# 独 创 性 声 明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京工业大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

签名：王先明 日期：2010.6.6

# 关于论文使用授权的说明

本人完全了解北京工业大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

（保密的论文在解密后应遵守此规定）

签名：王先明 导师签名： 日期：2010.6.6

# 摘要

公共对象请求代理体系结构 CORBA 是为了解决分布式异构环境下对象之间的互操作性问题而提出的基于中间件的分布式对象技术，其核心是一套标准的语言、接口和协议，以支持异构分布式应用程序间的互操作性以及独立于平台和编程语言的对象重用。由于 CORBA 在信息系统中的广泛应用，基于 CORBA 的分布式系统的安全问题成为人们关注的焦点。本文的目标是以 CORBA 和 CORBA 安全服务规范为基础，基于 PKI/CA 技术、SSL 技术和 OpenSSL 技术，提出一种基于 CORBA 的分布式系统安全架构，并给出安全架构各个模块的详细设计，从而满足基于 CORBA 的分布式系统的安全需求。

本文首先对 CORBA 技术、PKI/CA 技术、SSL 技术和 OpenSSL 技术进行了研究，并分析了分布式系统存在的威胁以及相应的安全机制。然后重点研究了证书管理、身份认证、访问控制以及安全通信的方法，并在此基础上提出了基于 CORBA 的分布式系统安全架构，包括证书管理模块、安全服务模块和安全通信模块。其中，证书管理模块主要基于 OpenSSL 提供的 API，包括证书生成和发放、证书吊销和证书吊销列表更新和 LDAP 目录服务等功能。身份认证和安全通信主要基于 SSL 协议和 OpenSSL 的 API，访问控制主要基于 CORBA 安全服务规范中的访问控制模型。

本文最后将文中提出的基于 CORBA 的分布式系统安全架构应用到了基于 CORBA 的网络税控系统中。

**关键词** CORBA；PKI；CA；SSL；OpenSSL

# ABSTRACT

CORBA is a distributed object technology based on middleware and its purpose is to solve the interoperability problems happened between objects in the distributed heterogeneous environment. The core of CORBA is a standard set of languages, interfaces and protocols which are used to support the interoperability of distributed heterogeneous applications and object reuse which is platform-independent and language-independent. Due to the widely use of CORBA in the information systems, the security issues of CORBA-based distributed system become the focus of attention. The target of this thesis is to present a security architecture for the CORBA-based distributed systems based on the PKI/CA technology, the SSL technology, the OpenSSL technology, CORBA and the CORBA security service specification and to give the detailed design of each module in the architecture so that it can meet the security needs of CORBA-based distributed systems.

This thesis presents the research on the technology of CORBA, PKI/CA, SSL and OpenSSL first, and analyzes the distributed system security threats and the corresponding security mechanism, and then focus on the research of the methods of certificate management, authentication and access control, and finally presents a security architecture for the CORBA-based distributed systems, which includes the certificate management module, the security service module and the secure communication module. The certificate management module which main features include certificate generation, certificate issue, certificate revocation, certificate revocation list update and LDAP directory service is mainly based on the API provided by OpenSSL. The authentication and secure communication are based on the SSL protocol and the API provided by OpenSSL, and the access control is mainly based on the access control model of the CORBA security service specification.

At the end of this thesis, the security architecture for the CORBA-based distributed systems is integrated into the CORBA-based network tax control system.

**Keywords** CORBA; PKI; CA; SSL; OpenSSL

# 目 录

# 第1章 绪论

## 1.1 课题背景

税收是国家为满足社会公共需要，凭借公共权力，按照法律所规定的标准和程序，参与国民收入分配，强制地、无偿地取得财政收入的一种方式，是国家保障经济有序发展、政治稳定的强制性管理手段，严格、准确、及时足额的收缴税款体现着国家意志和利益。商场消费，尤其是到大中小型超市消费是国民几乎天天要进行的经济活动，然而现如今超市企业的税收环节却出现了可有可无，可多可少的现象，对税务机关的管理工作造成了不好的印象，同时又影响了税收工作的严肃性。国家税务管理机关一直致力于依法治税，按律收税，也在不断的与偷税、漏税和逃税的行为做斗争，但是缺少第一手的税源数据是造成税收管理力度跟不上的重要原因之一。

CORBA(Common Object Request Broker Architecture，公共对象请求代理体系结构)是 OMG(Object Management Group，对象管理组织)为了解决分布式、异构软硬件环境下对象之间的互操作性问题而提出的基于中间件的分布式对象技术，其核心是一套标准的语言、接口和协议，以支持异构分布式应用程序间的互操作性以及独立于平台和编程语言的对象重用[1][2]。CORBA 有很广泛的应用，它易于集成各厂商的不同计算机，从大型机一直到微型内嵌式系统的终端桌面，是针对大中型企业应用的优秀的中间件。最重要的是，它使服务器真正能够实现高速度、高稳定性处理大量用户的访问。CORBA 将面向对象技术与客户/服务器计算模式结合起来，有效地解决了对象封装和分布式计算环境中资源共享、代码可重用、可移植及应用间的互操作性等问题。

网络税控系统是用于对商场超市进行税务监控的基于 CORBA 的分布式税控管理系统，并且已经制订了国标七规范，其核心功能是向国家税收部门提供大中型零售商场的销售信息即税源数据以加强对税收的管理。然而税源数据涉及到企业商品的销售价格，销售利润等保密信息，如被窃取会对企业造成巨大的损失。因而既要保证税源数据的准确性，没有虚假被篡改伪造的数据，又要保证数据的机密性，不被不法分子窃取，还要保证数据的完整性，只有完整的数据才有进一步研究的价值。而税控系统作为一个信息系统，又需要进行身份认证和访问控制以确保系统的安全性。因此，网络税控系统的安全性具有很现实的研究意义。

## 1.2 国内外研究现状

### 1.2.1 CORBA 研究现状

CORBA 是 OMG 为了解决分布式、异构软硬件环境下对象之间的互操作性问题而提出的基于中间件的分布式对象技术，其核心是一套标准的语言、接口和协议，以支持异构分布式应用程序间的互操作性以及独立于平台和编程语言的对象重用。

CORBA 是针对 OMA(Object Management Architecture，对象管理体系结构)参考模型中的 ORB（Object Request Broker，对象请求代理）而制订的规范。CORBA 规范定义了 IDL（Interface Definition Language， 接口定义语言）及其映射、接口仓库和实现仓库、动态调用接口、存根和框架、ORB 协议以及单个 ORB 和 ORB 间的互操作机制等。CORBA 的核心思想是采用标准的接口定义语言将接口与实现分离，CORBA 使得基于对象的软件在分布异构环境下具有良好的可重用性、可移植性以及互操作性，并且使得在由多种主流平台上运行多种操作系统构成的异构分布环境中方便地构造异构分布系统成为可能[3]。目前，OMG 组织已经制定并发布了 CORBA 的多个版本。

1991 年，OMG 发布了 CORBA 1.0，定义了接口定义语言 IDL，动态调用接口 DII，接口库以及 IDL 到 C 语言的映射。

1995 年，OMG 发布了 CORBA 2.0，定义了动态框架接口 DSI、支持客户端移植的初始应用解析器、接口库的扩展、互操作体系、层次化的安全和事务服务、IDL 到 C++和 Smalltalk 的语言映射等。CORBA 2.0 主要解决了两个问题：一是标准化了 IDL 到 C++的映射，由于 C++是当时主流的程序开发语言，因而使得 CORBA 也成为商业主流；二是提出 CORBA 中基于 TCP/IP 的 IIOP（Internet Inter-ORB Protocol，互联网内部对象请求代理协议）协议，提高了不同厂商的 ORB 产品之间的互操作性和通讯能力。

1999 年，OMG 发布了 CORBA 3.0，提出了 CORBA 的构件模型思想，并增加了容错 CORBA 部件和可移植拦截器，提出了新的安全互操作规范，改进了互操作体系。

CORBA 安全方面，OMG 于 1997 年发布了 CORBA 安全服务规范 1.1 版本，目前最新版本是于 2002 年发布的 CORBA 安全服务规范 1.8 版本。CORBA 安全规范对 CORBA 系统安全服务的标准概念、CORBA 安全参考模型、CORBA 安全体系架构、应用开发接口、互操作模型以及 CORBA 安全协议等进行了详细的定义和说明，为基于 CORBA 的安全系统的开发提供了基础。

### 1.2.2 信息系统安全研究现状

信息安全的基本目标是实现信息的机密性、完整性、可用性和资源的合法使用[4]。目前，国内外对于信息系统安全的研究主要针对以下几个方面：

1. 安全体系结构理论与技术的研究，主要研究如何利用形式化的数学描述和分析方法　建立信息系统的安全体系结构模型。
2. 安全协议理论与技术的研究，主要包括协议的安全性分析方法和各种实用安全协议 的设计与分析。协议的安全性分析方法主要有两类：一类是攻击检验法，通过使用各种有效攻击方法，逐一对使用安全协议的系统进行攻击，检验安全协议抵抗攻击的能力；另一类是形式化分析方法，即采用各种形式化的语言或者模型，建立安全协议模型，并按 照规定的假设和分析、验证方法等来证明协议的安全性。
3. 信息系统安全监控和保护技术的研究，主要包括网络安全整体解决方案的设计与分析，网络安全产品的研发等。网络安全监控是为了保障运行中的网络免受外来干扰和破坏而对网络实施的安全保护措施；网络保护技术主要是指网络访问控制和审计管理技术。包括防火墙、路由器、代理服务器、访问日志等。
4. 密码学及密码技术的研究，主要包括对称密钥体制、非对称密钥体制、数字签名与身份认证及非数学密码理论与技术的研究等。

### 1.2.3 SSL 及其研究现状

SSL(Secure Socket Layer，安全套接层)是 Netscape 公司设计的用于 HTTP 协议加密的安全传输协议，用以保障 Internet 上数据传输的安全。SSL 工作于一个可靠连接的通信协议之上，通常来说是 TCP（Transmission Control Protocol，传输控制协议）协议，采用数字证书进行身份认证和密钥交换，在传输过程中实现对信息机密性和完整性的保障[5]。SSL 协议可以在客户端和服务器之间建立一个安全的网络通道。它是一个基于 PKI（Public Key Infrastructure，公钥基础设施）的网络数据安全协议，具有保护传输数据以及识别通信机器身份的功能。SSL 对于传输的数据不加变更，客户端与服务器之间的数据是经过加密的，一端写入的数据完全是另一端读取的内容，这种透明性使得几乎所有基于 TCP 的协议稍加改动就可以在 SSL 上运行。为了防止通信过程中的监听、篡改以及消息伪造，SSL 提供了服务器认证和可选的客户端认证，通过在两个实体之间建立一个共享的秘密通道来提供保密性。

目前国内外对于数据传输这一领域的研究主要集中在网络层和传输层，一般而言，网络层次越低，安全服务的透明性就相对越差，但优点是灵活性强，应用

范围广。国外对传输层安全的研究起步较早，而且密码学、公钥基础设施 PKI 等相关技术的研究也比较成熟。1995 年 Netscape 公司制定了 SSL 协议并推出了其参考实现，虽然我国对网络安全的研究起步较晚，但已经在密码学和公钥基础设施等领域取得了丰硕的成果。

## 1.3 论文主要研究内容

本文的研究内容是基于 CORBA、SSL 以及 PKI/CA 的安全技术，并使用 OpenSSL、SSL 和 PKI/CA 技术来设计并实现基于 CORBA 的分布式网络税控系统的安全策略和安全架构。本文的主要内容包括以下几点：

- 研究了 CORBA 技术和 CORBA 安全规范，研究了基于 CORBA 的分布式系统所面临的安全威胁以及相应的解决方案。
- 研究了 SSL 技术、OpenSSL 技术和 PKI/CA 技术，并提出了基于 SSL、OpenSSL 和 PKI/CA 的基于 CORBA 的分布式系统的安全架构。
- 基于提出的分布式系统的安全架构，设计并实现了网络税控系统的安全策略。

## 1.4 论文结构安排

第一章 绪论。主要介绍了本论文所研究课题的背景和意义，分析了国内外的研究现状，介绍了课题的来源及研究内容。

第二章 基础理论。主要介绍了本论文所涉及的信息系统安全、CORBA、SSL、OpenSSL、数字证书、数字签名、身份认证、LDAP 以及 PKI/CA 等技术的相关概念，并介绍了分布式系统面临的安全威胁和所应具备的安全机制。

第三章 系统总体安全架构设计。本章在第二章介绍的基础理论的基础上，提出了一种基于 CORBA 的分布式系统安全架构。

第四章 系统安全架构详细设计。本章根据第三章提出的安全架构，对证书管理模块、安全通信模块和安全服务模块中的身份认证和访问控制部分进行了详细设计。

第五章 网络税控系统安全策略设计与实现。本章介绍了网络税控系统，并将本文中提出的基于 CORBA 的分布式系统安全架构应用到网络税控系统中。

# 第2章 基础理论

## 2.1 信息系统及信息系统安全简介

信息系统是指以计算机为主要工具，可对信息进行采集、处理，存储，管理，检索，传输，并可为人们提供有用信息的联合体[6]。近半个世纪以来，随着计算机技术的迅猛发展，尤其是计算机网络技术的发展，信息系统作为一个专门领域迅速形成，并被应用到社会的各行各业中。

在人们关注信息系统发展的同时，信息系统的安全问题也越来越引起人们的关注。随着 Internet 网络的发展，信息内容在网络中传输时的保密性、完整性、可用性和抗抵赖性等成为人们关注的焦点。

关于信息系统安全的概念，1991 年，英、法、德、荷四国联合提出了安全的信息系统较完整的概念，即一个安全的信息系统应在如下几个方面具有保障[7]：

- 保密性：确保信息不向未经授权者泄漏。
- 完整性：防止信息被未经授权者篡改，保证真实的信息无失真地传输到目的地。
- 可用性 ：保证信息及信息系统确实为授权使用者所用，防止由于计算机病毒或其 它人为因素造成的系统拒绝服务。
- 可控性：对信息及信息系统实施安全监控管理。
- 不可否认性：保证信息行为人不能否认自己的信息行为。

## 2.2 分布式系统存在的威胁及安全机制

分布式系统存在的威胁主要有以下几个方面[8]：

- 系统授权用户访问到应该对他隐藏的信息。
- 用户伪装成其他用户，并得到只有其他用户才能访问的信息。
- 在一个分布式系统中，用户可以将自己的权利授权给其他对象从而让其他对象作为他的代理完成某些操作，造成未授权访问的威胁。
- 安全控制被绕过。
- 窃听通信线路，并获得机密数据。
- 篡改对象之间的通信数据。
- 缺少足够的身份认证机制。

针对分布式系统存在的安全威胁，分布式系统所应具备的安全机制有[8]：

- 用户识别和认证，核实主体身份。
- 授权和访问控制，判断主体是否有权限访问他想要访问的对象。
- 安全审计，识别用户使用户对他所作的和安全相关的操作负责。
- 对象之间的安全通信，尤其是不安全的较低层的通信。
- 不可否认性，提供不可辩驳的活动证据，避免活动的发起者否认这一活动的发生。
- 安全信息的管理，管理安全策略等信息。

## 2.3 CORBA 介绍

CORBA 是 OMG 提出的一种分布式对象技术标准。CORBA 的核心是一套标准的语言、协议和接口，用于支持异构分布式应用程序间的互操作性以及独立于编程语言和平台的对象重用。CORBA 的主要目标是解决面向对象的异构应用之间的互操作问题。

OMG 组织成立后不久就制订了 OMA 参考模型，该模型描述了 OMG 规范所应遵循的概念化的基础结构。OMA 的参考模型如图 2-1 所示：

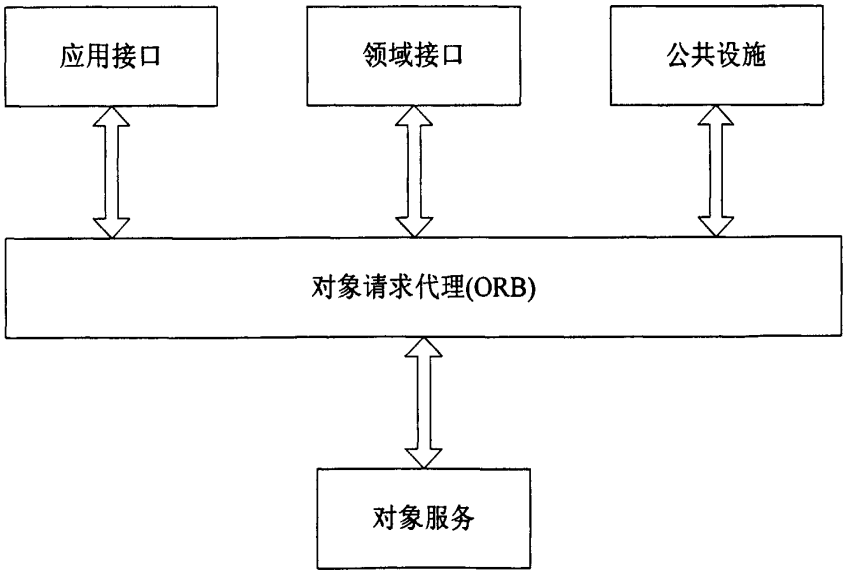


图 2-1 OMA 参考模型
Fig 2-1 OMA reference model

### 2.3.1 ORB 介绍

OMA 的核心部分是 ORB。对象服务是为了使用和实现对象而提供的基本服务集合，这些服务是与应用领域无关的接口，供分布对象调用[9]。公共设施是向终端用户应用程序提供的一组共享服务接口集合；领域接口是为应用领域服务而

提供的面向特定领域的接口；应用接口是由开发商提供的接口，不属于 OMG 标准的内容，OMG 只给出接口说明，而不开发任何具体应用。

CORBA 是针对 OMA 参考模型中的对象请求代理 ORB 而制订的规范。CORBA 规范定义了 IDL 语言及其映射、接口仓库和实现仓库、IDL 映射、动态调用接口、存根和框架、ORB 协议以及单个 ORB 和 ORB 间的互操作机制等。CORBA 的核心思想是采用标准的接口定义语言将接口与实现分离[10] [11]。CORBA 规范的基本组成如图 2-2 所示：

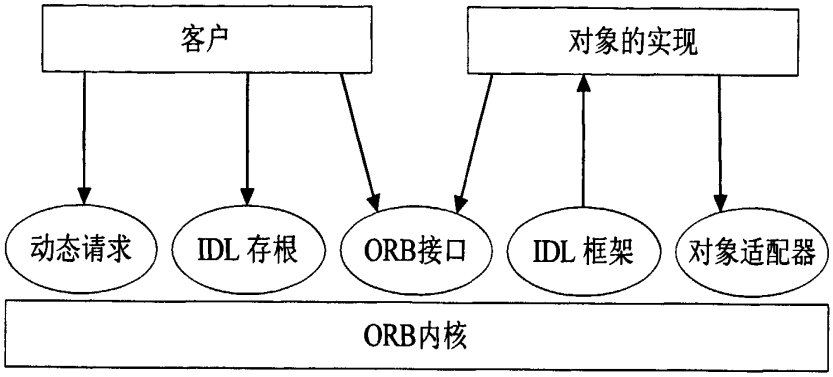


图 2-2 CORBA 体系结构
Fig 2-2 CORBA architecture

### 2.3.2 OMG 接口定义语言

当调用一个分布式对象时，在客户端向服务器发送请求之前，必须了解这个对象是否存在，如果存在必须了解这个对象所提供的接口和服务。同时，为了允许不同的编程语言和操作系统来处理 CORBA 对象，必须对描述对象接口和服务的标准达成一致。CORBA 就是通过 OMG 接口定义语言定义的对象接口来说明对象所能提供的服务。并通过接口定义语言使得 CORBA 做到编程语言无关。OMG 接口定义语言是一种用来描述客户端调用接口和服务器端对象实现接口的语言。OMG IDL 的语法与 C++类似，只是增加了一些支持分布式处理的关键字。

OMG IDL 仅仅是一种描述语言，不是编程语言，所以对象和应用程序不能用 IDL 实现，而是需要具体的编程语言来实现。IDL 作为一种描述语言，通过与 IDL Compiler 编译器的配合使用，从 IDL 文件中生成存根和框架样本[12]。这就把编程者从编写大量枯燥的样本代码中解救了出来。IDL 只说明对象支持的属性和方法，而不做具体的实现，编程者可以根据需要选择不同的编程语言按照 IDL 文件中的定义来实现具体的对象，这种允许程序以不同的编程语言来实现以便于程序的互操作性是 CORBA 支持分布式系统和独立开发的应用程序相互集成的关键。

## 2.4 CORBA 安全服务规范

CORBA 安全服务是 OMG 组织基于分布式系统安全保护的重要性以及 CORBA 所面临的安全性威胁而提出的安全服务规范，其主要思想如下[13]：

- 执行安全服务功能的模块与具体的安全策略相分离，即使具体的安全策略发生变化，应用程序也不需要进行相应的重组，大大提高了安全服务的可扩展性。
- 实现对象级别的保护，使得安全逻辑的实现与执行更加灵活,可以根据特定的需求做相应的设置。
- 与具体的安全技术相分离，实现安全功能的接口被对象调用时应该对上层的应用和 ORB 内核隐藏底层安全技术的具体细节。因此，无论是对称加密技术还是非对称加密技术,都可以应用到 CORBA 安全服务中来。

### 2.4.1 安全互操作/SecIOP 规范

CORBA 安全服务规范不仅描述了 CORBA 的安全服务，还定义了安全的 ORB 间协议 SecIOP，该协议和 GIOP/IIOP 一起为不同厂商的 CORBA 安全服务实现提供互操作性。为了实现安全互操作性，必须使用 IOR 标签和安全标志在客户机和目标对象间建立安全连接。ORB 安全服务下运行的对象，必须提供一个包含已标记安全组件的 IOR，并需要给出和对象相关联的安全策略信息[14]。当客户机要和该对象进行安全通信时，IOR 提供通信要求的安全级别初始信息。建立于 SecIOP 协议之上的通用安全互操作规范添加了认证机制以及加密算法等方面的细节，并提供了无委托的基于标识的安全策略。

### 2.4.2 ORB-SSL 集成规范

DCE(Distributed Computing Environment 分布式计算环境)和 RACF(Remote Access Control Facility 远程访问控制设备）安全服务是在制定 CORBA 安全服务规范和通用安全互操作规范时，最通用的系统安全服务。SSL 是为 Internet 应用而设计的传输层安全标准，并迅速成为安全 Internet 通信的广为接受的标准。随后，OMG 组织制定了一个规范，描述了 SSL 和 CORBA ORB 集成的基本需求。ORB-SSL 栈的图示如图 2-3 所示：

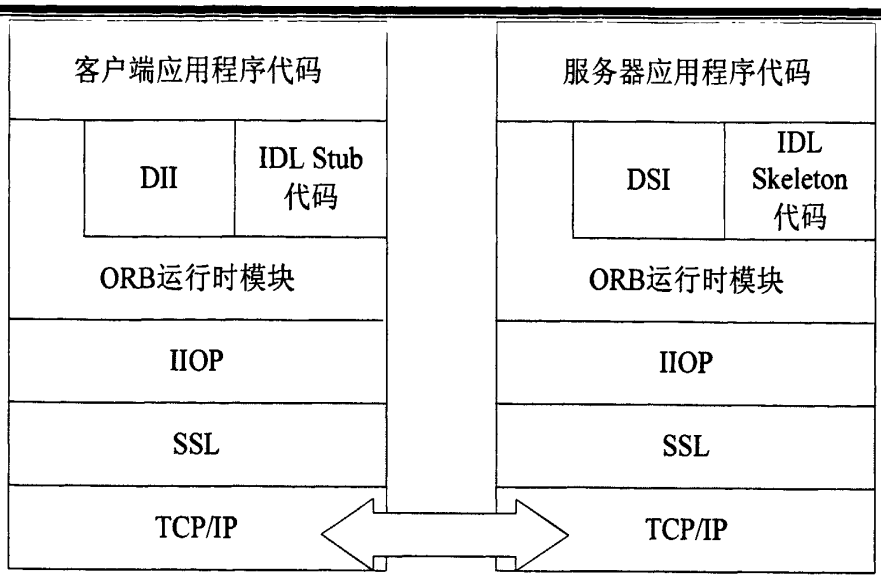


图 2-3 ORB-SSL 栈
Fig 2-3 ORB-SSL stack

### 2.4.3 CORBA/防火墙规范

CORBA/防火墙规范规定了防火墙如何处理 IIOP 请求，以使得防火墙所管理的 CORBA 对象能完成外界激发的操作。CORBA/防火墙规范规定了如何部署和设置能处理 IIOP 请求的防火墙，使得防火墙能像其他协议一样处理和授权 IIOP 请求。这些防火墙应该支持以下特性：

- 像普通的应用程序协议一样处理 IIOP 请求。由防火墙来决定用何种网络通信来实现 IIOP 和进行访问控制以及哪种 IIOP 通信可以通过防火墙。
- 保护内部目标对象免受无效 IIOP 数据流的攻击。

## 2.5 PKI 和 CA 概述

PKI(Public Key Infrastructure，公钥基础设施)，是一种通过使用公开密钥技术和数字证书来确保系统信息安全并负责验证数字证书持有者身份的信任体系，是一类构造巧妙的基础设施，可以以高效、统一的方式提供安全服务[15] [16]。PKI 是一个可以提供多种安全防护的长期解决方案，利用数字证书标识密钥持有人的身份，通过对密钥的规范化管理来建立和维护一个可信赖的系统环境，并透明地为应用系统提供身份认证、抗抵赖、数据保密性和完整性等各种必要的安全保障，从而满足各种应用系统的安全需求。简单的说，PKI 的目的是为了自动管理密钥和证书，保证网络信息传输的机密性、完整性、真实性、和不可否认性[17]。

PKI 体系主要由 KMC(Key Management Center，密钥管理中心)、

CA(Certificate Authority Center，数字证书认证中心)、RA(Registry Authority Center，注册权威中心) 和发布中心组成[18]。

- 密钥管理中心负责管理密钥的整个生命周期，为数字证书认证中心提供密钥服务。
- 数字证书认证中心是 PKI 的核心组成部分，负责对数字证书和证书撤销列表的签发与管理。数字证书认证中心是一个可信的权威机构，来对任何一个主体的公钥进行公证 并证明主体的身份以及主体与其公钥的匹配关系 CA 负责数字证书生命周期的管理。
- 注册权威中心是 PKI 与用户交互的平台，负责对证书用户的注册和管理，并完成数字证书的最终制作与发放。
- 发布中心负责发布对外信息，包括数字证书和证书吊销列表(CRL)等信息的发布。

典型 PKI 体系结构如图 2-4 所示：

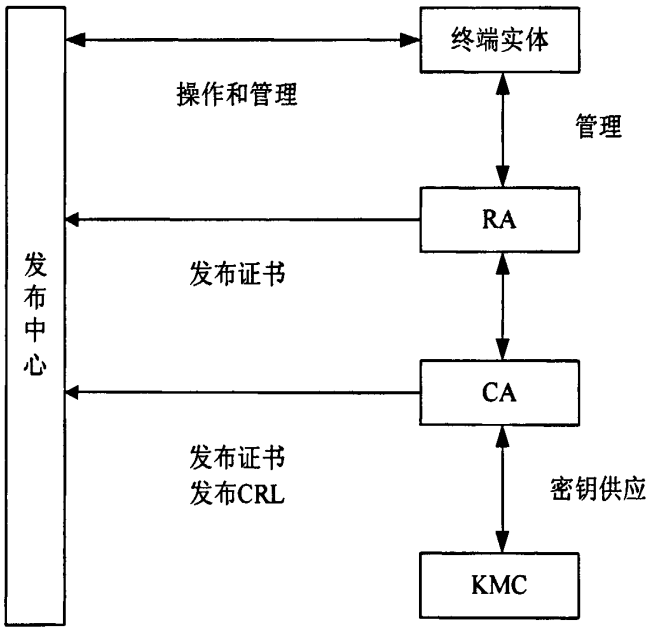


图 2-4 PKI 体系结构
Fig 2-4 PKI architecture

### 2.5.1 LDAP 概述

LDAP（Lightweight Directory Access Protocol，轻量级目录访问协议），是基于 X.500 标准的目录访问服务，LDAP 协议的设计目的是在不增加 X.500 目录访问协议资源需求的情况下访问 X.500 的目录[19]。目前，LDAP 协议已在 PKI 体系中被应用于证书信息发布、CRL 信息发布、CA 政策以及与信息发布相关的各个方面。LDAP 主要优点如下[20]：

- 简单通用，由于 LDAP 的高可靠性和良好性能，LDAP 目录服务能满足

绝大部分重要的目录服务需求；

- 易于定制，可以方便地提供不同的 LDAP 目录服务供用户使用；
- 快速搜索，LDAP 目录具有快速的搜索速度和很强的过滤器功能；
- 安全特性，LDAP 目录可以通过强制使用安全特定来保护目录信息不被窃取和篡改；
- 分布特性，LDAP 目录可以存放在网络中不同的机器上，实现分布式目录管理。

LDAP 目录中的基本数据单元是条目，条目由 DN 名和多个属性构成，属性则由属性类型以及属性值构成，属性类型由描述名以及 OID 构成，包含了属性取值所应遵循的文法以及属性查询时所应使用的匹配规则。创建条目时，必须同时定义条目所属的对象类以及提供对象类中必选属性类型的属性值。条目的目录结构如图 2-5 所示：

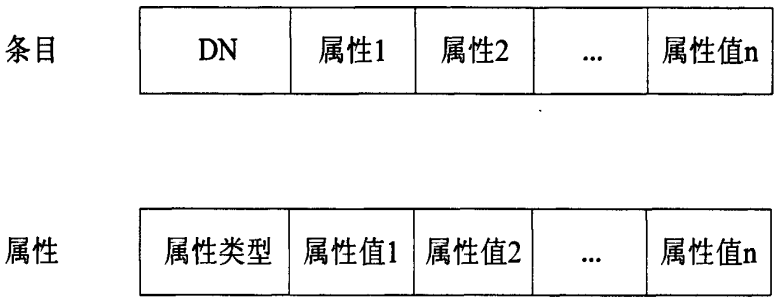


图 2-5 条目的目录结构
Fig 2-5 Entry directory structure

LDAP 目录的物理结构以树的形式描述，条目的 DN 名是条目各个属性的串联，是条目在整个树中的唯一名称标识，将 DN 名分开来，每一个部分称为 RDN，RDN 是条目在父节点下的唯一标识。

## 2.5.2 密码技术

将明文消息变成密文消息的过程称为加密，将密文消息变成明文消息的过程称为解密[21]。要加密明文消息，发送方要采用加密算法进行加密；要解密密文消息，接收方要采用解密算法进行解密[22]。目前密码算法主要分为两类：对称密码算法和非对称密码算法。

对称密码算法的加密密钥和解密密钥相同，或者虽然不相同，但是从其中一个可以很容易推得另一个。对称密钥算法要求发送方和接收方在安全通信之前，必须明确一个密钥。对称密码算法的优点是计算开销小，加密解密的效率高，缺点是密钥管理困难。

非对称密码算法，又称公开密钥算法，加密密钥和解密密钥不同。公开密钥算法的加密密钥可以公开，其他用户可以得到公开密钥并用来加密数据，但只有

用相应的私有密钥才能解密信息[23]。运用非对称加密算法的技术称为公钥加密技术，该技术是 PKI/CA 技术的基础。公钥加密技术主要应用在两个方面：一是身份认证，利用加密技术来鉴别用户身份，另一个数据加密，保证传输数据的机密性。

### 2.5.3 数字证书

数字证书，是由证书机构签名，并包含公开密钥拥有者信息、公开密钥、有效期、签发者信息以及其他一些扩展信息的数字文件[24]。数字证书分为加密证书和签名证书[25]。加密证书主要用于用户传送的数据进行加密，以保证数据的机密性；签名证书主要用于对用户信息进行签名，从而保证数据的完整性和行为的不可否认性。

数字证书的格式遵循 X.509 标准。X.509 是由国际电信联盟指定的数字证书标准，是随 PKI 的形成而发展起来的安全机制[26]。

数字证书主要包含三个基本组成部分：待签名证书、签名算法和数字签名[27]。待签名证书是数字证书的主题，包含了证书的基本内容，主要有：证书版本号、证书序列号、证书签发者、证书持有者、签名算法、证书有效期、持有者公钥信息以及证书扩展项等内容。

### 2.5.4 X.509 证书

X.509 证书结构主要包括三个基本部分：待签名证书、签名算法和数字签名。它的基本数据结构如下[28]：

```
Certificate ::= SEQUENCE{
    tbsCertificate    TBSCertificate, //待签名证书
    signatureAlgorithm    AlgorithmIdentifier, //签名算法标识符
    signatureValue    BIT STRING //数字签名值
}
```

待签名证书是证书的主体，包含了证书的基本内容，其基本数据结构如下：

```
TBSCertificate ::= SEQUENCE{
    version    [0]    EXPLICIT Version, //证书版本号
    serialNumber    CertificateSerialNumber, //证书序列号
    signature    AlgorithmIdentifier, //签名算法
    issuer    Name, //证书签发者
    validity    Validity, //证书有效期限
    subject    Name, //证书持有者
    subjectPublicKeyInfo    SubjectPublicKeyInfo, //公钥信息
```

issuerUniqueID [1] IMPLICIT UniqueIdentifier OPTIONAL, //签发者唯一标识

subjectUniqueID [2] IMPLICIT UniqueIdentifier OPTIONAL, //持有者唯一标识

extensions [3] EXPLICIT Extensions OPTIONAL //证书扩展项

}

### 2.5.5 身份认证

身份是实体的属性，通过身份可以唯一标识实体，在信息系统中，通常情况下，需要对用户的身份进行认证，以确定谁在使用系统以及可以赋予该用户何种操作权限[18]。身份认证目前主要依靠数字证书技术完成，常用的使用数字证书的身份认证流程如图 2-6 所示：

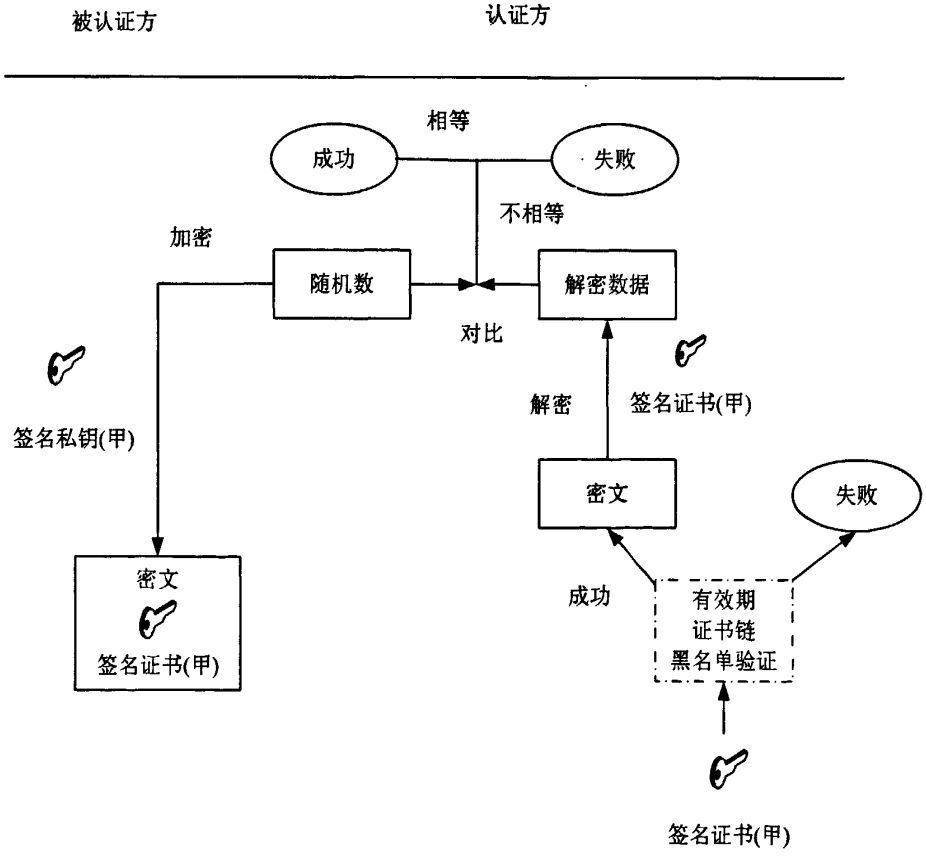


图 2-6 数字证书身份认证流程

Fig 2-6 Digital certificate authentication process

身份认证的步骤如下：

1. 认证方向被认证方发送一个随机数；
2. 被认证方使用自己的签名私钥将认证方提供的随机数进行加密；
3. 被认证方将自己的签名证书和密文发送给认证方；
4. 认证方验证被认证方所提供的签名证书的有效期、证书链；
5. 有效期、证书链和黑名单验证通过后，认证方使用被认证方的签名证书对甲所提供的密文进行解密，将认证方提供给被认证方的随机数与解密结果进行比对，相等则表明可以接受由被认证方提交的签名证书所申明的身份，身份认证过程结束。

### 2.5.6 数字签名技术

数字签名是数字证书的基本应用之一，是不对称加密算法的典型应用[29]。数字签名技术是在虚拟网络系统环境中用于身份确认的重要技术，完全可以代替现实过程中的亲笔签字。

常用的使用数字证书进行签名和验证签名的流程如图 2-7 所示：

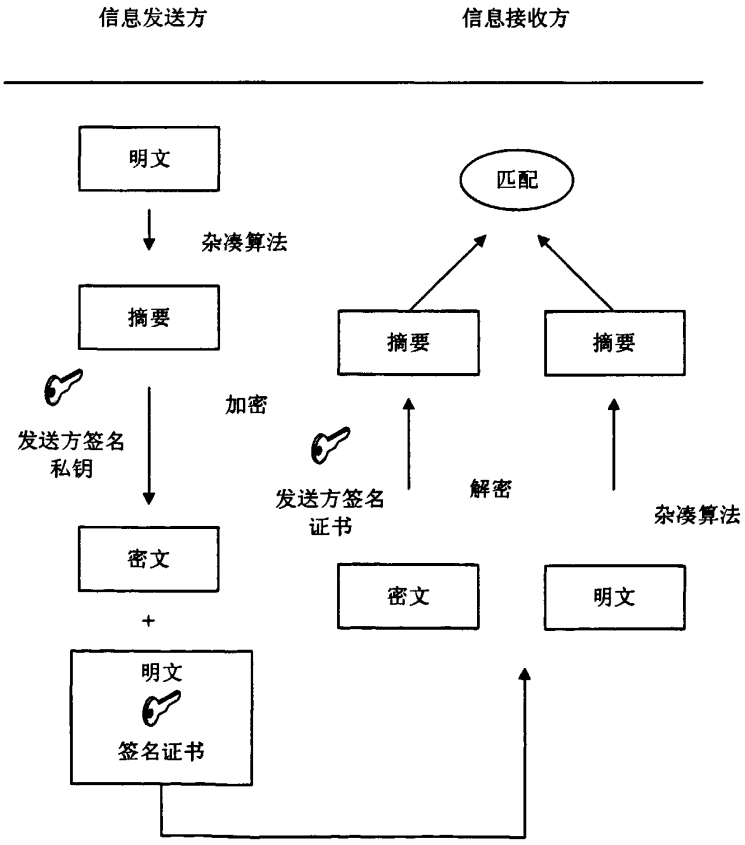


图 2-7 数字证书签名流程
Fig 2-7 Digital certificate signature process

数字签名的具体步骤如下:

1. 信息发送方将杂凑算法应用于原始数据，并生成一个摘要值;
2. 信息发送方使用私钥对摘要值进行加密，得到数字签名;
3. 信息发送方将原始数据、签名及发送方的签名证书发送给信息接收方;
4. 信息接收方验证签名证书的有效性;
5. 信息接收方将相同的杂凑算法应用于接收到的数据，并生成一个摘要值，同时将得到的签名通过发送方的签名证书进行解密得到摘要值;
6. 信息接收方对比两个摘要值，如果相同，则可以确认原始数据在传输过程中没有被更改，并且信息是由签名证书所申明身份的实体所发送的。

## 2.6 SSL 概述

SSL(Secure Socket Layer，安全套接层)是 Netscape 公司设计的用于 HTTP 协议加密的安全传输协议，用以保障 Internet 上数据传输的安全[30]。SSL 工作于一个可靠连接的通信协议之上，通常来说是 TCP 协议，采用数字证书进行身份认证和密钥交换，在传输过程中实现对信息机密性和完整性的保障。

SSL 协议提供的服务主要有[31]:

- 认证用户和服务器，确保数据发送到正确的客户机和服务器。
- 加密数据以防止数据中途被窃取。
- 维护数据的完整性，确保数据在传输过程中不被改变。

SSL 协议可以看成 TCP/IP 协议组中的另一层，位于应用层和传输层之间，独立于应用层协议，即建立在 SSL 之上的应用层协议可以透明地传输数据。SSL 协议被设计成使用 TCP 协议提供端到端的安全服务，SSL 并不是单个协议，而是由多个协议组合而成。SSL 协议是一个由两层协议组成的分层协议组，处于底层的是 SSL 记录协议（Record Protocol），位于高层的是握手协议（Handshake Protocol）、修改密钥协议（Change Cipher Spec Protocol）以及报警协议（Alert Protocol）[32]。SSL 协议的层次结构如图 2-8 所示:

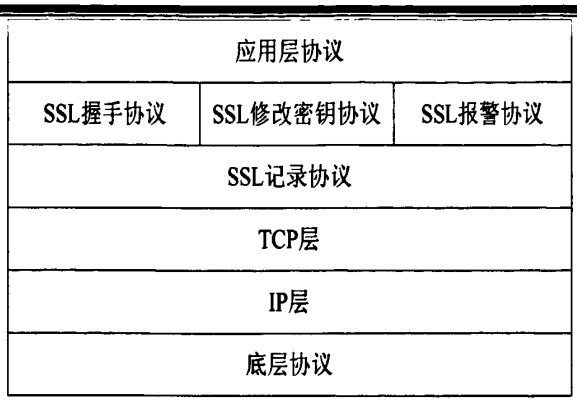


图 2-8 SSL 协议层次结构
Fig 2-8 SSL protocol architecture

## 2.6.1 SSL 握手协议

SSL 握手协议是客户端与服务器用 SSL 连接通信时使用的第一个子协议，握手协议包括客户端和服务器之间的一系列消息。当 SSL 客户端和服务器开始通信时，他们需要协商一个协议版本并选择密码算法，对彼此进行验证并使用公开密钥加密技术产生共享密钥。SSL 握手协议的流程如图 2-9 所示：

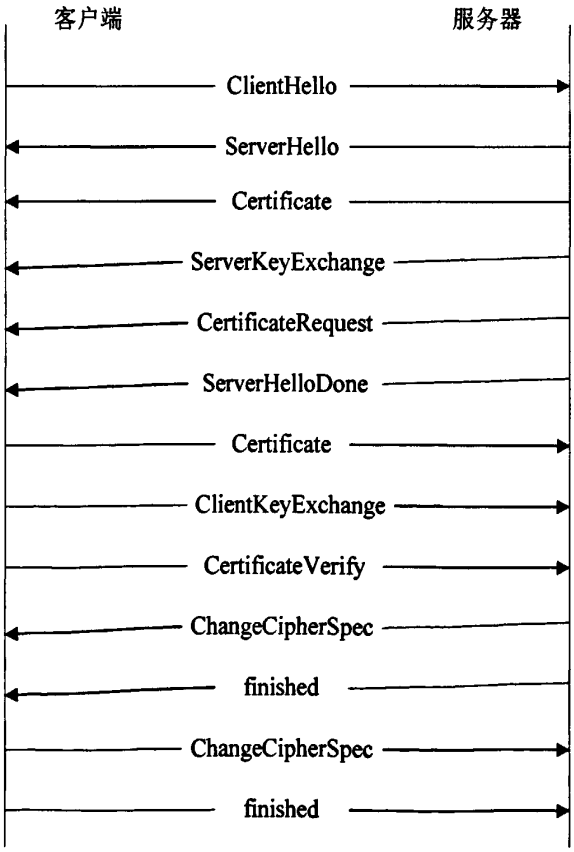


图 2-9 SSL 握手流程
Fig 2-9 SSL handshake process

SSL 握手的第一阶段启动逻辑连接，建立这个连接的安全能力，包括 ClientHello 和 ServerHello 两个消息，客户端与服务器直接协商确定协议版本号、会话号、密码组以及压缩方法。

服务器发起 SSL 握手的第二阶段，并且整个阶段服务器是唯一的发起方，客户端是第二阶段唯一的消息接收方。服务器首先向客户端发送数字证书，如果没有数字证书，则向客户端发送公钥，然后请求客户端的数字证书，ServerHelloDone 消息表示服务器的 ServerHello 消息部分已经完成，发送完这个消息后，服务器等待客户端的响应。

客户端启动 SSL 握手的第三阶段，如果第二阶段服务器请求了客户端的数字证书，则客户端发送数字证书，若没有数字证书，则发送 NoCertificate 消息，由服务器决定是否继续。如果服务器需要对客户端进行认证，则完成证书验证操作。

客户端启动 SSL 握手的第四阶段，使服务器结束。客户端和服务器生成一个主秘密(master secret)，用于生成密钥和秘密，从而进行加密和 MAC（Message Authentication Code ，消息认证码)计算。

## 2.6.2 SSL 记录协议

SSL 记录协议在客户端与服务器握手成功后起作用，即客户端与服务器彼此相互认证并确定安全信息交换所使用的算法后，进入 SSL 记录协议。SSL 记录协议可以为 SSL 连接提供保密性服务和消息完整性服务。保密性服务是通过握手协议建立的共享密钥来实现，消息完整性服务是通过握手协议建立的 MAC 来实现。

SSL 记录协议的过程如图 2-10 所示：

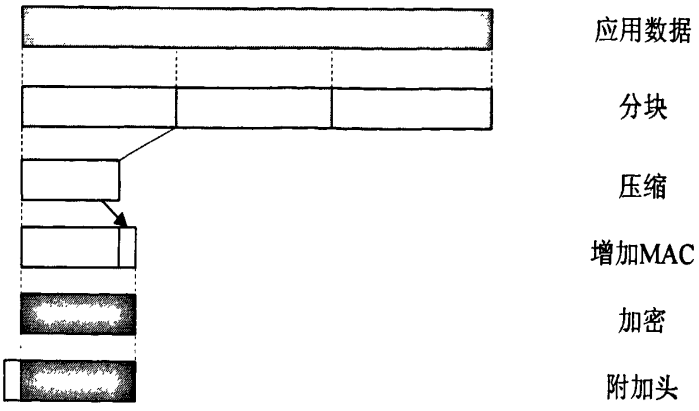


图 2-10 SSL 记录协议流程
Fig 2-10 SSL record protocol process

SSL 记录协议以应用数据作为输入，将需要发送的数据分为可供处理的数据块，并且对这些数据进行压缩、加密，然后附加头并传递给传输层，像其他 TCP 块一样经过 TCP 协议处理。作为数据接收方，则删除每个数据块的头，解密、解压缩，然后汇编成应用消息。

### 2.6.3 SSL 报警协议

SSL 报警协议是用来为对等实体传递 SSL 的相关警告，包括警告的严重程度以及该警告的描述等。如果在通信过程中某一方发现任何异常,就需要给另一方发送一条报警消息，报警消息有两种：一种是致命错误，例如在传递数据过程中发现错误的 MAC，此时双方需要立即中断会话，同时消除自己缓冲区相应的会话记录并使相应的会话标识符失效；第二种是非致命警报，此种情况下，通信过程通常不会造成任何影响，通信双方都只记录日志。

### 2.6.4 SSL 修改密钥协议

SSL 修改密钥协议时为了使密码策略能得到及时通知而存在的协议，协议由单个消息组成,该消息只包含一个值为 1 的单个字节，表明传输过程使用当前的加密约定来加密和压缩而不是改变后的加密约定。为了保障 SSL 传输过程的安全性,双方应该每隔一段时间改变加密规范。

### 2.6.5 OpenSSL

OpenSSL 是一个实现了 SSL 协议及其相关加密技术的开源软件包，OpenSSL 库中包含了完整的加密算法、数字签名算法及证书算法等。通过功能强大的加密算法来实现传输层的安全性[33]。OpenSSL 包含一套 SSL 协议的完整接口，应用程序通过这些接口可以很方便地建立起安全套接层，进而能够通过网络进行安全的数据传输，可以很好地保证数据的保密性和完整性。OpenSSL 软件包可以分为三个主要的功能部分：密码算法库、SSL 协议库以及 OpenSSL 命令行工具程序。

OpenSSL 主要提供以下功能[34]：

- 各类密钥以及密钥参数的生成和格式转换；
- 使用各种加密算法进行数据加密；
- 证书请求、证书生成和签发以及其他相关标准的转换；
- 消息摘要算法及其相关编码的实现；
- SSL 服务器和 SSL 客户端安全通信的实现等。

## 2.7 本章小结

本章主要介绍了信息系统安全、CORBA、SSL、OpenSSL、数字证书、数字签名、身份认证、LDAP 以及 PKI/CA 等技术的相关概念以及分布式系统面临的安全威胁和所应具备的安全机制。

# 第3章 安全架构总体设计

CORBA 由于支持异构分布式应用程序间的互操作性以及支持独立于编程语言和平台的对象重用，并且可以用于解决面向对象的异构应用之间的互操作问题，因而越来越多的应用到了信息系统和 Web 系统中。CORBA 系统由于自身的分布式特性，因而在安全方面会遇到比普通信息系统更多的安全威胁，如系统授权用户访问到未授权信息、安全控制被绕过、机密数据被窃取、通信数据被篡改以及缺少足够的身份认证等。因此，CORBA 系统的安全架构需要从多方面进行考虑。

## 3.1 安全架构分析

### 3.1.1 身份认证

身份认证是任何加密方案的第一步，因为只有知道对方是谁，通信加密才有意义。通信加密的目的是保护双方之间的通信，假如不知道对方是谁，保护这个通信则变得毫无意义。身份认证是实现网络安全的重要机制之一，身份认证是保证系统信息资源和操作被合法使用的关键[35][36]。在安全的网络通信中，涉及通信的各方必须通过某种形式的身份验证机制来证明自己的身份，同时还要验证与之通信的对方身份是否与所宣称的一致，只有双方身份确认后才可以进行后续的通信。因此，系统总体安全架构中需要有相应的模块用于处理身份认证。鉴于数字证书是目前完成身份认证的最为安全有效的技术手段之一，系统将引入 PKI /CA 机制，使用数字证书来进行身份认证。

### 3.1.2 访问控制

在确定用户的身份后，需要进行访问控制，来防止用户访问到未授权的信息。访问控制是指用一系列的方法和手段，以限制活动实体操作计算机中对象的能力，或限制其在网络上接收消息或访问服务的能力。访问控制的目标是，通过禁止未授权方读取信息、修改内容或消耗资源，来维护系统的保密性、完整性和可用性[21]。因此，系统总体安全架构中需要有相应的模块用于进行访问控制。

### 3.1.3 安全通信

在确认用户身份和权限后，通信过程中，如果是未加密的数据，则会面临数据丢失、数据重复或数据传送的自身错误，而且会遭遇信息攻击或欺诈行为，导致最终信息收发的差异性，因此，在信息传输过程中，还需要确保发送和接收的信息内容的一致性，保证信息接收结果的完整。因此，系统总体安全架构中需要有相应的模块用于保障通信安全。

## 3.2 安全架构总体设计

经过上述的讨论，最终设计系统安全总体架构如图 3-1 所示：

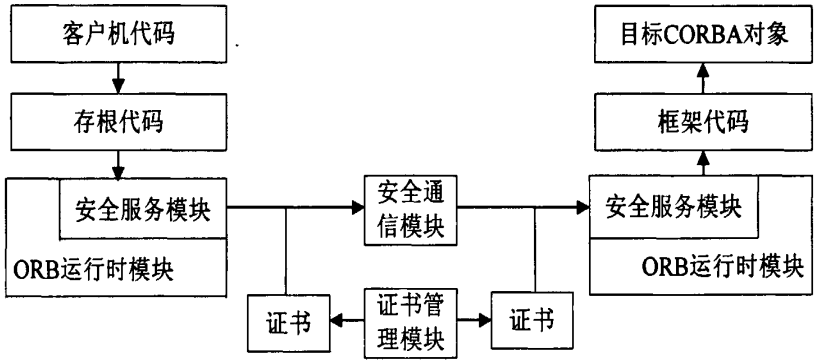


图 3-1 系统安全架构
Fig 3-1 System security architecture

系统安全总体架构中，安全服务模块负责身份认证和访问控制，安全通信模块负责信息安全完整传输，证书管理模块负责证书的签发和管理。

## 3.3 证书管理模块

CORBA 安全服务规范中身份认证部分规定，在安全的调用实施访问控制之前，CORBA 系统必须对客户身份进行验证。在分布式 CORBA 系统中引入 PKI 技术可以很方便得进行证书管理并完成身份认证的工作。

证书管理模块结构如图 3-2 所示：

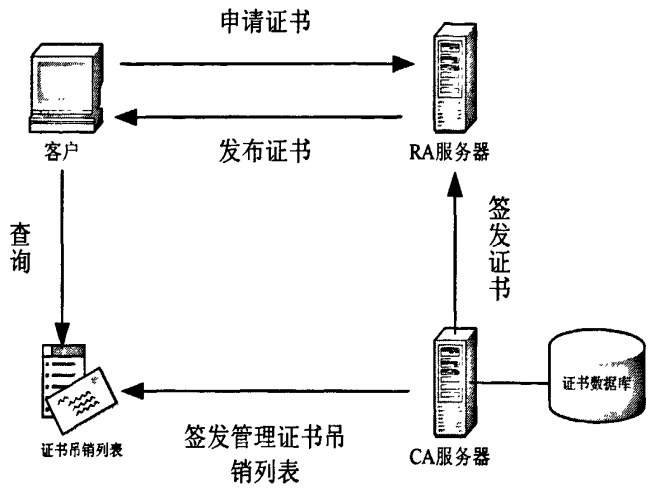


图 3-2 证书管理模块
Fig 3-2 Certificate management module

其中，CA 服务器负责完成数字证书和证书吊销列表的签发和管理，RA 服务器负责完成对证书用户的管理，并作为与用户交互的平台，完成数字证书的最终制作和发放。CA 服务器在签发数字证书后，需要将证书存入证书数据库备份以便于以后的查询和管理。客户直接与 RA 服务器交互来申请证书，客户可以查询证书吊销列表来判断需要验证的证书是否已经被吊销。

CA 服务器需要能够完成的具体任务如下:

- 密钥的生成和管理。
- 证书的生成和管理。
- 证书吊销列表的生成和维护。
- 密钥、证书和用户信息数据的存储、备份和恢复。
- 日志记录。

RA 服务器需要能够完成的具体任务如下:

- 用户申请的受理，用户信息的录入、核对。
- 证书信息的发布。
- 证书请求的删除。
- 日志记录。
- LDAP 目录服务。

证书管理模块工作流程如下：

1. 使用 RSA 算法建立非对称密钥，用于 CA 服务器；
2. 使用 CA 私钥对 CA 服务器相关信息进行签名，生成 CA 根证书；
3. 建立证书库，用于存放 CA 服务器所签发的证书；
4. 建立证书吊销列表；

5. 用户向 RA 服务器提交证书申请；
6. RA 服务器审核用户的证书申请；
7. RA 服务器审核用户证书申请通过后，交由 CA 服务器审核；
8. CA 服务器审核通过后，用自己的私钥对证书申请进行签名，生成一张符合 X.509 规范的证书，并在证书库中进行备份，生成证书号；
9. RA 服务器将 CA 服务器生成的证书发送给申请的用户，此证书将成为该用户在系统中身份的唯一标识。

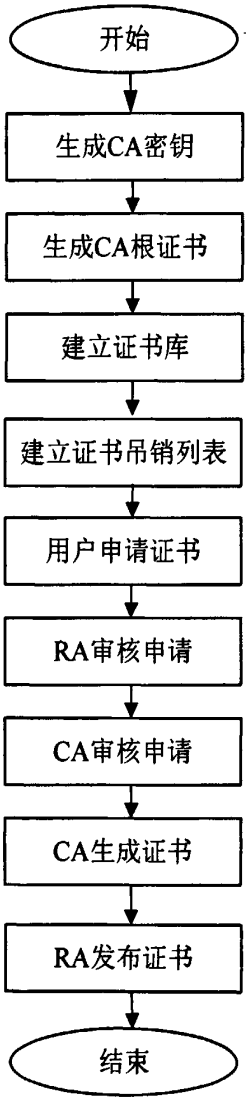


图 3-3 证书管理模块工作流程

Fig 3-3 Certificate management module working process

## 3.4 安全服务模块

### 3.4.1 身份认证

在对一个目标对象进行安全调用和访问控制之前，CORBA 系统必须对客户和目标对象进行身份认证，这一过程通过验证客户和目标对象的数字证书来实现[37][38]。认证证书包含两个方面的内容一是认证证书的真实性，二是认证证书确实为发送者所拥有。OpenSSL 在握手期间会对客户的数字证书进行验证，包括证书的签发实体、证书的有效时间和信任设置以及证书的吊销状态等，因此，将采用 OpenSSL 提供的功能来实现身份认证，具体过程如下：

首先，双方要建立连接，具体过程如下：

1. 双方建立 TCP 连接；
2. 绑定 SSL 套接字；
3. 双方建立 SSL 连接。

然后，双方进行身份认证，具体过程如下：

1. 获得对方的证书；
2. 验证证书的真实性；
3. 验证证书身份，获得证书所有者和证书颁发者信息从而验证证书的身份。

### 3.4.2 访问控制

在主体身份认证完成后，需要进行访问控制来判断主体对于其所请求的资源或对象是否有访问权限。一般情况下，实施访问控制的系统需要包含以下三种主要对象[39]：

1. 主体：试图去访问对象信息的主动者，例如用户或代理；
2. 客体：被访问的对象信息。例如系统中某一对象或对象的方法等，需要对其访问实施有效的控制；
3. 访问控制策略：一套规则，以确定主体是否对客体拥有某些操作权限。

访问控制模块需要实现三个功能：

1. 授权：用于指定访问控制策略，将客体的操作权限赋予主体，可以分为直接授权和间接授权。直接授权是指指定的访问控制策略直接将客体的访问权限分配给主体。间接授权是指指定的访问控制策略并不直接将客体的访问权限分配给主体，而是通过一个中间件来实现权限的分配。目前，间接授权主要以基于角色的访问控制模型为典型；
2. 权限回收：用于将指定权限从主体收回，防止该主体再拥有该权限从而进行非法访问。与授权相似，权限回收也分为直接权限回收和间接权限

回收；

3. 访问检查：授权在系统中的实施模块。通过访问检查与授权的配合来达到访问控制的目的。

访问控制模型如图 3-4 所示：

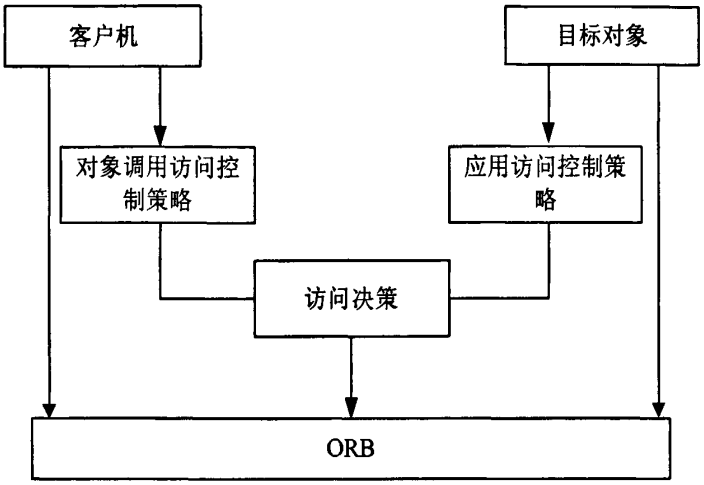


图 3-4 访问控制模型
Fig 3-4 Access control model

在上述访问控制模型中，对象调用控制策略在客户机调用目标对象时通过截获对象调用来实现；应用访问控制策略则是保证在客户和目标对象的实现过程中的访问控制，该策略能够控制对对象的内部函数和数据的访问，还可以进一步使用对象调用访问策略。

对象调用访问控制策略中定义了允许客户访问目标对象上指定操作的条件，包括当前客户主体的特权属性(主体的身份、角色、组别，主体的认证信息，主体被允许的相关操作)、访问控制特性（主体的有效访问时间）以及需要进行的相关操作等[40]。

应用访问控制策略则用来控制哪些主体可以对目标对象进行相应的访问操作，通过应用访问控制策略维护一个访问控制列表，用来判断允许进行访问操作的主体的身份及其访问权限。

访问决策则是用来判断客户机在当前访问控制策略下是否有权限访问请求的目标对象的相关操作。访问决策模型如图 3-5 所示：

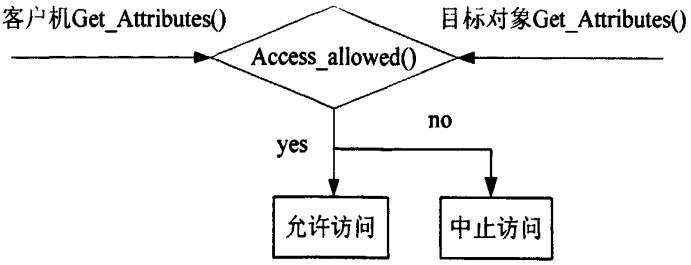


图 3-5 访问决策模型
Fig 3-5 Access decision-making model

访问决策通过调用 Get_Attributes()方法来获得判定所需要的参数，包括客户主机的身份、权限以及目标对象的名称、相关操作和访问控制列表等。获得相应的参数后，访问决策调用 Access_allowed()方法进行判断，并最终决定对目标对象的访问是允许还是拒绝。

## 3.5 安全通信模块

在主体认证和访问控制结束以后，需要保障客户端和服务器通信数据的保密性和完整性。数据的保密性方面，目前主要是采用加密算法来对传送的数据进行加密，数据完整性方面，主要是采用数字签名技术来保证数据的完整性。

在现代密码学中，主要有两类算法：对称密码算法和非对称密码算法[41]。对称密码算法又称传统密码算法或者单密钥算法，即加密密钥和解密密钥可相互推导，且在大多数的对称算法中，加密密钥和解密密钥是相同的。对称算法要求发送者和接收者在进行安全通信前，首先商定一个密钥，且对称密码算法的安全性依赖于密钥，泄漏密钥就意味着通信保密性的丧失;所以只要进行保密通信，就必须保密密钥。非对称密码算法也称为公开密钥算法，用于加密的密钥和用于解密的密钥不同，且解密密钥不能由加密密钥在合理的时间内计算而得出[42]。加密密钥可以公开，所以也称为公钥，解密密钥由用户自己妥善保管。陌生人只能用加密密钥对消息进行加密，但只有拥有解密密钥的人才能够解密该密文信息。

数字签名技术是在数据加密基础上的延伸应用，建立在公钥体制基础之上[43]。它的主要方式是，报文的发送方从报文文本中生成一个散列值，发送方用自己的私钥对这个散列值进行加密来形成发送方的数字签名。然后，这个数字签名将作为报文的附件和报文一起发送给报文的接收方。报文的接收方首先从接收到的原始报文中计算出散列值，接着再用发送方的公钥来对报文附加的数字签名进行解密。如果两个散列值相同，那么接收方就能确认该数字签名是发送方的，通过使用数字签名技术能够证实数据的来源及其完整性，同时对数据进行保护。

在本系统安全架构中，使用 SSL 技术来保证数据的机密性和完整性。SSL

对于数据机密性的保证是通过使用 SSL 握手协议定义的秘密密钥来实现的，而 SSL 对于数据完整性的保证是通过握手协议定义的 MAC 来实现的。SSL 的秘密密钥和共享秘密密钥的生成发生在 SSL 握手的第四阶段。在记录进行安全加密或完整性检查之前，客户机和服务器需要生成只有他们自己知道的共享秘密消息，长度为 48 个字节，称为主秘密，并通过主秘密来生成密钥和秘密，用于加密和 MAC 计算。主秘密的生成方法如图 3-6 所示：

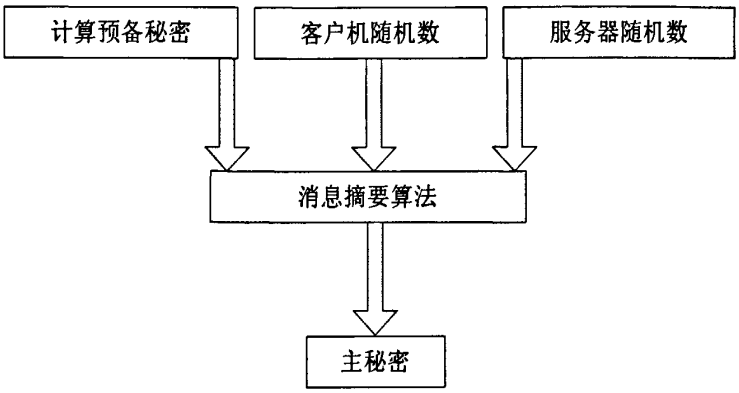


图 3-6 主秘密生成流程

Fig 3-6 Main secret generating process

计算主秘密的技术规范如图 3-7 所示：

$$Master\_secret = \begin{cases} MD5(pre\_master\_secret + SHA(A + pre\_master\_secret \\ +ClientHello.random + ServerHello.random)) + \\ MD5(pre\_master\_secret + SHA(BB + pre\_master\_secret \\ +ClientHello.random + ServerHello.random)) + \\ MD5(pre\_master\_secret + SHA(CCC + pre\_master\_secret \\ +ClientHello.random + ServerHello.random)) \end{cases}$$

图 3-7 主秘密计算技术规范

Fig 3-7 Main secret calculating technical specification

然后，生成客户机和服务器使用的用于数据加密的对称密钥，对称密钥生成方法如图 3-8 所示：

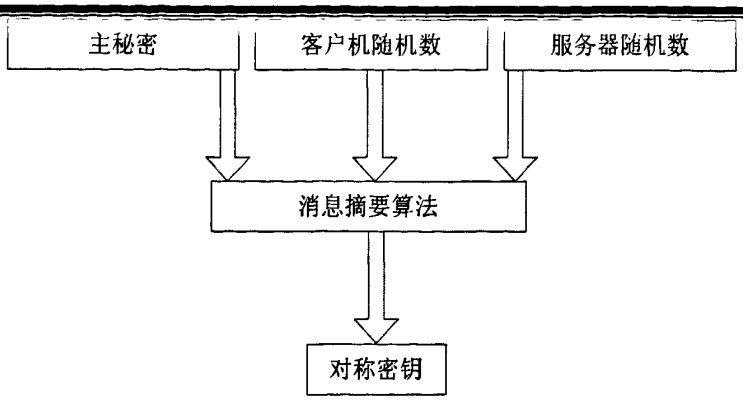


图 3-8 对称密钥生成流程
Fig 3-8 Symmetric key generating process

实际密钥生成公式如图 3-9 所示：

$$key\_block = \begin{cases} MD5(master\_secret + SHA(A + pre\_master\_secret \\ +ClientHello.random + ServerHello.random)) + \\ MD5(master\_secret + SHA(BB + pre\_master\_secret \\ +ClientHello.random + ServerHello.random)) + \\ MD5(master\_secret + SHA(CCC + pre\_master\_secret \\ +ClientHello.random + ServerHello.random)) \end{cases}$$

图 3-9 密钥生成公式
Fig 3-9 Key generating formula

MAC 的生成方式与对称密钥类似，对称密钥与MAC生成后，安全通信过程中，将使用对称密钥来保证消息的保密性，使用MAC来保证消息的完整性。

## 3.6 本章小结

本章在分析了 CORBA 分布式系统安全需求的基础上，提出了基于 CORBA 的分布式系统安全总体架构，包括证书管理模块、安全服务模块和安全通信模块，主要等实现了证书管理、身份认证、访问控制、数据加密和数据完整性保障等安全功能。

# 第4章 安全架构详细设计

本章将对系统安全架构的三个模块：证书管理模块、安全服务模块和安全通信模块进行详细设计，其中，证书管理功能主要基于 OpenSSL 提供的 API，安全通信模块和安全服务模块的身份认证部分主要基于 SSL 提供的 API，而安全服务模块的访问控制部分主要基于 CORBA 安全服务规范中的访问控制模型。

## 4.1 证书管理模块

### 4.1.1 证书文件生成

证书文件生成主要是使用 OpenSSL 提供的 API 来完成的，首先在 OpenSSL 安装目录下建立 CA 文件夹作为根目录，在 CA 文件夹下分别建立 certs、newcerts、crl 和 private 四个文件夹，并将 OpenSSL 自带的 openssl.cnf 文件拷贝到 CA 文件夹中，设置环境变量 OPENSSL_CONF 为 openssl.cnf 的路径，在 CA 文件夹中建立随机数文件，命令如下：

openssl rand –out private/.rand 1000

生成文本数据库文件，命令如下：

touch index.txt

生成证书序列号文件，命令如下：

Echo “01”>ca.srl

然后，利用 OpenSSL 提供的 API 制作证书文件，证书文件主要分为三类：CA 服务器自签名的根 CA 证书、服务器证书和客户端证书。

1. 制作 CA 服务器自签名的根 CA 证书以及私有密钥文件

1) 生成 CA 服务器的私有密钥文件。

在 CA 根目录下，输入以下命令：

openssl genrsa –des3 –out private/ca.key 2048

其中，des3 表示对私钥加密的算法，out 表示输出到文件，2048 表示私钥长度。

提示输入私有密钥文件密码：

Enter pass phrase for private/ca.key：

输入密码后，提示再一次输入密码：

Verifying – Enter pass phrase for private/ca.key：

输入完成后，将会在 CA 文件夹下创建 ca.key 私有密钥文件。

整个过程如图 4-1 所示：

```
root@gezunmin-desktop:~# cd /usr/local/ssl/CA
root@gezunmin-desktop:/usr/local/ssl/CA# openssl genrsa -des3 -out private/ca.ke
y 2048
Generating RSA private key, 2048 bit long modulus
..................+++
..............................+++
e is 65537 (0x10001)
Enter pass phrase for private/ca.key:
Verifying - Enter pass phrase for private/ca.key:
root@gezunmin-desktop:/usr/local/ssl/CA#
```

图 4-1 CA 服务器私有密钥生成过程

Fig 4-1 CA server private key generating process

2) 生成 CA 服务器的自签名的根 CA 证书

在 CA 根目录下，输入以下命令:

openssl req –new –x509 –days 1000 –key private/ca.key –out certs/ca.crt

其中，day 表示有效期时间的天数，key 表示采用的签名私钥。

提示输入 CA 服务器私有密钥文件的密码:

Enter pass phrase for private/ca.key:

输入密码后，提示输入证书的详细信息:

Country Name (2 letter code) [AU]：输入 CH，代表中国

State or Province Name (full name) [Some-State]：输入 Beijing，代表北京

Locality Name (eg. city) []：输入 Beijing，代表北京

Organization Name (eg. company) [Internet Widgits Pty Ltd]：输入 BJUT，代表北京工业大学

Organizational Unit Name (eg. section) []：输入 Computer Institute，代表计算机学院

Common Name (eg. YOUR name) []：输入 CA，代表证书持有者为 CA 自身

Emal Address []：输入 Email 地址

整个过程如图 4-2 所示：

```
root@gezunmin-desktop:~# cd /usr/local/ssl/CA
root@gezunmin-desktop:/usr/local/ssl/CA# openssl req -new -x509 -days 1000 -key
private/ca.key -out certs/ca.crt
Enter pass phrase for private/ca.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CH
State or Province Name (full name) [Some-State]:Beijing
Locality Name (eg, city) []:Beijing
Organization Name (eg, company) [Internet Widgits Pty Ltd]:BJUT
Organizational Unit Name (eg, section) []:Computer Institute
Common Name (eg, YOUR name) []:CA
Email Address []:allanwxm@emails.bjut.edu.cn
root@gezunmin-desktop:/usr/local/ssl/CA#
```

图 4-2 CA 服务器根证书生成过程
Fig 4-2 CA server root certificate generating process

2. 制作服务器的密钥文件和数字证书

1) 生成服务器的私有密钥文件，步骤与 CA 服务器私有密钥文件制作过程类似，在 CA 根目录下，输入以下命令:

openssl genrsa –des3 –out private/server.key 1024

整个过程如图 4-3 所示:

```
root@gezunmin-desktop:~# cd /usr/local/ssl/CA
root@gezunmin-desktop:/usr/local/ssl/CA# openssl genrsa -des3 -out private/serve
r.key 1024
Generating RSA private key, 1024 bit long modulus
.....................++++++
...++++++
e is 65537 (0x10001)
Enter pass phrase for private/server.key:
Verifying - Enter pass phrase for private/server.key:
root@gezunmin-desktop:/usr/local/ssl/CA#
```

图 4-3 服务器私有密钥文件生成过程
Fig 4-3 Server private key file generating process

2) 生成请求 CA 服务器签名的文件 server.req，步骤与 CA 服务器自签名证书的生成过程类似，在 CA 根目录下，输入以下命令:

openssl req –new –key private/server.key –out certs/server.csr

整个过程如图 4-4 所示:

```
root@gezunmin-desktop:~# cd /usr/local/ssl/CA
root@gezunmin-desktop:/usr/local/ssl/CA# openssl req -new -key private/server.ke
y -out certs/server.csr
Enter pass phrase for private/server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CH
State or Province Name (full name) [Some-State]:Beijing
Locality Name (eg, city) []:Beijing
Organization Name (eg, company) [Internet Widgits Pty Ltd]:BJUT
Organizational Unit Name (eg, section) []:Computer Institute
Common Name (eg, YOUR name) []:Server
Email Address []:allanwxm@emails.bjut.edu.cn

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:1234567890
An optional company name []:
root@gezunmin-desktop:/usr/local/ssl/CA#
```

图 4-4 服务器签名文件生成过程

Fig 4-4 Server signature file generating process

3) 生成 CA 服务器签名的服务器证书 server.crt，在 CA 根目录下，输入以下命令:

openssl x509 –req –days 1000 –CA certs/ca.crt –CAkey private/ca.key –in certs/server.csr–out certs/server.crt

整个过程如图 4-5 所示:

```
root@gezunmin-desktop:~# cd /usr/local/ssl/CA
root@gezunmin-desktop:/usr/local/ssl/CA# openssl x509 -req -days 1000 -CA certs/
ca.crt -CAkey private/ca.key -in certs/server.csr -out certs/server.crt
Signature ok
subject=/C=CH/ST=Beijing/L=Beijing/O=BJUT/OU=Computer Institute/CN=Server/emailA
ddress=allanwxm@emails.bjut.edu.cn
Getting CA Private Key
Enter pass phrase for private/ca.key:
root@gezunmin-desktop:/usr/local/ssl/CA#
```

图 4-5 服务器证书生成过程

Fig 4-5 Server certificate generating process

3. 制作客户端的密钥文件和数字证书

执行过程中，可以由用户端发出申请或者由 CA 服务器根据客户端的个数来制作相应的密钥文件和数字证书。

1) 生成客户端的私有密钥文件

过程与 CA 服务器私有密钥文件生成类似，在 CA 根目录下，输入以下命令:

openssl genrsa –des3 –out private /client1.key 1024

整个过程如图 4-6 所示：

```
root@gezunmin-desktop:~# cd /usr/local/ssl/CA
root@gezunmin-desktop:/usr/local/ssl/CA# openssl genrsa -des3 -out private/clien
t1.key 1024
Generating RSA private key, 1024 bit long modulus
.................................++++++
.........++++++
e is 65537 (0x10001)
Enter pass phrase for private/client1.key:
Verifying - Enter pass phrase for private/client1.key:
root@gezunmin-desktop:/usr/local/ssl/CA#
```

图 4-6 客户端私有密钥文件生成过程

Fig 4-6 Client private key file generating process

2) 生成请求 CA 服务器签名的文件

在 CA 根目录下，输入以下命令:

openssl req –new –key private/client1.key –out certs/client1.csr

整个过程如图 4-7 所示：

```
root@gezunmin-desktop:~# cd /usr/local/ssl/CA
root@gezunmin-desktop:/usr/local/ssl/CA# openssl req -new -key private/client1.k
ey -out certs/client1.csr
Enter pass phrase for private/client1.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CH
State or Province Name (full name) [Some-State]:Beijing
Locality Name (eg, city) []:Beijing
Organization Name (eg, company) [Internet Widgits Pty Ltd]:BJUT
Organizational Unit Name (eg, section) []:Computer Institute
Common Name (eg, YOUR name) []:Client1
Email Address []:allanwxm@emails.bjut.edu.cn

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:1231567890
An optional company name []:
root@gezunmin-desktop:/usr/local/ssl/CA#
```

图 4-7 客户端签名文件生成过程

Fig 4-7 Client signature file generating process

3) 生成 CA 服务器签名的客户端证书

在 CA 根目录下，输入以下命令:

openssl x509 –req –days 1000 –CA certs/ca.crt –CAkey private/ca.key –in certs/client1.csr –out certs/client1.crt

整个过程如图 4-8 所示：

```
root@gezunmin-desktop:~# cd /usr/local/ssl/CA
root@gezunmin-desktop:/usr/local/ssl/CA# openssl x509 -req -days 1000 -CA certs/
ca.crt -CAkey private/ca.key -in certs/client1.csr -out certs/client1.crt
Signature ok
subject=/C=CN/ST=Beijing/L=Beijing/O=BJUT/OU=Computer Institute/CN=Client1/email
Address=allanwxm@emails.bjut.edu.cn
Getting CA Private Key
Enter pass phrase for private/ca.key:
root@gezunmin-desktop:/usr/local/ssl/CA#
```

图 4-8 客户端证书生成过程

Fig 4-8 Client certificate generating process

最终生成的客户端证书文件 client1.crt 内容如下：

-----BEGIN CERTIFICATE-----

MIIDIzCCAgsCAQMwDQYJKoZIhvcNAQEFBQAwgZYxCzAJBgNVBAYTA
kNlMRAwDgYD

VQQIDAdCZWlqaW5nMRAwDgYDVQQHDAdCZWlqaW5nMQ0wCwYDV
QQKDARCSlVUMRswGQYDVQQLDBJDb21wdXRlciBJbnN0aXR1dGUxCzAJBg
NVBAMMAkNBMSowKAYJKoZIhvcNAQkBFhthbGxhbnd4bUBlbWFpbHMuYmp
1dC5lZHUuY24wHhcNMTAwNDE1MTQwMzMzWhcNMTMwMTA5MTQwMzM
zWjCBmzELMAkGA1UEBhMCQ0gxEDAOBgNVBAgMB0JlaWppbmcxEDAOBg
NVBAcMB0JlaWppbmcxDTALBgNVBAoMBEJKVVQxGzAZBgNVBAsMEkNvb
XB1dGVyIEluc3RpdHV0ZTEQMA4GA1UEAwwHQ2xpZW50MTEqMCgGCSqGS
Ib3DQEJARYbYWxsYW53eG1AZW1haWxzLmJqdXQuZWR1LmNuMIGfMA0G
CSqGSIb3DQEBAQUAA4GNADCBiQKBgQCjperuK/UNp4UT8N4AvQmWUD/U
ccCljk5LGSM4iNRPFR1yT1zhxv6DAtKgPl5QlrMVovYaal0+H+z8Qd9lSrS1Unx0/F
G7y1uoV3f97u7Qm6yUTBH/6RradEoEb42ryfaSYclhw6pDfDlocCD1mClmNBrlbnl
+by8kQWXOfUeM/wIDAQABMA0GCSqGSIb3DQEBBQUAA4IBAQCQbboii0vle
DajzPxaoslehFtvrGWAWeKz1QACFfFovyu3HhyeoVdCUg1L+1UIT4JujynniLxdH/o
vZMUE3HtFoRnIKtYc5GYDeuPexN3bp+YpIuHTg8DRJD5kr/s1nl911/Kfgpah3KO
AAG5e10a+Od62k78odraEtPdCalcqOOLMiibNpTi057VSmrD3wB/YRB+03m3hN4
gMaEKI3+fXXntOQN9WMwvBtTPhtgJkXssx0e3i7zBys1y2OJchTnKdiSYt2DmQcl
ewj9uyX+3uJHlnZ2dTjCyUH0jugwKvTs7CcEM08DC2tz9+iK3UK32G4aT0pJpPbj
Gx2J69MJVq

-----END CERTIFICATE-----

使用如下命令查看生成的客户端证书的详细信息：

openssl x509 –in certs/client1.crt –text –noout

显示内容如下：

Certificate:

Data:

Version: 1 (0x0)

Serial Number: 3 (0x3)

Signature Algorithm: sha1WithRSAEncryption

Issuer: C=CH, ST=Beijing, L=Beijing, O=BJUT, OU=Computer Institute, CN=C A/emailAddress=allanwxm@emails.bjut.edu.cn

Validity

Not Before: Apr 15 14:03:33 2010 GMT

Not After : Jan 9 14:03:33 2013 GMT

Subject: C=CH, ST=Beijing, L=Beijing, O=BJUT, OU=Computer Institute, CN= Client1/emailAddress=allanwxm@emails.bjut.edu.cn

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

Public-Key: (1024 bit)

Modulus:

00:a3:a5:ea:ee:2b:f5:0d:a7:85:13:f0:de:00:bd:
09:96:50:3f:d4:71:c0:a5:8e:4e:4b:19:23:38:88:
d4:4f:15:1d:72:4f:5c:e1:c6:fe:83:02:d2:a0:3e:
5e:50:96:b3:15:a2:f6:1a:6a:5d:3e:1f:ec:fc:41:
df:65:4a:b4:b5:52:7c:74:fc:51:bb:cb:5b:a8:57:
77:fd:ee:ee:d0:9b:ac:94:4c:11:ff:e9:1a:da:74:
4a:04:6f:8d:ab:c9:f6:92:61:c9:61:c3:aa:43:7c:
39:68:70:20:f5:98:29:66:34:1a:e5:6e:79:7e:6f:
2f:24:41:65:ce:7d:47:8c:ff

Exponent: 65537 (0x10001)

Signature Algorithm: sha1WithRSAEncryption

90:6d:ba:22:8b:4b:e5:78:36:a3:cc:fc:5a:a2:c9:5e:84:5b:
6f:ac:65:80:59:e2:b3:d5:00:02:15:f1:68:bf:2b:b7:1e:1c:
9e:a1:57:42:52:0d:4b:fb:55:08:4f:82:6e:8f:29:e7:88:bc:
5d:1f:fa:2f:64:c5:04:dc:7b:45:a1:19:c8:2a:d6:1c:e4:66:
03:7a:e3:de:c4:dd:db:a7:e6:29:22:e1:d3:83:c0:d1:24:3e:
64:af:fb:35:9e:5f:75:d7:f2:9f:82:96:a1:dc:a3:80:00:6e:
5e:d7:46:be:39:de:b6:93:bf:28:76:b6:84:b4:f7:42:6a:57:
2a:38:e2:cc:8a:26:cd:a5:38:b4:e7:b5:52:9a:b0:f7:c0:1f:
d8:44:1f:b4:de:6d:e1:37:88:0c:68:42:88:df:e7:d7:5e:7b:
4e:40:df:56:33:0b:c1:b5:33:e1:b6:02:64:5e:cb:31:d1:ed:
e2:ef:30:72:b3:5c:b6:38:97:21:4e:72:9d:89:26:2d:d8:39:
90:72:57:b0:8f:db:b2:5f:ed:ee:24:79:67:67:67:53:8c:2c:
94:1f:48:ee:83:02:af:4e:ce:c2:70:43:34:f0:30:b6:b7:3f:
7e:88:ad:d4:2b:7d:86:e1:a4:f4:a4:9a:4f:6e:31:b1:d8:9e:
bd:30:95:6a

### 4.1.2 密钥和证书管理

由于CA服务器的重要性，在完成证书生成和签发工作后，需要将CA服务器的根证书和私有密钥文件加密存储到数据库中，对于服务器和客户端的私有密钥文件和证书文件，需要存储到证书库中作为备份，以便日后恢复用户证书。证书目录的结构如图4-9所示：

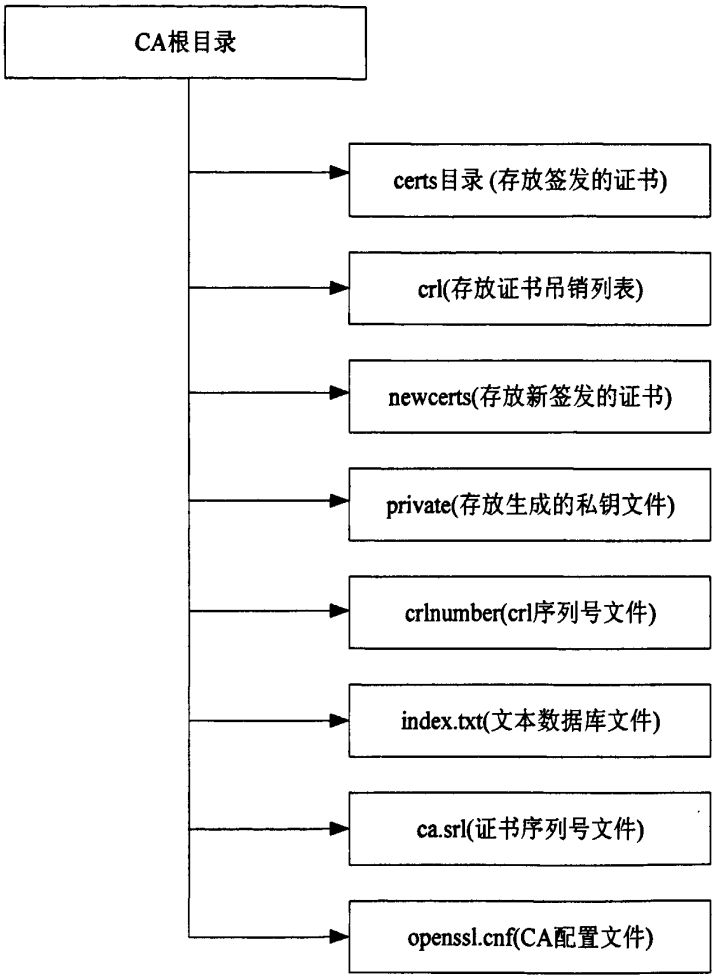


图4-9 证书目录结构
Fig 4-9 Certificate directory structure

### 4.1.3 证书吊销列表更新

证书吊销是一个十分简单的过程，只需要吊销服务器上存放的证书的备份即可。吊销证书需要用到OpenSSL中的ca命令，假设我们要吊销前面签发的客户端证书client1.crt，并签发一个client2.crt证书，然后吊销，吊销命令如下：

openssl ca –revoke certs/client1.crt

执行过程如图4-10所示：

```
root@gezunmin-desktop:~# cd /usr/local/ssl/CA
root@gezunmin-desktop:/usr/local/ssl/CA# openssl ca -revoke certs/client1.crt
Using configuration from /usr/local/ssl/openssl.cnf
Enter pass phrase for .//private/ca.key:
Adding Entry with serial number 03 to DB for /C=CH/ST=Beijing/L=Beijing/O=BJUT/O
U=Computer Institute/CN=Client1/emailAddress=allanwxm@emails.bjut.edu.cn
Revoking Certificate 03.
Data Base Updated
root@gezunmin-desktop:/usr/local/ssl/CA#
```

图 4-10 证书吊销过程

Fig 4-10 Certificate revocation process

证书吊销后，对于 client1.crt 和 client2.crt 证书文件来说，并没有任何改变，唯一的改变时 CA 的数据库文件 index.txt，该文件中多了如下信息：

R 130110072508Z 100416072927Z 04 unknown
/C=CH/ST=Beijing/L=Beijing/O=BJUT/OU=Computer
Institute/CN=Client2/emailAddress=allanwxm@emails.bjut.edu.cn
R 130109140333Z 100416073508Z 03 unknown
/C=CH/ST=Beijing/L=Beijing/O=BJUT/OU=Computer
Institute/CN=Client1/emailAddress=allanwxm@emails.bjut.edu.cn

使用命令行吊销证书后，只有 CA 本身知道该证书被吊销了，客户端和服务器都无法获得该信息，因此需要创建证书吊销列表来告知客户端和服务器哪些证书已经被吊销了。证书吊销列表有一个默认有效期限，可以在 openssl.cnf 文件中进行配置。证书吊销列表的创建命令如下：

openssl ca –gencrl –out crl/ca.crl

使用如下命令来显示创建的 ca.crl 证书吊销列表的信息：

openssl crl –in crl/ca.crl –text –noout

显示结果如图 4-11 所示：

```
root@gezunmin-desktop:~# cd /usr/local/ssl/CA
root@gezunmin-desktop:/usr/local/ssl/CA# openssl crl -in crl/ca.crl -text -noout
Certificate Revocation List (CRL):
        Version 2 (0x1)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: /C=CH/ST=Beijing/L=Beijing/O=BJUT/OU=Computer Institute/CN=CA/em
ailAddress=allanwxm@emails.bjut.edu.cn
        Last Update: Apr 16 08:04:30 2010 GMT
        Next Update: May 16 08:04:30 2010 GMT
        CRL extensions:
            X509v3 CRL Number:
                1
Revoked Certificates:
    Serial Number: 03
        Revocation Date: Apr 16 07:35:08 2010 GMT
    Serial Number: 04
        Revocation Date: Apr 16 07:29:27 2010 GMT
    Signature Algorithm: sha1WithRSAEncryption
        a9:66:4c:f4:61:e8:90:b0:85:2e:29:bb:3f:42:51:a7:cf:f8:
        ea:d4:ad:96:71:8d:57:ac:60:0e:33:74:09:b8:fc:1a:33:d2:
        94:a9:57:66:68:5b:c1:4f:40:be:08:6b:c2:b0:95:7f:81:00:
        ce:ce:65:6e:0c:9a:4b:b6:56:01:14:8d:74:c9:f2:6d:6e:83:
        b7:f2:69:6f:20:6d:29:72:37:a3:81:e2:eb:9c:ba:2b:0a:54:
        c0:1b:0d:a3:0b:9d:b6:a7:a4:e8:21:f7:ea:eb:13:df:48:fe:
        5b:bf:c7:61:ca:53:a2:7f:bb:79:b2:ea:c1:ad:27:10:3d:12:
        4d:08:fe:74:47:ad:fd:d9:26:6a:ef:ae:ad:f5:1b:67:3b:da:
        5e:97:87:77:4a:bb:21:4d:9c:5b:e6:ca:f9:a1:ab:fe:37:fc:
        e8:ba:16:47:31:12:d3:69:dd:d4:79:92:3b:38:6f:8b:c3:d1:
        93:47:64:8c:c8:98:05:31:a6:99:08:aa:4a:22:68:a9:e3:f8:
        18:59:c6:da:2c:a1:30:a7:6a:af:b7:79:90:a8:cb:c1:e6:a0:
        7c:55:51:d9:5d:ee:0c:9d:e2:f6:5b:13:08:de:28:cb:de:73:
        58:2a:71:3a:24:eb:11:77:60:12:df:d7:c1:8f:b5:0f:c2:ba:
        75:25:c1:8c
root@gezunmin-desktop:/usr/local/ssl/CA#
```

图 4-11 证书吊销列表生成过程
Fig 4-11 CRL generating process

### 4.1.4 LDAP 目录服务

轻量级目录访问协议 LDAP 是一种基于 X.500 目录的跨平台的目录访问服务，使用 LDAP 协议构建证书数据库可以大大简化证书的查询。LDAP 协议的一般模型是客户服务器模型客户服务器模型，在该模型中，由客户发起对目录服务器的协议操作并向目录服务器传输描述操作的协议请求；目录服务器在收到客户的操作请求后，在目录中执行必要的操作；服务器完成相应的操作后将回送结果或错误响应到请求的客户。LDAP 客户和 LDAP 服务器的一般交互过程如下：

1. 绑定：LDAP 客户与 LDAP 服务器之间建立会话，使客户绑定到服务器，客户需要制定 LDAP 服务器的 IP 地址以及监听的端口号；
2. 操作请求：LDAP 客户向 LDAP 服务器发出操作请求；
3. 操作响应：LDAP 服务器响应客户的操作请求，执行相应的操作并将结果返回给 LDAP 客户

4. 解除绑定：LDAP 客户完整相应的操作请求并得到 LDAP 服务器回送的结果后关闭与服务器之间的会话。

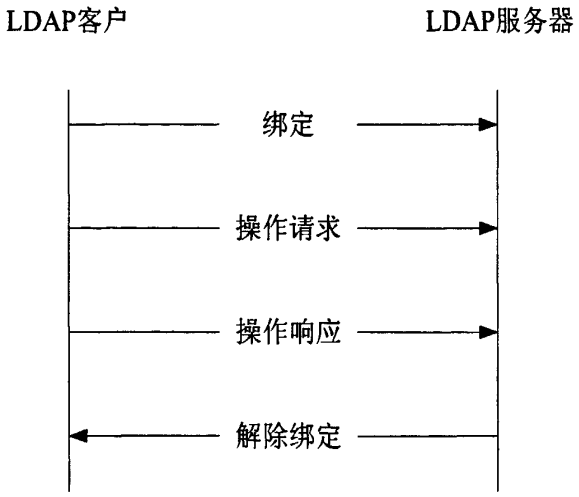


图 4-12 LDAP 客户端服务器交互过程
Fig 4-12 LDAP client and server interaction process

系统安全架构中，采用 LDAP 协议来完成证书存储和证书吊销列表的存储。由于 LDAP 目录以树状的层次结构来存储数据，因而需要根据安全架构的需求，设计 LDAP 的目录树。目录树第 0 层为为 dc=bjut.edu.cn，目录树的第 1 层按系统需求分为证书信息项 ou=certificate 和证书吊销列表信息项 ou=crl，证书信息项用于管理系统中的不同类型的证书，证书信息项下又分为服务器证书项 ou=server、客户机证书项 ou=client 以及 CA 服务器证书项 ou=CA，证书吊销列表信息项用于管理系统中的证书吊销列表。最终形成的目录树 DIT 的示例如图 4-13 所示：

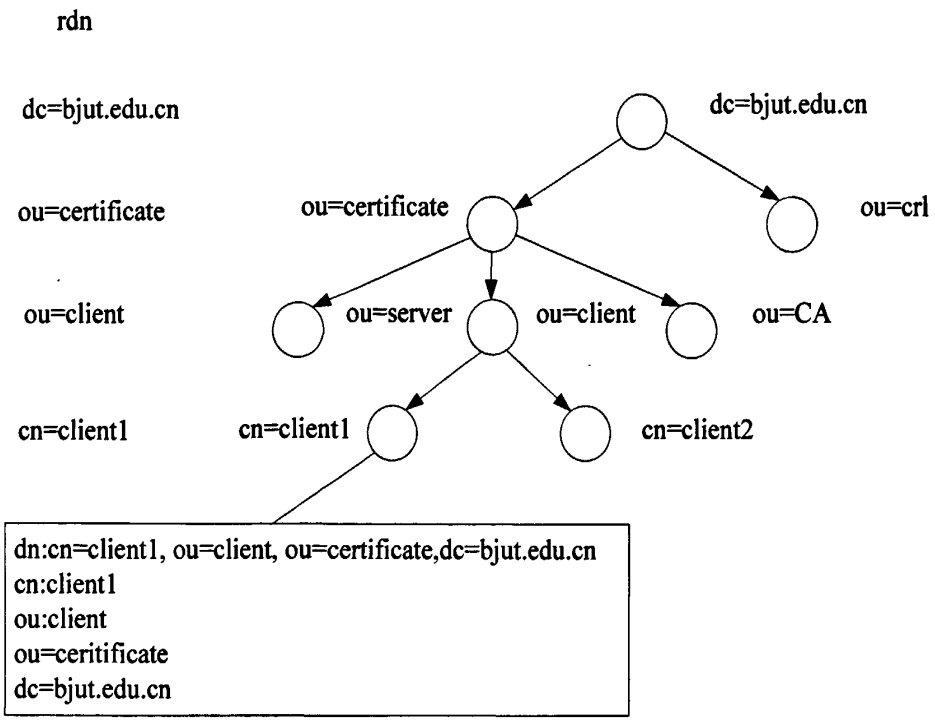


图 4-13 DIT 结构
Fig 4-13 DIT structure

## 4.2 身份认证模块

身份认证主要依靠 OpenSSL 的 API 来完成，建立连接过程如图 4-14 所示：

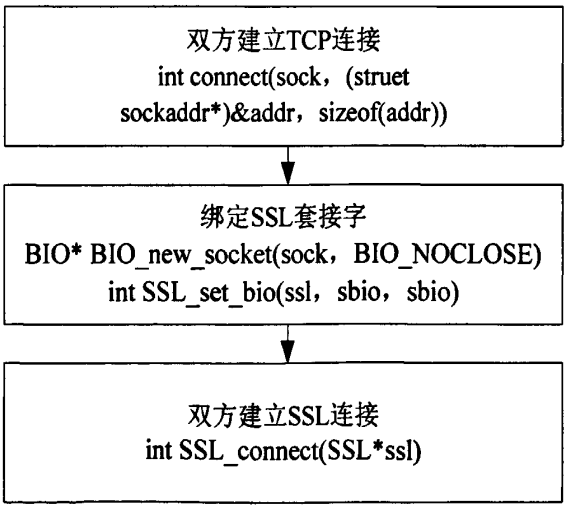


图 4-14 建立连接过程
Fig 4-14 Connection process

在双方建立连接以后，开始进行身份认证，过程如图 4-15 所示：

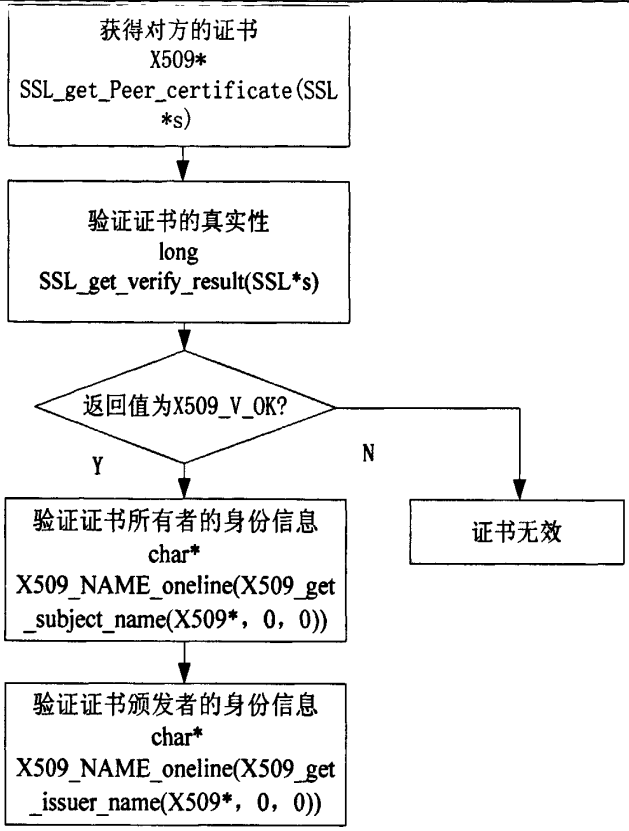


图 4-15 身份认证过程
Fig 4-15 Authentication process

## 4.3 访问控制模块

访问控制模块主要基于 CORBA 安全服务规范中的访问控制模型，在该模型中没有用户的概念，取而代之的是主体，每一个主体都有相应的角色属性，每一个角色属性又与具体的权限相对应，而目标对象方法的访问又需要相应的权限，这样就可以得出主体是否有目标对象方法的访问权限。

### 4.3.1 主体到角色属性的映射

主体是 CORBA 安全服务中最基本的实体，主体的角色属性表现为主体对某一特定目标对象的访问能力。主体与角色属性之间的映射关系示例如表 4-1 所示：

表 4-1 主体与角色属性映射关系
Table 4-1 Mapping relationship of entity and role property

| 主体 | 角色属性 |
|---|---|
| p1 | a1 |
| p2 | a1,a2 |

### 4.3.2 角色属性到权限的映射

访问控制策略中，角色属性对应的权限表现为拥有该角色属性的主体所具有的访问权限，角色属性与权限之间的映射关系示例如表 4-2 所示：

表 4-2 角色属性与权限映射关系
Table 4-2 Mapping relationship of role property and privilege

| 角色属性 | 权限 |
|---|---|
| a1 | r1,r2 |
| a2 | r3 |

### 4.3.3 目标对象访问权限

访问控制策略中，目标对象访问权限表现为访问某一特定目标对象的特定方法所需具备的权限，目标对象权限列表示例如表4-3所示：

表 4-3 目标对象权限列表
Table 4-3 Target object privilege list

| 目标对象方法 | 所需权限 | 组合方式 |
|---|---|---|
| o1m1 | r1 | all |
| o1m2 | r1,r2,r3 | all |
| o2m1 | r1,r2 | any |

在上述权限列表中，$o_i m_j$表示第 i 个目标对象的第 j 个方法，组合方式 all 表示只有具备所有的所需权限才能访问目标对象方法，组合方式 any 表示只要具备任一所需权限即可访问目标对象方法。

通过主体与角色属性映射表，角色属性与权限映射表以及目标对象访问权限列表，最终可以得出主体对某一特定目标对象的特定方法是否有访问权限，如表 4-4 所示：

表 4-4 目标对象权限列表
Table 4-4 Entity privilege list

| 主体 | 目标对象方法 | 是否有权访问 |
|---|---|---|
| p1 | o1m1 | Y |
| p1 | o1m2 | N |
| p1 | o2m1 | Y |
| p2 | o1m1 | Y |
| p2 | o1m2 | Y |
| p2 | o2m1 | Y |

访问控制过程如下：

1. 获得主体证书，并验证主体证书的合法性
2. 查询证书吊销列表，验证主体证书是否已被吊销
3. 读取主体信息
4. 根据主体信息，验证主体的身份
5. 读取主体要访问的目标对象的信息
6. 查询访问控制列表，验证主体是否有该目标对象方法的操作权限
7. 访问目标对象

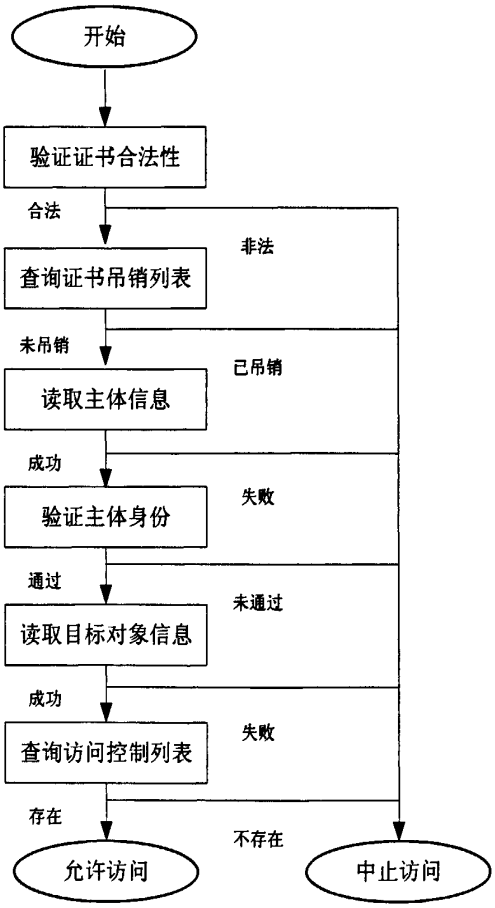


图 4-16 访问控制流程
Fig 4-16 Access control process

## 4.4 安全通信模块

安全通信模块既要保证数据传输的安全性，同时还要保证数据传输的完整性，安全通信模块的工作流程如下：

1. 利用预备秘密、客户机随机数和服务器随机数，运行消息摘要算法生成主秘密；

2. 利用主秘密、客户机随机数和服务器随机数，运行消息摘要算法生成对称密钥；
3. 利用主秘密、客户机随机数和服务器随机数，运行消息摘要算法生成MAC；
4. 数据发送方将应用数据分块；
5. 压缩分块数据；
6. 对压缩后的分块数据添加MAC；
7. 对数据进行加密；
8. 附加头信息；
9. 数据接收方删除头信息；
10. 对数据进行解密；
11. 验证数据的完整性；
12. 对数据进行解压缩；
13. 将分块数据合成完整的应用数据。

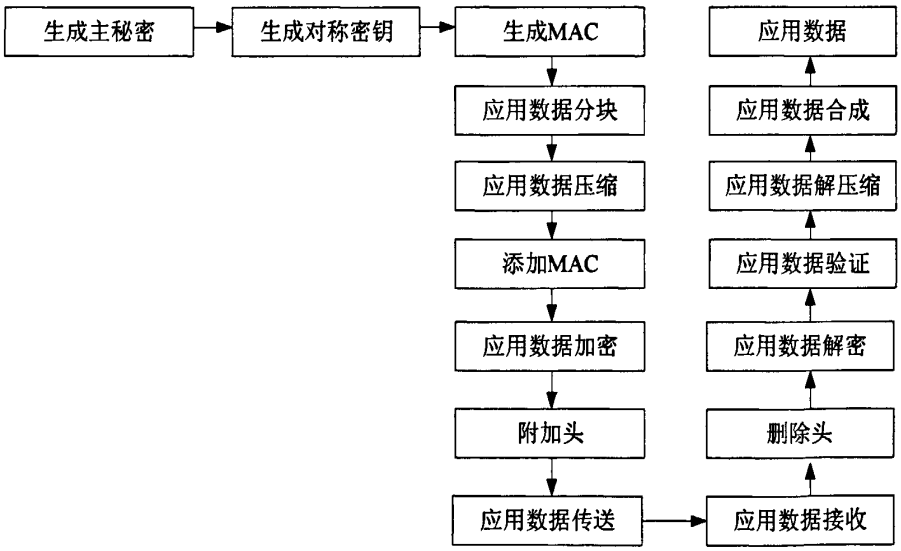


图 4-17 安全通信模块工作流程

Fig 4-17 Security communication module working process

## 4.5 本章小结

本章对于系统安全架构的证书管理模块、安全服务模块和安全通信模块进行了详细设计，其中，证书管理模块主要基于PKI技术，利用OpenSSL提供的API实现；安全服务模块的身份认证部分也是利用OpenSSL的API实现；安全服务模块的访问控制模块主要基于CORBA安全服务规范的访问控制模型实现；安全通信模块主要基于SSL技术实现。

# 第5章 网络税控系统安全策略设计与实现

## 5.1 网络税控系统介绍

网络税控系统是用于对商场超市进行税务监控的基于 CORBA 的分布式税控管理系统，并制订了国标七规范，其核心功能是向国家税收部门提供零售商场和快餐企业的销售信息即税源数据以加强对税收的管理。

网络税控系统的拓扑结构如图 5-1 所示：

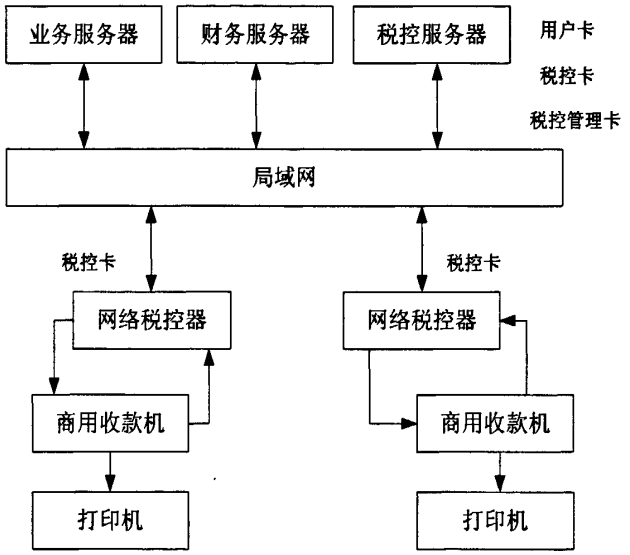


图 5-1 网络税控系统结构

Fig 5-1 Network tax control system architecture

### 5.1.1 税控服务器

税控服务器是指专用于收集网络税控器、业务服务器、财务服务器的税控数据，并完成税务管理部门税控功能要求的服务器。

在税控系统中，税控服务器为其它终端提供服务，是整个税控系统的核心，税控服务器的主要功能如下：

- 采集并分析业务服务器的业务数据和财务服务器的财务数据；
- 初始化商用收款机和网络税控器，检测其联机情况并记录其工作状态信息；
- 管理发票，包括发票的分发、回收、开退以及再分配；
- 回送税控监管数据到网络税控器；

- 对税控相关数据进行采集、保存、分析和统计，生成税控查询和统计数据；
- 满足税务机关进行稽查的要求，并提供供税务机关稽查所使用的应用交互界面；
- 提供报税功能，用于向税务机关提供税收数据。

### 5.1.2 网络税控器

网络税控器是指衔接 GB 18240.1，GB 18240.2，GB 18240.3 标准，用于生成税控码，完成与税控服务器之间传送税控数据的电子装置。通过网络税控器与税控服务器的合作，结合已有的网络体系以及硬件设备，来完成对原有的商用收款机的税控改造。网络税控器主要功能是接收税控数据采集模块的税源数据进行解析从而生成税控码，并向税控数据采集模块回传税控数据；向税控服务器上传税控数据。

### 5.1.3 业务数据监控模块

业务数据监控模块是指驻留于业务服务器的软件模块，其作用是向税控服务器提供必要的业务数据，进而生成业务报表。

### 5.1.4 财务数据监控模块

财务数据监控模块是指驻留于财务服务器的软件模块，其作用是向税控服务器提供必要的财务数据，进而生成财务报表。

### 5.1.5 商用收款机

商用收款机，简称 POS 机，是用于商品交易过程中，具有计算、记录、显示、打印等功能的电子设备。

### 5.1.6 税控数据采集模块

税控数据采集模块，是指驻留在商用收款机操作系统内核的软件模块，主要功能是在商用收款机工作时主动获取商用收款机销售数据并完成与网络税控器的数据交互，依照国家税务机关的格式要求执行税控电子发票的打印。

## 5.2 网络税控系统安全模型

在第四章提出的系统安全架构的基础上，设计网络税控系统安全模型如图5-2所示：

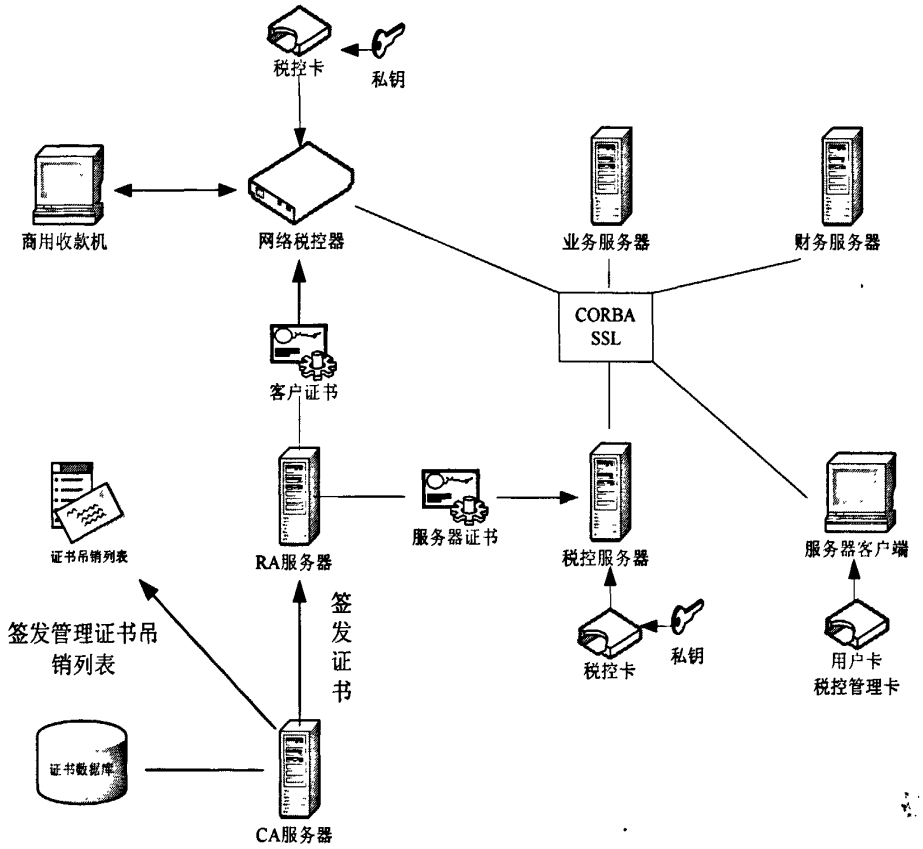


图 5-2 网络税控系统安全模型

Fig 5-2 Network tax control system security module

### 5.2.1 证书管理

网络税控系统中，各个组件之间的交互主要包括，税控服务器与服务器客户端交互、税控服务器与业务服务器交互以及税控服务器与财务服务器交互，这三种交互的途径是依靠CORBA/SSL，网络税控器与商用收款机之间依靠串口进行通讯。其中，税控服务器与网络税控器的交互是系统中最重要的交互，因而需要采取最严格的安全措施，由CA服务器签发并由RA服务器向网络税控器和税控服务器分别发放客户证书和服务器证书。

### 5.2.2 LDAP目录服务

网络税控系统中，使用LDAP协议来完成证书存储和证书吊销列表的存储。网络税控系统LDAP目录树DIT第0层-第1层设计如图5-3所示：

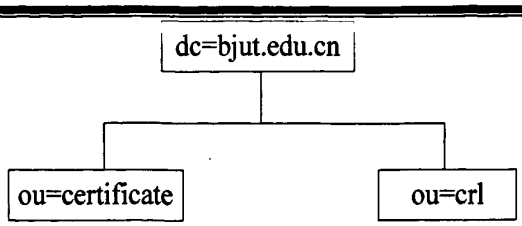


图 5-3 网络税控系统 DIT 结构
Fig 5-3 Network tax control system DIT structure

证书分支 ou=certificate 的子树如图 5-4 所示，划分为网络税控器证书、税控服务器证书和 CA 证书。

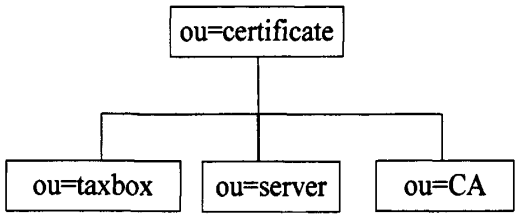


图 5-4 DIT 子树结构
Fig 5-4 DIT subtree structure

LDAP 服务器中，使用 OpenLDAP 目录服务和 Berkeley DB 数据库来存放数据，税控服务器和每一个网络税控器都是一个条目，都有相应的 inetPersonOrg 对象类与之相对应，并且类中有代表税控服务器或网络税控器证书的 userCertificate 属性，从而使 LDAP 服务器可以管理系统中的证书。税控服务器的数字证书在 LDAP 服务器中的条目的存储结构如下：

dn: cn=taxserver, ou=server, ou=certificate,dc=bjut.edu.cn
mail: taxserver@bjut.edu.cn
ou: certificate
ou: server
objectClass:inetorgPerson
objeetClass: Person
sn: taxserver
c: CN
cn: taxserver
…
userCertificate;binary::TEQMA4GA1UEAwwHQ2xpZW50MTEqMCgGCSqGSIb3DQEJARYbYWxsYW53eG1AZ
…

## 5.2.3 身份认证

网络税控系统中，税控服务器与服务器客户端、税控服务器与网络税控器之

间的通讯均需要身份认证机制。

网络税控系统中，主要存在三种类型的用户，分别是普通用户、系统管理员以及稽查人员。税控服务器客户端调用税控服务器相关操作时，必须首先进行身份认证，对于用户的身份认证主要通过用户卡和税控管理卡来实现。

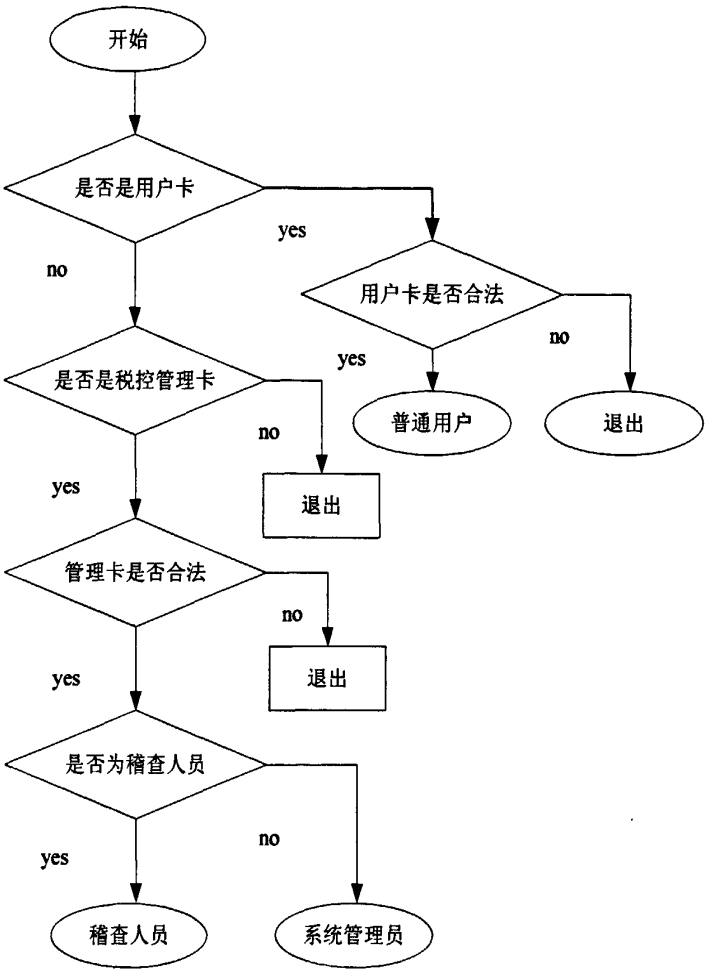


图 5-5 网络税控系统用户身份认证过程

Fig 5-5 Network tax control system user authentication process

税控服务器与网络税控器通讯过程中的身份认证主要通过数字证书来实现，过程如图 5-6 所示：

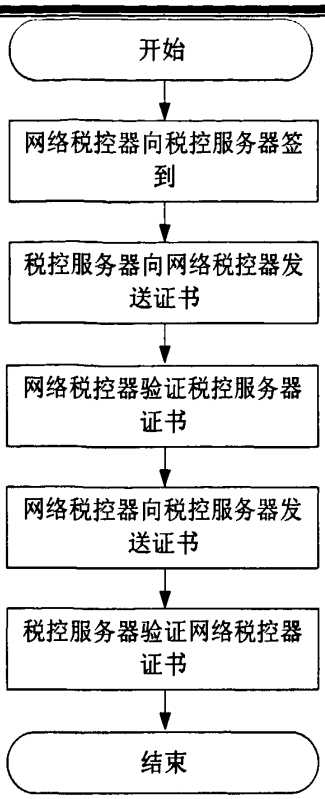


图 5-6 税控服务器与网络税控器身份认证过程
Fig 5-6 Authentication process between tax server and taxbox

网络税控器端实现：

1. 初始化并设置 SSL_CTX 对象
   1) 初始化
      SSL_library_init();
   2) 设置伪随机数生成器，该函数具体实现由于篇幅原因而未给出
      seed_prng();
   3) 初始化 SSL_CTX，设置 SSL 协议算法
      SSL_CTX *ctx = SSL_CTX_new(SSLv23_method());
   4) 加载受信任的 CA 证书
      if (SSL_CTX_load_verify_locations(ctx, "ca.pem", NULL) <=0)
      return LOAD_LOCATIONS_ERROR;
      由于该函数只支持 pem 格式的证书，需要使用以下命令将 crt 格式证书转换成 pem 格式，命令如下：
      openssl x509 –in ca.crt –inform DER –out ca.pem –outform PEM
   5) 寻找默认验证路径
      if (SSL_CTX_set_default_verify_paths(ctx) <=0)
      return SET_DEFAULT_PATHS_ERROR;
   6) 加载证书信息至 SSL_CTX 对象中

if (SSL_CTX_use_certificate_chain_file(ctx, "client1.pem") <= 0)
return USE_CHAIN_FILE_ERROR;

7) 加载私钥文件

if (SSL_CTX_use_PrivateKey_file(ctx, "client1.pem",
SSL_FILETYPE_PEM) <=0)
return USE_PRIVATEKEY_FILE_ERROR;

8) 设置证书的验证方式，该函数的第三个参数为处理验证的回调函数，回调函数的具体实现省略

SSL_CTX_set_verify(ctx, SSL_VERIFY_PEER, verify_callback);

9) 设置最大链长度

SSL_CTX_set_verify_depth(ctx, 10);

10) 设置 SSL_CTX

SSL_CTX_set_options(ctx, SSL_OP_ALL|SSL_OP_NO_SSLv2);

11) 设置加密算法

if (SSL_CTX_set_cipher_list(ctx, "DES-CBC3-SHA") <= 0)
return SET_CIPHER_LIST_ERROR;

2. 与服务器建立连接并验证服务器证书

1) 创建新的与服务器的连接

BIO *conn = BIO_new_connect("taxserver.bjut.edu.cn : 8001");
if (!conn)
return CREATE_BIO_ERROR;
if (BIO_do_connect(conn) <= 0)
return CONNECT_SERVER_ERROR;

2) 初始化并设置 SSL 对象

SSL *ssl = SSL_new(ctx);
SSL_set_bio(ssl, conn, conn);
if (SSL_connect(ssl) <= 0)
return CONNECT_SSL_ERROR;

3) 验证税控服务器证书，该函数具体实现由于篇幅原因而未给出

if ((post_connection_check(ssl, "taxserver.bjut.edu.cn")) != X509_V_OK)
return CHECK_SSL_ERROR:

4) 向服务器传递证书并关闭连接

if (do_client_loop(ssl))
SSL_shutdown(ssl);
else
SSL_clear(ssl);
SSL_free(ssl);

```
SSL_CTX_free(ctx);
```

5) 向服务器传递证书的函数

```
int do_client_loop(SSL *ssl) {
  int   err, nwritten;
  char buf[80];
  for (;;) {
    if (!fgets(buf, sizeof(buf), stdin))
    break;
    for (nwritten = 0;   nwritten < sizeof(buf);   nwritten += err) {
      err = SSL_write(ssl, buf + nwritten, strlen(buf) - nwritten);
    if (err <= 0)   return 0; }
  }
  return 1;
}
```

### 5.2.4 访问控制

网络税控系统的访问控制策略中，主体分别是：普通用户、系统管理员、稽查人员以及网络税控器，客体则是税控服务器和网络税控器所提供的接口，主要包括税控服务器和网络税控器初始化接口、时间校准接口、报税接口、稽查接口、监管数据回送接口、发票管理接口、人员管理接口和设备管理接口等。网络税控系统中，主体与角色属性的映射关系如表 5-1 所示：

表 5-1 主体与角色属性映射关系
Table 5-1 Mapping relationship of entity and role property

| 主体 | 角色属性 |
|---|---|
| 网络税控器 | a1 |
| 稽查人员 | a2 |
| 普通用户 | a3 |
| 系统管理员 | a3,a4 |

网络税控系统中，角色属性与权限之间的映射关系如表 5-2 所示：

表 5-2 角色属性与权限映射关系
Table 5-2 Mapping relationship of role property and privilege

| 角色属性 | 权限 |
|---|---|
| a1 | r1 |
| a2 | r2 |
| a3 | r3 |
| a4 | r4 |

网络税控系统中，具体操作与所需权限之间的对应关系如表 5-3 所示：

表 5-3 具体操作与权限映射关系
Table 5-3 Mapping relationship of role specific operation and privilege

| 操作 | 所需权限 | 组合方式 |
|---|---|---|
| 税控服务器网络税控器初始化 | r3 | any |
| 时间校准 | r4 | any |
| 报税 | r3 | any |
| 稽查 | r2 | any |
| 监管数据回送 | r3 | any |
| 发票管理 | r3 | any |
| 人员管理 | r4 | any |
| 网络税控器签到 | r1 | any |

### 5.2.5 安全通信

安全通信功能主要依靠 SSL 协议来实现，由于网络税控系统采用的 CORBA 库是 omniORB，而 omniORB 对 SSL 协议又有很好的支持，因此，只需要对 omniORB 进行相应的配置就可以利用 SSL 协议来实现安全通信，网络税控器端具体配置如下，税控服务器端配置与网络税控器端配置类似。

```
sslContext::certificate_authority_file="ca.crt";
sslContext::key_file="client1.crt";
sslContext::key_file_password="password";
```

其中：

ca.crt 指向了保存在与应用相同目录的 CA 证书文件。

client1.crt 指向了保存在与应用相同目录的网络税控器证书文件。

password 是网络税控器私钥的密码。

## 5.3 本章小结

本章首先给出了网络税控系统的拓扑结构，并对网络税控系统的主要组成部分进行了介绍，然后将本文提出的基于 CORBA 的分布式系统安全架构应用到网络税控系统中，从证书管理、LDAP 服务、身份认证、访问控制和安全通信等多个方面进行了详细的阐述。

# 结论

CORBA是为了解决分布式异构环境下对象之间的互操作性问题而提出的基于中间件的分布式对象技术，由于CORBA支持异构分布式应用程序间的互操作性以及独立于平台和编程语言的对象重用，因而近年来在信息系统中得到了广泛的应用。由于CORBA的分布式特性，基于CORBA的信息系统更容易受到入侵也更需要受到安全保护，本文主要的解决问题是如何保障基于CORBA的分布式系统的安全。

本文所作的具体研究工作和总结如下：

1. 首先介绍了PKI、CA、SSL、OpenSSL等相关技术并分析了分布式系统主要存在的威胁，在分析这些威胁的基础上，给出了分布式系统所应具备的安全机制，并提出了基于CORBA的分布式系统安全架构，包括证书管理模块、安全服务模块和安全通信模块。
2. 针对基于CORBA的分布式系统安全架构中的证书管理模块进行详细设计，应用OpenSSL技术和CA技术，实现了证书管理模块中的证书生成和发放、证书吊销、证书吊销列表更新等功能。
3. 针对基于CORBA的分布式系统安全架构中安全服务模块的身份认证和访问控制部分进行详细设计，应用SSL技术和CORBA安全服务规范中的访问控制模型，实现了安全服务模块中的身份认证和访问控制功能。
4. 针对基于CORBA的分布式系统安全架构中的安全通信模块进行详细设计，应用SSL技术来保证数据传输的安全性和完整性。
5. 将基于CORBA的分布式系统安全架构集成到网络税控系统中，并从证书管理、LDAP服务、身份认证、访问控制和安全通信等多个方面进行了详细的阐述。

由于时间关系和作者的经验有限，本文存在以下有待深入研究和进一步改进的地方，归纳如下：

1. 本文提出了基于CORBA的分布式系统安全架构，并将其应用到了网络税控系统中，但是该架构是在网络税控系统的基础上提出的，要成为可以应用到大多数信息系统中的安全架构，还需要进一步优化和改进。
2. 网络税控系统目前只是一个局域网内的系统，但是未来的发展方向是每一个商场超市的税控服务器作为国税局税控总服务器的前台通过互联网与税控总服务器进行交互，因此安全架构还需要针对这方面进行优化和改进。

# 参考文献


1 OMG. CORBA3.0 Specification. Object Management Group, 2003,9:6~10

2 OMG. CORBA Component Model Specification Version 4.0. Object Management Group, 2006.4:3~5

3 Dirk Slama, Jason Garbis, Perry Russell 著. 李师贤，郑红，吴涛 译. CORBA 企业解决方案. 北京：机械工业出版社，2001.1:60~68

4 Behrouz A.Forouzan 著. 马振晗，贾军保 译. 密码学与网络安全. 北京：清华大学出版社，2009.1:105~122

5 A.Frier, P.Karlton, P.Kocher. The SSL 3.0 Protocol. Netscape Communication Corp, 1996.11:5~12

6 林代茂. 信息安全-系统的理论与技术. 北京：科学出版社，2008:30~35

7 William Stallings. Cryptography and Network Security Principles and Practices. 北京：机械工业出版社，2006.11:25~32

8 OMG. CORBA Security Service Specification Version 1.8. Object Management Group. 2003,3:10~30

9 OMG. A Discuss of the Object Management Architecture. 1997,1:8~12

10 杨锡慧，吴国新. CORBA 的安全机制及实现. 计算机应用研究，2003(5):60~62

11 李晓东，周兴社. 通用 CORBA 安全服务的研究与实现. 计算机工程与应用，2003,39(1)：42~44

12 朱其亮，郑斌. CORBA 原理及应用. 北京：北京邮电大学出版社，2001.10:68~75

13 OMG. CORBA Services：Common Objeet Services Specification. 1997,11:10~20

14 Bob Blakley. CORBA Security：An Introduction to Safe Computing with Objects. Addison Wesley, 2000:160~173

15 宁宇鹏，陈昕. PKI 技术. 北京：机械工业出版社，2004.3:58~66

16 C.Adams, S.Farrell. Internet X.509 Public Key Infrastructure Certificate Management Protocols. RFC2510, 1999,3:15~22

17 佘堃，郑方伟. PKI 原理与技术. 成都：电子科技大学出版社，2007.8:78~86

18 《中国商用密码认证体系结构研究》课题组. 数字证书应用技术指南. 北京：电子工业出版社，2008.1:20~50

19 J.Hodges, R.Morgan, M.Wahl. Lightweight Directory Access Protocol(v3):Extension for Transport Layer Security. RFC2830, 2000.5:6~12

20 K.Zeilenga. Collective Attributes in the Lightweight Directory Access Protocol(LDAP). RFC3671, 2003.12:15~18

21 Mohammand S.Obaidat，Noureddine A.Boudriga 著，毕红军，张凯 译. 计算机网络安全导论. 北京：电子工业大学出版社，2009.5:170~183

22 Anon. Data encryption essentials. Software World, 2005,36(5):15~16

23 RSA Laboratories. RSA Encryption Standard. PKCS#1, 1993:8~10

24 Michael E.Whitman, Herbert J.Mattord 著，向宏，傅鹂 译. 信息安全管理. 重庆：重

庆大学出版社，2005.3:210~218

25 Feng Xiaoling. Analysis of Structure and Technology on Public-Key Infrastructure. Computer Development&Applications, 2003:78~83

26 Internet X.509 Public Key Infrastructure Online Certificate Status Protocol. RFC2560, PKIX Working Group, 1999:25~30

27 马臣云，王彦．PKI 网络安全认证技术与编程实现．北京：人民邮电出版社，2008.7:310~322

28 S.Berkovits, S.Chokhani, J.A.Geiter, J.C.guild. Public Key Infrastructure Study, Final Report. Produced by the MITRE Corporation for NIST, 1994,4:35~38

29 Mary R, Abdelilah E, Srilekha M. Cetificate-based Authorization Policy in a PKI Environment. ACM Transactions on Information and System Security, 2003,6(4):566~588

30 David Wagner, Bruce Schnerder. Analysis of the SSL3.0 Protocol, 1996.11:35~38

31 李涛．网络安全概论．北京：电子工业出版社，2004,11:260~265

32 王娟，邱宏茂，盖磊，王海军．SSL 及使用 OpenSSL 实现证书的签发和管理．西安：计算机技术与发展，2004(10):25~27

33 范恒英，何大可．用 OpenSSL 进行 TLS/SSL 编程．通信技术，2002,6:82~85

34 Pravir Chandra, Matt Messier, Joln Viega.Network Security with OpenSSL. O'Reilly, 2002,6:16~26

35 Gunter K. Authorization in CORBA Security. Journal of Computer Security, 2000,8(3):130~135

36 Canal C, Fuentes L, Pimentel E. Adding Roles to CORBA Objects. Software Engineering.IEEE Transactions, 2003，29(3):242~262

37 Maffeis S, Schmidt D. Constructing Reliable Distributed Communication Systems with CORBA. IEEE Communications Magazine, 1997,35(2):56~63

38 Basit Shafiq, James B.D.Joshi, Elisa Bertino, and Arif Ghafoor. Secure Interoperation in a Multidomain Environment Employing RBAC Policies. IEEE Transctions on Knowledge and Data Engineering, 2005,11(17):95~105

39 陈晓苏，熊晓萍，肖道举．CORBA 安全中的访问控制策略研究．华中科技大学学报(自然科学版)，2004,32(6):21-24

40 Doherty Conor, Uslaender Thomas. Enterprise CORBA Application Management Architecture. Network and Systems Management, 1997,7(1):127~132

41 谢冬青．计算机网络安全技术教程．北京：机械工业出版社，2007:162~168

42 贾晶．信息系统的安全与保密．北京：清华大学出版社，2004:265~270

43 魏利明，陈相宁．PKI 技术分析．网络安全技术与应用，2005,(3):19~21

# 攻读硕士学位期间所发表的学术论文

1 王先明, 于书举. 税源数据安全传输解决方案研究. 微计算机信息. 已录用

# 致谢

本论文是在我的导师于书举教授的悉心指导下完成的，三年来，于老师严谨的治学态度，对工作的严格要求和对学生的关心都深深影响并教育着我。在此谨向于老师表示最衷心的感谢，感谢他三年来对我的悉心培养和指导，使我不仅学到了专业知识，更在项目实践中培养了分析问题和解决问题的能力。

感谢许老师和杜老师在项目实践中给予我的指导和帮助，感谢曹增明、贾豪杰、汪寅等同学在学习过程给予我的宝贵的建议，感谢陪我一起走过研究生生涯的我的舍友，陈凯、邵华和李瑜，感谢他们使我的生活更加充实。

感谢我的父母，感谢他们一直以来对我的支持，使我在遇到困难时有人倾诉，遇到挫折时有人鼓励，他们对我的爱是我一生最大的财富。

感谢我的母校北京工业大学，今后无论走到哪里，我都会将母校的校训牢记于心——不息为体，日新为道。

最后，感谢答辩委员会的老师对我的论文的评审和提出的宝贵意见。