第32卷 第17期 计 算 机 工 程 2006年9月
Vol.32 № 17 Computer Engineering September 2006
·安全技术· 文章编号：1000—3428(2006)17—0227—03 文献标识码：A 中图分类号：TP309

高性能可信Web Service研究

陈荤祺，陈克非
(上海交通大学计算机科学与工程系，上海 200030)

摘 要：传统的Web Service以文本的方式传送SOAP包，存在安全性和性能等方面的问题。为了解决这些问题，提出了一种新的Web Service处理模型，通过将PKI技术、数据压缩技术与Web Service技术的结合，形成了可信、高性能的Web Service解决方案。并设计了平台无关、应用透明的实现方式。

关键词：Web Service；可信；安全；高性能

Research of Trustworthy High Performance Web Service

CHEN Luoqi, CHEN Kefei
(Dept. of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200030)

【Abstract】 To give the solution of the Web Service security and performance, this paper issues a trustworthy and high efficiency Web Service processing model by integrating the technology of PKI, data compressing and Web Service. And it provides a platform independency, application transparent realization.

【Key words】 Web Service; Trustworthy; Security; High performance

Web Service作为当前在计算机网络应用领域最热门的技术，已经有了许多成功的应用。它采用XML、SOAP、WSDL、UDDI等主要技术，面向对象的信息交换，可支持分布式环境下的信息共享和信息交换，特别适合异构环境中的系统互联，因此成了当前实现SOA架构的最理想的技术。

但是，Web Service是基于SOAP来传递消息的，而SOAP就是XML文档。因此Web Service存在两大先天的不足：一是安全性，SOAP消息都是以明文传送的，缺乏必要的认证、加密手段；二是性能，XML信息的高冗余，使它相对于其它远程调用技术，在通信传输上的开销要大得多。如果不能解决好Web Service的这两大问题，必然极大地制约Web Service的发展。

本文提出了一种可信、高性能的Web Service技术模型，通过引入PKI、数据压缩等技术，有效地解决了以上问题。并且，这种模型是对应用透明的，也不与具体的平台相关。

1 解决方案

1.1 可信的Web Service

可信的Web Service，就是要保证Web Service数据的保密性、完整性，参与方的身份认证和不可抵赖性。PKI技术对上述要求已经有了很好的解决方案，因此通过引入PKI相关技术，可以有效地解决可信的问题。

SOAP提供了标准的、可扩展的、组件化的XML消息打包和交换的框架。SOAP标准还提供了一个扩展模型，以便实现系统所需的各种特性。利用这个扩展模型，可以将PKI中的各种“元素”(如证书、签名、密钥)，加入到SOAP扩展中，形成一种安全的SOAP规范。引入PKI的核心问题就是解决在SOAP中如何描述PKI中各种元素的问题。

W3C在XML-Signature Syntax and Processing和XML Encryption Syntax and Processing中分别定义了数字签名，加密等如何在XML中表示。IBM、Microsoft、Verisign制定的W-S Security规范，定义了一套标准的SOAP扩展。参照W-S Security，我们这样定义PKI中的主要元素。

1.1.1 X.509数字证书

是一个BinarySecurityToken的安全性令牌，按照Base64编码。

```
<wsse:BinarySecurityToken ValueType="wsse:X509v3"
EncodingType="wsse:Base64Binary"
xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility"
wsu:Id="SecurityToken-d27a5215-0837-4620-ad40-ca4ff02d33f4">
MIIBtDC…
</wsse:BinarySecurityToken>
```

1.1.2 数字签名

```
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
<SignedInfo>
<CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
<SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<Reference URI="#Id-aba44a7d-34f4-4614-bbe0-5c904a6637d8">
<DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
<DigestValue>oA/iw6AUp40Bkwcu0AnxkrSZmc0=</DigestValue>
</Reference>
</SignedInfo>
<SignatureValue>NPb3V4euxhjgqP4tP7f7ybutluo/1…
</SignatureValue>
<KeyInfo>
```

作者简介：陈荤祺(1973—)，男，硕士生，主研方向：信息安全，PKI/CA的互联互通；陈克非，教授、博导

收稿日期：2005-09-23 **E-mail**：ldapa@sina.com

```
<wsse:SecurityTokenReference>
<wsse:Reference
URI="#SecurityToken-d27a5215-0837-4620-ad40-ca4ff02d33f4" />
</wsse:SecurityTokenReference>
</KeyInfo>
</Signature>
```

数字签名用 Signature 表示。里面分为 3 项：<SignedInfo>，<SignatureValue>，<KeyInfo>。

<SignedInfo>里说明规范化方法、签名算法、被签名数据在哪里、转换方法、摘要算法、摘要值等。

这里定义的规范化 Canonicalization 方法是一种唯一的规范化算法，基于 W3C 的“ Exclusive XML Canonicalization ”。由于 XML 是一种文本的表示数据的方法，在 XML 中，多一个空格，少一个换行，都不会影响 XML 的意义。但作为数字签名，对原文的任何修改都会引起验证签名失败。那很容易出现签名方和验证方因为使用不同的规范化方法，导致签名失效。Exclusive XML Canonicalization 就是为了防止这种情况的发生。在这种规范化算法下，XML 只会被规范化成唯一的输出。Exclusive XML Canonicalization 是基于 W3C 的“ Canonical XML ”的。

这里的 SignatureMethod 用的是 RSA-SHA1；<RefernceURI>指向本文档中的一段，表明是对指定的这一段数据签名的；<DigestMethod>，表示摘要算法是 SHA1；<DigestValue>，<SignatureValue>分别指明摘要的结果和签名的结果；<KeyInfo>指定了对一个 SecurityToken 的引用。从 URI 可以看出，就是对上面的 X.509 证书的引用。表示可以用这个证书验证数字签名。

1.1.3 加密

加密分成 2 块：一块是用一个随机的 Session Key 和对称加密算法加密数据；另一块是用数字证书加密 Session Key。

```
<xenc:EncryptedData
Id="EncryptedContent-ddf2320c-9596-4f32-8bef-4b73389dcf34"
Type="http://www.w3.org/2001/04/xmlenc#Content"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
<xenc:EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc" />
<xenc:CipherData>
<xenc:CipherValue>saU0vlu9oKoRJ3Jw…
</xenc:CipherValue>
</xenc:CipherData>
</xenc:EncryptedData>
```

这一段<EncryptedData>表明被对称加密后的密文数据。是用 TripleDES 加密的。

```
<xenc:EncryptedKey
Type="http://www.w3.org/2001/04/xmlenc#EncryptedKey"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
<xenc:EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5" />
<KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
<wsse:SecurityTokenReference>
<wsse:KeyIdentifier
ValueType="wsse:X509v3">e4k6bOdyzuPSAUkyrl+nwdXtNGo=
</wsse:KeyIdentifier>
</wsse:SecurityTokenReference>
</KeyInfo>
<xenc:CipherData>
<xenc:CipherValue>0lp6oZz5N2HIXO….
</xenc:CipherValue>
</xenc:CipherData>
<xenc:ReferenceList>
<xenc:DataReference
URI="#EncryptedContent-ddf2320c-9596-4f32-8bef-4b73389dcf34" />
</xenc:ReferenceList>
</xenc:EncryptedKey>
```

这一段表示一个被证书加密的对称密钥。被加密的密钥用<EncryptedKey>表示。加密的算法是 RSA。<KeyIdentifier>是证书的标识号。<CipherValue>是实际加密后的对称密钥。<DataReference>指向刚才的<EncryptedData>，说明那段加密的数据可以用这个对称密钥解密。

1.1.4 构造安全的 SOAP

有了这些基本元素的表示方法就可构造出安全的 SOAP 扩展：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 Web Service 传递的明文 SOAP。限于篇幅，不再把完整的安全 SOAP 定义列出来了。

1.1.5 从安全到可信

构建起了安全的 SOAP，仅仅解决了 Web Service 数据的保密性和完整性问题。如何保证只有授权的用户才能访问 Web Service？如何保证用户访问的 Web Service 不是一个恶意的假冒服务？在交易出现纠纷时，如何提供举证？这些问题仅靠安全的 SOAP 是不能解决的。

在可信 Web Service 处理模型中，基于安全 SOAP 的基本安全措施，采用在代理层实现双向的身份认证，权限控制，签名数据的存储举证等技术，使其真正成为一个可信的 Web Service。

1.2 高性能的 Web Service

XML 的文本表示方式和数据冗余，使 Web Service 通信时的数据量大增，大量时间消耗在网络通信中。例如，当 Northwind 中的表 orders 中的内容被序列化为 XML 后，数据可能达到 454kB。在可信的 Web Service 中，又引入了签名、加密、证书，数据量又有了增加。例如，为了 base64 编码，数据量增加到 1.33 倍。为了提高性能，需要把通信的 SOAP 包做压缩。对于文本数据，好的压缩算法可以压缩 80%的数据量，压缩比例是很高的，可以有效地减少通信包的大小。

同样，通过 SOAP 扩展，可以构建出压缩的 SOAP 扩展：在 SOAP 头中定义压缩的算法；SOAP Body 中加入经压缩的安全 SOAP 扩展。限于篇幅，具体的 SOAP 扩展不再写出。

2 可信高性能的 Web Service 模型

通过上一节的研究，可以归纳出如图 1 所示的扩展 Web Service 模型。

这是一个链式处理模型。客户端发起的 Web Service 调用首先进入可信 SOAP 处理链，在这里，将完成服务器端的身份认证，客户端签名、加密，生成安全的 SOAP 扩展，记录签名数据。接着进入压缩 SOAP 处理链，生成压缩的 SOAP 扩展。这样的处理次序保证先进行身份认证，签名是针对原文的，所有的数据都经过压缩。

服务器端执行相反的操作，先解开压缩的 SOAP 扩展，得到安全的 SOAP 扩展，验证客户端身份和权限，再还原出 Web Service 调用的原始 SOAP，交 Web Service 应用程序处

理业务。中间任何环节的错误都将使本次 Web Service 调用产生异常。

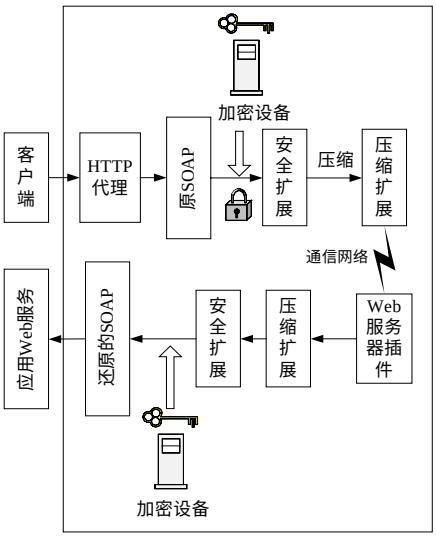

图 1 可信高性能的 Web Service 模型

服务器的应答 SOAP 也将类似处理，不再赘述。

3 平台无关和应用透明的实现

最后给出一个具体实现的方式。当今两大主流的平台 J2EE 和.NET 都对 Web Service 有很好的支持。微软的.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。本文中提到的两种扩展都可以用这种方式嵌入.NET 中。实际上，微软也提供了一个安全的解决方案：Web Service Enhancements，通过 SOAP 扩展增强 Web Service 的安全性。当然，仅仅是安全，没有达到可信的程度。

但是，考虑到通用性，实现要求做到平台无关。无论 J2EE 还是.NET 都可以以相同的方式实现，也不受 J2EE、.NET 互操作性的影响，因此不采用微软的解决方案。

在实现时，使用客户端 HTTP 代理加服务器端 Web Server 插件的技术来做到平台无关。客户端 HTTP 代理可以屏蔽浏览器、应用程序的差异，只要是 HTTP 协议，通过代理都能得到 Web Service 调用时的原始 SOAP 包，基于此 SOAP 包，可以加入两个扩展。服务器端，主流的 Web Server 一般都提供插件机制，允许在客户端的 HTTP 请求上送到应用程序前处理 HTTP 数据包。这样插件就能完成解压、解密、验证、还原 SOAP 的工作。

采用这种实现方式，对应用程序也做到了完全透明。应用开发调用 Web Service 的方式与原来完全相同，不需要做出任何额外的努力。已有的应用不需要修改，通过一些配置就能自动升级成可信高性能的 Web Service。可见，这是一种比较好的实现方案。

4 结论

本文提出了一种可信高性能的 Web Service 处理模型，通过将 PKI 技术，数据压缩技术与 Web Service 技术的结合，有效地解决了 Web Service 的安全性和性能的问题。并且给出了一种平台无关、应用透明的实现方式，有比较好的应用前景。

参考文献

1 Bartel M, Boyer J. XML-signature Syntax and Processing[Z]. W3C Proposed Recommendation 20, http://www.w3.org/TR/2001/PR-xmldsig-core-20010802, 2001-08.

2 XML Encryption Syntax and Processing. W3C Candidate Recommendation[Z]. http:/www.w3.org/TR/2002/CR-xmlenc- core -20020802, 2002-08-02.

3 Atkinson B, Libera G D. Web Services Security(WS-Security) Version 1.0[Z]. http://www.ibm.com/developeworks/library/ws-secure, 2002-04-05.

4 Box D, Ehnebuske D. Simple Object Access Protocol (SOAP) 1.1[Z]. W3C Note, http://www.w3.org/TR2000/NOTE-SOAP-20000508, 2000-05-08.

5 Gailey J H. Encrypting SOAP Messages Using Web Services Enhancements[Z]. http://msdn.microsoft.com/library/default.asp?url=library/enus/dnwse/html/wseencryption.asp, 2003-03.

（上接第 226 页）

5 结论

对于面向数字版权管理的搜索引擎技术作了一个详细的综述，基于拷贝检测的搜索引擎机制、内嵌 DRM 的搜索引擎机制以及外加 DRM 的搜索引擎机制作了详细介绍和比较。这项工作的深入研究将有助于互联网上的数字化图书馆、网络出版、远程教育以及互联网企业内容服务的版权内容查询。

参考文献

1 庄　超. 一种新型 Internet 内容版权保护的计算机制[D]. 北京: 中国科学院计算技术研究所, 1999-12.

2 Olin S. Securing the Content, Not the Wire, for the Information Commerce[Z]. http://www.intertrust.com.

3 Koch E. Copyright Protection for Multimedia Data[C]. Proc. of the International Conference on Digital Media and Electronic Publishing, 1994: 6-8.

4 Microsoft Palladium. Trusted Computing[Z]. http://www.microsoft.com.

5 Roscheisen D M. A Network-centric Design for Relationship-based Rights Management[D]. Stanford University, 1997-12.

6 Goldstein P. Copyright's Highway: The Law and Lore of Copyright from Gutenberg to Celestial Jukebox[M]. New York: Hill and Wang, 1994.

7 Stefik M. Letting Loose the Light: Igniting Commerce in Electronic Publishing, Draft[Z]. Xerox PARC, CA., 1995.

8 Stefik M. The Digital Property Rights Language, Manual and Tutorial[Z]. Xerox PARC: CA., 1996.

9 Choudhury, Maxemchuk. Copyright Protection for Electronic Publishing over Computer Networks[EB/OL]. ftp://ftp.research.att.com/dist/anonce/copyright.epub.ps.z.

10 Shivakumar N. SCAM: A Copy Detection Mechanism for Digital Documents[C]. Proc. of DL, 1995.