基于 JCE 的 CA 认证系统设计与实现①

Design and Implementation of CA Based on JCE

李海山　王永贵　(辽宁工程大学 电子与信息工程系 辽宁 葫芦岛 125105)

摘　要： CA(Certification Authority)是负责发放和管理数字证书的机构。JCE(Cryptography Extension)为应用程序采用 Java 加密和数字签名提供了一种统一和一致的方式。给出了一种建立在 JCE 基础上的 CA 认证系统的实现性。

关键词： CA JCE 数字证书

1　引言

随着互连网的发展，电子政务、电子商务等日益普及，随之出现的网络安全问题也日益突出和重要，只有有效地解决网络安全问题，真正意义上的电子政务和电子商务才能得到良好的发展。

数字证书是由证书管理机构即数字证书认证中心签发的不可伪造的在网络上标识通讯各方身份信息的一系列数据的电子文件。数字证书是互联网上的网络身份证，它具有电子签名及信息加密两大功能，能在互连网上起到身份确认，保障信息的安全和完整及信息的不可否认性等作用[1]。其在实际 应用中能有效地解决两个问题：一、能有效地解决如网上商店、网上银行、网上政府等相关操作方的身份认证问题，确保网络上身份是真实的、不可假冒的，并能有效地保障相关操作信息安全性、完整性及不可否认性，即采用数字证书能有效地建立一个安全可靠的网络上的可信任环境；二、应用数字证书特定的电子签名功能对相关的网络操作或传输信息进行电子签名并加盖时间戳电子印记，通过数字认证中心的第三方验证可起到防抵赖、防否认和防假冒的作用，能为处理可能出现的网上行为纠纷提供可靠有效的法律证据[2]。

本文在考虑各方面问题的基础上给出了一种基于 JCE 的认证中心系统——LCA 的设计与实现。

2　CA实现功能

CA 即 Certificate Authority，指数字证书认证中心，它采用 PKI(Public Key Infrastructure)公开密钥基础架构技术，基础性地构建网络上安全可靠的信任环境，专门提供网络身份认证服务，是负责数字证书的签发和管理，具有权威性和公正性的第三方信任机构，负责数字证书的整个生命周期的管理[3]。本 LCA 包括如下功能：

1) 证书签发：通过 CA 认证系统，能够申请、产生和分发数字证书，具有证书签发功能。

2) 证书生命周期管理：通过 CA 认证系统，可以实现证书的生命周期管理，证书申请最终用户使用浏览器，访问 CA 认证系统，可以进行证书申请，在线提交证书申请请求； 证书批准管理员登录管理员站点，完成证书批准功能，可以查看和审批最终用户的证书申请请求； 证书查询最终用户可以通过 CA 认证系统，查询自己或别人的数字证书；证书下载通过 CA 认证系统，可以下载签发的数字证书；用户吊销证书时，可以直接访问 CA 认证系统，在线的向 CA 提交证书吊销请求，CA 认证系统根据用户的选择，自动吊销用户的证书，并将吊销的证书添加到证书吊销列表(CRL)中，按照证书吊销列表的发布周期进行发布。

3) CRL 服务功：CA 认证系统支持证书黑名单列表(CRL)功能，CA 认证系统定时产生 CRL 列表，并将产生的 CRL 发布至 Web 层 CRL 服务模块，可以通过手工下载该 CRL。

4) 目录服务功能：CA 认证系统支持目录服务，支持 LDAP V3 规范，CA 认证系统在签发用户证书时

① 收稿时间:2009-02-24

或者对证书进行吊销处理时，会及时更新目录内容。证书目录服务的功能提供给用户进行证书查询的功能，用户可以通过电子邮件、用户名称、单位名称和部门名称(OU)等字段查找 CA 认证系统签发的用户证书。

5) 综合管理功能：管理员管理，包括初始化管理员申请、增加管理员、删除管理员以及其它信息的管理。

3 系统结构设计

LCA 系统的逻辑结构如图 1 所示。本系统采用模块化结构设计，由最终用户、RA-CA 管理员、注册中心(RA)、认证中心(CA)等构成，系统架构如图 1：

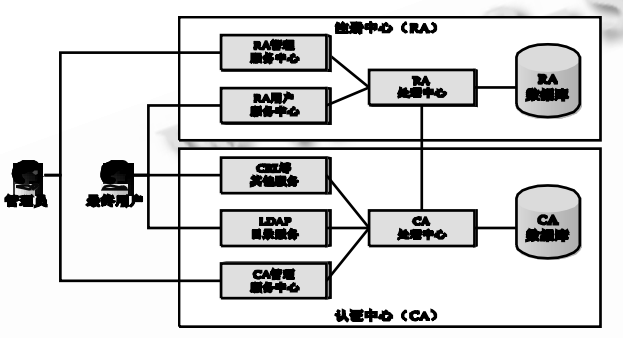

图 1　LCA 系统架构图

首先用户访问 RA 用户服务中心，提交证书申请请求，申请数字证书；管理员访问 RA 管理中心，审查和批准用户的证书申请请求；CA 认证中心根据管理员的批准，签发用户证书，然后 RA 处理中心与 CA 处理中心交互，决定是否授予用户证书，而后将数字证书发布到目录服务器中，用户以后可以访问目录服务器看自己的证书。

证书签发以后，当用户证书的私钥受到威胁、或者用户私钥丢失时，需要吊销用户的证书，根据用户信息系统的应用情况，本方案设计证书吊销由管理员进行，管理员访问 CA 管理服务中心，进行用户证书吊销工作，然后提交到 CA 处理中心。 CA 处理中心根据管理员的证书吊销请求，自动的吊销用户的证书，并将吊销的用户证书发布到证书吊销列表中，同时对数据库中保存的用户证书的最新状态进行更新，CA 认证系统给管理员返回证书吊销成功信息，同时给用户发送电子邮件，告诉用户证书已经被吊销，不能再使用自己的证书。

4 系统实现

4.1 实现工具

Java 平台为安全和加密服务提供了两组 API：JCA 和 JCE。JCA (Java Cryptography Architecture)提供基本的加密框架，如证书、数字签名、消息摘要和密钥对产生器； JCE(Cryptography Extension)在 JCA 的基础上作了扩展，包括加密算法、密钥交换、密钥产生和消息鉴别服务等接口。但是 JCE 对部分国家是限制出口的。因此，要实现一个完整的安全结构，就需要一个或多个第三方厂商提供的 JCE 产品，称为安全供应者。BouncyCastle JCE 就是其中的一个安全供应者。它提供了可以在 J2ME/J2EE/J2SE 平台得到支持的 API[4]。LDAP(轻量级目录服务访问协议，Light 书馆 weight Directory Access Protocol)基于 X.500 标准，支持 TCP/IP，使用简单方便。现在越来越多的网络应用系统都支持 LDAP。OpenLDAP 是 LDAP 的一种开源实现，本系统使用 OpenLDAP 作为目录服务器。

4.2 具体实现

本系统建立在 Windows 系统上，管理员信息等基本信息存储在 MySql 数据库中，认证系统签发的用户证书和证书吊销信息存储在 OpenLDAP 目录服务器中，WEB 表示层使用 JSP 技术，服务器采用 Tomcat 6.0。主要的证书签发模块使用 JBPM(Java Business Process Management)工作流实现。由于篇幅有限，有的程序代码已经做了精简，类结构并不完整，保留核心部分，剔除了异常处理等，下面将给出主要的模块实现。

1) 证书签发模块，本系统证书签发模块采用 JBPM 工作流机制实现，Eclipse 开发平台为 JBPM 提

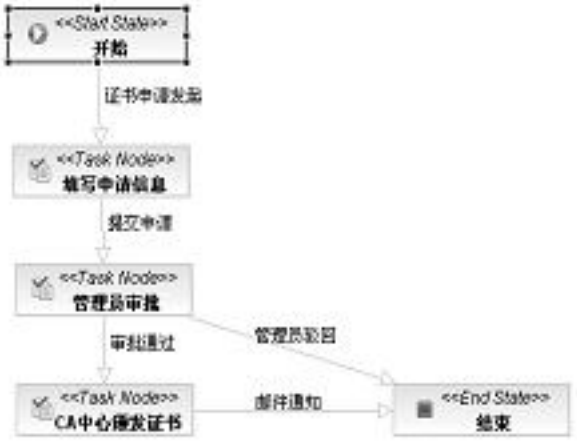

图 2　证书签发模块的流程定义图

供了良好的可视化支持，可以考虑签发证书是一个具体的工作流程，用户提出证书申请请求，填写申请信息，向处理中心提交。管理员查看到用户的申请，如果确认通过审批，则交由 CA 中心颁发证书，邮件通知用户申请成功。否则撤销用户本次申请。系统的证书签发模块的流程定义见图 2。

下面是其中 CA 中心签发证书的核心代码：

```
public void SignCert() {
// 获得根 CA 证书
CertificateFactory
certCF=CertificateFactory.getInstance("X.509");
FileInputStream certBIS = new FileInputStream("rootCA.cer");
X509Certificate caCert = (X509Certificate)
certCF.generateCertificate(certBIS);
// 构造一个证书生成器对象
X509V3CertificateGenerator certGen = new X509V3CertificateGenerator();
// 从 CA 的证书中读取 CA 的 DN 进行 DER 编码
DERInputStream dnStream = new DERInputStream(new ByteArrayInputStream (caCert.getSubjectX500Principal().getEncoded()));
// 从编码后的字节流中读取 DER 编码对象
DERConstructedSequence dnSequence =(DERConstructedSequence)
dnStream.readObject();
// 利用读取出来的 DER 编码对象创建 X509Name
certGen.setIssuerDN(new X509Name(dnSequence));
// 设置好证书生成器中的"接收方 DN"certGen.setSubjectDN(subjectDN);
// 设置好一些扩展字段,包括签发者和接收者的公钥标识certGen.addExtension(X509Extensions.SubjectKeyIdentifier,false,createSubjectKeyId(keyToCertify));certGen.addExtension(X509Extensions.AuthorityKeyIdentifier, false,createAuthorityKeyId(caCert.getPublicKey()));
// 设置证书的有效期和序列号
certGen.setNotBefore(startDate);
certGen.setNotAfter(endDate);
certGen.setSerialNumber(serialNumber);
// 设置签名算法，签名,并且设置好主体的公钥
certGen.setSignatureAlgorithm("MD5withRSA");
certGen.setPublicKey(keyToCertify);
X509Certificate cert =certGen.generateX509Certificate(caPrivateKey);
File userfile = new File("userCA.cer");
FileOutputStream caCerOut = new FileOutputStream("userfile");
caCerOut.write(cert.getEncoded());
caCerOut.close();}
```

2) 根 CA 生成，利用 openSSL 生成一个用于 CA 服务器的自签名证书。执行程序时，系统会提示输入一些相关信息，如申请人姓名、电子邮箱和地址等，然后用 CA 的自签名证书对证书申请签名。签名后，将证书分别拷贝到各个应用服务器，继续处理证书，最终完成配置工作[5]。具体的命令见表 1：

表 1　根证书生成的相关命令及描述

命令	描述
openssl genrsa -out root/root-key.pem 1024	创建私钥
openssl req -new -out root/root-req.csr -key root/root-key.pem	创建证书请求（这里会要求输入一些信息）
openssl x509 -req -in root/root-req.csr -out root/root-cert.pem -signkey root/root-key.pem -days 3650	产生受信任的证书文件
openssl pkcs12 -export -clcerts -in root/root-cert.pem -inkey root/root-key.pem -out root/root.p12	将受信任的证书导出成浏览器支持的.p12(PKCS12)格式

3) 证书吊销，通常最终用户发现自己的证书不安全，或者私钥被别人窃取，或者相关操作人员离职，都需要吊销证书，当用户向 CA 提出吊销请求的时候，需要输入吊销证书的信息，然后 CA 管理员使用 CA 管理员证书，访问管理站点，CA 管理员按照用户的通知，查询到需要吊销的用户证书,选择吊销原因,点击吊销，系统便自动吊销用户的证书。下面是核心代码:

其中，X509CrlParser 用于构建一个 crl 对象，支持从字节数组和内存流中获取数据。X509Crl 对象，包含证书吊销组织、吊销证书列表、时间戳等信息。

```
Public void makeCRL(){
//获取 obj
List<int> numbers = new List<int>();
X509CrlParser parser = new X509 CrlParser();
X509Crl crl = parser.ReadCrl((byte[])obj);
//获取所有的吊销证书
ISet crlSet = crl.GetRevokedCertificates();
if (crlSet != null && crlSet.Count > 0)
{foreach (object o in crlSet) {
X509CrlEntry crlEntry = (X509CrlEntry)o;
int serialNumber = crlEntry.SerialNumber. IntValue;
if (!numbers.Contains(serialNumber)) {numbers.Add(serialNumber); } }}}
```

5 系统分析

通过上述设计与实现的详细阐述，可以看出本 LCA 具有区别于其它 CA 的以下独立特点:

1) 将 CA 服务器与其他服务器隔离，任何通信采用人工干预的方式，以确保认证中心的安全。

2) 系统所有用户、管理员界面都是 B/S 模式，CA/RA 策略配置和定制以及用户证书管理等都是通过浏览器进行。易于部署和操作。

3) 引入 JBPM 工作流思想，对证书签发业务流程进行监控管理，提高业务工作效率。

4) 为了提高系统的安全性,将经过 DER 编码的私钥,进行加密,储存在文件系统中.在使用私钥的时候,如果没有正确的口令,无法还原私钥。

5) 系统在设计中遵循了相应的国际和工业标准，包括 X.509 标准、IETF 的 PKIX 工作组制定的 PKI 相关 RFC 标准，以及 HTTP、 SSL、LDAP 等互联网通讯协议。严格遵循这些标准，使得系统具有很好的开放性，能够与各种应用结合，成为真正的安全基础设施。

6) 支持多操作系统、多证书存储介质、多加密设备，拥有较高的兼容性。

6 结语

当前，比较知名的 CA 机构都是国外的，要想从他们那里获得认证证书，需要缴纳高昂的认证费用。而国内的 CA 还处于起步阶段，提供的公司也比较少，所以，对于国内较大的机构，如大型企业和政府部门来说，一个较好的方案就是构建自己的认证中心。本文提出了用 JCE 构建一种 CA 的方法，并给出了具体实现。LCA 由于使用 Java 语言所以可以在不同的平台之间移植。可以看出针对当前国内 CA 的现状，本 LCA 具有强大的功能和较好的安全性，应用和理论价值有广泛的前景。

参考文献

1 Andrew N, William D, Celia J,著. 张玉清,陈建奇,杨波,译.公钥基础设施(PKI)实现和管理电子安全.北京, 2002:166－169.

2 石峰,蒋亚丽.企业级桥 CA 系统的研究与设计.计算机应用, 2003,23(6):76－77.

3 钟鸣,冉春玉.基于身份认证技术的 CA 系统的研究.福建电脑, 2006,(6):12－13.

4 简艳红,周爱霞.Java 实现浏览器 CA 的签名和认证.商丘职业技术学院学报, 2007,6(2):42－43.

5 张浩然.用 Java 和 OpenSSL 实现认证中心.计算机应用研究, 2004,21(5):158－159.