基于.NET的SOAP加密方法研究与实现

孟 军 盛 雨 刘洪波

(大连理工大学计算机科学与工程系 大连 116023)

摘 要 本文首先从数据加密性能方面讲述可以利用公钥体制部分加密SOAP消息的方法,在此基础上提出一种利用Web Services架构高数据同步性系统的解决方案,并在性能和可行性上对方案进行了比较详细的分析。实验证明该方案是可行的。

关键词 数据加密,SOAP,Web Services

Research and Realization for SOAP Encryption Techniques Based on .NET

MENG Jun SHENG Yu LIU Hong-Bo

(Department of Computer Science & Engineering, Dalian Unv&ersity of Technology, Dalian 116023)

Abstract This paper discusses the method of encrypting SOAP messages partially using public key at first. Then using Web services architecture, a high data synchronization solution is presented. The performance and feasibility about this method are analyzed detailedly. The experimental result shows that this method is feasible.

Keywords Data encryption, SOAP, Web services

1 引言

Web Service向外界显示的是一个能够通过Web进行调用的函数。目前,在企业级应用中Web Service扮演着非常重要的角色,它为客户端提供实现某种功能的接口,并以SOAP消息的格式将运算结果返回给调用者[1],但是SOAP消息传播的途径是Internet,所以安全问题就显得非常重要,特别是一些重要的部门或企业(如银行、公安系统等)传播的数据是非常敏感和重要的,这些数据如果被怀有某些企图的人获得,将会造成严重的后果[2],所以必须对SOAP消息进行加密。

目前,加密SOAP消息常用的方法有SSL和WS_SECURITY这两种,其中SSL只能加密整个SOAP消息,这会影响效率(因为几乎所有的加密算法都会大大降低应用程序的效率);而WS_SECURITY是Microsoft联合IBM和其他几家公司提出的安全协议,它虽然可以选择性地加密SOAP消息中的部分信息,但是,实现起来比较繁琐。本文介绍一种能够自如而有效地加密SOAP消息中敏感数据的方法。

2 SOAP的消息组成及加密过程

2.1 SOAP的消息组成机构

SOAP消息是由XML格式定义的,它分为三个部分:Envelope、Header和Body,它们之间的关系如图1所示。

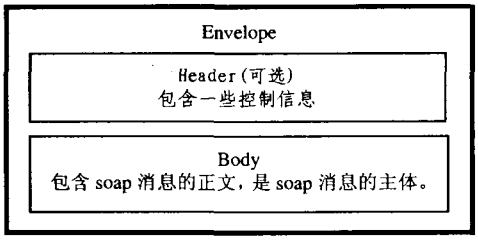

图1 soap消息的组成结构

Envelope包括Header和Body两个组成元素,其中Header是可选的,只有当需要对SOAP消息添加一些控制信息的时候,才需要Header;而Body存放消息的正文。这里介绍的方法的基本思路就是:调用者将请求信息连同公钥(放在Header中)通过SOAP消息传给提供者,提供者有选择地用公钥加密他认为需要加密的数据,然后将信息通过SOAP消息返回给调用者,最后调用者用自己的私钥解密数据,加密过程如图2所示。

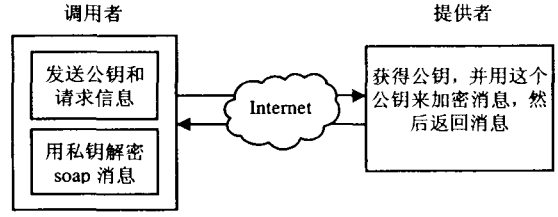

图2 加密过程示图

本文加密用到的算法是RSA算法,运用两个不同的密钥,即公钥和私钥。私钥用来生成一个可以发送到其它应用程序的公钥。通过运用公钥,应用程序可以加密数据,并把它们发送到私钥的所有者那里,数据可以在此被解密和使用。只有运用私钥才可以解密数据,所以如果没有私钥,数据就没有用了。

这种方法有两个难点:第一,怎样将公钥传送给服务提供者;第二,服务提供者怎样取得公钥,并用这个公钥加密数据。

为了解决第一个问题,由于soap消息中的header元素是专门用来存放控制信息的,因此可以将公钥放在header里传给提供者,但这需要自定义soap header类。在.NET中这样自定义header类的好处就是,它会让.NET框架自动解析soap的header元素而不必手动添加代码来完成解析任务[3]。可以在提供者处定义header类:

```
public class header:SoapHeader
{ private string_PublicKey;//这是一个属性
  public string PublicKey//这是一个字段
  { get{return _PublicKey;}
    set{//提供者将解析得到的公钥值赋给 header 的属性_Pub-
```

```
        licKey
    if (value ! = null &&  value ! = String. Empty)
    { _PublicKey = value;}
      else
    { throw new SoapException(
        "No PublicKey"+"Has Been Received!",null);}
      }
  }
}
```

这段代码实现了在提供者处定义 soap 消息 header，使其能够接收调用者传来的公钥，从而进行加密。

对于第二个问题，解决方法比较繁琐，首先需要弄清楚两个概念：soap 扩展类 和 soap 扩展属性类，这两个类一般是成对出现的。提供者用 soap 扩展属性类来收集调用者传来的 soap 消息中 header 部分的信息，然后通过相应的 soap 扩展类来用由 soap 扩展属性类收集到的信息进行操作，下面是一个具体的应用。

3 SOAP 消息加密方法研究

3.1 加密算法研究

首先，定义 soap 扩展属性类，在这个类中定义一个属性_AttributeNamesToEncrypt，这个属性就是用来制定那部分需要加密的，它只是一个字符串数组，用来存放 XML 元素的名称。相应地，在 SOAP 扩展类中也要定义一个字符串数组，用来存放从 SOAP 扩展属性类中的_AttributeNamesToEncrypt 拷贝过来的数据，然后，SOAP 扩展类就会根据这个数组中的名称来和将要返回的 XML 数据中的元素一一比较，只要匹配，就加密这个元素中存放的数据。下面是一个例子（C# 代码）：

```
[WebMethod]
//header 是前面定义的 SOAP 消息头的对象名
[SoapHeader ( " header",  Direction = SoapHeaderDirection. In, Re-
    quired=true)]
//将需要加密的数据的名称存放在 SOAP 扩展属性类中的属性里
//这里我需要加密的数据名称是"NAME"
[EncryptionExtension(AttributeNamesToEncrypt  = new string[]{"
    NAME"})]
//web service 提供的方法，返回一个数据集
public DataSet data()
{
  ......
  return dataSet11;
}
```

数据集对应的数据库中的表如表 1 所示：

表 1 STUDENT1 的结构

NAME	AGE	SEM(male:1,femal:0)
sy	25	1
lqy	24	0

如果没有进行加密，WEB SERVICE 以 XML 格式返回的 SOAP 消息（简化形式）如下面所示：

```
〈? xml version="1. 0" encoding="utf-8" ?〉
〈soap:Envelope〉
〈soap:Body〉
〈dataResult〉
〈DataSet1 xmlns="http://www. tempuri. org/DataSet1. xsd"〉
〈STUDENT diffgr:id="STUDENT1" msdata:rowOrder="0"〉
〈NAME〉sy〈/NAME〉
〈SEX〉nan〈/SEX〉
〈AGE〉24〈/AGE〉
〈/STUDENT〉
〈/DataSet1〉
〈/dataResult〉
〈/soap:Body〉
〈/soap:Envelope〉
```

可以看到〈NAME〉sy〈/NAME〉这个元素是明文的形式，而在经过加密以后，上面的 SOAP 消息就变成下面所示的形式：

```
〈? xml version="1. 0" encoding="utf-8" ?〉
〈soap:Envelope〉
〈soap:Body〉
〈dataResult〉
〈DataSet1 xmlns="http://www. tempuri. org/DataSet1. xsd"〉
〈STUDENT diffgr:id="STUDENT1" msdata:rowOrder="0"〉
〈                    NAME                    〉
YxwTxkiIciYgsE1rW4PFiqto33yiOVjxmj176n9gCrlkxzrKgZ9nGw-
d3Yb8q9mNndPn042q4GDs1bf5rXHh0UAggxKU33rtR3iQsrYi0DcuDLXZZ-
5GZF7NCTxzIusWyd5kTKuWyftOLflvYrxX
KnnZCTxLi2hyxHpmscGFQUt5M=〈/NAME〉
〈SEX〉nan〈/SEX〉
〈AGE〉24〈/AGE〉
〈/STUDENT〉
〈/DataSet1〉
〈/dataResult〉
〈/soap:Body〉
〈/soap:Envelope〉
```

可见，上述代码中〈NAME〉〈/NAME〉元素已经被加密成密文了。当然我们也可以加密〈SEX〉1〈/SEX〉或者〈AGE〉24〈/AGE〉。

选择性加密 SOAP 消息的方法，不仅达到了加密数据的目的，也大大提高应用程序的效率。此外，这种方法还可以实现一个 SOAP 消息一次发送给多个调用者而不影响安全性要求，图 3 显示了这个过程。

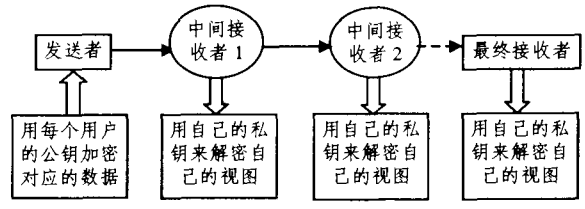

图 3 一信多发的过程示意图

如图 3 所示，当需要将一个消息同时发送给几个接收者，而且只能发送一次的时候，就要为每个接收者产生它们相应的视图（比如，接收者 1 只能看到字段 NAME，接收者 2 只能看到字段 NAME 和 AGE）。这样，就实现了在保证安全的前提下的"一发多送"的目的。

3.2 SOAP 消息加密实现

首先要在消息发送者那里设置 Web Services 传输路径，在路径的设置上可以采用 WS-Security。它是对 SOAP 协议在传输路径方面进行了扩展的一种协议[4]。本文主要讲述 Web Services 的加密问题，所以路径设置就不在这里具体描述了。

上述系统的消息加密要分成两种情况来考虑：

第一种情况：既要考虑系统外部的安全性，又要考虑系统内部的安全性。每个接收者的视图可能存在重复的部分，这时就要将重复数据拷贝多份，然后对每个接收者对应的数据用他们各自的公钥进行加密，在整个传输过程中，当 SOAP 消息经过每个接收节点时，接收者用它们各自的私钥对消息进行解密，这样就实现了一次传输多个接收者，并且保证了系统内部的安全性。拷贝视图交集情况如图 4 所示。

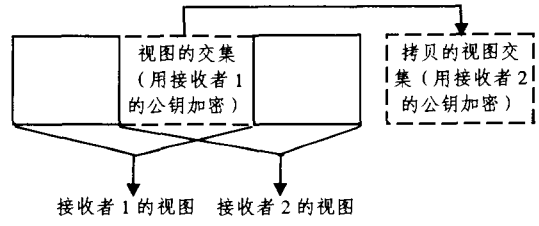

图 4 拷贝视图交集示意图

但这是在牺牲数据量和加密性能的前提下实现的，如果要传输的数据量很大，并且各个接收者要获取的数据之间有很大的重复度，上述方法将会影响消息传输性能[5]。

在大多数情况下，系统外部的安全性要比系统内部的安全性重要得多，因为系统内部的各个节点都应该是可信的，要牺牲性能来保证不必要的安全性就不合适了，这时就只需要考虑系统外部的安全性，这就是第二种情况。这时就应该改变加密的策略，对整个信息进行加密，发送者用它发布的公钥加密数据，并且保证每个接收者都有发送者的私钥，当消息到达某个接收者时，它就用文章刚开始讲的方法只对属于自己的视图解密，在获得数据后，在将消息发送给下一个接收者。整体加密情况如图 5 所示。

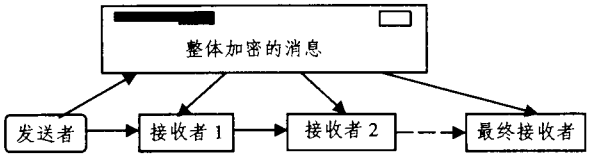

图 5　整体加密示意图

此外在路径的设置上还分为静态和动态两种，当需要在传输的过程中动态改变传输路径时，要考虑防止出现死循环的情况，一旦出现死循环，一部分节点就会永不停止地中转、加密和解密数据，造成传输的失败[6]，不过这不是本文考虑的问题，在此不再讲述。

结论　通过在局域网上进行的实验表明，该加密方法是可行的，在传输的数据量不是特别大的情况下，其性能也是合理的。本文的下一步研究工作将着重研究大数据量情况下的数据冗余和加密性能的问题。

参 考 文 献

1　柴晓路. 为什么需要 Web 服务. IBM developerWorks 专刊，2003
2　King S. CLSSP, threat and Solutions to Web Services Security. Network Security, 2003, 9: 8～11
3　Wahlin D. XML for ASP. NET Developers. SAMS, 2001
4　柴晓路. SOAP Header 扩展：WS-Routing 和 WS-Referral，IBM Corporation, 2001. http://www-900. ibm. com/developerWorks/cn/webservices/ws-soapheadext/part2/index. shtml
5　张勇，冯东雷，陈涵生，白英彩. Internet 密钥交换协议的安全缺陷分析. 软件学报，2002, 13(6)
6　刘怡，李伟琴. 密码协议的分层安全需求及验证. 北京航空航天学学报，2002, 28(5): 589～592

（上接第 39 页）

表 2　基于包个数调整系数后的测试模拟

d_i		32					64				
		Detections	Alarms	True	False	Times	Detections	Alarms	True	False	Times
P=22	1/2,1/2	31	1063	52	1011	348	24	659	33	626	185
	1/3,2/3	30	1163	53	1110	348	24	750	34	716	185
	2/3,1/3	31	1063	52	1011	348	24	803	35	768	186

表 3　调整系数前后(基于包个数)的检测结果比较

阈值因子 $P=22$	系数	detections	Alarms	True	False	False Rate
未使用系数(等价于系数相等，即都为 1/7)	1/7, 1/7, 1/7, 1/7, 1/7, 1/7, 1/7,	55	878	137	741	0.843
调整系数后	8/21, 8/21, 1/21, 1/21, 1/21, 1/21, 1/21	68	802	153	649	0.809

表 3 给出了基于时间序列的模拟检测结果，由表 3 可知，采用时间序列的检测结果要好于基于包的个数，主要原因是基于包的个数的测度选择方法中测度个数太少，不能准确地描述异常。对于基于时间序列的测度方法，测度的个数多，更能准确地反应异常。第二种方法使用了修改过的系数(8/21, 8/21, 1/21, 1/21, 1/21, 1/21, 1/21)，误警率也有所降低，由 0.843 变为 0.809，检测率有所增加，检测到的入侵数由 55 变为 68，可见使用系数调整后系统的检测效果有了显著的改善。

总结　本文提出了一种新的流量入侵检测的模型 $T^2=\sum_1^2 d_i \frac{(K_i-\overline{K}_i)^2}{\overline{K}_i}$，并且着重分析了基于时间序列的实现以及测度的选取方法，并在对数据深入分析的基础上对测度进行了优化，如合并了三种协议的平均时间间隔测度。文中给出了 d_i 的选取方法，有较强的可操作性，但是仍然需要对网络流量有较深入的了解才能真正地确定出一组适合系统的系数 d_i。另外由于测度本身的选取还有待改进，使得总的检测效果还是不够理想。以后的工作将从两个方面展开，第一是对每天的网络流量进行更细致的建模，将不同的时间段采用不同的概率分布模型，第二是对测度的选取进一步研究，优选出更加有效的测度以提高检测率。

参 考 文 献

1　Nassehi M. Anomaly detection for Markov models: [Research report]. IBM Research Division, Zurich Research Laboratory, 8803 Ruschlikon, Switzerland, 1998
2　Deri L, Suin S, Maselli G. Design and Implementation of an anomaly detection System: An empirical approach. http://jake. unipi. it/～deri/ADS. pdf. Aug. 2001
3　Huang P, Feldmann A, Willinger W. A nonintrusive, wavelet-based approach to detecting network performance problems. ACM SIGCOMM internet measurement workshop 2001, San Francisco, USA, Nov. 2001
4　Alarcon-Aquino V, Barria J A. Anomaly detection in communication networks using wavelets. IEE Proc. -Commun. , 2001, 148(6)
5　邹伯贤，李忠诚. 基于 AR 模型的网络异常检测. 微电子学与计算机，2002(12)
6　Denning D E, Neumann P G. Requirements and Model for IDES - A Real-Time Intrusion Detection System. Computer Science Laboratory, SRI International, 1985
7　Ye Nong, Chen Qiang. An Anomaly Detection Technique Based on a Chi-Square Statistic For Detecting Intrusions Into Information Systems. Quality and Reliability Engineering International
8　Montgomery DC. Introduction to Statistical Quality Control. John Wiley & Sons: New York, 2000
9　Banks J. Principles of Quality Control. John Wiley & Sons: New York, 1989
10　MIT Lincoln Laboratory. A public Web site. http://www. ll. mit. edu/IST/ideval/index. html