第34卷 第21期 计 算 机 工 程 2008年11月
Vol.34 No.21 Computer Engineering November 2008
·安全技术· 文章编号：1000—3428(2008)21—0109—03 文献标识码：A 中图分类号：TP309

基于 J2EE 多层结构的认证中心

马 骥，江为强，杨义先
(北京邮电大学网络与交换国家重点实验室信息安全中心，北京 100876)

摘 要：分析证书签发系统的实现机制和私钥签名的相关流程，给出一个基于 J2EE 多层模式的证书签发管理系统设计方案。该方案采用 EJB 组件等技术，实现证书/CRL 的生成、签发、查询和下载、证书验证、证书撤销及证书模板管理等功能，降低应用系统的建设成本和部署难度，具有良好的可扩展性。

关键词：公钥基础设施；证书签发管理系统；认证中心

Certificate Authority Based on J2EE Multilayer Architecture

MA Ji, JIANG Wei-qiang, YANG Yi-xian
(Information Security Center, State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, Beijing 100876)

【Abstract】 This paper analyzes the mechanism of issuing and managing system of certificates and its signature process, and describes the scheme of issuing and managing system of certificates based on J2EE multilayer model. This scheme utilizes EJB component technology to implement relevant functions, including certificate/CRL creating, issuing, query, downloading, certificate verifying, revoking and certificate templates management, etc. It decreases the cost and difficulty of development and of application systems, and ensures good extensibility.

【Key words】 Public Key Infrastructure(PKI); issuing and managing system of certificates; Certificate Authority(CA)

1 概述

公钥基础设施(Public Key Infrastructure, PKI)技术采用证书管理公钥，通过认证中心(Certficate Authority, CA)把用户公钥和其他用户标识信息捆绑起来，用于验证用户身份[1-6]。J2EE平台是Sun公司为了快速设计并开发企业级应用程序而推出的一种全新概念的模型，其基本思想是通过一个基于组件的应用程序模型为分布式应用程序提供统一标准。

运用 J2EE 技术进行 PKI 系统开发具有很多优点，例如利用 Java 语言中已有的资源提高开发速度并满足系统安全性需求，采用 EJB 组件技术使 PKI 系统有利于扩展。J2EE 的跨平台性使 PKI 系统适用于使用不同平台的用户，且系统易于移植。

本文论述的 PKI/CA 认证系统主要分为 2 个部份：证书签发管理系统和证书注册审核系统。本文重点讨论证书签发管理系统的设计与实现。

2 证书签发管理系统框架

证书签发管理系统采用 J2EE 技术开发，遵循 J2EE 规范。如图 1 所示，整个系统采用 4 层结构的设计模式，具体如下：

(1)用户界面层。处理用户与应用程序的交互。

(2)表示逻辑层。定义用户内容和如何处理用户请示。Web 组件可以是 Servlet 或 JSP 页面。

(3)业务逻辑层。确定应用程序的业务规则，业务逻辑代码表示与特定商业领域相适应的逻辑，它由运行在业务逻辑层的 EJB 处理。作为 J2EE 架构中最重要的构件，EJB 是实现服务器端分布式计算的核心。

(4)企业信息系统层。处理企业信息系统软件并包含诸如企业资源计划(ERP)、主机事务处理、数据库系统和其他类似的传统系统底层。本文称此层为数据库层。

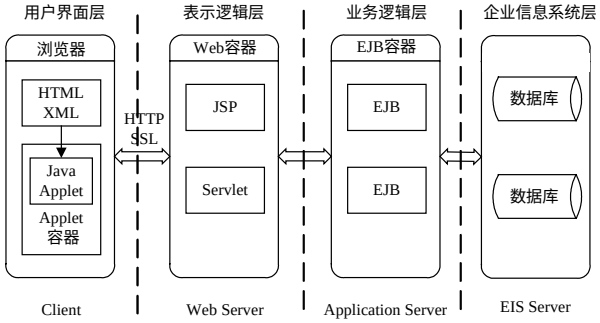

图 1 系统软件结构

3 证书签发管理系统的设计

CA 认证系统主要包括以下流程：

(1)证书申请流程。用户在 RA 提出证书申请后，把此申请转发给 CA，并由 CA 签发用户证书。用户通过证书下载模块获得证书，并把证书和相应的私钥写到终端实体证书储存介质 EEkey 或导入本地系统中。

(2)证书撤销流程。用户提交证书撤销口令 PIN，RA 接收后向 CA 转发证书撤销申请，CA 接收后将此用户证书状态设为证书撤销状态，此时 CA 向 KMDB 请求销毁与证书对应的托管加密/解密密钥对。CA 还要将撤销证书写入 CRL 表，发布到 LDAP 上。

(3)密钥恢复流程。密钥恢复是在用户解密私钥丢失等情况下，向 RA 申请下载丢失的解密私钥的过程。

(4)密钥和证书更新流程。CA 认证系统设计为自动完成

作者简介：马 骥(1984－)，男，硕士研究生，主研方向：计算机网络，网络安全；江为强，博士研究生；杨义先，教授、博士生导师
收稿日期：2008-01-22 **E-mail：**mjhorse@sina.com

密钥和证书的更新，系统接收用户的密钥与证书的更新申请后，会检查有效期。如果证书已过有效期，则接受更新申请，生成一个新证书代替旧证书。

证书签发管理系统具备向密钥管理中心申请密钥、用自己的私钥签发数字证书、子 CA 根证书和私钥管理以及对证书资料信息的管理、对用户的访问行为进行日志记录等功能。它还具备操作员的管理功能。

证书签发管理系统结构如图 2 所示，主要包括业务管理模块、证书管理模块、证书签发模块和操作员管理模块。

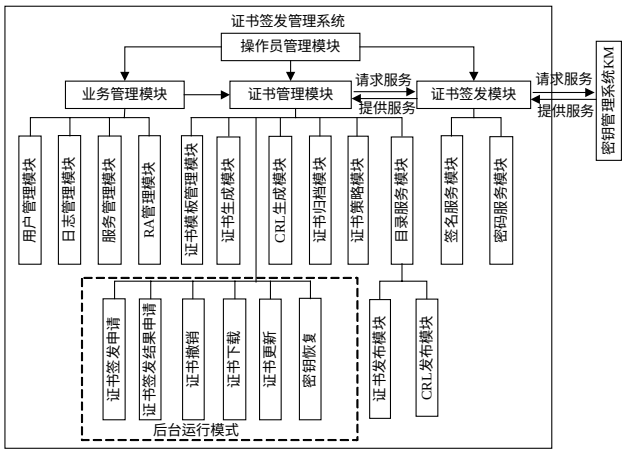

图 2　证书签发管理系统结构

对证书签发管理系统各主要部分表述如下：

(1)CA 服务器。是整个证书签发管理系统的核心，主要功能包括：制定证书策略，初始化 RA，接收 RA 签发、撤销和更新证书请求，为端实体或下属 CA 生成密钥对，签发、撤销和更新证书，发布证书和证书撤销列表(CRL)，签发交叉证书以及密钥托管和恢复。CA 服务器是整个结构中最重要的部分，存有 CA 的私钥和发行证书的脚本文件。根据实际需要，CA 服务器可决定为用户签发双证书或单证书。

(2)LDAP 服务器。LDAP 服务器提供目录浏览服务，负责将用户信息及数字证书加入到服务器上，使用户可以通过访问 LDAP 服务器得到其他用户的数字证书。

(3)数据库服务器。数据库服务器是认证机构中的核心部分，用于认证机构中数据(如密钥和用户信息等)、日志统计信息的存储和管理。

CA 认证系统的其余部分包括证书注册审核系统和终端实体即证书用户。

4　证书签发管理系统的实现

本系统采用 Windows2000 Server 和 RedHat Linux 7.2 进行测试，Web 服务器采用 Tomcat4.1.24 进行部署，应用服务器采用 JBoss3.0.8 进行部署、LDAP 服务器采用 OPENLDAP，关系数据库采用 MySql3.23，系统开发工具采用 JBuilder9.0。本系统涉及大量密码相关操作，这些密码操作主要由 Java 语言中的 2 组与实现无关的加密函数 API 提供，分别是 Java 密码构架(Java Cryptography Architecture, JCA)和 Java 密码扩展(Java Cryptography Extension, JCE)。本系统的设计重点是对会话 bean 和实体 bean 的设计及各 EJB 之间的交互过程设计。

会话 bean 专用于表达用户与 CA 认证系统在一次会话中完成的动作，即会话中的业务流程。重要的会话 bean 包括：

(1)证书/CRL 签发会话 bean——CertCrlSignSession，是证书签发模块的核心会话 bean。它接收证书生成模块的签发申请，用子 CA 系统的根证书签发证书或 CRL，并通过证书/CRL 存储会话 bean 进行签发证书的存储操作，通过密钥对管理会话 bean 向 KM 服务器申请并获得加密/解密密钥对。

(2)证书/CRL 存储会话 bean——CertCrlStoreSession，是证书管理模块中最重要的会话 bean。证书管理模块中相关子模块通过证书/CRL 存储会话 bean 完成各种与数据库中证书及 CRL 相关资料的管理操作(包括数据的插入、查询、删除和更新等)。

(3)证书申请接收会话 bean——CertReqRecieveSession，用于证书签发申请等模块中，负责将审核通过的用户申请存入 CA 端的证书申请接收表 CertRequest 中，等待证书签发。

(4)证书撤销管理会话 bean——CertRevokeManageSession，是证书撤销模块中的重要会话 bean，管理证书撤销的整个过程，包括接收证书撤销的申请、修改证书状态、通过调用相关会话 bean 生成、存储和发布 CRL 以及销毁加密/解密密钥对等。

(5)生成 CRL 会话 bean——MakeCRLSession，管理 CRL 的生成过程，归属于 CRL 生成模块，主要进行已撤销证书的获取及调用证书/CRL 签发会话 bean 进行 CRL 的签发。

(6)密钥对管理会话 bean——KeyPairManageSession，在证书签发模块中用于管理 CA 服务器与 KM 服务器中的交互过程。主要任务是向 KM 服务器申请并获得密钥对，以及向 KM 服务器申请销毁密钥对。

(7)密钥恢复会话 bean——PriKeyRecoverySession，在密钥恢复模块中管理密钥恢复的整个过程，包括接收密钥恢复申请，通过调用 KeyPairManageSession 从 KM 服务器重新获得解密私钥等。

(8)基类 bean——OriSessionBean，实现上述会话 bean 的一些基本方法，上述会话 bean 继承了 OriSessionBean 中 ejbActivate(),ejbPassivate(),ejbLoad(),ejbStore()和 ejbRemove()等公共方法。

实体bean描述存储在数据库表中、持久稳固的数据，确定了数据存储的模型，是数据的一个对象包装器。CA认证系统中的重要性数据，如证书、CRL和用户信息，采用实体bean表示，并通过操作实体bean实现对它们的持久性管理[2]。

在证书签发管理系统的设计过程中，持久性的数据，如证书、CRL、证书模板、加密证书中的加密/解密密钥对均采用容器管理方式的实体 bean 来表示，包括 CertDataBean(证书实体 bean)、CRLDataBean (CRL 实体 bean)、CertTemplate DataBean(证书模板实体 bean)和 KeyPairDataBean(密钥对实体 bean)等。

4.1　证书签发设计

证书签发管理系统要设计的模块主要包括系统初始化、证书签发、证书信息查询、证书及私钥下载、证书撤销、证书更新、密钥恢复、证书验证等。下文重点对证书签发模块、证书撤销模块和证书验证模块进行描述。

证书签发管理系统要签发的证书包括服务器证书和用户证书。其中，用户证书签发的总体思路是 CA 服务器先查询已通过 RA 审核并已将申请转发给 CA 请求签发证书的用户，从 KM 为该用户申请加密/解密密钥对，并获取加密公钥来签发证书。其具体步骤如下：

(1)在 CA 数据库的表 CertRequest 中查询得到需要生成证书的 UserName(记录集查询操作)，系统根据 UserName 查询 UserInfomation 表得到该用户的相关信息。

(2)CA服务器将密钥对请求结构数据StruApplyKey(含证书申请唯一编号ApplyUniqueID、用户名UserName、EEKeyID、密钥对强度、密钥对生成算法等信息)用CA服务器证书签名后得到E_{KS_CA}[StruApplyKey](KS_CA为CA服务器的签名私钥)并传递到KM，KM收到后用CA服务器证书验签，得出密钥对请求信息。

(3)KM生成回执结构数据StruKeyBack(含UserName, EEKeyID和用户加密公钥KP_USER等信息)，将StruKeyBack和KM服务器证书对StruKeyBack的签名信息合并后，用CA服务器证书进行加密，得到E_{KP_CA}[StruKeyBack|E_{KS_KM}[StruKeyBack]](KP_CA为CA服务器的加密证书对应的公钥，KS_KM为KM服务器的签名证书对应的私钥，KP_USER为用户证书生成所需的公钥)，KM将此信息传送至CA服务器。

(4)CA先用CA服务器证书私钥KS_CA解密得KM所传公钥，并用KM服务器证书验签，然后用子CA系统根证书为该用户签发证书，将签发后的证书信息写入CA数据库表CertInfomation和CertPublish中。此时CertInfomation中该证书的证书状态为CERT_GENERATED，CA可以将它们发布到LDAP服务器上。

在整个证书签发过程中，通过CertSignManagerServlet来管理操作员的查询工作并统一调用CertCrlSignSession实现证书的签发及存储入库。CertCrlSignSession继承的类主要是java.secutity.*和org.bouncycastle.*。

4.2 证书撤销

证书撤销的体系结构与流程如图3所示。

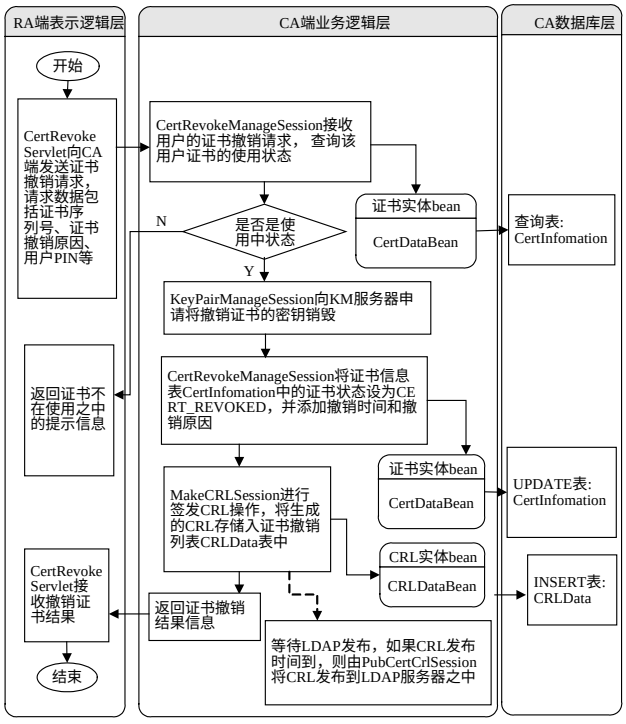

图3 证书撤销的体系结构与流程

证书撤销模块的功能包括接收撤销申请、证书记录状态修改、KM服务器中密钥对移除、CRL生成、CRL导出、CRL发布和CRL下载等。整体流程是证书签发管理系统接收RA请求，将需要撤销的证书状态设为证书撤销状态，并将撤销证书的信息传递给KM，在KM中将撤销证书的密钥销毁，并将撤销证书信息写入CRL表，等待LDAP发布。它还提供本地数据库CRL下载等其他功能。

在以上流程中，CertCrlSignSession的creatCoreCRL()是核心，它通过CRL生成器类X509V2CRLGenerator的generateX509CRL()方法，采用CA根证书对应的私钥进行签名，从而生成CRL。

4.3 证书验证

证书的验证主要包括对证书有效性的认证和对证书链的验证。

4.3.1 证书有效性认证

证书有效性认证的主要内容包括可信CA是否在证书上签名、证书是否有良好的完整性、证书是否在有效期内、证书有没有被撤销、证书的使用方式与任何声明的策略和使用限制是否一致，具体如下：

(1)验证证书的有效期。首先获取X509Certificate证书对象和当前日期，然后通过调用X509Certificate类的checkValidity()方法来判断证书是否过期。

(2)验证可信CA是否在证书上签名、证书是否有良好的完整性。此过程需要用到子CA系统的根证书cacert，先通过X509Certificate类的getPublicKey()方法提取根证书的公钥，再通过X509Certificate类的verify()方法进行证书签名验证。

(3)验证证书是否被撤销。CRL中存放了由CA签署的已被撤销的证书序列号。通过CRL验证证书的过程可通过X509CRL类的isRevoked( )方法判断证书状态来实现。

4.3.2 证书链验证

对证书链的验证是指通过证书链追溯到可信赖的根CA并沿证书链的反向路径进行证书的拆封，以获取关于终端实体证书是否有效的信息。图4描述了证书链的验证过程，如图4箭头所示，先沿着信任路径追溯到根CA，然后沿原路返回进行证书的拆封，即进行CA签名的验证。证书链的验证包括对证书序列号的验证，证书序列号的验证是指检查终端实体证书中的“颁发者证书的序列号”是否与CA证书中的“证书序列号”一致，以验证证书的真伪。

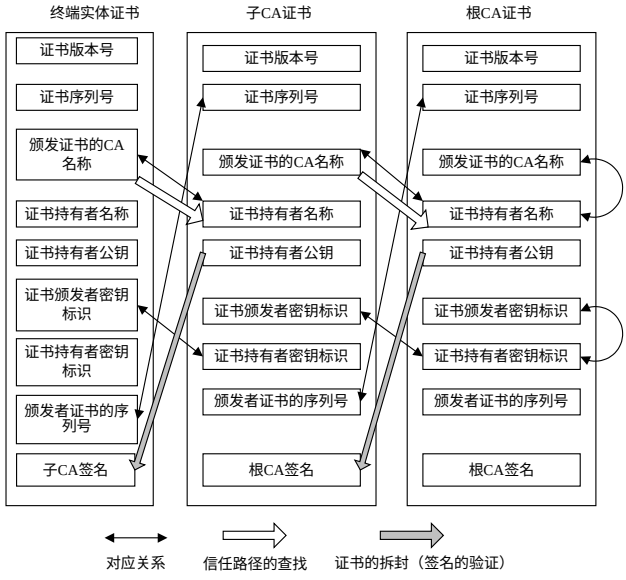

图4 证书链的验证过程

本系统证书链的验证使用CertPathValidator类直接对CertPath类型的对象进行验证，使用TrustAnchor对象设置ROOTCA的证书。CertPathValidator类中的validate()方法可以使用现成的PKIX certification path验证算法直接验证CertPath类型的对象。 （下转第114页）