

网络空间安全

专业课程实验指导书

山东大学软件学院

前 言

在飞速发展的网络时代，特别是电子商务时代，信息安全（特别是网络安全）越来越表现出其重要性，研究和学习信息安全知识,并掌握相应主流技术迫在眉睫。自 2002 年开始,我们尝试在全校开设通选课，然后在计算机科学与技术学院计算机科学与技术专业的毕业班中作为必选选修课进行讲授，近两年在电子商务专业进行了更系统的课程规划，内容涉及信息安全的基本理论框架，包括网络安全框架、对称密码技术、公开钥密码技术、HASH 函数、MAC 函数等基本密码学理论，同时也涉及到更高层的基于密码技术的安全协议分析和应用，也兼顾网络入侵、恶意软件、防火墙等网络安全技术。本实验将主要集中在涉及密码技术的实验内容上。

信息安全理论和技术作为一门综合性科目，要求学生应具备较全面较扎实的理论基础，课程基础涉及范围广，课程理论相对比较抽象和繁杂，因而同学们在学习中会有一定难度。为了使理论教学与实践教学紧密结合，注重学生的理解和动手能力培养，我们安排了信息安全系列实验内容来配合教学环节，希望同学们能认真独立的完成实验内容，增进对课程内容的理解，提高自己理论联系实际的能力，提高自己独立思考解决问题的能力。

本实验采用了一些信息安全方面开放源码的较成熟的软件包和部分商业化并可用于教学目的的软件产品作为实验的基本平台，这有利于同学们能够充分利用因特网进行更多的实验内容的收集和进一步研究的展开，充分利用网络信息安全相关资源，将更有助于本实验内容的良好完成。

山东大学软件学院

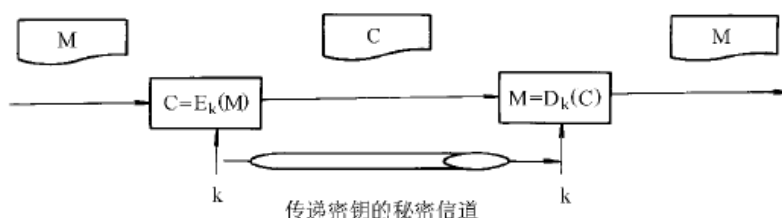
目 录

一、 信息安全基本理论简介	4
二、 实验基本环境简介	9
三、 系列实验	10
实验一 VMWARE 虚拟机和 WIN2000 安装	10
实验二 Sniffer 网络侦听和 pcap 编程	14
实验三 对称加密算法实验	20
实验四 公钥算法实验	24
实验五 微软 CA 组件安装和配置	28
实验六 EJBCA 系统的安装和配置	30
实验七 SureCA 系统的安装和配置	45
实验八 安全电子邮件的配置和使用	62
实验九 安全 WEB 访问的配置和使用	68
实验十 OPENSSL 软件包的使用	76
实验十一 OpenVPN 配置和使用	82

一、 信息安全基本理论简介

1. 对称密码技术

对称密码加密也称常规密码加密、单钥密码加密、秘密密钥加密，它包括许多数据加密方法。对称密码系统的基本模型见下图：

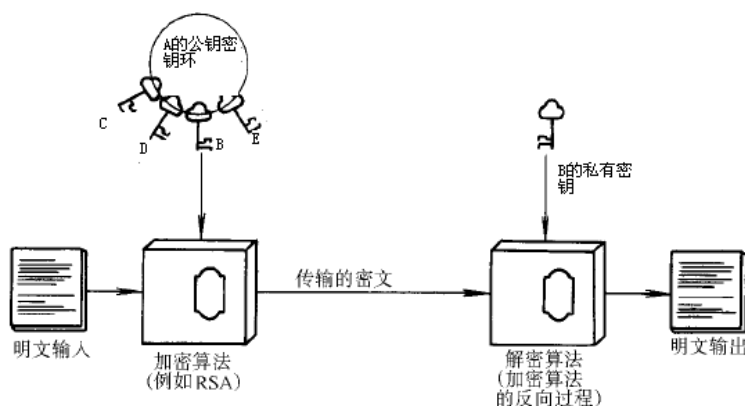


其基本特征是：数据加密和解密使用同一个密钥；在算法公开的前提下所有秘密都在密钥中，因此密钥本身应该通过另外的秘密信道传递。对称密码系统的安全性依赖于两个因素：其一，加密算法强度至少应该满足：当敌手已知算法，通过截获密文不能导出明文或者发现密钥。更高的要求是当敌手即使拥有部分密文以及相应明文段落也不能导出明文或者发现密钥系统。其二，发送方和接收方必须以安全的方式传递和保存密钥副本，对称加密的安全性取决于密钥的保密性而不是算法的机密性。

2. 公开钥密码技术

公钥密码也称为非对称密码，公钥密码系统的核心是信源端对明文加密和信宿端对密文解密时分别使用两个相互对应，但计算上只能单向推导的一对密钥。根据应用的需要，将其中一个称为公钥，另一个称为私钥。

传统的对称密码系统主要是建立在位操作基础之上，而公钥密码算法和密钥生成则是建立在数学函数基础之上。目前，公钥密码理论中大量使用数论等数学理论和方法，对现代密码学产生了深远的影响。公钥加密方法的安全性主要基于复杂数学问题的难解性假设，根据所基于的数学难题来分类，以下三类系统目前被认为是安全和有效的：基于大整数因子分解的公钥密码系统（如 RSA）、椭圆曲线离散对数系统（如 ECC），离散对数系统（如 DSA）。



在数据保密通信中，加密钥匙是公开的，解密私钥原则上不传递，因此密钥的分配和管理较对称密码系统简单。公开密钥加密系统还能够很容易地实现数字签名。因此，公钥密码技术适应了电子商务应用需要。

3. HASH 函数

HASH 函数，又称杂凑函数，是在信息安全领域有广泛和重要应用的密码算法，它有一种类似于指纹的应用。在网络安全协议中，杂凑函数用来处理电子签名，将冗长的签名文件压缩为一段独特的数字信息，像指纹鉴别身份一样保证原来数字签名文件的合法性和安全性。SHA-1 和 MD5 是目前最常用的杂凑函数。经过这些算法的处理，原始信息即使只更动一个字母，对应的压缩信息也会变为截然不同的“指纹”，这就保证了经过处理信息的唯一性。为电子商务等提供了数字认证的可能性。

HASH 函数，又称杂凑函数，是在信息安全领域有广泛和重要应用的密码算法，它有一种类似于指纹的应用。在网络安全协议中，杂凑函数用来处理电子签名，将冗长的签名文件压缩为一段独特的数字信息，像指纹鉴别身份一样保证原来数字签名文件的合法性和安全性。SHA-1 和 MD5 是目前最常用的杂凑函数。经过这些算法的处理，原始信息即使只更动一个字母，对应的压缩信息也会变为截然不同的“指纹”，这就保证了经过处理信息的唯一性。为电子商务等提供了数字认证的可能性。

设有散列函数 $h = H(M)$ ，这里 M 是可变长度消息， h 是固定长度的函数值。散列函数值行的作用是对消息 M 产生一个“摘要”，使得接收方能够对消息 M 的完整性进行检验。散列方法本身并不需要保密。

4. MAC 函数

MAC 函数也称密码校验和，它由如下形式的函数 C 产生： $MAC = CK(M)$ ，其中 M 是一个变长消息， K 是收发双方共享的密钥， $CK(M)$ 是定长的认证符。在假定或已知消息正确时，将 MAC 附于发送方的消息后；接收方可通过计算 MAC 来认证该消息。

5. 数字签名

在收发双方不能完全信任的情况下，数字签名是解决该问题的最好方法。其作用相当于手写签名。数字签名满足：必须能验证签名者、签名日期和时间；能认证被签的消息内容；应能由第三方仲裁，以解决争议。

数字签名必须是与消息相关的二进制位串，签名必须使用发送方某些独有的信息，以防止伪造和否认，产生数字签名比较容易，识别和验证签名比较容易，伪造数字签名在计算上是不可行的，保存数字签名的拷贝是可行的。

基于密码技术的数字签名一般采用具有数字签名功能的公开钥密码算法，如 RSA、DSA 等。其基本原理是使用秘密钥加密实现数字签名，使用公开钥解密实现签名验证。在实际应用中，数字签名包括两个步骤：（1）对待签名信息计算 HASH 值（2）对 HASH 值采用秘密钥加密得到数字签名值；验证过程是首先将数字签名值用对应的公开钥解密，并和重新计算的信息的 HASH 值进行比较，如果相同，证明验证签名正确，否则认为是错误。

6. 数字证书

数字证书是指利用数字签名技术实现的经由第三方可信的、权威的机构 CA 签发的，将被认证对象（用户）的身份信息和其公开钥进行有效捆绑而编码形成的数字认证信息。通过验证者对数字证书的验证，确保用户身份和用户公开钥的一一对应性，从而确认用户对该公开钥的合法拥有，从而利用该公钥进行安全的信息加密。

X509 标准是数字证书的主要标准之一。在一个标准的数字证书中，包含证书版本号、证书序列号、证书签发者身份信息、证书拥有者（用户）身份信息、证书有效期、证书拥有者公钥信息、某些扩展信息、签名方法以及证书签发者用自己的私钥对以上信息所做的签名产生的签名信息。证书的验证主要包括验证数字签名是否正确（确认证书是否被修改）、证书有效期是否有效、证书签发者是否可信、证书中其他信息是否符合政策、证书是否已经被注销等，在证书链中，还应验证证书链中所有的证书是否符合信任链关系等。

7. 证书注销列表（黑名单）

数字证书在有效期内因各种原因（对应秘密钥丢失、身份信息变更等）可能变得不安全，需要申请注销。证书注销列表是由可信的、权威的第三方机构 CA 审核签发的所有在证书有效期内，但是被注销的证书的列表。该列表经由 CA 机构签名保证可信。用户通过定期下载，在验证证书有效性时使用。

8. CA 中心

CA 中心（Certificate Authority）即数字证书认证机构，是一个可信的、权威的第三

方机构，其主要功能就是为用户（包括人和设备等）签发数字证书，并实施相应的管理。

CA 的核心功能就是发放和管理数字证书，具体描述如下：

- (1) 接收验证最终用户数字证书的申请。
- (2) 确定是否接受最终用户数字证书的申请—证书的审批。
- (3) 向申请者颁发、拒绝颁发数字证书—证书的发放。
- (4) 接收、处理最终用户的数字证书更新请求—证书的更新。
- (5) 接收最终用户数字证书的查询、撤销。
- (6) 产生和发布证书注销列表（CRL）。
- (7) 数字证书的归档。
- (8) 密钥归档。
- (9) 历史数据归档。

CA 的数字签名保证了证书的合法性和权威性。主体的公钥可有两种产生方式：(1) 用户自己生成密钥对，然后将公钥以安全的方式传给 CA，该过程必须保证用户公钥的可验证性和完整性。(2) CA 替用户生成密钥对，然后将其以安全的方式传送给用户，该过程必须确保密钥的机密性、完整性和可验证性。该方式下由于用户的私钥为 CA 所产生，故对 CA 的可信性有更高的要求。

RA (Registry Authority, 注册中心)，是数字证书注册审批机构。RA 系统是 CA 的证书发放、管理的延伸。它负责证书申请者的信息录入、审核等工作；同时，对发放的证书完成相应的管理功能。

RA 系统是整个 CA 中心得以正常运营不可缺少的一部分。但有的系统中，将 RA 合并到 CA 中。一般说来，注册机构控制注册、证书传递、其他密钥和证书生命周期管理过程中主体、最终实体和 PKI 间的交换。

9. PKI

公钥基础设施 (Public Key Infrastructure, PKI) 是一个用公开密钥算法原理和技术实现并提供安全服务的具有通用性的安全基础设施。PKI 是一种遵循标准的利用公钥加密技术为电子商务、电子政务的开展提供一整套安全的基础设施。用户利用 PKI 平台提供的安全服务进行安全通信。PKI 这种遵循标准的密钥管理平台，能够为所有网络应用透明地提供采用加密和数字签名等密码服务所需要的密码和证书管理。使用基于公开密钥技术平台的用户建立安全通信信任机制的基础是，网上进行的任何需要提供安全服务的通信都是建立在公钥的基础之上的，而与公钥成对的私钥只掌握在他们与之通信的对方。这个信任的基础是通过公钥证书的使用来实现的。公钥证书就是用户的身份与之所持有的公钥的结合，在结合之前，由一个可信任的权威机构——认证机构 (CA) 来证实用户的身份。然后由可信任的 CA 对该用户身份及对应公钥相结合的证书进行数字签名，用来证明证书的有效性。PKI 首先必须具有可信任的认证机构，在公钥加密技术基础上实现证书的产生、管理、存档、发放以及证书撤销管理等功能，并包括实现这些功能的硬件、软件、人力资源、相关政策和操作规范以及为 PKI 体系中的各成员提供全部的安全服务，例如，身份认证、数据保密性、完整性以及不可否认性服务等。

构建实施一个 PKI 系统主要包括以下内容：

（1）认证机构证书的签发机构，它是 PKI 的核心，是 PKI 应用中权威的、可信的、公正的第三方机构。

（2）证书库证书的集中存放地，提供公众查询。

（3）密钥备份及恢复系统对用户的解密密钥进行备份，当丢失时进行恢复，而签名密钥不能备份和恢复。

（4）证书撤销处理系统证书由于某种原因需要作废，终止使用，将通过证书撤销列表 CRL 来实现。

（5）PKI 应用接口系统

综上所述，PKI 是一种新的安全技术，它基于公开密钥密码技术，通过数字证书建立信任关系。PKI 是利用公钥技术实现电子商务安全的一种体系，是一种基础设施，可以保证网络通信、网上交易的安全。

二、 实验基本环境简介

基本的实验环境主要包括以下：

1. 虚拟安全环境的建立

为了保持本实验的独立性和安全性，搭建一个相对独立的安全环境很有必要。本实验建议在微机实验平台上首先安装一个硬件虚拟软件环境，如 VMWARE 等。在其中再安装一个操作系统平台，如 WINDOWS2000 SERVER 等。在这样一个虚拟硬件环境和操作系统平台上，安装和配置任何的安全实验环境将不会影响主系统平台的安全性。所以实验内容之一就是建立一个这样的安全环境。

2. 安全实验环境内容的总体描述

本系列实验的总体目标是：通过安装和配置一系列 CA 系统（包括微软公司操作系统中提供的 CA 模块、商业化 CA 系统以及一些开放源码的 CA 系统），理解并掌握 CA 系统的原理和实际使用方法；进而根据实际应用系统的安全需要，通过数字证书的使用，搭建一些基本的 PKI 安全环境，从而理解并掌握在实际应用中是如何做到基于密码技术来完成安全化改造的，进一步掌握 PKI 技术的开发，能够做到对系统进行安全需求分析，并做一些较为简单有效的安全方案，提高实际动手能力。

本系列实验包括的环境是：

- 搭建微软操作系统上自带的一套 PKI 安全环境。
- 在 WINDOWS 平台上搭建一个基于 JAVA 和 WEB 环境的大型 CA 系统平台。
- 配置一个在 WINDOWS 平台下基于 C/S 环境的 CA 系统环境。
- 利用各种 CA 平台，签发各种类型的数字证书，并在实际应用环境中进行配置（典型和成熟的应用包括：安全电子邮件系统、安全 WEB 访问等）。
- 在 C 语言环境下，利用已经提供的安全开发包，进行二次开发，构造一个比较实际的安全应用，提高实际安全分析和动手能力。

在该系列实验内容里，分三个层次：（1）了解和掌握 CA 系统是干什么的，为什么要这样，原理是什么（2）CA 签发的数字证书是如何被使用的，为什么可以这样使用（3）是否可以利用一些现成的开发包去构造基于数字证书的安全应用。

如果通过实验，达到了这三个目的，就算是一个成功的实验课了。

三、 系列实验

实验一 VMWARE 虚拟机和 WIN2000 安装

1.1 实验目的

通过在主系统平台上安装和配置一个虚拟机环境，为下面的实验打好实验环境基础。

1.2 实验内容

1. 在主系统平台上安装和配置 VMwareworkstation-4.5.1 。
2. 在 VMWARE 中安装 WINDOWS 2000 SERVER 操作系统。

1.3 实验指导

1.3.1 软件准备

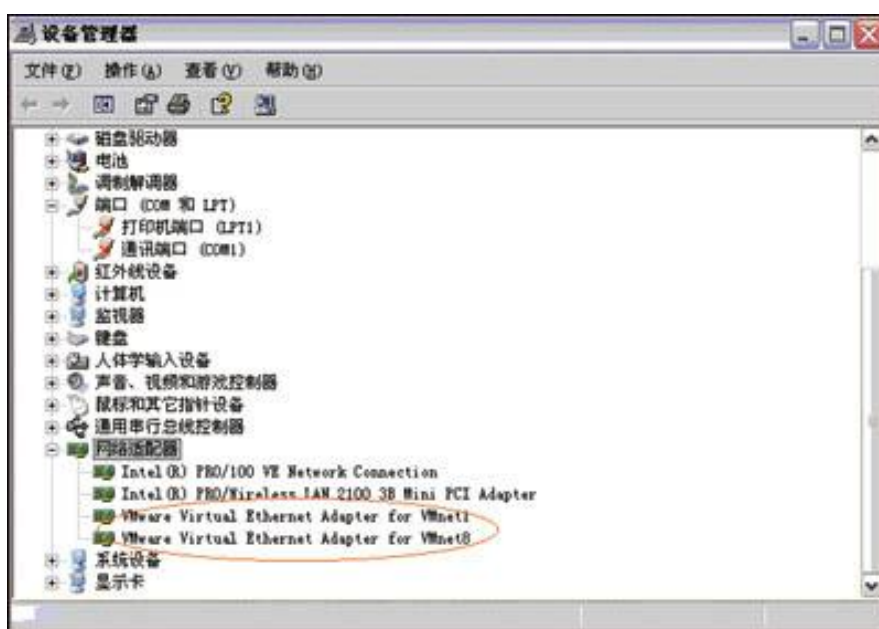
VMware 是 VMware 公司出品的一个多系统安装软件。利用它，你可以在一台电脑上将硬盘和内存的一部分拿出来虚拟出若干台机器，每台机器可以运行单独的操作系统而互不干扰，这些“新”机器各自拥有自己独立的 CMOS、硬盘和操作系统，你可以像使用普通机器一样对它们进行分区、格式化、安装系统和应用软件等操作，还可以将这几个操作系统联成一个网络。在虚拟系统崩溃之后可直接删除不影响本机系统，同样本机系统崩溃后也不影响虚拟系统，可以下次重装后再加入以前做的虚拟系统。同时它也是唯一的能在 Windows 和 Linux 主机平台上运行的虚拟计算机软件。VMware 虚拟机软件不需要重开机，就能在同一台电脑使用好几个 OS，不但方便，而且安全,为了更好的运行虚拟机,一般对要求宿主计算机有比较大的内存(大于 256M,推荐 512M)。

Windows 2000 SERVER 操作系统就不用介绍了。在安装有 VMware 的系统上可以

在其基础上安装更多新的操作系统。

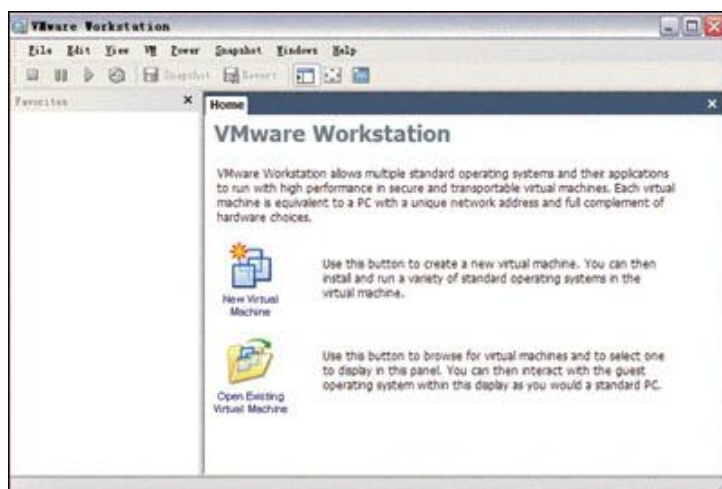
1. VMWARE 虚拟机安装

在安装上没必要作过多的介绍，请大家注意的是 VMware 只能安装在 WinNT/2000/XP 或 Linux，以及 FreeBSD 下。装好之后，你可以发现你多了两块名为 VMware Virtual Ethernet Adapter (basic host-only support for VMnet1)和 VMware Virtual Ethernet Adapter (Network Address Translation (NAT) for VMnet8)的虚拟网卡，这也是 VMware 的特色所在，因为在 VMware 下你可以使用虚拟网卡进行联网设置及其实验。



2. 建立一个虚拟系统：WIN2000 SERVER 在虚拟机中的安装

双击桌面的“VMware”图标，即可进入 VMware 的主窗口，点击右侧的 New Virtual Machine 即可新建一个虚拟系统，之后选择 Typical，再选择一种要安装的操作系统。



现在选择你建立的虚拟系统，点击上方工具栏中的 POWER ON 键便可开机了。其他按钮分别是 POWER OFF: 关机。Suspend: 挂起，可以让虚拟机记录下当前虚拟系统的状态，下次可以用 Resume 重新恢复选择挂起时的运行状态，以便接着工作。Reset: 重启，相当于物理计算机的 RESET 按钮。Full Screen: 全屏，将虚拟机全屏显示。现在一台和真实计算机一样的机器已经建起来了，点击虚拟机的窗口，你的鼠标就融入虚拟系统了，你可以和使用一台真的计算机一样使用它了，而且任何设置都不会影响到你本机。如果想回到主机系统，则可以按 Ctrl+Alt 使鼠标回到主机系统。不要在虚拟机中使用 Ctrl+Alt+Del 组合键，因为主机系统同样也会对这个组合键做出反应，你应当使用 Ctrl+Alt+Ins 来代替。

虚拟机的启动过程和你的 PC 的启动过程也是没有什么不同的，一开始是自检，这时按 F2 可以进入 BIOS 设置。每一台虚拟机都有它自己的 BIOS。

请到 BIOS 中去设置启动顺序（这里的 BIOS 当然是虚拟机中的 BIOS）。之后便可使用光盘来启动安装（虚拟机可以用光盘镜像如：ISO、vcd 文件作为光盘。比如从网上下载的 Linux ISO 文件，不需刻盘，可直接安装。软驱同理）。你会发现在虚拟机中的设备和你实际的设备完全不一样，VMware 为了保证系统的兼容性和稳定性，把现有的设备都虚拟成了最标准的、兼容性最好的设备。由于实际驱动设备的程序仍是在本机系统上运行的驱动程序，实际上的效率并没有多少降低。所以不要试图按照自己的机器配置系统。除此之外，在虚拟机中不用也不能安装任何驱动程序。

通常在安装完虚拟机后，需要安装 VMware tools (菜单 VM->Install VM Tools), 以便更好的支持各种驱动，如显示等。

实验二 Sniffer 网络侦听和 pcap 编程

2.1 实验目的

- (1) 了解和验证网络安全威胁最基本的一种形式：网络窃听。
- (2) 掌握 Ethereal 等网络侦听工具的用途和用法。

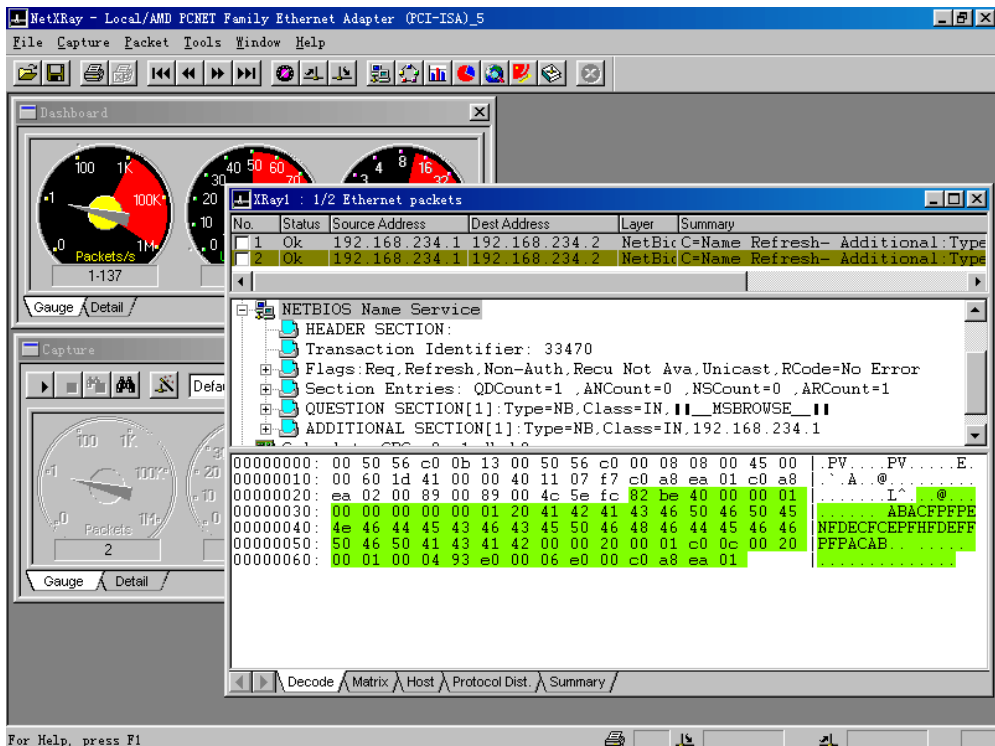
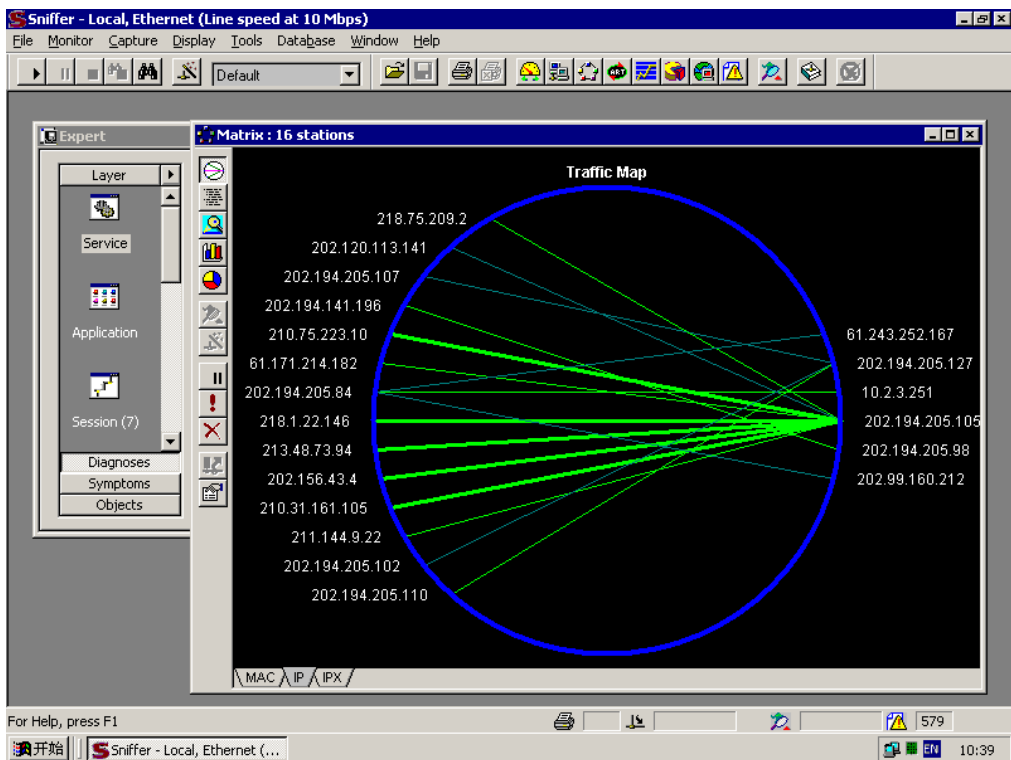
2.2 实验内容

- (1) 查阅资料，回顾以太网的工作原理。
- (2) 使用 Sniffer 工具软件侦听网络传输信息，包括通信内容及账户/口令等敏感信息。
- (3) 用 winpcap/libpcap 开发库进行编程开发(选做)。

2.3 实验指导

网络是以分组交换技术为基础的，即数据分组(报文)在网络中经过交换/路由设备的存储-转发接力完成递交，因此分组完全有机会被沿途的设备查看、篡改甚至假冒。局域网特别是以太网是一个完全不设防的共享式传输系统，尤其是在集线器(Hub)时代窃听是极其容易的。现在交换机(Switch)已成为主流，但是交换机并没有从根本上解决窃听问题。

按照以太网原理，以太网卡可以收听到本网段上的所有帧，也包括和本机无关的其它机器之间的通信。一般开机后网卡正常被设置为过滤模式，即只过滤接收和本机有关的帧。如果把网卡设置为混杂模式(promiscuous)，则可以接收到所有帧。下图第一个可以看到其它机器之间的流量，第二个图是对一个 NetBIOS 分组的格式字段解析。(注：后面使用了不同时期的版本程序的屏幕截图，但是和新版的基本类似)。



这种软件已经有很成熟的了，比 tcpdump/windump、NetXray、Sniffer Pro、Ethereal 等，统称为 Sniffer 软件。当然，这些工具软件的起始目的是为了网络管理和排错，用来演示窃听只是顺便。此类软件一般都具有协议解析及流量分析能力，操作界面也都类似。

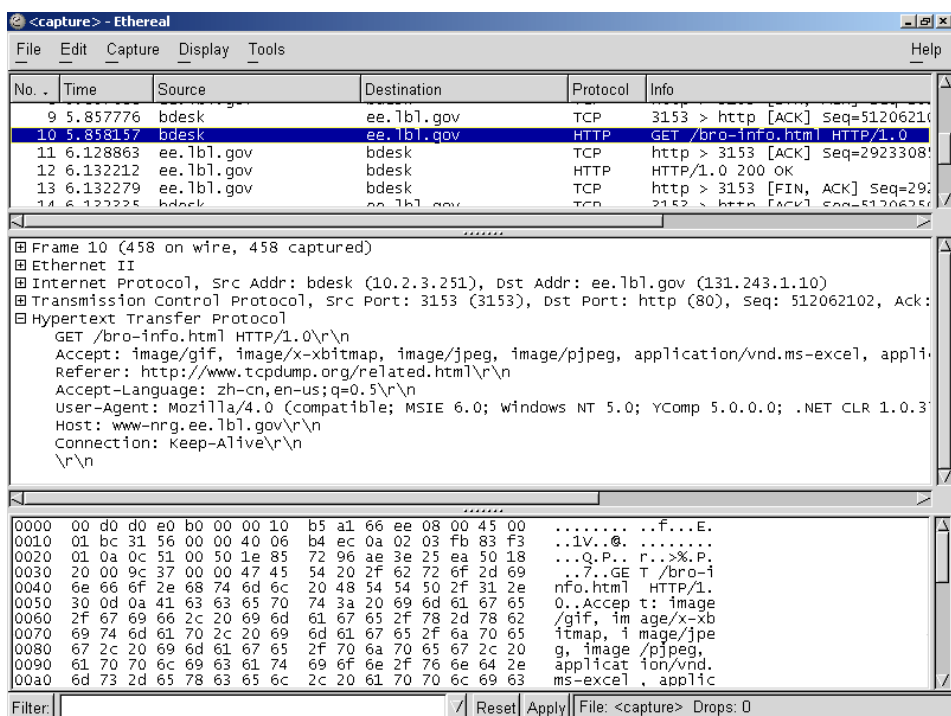
推荐使用 Ethereal 或 Sniffer Pro。Ethereal 功能够强大，而且是开源的，可以从 Ethereal 网站下载，安装是直接的。

2.3.1 用 Sniffer 工具观察协议分组

观察 ARP、UDP、TCP、DNS、HTTP、FTP、POP3、SMTP 等协议的网络分组格式。这些标准格式的协议分组 Sniffer 程序都可以解析。其它程序使用自己的报文格式，比如 QQ 报文等则一般不能被解析。

为了观察特定的分组，开启 Sniffer，制造期望的分组。方法：

- (1) ping 一个本网段但是未开机的机器的 IP 地址，可以引发 ARP 报文。
- (2) ping 一个最近未曾访问过的网站的域名，可引发本机和 DNS 服务器之间的交互，可以观察到 DNS 报文，它是封装在一个 UDP 报文中的。
- (3) 访问某个网页，可以制造 HTTP 流量，顺便可以看到 TCP 会话，包括三次握手的过程。也可能会有 DNS 报文。
- (4) 使用 Outlook 或 Foxmail 收发邮件，可以引发 POP3 和 SMTP 流量，都是封装在 TCP 协议中。
- (5) FTP 到某个站点，可以观察 FTP 流量。

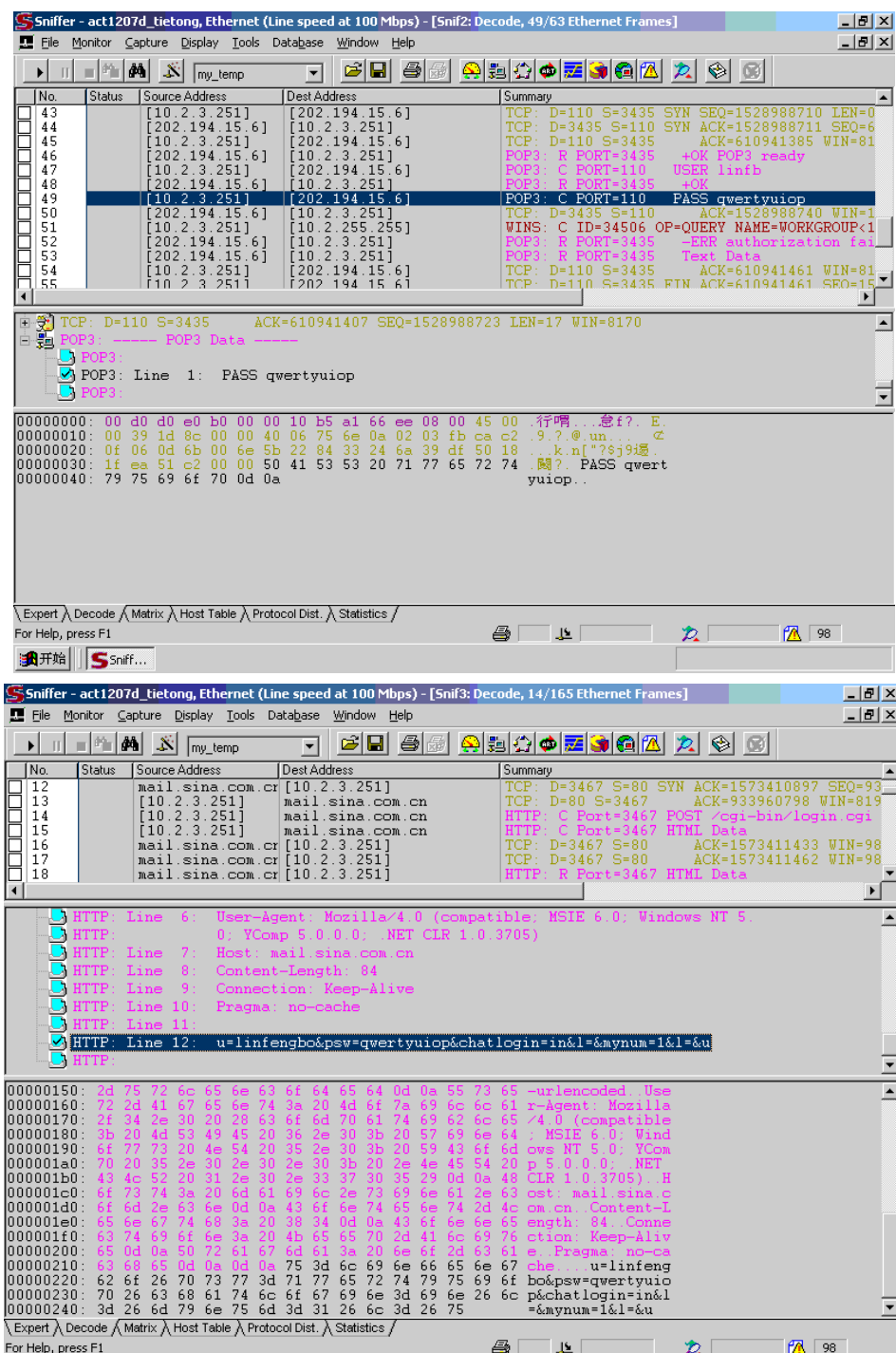


2.3.2 观察口令

推荐三类可以观察到口令的情形。

- (1) 邮件口令。使用 Outlook 或 Foxmail 收发邮件，可以捕获账户口令的明文。当然也可以看到邮件正文，不过大部分是编码过的，不能直接看懂。
- (2) Web 邮件口令。很多人习惯使用 Web 界面的邮件。捕获登录并查看 Web 邮件的流量，可以看到明文口令和邮件正文内容。
- (3) 登录注册论坛等。登录学习论坛，记录流量，一般可以观察到帐号和口令。

示例如下：



2.3.3 开发

在 Linux 下可以使用 libpcap 库编程实现包捕获功能。一个现成可参考的例子就是基于 libpcap 的 tcpdump，它是开源的。

Windows 下一般使用 Winpcap，它是 libpcap 的移植版本，大部分接口函数相同。Windump 基于 Winpcap 是 tcpdump 的翻版。Winpcap 是开源的，提供开发包以及示例程序的源文件。

请下载 winpcap 的开发包，编译并运行其中的示例程序，然后观察源程序及相关文档，试演并提炼出简短报告，内容包括：

- (1) winpcap 库的用法说明。
- (2) 一个包捕获程序的基本框架，要说明所调用的 winpcap 关键函数。
- (3) 选做：发送分组(帧)，实现假冒和重放。

2.3.4 要求和建议

一次实验：某楼层以太网，运行 Sniffer 约半个小时(10:30 2003-2-22)，实测捕得 10 万个包，约 100M 字节。经搜索“pass”、“psw”等关键字，发现有效的帐号/口令出现约有 8 次。这种威胁不仅发生在局域网上，显然也可以发生在分组经由的沿途交换设备及其网段上。

建议：

(1) 为了看清关心的流量，在逮包期间尽量关闭其它无关程序，尤其是 P2P 下载等程序。

(2) 某些 Windows 版本中（如 2000 服务器版）自带的网络监视器就是一个 Sniffer 程序，可以直接使用。

(3) tcpdump/windump 是命令行界面的 Sniffer 工具程序，灵活方便，功能强大，建议了解并试用。

思考如何保护网络传输数据安全：

- (A) 抵御窃听；(B) 发现窃听；(C) 不被察觉地窃听。

2.3.5 示例程序 udpdump.c

一个基于 winpcap-3.1 的逮包程序的框架，请参见 udpdump.c in WpdPack_3_1.zip, at <http://www.winpcap.org/>。

实验三 对称加密算法实验

对称加密算法曾经是唯一，后来才有了非对称算法（公钥算法）。现在一般使用公钥算法做鉴别、认证以及协商会话密钥，然后使用对称算法加密批量数据。

3.1 实验目的

- (1) 了解对称算法的基本工作流程。
- (2) 掌握对称算法的使用方法。

3.2 实验内容

- (1) 认识 OpenSSL 工具包。
- (2) 用简短的程序代码演示：分组加密算法（DES、AES）和流密码算法（RC4）的使用，其中包括分组算法的四种应用模式 ECB、CBCCFB、OFB。
- (3) 编写一个简单但是安全的文件加密程序。

3.3 实验指导

分组算法在程序实现时一般首先被抽象为这样一个基本函数：

$$F(K, IN, ENC/DEC) = OUT$$

其中 IN 是固定长度的输入分组（DES 算法是 8 字节，AES 是 16 字节），OUT 是同样长度的输出分组，密钥 K 是随机比特串（DES 是 56 比特，AES 是 128 比特）。

OpenSSL 中实现的对称算法太多，重点关注三个：DES、AES、RC4。

3.3.1 DES 算法函数接口

在 OpenSSL 中，DES 算法的基本函数就是 ECB 操作模式对应的函数 `DES_ecb_encrypt()`。该函数把一个 8 字节明文分组 `input` 加密成为一个 8 字节密文分组 `output`。参数中密钥结构 `ks` 是用函数 `DES_set_key()` 准备好的，而密钥 `key` 是用随机数算法产生的 64 个随机比特。参数 `enc` 指示是加密还是

解密。该函数每次只加密一个分组，因此用来加密很多数据时并不方便。

```
void DES_ecb_encrypt(const_DES_cblock *input, DES_cblock *output,  
DES_key_schedule *ks, int enc);  
  
int DES_set_key(const_DES_cblock *key, DES_key_schedule  
*schedule);
```

DES 算法 CBC 操作模式加解密函数是 DES_ncbc_encrypt()。参数 length 指示输入字节长度。如果长度不是 8 字节的倍数，则会被用 0 填充到 8 字节倍数。因此，输出可能比 length 长，而且必然是 8 字节的倍数。

```
void DES_ncbc_encrypt(const unsigned char *input, unsigned char  
*output, long length, DES_key_schedule *schedule, DES_cblock *ivec,  
int enc);
```

DES 算法 CFB 操作模式加解密函数是 DES_cfb_encrypt()。参数 length 指示输入字节长度。参数 numbits 则指示了 CFB 每次循环加密多少明文比特，也即密文反馈的比特数目。ivec 是初始向量，被看做第 0 个密文分组，是不用保密但应随机取值的 8 个字节。如果在一次会话中数次调用 DES_cfb_encrypt()，则应该记忆 ivec。由于 CFB 模式中每次 DES 基本操作只加密 numbits 比特明文，因此如果 numbits 太小则效率太低。

```
void DES_cfb_encrypt(const unsigned char *in, unsigned char *out,  
int numbits, long length, DES_key_schedule *schedule, DES_cblock  
*ivec, int enc);
```

另有一个 numbit 是 64 比特的版本，既高效又没有填充的麻烦，推荐使用。num 中的返回值指示了 ivec 中的状态，是和下次调用衔接的。

```
void DES_cfb64_encrypt(const unsigned char *in, unsigned char *out,  
long length, DES_key_schedule *schedule, DES_cblock *ivec, int *num,  
int enc);
```

OFB 和 CFB 类似，也有两个函数，用法一样。

```
void DES_ofb_encrypt(const unsigned char *in, unsigned char  
*out, int numbits, long length, DES_key_schedule *schedule, DES_cblock  
*ivec);
```

```
void DES_ofb64_encrypt(const unsigned char *in, unsigned char  
*out, long length, DES_key_schedule *schedule, DES_cblock *ivec, int
```

```
*num);
```

3.3.2 AES 算法函数接口

典型参数的 AES 的基本操作是把 128 比特明文在 128 比特密钥指引下加密成 128 比特密文。OpenSSL 中关于 AES 的函数名和参数接口和 DES 的雷同。相关函数名如下(参数略)。

```
int AES_set_encrypt_key();
int AES_set_decrypt_key();
void AES_ecb_encrypt();
void AES_cbc_encrypt();
void AES_cfb128_encrypt();
void AES_ofb128_encrypt();
```

3.3.3 RC4 算法函数接口

RC4 密码算法是流算法，也叫序列算法。流算法是从密钥作为种子产生密钥流，明文比特流和密钥流异或即加密。RC4 算法由于算法简洁，速度极快，密钥长度可变，而且也没有填充的麻烦，因此在很多场合值得大力推荐。

OpenSSL 中 RC4 算法有两个函数：RC4_set_key() 设置密钥，RC4() 加解密。可以把 RC4 看作异或，因此加密两次即解密。

```
void RC4_set_key(RC4_KEY *key, int len, const unsigned char
*data);

void RC4(RC4_KEY *key, unsigned long len, const unsigned char
*indata, unsigned char *outdata);
```

3.3.4 一个文件加密例子程序

一个最简单的（但是相当安全的）文件加密程序例子，参见“rc4.zip”和“myrc4.zip”。

3.3.5 例子程序(仅作参考)

DES 示例程序	 Demo_des.zip	
AES 示例程序	 Demo_aes.zip	
RC4 示例程序	 enc.c	需要使用 OpenSSL 库
	 Rc4.zip	RC4 算法从 OpenSSL 中分离出来了，不需要 OpenSSL 库
	 Myrc4.zip	一个参考 AC2ED 的教学实现，很慢

3.3.6 提示和建议

基于口令的文件加密的安全性依赖于几个方面：

- (1) 算法选择。需要使用经过考验的公开的加密算法，如 DES/AES/RC4 等；
- (2) 口令的质量，以及从口令衍生密钥的方法。建议使用 PKCS#5 或者类似的思路从口令产生密钥。
- (3) 发现错误的密文或口令保护。在加密前对明文添加校验保护，比如添加 MD5/SHA1 校验值，如果解密后校验不符则可以断定密文有篡改或口令不对。

做一个练习题。如何评价市面上有的“高强度文件夹加密”软件声称“上百 G 的数据仅需 1 秒钟完成”？请进一步查阅网络资料完成分析。

实验四 公钥算法实验

使用公钥密码算法可以克服协商对称密钥的困难，也可以用来认证和签名。为了避免公钥算法的速度缺陷，当前普遍使用混合密码体制，即使用公钥算法做鉴别和协商会话密钥，使用对称算法加密批量数据。

公钥密码算法当前仍是 RSA 算法占统治地位。OpenSSL 加密函数库中提供了对 RSA 等算法的支持。

4.1 实验目的

- (1) 掌握公钥算法的使用方法
- (2) 掌握混合密码体制的工作原理

4.2 实验内容

- (1) 使用 RSA 算法加解密和签名验证
- (2) 使用混合密码体制的文件加密

4.3 实验指导

不同于对称加密算法中加密和解密使用同样的密钥，公钥算法分为加密密钥 K1 和解密密钥 K2 两部分，而且从 K1 很难计算推导出 K2。这样就可以保密 K2 而公布 K1，从而大大简化了密钥管理。习惯上 K1 称为公钥，K2 称为私钥。

加密使用公钥，解密使用私钥。

$$\text{ENC}(P, K1) = C$$

$$\text{DEC}(C, K2) = P$$

RSA 加密算法的步骤是这样的：

- (1) 找两个随机大素数 p 和 q ；
- (2) 计算模 $n=pq$ 和 Euler 函数 $\phi(n) = (p-1)(q-1)$ ；
- (3) 选取数 e 后用扩展 Euclid 算法求数 d 满足 $ed \equiv 1 \pmod{\phi(n)}$ ；
- (4) 保密私钥 $K2=(d, n)$ ，发布公钥 $K1=(e, n)$ ；
- (5) 加密明文 p 时，计算密文 $c = p^e \pmod{n}$ ；

(6) 解密 c 时, 计算 $p = c^d \bmod n$ 。

RSA 算法也可以用来签名:

(7) 对消息 m , 其签名 $s = m^d \bmod n$;

(8) 验证 (m, s) 即判断 $m \stackrel{?}{=} s^e \bmod n$ 。

下面介绍 OpenSSL 中 RSA 算法的函数接口。

4.3.1 RSA 密钥产生

RSA 密钥产生函数 `RSA_generate_key()`, 需要指定模长比特数 `bits` 和公钥指数 `e`。另外两个参数为 `NULL` 即可。

```
RSA * RSA_generate_key(int bits, unsigned long e, void
(*callback) (int,int,void *),void *cb_arg);
```

目前对于长达 663 比特的 RSA 模数已经有成功分解的先例, 因此当前典型的应用场合使用 1024 比特模长的 RSA 算法, 此时一个分组是 128 字节。

如果从文件中读取密钥, 可使用函数 `PEM_read_bio_PrivateKey()`/`PEM_read_bio_PUBKEY()`, 其中 `EVP_PKEY` 中包含一个 RSA 结构, 可以引用。

```
EVP_PKEY *PEM_read_bio_PrivateKey(BIO *bp, EVP_PKEY **x,
pem_password_cb *cb, void *u);
```

4.3.2 RSA 加密和解密

RSA 加密函数 `RSA_public_encrypt()` 使用公钥部分, 解密函数 `RSA_private_decrypt()` 使用私钥。填充方式常用的有两种 `RSA_PKCS1_PADDING` 和 `RSA_PKCS1_OAEP_PADDING`。出错时返回 -1。输入必须比 RSA 钥模长短至少 11 个字节 (在 `RSA_PKCS1_PADDING` 时?)。输出长度等于 RSA 钥的模长。

```
int RSA_public_encrypt(int flen, const unsigned char
*from,unsigned char *to, RSA *rsa,int padding);
```

```
int RSA_private_decrypt(int flen, const unsigned char
*from,unsigned char *to, RSA *rsa,int padding);
```



4.3.3 RSA 签名和验证

签名使用私钥, 验证使用公钥。RSA 签名操作是把被签署消息的散列值编

码后用私钥加密，因此函数中参数 type 用来指示散列函数的类型，一般是 NID_md5 或 NID_sha1。正确情况下返回 0。

```
int RSA_sign(int type, const unsigned char *m, unsigned int
m_length, unsigned char *sigret, unsigned int *siglen, RSA *rsa);
int RSA_verify(int type, const unsigned char *m, unsigned int
m_length, unsigned char *sigbuf, unsigned int siglen, RSA *rsa);
```

4.3.4 RSA 算法例子程序(供参考)

RSA 加解密例子	 demo_rsaenc, v2. zip	
RSA 签名和验证	 demo_sign, v2. zip	

4.3.5 设计：签名并加密(选做)

在公钥体制中，每个人都有一对公钥私钥，可以使用这对密钥加密保护自己的文件。通常使用对称算法加密文件，而用公钥算法加密对称算法的密钥，此即混合密码体制。一般使用随机数产生函数（OpenSSL 库中有）产生随机比特用做对称密钥 Key，选择使用 DES/AES/RC4 等算法加密文件正文，而使用 RSA 算法加密对称密钥。公式化表示为：

$$RSA (Key, K1) || RC4 (Message, Key)$$

解密时先用私钥解密得到 Key，再用 Key 解密得到 Message。

如果想对消息签名以发现变化或防篡改，通常则使用私钥对原文的散列值进行签名，签名值和原文一起存放：

$$Message || RSA (Message, K2)$$

签名和加密可以结合起来，比如先用自己的私钥签名，再用对方的公钥加密，这就是 PGP 加密邮件系统的基本思路。

关于私钥的保密问题。私钥一般使用口令保护（参见前一个实验），并且操作系统往往也限制对私钥密文的访问。

4.3.6 要求和建议

根据上面讲述的思路，进一步细化方案，并争取能编程实现这样一个功能的程序。

除了 OpenSSL 之外，还有一些其它的环境可以支持对称及非对称加密算法，比如 Java/JDK 中。不熟悉 C/C++ 而擅长 Java 的可以考虑使用 Java 环境实现本实验题目。

关于随机数产生和散列函数，可以提前看一下相关资料。

实验五 微软 CA 组件安装和配置

5.1 实验目的

通过安装和配置虚拟机的 WINDOWS2000 SERVER 操作系统中集成的 CA 组件,实现浏览器证书和 WEB 证书的签发,进一步了解微软公司的 PKI 技术框架,并能够使用该模块,在实际系统中进一步应用(也可以在安装 WINDOWS 2000 SERVER 的时候一并选择安装).

5.2 实验内容

1. 安装 CA 组件。
2. 配置 CA 组件。
3. 申请证书、签发证书、下载证书等。

5.3 实验指导

证书授权服务器是 Windows 2000Server 的一个附件,它放在 Windows 2000 Server 的安装盘上。默认安装 WINDOWS 操作系统时是不安装的(除非特别选择)。它可让你为建立和管理 X509 版本 3 的数字证书创建一个定制的服务以作证书之用。你可以为 Internet 或者公司的内部网创建服务器证书,从而可让你的组织完全控制它自己的证书管理策略。

设置 CA:

1. 单击“开始”,指向“设置”,然后单击“控制面板”。
2. 双击“添加/删除程序”。
3. 单击“添加/删除 Windows 组件”。
4. 单击“下一步”。
5. 单击“证书服务”复选框,将其选中,然后单击“下一步”。
6. 单击相应的 CA 类型。可用选项右侧显示每个颁发机构的说明。
7. 如果要更改默认密码设置,请单击“高级选项”复选框,将其选中。只有在您知道的确需要这么做时,才选中此复选框。
8. 单击“下一步”。
9. 如果选中了“高级选项”复选框,系统将提示您更改“公钥和私钥对”选项。如果未选中“高级选项”复选框,则继续执行下一步。
10. 显示证书颁发机构标识信息窗口。填入相应的站点和组织信息。请注意,CA 信

息非常重要，因为要用它来标识所创建的 CA 对象。完成后单击 “下一步” 。

11. 系统会提示您定义证书数据库的位置、配置信息和 “证书吊销列表” (CRL)。

企业 CA 始终将其信息 (包括 CRL) 存储在 Active Directory 中。Microsoft 建议您选中 “共享文件夹” 复选框。这样就指定了存储 CA 配置信息的文件夹位置。所有 CA 配置信息都应该存储在一个文件夹中。

12. 单击 “下一步” 。

13. 如果 IIS 处于运行状态，请将其关闭。单击 “确定” 以停止 IIS。安装 Web 组件之前必须先停止 IIS。如果没有安装 IIS，则继续执行下一步。

14. 安装次级 CA 时，您需要：单击 “浏览” ，找到联机 CA，或者，如果您的请求是定向到专用 CA 或无法从网络访问的 CA，则需要单击 “将申请保存到一个文件” 。

15. 等待安装完成。

16. 单击 完成 。

验证 “证书服务器” 安装：

要验证安装，您可以使用以下任一方法：

(1) 在命令提示下键入 `net start` ，确认 “证书” 服务正在运行。

(2) 申请证书，方法是单击 “开始” ，指向 “运行” ，键入 `mmc` ，单击 “确定” ，在 “控制台” 菜单上单击 “添加/删除管理单元” ，添加 证书 管理单元，单击要管理的 我的用户帐户 ，右键单击 个人 文件夹，单击 所有任务 ，然后单击 “申请新证书” 。“证书申请向导” 应该启动。

(3) 对于独立 CA，您可以使用 Internet Explorer 5 连接到 “[http:// 服务器名 /CertSrv](http://服务器名/CertSrv)” (其中 服务器名 是服务器的名称)，以此来申请新证书。

具体证书申请方法，请参见帮助内容。

实验六 EJBCA 系统的安装和配置

6.1 实验目的

通过 EJBCA 系统的安装和配置，一方面了解大型软件的基本安装步骤，更重要的是了解和掌握作为一个典型的大型 CA 系统软件，其功能模块是如何构成的，实现的基本功能有哪些，掌握其优缺点，并能根据目前的应用环境，写出一个结构性分析报告，尽量提出进一步完善的方面，使之更适合实际应用的需要。

6.2 实验内容

独立完成 EJBCA 系统的各个模块的安装和配置，并能完成多种数字证书的申请、签发、下载、注销，黑名单的生成、下载等 CA 基本功能。

实验所需的具体软件环境和所需的软件包

VMWARE 虚拟硬件环境下的 WINDOWS2000 操作系统，作为基本安装环境。

EJBCA 完全采用 Java 编写，能够在任何采用 J2EE 服务器的平台上运行。开发和测试是在 Linux 和 Windows2000 上进行的。本文主要介绍针对于 Windows XP SP2 操作系统平台的安装、部署与应用。

构筑系统默认的 configuration 平台，所需软件除 ejbca_3_0_2.zip (<http://ejbca.sourceforge.net>)外，尚需如下软件：

1. j2sdk-1_4_1_01-windows-i586.exe (<http://www.sun.com/>)

JDK 是 Sun 公司开发的 Java 虚拟机，及 Java 开发工具，j2sdk-1_4_1_01-windows-i586.exe 是 JDK1.4 版本的 Windows 平台安装软件。JDK 目前最新版本为 1.5，但 ejbca_3_0_2 在 JDK1.5 平台下，编译、部署、安装会出现诸多异常，建议使用 JDK1.4。

2. jboss-3.2.5.zip (<http://www.jboss.org/>)

JBoss 是一个运行 EJB 的 J2EE 应用服务器。它是开放源代码的项目，遵循最新的 J2EE 规范。从 JBoss 项目开始至今，它已经从一个 EJB 容器发展成为一个基于的 J2EE 的一个 web 操作系统（operating system for web），它体现了 J2EE 规范中最新的技术，并且它还在 the JavaWorld Editors' Choice 2002 评选中获得“最佳 Java 应用服务器”大奖。现在 SUN 公司已经把 JBOSS 作为 J2EE1.4 的标准实现服务器。

3. jce_policy-1_4_2.zip (<http://www.sun.com/>)

JCE 是 Java(TM) Cryptography Extension 的简写。由于进口控制的限制，J2SDK 绑定的 JCE policy 文件，仅允许有限的加密强度。所以，运行 EJBCA 需要下载无

加密强度限制的加密扩展文件。

4. apache-ant-1.6.2-bin.zip

Apache Ant 是一个基于 Java 的生成工具，这个工具的名称是 another neat tool（另一个整洁的工具）的首字母缩写。Ant 与 C 或 C++ 中的传统项目经常使用 make 工具类似，它也定义生成文件之间的依赖关系；然而，与使用特定于平台的 shell 命令来实现生成过程所不同的是，它使用跨平台的 Java 类。使用 Ant，您能够编写单个生成文件，这个生成文件在任何 Java 平台上都一致地操作，Ant 的其他关键优势包括其突出的简单性和无缝地使用自定义功能来扩展它的能力。

6.3 实验指导

EJBCA 软件是 OSI 认证的开源软件，它是一个全功能的 CA 系统软件，它基于 J2EE 技术，并提供了一个强大的、高性能并基于组件的 CA。EJBCA 兼具灵活性和平台独立性，能够独立使用，也能和任何 J2EE 应用程序集成。它具有以下特征：

LGPL 开源许可

建立在 J2EE 1.3（EJB2.0）规范之上

灵活的、基于组件的体系结构

多级 CA

多个 CA 和多级 CA，在一个 EJBCA 实例中建立一个或者多个完整的基础设施单独运行，或者在任何 J2EE 应用中集成它

简单的安装和配置

强大的基于 Web 的管理界面，并采用了高强度的鉴别算法

支持基于命令行的管理，并支持脚本等功能

支持个人证书申请或者证书的批量生产

服务器和客户端证书能够采用 PKCS12, JKS 或者 PEM 格式导出

支持采用 Netscape, Mozilla, IE 等浏览器直接进行证书申请

支持采用开放 API 和工具通过其它应用程序申请证书

由 RA 添加的新用户可以通过 email 进行提醒

对于新用户验证可以采用随机或者手工的方式生成密码

支持硬件模块，来集成硬件签发系统（例如智能卡）

支持 SCEP

支持用特定用户权限和用户组的方式来进行多极化管理

对不同类型和内容的证书可以进行证书配置

对不同类型的用户可以进行实体配置

遵循 X509 和 PKIX(RFC3280)标准

支持 CRL

完全支持 OCSP，包括 AIA 扩展

CRL 生成和基于 URL 的 CRL 分发点遵循 RFC3280，可以在任何 SQL 数据库中存储证书和 CRL（通过应用服务器来处理）。

可选的多个发布器，用来在 LDAP 中发布证书和 CRL
支持用来为指定用户和证书来恢复私钥的密钥恢复模块
基于组件的体系结构，用来发布证书和 CRL 到不同的目的地
基于组件的体系结构，用来在发布证书时采用多种实体授权方法
容易集成到大型应用程序中，并为集成到业务流程进行了优化

可以通过访问 <http://ejbca.sourceforge.net/> 获得关于 EJBCA 的详细资料。

(i) 准备工作:

- 1) 为管理方便，新建文件夹“EJBCA”（如 F:\EJBCA），作为安装文件的根目录，以后安装的所有相关程序，均在此目录下；
- 2) 运行 j2sdk-1_4_1_01-windows-i586.exe，安装 J2SDK；设置安装路径 F:\EJBCA\j2sdk1.4；
- 3) 解压缩 jce_policy-1_4_2.zip，得到一个 JCE 文件夹，将其中的 local_policy.jar 和 US_export_policy.jar 文件复制到 C:\Program Files\Java\j2re1.4.1_01\lib\security 路径及 F:\EJBCA\j2sdk1.4\jre\lib\security 路径，覆盖原先的同名文件即可；
- 4) 解压缩 apache-ant-1.6.2-bin.zip 到安装路径 F:\EJBCA 下；
- 5) 解压缩 jboss-3.2.5.zip 到安装路径 F:\EJBCA 下；
- 6) 解压缩 ejbca_3_0_2.zip 到安装路径 F:\EJBCA 下；
新建环境变量 JAVA_HOME、CLASSPATH、ANT_HOME、JBOSS_HOME、EJBCA_HOME，并根据安装路径具体设置；如设置为：
JAVA_HOME=F:\EJBCA\j2sdk1.4
CLASSPATH=F:\EJBCA\j2sdk1.4\jre\lib\security;.;F:\EJBCA\j2sdk1.4\lib;F:\EJBCA\jboss-3.2.5\server\default\lib;
ANT_HOME= F:\EJBCA\apache-ant-1.6.2
JBOSS_HOME=F:\EJBCA\jboss-3.2.5
EJBCA_HOME= F:\EJBCA\ejbca
- 7) 编辑环境变量 Path，在其中加入 JDK, ANT, JBOSS 的 bin 路径,本文设置后为：
Path= %SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;F:\EJBCA\j2sdk1.4\bin;F:\EJBCA\jboss-3.2.5\bin;F:\EJBCA\apache-ant-1.6.2\bin;F:\EJBCA\openldap;F:\EJBCA\mysql\bin;

(ii) ant build.xml 过程

点击‘开始’菜单，运行 cmd，打开控制台窗口，切换到 ejbca 的安装目录，运行 ant 命令。ant 会根据 ejbca 目录下的 build.xml 文件,创建、编译、打包、复制相关文件，并最后提示 BUILD SUCCESSFUL。

(iii) ant deploy 过程

运行 ant 顺利通过后，键入 ant deploy 命令。最终提示你：BUILD SUCCESSFUL。这时你的 JBOSS_HOME\server\default\deploy 下面发现多了 ejbca-ca.ear 文件，ejbca-ca.ear 是上一步命令运行中 ca.ear:时生成的，这个文件里包含了 ant 时打的所有包。启动 JBOSS 你会发现有 ejbca-ca.ear 部署成功的信息，在启动 JBOSS 的过程中，JBOSS 会对 ejbca-ca.ear 作一些处理。

(iv) install 过程

在安装之前必须启动 JBOSS 服务器，并确保贮备工作中的第三点 JCE 文件配置无误；否则，install 命令会给出相应的提示，并停止安装工作。

在命令行下键入 run 命令。经过一系列的配置加载，最终命令行提示，表明 JBOSS 服务启动结束。这时在浏览器地址栏中键入 <http://localhost:8080> 或 http://(本机 IP 或域名): 8080,会有如下提示：



在命令提示符下，接着键入 install 命令：（粗体为系统提示）

Welcome to EJBCA Installation

This script acts as a wizard helping you with the installation of your Certificate Authority.

Before the installation will begin make sure of the following preparations havebeen done:

1. The EJBCA application is deployed to the application server. ('ant deploy')

2. You run this installation with access to administrative privileges.
Is these requirements meet (Yes/No) :

程序提示必须经过 ant deploy 将 EJBCA 部署到应该服务器，并且具有 administrative 权限，键入 **Yes** 将继续下一步的安装配置。

This installation will create a first administrative CA. This CA will be used to create the first super administrator and for the SSL server certificate of administrative web server.

When the administrative web server have been setup you can create other CAs and administrators.

Please enter the short name for the CA.

This is only used for administrative purposes,

avoid spaces or odd characters (Ex 'AdminCA1') : DragDragonCA①

Enter the Distinguished Name of the CA. This is used in the CA certificate to distinguish the CA. (Ex 'CN=AdminCA1,O=PrimeKey Solutions

AB,C=SE') :CN=DragDragonCA, O=QLSC, C=CN②

Enter the keysize in bits of the CA, only digits. (Ex '2048') :2048③

Enter the validity in days for the CA, only digits (Ex '3650') :3650④

Enter the policy id of the CA. Policy id determine which PKI policy the CA uses.

标有数字的下划线部分，是要设置的地方。其中①处为设置创建的 CA 的管理名称，仅为管理之用。②处为填写所创建 CA 的唯一甄别名 (**Distinguished Name**)，此名称将会出现在 CA 根证书及其所签发的所有证书中。①处的名称与②处的 CN 名称不必相同，因为前者为管理名称，而后者才是 CA 机构的对外公布名称。接下来将输入 CA 的密钥长度合 CA 的有效期。

Type your policy id or use '2.5.29.32.0' for any policy or 'NO' for no policy at all. (Ex '2.5.29.32.0') : 2.5.29.32.0⑤

Now for some information required to set up the administration web interface.

在⑤处，提示输入 CA 系统的配置 ID，根据提示输入 **2.5.29.32.0** 即可，接着安装过程为 EJBCA 的 administrative web GUI 连接创建一个 SSL 服务器端证书，以备管理员通过 SSL 协议连接管理服务器验证之用。

Please enter the computer name of CA server.

(Ex 'caserver.primekey.se') :222.194.65.156⑥

Enter the Distinguished Name of the SSL server certificate used by the administrative web gui (Ex 'CN=caserver.primekey.se,O=PrimeKey Solutions AB,C=SE') :CN=222.194.65.156, O=QLSC,C=CN⑦

(ssl 服务器端证书的 DN)

Enter a good password for the super administrators keystore. Please remember this one: ⑧

⑥处为输入 CA 服务器的名称，可以输入服务器的 IP 地址或域名，也可输入 localhost 代指本机名称，为使管理员可以远程登录本机进行管理，建议输入域名或 IP 地址。⑦处为 SSL 服务端的证书唯一甄别名，⑧要求填写超级管理员证书库的密码，此密码要求安全可靠并不可忘记，将 SuperAdmin.p12 导入浏览器时将输入这个密码以操作证书库。

You have entered the following data :

CA short name : DragDragonCA

Distinguished Name CA : CN=DragDragonCA,O=QLSC,C=CN

Keysize of the CA : 2048

Validity in days for the CA : 3650

Policy id of the CA : 2.5.29.32.0

Computer name of CA server : 222.194.65.156

Distinguished Name of the SSL server certificate :

CN=222.194.65.156,O=QLSC,C=CN

Password for the super administrators keystore : *****

Is this correct (Yes/No/Exit) :Yes

The installation will now start, please wait

Initializing CA

Generating rootCA keystore:

DN: CN=DragDragonCA,O=QLSC,C=CN

Keysize: 2048

Validity (days): 3650

Policy ID: 2.5.29.32.0

Initializing Temporary Authorization Module.

Creating CA...

CAId for created CA: -1912334509

-Created and published initial CRL.

CA initialized

Setup of Administration Web Interface have started, this will take a minute to complete

认证已添加至 keystore 中。

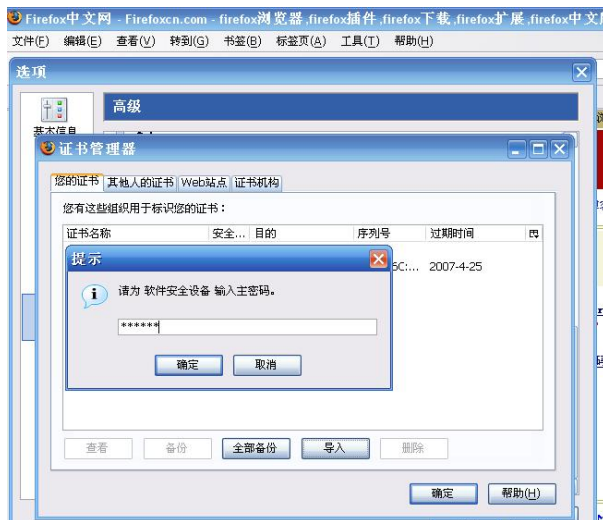
The installation is now complete. Proceed with the following steps in order to start administrating EJBCA.

1. Restart the application server.
2. Import the p12/superadmin.p12 file in your browser.
3. Go to the following URL: <https://:8443/ejbca/adminweb>
4. And now you are all set to start using EJBCA.

If you are interested in professional support of EJBCA and PKI related questions, please contact PrimeKey Solutions AB, Sweden at ejbca@primekey.se or www.primekey.se for more information.

这一段过程，提示你确认或重填你的设置，然后根据你的设置，创建 CA，生成 CA 的证书，初始化 user 状态，发布 CRL，把证书添加到 KEYSTORE 中，并初始化管理员界面等一系列活动。

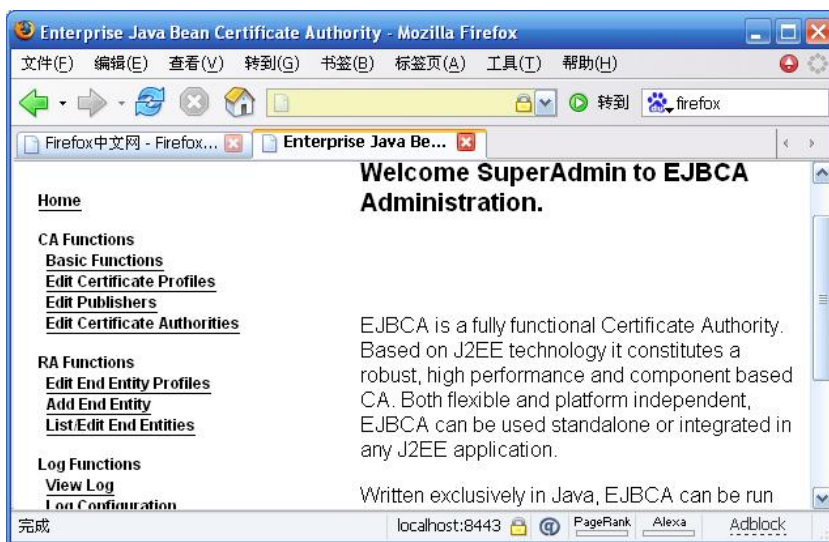
最后根据系统的提示，重启 JBOSS 服务器，将 p12 目录下的 superadmin.p12 文件导入到浏览器中，本人使用开放源码的 Web 浏览器 Firefox，导入界面如下：



在此需要输入安装过程中设定的证书库密码。在浏览器地址栏中键入相应的 URL 地址 <http://localhost:8080/ejbca>，出现系统登陆界面：



键入 `https://localhost:8443/ejbca/adminweb/index.jsp`，会提示你即将建立安全的 SSL 连接，并进入管理员界面：



至此，安装工作顺利完成。

EJBCA 系统的管理功能可以划分为四大板块，包括 CA 功能，RA 功能，日志功能和系统功能。

CA 功能：

1) Basic Functions

通过此功能模块，可以查看 CA 证书或相关信息，将根 CA 的证书下载到 IE 浏览器

或 Netscape 浏览器，或者下载为 pem 文件；并可以得知最新 CRL 的有效时间，获取 CRL 或更新 CRL。

2) Edit Certificate Profiles

该功能模块列举了系统默认的三种证书配置，最终用户、根 CA 和子 CA 证书配置。这三种配置是不可以更改或删除的，除此之外，管理员可以新建新的配置，并可以修改和删除自定义的配置。

3) Edit Publishers

通过此功能模块，管理员可以添加并配置新的 LDAP 服务器，作为证书发布点。

4) Edit Certificate Authorities

此功能单元，可以新建、编辑、删除 CA 中心。

RA 功能：

1) Edit End Entity Profiles

类似于 CA Functions 中的 Edit Certificate Profiles, 该功能单元可以对最终实体的证书类型进行相应的配置。

2) Add End Entity

此功能单元，有 RA 管理员来审核最终用户，并未最终用户录入相关数据，以备生产数字证书之用。

3) List End Entities

RA 管理员通过此功能单元，查询相关用户的某些信息，并进行相应的操作。

日志功能:

1) View Log

可以设定条件，查询相关日志。

2) Log Configuration

对系统的日志功能进行设置。

系统功能:

1) System Configuratin

对 Web 页面的显示进行设置。

2) Edit Administrator Privileges

对管理员权限进行相应的设置。

(V) 生产及获取证书

在 EJBCA 系统中，用户有两种方式可以获取数字证书，分别详述如下：

一、用户通过向 RA 注册相应信息，由 RA 审核并向 CA 提交用户信息，然后由 CA 对用户信息签名，生产相应的数字证书。

1. 用户向 RA 注册，RA 管理员通过系统 Add End Entity 功能单元审核并填写相关信息。如图，添加一用户名为 peng 的用户，

End Entity Profile	EMPTY	Required
Username	peng	<input checked="" type="checkbox"/>
Password	*****	<input checked="" type="checkbox"/>
Confirm Password	*****	
Batch	<input type="checkbox"/>	
Email	pengyanhai@mail.sdu.edu.cn	<input type="checkbox"/>
Subject DN Fields		
E, EmailAddress in DN	Use data from Email field : <input checked="" type="checkbox"/>	<input type="checkbox"/>
UID, Unique Id		<input type="checkbox"/>
CN, Common Name	pengyanhai	<input checked="" type="checkbox"/>
SN, Serial Number		<input type="checkbox"/>

详细填写相关信息后，点击页面下方的 Add End Entity 按钮，会显示
End Entity peng added successfully.

2. 用户到：

http://222.194.65.156:8080/ejbca/publicweb/apply/apply_main.jsp，输入用户名及密码，进行证书注册。

Welcome to certificate enrollment.

Please give your username and password, then click OK to generate your token.

Username:	peng
Password:	*****
<input type="button" value="OK"/>	

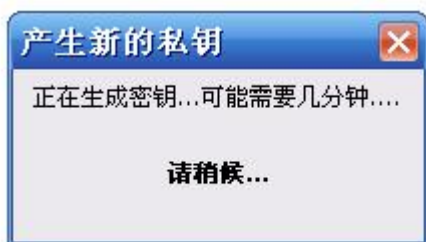
3. 获取数字证书，你可以手动安装证书链，或在获取证书的同时，自动安装证书链。

Install CA certificates:

• Certificate chain

Please choose keylength, then click OK to fetch your certificate.

Key length	2048 (高级)	<input type="button" value="OK"/>
------------	-----------	-----------------------------------



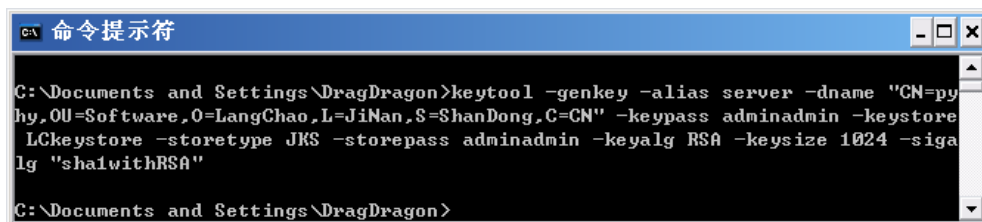
4. 到浏览器的证书管理器，可以看到新申请的证书，已经真确安装到浏览器中，并可通过“查看”按钮，查看证书的详细内容。



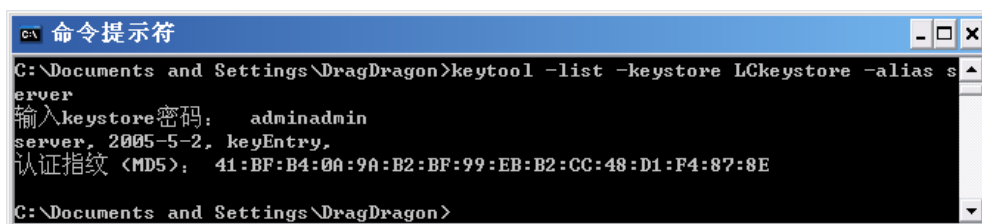
二、用户自己建立 Token，通过发送证书签名申请 CSR（Certificate Signing Request），CA 对发送信息签名，发送证书给用户。本文此过程用的 J2SDK 自带的 keytool 工具。

Keytool 是一个有效的安全密钥和证书的管理工具，它管理一个存储了私有密钥和验证相应公共密钥的与它们相关联的 X.509 证书链的 keystore(相当一个数据库)。它能够使用户使用数字签名来管理他们自己的私有/公共密钥对，管理用来作自我鉴定的相关的证书，管理数据完整性和鉴定服务，它还能使用户在通信时缓存它们的公共密钥。

1. 使用 keytool 工具的非交互模式建立密钥对，比如，要建立别名为 sever，通用名为 pyhy，公司为 LangChao，部门为 Software，城市为 JiNan，省份为 ShanDong，国家为 CN 的证书，把它存到了被密码 adminadmin 保护的密钥库 LCkeystore 的密钥库中，密钥长度为 1024bit



2. 查看证书库，可见证书已生成



3. 生成证书签发申请 CSR 文件

```

C:\命令提示符
inadmin -keystore LCKeystore -storepass adminadmin -storetype JKS -file pyhy_certreq.pem -sigalg "SHA1withRSA"

C:\Documents and Settings\DragDragon>

```

4. 到 keytool 运行目录下，查找 pyhy_certreq.pem 文件，用纯文本方式打开，如图所示：

```

-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBPTCCAQ4CAQAwZTELMAkGA1UEBhMCQ04xETAPBgNVBAGTCFNoYW5Eb25nMQ4wDAYDVQQHEwVK
aU5hbGJERMA8GA1UEChMITGFuZ0NoYW8xETAPBgNVBAsTCFNoYW5Eb25nMQ4wDAYDVQQDEwRQWU
hZMIHGMAGCSqGSIb3DQEBAQUAA4GNADCBiQKBggQClEBQe5pKj/dVV3cY0m4R79sAbmLTSwq+iCbTu
mcP+J0gwq6EHgc/ZwhJX9qbW+FxL4wI0epN4H0sIVigpnXzL35Pr2chvblWNvDcWZpI/SGSF8bvx
4zRayWeGt07JeZ8LZydtu0fh8zzZ79Ty1F3KulnttHuJnppCJVT24/P42wIDAQABoAAwDQYJKoZI
hvcNAQEFBQADgYEAaVn+aIMjiQWDYysxFEVHaF1neY0UxByBYIZY1m05wsixCXjvu4obZJG6Nmi
+mWLBxSjRk3zLDbENFX0op+d4ur5n40Lb3DWjVmsUSrDDHEu9w10zKMMIjsG9c5YVBxJRvBXfHNZ
yUa8/f5vKn79oS3C+wtcKqOSJy33EDkFN5w=
-----END NEW CERTIFICATE REQUEST-----

```

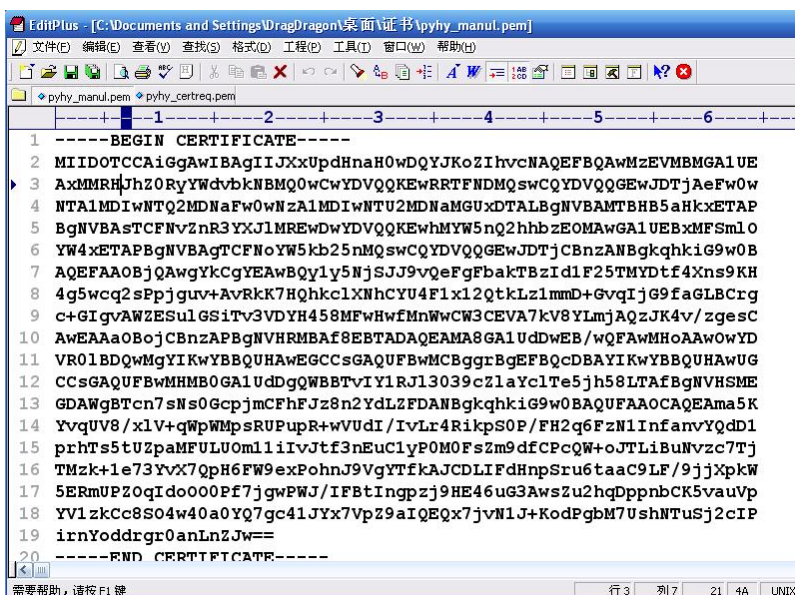
5. 在 CA 的管理端建立用户 user001，cn=pyhy 的用户

End Entity Profile EMPTY	Required
Username user001	
Password *****	
Confirm Password *****	
Batch <input type="checkbox"/>	
Email <input type="text"/> @	<input type="checkbox"/>
Subject DN Fields	
E, EmailAddress in DN Use data from Email field: <input type="checkbox"/>	<input type="checkbox"/>
UID, Unique Id <input type="text"/>	<input type="checkbox"/>
CN, Common Name pyhy	<input checked="" type="checkbox"/>
SN, Serial <input type="text"/>	<input type="checkbox"/>

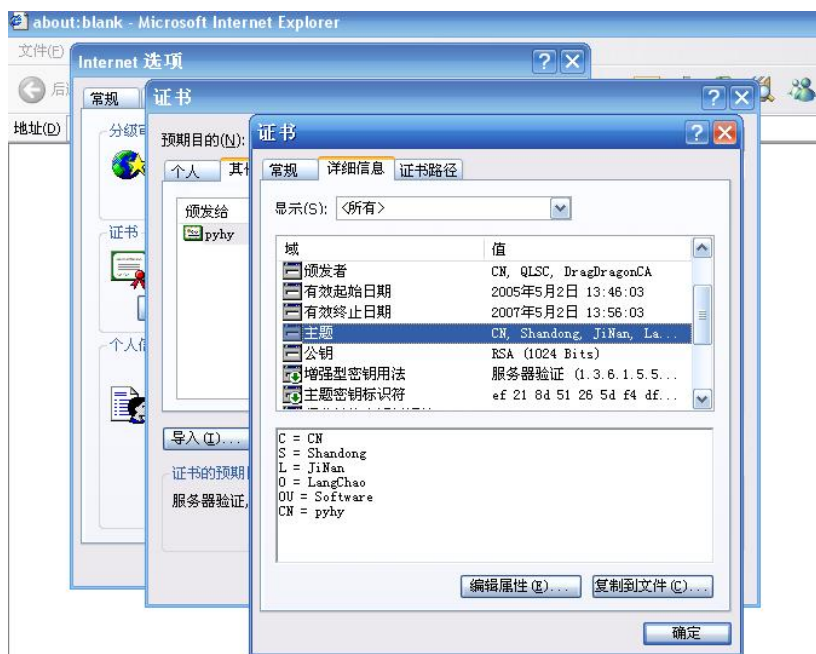
6. 打开页面 http://localhost:8080/ejbca/publicweb/apply/apply_man.jsp 将 PYHY_certreq.pem 中的内容粘贴到中的文本框中，输入用户名密码取得证书。



7. 将 cert.pem 保存到磁盘，用纯文本方式打开，可见文件内容与原先生成的已有很大的区别。



8. 将下载的证书文件导入到 IE 浏览器，可以查看证书的内容：



其他功能在此不赘述。

实验七 SureCA 系统的安装和配置

7.1 实验目的

通过 SureCA 系统的安装和配置，一方面了解大型软件的基本安装步骤，更重要的是了解和掌握作为一个典型的大型 CA 系统软件，其功能模块是如何构成的，实现的基本功能有哪些，掌握其优缺点，并能根据目前的应用环境，写出一个结构性分析报告，尽量提出进一步完善的方面，使之更适合实际应用的需要。

7.2 实验内容

独立完成 EJBCA 系统的各个模块的安装和配置，并能完成多种数字证书的申请、签发、下载、注销，黑名单的生成、下载等 CA 基本功能。

《Sure 网络安全认证中心（CA）系统》（以下简称《CA 系统》）是一套遵照 PKI 技术规范设计的，为一般电子商务系统网络用户提供基于公开钥密码技术的身份认证的，为用户提供公开密钥和数字证书管理服务的综合系统。它通过采用并实现符合国际标准的公开钥和对称钥密码算法，以及数据加密、解密、数字签名、签名验证等核心技术，构造了符合国际相关标准的认证体系，实现了 PKI 技术的核心基础设施---CA 系统。

《CA 系统》采用了符合国际潮流的设计方案。其基本网络拓扑结构为客户/服务器（C/S）模式的，其应用模式为 RA 系统/CA 系统。其中 RA 系统主要完成与安全域用户的交互，负责用户认证服务请求（如委托证书/密钥请求、证书注销请求、证书/密钥恢复请求等）资格的审查与确认，以及认证结果（数字证书和密钥/CRL 等）的发放和管理。CA 系统不直接与客户交互，而是与各个合法的 RA 系统交互，接收 RA 系统审核过的用户认证服务请求，执行认证服务，返回认证结果。

《CA 系统》中牵扯的技术主要包含各种公开钥、对称钥和摘要算法的底层实现，数据加密、解密、签名、验证签名运算的标准化，数字信息的 ASN1 描述与 DER 数据编码实现，数字证书、黑名单的标准化编码、解码，数据安全通信协议的设计与实现，目录服务器应用技术实现，数据库接口实现，用户界面的实现等。

(1) 实验所需的具体软件环境和所需的软件包

在 VMWARE win2000 环境下,导入已经安装但尚未配置过的系统软件 SureCA 系统的服务器端 CA 和客户端 RA 软件包和 SQL SERVER 数据库以及 NETSCAPE 目录服务器软件包。

(2) 进行 CA 端软件、RA 端软件、通信服务器软件、数据库、目录服务器软件的配置。

(3) 进行两种方式（委托证书申请和 PKCS10）的数字证书申请、签发、下载、安装。

7.3 实验指导

该软件是一个具有较完善功能的 CA 系统。采用 C / S 结构设计。RA 端是注册申请证书的操作端，可以分布在不同地理位置，面向用户服务；CA 端是该系统的核心部分，负责将 RA 端上传的信息进行处理。CA 端系统和 RA 端系统通过网络互连。系统软件有两个软件安装包，一个是 CA 系统软件包，一个是 RA 系统软件包。CA 端软件安装后包括两部分，一个是后台管理软件（负责 CA 本地配置，策略制定，RA 管理，密钥生成，证书签发，证书注销，证书归档，证书查询，生成和发布 CRL，权限管理等核心服务），一个是通信服务器（通信服务器负责接受来自各个 RA 的业务申请）。RA 端系统只包含一套管理软件，功能是完成多种方式的证书业务信息录入、审核、上传、下载等任务。详细功能见软件使用说明书。

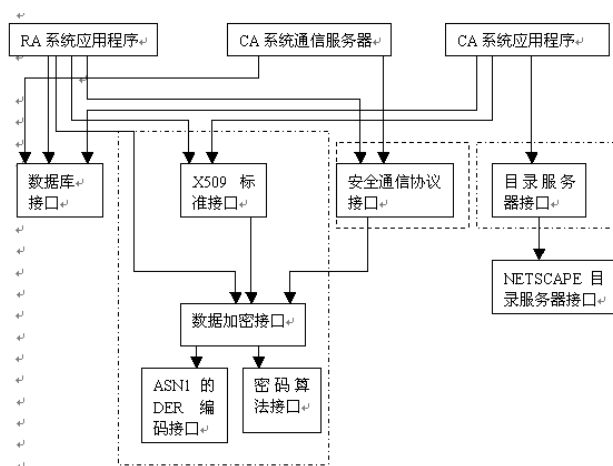
对于一个典型的 CA 综合系统，应当安装 CA 管理中心系统安装软件包和至少一套 RA 登记中心系统安装软件包，根据实际情况，可以安装多套 RA 登记中心系统安装软件包，形成典型的 C/S 应用模式。

CA 管理中心系统安装软件包是安装 CA 系统后台管理系统和 CA 系统通信服务器两部分，这两部分共同配合完成 CA 中心端的核心功能，RA 登记中心系统安装软件包完成 CA 系统前台登记功能，直接面向用户。

对于 CA 中心端系统，需要数据库服务器和目录服务器（可选）支持，所以在安装系统时，可以有不同的机器承担不同的功能，如选配三台机器，一台机器安装应用系统，一台机器安装数据库系统，一台机器安装目录服务器系统；也可以选配两台机器，一台机器安装应用系统和数据库系统，一台机器安装目录服务器系统，机器之间通过网络互连。

对于 RA 登记中心端系统，需要数据库服务器支持，又由于操作员的原因，在安装系统时，可以有不同的机器承担不同的功能，如选配两台机器，一台机器安装应用系统，一台机器安装数据库系统；也可以选配一台机器，一台机器安装应用系统和数据库系统；也可以选配三台机器，一台机器安装数据库系统，两台机器分别安装应用系统，分别用作录入机和核心功能机，机器之间通过网络互连。

系统模块组成及相互关系如下所示：



(i) CA 端系统软件环境

CA 中心端应用系统软件包。

MS SQL SERVER 6.5 或 7.0 版本的数据库服务器软件包。

NETSCAPE DIRECTORY SERVER 3.0 目录服务器系统。

(ii) CA 端数据库配置

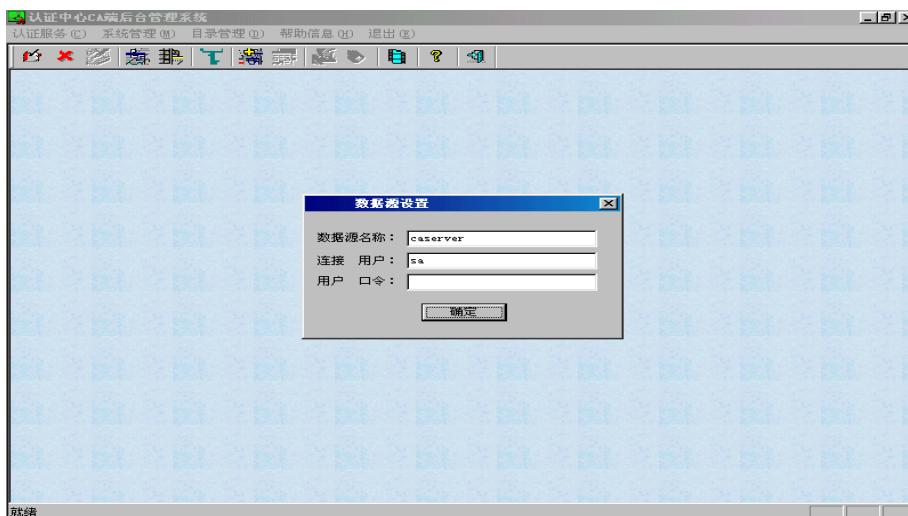
配置数据库服务器的用户 (DBUSER) 为 sa, 口令 (DBPASSWD) 为 12345678 (任意), 创建一个数据库, 名称为: caserver, 根据提供的脚本文件 CASQL.TXT, 创建一系列表。创建 ODBC 数据源, 连接用户为: sa, 连接口令为: 12345678。

(iii) CA 端目录服务器配置

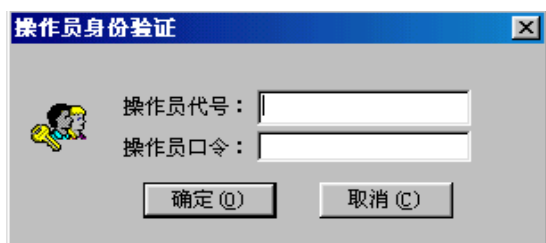
目录服务器的服务端口默认为 389, 用户为: CN=Directory Manager, 口令为: 12345678 (任意), 根节点为: O=SDU_ca_system (CA 系统的组织名称)。

(iv) CA 系统配置

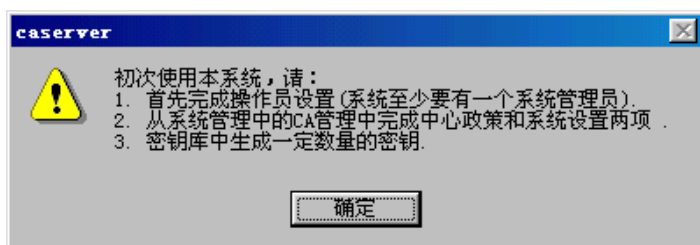
(1) 执行“CA 系统后台管理系统”, 出现初始画面:



(2) 输入数据库连接数据源的名称和口令。接着出现用户登录窗口：



因为首次登录，直接按“确定”。出现一个提示框：



(3) 根据提示，进入“系统管理”——“操作员管理”对话框，添加一个系统管理员。

系统操作员管理

系统现有操作员列表:

操作员编号	操作员姓名	操作员权限

添加操作员输入信息框

操作员编号: 操作员口令:

操作员姓名: 重复 口令:

操作员权限: ☒ 管理员 ☐ 操作员

添加系统管理员成功，退出系统，并以该系统管理员身份登录系统。
出现提示信息：

caserver

 请以系统管理员身份登陆系统，执行<证书政策设定>功能初始化配置信息!!

确定之后，出现如下对话框：

CA 中心端系统配置----CA 政策设定

一次性重要信息设定

生成签名密钥:

更新加密密钥:

签名 算法:

CRL发布时间间隔: 天 小时

用户证书有效期

EMAIL 证书:

SSL-CLIENT证书:

SSL-SERVER证书:

个人 签名 证书:

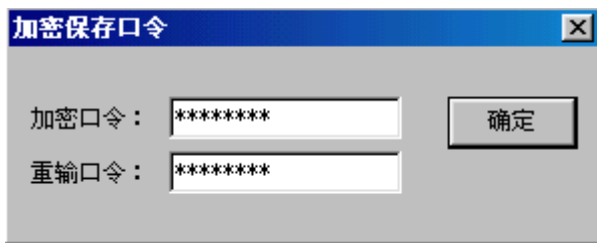
个人 加密 证书:

单位 签名 证书:

单位 加密 证书:

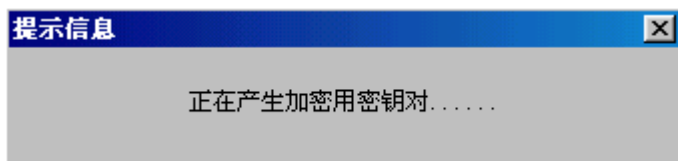
它实际上是“系统管理”——“CA 中心管理”——“证书政策设定”对话框：

点击“生成签名密钥”按钮，出现提示框，输入用于加密保存该密钥的口令字：

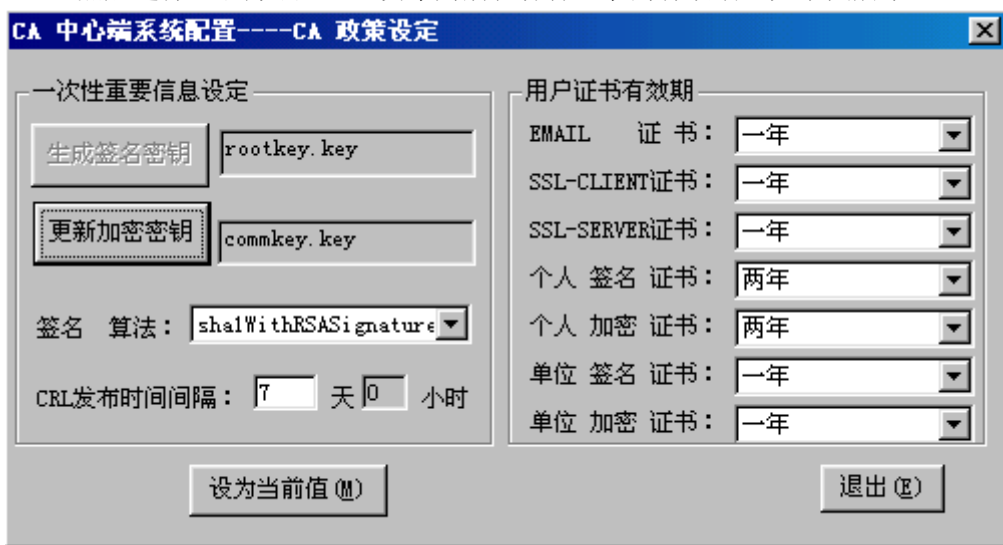


然后系统开始生成该签名密钥，并用该口令生成的对称密钥对称加密保存，密钥文件一般保存在系统当前目录。该密钥用于签发数字证书。当该密钥生成成功后，该按钮变灰。不允许再次更新。

同样的方法，点击“生成加密密钥”，该密钥用于 CA-RA 之间加密通信。出现提示：



(5) 然后选择签名方法、CRL 发布间隔和各种证书的有效期，如下图所示：



然后点击“设为当前值”按钮，保存当前配置，退出。

(6) 选择“系统管理”——“CA 中心管理”——“系统配置设定”，出现对话框，填写左侧相应的项目，如下图所示：

认证中心CA系统参数配置

证书中心参数配置

国家代码:	CN
组织名称:	stw ca center
部门名称:	stw ca center
通用名称:	stw ca center
证书签名算法:	sha1WithRSA signature
根证书有效期:	2001年 8月 1日至 2011年 8月 1日
所在省份名称:	beijing
所在城市名称:	beijing
通信地址:	beijing stw ca center
电子邮箱地址:	cacenter@stw.com
邮政编码:	100037
电话号码:	32233457

CA根证书信息:

CA-RA 通信加密证书信息:

目录服务器信息设定

目录服务器名称

或 主机名: 192.168.1.99

连接端口: 389

用户名称: CN=Directory Manager

用户口令: *****

基本 DN: O=stw_ca_system

设为当前值 (Q)

重要操作

生成自签名根证书 (R)

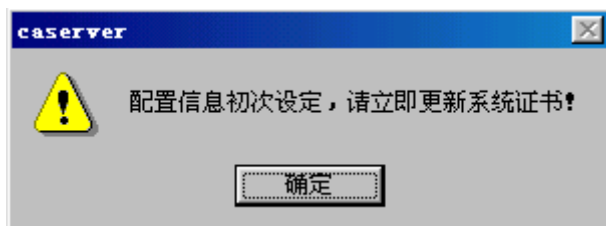
更新CA系统通信证书 (C)

生成CA证书申请 (G)

输出证书 (P) 证书目录发布 (I)

退出 (E)

点击“设为当前值”按钮，出现提示框：



(7) 完成生成系统证书的重要操作

点击“生成自签名根证书”按钮，提示口令并输入，生成根证书。

点击“生成CA系统通信证书”按钮，提示口令并输入，生成通信证书。

系统显示生成的编码后的证书，如下图：



(8) 输出系统证书(签名根证书和加密通信证书)。

点击“输出证书”按钮，出现提示对话框，以文件方式保存系统证书，如命名根证书为：root.der(root.pem), 通信证书为：comm.der。

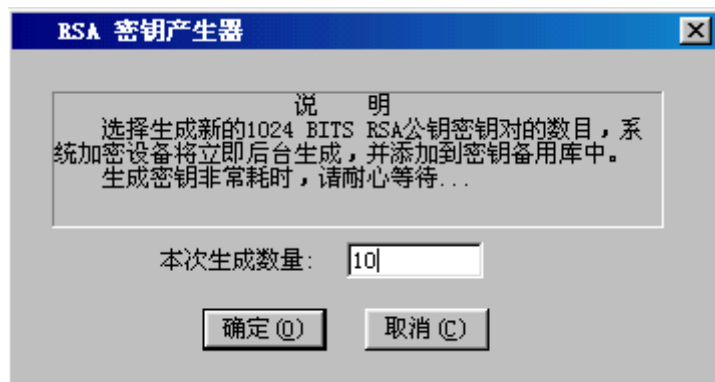
(9) 系统证书目录发布。

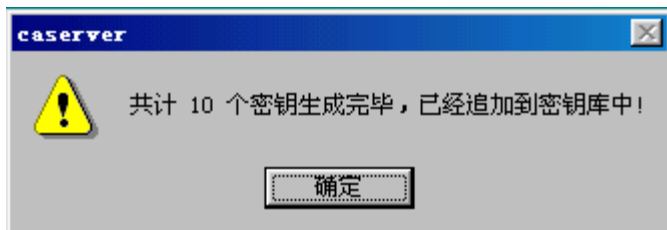
点击“证书目录发布”按钮，系统将根据目录服务器连接配置，连接目录服务器，将系统证书发布到目录服务器。

(10) 批量生成备用密钥。

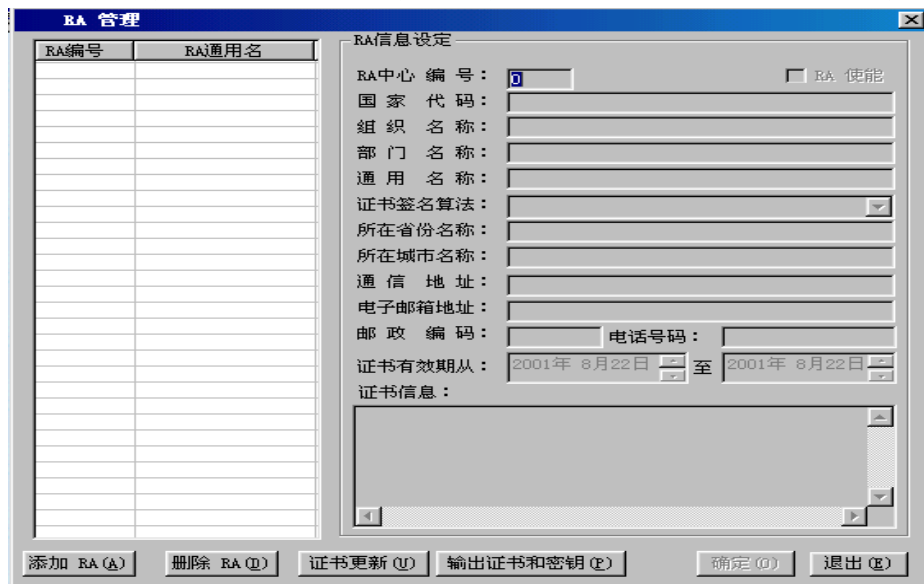
选择菜单“认证服务”—》“批量密钥生成”，出现对话框，输入一次生成的密钥数量，系统开始生成密钥。由于密钥生成比较花费时间，所以一般在系统在提供认证服务的空闲时间，要经常利用该功能产生一定数量的密钥，以备委托方式签发证书和密钥之用。

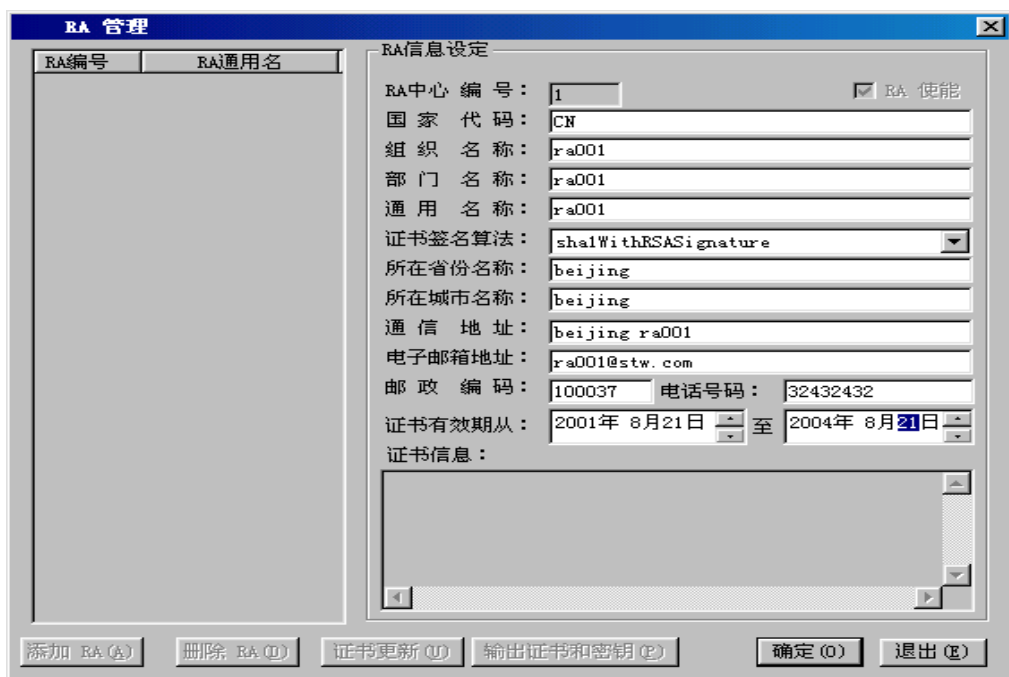
系统提示框如下：



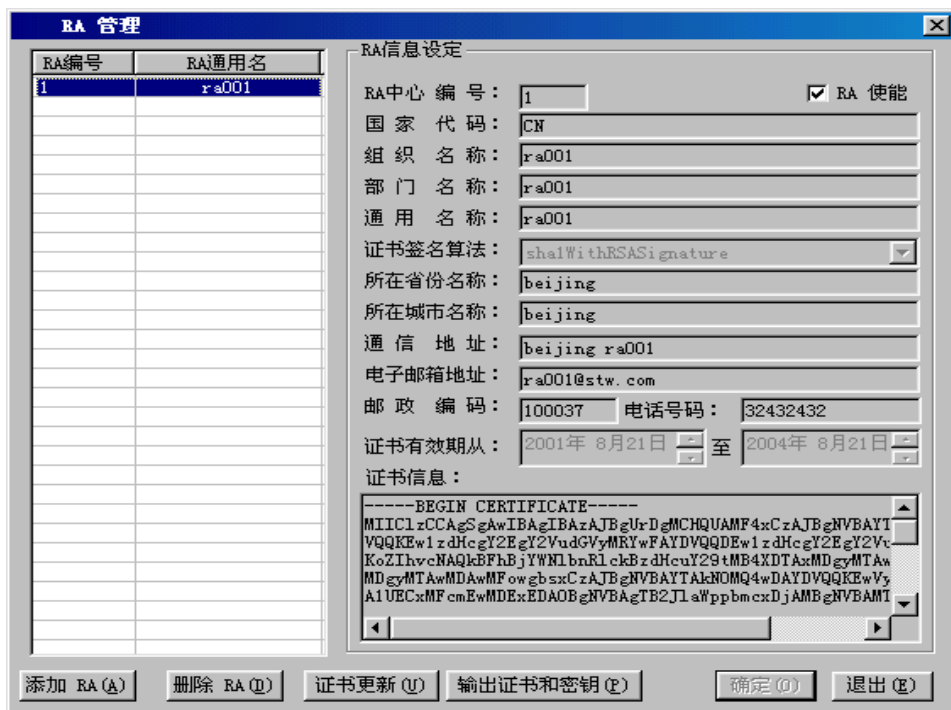


进入“系统管理”—》“RA 中心管理”菜单，显示 RA 中心系统配置对话框：





单击“确定”按钮，系统增加一个 RA 系统。如下图所示：

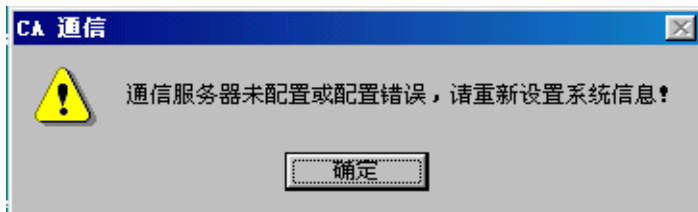



然后单击“输出证书和密钥”按钮，输出该 RA 系统的密钥和证书。其中密钥用用

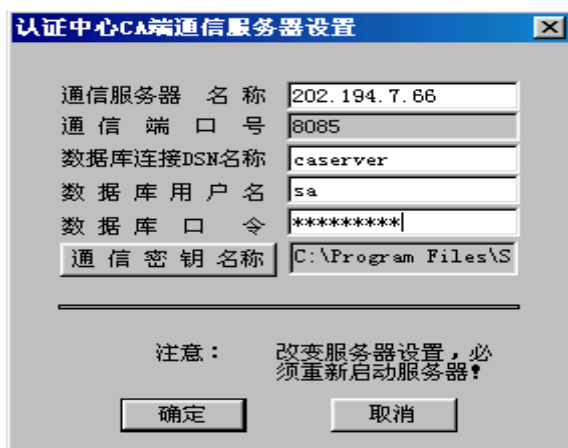
户提供的口令加密，保存在文件中（如：rakey.der），证书保存在文件中（如：racert.der）。在安装该 RA 系统时需要安装这两个文件和 CA 系统的证书。

（v）通信服务器的配置

（1）初次启动。点击“CA-RA 通信服务器”图标，出现提示框：



这时，在桌面右下角出现图标：， “确定”后显示通信服务器配置对话框：



（2）修改通信服务器名称、端口、数据库连接参数以及通信密钥的位置等，然后退出重新启动。出现提示输入通信密钥加密保存口令：



输入正确的口令字，通信服务器配置即可结束，在正常使用时，CA 端一般运行通信服务器就可以接收各个 RA 系统的服务请求，而认证服务需要由后台系统提供，二者通过数据库交换数据。

(VI) RA 系统所需的软件准备

RA 中心端应用系统软件包。

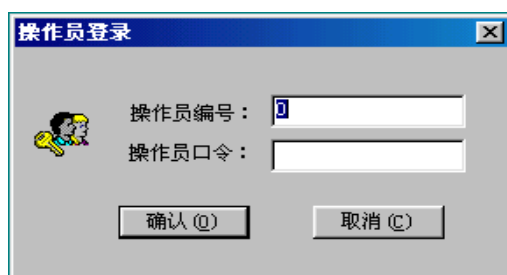
MS SQL SERVER 6.5 或 7.0 版本的数据库服务器软件包。

(VII) RA 系统数据库配置

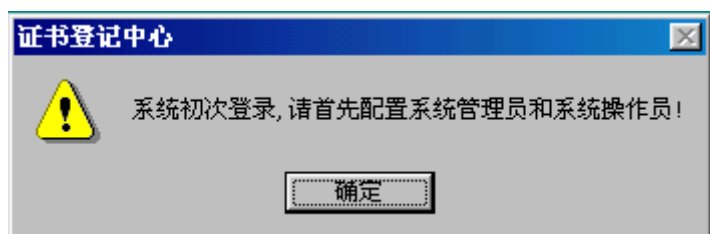
配置数据库服务器的用户(DBUSER)为 sa, 口令(DBPASSWD)为 12345678 (任意), 创建一个数据库, 名称为: caserver, 根据提供的脚本文件 RASQL.TXT, 创建一系列表。创建 ODBC 数据源, 连接用户为: sa, 连接口令为: 12345678。

(VIII) RA 登记中心系统软件配置

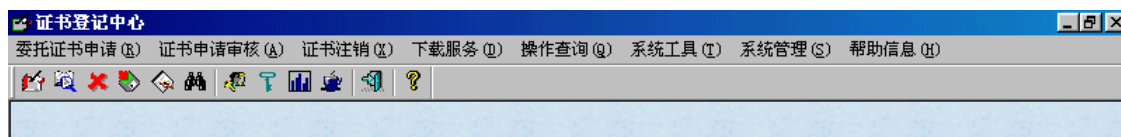
(1) 启动 RA 登记中心系统, 出现:



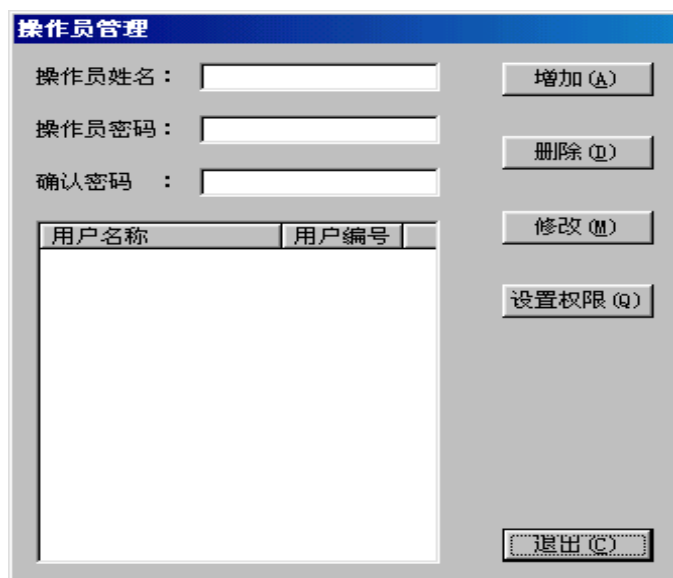
直接执行“确定”，显示：



然后显示主菜单：

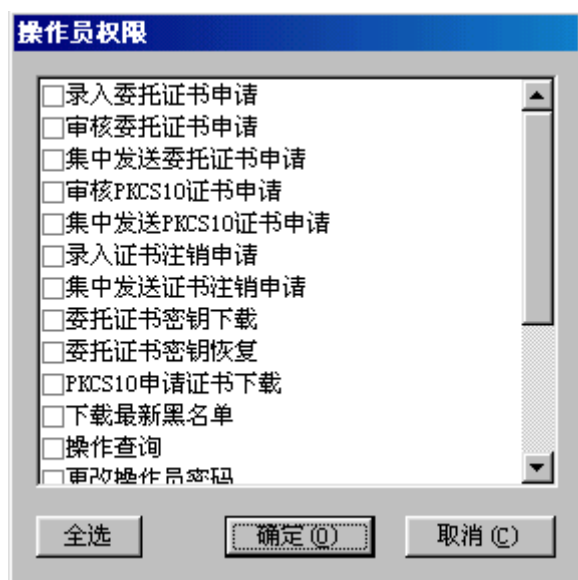


- (1) 设置系统管理员和系统操作员。 进入“系统管理”—》“操作员管理”菜单。
出现系统操作员管理对话框：



The dialog box titled "操作员管理" (Operator Management) contains three input fields on the left: "操作员姓名:" (Operator Name), "操作员密码:" (Operator Password), and "确认密码:" (Confirm Password). To the right of these fields are four buttons: "增加(A)" (Add), "删除(D)" (Delete), "修改(M)" (Modify), and "设置权限(Q)" (Set Permissions). Below the input fields is a table with two columns: "用户名称" (User Name) and "用户编号" (User ID). The table is currently empty. At the bottom right of the dialog is a "退出(C)" (Exit) button.

输入操作员姓名（如：aaa）、密码(如：1000)，然后点击“增加”按钮，在列表框中出现新操作员，该操作员分配的用户编号为：1000，然后点击“设置权限”按钮，出现对话框：



The dialog box titled "操作员权限" (Operator Permissions) displays a list of system functions with checkboxes. The functions are: 录入委托证书申请, 审核委托证书申请, 集中发送委托证书申请, 审核PKCS10证书申请, 集中发送PKCS10证书申请, 录入证书注销申请, 集中发送证书注销申请, 委托证书密钥下载, 委托证书密钥恢复, PKCS10申请证书下载, 下载最新黑名单, 操作查询, and 更改操作员密码. At the bottom of the dialog are three buttons: "全选" (Select All), "确定(O)" (OK), and "取消(C)" (Cancel).

图 30

对话框中列出系统的全部功能, 选择该操作员的操作功能权限, 以后该操作员登录系统只能执行权限内的操作。系统首先应增加一个具有全部功能的超级系统管理员。以后可以通过该超级系统管理员完成对各种权限的操作员的管理。

一般来说，系统管理员具有操作全部功能的权限。

信息录入员具有以下功能：

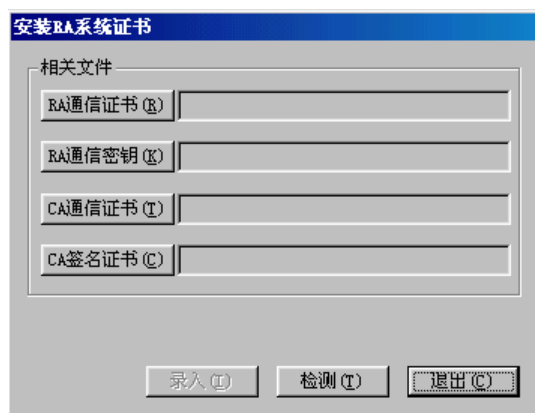
录入委托证书申请
操作查询
更改操作员密码
更换操作员
锁定屏幕

核心功能操作员具有以下功能：

审核委托证书申请
集中发送委托证书申请
审核 PKCS10 证书申请
集中发送 PKCS10 证书申请
录入证书注销申请
集中发送证书注销申请
委托证书密钥下载
委托证书密钥恢复
PKCS10 申请证书下载
下载最新黑名单
操作查询
更改操作员密码
更换操作员
锁定屏幕
统计报表管理

(4) 安装 RA 端的证书和密钥

进入“系统管理”—》“系统证书密钥管理”，出现对话框：



通过点击相应的按钮，可以选择将要安装的相应证书或密钥文件的位置，如：

安装RA系统证书

相关文件

RA通信证书 (R) D:\My Documents\racert.der

RA通信密钥 (K) D:\My Documents\rakey.der

CA通信证书 (T) D:\My Documents\comm.der

CA签名证书 (C) D:\My Documents\root.der

录入 (I) 检测 (T) 退出 (C)

然后点击“检测”按钮，检查输入信息之间的对应关系是否正确，如果正确，允许装入系统。单击“录入”按钮，出现对话框，显示 RA 系统证书信息：

用户证书

用户信息

用户名称：ra001

国 家：cn

省份名称：beijing

城市名称：beijing

单位名称：ra001

部门名称：ra001

通信地址：beijing stw ra001

邮政编码：250100

电子邮箱：ra001@stw.com

联系电话：54365356

证书信息

证书序列号：2 RSA密钥模长：1024

签名算法：sha1WithRSAEncryption

证书类型：[v]

有效期从：2001.08.01 至 2003.08.01

其他信息

录入时间：[] 操作员：0

审核时间：[] 操作员：0

制卡时间：[] 操作员：0

恢复时间：[] 操作员：0

注销时间：[] 操作员：0

注销原因：[v]

录入 (I) 退出 (C)

单击“录入”按钮，出现对话框：

请输入本地RA编号

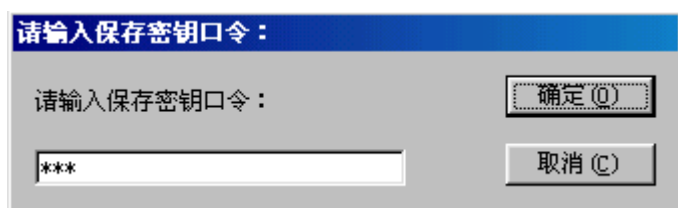
请输入本地RA的编号：

确定 (O)

取消 (C)

输入该 RA 系统的编号（该编号是在 CA 系统管理中赋予的，如：001）。又出现对话

框，提示输入口令加密保存本地的密钥文件。



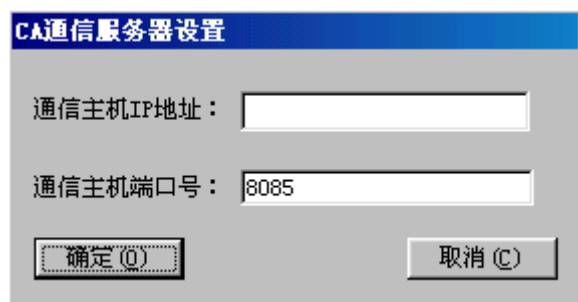
接着会显示成功信息，至此，RA 端的证书和密钥安装成功。



(5) 设置通信服务器参数

为了使 RA 系统能够与 CA 通信服务器网络连接，交换数据，需要设置 CA 通信服务器参数。

进入“系统管理”——“系统通信设置”菜单，显示对话框：



输入 CA 通信服务器的主机名或 IP 地址（如：202.194.7.26），端口采用默认值。

至此，RA 系统的初始化配置全部完成，接下来，就可以进行正常的业务流程了。

(IX) 系统业务操作说明

RA 系统业务操作包括：

- 委托方式的证书密钥申请
- 委托方式的证书密钥申请审核
- 委托方式的证书密钥申请上传

委托方式的证书密钥下载
委托方式的证书密钥恢复
PKCS10 方式的证书申请
PKCS10 方式的证书申请上传
PKCS10 方式的证书下载
证书注销申请
证书注销申请上传
下载黑名单
业务操作查询
统计报表
系统日志

CA 中心后台管理系统业务操作包括：

委托方式的证书密钥签发
PKCS10 方式的证书签发
用户证书注销
证书目录发布
黑名单签发
黑名单目录发布
黑名单输出
委托方式的批量证书密钥签发
PKCS10 方式的批量证书签发
批量密钥生成
测试证书生成
证书信息查询
委托拒签信息查询
PKCS10 拒签信息查询
密钥库信息查询
系统日志管理
目录服务器信息查询

具体操作请参见系统使用说明书，在此不赘述。

实验八 安全电子邮件的配置和使用

8.1 实验目的

利用前面实验中的 CA 系统,颁发安全电子邮件证书,并在一个支持电子邮件证书的邮件客户端系统中安装配置,实现加密和签名电子邮件系统。

8.2 实验内容

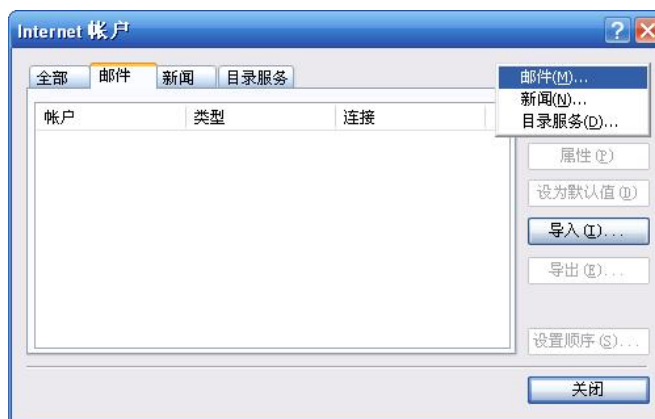
1. 在浏览器中加载电子邮件证书。
2. 配置 OUTLOOK 邮件客户端系统,实现签名电子邮件和加密签名电子邮件的发送与接收。

8.3 实验指导

注:如果是使用 EJBCA 系统签发的数字证书,则数字证书自动加载浏览器,如果是使用 SureCA 系统签发的证书,应使用委托方式签发电子邮件证书,下载证书时要下载 PKCS12 编码的证书和密钥信息包。PKCS12 信息包在 WINDOW 环境下可以通过双击加载浏览器。

1. 在 Outlook Express 中设定邮件

首先,打开 Outlook Express,然后选择菜单中的“工具”菜单中的“帐户”选项,出现“Internet 帐户”对话框,我们点击右边的“添加”按钮,选择“邮件”选项,如图所示。



然后根据向导提示, 输入用户 psdu (pengyanhai@mail.sdu.edu.cn) 和 pnet (pyhy68246482@163.com) 的邮件显示姓名、电子邮件地址、接收邮件服务器和发送邮件服务器的域名或 IP 地址 (在实验的时候, 自行决定邮箱地址)。最终, 系统显示有如下帐户信息:

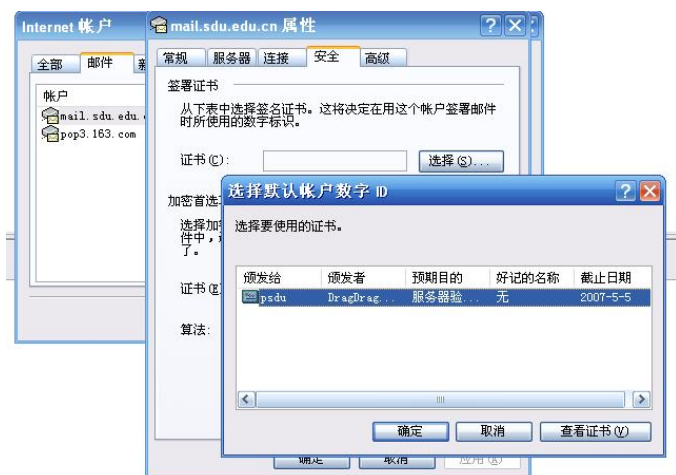


2. Outlook Express 中设置邮箱与数字证书的绑定

在 Outlook Express6.0 单击菜单中的“工具”, 选择“账号”, 选取“邮件”选项卡中的用于发送安全电子邮件的邮件账号, 然后单击“属性”。

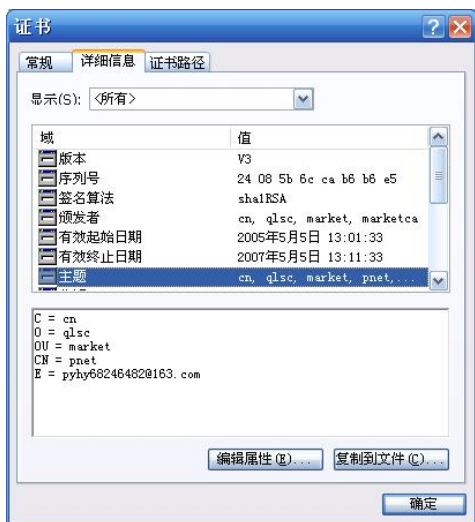


选择上面的“安全”标签，可以看到“签署证书”和“加密首选项”两栏。通过相关设置，我们可以进行邮件的签署和加密。



在“签名证书”项后，点击“选择”按钮，可以看到我们在 EJBCA 系统中申请的证书。选择读者的数字证书，点击“确定”完成邮箱与证书的绑定，读者也可以点击“查看证书”，了解自己证书的详细信息。

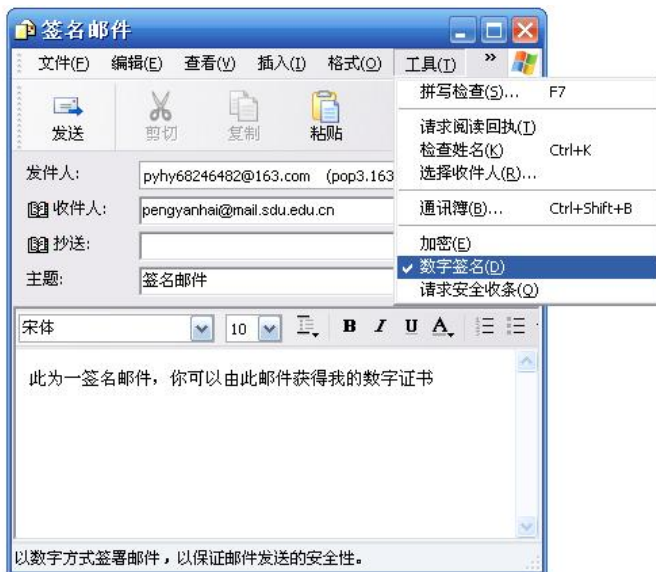
注意：如果点击“选择”按钮，没有相关的证书弹出来，请确认读者的证书已经正确安装且没有过期。同时要确认读者在 Outlook Express 中所设置的邮箱与读者在申请数字证书时所提供的邮箱一致。查看读者在申请数字证书时所提供的邮箱方法：在 Internet Explorer 中，依次点击“工具”中的“Internet 选项”，选择“内容”选项卡中的“证书”，选中读者的数字证书，点击“查看”，找到“详细信息”中的“主题”，读者就可以看到邮箱。



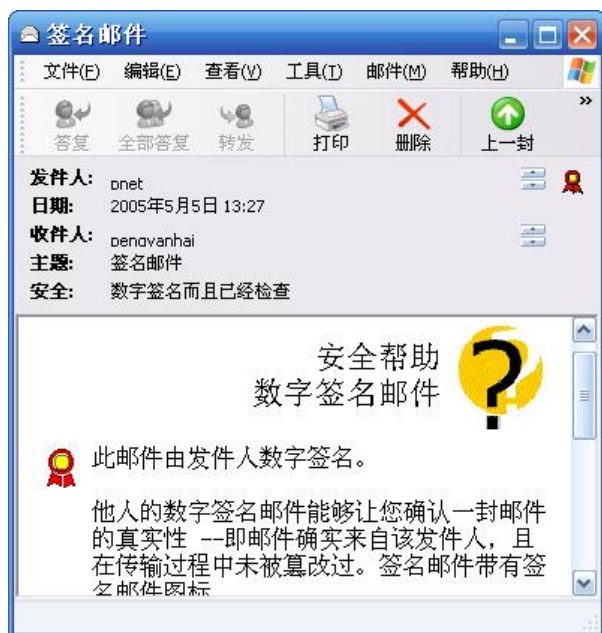
按照同样的方法，读者也可以在“加密首选项”中把读者自己的证书选中。

1. 发送签名的电子邮件

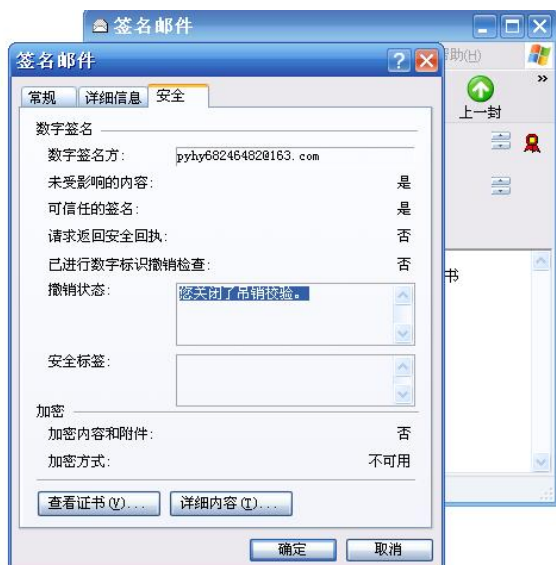
发送加密邮件前必须先获得接收方的数字证书，读者可以首先让接收方给读者发一份签名邮件来获取对方的数字证书。如 psdu 要给 pnet 发一封加密电子邮件，则 psdu 需要查询并下载 pnet 用户的数字证书，或者先由 pnet 给 psdu 发一封签名邮件，如下：



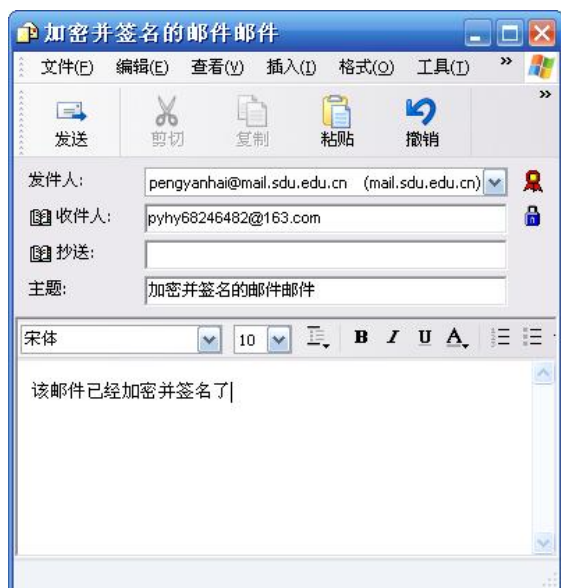
点击“发送”，签名邮件发送成功。当收件人收到并打开有数字签名的邮件时，将看到 数字签名邮件 的提示信息，按“继续”按钮后，才可阅读到该邮件的内容。



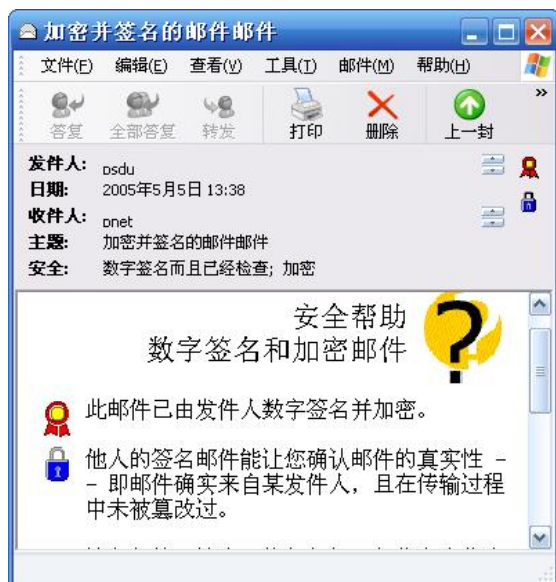
若邮件在传输过程中被他人篡改或发信人的数字证书有问题，将出现“安全警告”提示。用户 psdu 收到邮件后，我们可以看到，邮件的右边中间有一个小图标，点击它，可以看到相关的数字证书信息，包括把查看他的相关信息、把发信人的数字证书添加到自己的通讯簿。



现在，psdu 已经获得了 pnet 的数字证书，则可以给她发一封加密邮件了。



当用户 pnet 收到 psdu 加密并签名的邮件后，会有如下提示：



点击提示页面下方的“继续”按钮，即可阅读加密的邮件了。

实验九 安全 WEB 访问的配置和使用

9.1 实验目的

利用前面实验中的 CA 系统,颁发浏览器 SSL 客户端证书以及 WEB SERVER 证书,并在一个支持 SSL 安全连接的系统中安装配置,实现基于数字证书安全访问的 WEB 系统。

9.2 实验内容

1. 申请并获取安装浏览器 SSL-CLIENT 证书。
2. 申请并获取安装 IIS WEB 服务器证书。
3. 配置 WEB 服务器,实现 SSL 单向认证 WEB 登录和 SSL 双向认证 WEB 登录。

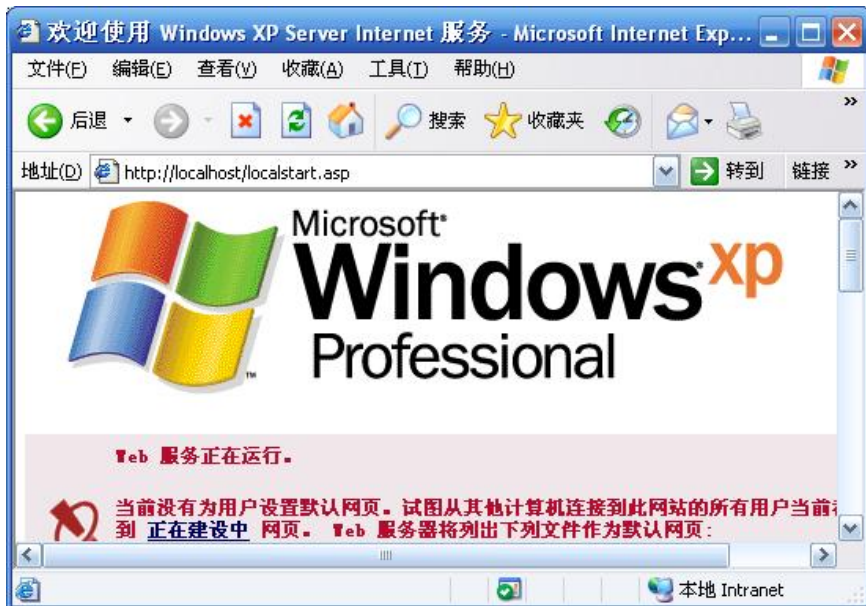
9.3 实验指导

默认情况下我们所使用的 HTTP 协议是没有任何加密措施的,所有的消息全部都是明文形式在网络上传送的,恶意的攻击者可以通过安装监听程序来获得我们和服务器之间的通讯内容。通过 SSL (Security Socket Layer) 安全机制使用数字证书,建立 SSL 安全通道后,只有 SSL 允许的客户才能与 SSL 允许的 Web 站点进行通信,并且在使用 URL 资源定位器时,输入 https://,而不是 http://。SSL (加密套接字协议层)位于 HTTP 层和 TCP 层之间,建立用户与服务器之间的加密通信,确保所传递信息的安全性。SSL 是工作在公共密钥和私人密钥基础上的,任何用户都可以获得公共密钥来加密数据,但解密数据必须要通过相应的私人密钥。使用 SSL 安全机制时,首先客户端与服务器建立连接,服务器把它的数字证书与公共密钥一并发送给客户端,客户端随机生成会话密钥,用从服务器得到的公共密钥对会话密钥进行加密,并把会话密钥在网络上传递给服务器,而会话密钥只有在服务器端用私人密钥才能解密,这样,客户端和服务端就建立了一个惟一的安全通道。

下面,通过微软 IIS 服务器来演示数字证书的使用过程。

1. 申请服务器证书

未使用数字证书进行验证时，在 IE 中输入：<http://localhost> 显示如下：

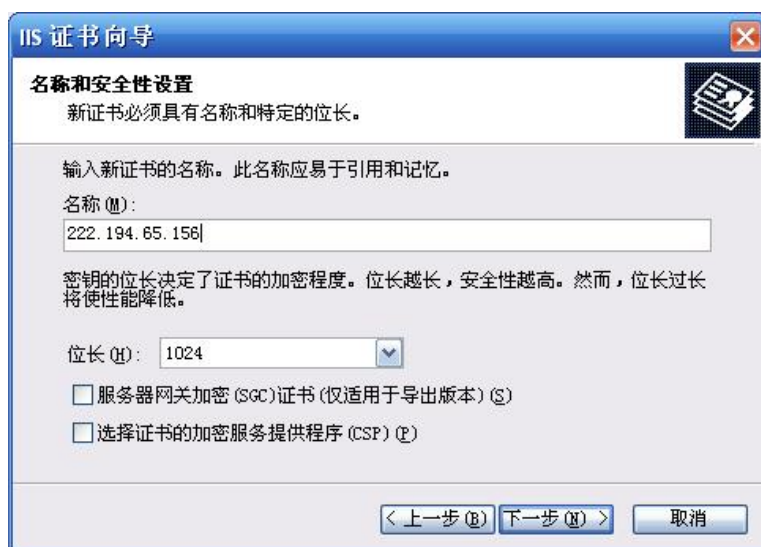


打开 Internet 信息服务管理器，然后打开要为之申请证书的站点的属性。找到“目录安全性”对话框。



选择“服务器证书”按钮，根据服务器证书的状态，选择“下一步”。在出现的选项中，选择“创建一个新证书”。现在可以准备证书请求了。点击“下一步”，输

入站点名，并选择私钥的长度



点击“下一步”，输入单位信息，站点公用名称，和地理信息。接着选择存储证书请求的文件，确认所填信息无误后，会由如下提示：



2. 获得服务器证书

获得服务器证书的方法和前面手工获得个人数字证书的方法有点类似。首先，向 EJBCA 系统 RA 管理员提出申请并由 RA 新建用户

End Entity Profile	EMPTY	Required
Username	website	<input checked="" type="checkbox"/>
Password	*****	<input checked="" type="checkbox"/>
Confirm Password	*****	
Batch	<input type="checkbox"/>	
Email	<input type="text"/> @ <input type="text"/>	<input type="checkbox"/>
Subject DN Fields		
E, EmailAddress in DN	Use data from Email field : <input type="checkbox"/>	<input type="checkbox"/>
UID, Unique Id	<input type="text"/>	<input type="checkbox"/>
CN, Common Name	222.194.65.156	<input checked="" type="checkbox"/>
SN, Serial Number	<input type="text"/>	<input type="checkbox"/>
GivenName, Given Name	<input type="text"/>	<input type="checkbox"/>

然后下载证书，输入用户名和密码，并将前面生产的证书申请文件 certreq.txt 内容以纯文本方式粘贴至目标文本框中，

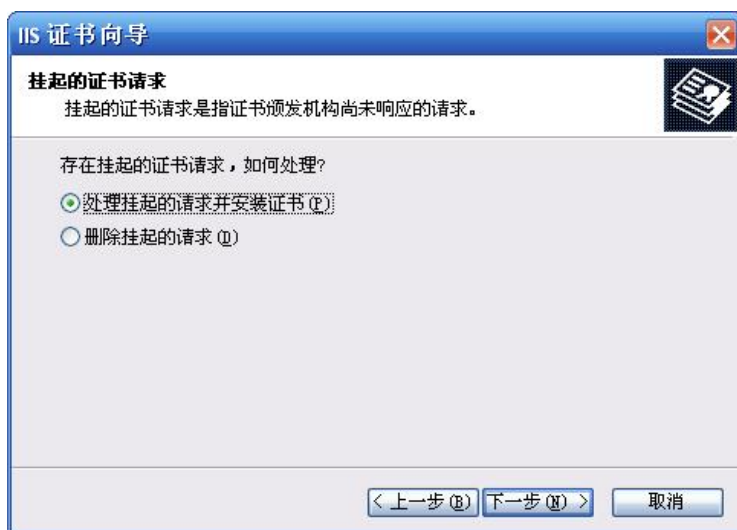
Username:	website
Password:	litng
<pre> -----BEGIN NEW CERTIFICATE REQUEST----- MIIDRTCCAq4CAQAwajEXMBUGA1UEAxMOMjIyLjE5NC42NS4xNTYxETAPBgNVBAsE CI9vTVZbZpZiMREwDwYDVQQKHGhccU4cWSdbZjENMA5GA1UEBx4EbU5TVzENMA5G A1UECB4EXHF0HDELMAkGA1UEBhMCQ04wZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ AoGBAAeLrDtp+XRW+1CshUgZyEqJT4kUfu4FojpPnJaaAX1eWzOUYj0C1Wd9CU8 z+1tYg+CilhzeoFLH8g5yQYzQqM1zQdhczWv+UKOCk9EQZxPV5P9FAYuEBKQ7xub Oc9SGJ0V3eiCMX9Uke84f5WtBSZcUtoBbOiftoxqUWGk+CGxAgMBAAGggGZMB0G CisGAQQBgjcNAgMxDBYKNS4xLjI2MDAuMjB7BgorBgEEAYI3AgEOMW0wazAOBgNV HQ8BAf8EBAMCBFAwRAYJKoZIhvcNAQkPBDCwNTAOBgghkiG9w0DAgICAIAwDgYI KoZIhvcNAQcAgCAMAcGBSS0AwIHMAoGCCqGSIb3DQMHBMBGA1UdJQQMMAoGCCsG AQUFBwMBMIH9BgorBgEEAYI3DQICMYHuMIHrAgEBH1oATQBpAGMAcBvAHMAbwBm AHQAIABSAFMAQQAgAFMAQwBoAGEAbgBuAGUAbAAgAEMAcgB5AHAAdABvAGcAcgBh AHAAaABpAGMAIABQAHIAbwB2AGkAZAB1AHIDgYkAk0kfH5skr4jsEVya3mgUoyaYM O456ECNzr4Cb+WhPgexfj005qwOG1oDOTaKycrk5pG+IPBQnq+4cotT8hWJQwpc ----- </pre>	
PEM Certificate <input type="button" value="OK"/>	

点击 OK 按钮，弹出对话框，将生成的 cert.pem 文件保存到磁盘。



3. 安装服务器证书

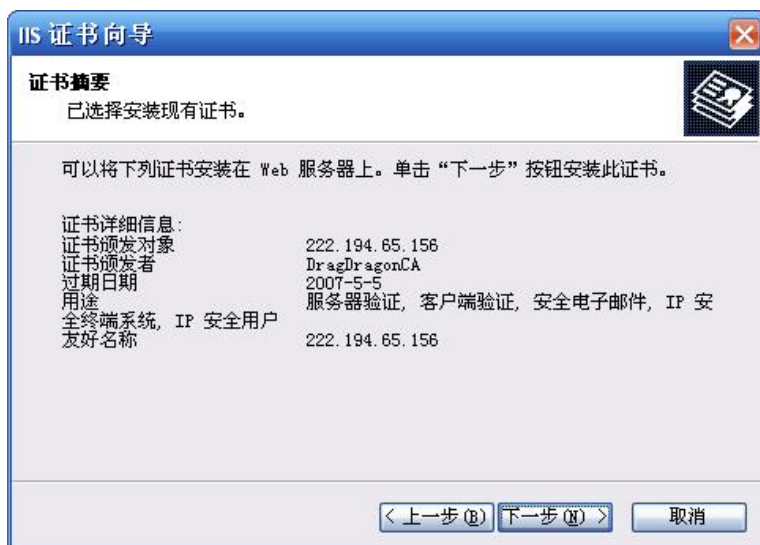
安装证书的时候，同生成证书申请文件相同，先打开 Internet 信息服务管理器，然后打开要为之申请证书的站点的属性。还是选择“目录安全性”，选择“服务器证书”，现在我们发现，服务器已经挂起了一个证书请求。根据证书向导，处理挂起的请求并安装证书。



点击下一步，输入包含证书颁发机构相应的文件路径和名称，

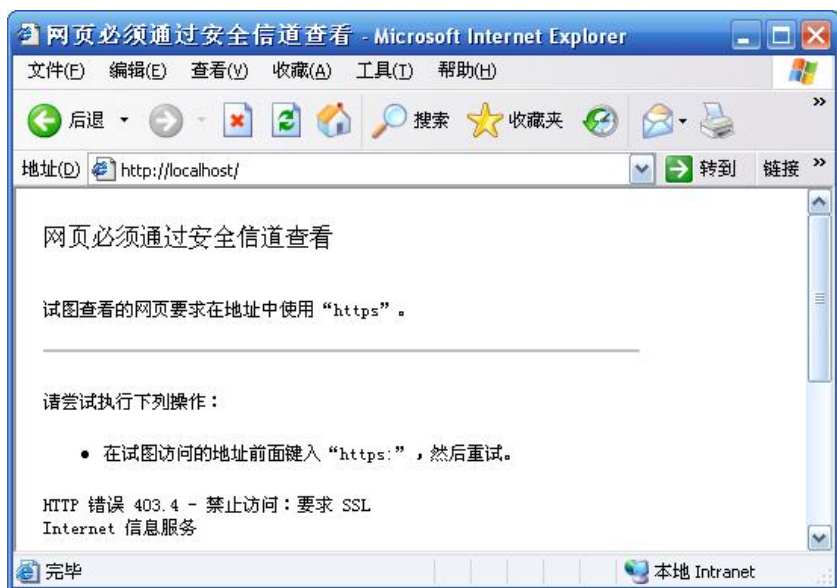


再下一步，会显示证书内容摘要，继续，便完成了证书的安装工作。



再次打开网站属性页面的“目录安全性”选项卡，“安全通信”的“编辑”按钮变为可响应状态, 点击“编辑...”，选定“要求安全通道（SSL）”





之后，如果在此通过 URL <http://localhost> 访问 IIS 服务，会提示“网页必须通过安全信道查看” URL 改为：<https://localhost> 之后，将会显示最初的界面，如下：



证明浏览器访问 WEB 服务器已经是通过 SSL 安全通道了。

实验十 OPENSSL 软件包的使用

10.1 实验目的

了解一个开放源码的密码库实现，并能够通过对核心函数库的掌握，进行二次开发，编制较简单的安全应用。

10.2 实验内容

1. 对 OPENSSL V0.9.7a 开发包,按照说明(README),编译生成在 WINDOWS 下的动态库。
2. 通过研究示例程序,基本掌握核心函数的实现方法。
3. 自己动手编写程序(调用开发包),实现密钥生成、对称加解密、公钥加解密、HASH 生成与验证、数字证书生成等基本功能,使用数字证书机制设计一个安全通信的示例程序,保证数据传输的机密性、完整性、数字签名等功能。

10.3 实验指导

参考中文网站: <http://openssl.cn/>

Eric A. Young 和 Tim J. Hudson 自 1995 年开始编写后来具有巨大影响的 OpenSSL 软件包,它是一个完全开放源代码的软件包。OpenSSL 的算法已经非常完善,对 SSL2.0、SSL3.0 以及 TLS1.0 都支持。OpenSSL 目前最新的版本是 0.9.7b 版。

OpenSSL 采用 C 语言作为开发语言,这使得 OpenSSL 具有优秀的跨平台性能。OpenSSL 支持 Linux、Windows、BSD、Mac、VMS 等平台,这使得 OpenSSL 具有广泛的实用性。

OpenSSL 整个软件包大概可以分成三个主要的功能部分:密码算法库、SSL 协议库以及应用程序。OpenSSL 的目录结构自然也是围绕这三个功能部分进行规划的。

首先,我们注意到 OpenSSL 的根目录下有不少文件,这些文件包含 OpenSSL 各个平台下编译安装的说明文档、编译安装的配置文件以及 OpenSSL 本身版本变化的一些说明文档。诸如 INSTALL.*这样名称的文件,都是安装编译说明文件,后缀名是平台的名称。比如 INSTALL.w32,就是 Windows 平台的 OpenSSL 安装编译说明文件。只有

Linux 的安装编译说明文件是不带后缀的，就是 INSTALL，由此可见 OpenSSL 肯定是出身 Linux 家族了。表 13-1 列出了系统平台 and 对应安装文件的关系。其它一些文件的作用根据其文件名就可以知道个大概，如果你刚刚接触 OpenSSL，这些文件还是值得一读的。

Crypto 目录是 OpenSSL 所有密码算法和一些 PKI 相关标准源码存放的目录，也是 OpenSSL 最重要的一个目录。SSL 目录是 SSL 协议各个版本的实现源码存放的目录。Doc 目录是 OpenSSL 使用的说明文档存放的目录，这个目录对于 OpenSSL 使用者来说具有“芝麻开门”的作用。Apps 目录存放了 OpenSSL 所用应用程序的源代码文件，也是研究 OpenSSL 的 API 很好的例子。Demos 目录就是一些乐意奉献的人写的 OpenSSL 应用的例子了，在你开始使用 OpenSSL 进行工作之前，可以看看这个目录，或许会有所帮助。Include 目录是使用 OpenSSL 的库进行编程的时候可能需要使用到的一些头文件。Test 目录测试 OpenSSL 一些自身测试程序源文件所在的地方。

如果你在 Windows 平台下将 OpenSSL 编译成功后，还会增加三个新的目录：inc32、out32dll、tmp32dll。Inc32 目录根 Include 目录相似，存放的是 Windows 平台下使用 OpenSSL 进行编程需要包含的头文件。Out32dll 则存放了 OpenSSL 编译成功后的可执行应用程序、链接库 LIB 文件和动态 DLL 文件。Tmp32dll 则是在编译过程中存放 OBJ 等临时文件的目录。

表 13-1 OpenSSL 安装说明文件名和相应的平台

文件名	系统平台简单描述
INSTALL	Linux 等 Unix 平台
INSTALL.W32	Windows 平台，包括 Windows98、Windows2000、WindowsNT 和 WindowsXP 等
INSTALL.WCE	WinCE 平台
INSTALL.MacOS	苹果电脑的操作平台 MacOS
INSTALL.OS2	OS2 操作平台
INSTALL.VMS	VMS 平台，一种在 Windows 系统下虚拟操作系统
INSTALL.DJGPP	DJGPP 平台，一种在 Windows 系统下的虚拟操作系统

表 13-2 OpenSSL 部分目录的功能说明

目录名	功能描述
Crypto	存放 OpenSSL 所有加密算法源码文件和相关标注如 X.509 源码文件，是 OpenSSL 中最重要的目录，包含了 OpenSSL 密码算法库的所有内容。
SSL	存放 OpenSSL 中 SSL 协议各个版本和 TLS 1.0 协议源码文件，包含了 OpenSSL 协议库的所有内容。
Apps	存放 OpenSSL 中所有应用程序源码文件，如 CA、X509 等应用程序的源文件就存放在这里。
Doc	存放了 OpenSSL 中所有的使用说明文档，包含三个部分：应用程序说明文档、加密算法库 API 说明文档以及 SSL 协议 API 说明文档。
Demos	存放了一些基于 OpenSSL 的应用程序例子，这些例子一般都很简单，演示怎么使用 OpenSSL 其中的一个功能。
Include	存放了使用 OpenSSL 的库时需要的头文件。
Test	存放了 OpenSSL 自身功能测试程序的源码文件。

表 13-3 Crypto 子目录列表

目录名称	目录类型	内容或功能描述
Aes	对称算法	美国新的对称加密算法标准 AES 算法源码。
Bf	对称算法	Blowfish 对称加密算法源码。
Cast	对称算法	CAST 对称加密算法源码。
Des	对称算法	包括了 DES 和 3DES 对称加密算法源码。
Idea	对称算法	IDEA 对称加密算法源码。
Rc2	对称算法	RC2 对称加密算法源码。
Rc4	对称算法	RC4 对称加密算法源码。
Rc5	对称算法	RC5 对称加密算法源码。
Dh	非对称算法	DH 非对称密钥交换算法源码。
Dsa	非对称算法	DSA 非对称算法源码，用于数字签名。
Ec	非对称算法	EC 椭圆曲线算法源码。
Rsa	非对称算法	RSA 非对称加密算法源码，既可以用于密钥交换，也可以用于数字签名。
Md2	信息摘要算法	MD2 信息摘要算法源码。
Md5	信息摘要算法	MD5 信息摘要算法源码。
Mdc2	信息摘要算法	MDC2 信息摘要算法源码。
Sha	信息摘要算法	SHA 信息摘要算法源码，包括了 SHA1 算法。
Ripemd	信息摘要算法	RIPEMD-160 信息摘要算法源码。
Comp	数据压缩算法	数据压缩算法的函数接口，目前没有压缩算法，只是定义了一些空的接口函数。

Asn1	PKI 相关标准	ASN.1 标准实现源码, 只实现了 PKI 相关的部分, 不是完全实现。包括 DER 编解码等功能。
Ocsp	PKI 相关标准	OCSP (在线证书服务协议) 实现源码。
Pem	PKI 相关标准	PEM 标准实现源码, 包括了 PEM 的编解码功能。
Pkcs7	PKI 相关标准	PKCS#7 标准实现源码。PKCS#7 是实现加密信息封装的标准, 包括了证书封装的标准和加密数据的封装标准。
Pkcs12	PKI 相关标准	PKCS#12 标准实现源码。包括了 PKCS#12 文件的编解码功能。PKCS#12 是一种常用的证书和密钥封装格式。
X509	PKI 相关标准	X.509 标准的实现源码。包括了 X.509 的编解码功能, 证书管理功能等。
X509v3	PKI 相关标准	X.509 第三版扩展功能的实现源码。
Krb5	其它标准支持	支持 Kerberos 协议的一些接口函数和结构定义。
Hmac	其它标准支持	HMAC 标准的支持结构和函数源代码。
Lhash	其它标准支持	动态 HASH 表结构和函数源代码。
Bio	自定义	OpenSSL 自身定义的一种抽象 IO 接口, 封装了各种平台的几乎所有 IO 接口, 如文件、内存、缓存、标准输入输出以及 Socket 等等。
Bn	自定义	OpenSSL 实现大数管理的结构及其函数。
Buffer	自定义	OpenSSL 自定义的缓冲区结构体。
Conf	自定义	OpenSSL 自定义的管理配置结构和函数。
Dso	自定义	OpenSSL 自定义的加载动态库的管理函数接口。如使用 Engine 机制就用到了这些函数提供的功能。
Engine	自定义	OpenSSL 自定义的 Engine 机制源代码。Engine 机制运行 OpenSSL 使用第三方提供的软件密码算法库或者硬件加密设备进行数据加密等运算。相当于 Windows 平台的 CSP 机制。
Err	自定义	OpenSSL 自定义的错误信息处理机制。
Evp	自定义	OpenSSL 定义的一组高层算法封装函数, 包括了对称加密算法封装、非对称加密算法封装、签名验证算法封装以及信息摘要算法封装, 类似 PKCS#11 提供的接口标准。
Objects	自定义	OpenSSL 管理各种数据对象的定义和函数。事实上, Objects 的 OID 是根据 ASN.1 的标准进行命名的, 不完全是 OpenSSL 自定义的结构。
Rand	自定义	OpenSSL 的安全随机数产生函数和管理函数。
Stack	自定义	定义了 OpenSSL 中 STACK 结构和相关管理函数。
Threads	自定义	OpenSSL 处理线程的一些机制。
Txt_db	自定义	OpenSSL 提供的文本证书库的管理机制。
Ui	自定义	OpenSSL 定义的一下用户接口交换函数。
Perlasm	自定义	编译的时候需要用到的一些 Perl 辅助配置文件。

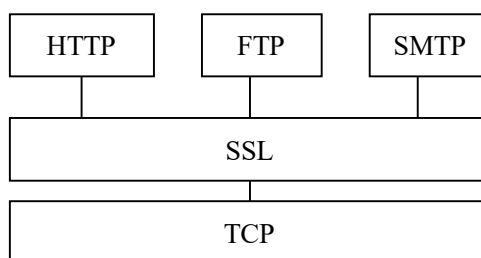
10.3.1 使用 OpenSSL.exe

使用 OpenSSL.exe(Linux 中可执行文件名是 openssl)可以做很多工作, 是一个很好的测试或调试工具。

显示版本和编译参数: >openssl version -a
 查看支持的子命令: >openssl ?
 SSL 密码组合列表: >openssl ciphers
 测试所有算法速度: >openssl speed
 测试 RSA 速度: >openssl speed rsa
 测试 DES 速度: >openssl speed des
 产生 RSA 密钥对: >openssl genrsa -out 1.key 1024
 取出 RSA 公钥: >openssl rsa -in 1.key -pubout -out 1.pubkey
 RC4 加密文件: >openssl enc -e -rc4 -in 1.key -out 1.key.enc
 RC4 解密文件: >openssl enc -d -rc4 -in 1.key.enc -out 1.key.2
 计算文件的 MD5 值: >openssl md5 < 1.key
 计算文件的 SHA1 值: >openssl sha1 < 1.key

10.3.2 SSL 函数编程 API

SSL 协议的主要功能即把 TCP 的字节流变成了一个安全的流, 所有基于 TCP 的程序可以很容易地采用 SSL 协议。主要的变化在于 accept/connect() 变化为使用 OpenSSL 提供的 SSL_accept/SSL_connect(), read/write() 变化为 SSL_read/SSL_write()。



初始化函数库

```
SSL_load_error_strings();
OpenSSL_add_all_algorithms();
```

客户端

```
SSLv3_client_method();
SSL_CTX_new();
SSL_CTX_set_accepted_Cas(); // 认可的 CAs
SSL_CTX_use_certificate_file(); // 自己的证书
SSL_CTX_use_PrivateKey_file(); // 自己的私钥
SSL_CTX_check_private_key(); // 检查证书-私钥一致性
SSL_CTX_set_cipher(); // 自己喜欢的算法组合
```

服务器端

```
SSLv3_server_method();
SSL_CTX_new();
SSL_CTX_set_accepted_Cas();
```

```
SSL_CTX_use_certificate_file();  
SSL_CTX_use_PrivateKey_file();  
SSL_CTX_check_private_key();  
SSL_CTX_set_cipher();
```

10.3.3 SSL 示例程序



demo_mini.zip

10.4 建议和要求

要求使用 Sniffer 观察 SSL 协议，核实数据加密效果。

证书的使用分 3 步走：首先使用例子程序中附带的密钥对和证书，然后尝试使用 openssl.exe 签发证书，然后使用 CA in Windows、ejbca 或 SureCA 等签发证书。这些证书有可能存在兼容性问题。

先使用命令行界面实现功能，然后考虑改进成窗口界面程序。

本实验可以两同学合作。

实验十一 OpenVPN 配置和使用

VPN 的优点是对应用层透明，因此特别方便部署。终端用户自己部署 VPN 典型的有三种方式：IPSec 方式、基于 SSL 的 OpenVPN 方式、PPTP/L2TP 等拨号方式。其中 IPSec 方式是 IPv6 强制要求的，OpenVPN 是开源的，PPTP 在 Windows 中支持。

11.1 实验目的

- (1) 掌握 OpenVPN 的部署和使用方法
- (2) 了解其它 VPN 的部署和使用方法

11.2 实验内容

- (1) OpenVPN 软件的安装
- (2) 基于共享秘密的 OpenVPN
- (3) 基于公钥证书的 OpenVPN （选做）

11.3 实验指导

OpenVPN 通常被用来建立端到端的安全隧道。OpenVPN 使用 SSL 协议保护两端之间的通信，这样就充分利用了 SSL 的易用性，简化了管理成本。

IPSec 是 IPv6 中内置的安全特性，可以用传输模式来保护 IP 分组的数据载荷部分。

11.3.1 OpenVPN 安装

OpenVPN 是开放软件，可以运行在 Windows、Linux 等各种平台上。下载可执行 exe 安装文件，安装 OpenVPN 过程中会在操作系统中添加一个虚拟网卡。该网卡的比特传输（物理层）是用运行在实际网卡上的一条 SSL 连接模拟的。另外，在 Windows 上还可以看到增加了一个名为 OpenVPNService 的后台服务项。

OpenVPN 的配置文件是 ./conf/*.ovpn，每个文件对应着一个端到端连接。

注：SSL 协议结合使用公钥算法、对称算法和散列算法等密码算法，运行在 TCP 等可靠流协议之上，提供认证、加密和完整性等安全服务，可以把 SSL 协议看作是经过安全改造的 TCP 协议。

11.3.2 基于共享秘密文件的 OpenVPN

为了能建立 VPN 连接，需要指明鉴别认证手段，目前常用两种方式：共享秘密或公钥证书。

下面是一个使用共享秘密的 VPN 客户端的配置文件实例：

```
remote 211.86.49.238
port 19201
dev tap
dev-node OpenVPN_CARD_1
ifconfig 192.168.0.2 255.255.255.0
secret key.txt
ping 10
comp-lzo
verb 4
mute 10
```

其中指明了远程 VPN 服务器的地址及端口号，本地虚拟网卡名字、IP 地址及子网掩码，共享秘密文件的名字等。秘密文件中一般是一个口令，或者别的不易被猜到的随机串，但是必须保证两端内容一致。

相应的服务端的配置文件只需要去掉第一行，并把其余诸行按照实际修正即可。

在两台机器上准备好配置文件和秘密文件后重起后台服务，使用 192.168.0.0/24 网段地址就可以 ping 通对方。使用 Sniffer 工具查看实际网卡上的流量，就可以发现只能看到 SSL 协议的密文。

11.4 一个实际 OpenVPN 配置文件样例

OpenVPN Client	 OpenVPN_client.ov pn	 key.txt
OpenVPN Server	 OpenVPN_server.ov pn	 key.txt

11.5 要求和建议

必须使用 Sniffer 查看并核实其确实加密了。

如果 VPN 连接建立出错，相应的错误信息在 ./log/*.log 文件中。

本实验可以两个同学联合试验，分别是客户端和服务端。也可以 1+n 方式，即 1 个服务端，n 个客户端，此时可以通过修正服务器的路由表，把它变成一个安全的路由器（或者更像一个交换机），让 n 个客户端之间互通。{注：理论上完全可以，实际效果有待验证。}

如果尝试证书模式，建议使用 OpenSSL 中的工具产生密钥对及制作证书。细节请查询网络。www.openssl.cn