基于CAPICOM和IAIK的信息安全传输系统

吴洁明 史建宜 李硕征

(北方工业大学信息工程学院 北京 100144)

摘 要 介绍了几种常用的网络数据安全传输的技术,在此基础之上,提出了信息安全传输系统的设计思想。在此系统中,通过使用开源的EJBCA系统完成对数字证书的管理。客户端采用CAPICOM技术简化数字签名和数字信封的实现过程。服务端使用第三方库IAIK实现对PKCS#7格式数据的解析和验证。最后,给出了一些实现信息安全传输系统的关键代码。

关键词 CAPICOM,数字签名,数字信封,PKI,EJBCA,IAIK,PKCS#7

中图法分类号 TP393.08 **文献标识码** A

Information Secure Transmission System Based on CAPICOM and IAIK

WU Jie-ming SHI Jian-yi LI Shou-zheng

(Information Engineering Institute, North China University of Technology, Beijing 100144, China)

Abstract This paper describes some techniques which can ensure the secure transmission of network data. On this basis, proposes the design ideas of the information secure transmission system. In this system, completes the management of digital certificates by using the EJBCA open source system. And, the client using CAPICOM technology simplifies the digital signatures and digital envelope implementation process. Besides, the server uses third-party libraries IAIK to achieve analysis and validation PKCS#7 format data. Finally, provides some critical code of the information secure transmission system.

Keywords CAPICOM, Digital signature, Digital envelope, PKI, EJBCA, IAIK, PKCS#7

随着互联网技术的飞速发展,计算机网络技术广泛应用到社会的各个领域,通过网络进行信息的传输变得越来越普遍,它给我们的生活带来了极大的方便。与此同时,也出现了很多信息传输方面的安全隐患,比如:信息截获和窃取、信息篡改、信息伪造、信息抵赖等。如何保障信息传输的机密性、完整性、真实性、不可抵赖性成为信息安全传输所面临的最大挑战。针对以上问题,应运而生了一系列保障网络数据安全性的技术,其中包括:数字证书、加密技术、数字签名、数字信封、公钥基础设施(Public Key Infrastructure, PKI)等等。

1 信息安全技术

1.1 数字签名

数字签名是指发送方使用加密算法对原文进行加密,生成一段信息,再把此加密信息和原文一起发送给接收方,接收方对其进行解析,并判断原文在发送过程中是否被篡改[1]。数字签名算法包括签名和验证两项操作,遵循"私钥签名,公钥验证"的签名/验证方式。数字签名的过程如下:

(1)发送方使用Hash函数对原文生成信息摘要。

(2)发送方使用自己证书的私钥对信息摘要进行签名,形成数字签名。

(3)发送方把原文和数字签名一同发给接收方。

(4)接收方用发送方证书的公钥解析数字签名,得到信息摘要;接收方使用与发送方一样的Hash函数,对原文进行信息摘要,此信息摘要与解析数字签名得到的消息摘要信息对比,从而验证数字签名。

数字签名中用到了单向Hash函数,它有一个十分显著的特点是输入数据的丝毫变化都会引起信息摘要极大的变化,从而保证了数据的完整性。数字签名采用"私钥签名,公钥验证"的模式,可以保证数据的完整性、真实性和不可抵赖性。

1.2 数字信封

数字信封中采用了单钥密码体制和公钥密码体制。信息发送者首先利用随机产生的对称密码加密信息,再利用接收方的公钥加密对称密码,被公钥加密后的对称密码称为数字信封[1]。信息接收方解密信息时,必须先用自己的私钥解密数字信封,得到对称密码,才能利用对称密码解密所得到的信息。这样就保证了数据传输的真实性和完整性。数字信封的过程如下:

(1)当发送方需要发送信息时,首先随机生成一个对称密钥,这个对称密钥是用来加密待发送信息的,生成密文。

(2)发送方用接收方证书的公钥加密对称密钥,形成数字信封。

本文受国家科技部支撑计划课题:第三方版权服务模式与标准研究基金(2012BAH04F01)资助。

吴洁明(1958-),女,硕士,教授,主要研究方向为软件工程,E-mail:wujieming@263.com;史建宜(1988-),女,硕士生,主要研究方向为软件工程、信息安全;李硕征(1985-),男,硕士生,主要研究方向为软件工程、信息安全。

(3)发送方将密文和数字信封发送给接收方。

(4)接收方使用自己证书的私钥解密数字信封,得到对称密钥。

(5)接收方用得到的对称密钥解密密文,得到发送方发送的信息。

数字信封采用了对称密钥加密算法和非对称密钥加密算法相结合的方式,能够保证信息传输的机密性[2]。

1.3 PKI

PKI是Public Key Infrastructure的缩写,即公开密钥基础设施,它是国际上解决开放式互联网络信息安全要求的一套体系。PKI体系支持身份认证,信息传输、存储的完整性,消息传输、存储的机密性,以及操作的不抵赖性。PKI的主要服务组件包括:认证机构(CA)、注册机构(RA)、证书服务器(LDAP)、证书库、证书验证机构,其中CA是PKI系统的核心。

2 信息安全传输系统的整体设计

信息安全传输系统采用的是B/S架构,客户端(浏览器端)的开发语言是JSP,服务端的开发语言是Java。系统的主要业务流程是:用户把待上报数据经过客户端安全操作,得到密文;把密文经过网络传输到服务端;密文经过服务端操作进行解析和验证,验证通过得到待上报数据,将其存储到业务系统的数据库中;在此过程中所需要的使用的证书,由EJBCA统一管理。信息安全传输系统的主要流程如图1所示。

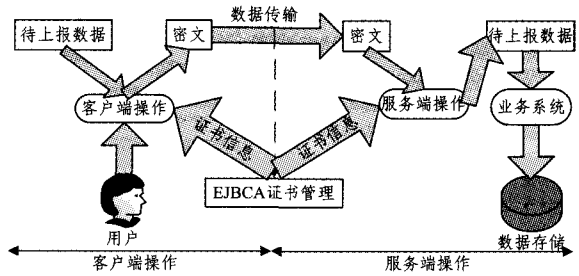

图1 信息安全传输系统的主要流程图

现在,详细介绍客户端(浏览器端)和服务端的详细功能,如图2所示。客户端的主要操作:1)用户填写待上报数据,经过信息摘要和数字签名,形成数字签名;2)随机产生一个对称密钥,用服务端公钥对此对称密钥进行加密,形成数字信封;3)把待上报数据、数字签名、客户端公钥用对称密钥进行加密,形成加密信息;4)把加密信息、数字信封经过网络发送到服务端。

服务端的主要操作:1)用服务端私钥解析数字信封,得到对称密钥;2)用对称密钥解密加密信息,得到待上报数据、数字签名和客户端公钥;3)用客户端公钥验证数字签名,得到信息摘要。4)用与客户端同样的Hash函数对待上传数据进行信息摘要,得到重新计算的信息摘要,此信息摘要与3)中的信息摘要进行对比。从而验证上报数据的机密性、完整性、真实性和不可抵赖性。

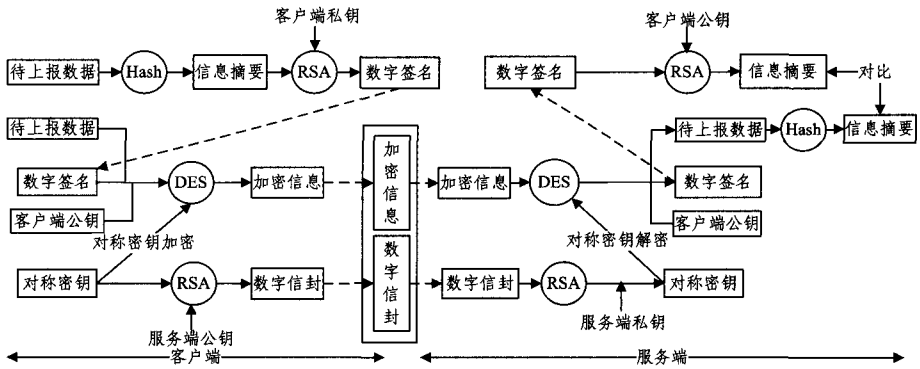

图2 信息安全传输系统的详细功能图

3 CAPICOM实现加密

3.1 概述

Window操作系统采用分层的加密体系模型密码服务提供者(Cryptographic Service Provider,CSP)架构系统安全。CSP提供整套的具有加密功能的CryptoAPI,供开发者使用。CryptAPI使用过程相当的复杂,也不适合在Web上直接使用。所以,微软公司提出了CAPICOM(Cryptographic API Component Object Model)组件,封装CryptAPI中的函数。CAPICOM是一个COM模型的组件,以一种中间件的形式提供了一个标准的密码应用层接口,它介于各种应用与CSP之间[3]。它们之间的关系如图3所示。

CAPICOM可在Windows环境下的各种开发语言中使用,而且,CAPICOM中的大多数接口都是"脚本安全"的,这意味着可以在浏览器网页脚本中安全地使用CAPICOM所提供的各种服务接口。正是由于CAPICOM具有以上优点,所以,在此信息安全传输系统中,我们使用它对数据进行加密操作。

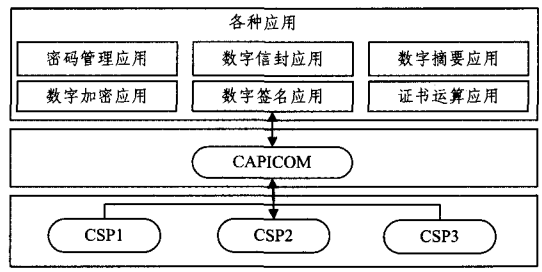

图3 CSP和CAPICOM的体系结构

3.2 数字证书管理

CAPICOM的应用是基于数字证书进行的,如果没有数字证书的支持,CAPICOM所进行的一切安全性操作都将失去意义。

为了实现对数字证书的管理,我们选择使用EJBCA。EJBCA(Enterprise Java Bean Certificate Authority)是一个基于J2EE技术的企业级PKI(Public Key Infrastructure)证书

颁发机构[4]。EJBCA 由 CA(Certification Autohrity,认证中心)、RA(Registration Authority,注册中心)、数据库、CRL(Certification Remove List,证书撤销列表)组成,其中 CA 是 EJBCA 的核心组成部分。根据信息安全传输系统对证书的实际需求,动手搭建了 EJBCA 系统,它对证书的实际管理情况如图 4 所示。

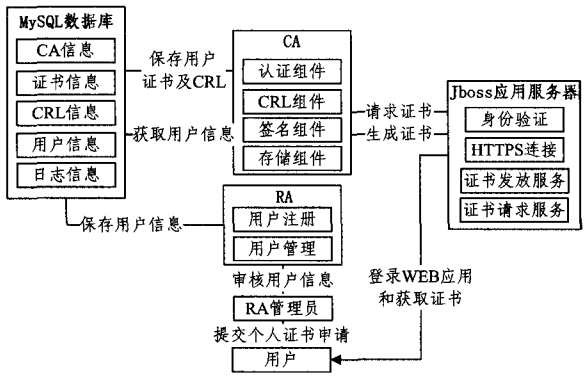

图 4 EJBCA 证书管理

3.3 数字签名和数字信封

目前,CAPICOM 共有两个版权:v1.0 版本和 v2.0 版本。后者在前者的基础上,提供了一些额外的功能。v1.0 提供的主要功能如下:(1)产生和验证 PKCS#7 格式的数字签名;(2)使用证书加密/解密数据;(3)使用口令加密、解密数据。v2.0 提供的额外功能包括:(1)产生和验证代码签名;(2)支持文件形式的证书;(3)快速证书搜索;(4)产生任意数据的哈希值;(5)支持 AES 算法;(6)支持高级证书属性[5]。

在信息安全传输系统的实现过程中,我们选择使用的 CAPICOM 版本为 v2.0。主要实现的工作有本地证书库管理、信息摘要、数字签名、数字信封等功能。在此,主要介绍数字签名和数字信封的实现过程。

(1)数字签名

CAPICOM 中的 SignedData 对象的主要功能是进行数字签名和验证数字签名。通过 SignedData 的 Content 属性来设定待数字签名的数据。调用 SignedData.Sign(Singer,bDetached,EncodingType)方法实现数字签名,其中最重要的参数是 Singer,通过 Singer 的 Certificate 属性来设置数字签名者的证书。最后生成的数字签名为 PKCS#7 格式。数字签名的核心代码如下:

```
Signer.Certificate=clientCert;(客户端证书)
SignedData.Content=hash;(原文的信息摘要)
signature= SignedData.Sign(Signer, false, CAPICOM_ENCODE_BASE64);
```

(2)数字信封

CAPICOM 中的 EnvelopedData 对象的主要功能是创建、发送和接收数字信封。通过 EnvelopedData 的 Content、Algorithm 和 Recipients 属性,分别设置待加密数据、加密算法和接收者证书。调用 EnvelopedData.Encrypt(EncodingType)来形成数字信封,最终得到的数字信封为 PKCS#7 格式。数字信封的核心代码如下:

```
EnvelopedData.Content=signature;(数学签名)
EnvelopedData.Algorithm.Name=2;(DES 加密算法)
EnverlopedData.Algorithm.KeyLenght=0;(密钥为最大长度)
EnvelopedData.Recipients.Add(serverCert);(服务端证书)
envelope=EnvelopedData.Encrypt(CAPICOM_ENCODE_BASE64);
```

4 IAIK 解密

4.1 概述

经过 CAPICOM 的加密操作后,得到的数字签名和数字信封均为 PKCS#7 格式的数据。

PKCS#7 标准定义了加密信息语法标准,描述了待加密数据的一般语法,比如数字签名和数字信封[6]。PKCS#7 格式的数据的验证,主要有两种方式。第一种方式是使用第三方库 IAIK 库,另一种方式是使用 Bouncy Castle。在此信息安全传输系统中,选择使用 IAIK 库验证 PKCS#7 格式的数字签名和数字信封。

IAIK 是 JDK 的扩展库,可以从网上免费下载,它主要包括 4 个 jar 包,分别是:iaik_javax-crypto.jar、iaik_jce.jar、iaik_jce_demo.jar 和 iaik_jce_native_aes.jar。把以上 jar 包引入到信息安全传输系统的项目中,用它实现解析和验证 PKCS#7 格式的数字签名和数字信封。

4.2 验证数字签名和数字信封

服务端验证 PKCS#7 格式的数据签名和数字信封需要经过如下两步:1)解析 PKCS#7 格式的数字信封,从中提取数字签名信息;2)验证数字签名。下面是主要的程序代码:

第 1 步 解析 PKCS#7 格式的数字信封

在解析数字信封的时候,需要两个参数:服务端证书私钥(serverPrivateKey)和数字信封(envelope);返回结果:数字签名(signature);

```
ASN1 obj=new ASN1(new ByteArrayInputStream(envelope));(1)
ASN1Object asn1=obj.toASN10Bject();(2)
ContentInfor ci=new ContentInfo(asn1);(3)
EnvelopedData envelopedData=(EnvelopedData)ci.getContent();(4)
envelopedData.setupCipher(serverPrivateKey,0);(5)
byte[] signature=envelopedData.getContent();(6)
```

(1)数字信封转换成 InputStream 流,再构造 ASN1 对象;

(2)通过 toASN1Object()方法生成 ASN1Object 对象;

(3)创建 PKCS#7 语法中的 ContentInfo 对象;

(4)从 ContentInfo 对象中提取出 EnvelopedData;

(5)利用服务端证书私钥进行解数字信封;

(6)从 EnvelopedData 中提取出 content,content 具体内容由 CAPICOM 进行数字信封时对 EnvelopedData 的 Content 属性进行的设置,此处为数字签名;

第 2 步 验证数字签名

验证数字签名,需要两个参数:数字签名(signature)和待上报数据(data);返回结果:验证数字签名的结果(flag);与第一步中类似的代码此处忽略,只简述验证数字签名的关键代码。

```
String newHash=SHA1.hex_sha1(data);(1)
newHash=newHash.toUpeerCase();(2)
byte[] unicode()fnewHash=newHash.getBytes("UTF-16LE");(3)
SingerInfo[] signer_infos=signedData.getSignerInfos();(4)
for (int i=0;i<signer_infos.length;i++){(5)
X509Certificate signer_cert=signedData.verify(i);
}
```

byte2hex(unicode()newHash). equals(byte2hex(signedData. getContent())))(6)

(1)计算待上报数据的消息摘要,选择与客户端相同的SHA1算法;

(2)消息摘要转换为大写;

(3)消息摘要转换成字节;

(4)获得数字签名者信息;

(5)验证数字签名证书的有效性;

(6)把SignedData中提取出的content(此处为消息摘要)和重新计算的消息摘要均转换成16进制字符串,再判断是否相等;

结束语 在网络上实现信息的安全传输,成为越来越多人的普遍诉求。本文提出了一种实现信息安全传输的方案。在客户端(浏览器端),通过引入CAPICOM技术对待传输的数据实现数字签名和数字信封。使用CAPICOM技术,使得浏览器对数据的加密操作变得十分简单和高效。在服务端,使用第三方库IAIK来解析PKCS#7格式的数字信封,并验证数字签名。同时,作为CAPICOM的基础证书管理部分,信息安全传输系统搭建了EJBCA系统,完成对证书的各种管理。此信息安全传输系统,通过结合数字证书、加密、数字签名、数字信封、PKI等各种网络信息安全传输的技术,实现了信息传输的机密性、完整性、真实性、不可否认性。

参 考 文 献

[1] 宋玲,李陶深,陈拓. 用CAPICOM组件实现应用系统安全性的方法[J]. 计算机工程,2004,30(16):128-129

[2] 李志民. 数字签名和数字信封的比较与应用[J]. 经济师,2006,4:137-138

[3] 张敏,卢巍. 基于CAPICOM的文档管理系统[J]. 计算机应用,2012,32(S1):56-57

[4] 陈勤,凌青山,丁宏. 安全CA实例——EJBCA的研究[J]. 计算机工程与设计,2005,26(12):3222-3224

[5] 谭文学,张健钦,王细萍. 密码中间件CAPICOM的应用研究[J]. 微计算机信息,2006,22(11):112-114

[6] 唐辉天. 将微软CAPICOM组件引入J2EE平台进行数字签名的研究与实现[D]. 西南石油大学,2006

[7] 张文奇,肖衡,段斌,等. Linux平台上基于PKCS#11的PKCS#7 Signed-data的实现[J]. 计算机应用,2003,23(11):103-105

[8] 张青凤,张凤琴. CryptoAPI在基于数字证书身份认证系统中的应用[J]. 现代计算机,2011(24)

[9] 周志刚,徐芳,肖晓华,等. 在Java中进行数字签名的一种实现方法[J]. 科学技术与工程,2006,6(17):2752-2754

[10] 李刚. 轻量级Java EE企业应用实战[M]. 北京:电子工业出版社,2010

[11] 梁栋. Java加密与解密的艺术[M]. 北京:机械工业出版社,2010

[12] 马臣云,王彦. 精通PKI网络安全认证技术与编程实现[M]. 北京:人民邮电出版社,2008

[13] 黄智诚,谢静贤,黄恺昕. 中文WORD 2000使用指南[M]. 北京:中国石化出版社,2000

[14] 胡凯,李腊元. 一个电子商务模型的认证和加密的设计与实现[J]. 计算机工程与设计,2003,24(2):41-43

[15] 王细萍,谭文学,张健钦. CAPICOM在安全电子商务中的应用研究[Z]. 计算机与信息技术,14-17

[16] 王金伟,孙德兵. 基于OpenSSL和CAPICOM身份认证的研究与实现[J]. 福建电脑,2010(08):5-7

[17] 张小江,高翔. CAPICOM组件技术实现数字签名的研究[J]. 商业现代化,2007,19

[18] 徐歆恺,梁军. 巧用CAPICOM进行安全通信[Z]. 计算机与网络,2009(18):53-55

[19] 吴艳. 使用数字证书进行PKCS#7数字签名[J]. 电脑编程技巧与维护,2011(16):130-134

[20] IAIK-JCE 3.13 API:Documentation[OL]. http://javadoc.iaik.tugraz.at/iaik_jce/3.13/overview-summary.html

[21] CAPICOM Reference(Windows) [OL]. http://msdn.microsoft.com/en-us/library/windows/desktop/aa375732(v=vs.85).aspx

[22] EJBCA[OL]. http://www.ejbca.org/

(上接第159页)

结束语 针对传统-KNN算法和距离加权-KNN算法在距离的定义及类别决定上的不足而导致准确率的问题,对这两类算法的特点和不足进行分析,提出了一种基于属性值相关距离的FCD-KNN算法。该算法考虑属性值对分类的重要性,定义样本间的距离为属性值的相关距离,此距离能有效度量两个样本间的相似程度,最大程度地提取与待测样本相似的近邻样本。仿真实验结果表明,该算法能够较大幅度地提高分类的准确率,且在相同测试条件下其并分类的准确率都高于马氏-KNN和距离加权-KNN,证实了FCD-KNN算法的有效性。但由于在实际应用中样本数据集都存在着不同程度的模糊性,这会对算法中样本属性值间的相关度的计算造成较大的干扰,如何解决这一问题并提高FCD-KNN的健壮性是下一步的研究重点。

参 考 文 献

[1] 王增民,王开珏. 基于熵权的K最临近算法改进[J]. 计算机工程与应用,2009,45(30):129-131

[2] 周靖,刘晋胜. 特征联合熵的一种改进k近邻分类算法[J]. 计算机应用,2011,37(7):1787-1792

[3] 陆微微,刘晶. 一种提高k-近邻算法效率的新算法[J]. 计算机工程与应用,2008,44(4):163-165

[4] 周靖,刘晋胜. 一种采用类相关度优化距离的KNN算法[J]. 微计算机应用,2010,31(11):7-12

[5] 杨立,左春,王裕国. 基于语义距离的K-最近邻分类方法[J]. 软件学报,2005,16(12):2054-2062

[6] Wu Xin-dong,Kumar V,Quinlan J R,et al. Top 10 Algorithms in Data Mining[J]. Knowledge and Information Systems,2008,14(1):1-37

[7] 童先群,周忠眉. 基于属性值信息熵的KNN改进算法[J]. 计算机工程与应用,2010,46(3):114-117

[8] 周靖,刘晋胜. 基于特征熵相关度差异的KNN算法[J]. 计算机工程,2011,37(17):146-148