基于JAVA的XML加解密系统的设计与实现

李海华
(中国人民解放军信息工程大学 国家数字交换系统工程与技术中心,河南 郑州 450002)

摘 要:网络银行、网上交易在实际中已被广泛应用,信息的安全成为网络应用可信度的重要因素。Web中越来越多的数据以XML(The eXtensible Markup Language)格式出现,XML的加密技术引起了广泛的关注。基于JAVA安全应用程序接口,设计了XML的加密系统和解密模型,提出了XML加密系统的实施策略,给出了整个XML文档、XML文档中元素、XML文档中元素内容的加密方法,实现了加解密模块的功能,使该系统对XML文档的加密可嵌入到XML文档内部,把加密粒度细化到XML文档元素级别,保证了信息收发的安全性。

关键词:XML;元素;加密;解密
中图分类号:TP309 **文献标识码**:A **文章编号**:1673-629X(2011)08-0246-04

Design and Implementation of XML Encryption System Based on JAVA

LI Hai-hua
(National Digital Switching System Engineering & Technological Center, PLA Information Engineering University, Zhengzhou 450002, China)

Abstract: Internet banking, online transaction has been widely used in practice, information security has become an important factor in the credibility of network applications. As more and more data appeared by the form of XML(The eXtensible Markup Language) in the web, the XML encryption technology has brought an extensive attention. An encryption system and decryption model is designed to XML based on JAVA, give methods of encryption for the XML document, XML document elements, and XML element content, realize the function of encryption and decryption modules, so this system can put the XML documents encryption into the XML files and tessellate the encryption compactness up into the XML file's element, to ensure the security of information sent and received.

Key words: XML; element; encryption; decryption

1 概 述

JDK1.2中的加密体系结构如图1所示,Java中定义了一组服务提供者接口(SPI),在此基础上形成了完整的平台独立的加密API,它主要包括加密服务、认证和密钥管理三个方面[1]。

1.1 加密服务

Java的加密主要由以下几个类提供:

(1)Cipher,这个类提供加密、解密的功能,是JCE的核心类。

(2)CipherInputStream,用于加密和解密的输入流,通过指定与该流关联的Cipher决定该流是用于加密或解密。

(3)CipherOutputStream,这个类是加密和解密的输出流[2]。

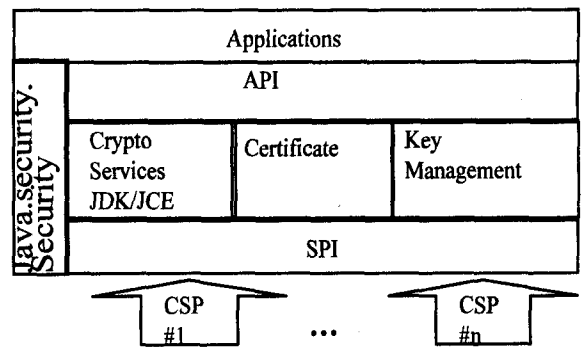

图1 Java加密的体系结构

1.2 证书管理

证书包括公钥、实体标识、数字签名三部分。JDK1.2中与证书相关的类和接口有:

(1)Certificate,这个类是各种格式证书公共接口的抽象。每种具体的证书是Certificate的子类;

(2)CertificateFactory,用于产生证书对象和证书取消列表对象;

收稿日期:2011-01-07;**修回日期**:2011-04-22
基金项目:国家973计划课题(2007CB307100)
作者简介:李海华(1965-),男,副教授,河南省学术技术带头人,河南省教学标兵,主要研究方向为网络协议、网络安全。

(3)X509Certificate,定义了 X.509 证书的抽象接口[3]。

1.3 密钥管理

密钥管理主要负责管理两种入口:

a. 密钥入口,存放用于加密的密钥;

b. 可信认证书入口,存放另一实体的公钥证书。

Java 中用于密钥管理的类主要有:

(1)Key,各种加密算法密钥的抽象接口;

(2)KeyStore,它提供一组接口用于访问和修改密钥库中的信息;

(3)一个缺省的 KeyStore 类,基于文件的密钥库管理类,其属性名叫做"JKS"。它用密码保护加密的私钥;

(4)KeyPairGenerator,生成公开与私有密钥对;

(5)KeyFactory,在安全密钥与密钥规范之间转换;

(6)Key 描述接口,用于描述不同算法的密钥参数,主要用于保护 Key 接口对上层的透明性[4,5]。

2 加解密程序的设计和实现

2.1 加解密程序的设计

在加解密程序中创建了 XmlEncApp 类、XmlDecryption 类和 XmlEncryption 类,其中 XmlEncryption 类调用四个主要的方法:EncryptCompleteXmlFile( )方法、EncryptElementOfXmlFile ( ) 方法、EncryptElementContentOfXmlFile( )方法和 generateKey( )方法。

部分编程接口如下:

```
public class XmlEncryption {
//原文件名和生成后的文件名
private String fileSource = null;
private String fileResult = null;
//用于进行加密的算法名称
private String algoName = null;
//用密钥标识符保存已协商好的秘密密钥
private String keyName = null;
//主结构的 Id 属性
private String encId = null;
//文档对象构造器
private DocumentBuilder docBuilder = null;
//默认构造器
public XmlEncryption( ) {
//创建一个 DocumentBuilder 对象
try {
docBuilder = DocumentBuilderFactory. newInstance
( ). newDocumentBuilder( );
} catch ( ParserConfigurationException e ) { doc-
Builder = null; }
}
//生成一个新的文档对象
private Document getNewDocument( ) {
if ( docBuilder ! = null)
return docBuilder. newDocument( );
else
return null;
}
//对整个 XML 文档加密
public void encryptCompleteXmlFile( )
//对 XML 文档元素加密
public void encryptElementOfXmlFile ( String ele-
mentName)
//对 XML 文档中元素内容加密
public void encryptElementContentOfXmlFile ( String
elementName)
}
```

2.2 加解密程序的实现

XmlEncApp 类中函数调用 XmlEncryption 对象的一些方法来设置以下的属性:

(1)clearDoc:XML 文档的 DOM 对象被加密;

(2)encKey:用于加密的密钥;

(3)algoName:加密算法的名字;

(4)keyName:加密密钥的名字;

(5)encId:在文档中给 EncryptedData 标签唯一的名字。

2.2.1 整个 XML 文档的加密

encryptCompleteXmlFile( )方法实现对整个文档的加密功能,它依次调用以下方法实现对整个文档的加密[6]:

· getEncryptedDataDoc ( ):返回 EncryptedData 类生成的对象;

· getEncryptionMethodDoc ( ):该方法是使用 EncryptionMethod 类来定义 XML;

· getKeyInfoDoc( ):它使用 GenericKeyInfo 类来定义 XML;

· getString( ):对整个 XML 文档进行序列化,并存为字符格式;

· getEncryptedData( ):将明文加密成密文的过程,返回一个 Base64 编码的加密文本串;

· getCipherDataDoc( ):返回对应于 Object 元素的 XML 结构;

· addChild( ):多次调用 addChild( )方法,使加密后的 XML 文档仍然保持其良好的格式。

2.2.2 XML 文档中元素的加密

encryptElementOfXmlFile( )方法依次调用以下方

法实现对元素加密[7]：

· getEncryptedDataDoc()：返回一个对应于 Encryption 元素的 XML 结构；

· getEncryptionMethodDoc()：返回一个对应于 EncryptionMethod 元素的 XML 结构；

· getKeyInfoDoc()：返回一个对应于 KeyInfo 元素的 XML 结构；

· getClearNode(elementName).toString().trim()：获得将要加密的一个或多个元素结点(其中可以包含自己的子元素)，并将其转换成字符格式；

· getEncryptedData()：将明文加密成密文的过程，返回一个 Base64 编码的加密文本串；

· getCipherDataDoc()：返回一个对应于 Object 元素的 XML 结构；

· addChild()：多次调用 addChild()方法，使加密后的 XML 文档仍然保持其良好的格式。

2.2.3 XML 文档中元素内容的加密

encryptElementContentOfXmlFile()方法实现对 XML 文档中元素内容的加密功能，它依次调用以下方法实现对元素的加密[8]：

· getEncryptedDataDoc()：返回 EncryptedData 类生成的对象，它包含一个对应于 Encryption 元素的 XML 结构；

· getEncryptionMethodDoc()：返回一个对应于 EncryptionMethod 元素的 XML 结构；

· getKeyInfoDoc()：返回一个对应于 KeyInfo 元素的 XML 结构；

· getElementContent(elementName).trim()：获得将要加密的一个或多个元素的内容，并将其转换成字符格式；

· getEncryptedData()：将明文加密成密文的过程，返回一个 Base64 编码的加密文本串；

· getCipherDataDoc()：返回一个对应于 Object 元素的 XML 结构；

· addChild()：多次调用 addChild()方法，使加密后的 XML 文档仍然保持其良好的格式。

2.2.4 getEncryptedData()方法

encryptElementOfXmlFile()方法、encryptElementContentOfXmlFile()方法和 encryptCompleteXmlFile()方法都调用了 getEncryptedData()方法，它得到了 JCA/JCE 中相应类的支持，其工作过程如图 2 所示。

getEncryptedData()方法创建加密对象，初始化矢量 IV 并生成有矢量前缀的文本字符串，最后调用 JCE 中的类加密生成 Base64 编码的密文[9]。

2.2.5 XML 文档的解密

decryptedData()方法依次调用以下方法实现对加密元素的解密[10,11]：

· getElementsByTagName()：该方法从已加密的 XML 文档中得到需要解密的内容，并根据属性值判断属于哪一种加密类型；

· getNamedItem().getNodeValue()：该方法可得到所用的加密算法；

· Decrypt().trim()：对加密的内容进行解密；

· getParentNode().replaceChild()：将解密后的内容还原到文档相应的部分，使其成为格式良好的 XML 文档。

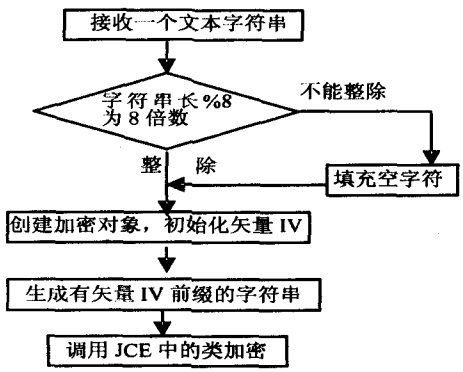

图 2 getEncryptedData()方法的工作过程

2.2.6 嵌套加密

根据加密需求，加密后的 XML 文档仍是格式良好的，当嵌套加密时，仍用加解密功能模块进行，只是入口参数不同而已。但要注意在嵌套加密时，不能对元素<EncryptedData>的子元素加密[12]。

3 实例分析

一个在线书销售商与书出版商以 XML 文档的形式交换信息的示例，书销售商制定一个安全策略，在订单以 XML 文档方式传到书出版商时，书出版商的销售部门只能获取订单的书名、书号和订书的数量，书出版商的会计部门才能看到支付信息。书销售商订单信息如图 3 所示。

```
<? xml version="1.0"? >
<! --书销售商方提供的 XML 文档订单  -->
<书销售商订单>
<订单>
<书名>计算机技术与发展<书名>
<书号> CN61-1450/TP </书号>
<数量>70</数量>
</订单>
<支付信息>
<卡号>60987997</卡号>
<持卡人>常青</持卡人>
<支付日期>2010/03/12</支付日期>
</支付信息>
</书销售商订单>
```

图 3 书销售商的订单信息

书销售商 XML 订单在加密前显示完整信息,书名:计算机技术与发展,书号:CN61-1450/TP,数量:70,卡号:60987997,持卡人:常青,支付日期:2010-03-12。

传送到出版商的 XML 订单已加密,订单信息见图 4。对每行的解释如下:

[01~23] 这是一个书销售商发送给书出版商的对信用卡信息加密后的 XML 文档订单;[04~08] 这是有关订单的详细信息,是非敏感信息,以供书出版商的销售部门查看;[09~22] 其中包含的是信用卡加密后的信息,仅供会计部门解密查看;[11~18] 提供了相关的加密信息,如密钥标识符、加密算法,以便会计部门解密;[19~21] 加密的密文信息。

```
[01]<?xml version="1.0"?>
[02]<!--书销售商方提供加密后的 XML 文档订单
-->
[03] <书销售商订单>
[04] <订单>
[05] <书名>计算机技术与发展<书名>
[06] <书号>CN61-1450/TP</书号>
[07] <数量>70</数量>
[08] </订单>
[09] <EncryptedData
Type='http://www.w3.org/2001/04/xmlenc#Elem
ent'
[10]
xmlns='http://www.w3.org/2001/04/xmlenc#'>
[11] <EncryptionInfos>
[13] <KeyInfo>
[14] <KeyName>1234</ KeyName >
[15] </KeyInfo>
[16] <EncryptionMethod
[17]
Algorithm='http://www.w3.org/2001/04/xmlenc
#tripledes-cbc '/>
[18] </EncryptionInfos>
[19]                                    <CipherData>
euriekeddedleiffmfmroafjkdvmrektklkprjgfgks
fkflfkg
lfksgfdddjhhkfffflgtbmtglkgkgklfkgmfienjeahd
stewrootmjhhkjh
[20]
jhhkfffflgtbmtglkgkgklfkgmfienjeahdstewrootm
fxbvbkfkrjimghjg
[21] uhghfgf</ CipherData >      <!—加密的支
付信息-->
[22] </ EncryptedData >
[23] </书销售商订单>
```

图 4　订单信息

对已加密的 XML 文档进行修剪处理后,在书出版商的销售部门所能看的订单信息中支付信息(持卡人:常青,支付日期:2010-03-12)被隐藏,重要信息得到了保护。

4　结束语

文中提出了一个 XML 加密系统,该系统把加密的粒度细化到 XML 文档的元素级,实现对同一文档的不同部分实施不同类型的加密,对不同的用户呈现不同的视图,用户只能看到被授权的那部分内容,这一功能在电子商务、企业和政府管理等领域很有价值,而其它加密系统无法实现这一功能。

参考文献:

[1] 张　艳,周明天,佘　堃. 基于 Web services 的 XML 引擎安全模型研究[J]. 计算机应用研究,2008,25(7):61-63.

[2] Goldfarb C F,Prescod P. XML 用户手册[M]. 潇湘工作室译. 北京:人民邮电出版社,2000.

[3] 曲巨宝. XML 网络服务安全策略应用[J]. 计算机技术与发展,2007,17(12):151-153.

[4] Imamura T,Maruyama H. Specification of Element-wise XML Encryption[EB/OL]. 2000. http://lists. w3. org/Archives/Public/xml-encryption/2000Aug/att-0005/01-xmlenc-spec. html.

[5] 况　旭,刘　波. XML 的面向对象语言特性[J]. 计算机技术与发展,2010,20(1):54-57.

[6] W3C. XML-Signature Syntax and Processing [EB/OL]. 2002. http://www. w3. org/TR/2002/REC-xmldsig-core-20020212/.

[7] IBM. XML Security Suite [EB/OL]. 2000. http://www. alphaworks. ibm. com/tech/xmlsecuritysuite/.

[8] Schneier. 应用密码学:协议、算法、与 C 源程序[M]. 吴世忠译. 北京:机械工业出版社,2000.

[9] Zhang Jimmy. XML on a Chip[EB/OL]. 2005. http://www. xml. com/pub/a/2005/03/09/chip. html.

[10] 张洪伟. JSP 网络开发技术与整合应用[M]. 北京:清华大学出版社,2007.

[11] 曾春平,张　鹏,王　超. InsideXML XML 编程从入门到精通[M]. 北京:北京希望电子出版社,2010.

[12] Geer D. Will Binary XML Speed Network Traffic[J]. IEEE Computer Society,2005,38(4):16-18.

(上接第 216 页)

南大学,2008.

[8] 原　玲. 第三代移动通信系统网络规划技术[J]. 计算机应用,2006,26(S1):181-185.

[9] 田　畅,王　海,郑少仁. 基于用户行为的网络流量模型及自相似性分析[J]. 通信学报,2000,21(9):19-25.

[10] 王西峰,高　岭,张晓孪. 自相似网络流量预测的分析和研究[J]. 计算机技术与发展,2007,17(11):42-45.

[11] 谢高岗,闵应骅,张大方,等. 一个基于实际测试的网络流量模型[J]. 计算机工程与科学,2001,23(5):51-53.

[12] 李　捷,刘瑞新,刘先省,等. 一种基于混合模型的实时网络流量预测算法[J]. 计算机研究与发展,2006,43(5):806-812.

[13] 尚凤军,潘英俊,唐　红. 一种基于回归方程的流量矩阵研究[J]. 计算机工程与应用,2005(9):9-12.