Web Services with WSE

XU Ya-feng, CHEN Zhi-guo, LI Han
(College of Computer and Information Engineering, Henan University, Kaifeng 475004, China)

Abstract: It is vital important to protect the data integrity and security in Web services, while the Web data transmission is insufficient to be transmitted successfully and safely because of the current develping of Web services. Using WSE to expand the secruity of Web service is a new solution under the existing framework of Web services. Aims at an entire description of constructing a safe Web services with WSE through the X.509 certificate's signatures and encryption. In addition, a partial code of College Students Entrance Candidates Information Collection System is given at the end of the paper to show how to construct a safe and reliable Web services with WSE.

Key words: Web services; WSE; X.509 certificate; policy

0 引 言

Web Services 作为一种新的 Web 应用程序分支,是基于网络的、分布式的模块化组件,它通过借鉴和利用现有的 Internet 开放互联标准,在现有的各种异构平台上构造了一个通用的与平台、语言无关的技术层规范,并实现了不同软硬件平台上应用的互联与互操作。Web Services 以微软的"软件即服务"(Saas, Software as a Service)思想为理念,通过对 Web 服务的调用,可以实现从简单请求到复杂商务处理的任何功能[1]。

在使用 Web Services 的过程中,经常需要传输一些隐私或机密数据,譬如信用卡信息、股票交易信息,或者高招志愿数据、高招录取信息等,而这些数据在公网上的直接传输是不安全的,很容易受到某些人的窃取和非法篡改,通常所采用的基于传输级的通信方法,譬如 SSL 或 IPSec,只能解决点到点的通信安全,且需要加密所有通信数据。而基于消息级别的解决方案,则可以有效地实现端到端的安全:在 SOAP 消息中嵌入安全信息,对消息数据有选择地进行签名和加密,消息在通过多个中间应用程序节点路由后由目的节点进行验证和解密。正是基于消息级别数据传输的优越性和高效性,微软推出了用来解决 Web Services 安全问题的 WSE 扩展框架。

收稿日期:2007-11-15
基金项目:河南省自然科学基金项目(0411014100)
作者简介:续亚锋(1979-),男,河南灵宝人,硕士研究生,研究方向为 Web Services 和语义网格;陈志国,教授,研究方向为软件理论、人工智能。

1 使用 WSE 构建安全 Web Services

1.1 WSE 框架介绍

WSE(Web Services Enhancements for .NET)是微软为.NET 开发人员推出的 Web Services 安全增强包,它支持最新的 Web Services 协议,可以实现 WS-Security, WS-SecureConversation, WS-Trust, WS-Policy, WS-SecurityPolicy, WS-Addressing 和 WS-Attachments 等高级 Web Services 协议。WSE 框架实现了在 SOAP 中引入了 XML 数字签名和 XML 加密,解决了如何在多点消息路径中维护一个安全的数据传输环境。目前 WSE3.0 已经与 VS2005 的 IDE 集成在一

起,开发人员可以快速便捷地建设一个跨平台、安全的、可升级的 Web Services。在 WSE 中可以使用用户名/密码、X.509 证书、Kerberos 票证以及其它自定义二进制文件或基于 XML 的安全令牌来作为消息签名和加密的凭据。

1.2 运行流程

实现 Web Services 安全通信的重要手段是对 SOAP 消息的签名和加密。签名可以保证数据完整性,防止被篡改;加密可以保证数据的机密性,防止数据被非法获取。实现时,使用者可以同时应用身份验证、授权和安全通信来建立客户端和 Web 服务器之间的会话安全。基于 X.509 证书的消息通信过程如下:

客户端(如图 1 所示):客户端从证书存储区(Certificate Store)获取服务器端证书和客户端证书,将客户端的证书绑定在消息信息上,使用客户端证书的私钥对消息进行签名,使用服务器端证书的公钥对消息进行加密,最后将加密后的消息发送至服务器端。

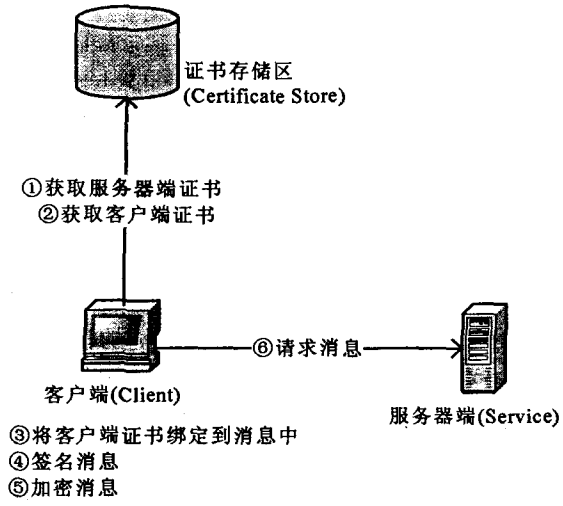

图 1 客户端请求流程

服务器端(如图 2 所示):服务器端接收到客户端消息后,先校验客户端证书是否合法,然后通过证书中心(CA,Certificate Authority)检查证书信任链,接着查询证书的撤销状态,若以上步骤均得到验证,则服务器端用自身证书的私钥对消息进行解密,利用消息中附加的客户端证书私钥验证消息签名,最后,服务器端根据客户端请求返回反馈信息。

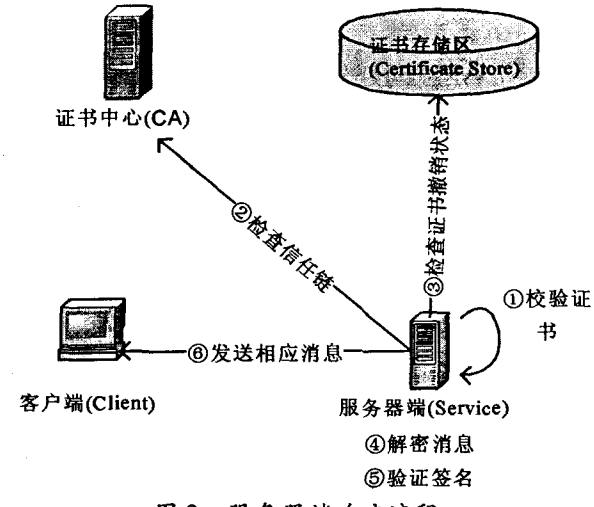

图 2 服务器端响应流程

2 用 WSE 保证 Web Services 安全性

在笔者开发的高招志愿项目中,由于需要通过 Internet 传输一些敏感信息,因而选择了基于 WSE 框架的 X.509 证书来保证数据传输的机密性和完整性。WSE 策略框架描述了与 ASMX 或 WSE Web 服务进行通信的约束和要求,在此框架下可以比较容易地实现 Web Services 的数据处理。后面的代码片段简要说明如何在 WSE 框架下实现客户端和 Web Services 之间的安全通信和会话。文中的代码,均在 Visual Studio 2005 的 C# 语言和 WSE3.0 环境下得到验证。

2.1 制作 X.509 证书

证书生成方法有多种,此处仅简要说明几种基于 Windows 的证书生成方法:

(1)通过 CA 获取证书。

在 Windows 2003 Server 或 Windows 2000 Server 中的安装证书服务组件,利用 Microsoft 证书服务进行证书申请[2]。

(2)通过微软提供的 makecert 工具制作测试证书[3]。

例如使用 makecert -r -pe -n"CN=XYF-Company" -b 01/01/2007 -e 01/01/2010 -sky exchange -ss my 可以创建一个自我签署的证书,指定使用者名称为"CN=XYF-Company",有效期为 3 年(2007.01.01 至 2010.01.01),并将密钥放入 my 存储区(证书的个人文件夹),指定并交换密钥,且私钥可导出。

(3)利用. Net 的 X509Certificate2 类编程创建证书[4]。

2.2 WSE 环境配置

要实现 WSE 中的方法及功能,首先要在开发环境下加入了 WSE3.0 运行支持部件,即确保在运行环境中已经安装 WSE 3.0 增强包、在代码中已加入了 Microsoft. Web. Services3. dll 引用和 Microsoft. Web. Service3 命名空间,然后需要分别配置 Web 服务器端和客户端。

WSE 的配置可以通过两种方法实现:一种是在 .NET开发环境中利用代码来实现;另一种是通过 WSE 的配置工具来进行设置。在代码中以命令方式定义策略时,可以完全控制为使用服务而需要满足的

特定要求,程序发布后,如果需要改变策略,则需要重新编译程序[5];而使用 XML 策略文件(通常使用 WSE 配置工具来设置),管理员则可以在程序部署后,根据不同的需要灵活地选择和使用策略。针对不同级别和不同层次的安全要求,配置时可以选择不同的认证方式和加密方式,文中采用 XML 策略文件和基于 X.509 证书的认证方法来签名和加密 SOAP 消息。

通过对 WSE 进行配置,最终在客户端和服务器端的配置文件中会增加以下代码:

```
<microsoft.web.services3>
……
  <policy fileName="wse3policyCache.config" />
  <security>
    <binarySecurityTokenManager>
      <add valueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" >
      </add>
    </binarySecurityTokenManager>
  </security>
</microsoft.web.services3>
```

同时,客户端和服务器端会建立其相应的 XML 策略文件(wse3policyCache.config),客户端的策略文件内容片断如下(服务器端略):

```
<policies xmlns="http://schemas.microsoft.com/wse/2005/06/policy">
  <policy name="ClientEncrypt">
  <mutualCertificate11Security establishSecurityContext="true" requireSignatureConfirmation="true" messageProtectionOrder="SignBeforeEncryptAndEncryptSignature" requireDerivedKeys="true">
      <clientToken>
      <x509 storeLocation="CurrentUser" storeName="My" findValue="CN=HENUUser" findType="FindBySubjectDistinguishedName"/>
      </clientToken>
      <serviceToken>
      <x509 storeLocation="CurrentUser" storeName="AddressBook" findValue="CN=HENUZY" findType="FindBySubjectDistinguishedName"/>
      </serviceToken>
      </mutualCertificate11Security>
  <requireActionHeader/>
</policy>
……
</policies>
```

通过 XML 策略文件,程序运行时会在指定的位置查找相应证书,并将策略中的要求转换为运行时组件,然后使用各运行时组件分别处理其输入输出消息。另外,为了使程序能顺利运行,必须保证所使用证书安装在正确位置上:在客户端的"个人的"文件夹下安装客户端证书,在服务器端的"信任用户"文件夹下安装客户端证书,在服务器端的"个人的"文件夹下安装服务端证书,在客户端的"其他人"文件夹下安装服务端证书。

2.3　程序实现

* 客户端程序。

客户端在调用 Web Services 前,需要先对其进行实例化,即创建一个 Web 服务代理类,Web 服务代理类在客户端充当 Web Services 的角色,并在后台实现和远程 Web Services 的通信。客户端根据其策略配置文件的要求和服务器端进行数据交互。通信过程中客户端先发送 SOAP 请求,从安全令牌服务处获得安全令牌,然后用得到的安全令牌签名和加密调用 Web 服务的请求,接着调用目标 Web Services,并验证来自目标 Web Services 的响应是否也经过签名和加密。以下是客户端的代码片段:

```
//创建一个 Web Services 代理的实例
MyServiceWse UpLoadService = new MyServiceWse();
//利用 WebServicesClientProtocol 提供的 SetPolicy 方法为 Soap 信息指定策略
UpLoadService.SetPolicy("MyClient");
//从程序配置文件 App.config 中获取 Web Services 的 URL
UpLoadService.Url = ConfigurationManager.AppSettings["WseUrl"];
//调用 Web Services,并返回相应的信息
return UpLoadService.UpLoadKSXX (strUserName, strToken, FileFactory);
……
```

* Web 服务端程序。

服务器先根据保存在信任文件夹下的用户端证书,判断所接受的用户请求是否合法,如果请求合法,则使用服务器端 X.509 证书的私钥对消息进行解密,解密后利用附加在消息上的客户端证书的密钥解密正文,然后再对消息的正文做进一步处理。在应用程序域中首次实例化服务类时,服务器端将 WSE 的 PolicyAttribute 属性所指定的策略文件解析为一个 Policy 实例,以便 SOAP 消息应用其要求。以下是服务端代码片段:

```
[WebService(Namespace="http://work.henu.edu.cn/Myserver")]//Web Services 声明
[WebServiceBinding(ConformsTo=WsiProfiles.BasicProfile1_1)]
[Policy("ServerPolicy")]//指定服务器端策论
  public class MyService:WebService
  {
```

```
    public string UpLoadKSXX(string strUserName, string Token,
FilePackage KSXX){
    //调用存储过程在服务器端进行数据处理
    GetProcCommand("Import_KSXX");
    ......
  }
  ......
}
```

这样,通信的双方(客户端应用程序和 Web 服务)就在 WSE 框架下建立了一个基于 X.509 签名和加密的安全通信环境。

3 结束语

XML 技术和 Web Services 技术的诞生为应用 XML 进行独立于平台的数据与系统集成提供了解决方案,以 Web Services 为主的软件实体以开放、自主的方式存在于 Internet 的各个节点之上,并以各种协同方式与其他软件实体进行跨网络的互联和协作,从而 Internet 逐渐从信息发布共享平台演变成一个大规模的分布式计算平台。

WSE 的出现,在一定程度上解决了制约 Web Services 发展的安全问题[6],保证了 Web Services 的安全通信,加速了在世界范围内资源的共享。

参考文献:

[1] 张尧学,方存好. 主动服务——概念、结构和实现[M]. 北京:科学出版社,2005.

[2] 从 Microsoft 证书服务导入 CA 证书[S/OL]. Microsoft Corporation. 2005 - 01. http://www.microsoft.com/technet/prodtechnol/windowsserver2003/zh - chs/library/ServerHelp/e4c46832 - 6fd2 - 4e42 - 9f1d - 8ebe84ce3ff1.mspx?mfr=true.

[3] 证书创建工具(Makecert.exe)[EB/OL]. Microsoft Corporation. 2003 - 03. http://msdn2.microsoft.com/zh - cn/library/bfsktky3.aspx.

[4] 使用 X509Certificate2 类编程[EB/OL]. Microsoft Corporation. 2005. ms - help://MS.MSDNQTR.v80.chs/MS.MSDN.v80/MS.NETDEVFX.v20.chs/CPref18/html/T_System_Security_Cryptography_X509Certificates_X509Certificate2.htm.

[5] Janczuk T. Protect Your Web Services Through The Extensible Policy Framework In WSE 3.0[J]. MSDN Magazine, 2006,21(2):50 - 62.

[6] 岳 昆,王晓玲,周傲英. Web 服务核心支撑技术:研究综述[J]. 软件学报,2004,15(3):436 - 438.

(上接第 154 页)

能,而现在的通用操作系统也都根据这个硬件特点来把操作系统划分成不同的运行级别,以保护内存数据。

3 发展趋势

随着安全性需求不断渗透到电子系统设计的各个环节,嵌入式系统设计面临着前所未有的挑战。过去,只有少量的电子设备用户才会考虑安全性问题,而且主要集中在金融行业、军用产品、门禁控制等,大多采用相关的软件技术或专用硬件实现。这些专用的层出不穷的安全标准和需要获得相关的产品认证使开发难度大大增加。因此迫切需要一个统一的安全标准来指导开发。

由于通过软件/固件设计很难保证全面的系统安全性,这就需要借助硬件保证系统的安全性,降低设计的复杂度。加密算法已不再是防范攻击的重点。攻击者会通过各种途径窃取密钥。因此,嵌入式系统的设计不在加密算法上投入过多精力,而是将注意力转向硬件保护的设计上。就目前的软硬环境而言,未来的嵌入式系统安全技术的发展趋势还将是以软硬件增强方式为主导的。

4 结束语

随着嵌入式系统的网络化,嵌入式系统的安全必将提上日程。因此研究嵌入式系统的安全技术是非常有必要的。嵌入式系统的安全不是安全嵌入式系统的附加功能,安全嵌入式系统要在整个设计过程中考虑安全因素,从而决定采用哪种安全技术。

参考文献:

[1] Festa P, Wilcox J. Experts Estimate Damages in the Billions for BUG[EB/OL]. 2000 - 05 - 05. CNET News.com.

[2] Ravi S, Raghunathan A, Kocher P, et al. Security in Embedded Systems: Design Challenges[J]. ACM Transactions on Embedded Computing Systems (TECS), 2004, 3(3): 461 - 491.

[3] 洪 帆,陈 卓,王瑞民. IPSec 安全机制的体系结构与应用研究[J]. 小型微型计算机系统,2002(8):946 - 949.

[4] 曹煦晖,李传目. 信息安全与加密技术研究[J]. 微机发展,2003(增刊):76 - 78.

[5] 陈志平,雷 航,杨 霞,等. 嵌入式安全操作系统的研究和实现[J]. 计算机工程,2007(1):83 - 85.

[6] Proctor P E. 入侵检测实用手册[M]. 邓琦皓,许鸿飞,张斌,译. 北京:中国电力出版社,2002.