

北京飞天诚信科技有限公司（以下简称“飞天”）尽最大努力使这篇文章中的内容完善且正确。飞天对于由这篇文档导致的任何形式的直接或间接损失不负有责任。这篇文章的内容会跟随产品的升级而有所变化。

修改记录：

日期	版本	修改
2002 年 5 月	1.0	第一版
2003 年 2 月	1.1	第一版第一次修订
2004 年 1 月	1.2	第一版第二次修订
2004 年 5 月	2.0	第二版
2004 年 11 月	2.1	第二版第一次修订

北京飞天诚信科技有限公司

软件开发协议

北京飞天诚信科技有限公司（以下简称飞天）的所有产品，包括但不限于：开发工具包，磁盘，光盘，硬件设备和文档，以及未来的所有定单都受本协议的制约。如果您不愿接受这些条款，请在收到后的 7 天内将开发工具包寄回飞天，预付邮资和保险。我们会把货款退还给您，但要扣除运费和适当的手续费。

1. 许可使用

您可以将本软件合并、连接到您的计算机程序中，但其目的只是如开发指南中描述的那样保护该程序。您可以以存档为目的复制合理数量的拷贝。

2. 禁止使用

除在条款 1 中特别允许的之外，不得复制、反向工程、反汇编、反编译、修改、增加、改进软件、硬件和产品的其它部分。禁止对软件 and 产品的任何部分进行反向工程，或企图推导软件的源代码。禁止使用产品中的磁性或光学介质来传递、存储非本产品的原始程序或由飞天提供的产品升级的任何数据。禁止将软件放在服务器上传播。

3. 有限担保

飞天保证在自产品交给您之日起的 12 个月内，在正常的使用情况下，硬件和软件存储介质没有重大的工艺和材料上的缺陷。

4. 修理限度

当根据本协议提出索赔时，飞天唯一的责任就是根据飞天的选择，免费进行替换或维修。飞天对更换后的任何产品部件都享有所有权。

保修索赔单必须在担保期内写好，在发生故障14天内连同令人信服的证据交给飞天。当将产品返还给飞天或飞天的授权代理商时，须预付运费和保险。

除了在本协议中保证的担保之外，飞天不再提供特别的或隐含的担保，也不再对本协议中所描述的产品负责，包括它们的质量，性能和对某一特定目的的适应性。

5. 责任限度

不管因为什么原因，不管是因合同中的规定还是由于刑事的原因，包括疏忽的原因，而使您及任何一方受到了损失，由我方产品所造成的损失或该产品是起诉的原因或与起诉有间接关系，飞天对您及任何一方所承担的全部责任不超出您购买该产品所支付的货款。在任何情况下，飞天对于由于您不履行责任所导致的损失，或对于数据、利润、储蓄或其它的后续的和偶然的损失，即使飞天被建议有这种损失的可能性，或您根据第 3 方的索赔而提出的任何索赔均不负责任。

6. 协议终止

当您不能遵守本协议所规定的条款时，将终止您的许可和本协议。但条款 2, 3, 4, 5 将继续有效。

EC Attestation of Conformity



The equipment complies with the principal protection requirement of the EMC Directive (Directive 89/336/EEC relating to electromagnetic compatibility) based on a voluntary test.

This attestation applies only to the particular sample of the product and its technical documentation provided for testing and certification. The detailed test results and all standards used as well as the operation mode are listed in

Test report No. 70407310011

Test standards: EN 55022/1998 EN 55024/1998

After preparation of the necessary technical documentation as well as the conformity declaration the CE marking as shown below can be affixed on the equipment as stipulated in Article 10.1 of the Directive. Other relevant Directives have to be observed.

FCC certificate of approval



This Device is conformance with Part 15 of the FCC Rules and Regulations for Information Technoogy Equipment.

USB



This equipment is USB based.

目 录

第一章 ePass2000 架构	1
1.1 什么是ePass2000	2
1.2 为什么使用ePass2000	2
1.3 ePass2000 的优点	3
1.4 ePass2000 的硬件特性	5
1.5 ePass2000 体系架构	6
第二章 ePass2000 安装与配置	8
2.1 ePass2000 支持的平台	9
2.2 准备安装ePass2000	9
2.3 安装ePass2000 SDK	9
2.4 安装ePass2000 驱动和运行库	15
2.5 卸载ePass2000 驱动和运行库	18
2.6 使用ePass2000 管理器	20
2.6.1 启动ePass2000 管理器	21
2.6.2 配置ePass2000	22
2.6.3 证书管理	28
第三章 计算机密码学与PKI体系	35
3.1 什么是密码学	36
3.2 密码学的起源	36
3.3 什么是加密算法	36
3.4 对称加密算法与非对称加密算法	37
3.5 什么是RSA算法	38
3.6 什么是公开密钥体系(PKI)	39
3.7 什么是SSL	42
第四章 ePass2000PKI应用指南	46
4.1 配置证书颁发机构	47
4.1.1 安装证书颁发机构	47
4.1.2 安装根证书	56
4.2 配置SSL加密站点	64
4.3 使用ePass2000 进行客户证书申请	86

4.4 使用ePass2000 访问SSL加密站点	89
4.5 使用ePass2000 收发签名与加密邮件	89
4.5.1 获取数字证书	90
4.5.2 设置Email帐号的安全性	96
4.5.3 使用Outlook Express发送附加数字签名的邮件	103
4.5.4 获取收件人的公钥和证书	103
4.5.5 使用Outlook Express发送属性加密的邮件	104
4.6 使用ePass2000 进行Windows智能卡登录	106
4.6.1 颁发智能卡证书管理	106
4.6.2 申请智能卡证书	111
4.6.3 锁定工作站	112
4.7 使用ePass2000 进行VPN远程登录	113
第五章 ePass2000 开发指南	120
5.1 ePass2000 的应用开发接口	121
5.2 使用PC/SC接口开发ePass2000 应用	121
5.2.1 智能卡数据库查询函数	122
5.2.2 智能卡数据库管理函数	123
5.2.3 资源管理器句柄函数	124
5.2.4 资源管理器工具函数	124
5.2.5 智能卡监视函数	124
5.2.6 智能卡和读卡器访问函数	124
5.2.7 直接卡访问函数	125
5.3 使用MS CryptoAPI开发ePass2000 应用	126
5.3.1 信息隐藏	126
5.3.2 身份鉴别	127
5.3.3 完整性检测	128
5.3.4 CSP与加密处理	128
5.3.5 CSP上下文	129
5.3.6 CryptoAPI体系架构	130
5.3.7 ePass2000 的CSP模块	131
5.4 使用PKCS#11 接口开发ePass2000 应用	133
5.4.1 ePass2000 支持的PKCS#11 类对象	134
5.4.2 ePass2000 支持的加密算法	135

5.4.3 ePass2000 支持的PKCS#11 接口函数	136
附录 一 ePass2000 技术参数	141

第一章 ePass2000 架构

随着 Internet 的日益普及，人们在信息交换的效率和形式得到飞速进步的同时，个人信息与隐私的安全也日益遭受到前所未有的威胁。对于信息安全有较高要求的用户来说，ePass2000 无疑是一个非常好的选择。

本文将向您介绍 ePass2000 的应用原理与安全特性以及如何使用 ePass2000 提高关键数据的安全性。

本章包括如下主题

- 什么是 ePass2000
- 为什么使用 ePass2000
- ePass2000 的优点
- ePass2000 的硬件特性
- ePass2000 的体系架构

1.1 什么是 ePass2000

ePass2000 是结合了 USB 接口技术和智能卡技术的新一代数据安全产品。它既完美地继承了已有智能卡技术的安全性又结合了新型 USB 接口的数据传输能力。它比传统的智能卡设备更加低廉，而应用更加灵活方便。它体积小巧，方便使用者随身携带。它无需任何附加电源，直接与计算机连接通讯，无需任何读卡器设备。

ePass2000 的驱动程序兼容 PC/SC 标准，任何兼容 PC/SC 的应用程序都可操纵 ePass2000 系列产品。

1.2 为什么使用 ePass2000

传统的智能卡的缺点是价格昂贵，需要额外的读卡器设备，不便于携带，使用很不方便，不能即插即用。而 ePass2000 的出现就很好的解决了这些不足。

ePass2000 采用 Intel 公司制定的新一代个人电脑外设通讯标准 USB (通用串行总线)接口与计算机进行通讯，具有传输数据快，支持即插即用，支持电源管理，可同时串接 127 个外设的优点。相对传统的串行端口和并行端口具有兼容性更好，数据传输更快等优势。只要智能卡的 COS 系统支持，数据传输速率可高达 115,200 波特率，比传统的 9,600 波特率提高将近十倍。

由于 ePass2000 完美集合了智能卡技术，现有的大量智能卡应用无需修改或升级就可以直接使用，为智能卡用户提供了灵活选择的可能。ePass2000 的驱动程序兼容 PC/SC 标准，因此避免了开发者重新学习新的接口，缩短了开发周期。

除了传统的智能卡应用之外，ePass2000 还可广泛地应用到 PKI 体系应用的很多地方。用户可以通过将私钥和数字证书保存到 ePass2000 中，来确保私钥的安全，解决了计算机密码学应用于信息安全中的薄弱环节——存储介质的安全性。

ePass2000 内部使用了运算能力强大的专用芯片，使得计算非常耗时的 RSA 运算可以在芯片的内部实现。这也就是意味着，私钥从生成的时候开始就一直保存在 ePass2000 内部，即使是私钥的所有者也没有权限读取私钥的内容。同时由于 ePass2000 专用芯片所使用的安全封装技术，使得从 ePass2000 中强行获取用户私钥的信息的尝试变得不可能。

众所周知，个人电脑的运行环境是非常不安全的，各种系统漏洞，黑客工具都严重威胁着用户敏感信息（如数字证书，私钥，密码等）的安全。各种信息加密应用程序的安全性，如数字签名，加密电子邮件，用户身份认证等操作都可能受到威胁。这些敏感信息通常都被保存在硬盘中，很容易被盗取。而 ePass2000 则提供了一个独立于个人电脑运行环境的安全存储区域。这些敏感信息都可保存在 ePass2000 内部，加密运算也都在 ePass2000 内部执行，这样就保证了信息的安全。

ePass2000 可用于安全的存储数字证书，私钥和密码，进行电子文档签名，加密操作，保护硬盘数据，对用户身份进行远程或本地的验证。

1.3 ePass2000 的优点

ePass2000 采用了一流工艺制造的智能卡芯片，是保护用户敏感数据的理想设备。其优点包括：

1. 高安全性：

使用基于硬件 RSA 算法的 ePass2000 比使用单纯的软件实现的 RSA 应用更加安全可靠。因为敏感数据都被安全地保存在 ePass2000 的安全存储区域中，未经授权用户是无法接触到这些信息的。数据的签名和加密操作全部在 ePass2000 内部完成，私钥从生

成的时刻起就一直保存其中，可有效的杜绝黑客程序的攻击。
ePass2000 的安全性还在于 ePass2000 使用的加密算法都是被广泛公开，业界公认的，经受了多年考验的算法。同时，一流的芯片封装工艺也保证了芯片内数据的安全性。

2. 灵活易用

使用 ePass2000 无需任何附加的外部设备。用户只要简单的将 ePass2000 插入任何带有 USB 接口的桌面电脑，笔记本，键盘，显示器的 USB 端口中就可以使用 ePass2000。用户不需要关闭计算机或关闭正在运行的程序。使用完毕之后，直接拔下 ePass2000 就可以了。

3. 造价低廉

ePass2000 比任何传统的基于硬件的安全系统都节省开支。由于使用 ePass2000 无需任何附加设备，因此很适合大范围的发行。ePass2000 能够提供智能卡设备提供的所有功能，但是不需要智能卡读卡器。

4. 便于携带

ePass2000 体积十分小巧，重量很轻，可以随身携带。

5. 无缝集成

ePass2000 提供符合业界广泛认可的 PKCS#11 和 Microsoft CryptoAPI 两种标准的接口。任何兼容这两种接口的应用程序，都可以立即集成 ePass2000 进行使用。同时，ePass2000 也针对多个第三方的软件产品进行了兼容性优化。此外，ePass2000 内置大容量的安全存储器，可以同时存储多个数字证书和用户私钥及其他数据。也就是说，多个 PKI 应用程序可以共用同一个 ePass2000。

6. 高可靠性

ePass2000 使用严格工艺制造，非易失性存储区可长期安全的保存用户的数据。

1.4 ePass2000 的硬件特性

➤ 硬件实现的加密算法

ePass2000 采用先进的智能卡技术，智能卡芯片内部可实现下列算法：

- ✧ 1024 位的 RSA 非对称加密算法和签名、校验操作
- ✧ 对称加密算法 DES、3DES
- ✧ 散列函数 SHA-1 和 MD5。

由于关键的加密算法都在硬件内实现，这就保证了进行加密运算的密钥的安全性。

➤ 硬件 RSA 密钥对生成

ePass2000 的 RSA 密钥对在硬件内部实时生成。用于生成密钥的大素数依靠硬件真随机数发生器产生。

➤ 硬件随机数发生器

ePass2000 内置硬件真随机数发生器。ePass2000 在内部使用这个随机数发生器进行密钥对生成，随机消息鉴别码的生成等操作。

➤ 多级访问权限

ePass2000 的文件系统具有多达 16 级的安全权限级别。用户可以定义一个或多个密钥管理安全权级。用户可以根据应用的需要定义出复杂的安全权级关系。

➤ 片内安全存储区域

ePass2000 的数据存储区（RAM）、固件存储区（ROM）及运算部件全都集成在一块芯片内，保证了数据存储的安全。

➤ 加密 USB 数据包

ePass2000 与用户计算机主机的通讯过程是经过加密处理的。有效防止了后台木马程序的监听。

1.5 ePass2000 体系架构

ePass2000 为开发者提供了标准的 PC/SC 接口。开发者可以直接使用微软 Win32 函数集中标准的 PC/SC 函数对 ePass2000 进行操作。

ePass2000 的软件架构，如图 1-1 所示：

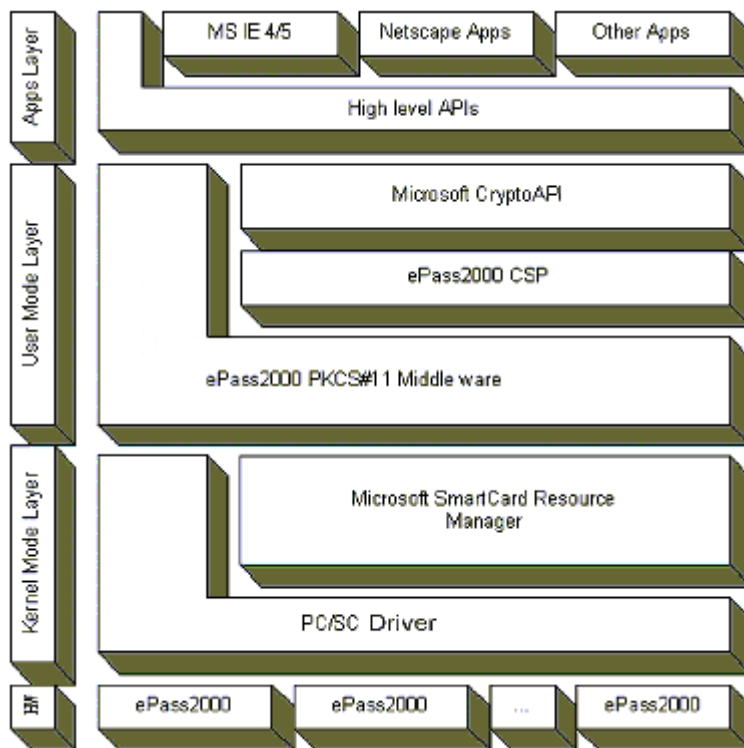


图 1-1 ePass2000 的软件架构

由上图可以看出，ePass2000 的产品架构可分为四个层次：硬件层，核心驱动层，用户接口层和应用层。

硬件层

这一层位于整个架构的最底层。它包括 ePass2000 硬件线路，固件程序和接线。这一层与用户主机之间通过 USB 端口连接，使用标准 USB 通讯协议交换数据。

本手册的附录列出了更详细的硬件细节。

核心驱动层

这一层运行于操作系统的 Ring 0 权级，负责协调用户主机与硬件层之间的数据交互操作和处理上层应用对 ePass2000 的访问请求。这一层与上层应用的接口符合微软 PC/SC 规范的标准驱动接口。也就是说上层应用可以通过标准的 Win32 PC/SC 函数集完成对 ePass2000 的访问。

用户接口层

用户接口实际上可以分为两个层次，低层次的接口负责 ePass2000 的基本 APDU 指令传输和其他一些管理的操作，而高层次的接口本身则建立在低层次接口之上，提供对于 PKI 应用开发必备的功能。

低层次的应用接口实际上提供了智能卡开发者需要的功能。通过这种接口应用程序可以直接向 ePass2000 发送 APDU 指令，对 ePass2000 进行直接的硬件操作。我们提供符合微软 PC/SC 规范的 Win32 PC/SC 应用接口。开发者可以选择使用已经非常熟悉的 Win32 PC/SC 函数集对 ePass2000 进行开发。

高层次的接口主要指 ePass2000 的 PKCS#11 标准接口和 MS CryptoAPI 接口。这两个接口本身就是使用 ePass2000 的低层接口实现的。这两个接口的提供一是为了于现有的应用程序兼容，另外也可供开发商进行二次开发。例如，有的应用需要使用 ePass2000 为用户在浏览器中提交的内容进行数字签名，则可以使用高层次的接口。

应用层

应用层包括已经广泛使用的应用程序和针对 ePass2000 开发的应用程序。由于 ePass2000 提供兼容多种业界标准的接口，即可与现有应用程序兼容也使开发者可以针对已经熟悉的编程接口进行开发。

第二章 ePass2000 安装与配置

在使用之前必须在您的计算机上正确安装 ePass2000 的软件。本章提供在 Windows 环境下安装 ePass2000 软件的详细说明。

本章包含以下主题：

- ePass2000 支持的平台
- 准备安装 ePass2000
- 安装 ePass2000 驱动和运行库
- 卸载 ePass2000 驱动和运行库
- 使用 ePass2000 管理器

2.1 ePass2000 支持的平台

ePass2000 目前支持下列操作系统：

- Windows 98 SE
- Windows ME
- Windows 2000
- Windows XP
- Windows Server 2003
- MAC OS
- Linux

注意：安装 ePass2000 软件之前，请以管理员身份登录系统。

2.2 准备安装 ePass2000

在开始安装 ePass2000 运行库之前，请确定满足以下要求：

- 操作系统为以上列出的操作系统。
- Internet Explorer 5.0 以上版本或 Netscape Communicator 4.7 以上版本。
- 主机上带有至少一个 USB 端口
- 计算机的 BIOS 支持 USB 设备，并且在 CMOS 设置中将 USB 支持功能打开。
- USB 设备延长线或 USB Hub（可选）。
- ePass2000。（在安装驱动之前，不要将 ePass2000 插入计算机的 USB 端口）

2.3 安装 ePass2000 SDK

ePass2000 SDK 会在您的计算机上安装下列组件：

- ✓ ePass2000 图形界面管理器
- ✓ ePass2000 SDK 文档
- ✓ ePass2000 头文件及库文件

- ✓ ePass2000 可分发包
- ✓ ePass2000 示例

注意：安装新版本之前，请卸载旧版本的 ePass2000 SDK。

将安装光盘放入光盘驱动器，然后执行“Setup.exe”，就开始了 ePass2000 SDK 的安装过程。

这时会显示如图 2-1 所示的语言选择窗口：



图 2-1

点击“ok”按钮继续，显示欢迎界面如图 2-2：

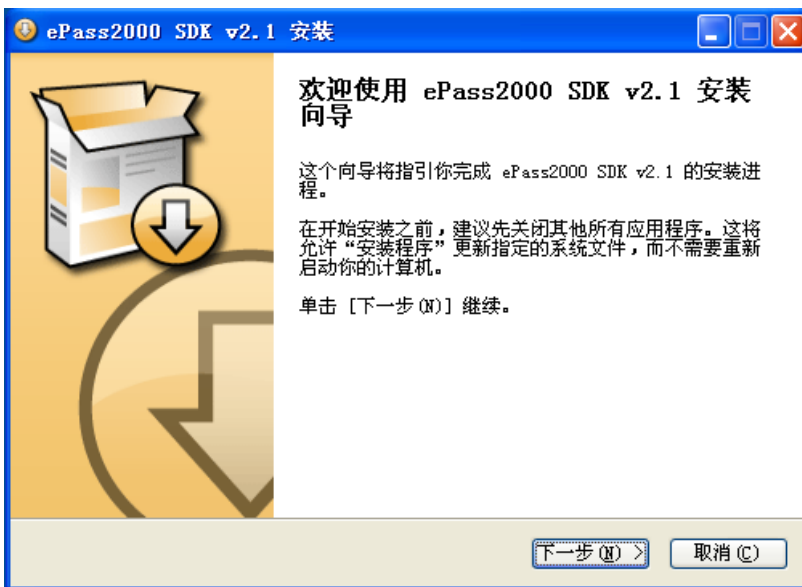


图 2-2

点击“下一步”按钮继续，会显示许可证窗口如图 2-3：

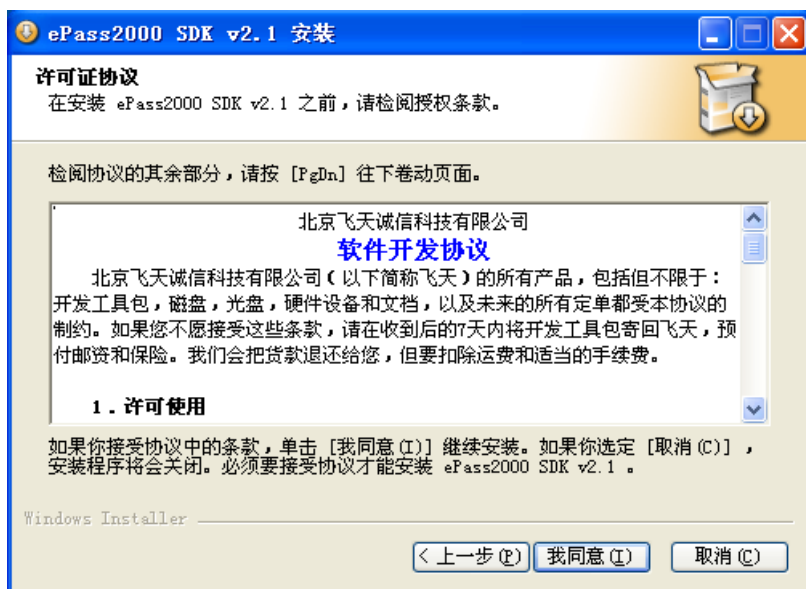


图 2-3

请详细阅读许可证协议。如果您同意这些条款，请选择“我同意”按钮接受许可协议，将进入安装程序的下一步。出现安装选项,如图 2-4 的窗口：

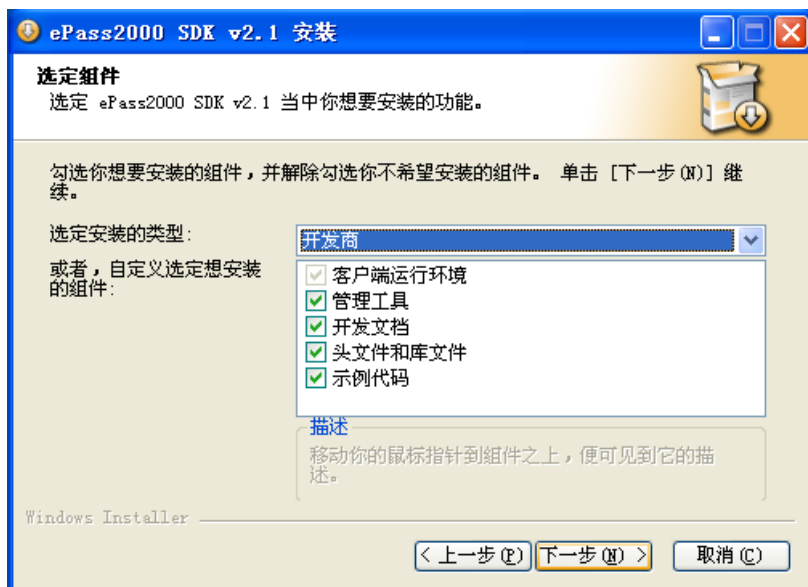


图 2-4

用户可选择安装类型，点击“下一步”按钮继续。出现如图 2-5 窗口：

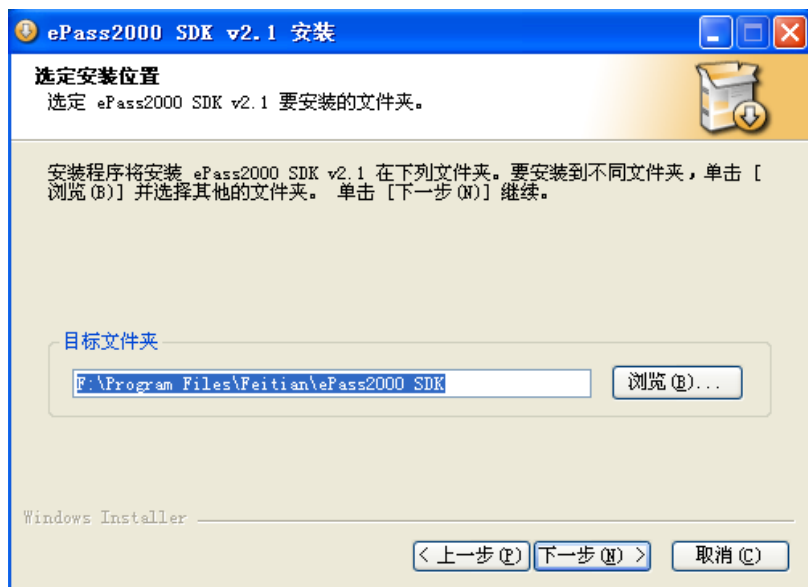


图 2-5

在这个窗口中，用户可以更改安装路径。设置完成后，点击“下一步”按钮，出现如图 2-6 窗口：

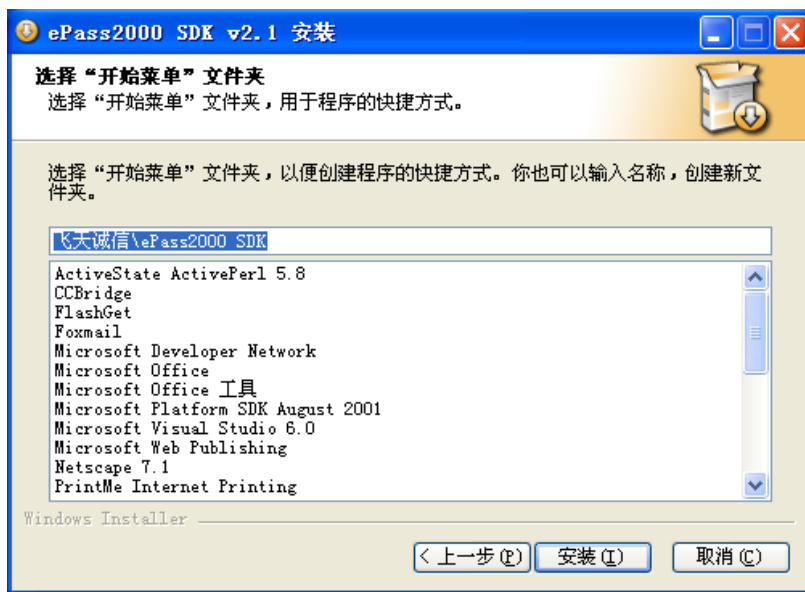


图 2-6

用户可选择“开始菜单”文件夹位置，设置完成后，点击“安装”按钮，安装程序会进入正在安装的界面，如图 2-7。

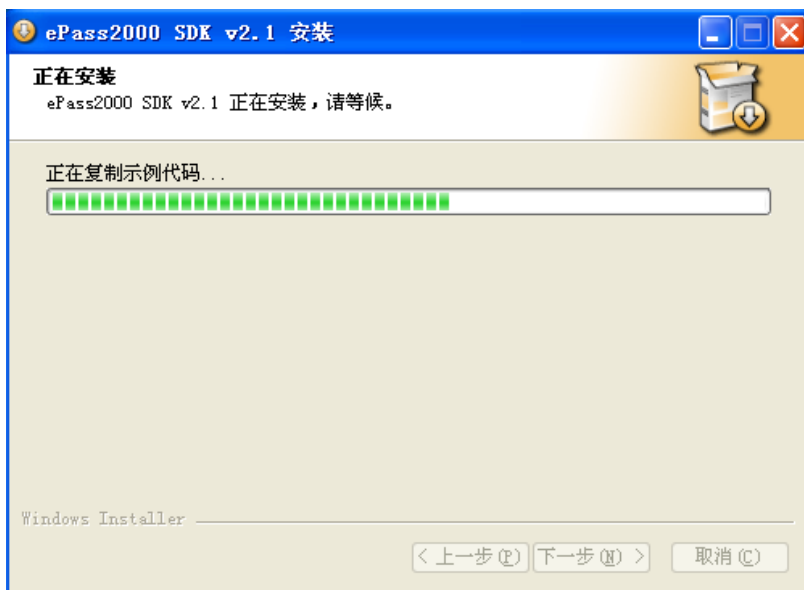


图 2-7

安装完成以后, 出现如图 2-8 所示的界面:

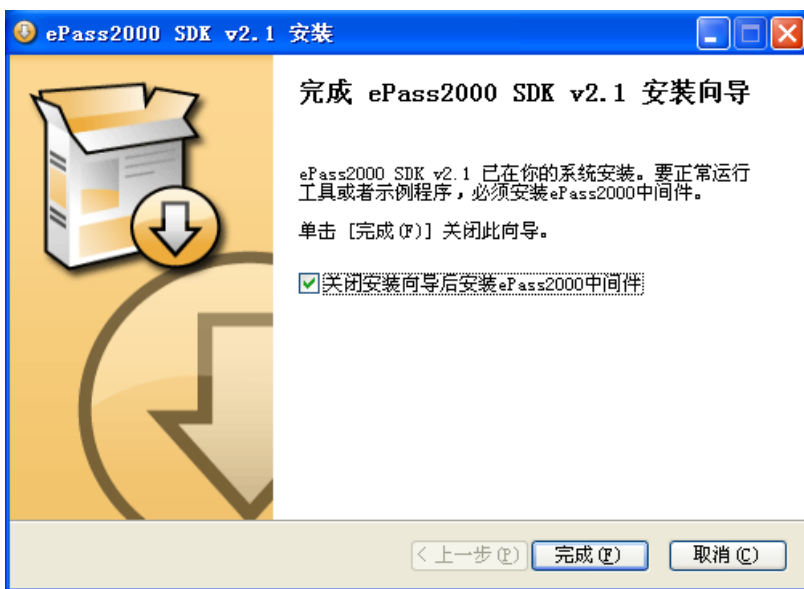


图 2-8

在安装完成的界面，用户可以选择是否接着安装 epass2000 的中间件，如果用户在这里选择不安装，也可以在安装目录的 redist 目录下找到中间件的安装包。

2.4 安装 ePass2000 驱动和运行库

使用 ePass2000 之前，您必须安装 ePass2000 运行库。所有的 ePass2000 运行库组件都包含在 ePass2000 可分发包中。运行库位于光盘 Redist 目录下，在这里，我们根据不同的平台，不同的语言有不同的安装包。

在安装完 ePass2000SDK 时，选择安装 ePass2000 中间件，安装程序将根据用户前面的语言选择和目前用户的平台自动选择 Redist 目录下的安装包进行安装。

开始安装 ePass2000 运行库，首先进入欢迎界面，如图 2-8：



图 2-8

要安装运行库，请点击“下一步”按钮。安装向导会提示是否选择支持智能卡登录操作系统或者 VPN，如图 2-9。

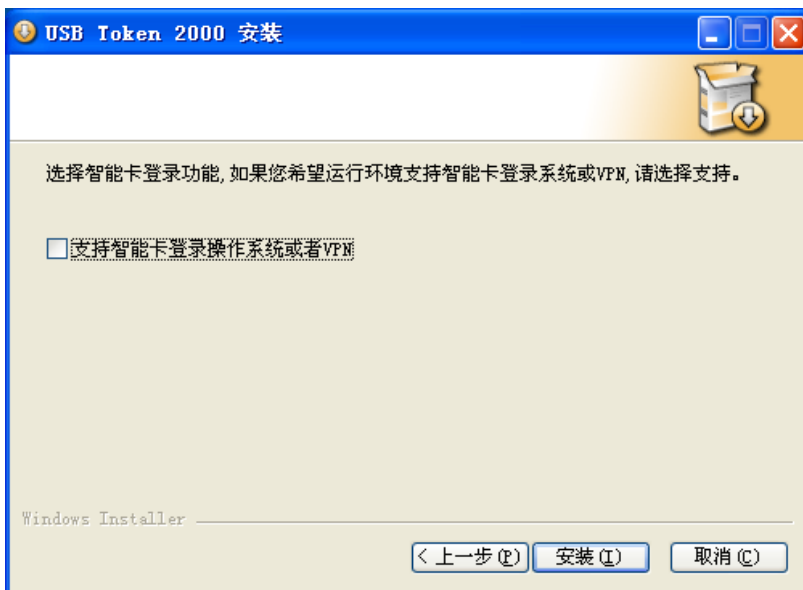


图 2-9

点击“安装”进入正在安装的界面，如图 2-10：

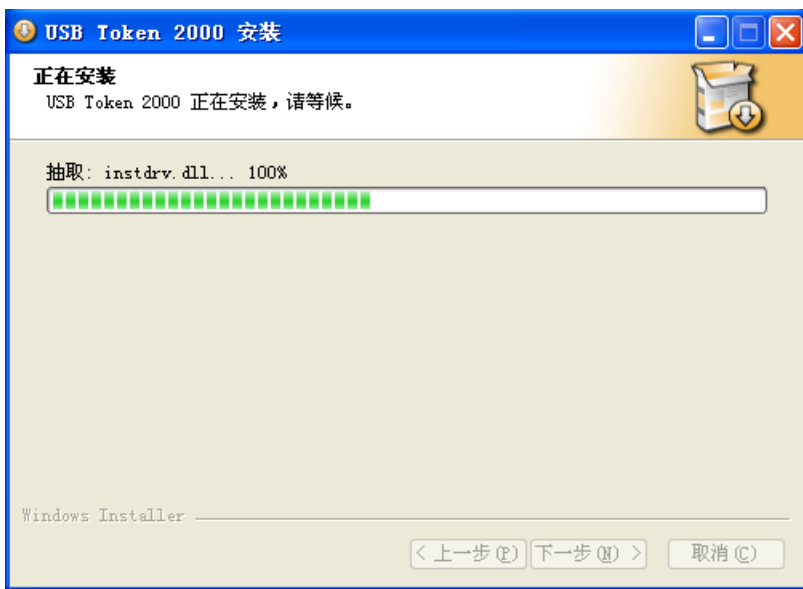


图 2-10

安装完成以后的界面如图 2-11:



图 2-11

点击“完成”按钮即完成了驱动和运行库的安装。

在安装的过程中,如果安装向导检测到 Netscape Navigator 已经被安装则它会自动配置 Netscape 的 PKCS#11 中间件设置,使 Netscape 能够访问 ePass2000。

注意: 如果安装向导提示您需要重新启动计算机,请不要在重新启动之前将 ePass2000 插入到您的计算机上!

如果安装向导提示您需要重新启动计算机,请使用“开始菜单—关闭系统—重新启动计算机”的步骤重新启动计算机,以使刚才安装的驱动程序启动。

成功地结束安装之后,请插入 ePass2000 到您的计算机的 USB 端口上。系统会提示找到新硬件,并且会为检测到的硬件自动激活驱动程序。

2.5 卸载 ePass2000 驱动和运行库

安装了 ePass2000 驱动和运行库之后，您可以通过控制面板的“添加/删除程序”管理工具来卸载它。

请使用“开始—设置—控制面板”的步骤打开控制面板，然后双击“添加/删除程序”打开“添加/删除程序 属性”对话框，在“安装/卸载”的列表选中“USB Token 2000 运行环境”项，然后单击“添加/删除...”按钮，这会启动 ePass2000 安装向导，如图 2-12 所示：



图 2-12

如果确认要卸载，请单击“是”按钮，安装向导会进入卸载欢迎界面，如图 2-13。



图 2-13

点击“移除”按钮，会出现正在卸载的界面，如图 2-14。

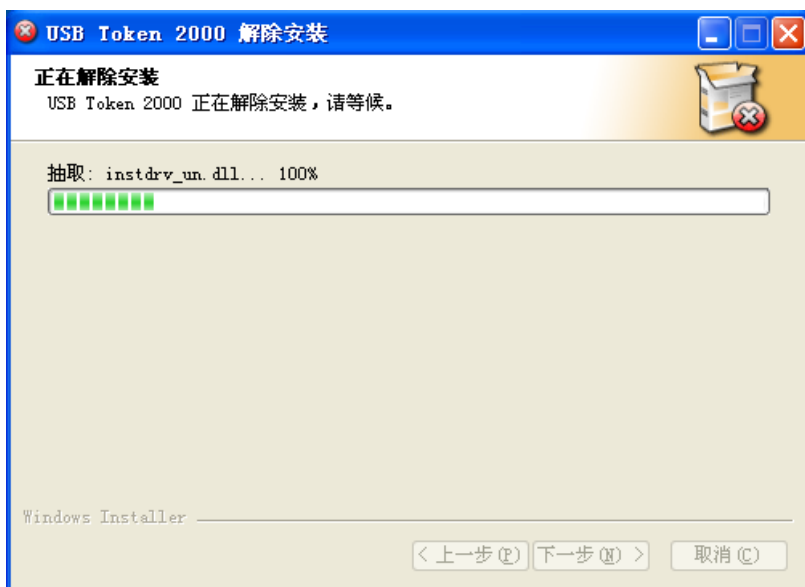


图 2-14

卸载完成以后出现如图 2-15 所示的界面。



图 2-15

在这里，用户可以根据情况选择“重新启动”和“稍候再启动”，点击“完成”按钮即可完成卸载操作。

在卸载的过程中，可能会遇到有些文件因被程序占用无法删除的情况。此时，安装向导会提示用户重新尝试删除，重新启动时删除或不删除。出现这种情况时，用户应检查是否有程序正在访问 ePass2000，如果有，将其关闭，然后重新尝试删除文件，如果还是无法删除，则最好选择在重新启动系统时删除文件。

注意：如果安装向导提示您需要重新启动计算机，请不要在重新启动之前再次安装 ePass2000 驱动程序和运行库或者将 ePass2000 插入到您的计算机上！

2.6 使用 ePass2000 管理器

注意：ePass2000 管理器分开发者版和最终用户版两个版本，二者的区别在于最终用户版没有初始化 ePass2000、用户 PIN 码解锁及改变管理员

PIN 码三个功能。

2.6.1 启动 ePass2000 管理器

如果您还没有插入 ePass2000，或者插入了多个 ePass2000，管理器将拒绝运行。如图 2-16 就显示了没有插入 ePass2000 的错误提示：



图 2-16 没有插入 ePass2000 时管理器拒绝运行

确认已插入 ePass2000 后，单击“是”按钮，会出现如图 2-17 所示的管理器主界面：



图 2-17 ePass2000 管理器主界面

此界面显示了 ePass2000 管理器的版本号、版权等相关信息。

2.6.2 配置 ePass2000

单击“配置”导航按钮，便会看到如图 2-18 所示的界面，此时，在“配置”导航按钮下边出现了 5 个子导航按钮。



图 2-18 配置 ePass2000 的主界面

下面依次对这几项功能进行介绍：

➤ 初始化（最终用户版无此功能）

这是一个基于 ePass2000 的 PKI 应用者首先要做的事情。只有完成了初始化操作，才能进行其他各项操作。在 ePass2000 初始化过程中，将在其硬件的存储空间中划分出一个结构化的区域来存放证书数据。初始化除了要求分配合理的公有和私有数据存储区的大小外，还要对 PIN 码及令牌名进行设定。

图 2-19 为单击“初始化”按钮后的管理器界面：



图 2-19 初始化 ePass2000 的管理器界面

首先您需要输入管理员 PIN 码 (超级用户密码), 初始的管理员 PIN 码为 rockey, 初始化操作是一个不可逆操作, 此操作将对 ePass2000 进行格式化, 原有的数据将全部被删除, 所以只有 ePass2000 发行者才可以进行此操作。

然后您需要设定用户 PIN 码 (普通用户密码), 这里的用户 PIN 码由 ePass2000 发行者设定, 最终用户也可以在“改变用户 PIN 码”功能中更改自己的密码。

最后是设置令牌名。令牌名在某些 PKCS#11 中会被用到。您可以在初始化过程中设置这支 ePass2000 的名称, 也可以在“改变令牌名”功能中更改。名称可以任意设置, 但不要超过 32 个字符。

设置完成后可以看到“初始化”按钮变为可用状态, 单击此按钮, 管理器开始初始化 ePass2000。

注意: 如果您已经填写了各项设置, 但是“初始化”按钮仍然不可用, 那么请检查您的输入是否超出了该项允许的范围。

如果没有成功，系统会自动提示出错信息。

初始化成功后，管理器界面会返回到配置 ePass2000 的主界面，如图 2-18 所示。

➤ 改变用户 PIN 码

在保证已有数据完整性的同时，用户可以改变自己的 PIN 码。在管理器配置 ePass2000 主界面（如图 2-11 所示）中单击“改变用户 PIN 码”或者单击“改变用户 PIN 码”所对应的“开始”按钮，管理器会进入如图 2-20 所示的界面：



图 2-20 改变用户 PIN 码

要改变用户 PIN 码，您必须输入原来的用户 PIN 码，然后设定新的用户 PIN 码。如果您的输入超出了 PIN 码的范围（4~8 个字符），“确定”按钮将无法使用。

注意：如果原来的用户 PIN 码不正确，那么点击“确定”按钮之后，设置操作会失败。如果设置失败，请确定您的旧用户 PIN 码是否输入正确。

➤ 改变令牌名

此功能可以改变令牌名。在管理器配置 ePass2000 主界面（如图 2-18 所示）中单击“改变令牌名”或者单击“改变令牌名”所对应的“开始”按钮，管理器会进入如图 2-21 所示的界面：



图 2-21 改变 ePass2000 令牌名

此项功能可以查看到当前的令牌名，如果您要修改您的令牌名，请在输入框中输入，然后单击“确定”按钮。否则，请单击“取消”按钮。

需要注意的是，修改令牌名不需要任何权限。

➤ 用户 PIN 码解锁（最终用户版无此功能）

当用户忘记密码而无法登录，就需要 ePass2000 分发者解锁。在管理器配置 ePass2000 主界面（如图 2-11 所示）中单击“用户 PIN 码解锁”或者单击“用户 PIN 码解锁”所对应的“开始”按钮，管理器会进入如图 2-22 所示的界面：



图 2-22 用户 PIN 码解锁

因为此项操作涉及到用户信息的安全问题，所以必须由 ePass2000 分发者来完成。首先要输入管理员 PIN 码才能进行，然后设置并确认新的用户 PIN 码（注意不要超过 4 至 8 个字符的限制）。用户现在可以用新的用户 PIN 码来登录了。

➤ 改变管理员 PIN 码（最终用户版无此功能）

在管理器配置 ePass2000 主界面（如图 2-18 所示）中单击“改变管理员 PIN 码”或者单击“改变管理员 PIN 码”所对应的“开始”按钮，管理器会进入如图 2-23 所示的界面：



图 2-23 改变管理员 PIN 码

要改变管理员 PIN 码，您必须输入原来的管理员 PIN 码，然后设定新的管理员 PIN 码。如果您的输入超出了 PIN 码的范围（4~8 个字符），“确定”按钮将无法使用。

2.6.3 证书管理

➤ 登入证书管理工具

由于证书中包含受保护的私有信息，所以只有登入后才能查看完整的证书信息。单击“证书管理”导航按钮便会看到如图 2-24 所示的登入界面：



图 2-24 登入证书管理工具

输入正确的用户 PIN 码后单击“登入”按钮后便可进入如图 2-25 所示的证书管理主界面：

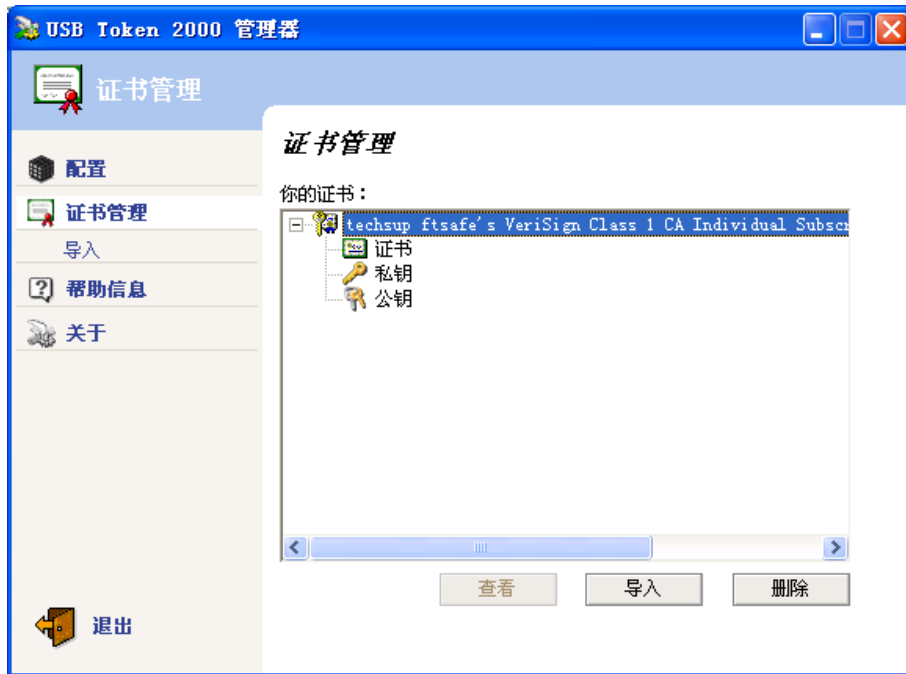


图 2-25 证书管理主界面

如果 ePass2000 中已经存放了证书，那么就会显示该证书的简要信息和状态。证书的简要信息包括证书的颁发对象、证书的颁发机构以及证书的有效期限（起止日期）。证书的状态有三种：有效的证书、无效的证书以及其它数据。

请注意无效的证书和其它数据之间的差别：无效的证书表示这个证书是一个合格的证书，只是现在已经无效，不能使用。无效的原因可能是证书尚未启用、证书已经过期、证书已经被吊销等等。而其它数据表示 ePass2000 中包含一块数据，这块数据不包含证书信息。产生其它数据的原因一般有两种，一种是颁发证书的 CA 使用的证书格式是自己定义的，无法按标准格式解析，一种是在申请证书的过程中发生错误，没有将完整的证书存放到 ePass2000 中。

➤ 查看证书

要查看一个证书的详细内容，只需要在证书列表中双击该证书，或者

选中要查看的证书后单击“查看”按钮。您将看到如果 2-26 所示的证书详细内容窗口：

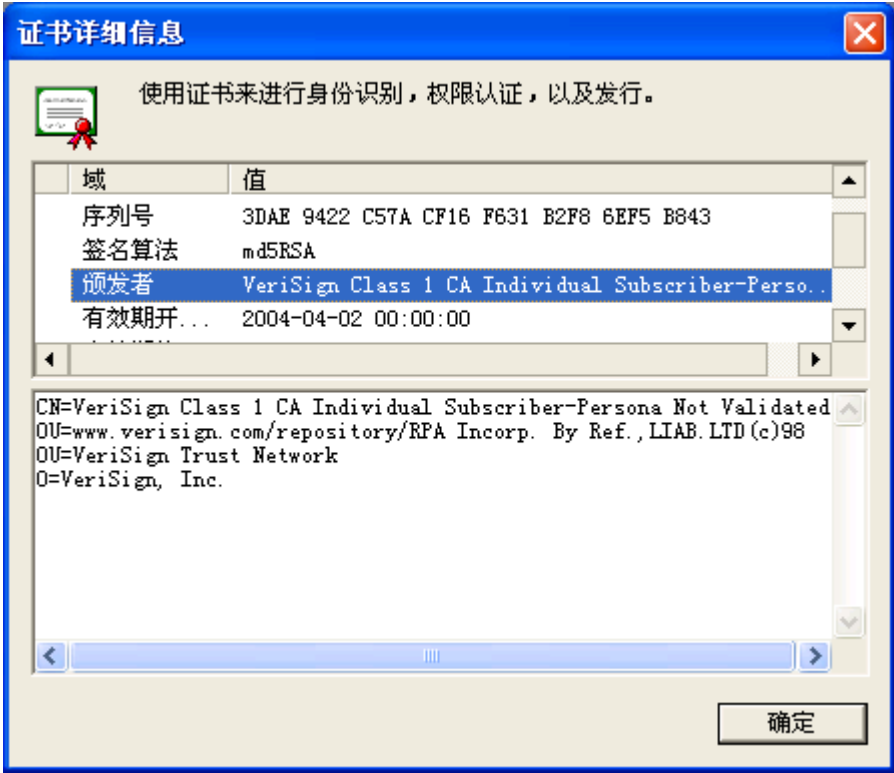


图 2-26 查看证书详细信息

在证书详细信息窗口中，列出了证书的版本、序列号、颁发者、有效日期、公钥信息等详细内容。如果要查看某一项，单击该项，在下面的详细信息框中就会列出该项的更详尽的内容。图 2-26 就列出了详尽的证书颁发者的内容。

如果一个证书已经无效，那么证书详细信息窗口中的列表会在导致无效的项目前面做上红色的标记，如图 2-27 就示范了一个已经无效的证书，在证书截至日期前面有一个红色的标记，方便用户查看：

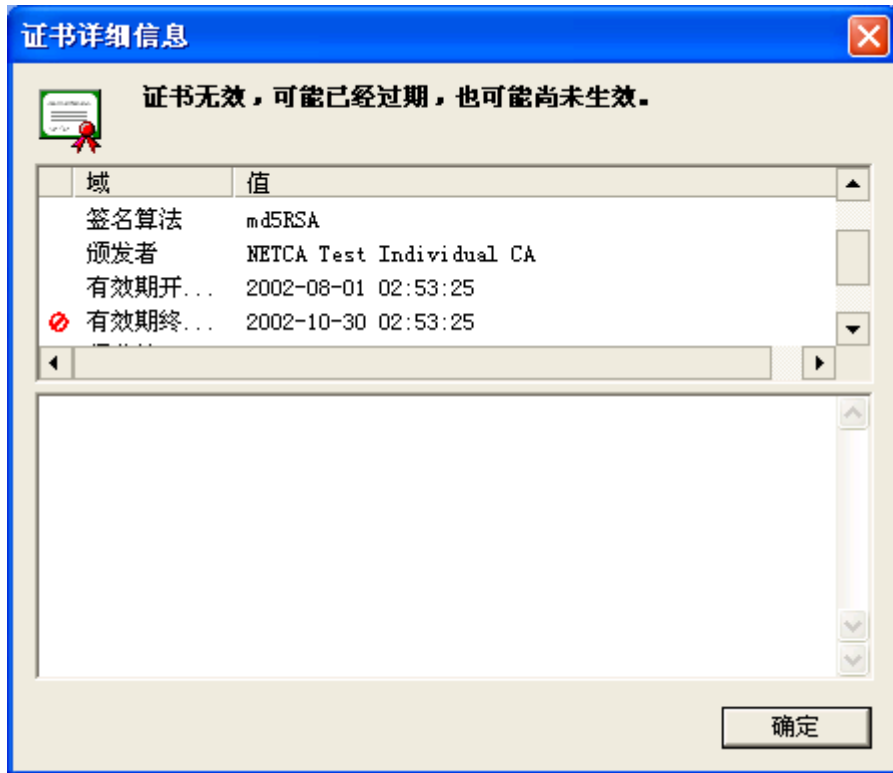


图 2-27 一个无效的证书的内容

在证书管理主界面中无法查看“其它数据”，但您可以将数据块删除，以回收被占用的 ePass2000 存储空间。

注意：因为某些证书颁发机构使用了自定义证书格式，由这些证书颁发机构颁发的证书在管理器中也会显示其它数据，如果删除了这些数据，可能会导致证书无法使用。所以如果没有确切的把握，请不要删除！！

➤ 导入证书

有时您可能需要导入一个证书到 ePass2000 中，例如您原来的证书存放在操作系统中，现在您需要将证书存放到 ePass2000 中以保证证书的安全，就可以使用 ePass2000 管理器提供的证书导入功能。

在 ePass2000 证书管理主界面上单击“导入”按钮，就进入了证书导入界面，如图 2-28 所示：



图 2-28 证书导入界面

您可以在证书文件输入框中输入要导入的证书文件，也可以单击输入框后面的“...”按钮打开文件选择窗口选择一个证书文件，然后您需要输入该证书文件的保护密码，最后单击“确定”按钮就可以了。

注意这里的证书密码，当您从其他的 PKI 应用程序（如 Internet Explorer）中导出证书到一个文件中时，如果导出的数据中包含了私钥，那么为了保证私钥的安全，一般会让您设置一个私钥保护密码，也就是这里需要您输入的证书密码。

另外要注意的是，如果您想将一个现有的证书从其他的 PKI 应用程序导出到文件后再导入到 ePass2000 中保存、使用，那么在导出时必须包括私钥，否则将无法使用该证书。

需要说明的是，证书导入（或者在申请时就直接在 ePass2000 中生成）到 ePass2000 后，私钥将无法从 ePass2000 中导出，这就极大地保护了私钥

的安全。

➤ **删除证书**

如果一个证书已经无效，或者因为其他原因，您可能需要删除 ePass2000 中存储的证书。在证书管理主界面中的证书列表中选中您要删除的证书，然后单击“删除”按钮，此时管理器会要求您确认删除操作。确认无误后，管理器会将该证书从 ePass2000 中删除并收回被占用的存储空间。

第三章 计算机密码学与 PKI 体系

“世界上有两种密码：一种是防止你的小妹妹偷看你的文件；另一种是防止当局阅读你的文件资料。”

——摘自 Bruce.Schneier 《应用密码学》

本章讲的是后一种情况。

- 什么是密码学
- 密码学的起源
- 什么是加密算法
- 对称加密算法与非对称加密算法
- 什么是 RSA 算法
- 什么是公开密钥体系(PKI)
- 什么是 SSL

3.1 什么是密码学

“如果把一封信所在保险柜中，把保险柜藏在纽约的某个地方...，然后告诉你去看这封信，这并不是安全，而是隐藏。相反，如果把一封信锁在保险柜里，然后把保险柜及其设计规范和许多同样的保险柜给你，以便你和世界上最好的开保险柜的专家能够研究锁的装置，而你还是无法打开保险柜去读这封信，这才是安全的概念。”

——摘自 Bruce.Schneier 《应用密码学》

密码学属于信息安全科学的范畴，它的任务就是保护关键信息和敏感数据的安全。

3.2 密码学的起源

最早的密码学应用可追溯到公元前 2000 年古埃及人使用的象形文字。这种文字由复杂的图形组成，其含义只被为数不多的人掌握着。而最早将现代密码学概念运用于实际的人是恺撒大帝（尤利西斯.恺撒 公元前 100 年 — 前 44 年）。他不太相信负责他和他手下将领通讯的传令官，因此他发明了一种简单的加密算法把他的信件加密。

第二次世界大战以后，由于与计算机技术的结合，密码学的理论与实际应用得到了飞速的发展，随之产生了很多新的分支理论，如微粒照片，数字图片水印技术和其他很多隐藏被传递和存储的信息的方法。其中，最常见的就是利用计算机将明文和密码变成密文和将密码和密文变成明文。

3.3 什么是加密算法

所谓加密算法就是指将信息变成密文的计算方法。有的加密算法就是对信息进行简单的替换或乱序，这种加密算法最明显的缺陷就是，算法本身必须保证是保密的。现代加密算法通常都需要密钥来完成对信息的加密运算，算法本身可以公开，理论上，只要保证密钥的安全就能保证信息的安全。

最早的恺撒密文就是一种简单的字母替换加密算法。算法本身非常简单，但同时也是最容易破解的算法。其加密方式就是，按照其在英文字母表里的顺序，将字母循环移位。整个算法可归结为下面的公式：

$$F(x) = (x + s) \bmod 26$$

其中 x 是原文字母， s 是一个常数。例如，如果 s 等于 3，则字母 A 就被加密为 D，而字母 Z 就被加密为 C。这种加密方法虽然简单，但是缺点也是显而易见的。比如，明文中的字母 C 出现的次数是 5 次的话，则加密后对应的字母出现的次数也是 5 次，也就是说字母出现的频率没有变化。比如 E 是英文中最常用的字母，那么给定一个足够大的密文，该文中出现最多的字母很可能就是 E，如果不是，那可能是 A、I 或 Q。密码学专家只用十几个密码字母就能很快的进行这种统计攻击。

现代加密算法与这种简单的字母替换算法不同的地方在于，加密算法的安全性基于用于加密的密钥而不是算法本身。对于好的加密算法，即使公开其算法设计原理也不会对其安全性产生丝毫的影响。只要用于加密的密钥是安全的，则被加密的信息也就是安全的。

3.4 对称加密算法与非对称加密算法

基于密钥的加密算法可以分为两大类：对称加密算法和非对称加密算法(也叫公钥算法)。所谓的对称密钥算法就是用加密数据使用的密钥可以计算出用于解密数据的密钥，反之亦然。绝大多数的对称加密算法加密密钥和解密密钥都是相同的。对称加密算法要求通讯双方在建立安全信道之前，约定好所使用的密钥。对于好的对称加密算法，其安全性完全决定于密钥的安全，算法本身是可以公开的，因此一旦密钥泄漏就等于泄漏了被加密的信息。

所谓非对称加密算法是指用于加密的密钥与用于解密的密钥是不同的，而且从加密的密钥无法推导出解密的密钥。这类算法之所以被称为公钥算法是因为用于加密的密钥是可以广泛公开的，任何人都可以得到加密密钥并用来加密信息，但是只有拥有对应解密密钥的人才能将信息解密。在公开密钥算法体系中，用于加密的密钥被称为公钥，而用于解密的密钥

则称为私钥。

3.5 什么是 RSA 算法

RSA 算法是当今使用最为广泛的非对称加密算法。这个算法是由 Ron. Rivest, Adi. Shamir 和 Leonard. Adleman 三人于 1977 年共同发明的。算法的名称就来自他们三人名字的首字母。

RSA 算法本身是公开的，所有关于这个算法的原理细节都可以从 RSA 公司的网站上找到。RSA 算法的安全性是基于分解一个由两个大素数（素数是只能被 1 和它本身整除的数）相乘所得到的的大数在数学上是非常困难的这一事实。

这两个大素数是随机挑选产生的，用于加密和解密的密钥就是由这两个素数计算产生。这两个素数必须是安全的，因为一旦它们被泄漏则等于泄露了私钥的内容。通常，计算出公钥和私钥后这两个素数都会被销毁，但在某些应用中也可能保留用来加速私钥操作的速度。

在使用 RSA 算法进行加密通讯的过程中，发送信息的一方使用接受方的公钥加密信息，接受的一方收到信息后，用自己的私钥解密信息。发送信息的一方也可以用自己的私钥加密信息，接受方用对应的公钥尝试解密信息以此来确定发送信息方的真实身份。

在 RSA 算法协议中，两个大素数称为 p 和 q ，它们相乘的结果称为模数 n 。公式描述如下：

$$n = p q$$

选择一个数 e ，小于 n ，且与 $(p-1)(q-1)$ 互为质数，也就是说 e 和 $(p-1)(q-1)$ 只有唯一的最大公约数 1。目前，业界的做法是取 $e = 3$ 或者 65537。

然后就是计算 d ，使得 $(ed-1)$ 能被 $(p-1)(q-1)$ 整除。公式如下：

$$d * e \equiv 1 \pmod{(p-1)(q-1)}$$

通常所说的 RSA 公钥就是 (n, e) 二元组，而私钥就是 (n, d) 二元组。而最初选择出来的两个大素数 p 和 q ，则可根据应用的需要销毁，或保存下来加速 RSA 运算。

如果要使用 RSA 算法加密一段信息 m ，则首先要将 m 分割成长度小于 n 的长度的多个数据块。也就是说如果 p 和 q 是 512 位的素数，则 n 就为 1024 位的合数，而每段被加密的信息的位数必须小于 1024。加密的过程就是对信息进行一次模运算，公式如下：

$$C = m^e \text{ MOD } n$$

而解密的过程则为：

$$m = C^d \text{ MOD } n$$

签名与校验签名的过程与加密和解密类似。

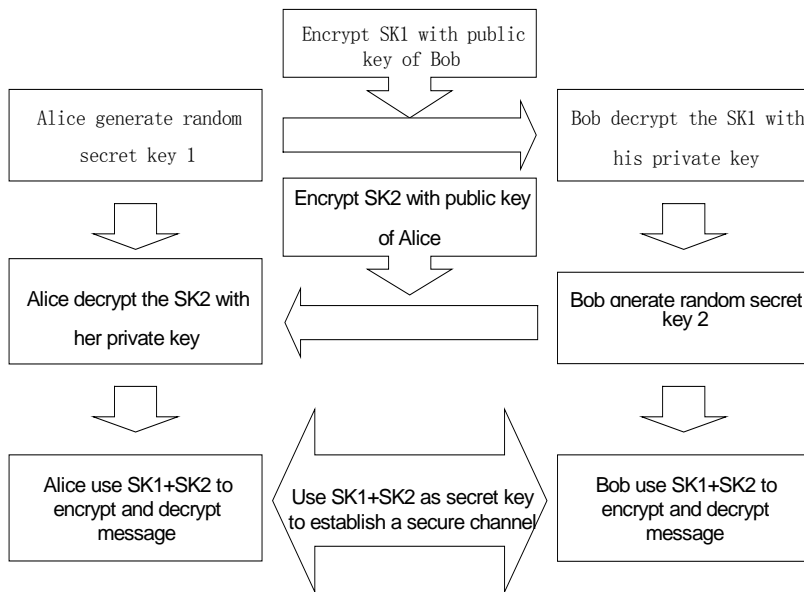
3.6 什么是公开密钥体系(PKI)

传统的加密系统是基于对称加密理论的，也就是信息采用单密钥加密，其特点是加密密钥与解密密钥可互相推导，信息的发送者和接收者在建立安全通讯信道之前必须商定一个密钥。对称加密算法的安全性依赖于密钥，泄漏密钥意味着任何人都能对信息进行加密和解密。随着对称加密理论的发展，出现了很多对称加密算法。对称加密算法具有速度快，实现简单等特点，能很好地解决数据的保密传输的问题，但是对称加密算法系统的致命弱点在于密钥的安全性，致使对称加密体系解决不了密钥分配和管理的问题。

1976 年，Whitfield . Diffie 和 Martin . Hellman 首次公开提出了公开密钥理论，奠定了 PKI 体系的基础。PKI 即 Public Key Infrastructure 的缩写，也就是所谓“公开密钥体系”，是一种利用现代密码学的公钥密码技术在公开的网络环境中提供数据加密以及数字签名服务的，统一的技术框架。常用的公开密钥算法有 RSA，DSA 和 Deffie . Hellman (DH) 算法等。使用公开密钥算法（有时也叫非对称加密算法）的用户同时拥有匹配的公钥和私钥。私钥由用户保存，且不能泄漏，公钥则要广泛公开的发布。私钥无法通过公钥计算获得。公开密钥理论最大的优势是解决了对称加密系统无法

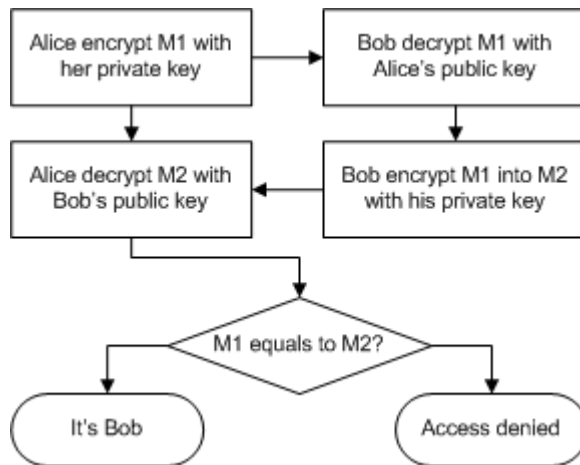
很好解决的密钥交换问题。

公开密钥的算法速度比相同强度的对称算法要慢得多，并且由于任何人都能得到用户的公钥，公开密钥算法对选择明文攻击十分脆弱，因此公钥加密/私钥解密不适用于大量的数据加密传输。为了实现数据的加密传输，公开密钥算法需要与对称加密算法结合使用，即公开密钥算法负责密钥交换，而对称加密算法则负责实际的数据加密。下图显示了一种常用的密钥交换方法：



在使用公开密钥算法进行密钥交换的过程中，密钥数据使用公钥加密。在保证用户私钥安全的前提下，攻击者即使截获传输的信息也不能得到用于加密数据的密钥。在整个密钥交换过程中，只要保证用户私钥安全，公钥不被篡改，就能保证通信的安全。

公开密钥体系除了可用于安全密钥交换之外，还可用于鉴别用户身份。下面是一个简单的鉴别用户身份的例子：



同样原理，公开密钥算法可以进行数据的签名和校验操作，保证数据的一致性和完整性。Alice 将数据和自己对数据的签名一起发送给 Bob, Bob 使用 Alice 的公钥解密得到数据的签名，然后与接收到的数据进行比较，如果一致则证明数据没有被第三者篡改。

因为非对称算法运算速度比较慢，对数据签名一般不采用直接加密数据的方式，而是加密数据的散列值。数据块的散列值是通过消息摘要算法计算生成。消息摘要算法实际上是一种单向散列函数。数据经过单向散列函数计算得到一个固定长度的值，消息不同得到的散列值也有很大差异。由于是单向函数，用户不可能从散列值推算出原数据，这样就保证了攻击者无法通过散列值伪造数据块。比较消息的散列值与比较消息本身是等价的。常用的消息散列算法有 MD5 和 SHA-1 等。

公钥算法仍然要面临公钥分发，公钥、私钥与用户真实身份绑定的问题。PKI 引入了证书机制解决了这个问题。证书由证书注册中心，也就是常说的 CA 中心统一颁发。

用户获得自己的证书之后，就可以使用证书来表明自己的身份，接受方只要使用 CA 中心的公钥验证用户的证书，如果验证成功，就可以信任该证书的用户的身份。证书的颁发和验证充分利用了公开密钥算法的数据

签名和验证功能，从原理上杜绝了冒充身份的可能。

3.7 什么是 SSL

所谓 SSL 就是安全套接字层，是当今互联网环境中使用最为广泛的密码学应用之一。

Internet 是一个开放的网络环境，开放的网络环境意味着在网络上传输的任何数据都有被截取和监听的可能。当你在一个 WWW 网站提交一些私人信息的时候，这些信息从你的计算机传出，通过 Internet 上的若干节点之后，达到服务器，在这中间的过程中，这些数据是完全暴露的，任何人只要有适当的工具都可以截取这些数据。

正是由于这一安全隐患的存在，Netscape 公司开发了 SSL 来解决这一问题。SSL 是由客户端浏览器创建的一个用来进行信息的安全传输的通讯层。这个安全套接字层位于 TCP/IP 协议簇层和浏览器及 HTTP 等应用层之间的一层，透明的完成信息的加密传输任务。浏览器的 SSL 实现采用了 RSA 公司的公、私钥加密系统和数字证书应用系统。所有当今主流的浏览器都内置了对 SSL 的支持，当用户访问需要 SSL 认证的页面时，浏览器会正确识别和处理相应的请求。SSL 可以有效的防止传输数据被监听和篡改，客户端和服务端可以经过相互验证来建立一条安全的通讯信道。

在 SSL 应用当中，用户敏感信息的安全是依靠高强度的加密算法来保障的。SSL 使用 RSA 加密算法，数字证书系统和数字签名机制来确保数据的安全性。用户的数据通过由 SSL 建立起来的网络连接进行传输时可以保证其隐秘性，一致性和完整性。由于 RSA 运算速度的问题，在每次 SSL 会话当中，RSA 运算只在服务器端和客户端各进行一次。在这个过程中，服务器和客户机经过一系列的交互建立其一个安全的通讯信道，这个过程称作 SSL 握手。

SSL 的握手过程可分为如下几个步骤：

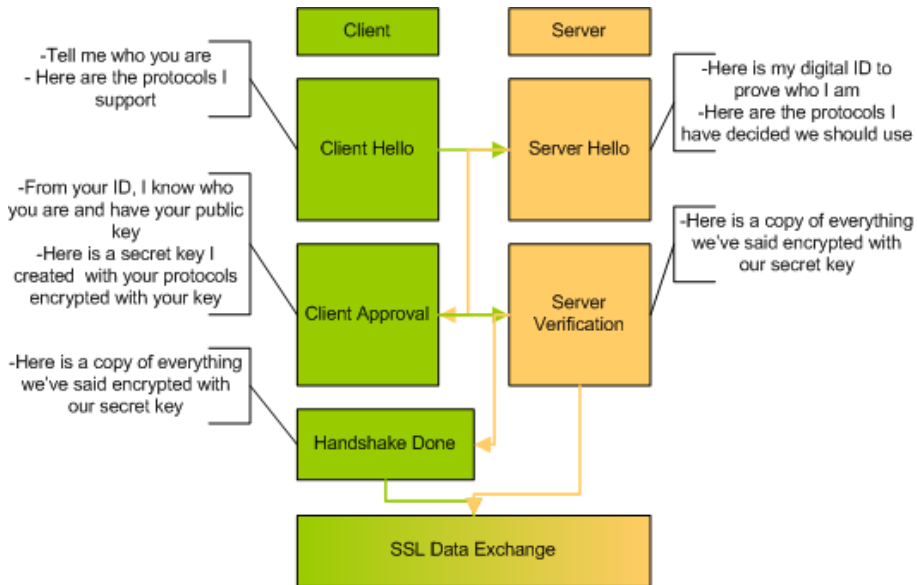
1. 客户端向服务器端发送客户端的 SSL 版本号，加密算法设置，

随机产生的数据和其他服务器需要用于跟客户端通讯的数据。

2. 服务器向客户端发送服务器的 SSL 版本号，加密算法设置，随机产生的数据和其他客户端需要用于跟服务器通讯的数据。另外，服务器还要发送自己的证书，如果客户端正在请求需要认证的信息，那么服务器同时也要请求获得客户端的证书。
3. 客户端用服务器发送的信息验证服务器的身份。如果验证不成功，用户就将得到一个警告，然后加密通讯连接将无法建立。如果成功，则继续下一步。
4. 用户用握手过程至今产生的所有数据，创建连接所用的 **Premaster Secret**，用服务器的公钥加密（在第二步中传送的服务器证书中得到），创送给服务器。
5. 如果服务器也请求客户端验证，那么客户端将对另外一份不同于上次用于建立加密连接使用的数据进行签名。在这种情况下，客户端会把这次产生的加密数据和自己的证书同时传送给服务器用来产生 **Premaster Secret**。
6. 如果服务器也请求客户端验证，服务器将试图验证客户端身份。如果客户端不能获得认证，连接将被中止。如果被成功认证，服务器用自己的私钥加密 **Premaster Secret**，然后执行一系列操作产生 **Master Secret**。
7. 服务器和客户端同时产生 **Session Key**，之后的所有数据传输都用对称密钥算法来交互数据。
8. 客户端向服务器发送信息说明以后的所有信息都将用 **Session Key** 加密。至此，他会传送一个单独的信息标识客户端的握手部分已经完成。

9. 服务器也向客户端发送信息说明以后的所有信息都将用 Session Key 加密。至此，他会传送一个单独的信息标识服务器端的握手已经完成。
10. SSL 握手至此成功结束，客户机和服务器开始用 Session Key 加密和解密双方交互的所有数据。

下图是一个常见的 SSL 握手过程：



在上面使用数字证书进行身份验证的过程中，服务器端的验证与客户端有些差别。客户端使用数字证书验证服务器的过程如下：

1. 服务器端传送的证书中获得相关信息。
2. 当天的时间是否在证书的合法期限内。
3. 签发证书的机关是否客户端信任的
4. 签发证书的公钥是否符合签发者的数字签名
5. 证书中的服务器域名是否符合服务器自己真正的域名。
6. 服务器被验证成功，客户继续进行握手过程

服务器端使用数字证书验证客户端的过程如下：

1. 客户端传送的证书中获得相关信息。
2. 用户的公钥是否符合用户的数字签名。
3. 当天的时间是否在证书的合法期限内。
4. 签发证书的机关是否服务器信任的。
5. 用户的证书是否被列在服务器的 LDAP 里用户的信息中。
6. 得到验证的用户是否仍然有权限访问请求的服务器资源。

在通过 SSL 应用获得较高的通讯安全性的同时，服务器也需要付出很大的代价。对于一个点击率很高的商业网站来说，如果使用 SSL 保护其站点，会极大的增加服务器的负担。因为服务器需要为每一个请求的用户进行 RSA 运算。解决这一问题的方法就是使用更多的辅助服务器分担客户访问的処理和使用附加的 RSA 运算加速硬件产品分担服务器 CPU 进行 RSA 运算的负荷。

第四章 ePass2000PKI 应用指南

ePass2000 的设计目标之一就是与现有的 PKI 体系应用无缝的集成。PKI 应用开发商无需对 ePass2000 进行任何形式的编程就能通过配置相关服务而开始将 ePass2000 集成于 PKI 应用当中。本章主要讲述如何配置 ePass2000 的 PKI 应用。

- 配置证书颁发机构
- 配置 SSL 加密站点
- 使用 ePass2000 申请数字证书
- 使用 ePass2000 访问 SSL 加密站点
- 使用 ePass2000 收发签名与加密邮件
- 使用 ePass2000 进行 Win2000 智能卡登录
- 使用 ePass2000 进行 VPN 远程登录

4.1 配置证书颁发机构

证书颁发机构亦即通常所说的 CA 中心, 是 PKI 应用的核心。任何 PKI 应用都需要 CA 中心的支持。Windows Server 2003 系统内建了很多对 PKI 应用的支持, 通过适当的配置可实现智能卡登录, 锁定工作站, VPN 远程登录, SSL 加密站点访问等功能。下面我们将以 Windows Server 2003 自带的证书颁发机构为例, 讲解配置证书颁发机构的一般步骤。

4.1.1 安装证书颁发机构

Windows Server 2003 的安装程序缺省设置下并不会自动安装证书服务。这是由于安装完证书服务后, Windows Server 2003 计算机就无法再更改计算机名称了。为了提高系统管理灵活性, 所以 Windows Server 2003 并未将证书服务安装到用户的 Windows Server 2003 计算机上。所以, 当用户要在 Windows Server 2003 计算机上安装证书服务时, 用户需要由“添加/删除程序”中的“Windows 组件”, 选择安装证书服务。

注意: 如果用户没有安装 IIS, 请参照 4.2 先安装 IIS。

若要在 Windows Server 2003 计算机上安装证书颁发机构(CA), 请按照下列的步骤进行操作:

1. 以系统管理员权限的帐号登录 Windows 2003 系统。
2. 请依序打开“开始”菜单→“设置”→“控制面板”选项, 以启动 Windows 2003 控制面板。
3. 接着, 选择“添加/删除程序”, 启动添加/删除程序, 如图 4-1 所示。

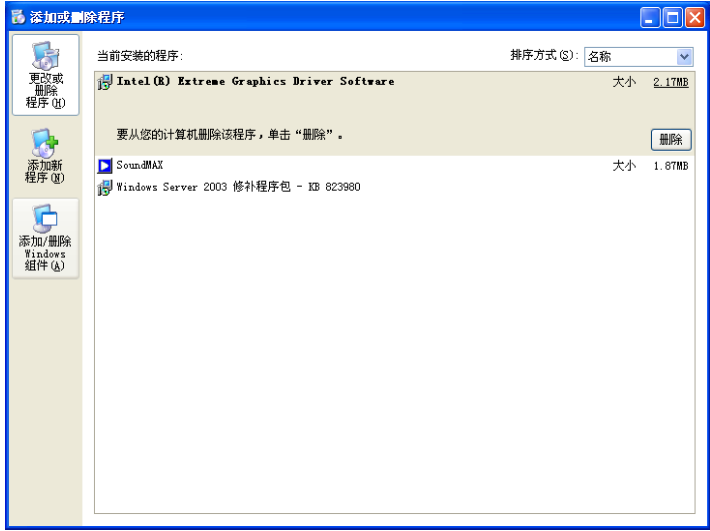


图 4-1 添加/删除程序

4. 接着，请选择“添加/删除 Windows 组件”选项，这时候，系统会启动 Windows 组件向导，让用户选择想安装的 Windows Server 2003 操作系统的相关服务或工具的组件。如图 4-2 所示。

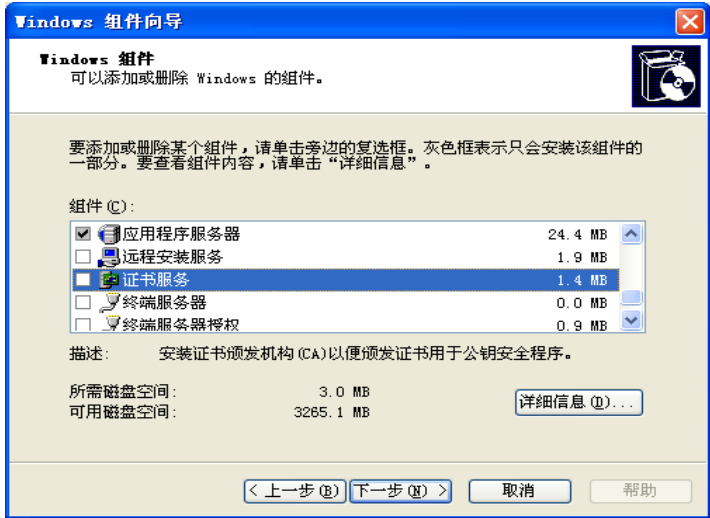


图 4-2 添加/删除 Windows 组件向导

5. 请在 Windows 组件向导的“组件”列表里，选择“证书服务”的选项，以便在 Windows Server 2003 计算机上安装证书服务。当在 Windows Server 2003 计算机上安装了证书服务后，这部 Windows Server 2003 计算机就会成为证书颁发机构主要的参考计算机，因此，就无法在 Windows Server 2003 计算机上重新为计算机命名了，而且也无法加入其他的域、或者由现存的域中删除。因此，当用户要安装证书服务前，请先确定这部 Windows Server 2003 计算机的稳定性。
6. 当勾选“证书服务”的选项后，请接着按“下一步”按钮。接下来，系统会出现证书授权类型的设置过程。只需要按照需要，选择要安装的证书颁发机构(CA)的类型即可。用户可以选择设置的各种证书颁发机构的类型以及用途如下(参见图 4-3)：

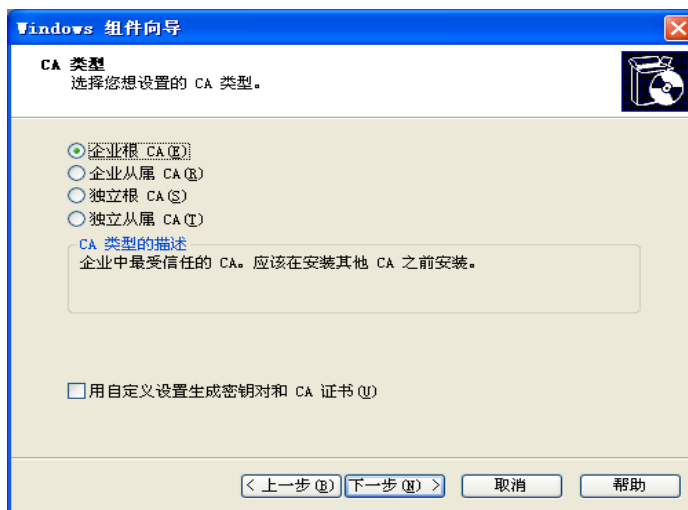


图 4-3 选择证书颁发机构类型

企业根证书颁发机构(Enterprise Root CA):如果所设置的证书颁发机构要将证书发行到企业Active Directory域内所有的个体上，用户就必须选择此选项。请注意，此部证书颁发机构将会登记在Active Directory域内。如果企业的硬件资源足够时，建议只将企业根证书颁发机构(Root CA)使用在发行授权(证书)

给企业从属证书颁发机构(Subordinate CA)之用,因为这样可以确保较好的安全性。如果企业域内部目前并没有任何的证书颁发机构,也必须选择安装根证书颁发机构(Root CA)。

企业从属证书颁发机构(Enterprise Subordinate CA):如果设置的证书颁发机构要将证书发行到企业Active Directory域内的每一个个体上,而且企业域上已经有一台企业根证书颁发证书机构,就可以选择此选项。请注意,此部证书颁发机构将会登记在Active Directory域内。

独立根证书颁发机构(Stand-alone root CA):如果所安装这部证书颁发机构将要发行证书给企业域外部的个体使用时,就必须选择这种证书颁发机构方式。选择了这种方式的证书颁发机构,将会成为一个证书颁发机构层次架构的独立根证书颁发机构。

独立从属证书颁发机构(Stand-alone subordinate CA):如果要将此部证书颁发机构设置为一个已经设置好的证书层次架构里的一员,就应该选择此选项。证书层次架构组织可以是用户之前所安装的独立证书系统,也可以是存在于企业外部的一个商用性证书颁发机构。

在图 4-3 中显示选择了企业根证书颁发机构。

在 Windows Server 2003 操作系统的证书服务器上已经采用了默认的加密系统,提供证书的安全机制。若要设置证书颁发机构一些高级设置值(例如证书颁发机构所使用的加密服务提供者(CSP)、数字签名或信息完整性检查所使用的散列算法、证书所使用的密钥长度、所使用的密钥类型等),可以勾选下方的“高级选项”复选框。若勾选了此选项的话,当按下“下一步”按钮时,接下来会出现“公钥/私钥对”的设置窗口。如图 4-4 所示:



图 4-4 公钥/私钥对高级设置

此对话框可以让您更改系统默认的加密功能，像是使用哪一种加密服务提供者（CSP）、使用哪一种散列算法等等。在上面的对话框里的每一种加密功能的选项（例如加密服务提供者、散列算法等），是根据目前您这部 Windows Server 2003 计算机上所有的软硬件的支持能力所提供的选项出现在上图的设置窗口里。

用户可以在“密钥长度”的选择框里调整数据加密时所使用的密钥长度。一般来说，密钥长度越长，加密出的密文越安全，但是所需要的加密/解密时间越久。如果选择“默认”的密钥长度，系统会根据所选择的加密服务来自动设置所需要的密钥长度。我们建议用户在许可的范围内，尽量选择较长的密钥长度，所需要计算加密/解密的时间可能会较长，而且可能有些硬件会无法支持较长的密钥长度（因为有些硬件设计空间或其他因素，限制密钥长度的使用）。如果要使用目前存在系统内的一些密钥建立证书颁发机构，请选择下方的“使用现有密钥”框以及“导入”按钮来设置此证书颁发机构所使用的密钥。

完成上述的设置后，请按“下一步”按钮，继续证书颁发机构的安装设置。

7. 接下来，向导会出现“CA 标识信息”的设置窗口。用户必须在此窗口里设置此证书颁发机构的标识信息，如图 4-5 所示：



图 4-5 证书颁发机构标识信息

在这里请用户要特别注意，在“CA 名称”的字段上，用户务必为此证书颁发机构命名一个名称，因为稍后将会使用此名称来标识建立在证书服务器上的证书颁发机构对象。

如果用户建立的是企业型的证书颁发机构，此名称将会使用来标识建立在 Active Directory 域内的证书颁发机构对象，如果用户建立的是独立证书颁发机构，此名称将会使用在标识此证书颁发机构上。在这里，还需要请读者注意另外一点，如果所设置的是根证书颁发机构(Root CA)，那证书颁发机构的“有效期限”需要比较长的时间，至少都需要比从属证书颁发机构的有效时间长。如果设置的根证书颁发机构，请将“有效期限”设置在一个合理的时间。当然用户必须考虑到安全以及系统管理的负担，在这两个相反的考虑上获取一个平衡点。当根证书服务器的有效期限过期时，系统管理人员就必须重新刷新一次所有的信任关系。

当完成上述的设置后，请按“下一步”按钮，继续下一个证书颁发

机构的设置过程。

8. 接下来, 向导会出现“数据储存位置”的窗口 (参见图 4-6), 此窗口主要的目的是要指定证书数据库的储存位置、证书服务器设置信息的储存位置、储存证书撤销列表的位置以及证书数据库记录文件的位置。



图 4-6 指定相关数据储存位置

如果所设置的证书颁发机构类型为企业型的证书颁发机构, 则企业型的证书颁发机构会将它的一些设置信息以及属性信息存储在域里 (域控制器上)。

若不是在域控制计算机上设置证书服务器的话, 请选择“共享文件夹”选项, 并输入一个位于本地上的共享文件夹路径, 用来指定证书颁发机构设置信息的存储位置 (用户可以指定在共享文件夹里, 这样即使未参与域的客户端机器, 也能够获取证书撤销列表的相关信息)。

当完成上述的设置后, 请按“下一步”按钮, 继续下一个证书颁发机构的设置过程。

9. 如果安装的是一个从属证书颁发机构，用户将会看到“CA 证书申请”的设置窗口（如果您安装的不是从属证书颁发机构，请跳到第 10 个步骤继续证书颁发机构的设置过程）。之前，我们曾经提到过，从属证书颁发机构会直接向根证书颁发机构获取证书信息，在这里，就是要设置此从属证书颁发机构要向哪一台 Windows 2000 计算机上的根证书颁发机构，获取证书颁发机构的证书信息。用户可以选择采用网络直接传输的方式，或者以文件形式的方式，来获取证书颁发机构的证书信息。若采用网络直接传输的方式，用户只要指定根证书颁发机构计算机名称、以及证书颁发机构的名称即可。若采用文件形式来获取根证书颁发机构的证书信息，必须指定存储证书信息的文件位置（参见图 4-7）。

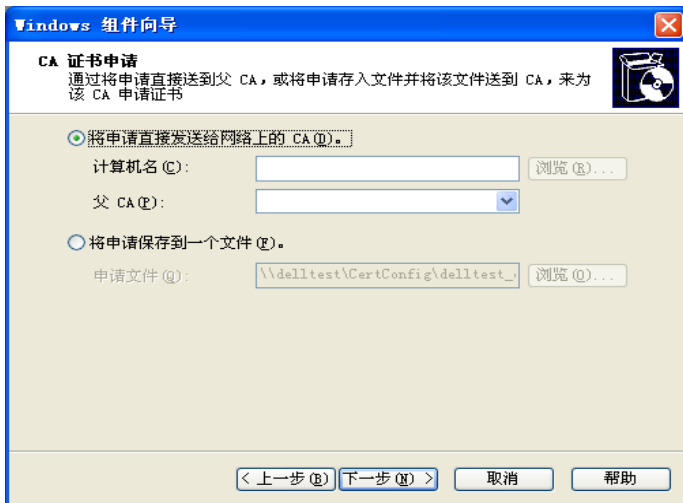


图 4-7 选择获取证书授权的主要证书颁发机构

用户可以选择“将申请直接发送给网络上的 CA”选项，并按下“浏览”按钮，选择一台可以获取证书授权的根证书颁发机构计算机以及证书颁发机构。如果必须由特定的商用证书颁发机构获取授权证书信息，或者需要获取授权的证书颁发机构无法由网络上获取授权信息时，用户可以选择“将申请保存到一个文件”

的选项，并将此文件带到指定的主要证书颁发机构上处理，获取发行证书的授权。

当完成上述的设置后，请按“下一步”按钮，继续下一个证书颁发机构的设置过程。

10. 因为Microsoft的证书服务也直接支持其他IIS服务器的运行，因此，如果这时候您的Microsoft Internet 信息服务器 IIS (Microsoft Internet Information Server) 还在运行阶段，系统会出现提示信息，要求您先停止 IIS 的运行，以便顺利安装证书服务器 (参见图 4-8)。

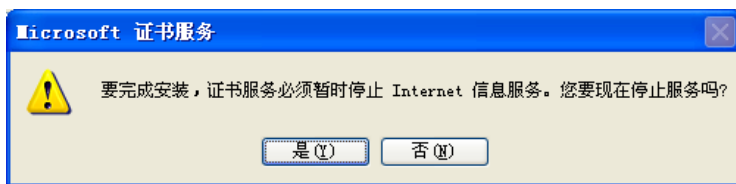


图 4-8 要求停止 IIS 的执行

11. 当确定后，接下来系统便开始安装证书服务器相关的组件以及程序，如图 4-9 所示。



图 4-9 证书服务器组件安装

12. 请注意一下%SystemRoot%\system32\CerSrv\CertEnroll文件夹是共享的。因为证书服务的客户端计算机需要获取此目录下的信息，以便核对撤销的相关信息。如果此磁盘文件夹没有处于共享状态，可能证书服务客户端计算机无法正常运行。
13. 这时候证书服务器已经成功地安装在 Windows Server 2003 计算机上了。已经可以由“开始”菜单→“程序”→“管理工具”→“证书颁发机构”选项，启动证书颁发机构系统管理工具(见图 4-10)，来管理证书服务器。

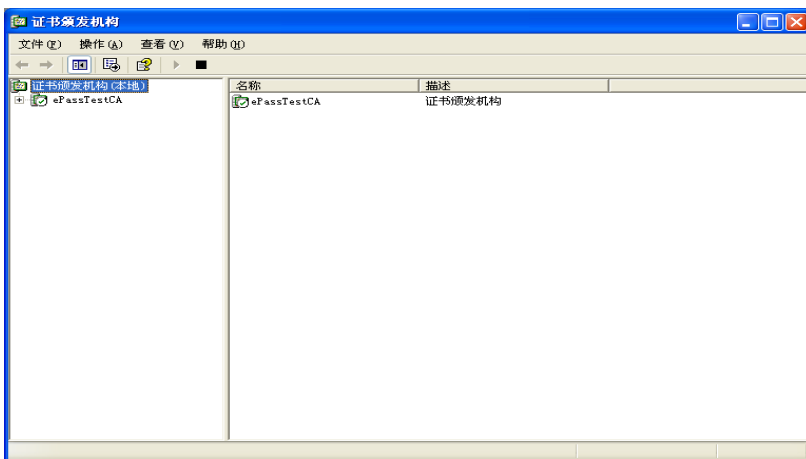


图 4-10 证书授权系统管理工具

4.1.2 安装根证书

若要开始向该证书颁发机构要求证书前，必须先安装该证书颁发机构的标识证书，这时候，当向该证书颁发机构要求其他证书信息时，该证书颁发机构就会先检查您有无该根证书颁发机构所发行的许可证书，若有，就会提供给您发行证书的服务。

所以，若企业内部已经具有证书服务器后，用户必须先向企业内部的根证书颁发机构获取属于用户帐号的根证书，当获取了根证书后，系统才

会启动由该证书颁发机构所发行出来的所有证书（启动这些证书的有效性）。下面我们以简单的操作步骤来说明如何由证书颁发机构获取信息，来安装根证书。

1. 首先，在企业内部域上安装证书服务器。关于证书服务器的安装方式，请参考以上的说明。
2. 启动Internet Explorer，并连接上企业内部的证书服务器。（例如 <http://企业提供根证书颁发机构的Windows Server 2003 计算机的DNS名称/certsrv>，例如假设在delltest这部Windows Server 2003 计算机上安装根证书颁发机构时，用户就可以用 <http://delltest/certsrv> 网址）。接下来，就可以直接进入到了证书颁发机构的证书发行网页，如图 4-11 所示。

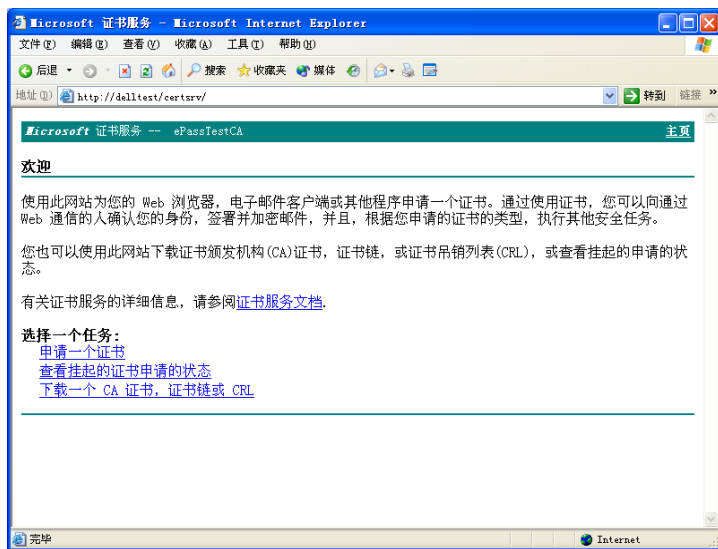


图 4-11 证书授权网页

因为现在需要先获取此部证书颁发机构的根证书，因此，请选择“检索 CA 证书或证书吊销列表”的选项，

3. 系统会呈现此证书颁发机构的安装证书或下载证书的选项网页，如图 4-12 所示。

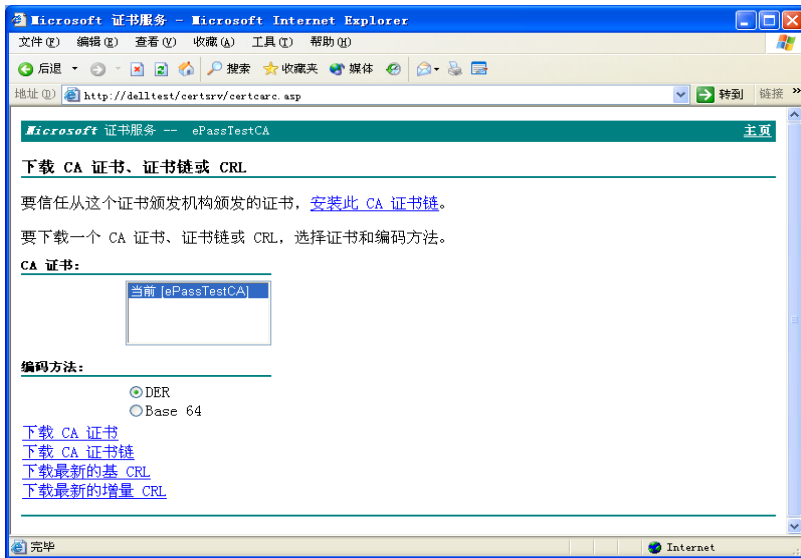


图 4-12 证书颁发机构的证书下载安装窗口

用户可以直接选择“安装此 CA 证书链”的链接，当按下此链接后，系统会自动将该证书颁发机构的证书链(证书信任关系)安装到您的 Windows Server 2003 计算机上，这时候，用户的 Windows Server 2003 计算机就可以使用该证书颁发机构所发行的证书，来完成身份验证或其他安全性的处理(参见图 4-13)。

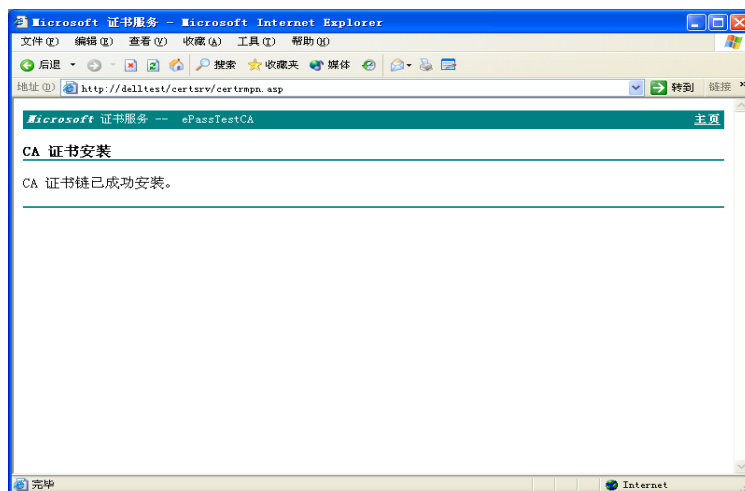


图 4-13 自动安装 CA 证书链

除了选择上述的“安装此 CA 证书链”的链接方法以外，用户还可以选择下方的“下载 CA 证书”的链接，以手动的方式直接获取证书颁发机构所发出的证书信息。可以采用 DER 编码的方式、或者以 Base 64 编码的方式，让证书颁发机构以这两种数据编码的方式将该证书颁发机构的证书信息打包成证书导出文件的形式，用户直接通过 Internet Explorer 下载该证书颁发机构的代表证书、或者相关的数据（包含下载证书路径（也就是证书授权信任的关系）、以及证书吊销列表）。

4. 选择编码形式后，直接按下“下载 CA 证书”的链接，这时候系统就会以用户所选择的证书编码形式，将该 CA 的代表证书下载到用户所使用的 Windows 计算机上，如图 4-14 所示。

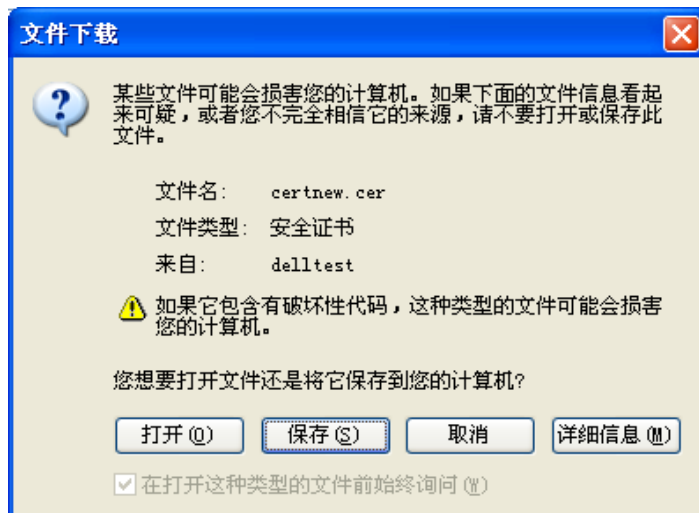


图 4-14 下载 CA 证书

- 若要查看此证书的话，可以选择“打开”按钮。这时候，系统便会立即打开这个 certnew.cer(也就是该证书颁发机构发行给用户的证书导出文件)，如图 4-15 所示。

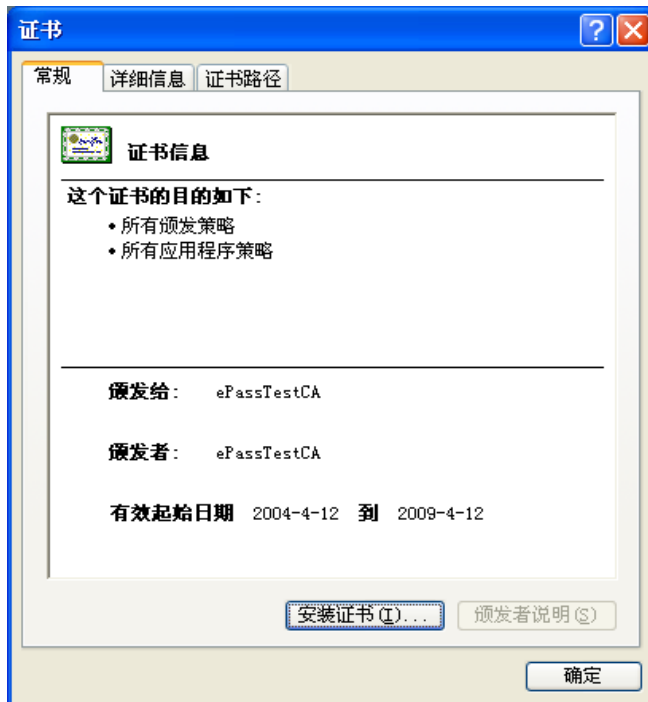


图 4-15 打开证书

6. 用户可以查看此证书的相关信息，若确定无误后，可以按下“常规”页面下方的“安装证书”按钮，以便将此证书安装到用户的作业环境上。当按下“安装证书”按钮后，系统会启动证书导入向导。因为当使用 Internet Explorer 自证书颁发机构下载该 CA 的代表证书时，该 CA 是以用户所选择的证书导出的文件格式 (DER 编码或者 Base 64 编码) 来打包此证书信息的，因此，必须通过 Windows 操作系统的证书导入向导，才能将该证书信息顺利安装到您的系统环境上。

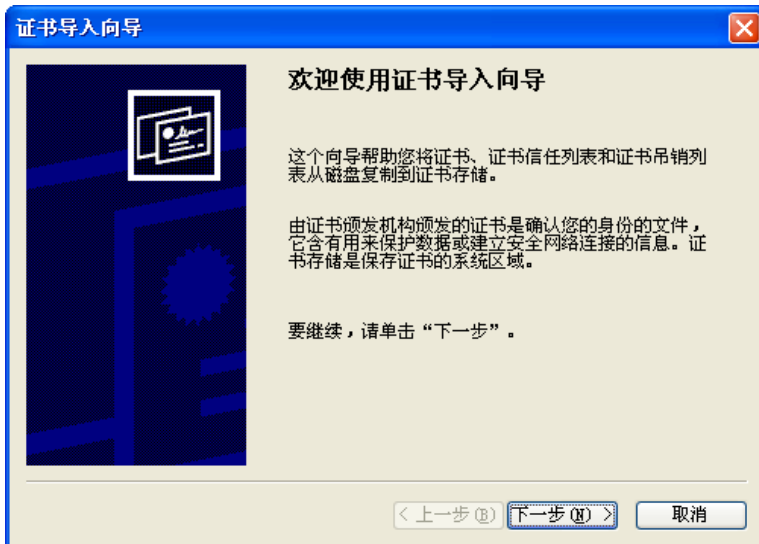


图 4-16 证书导入向导

用户只需要按照证书导入向导的提示步骤，依序进行操作，即可将证书顺利安装到用户计算机的运行环境上。

以上是下载 CA 代表证书的操作方式。也可以采用同样的方法来下载 CA 证书链导出文件或者该 CA 的基证书吊销列表和增量证书吊销列表的导出文件。若要下载 CA 证书链，只需要按下“下载 CA 证书链”的链接即可（参见图 4-12）。

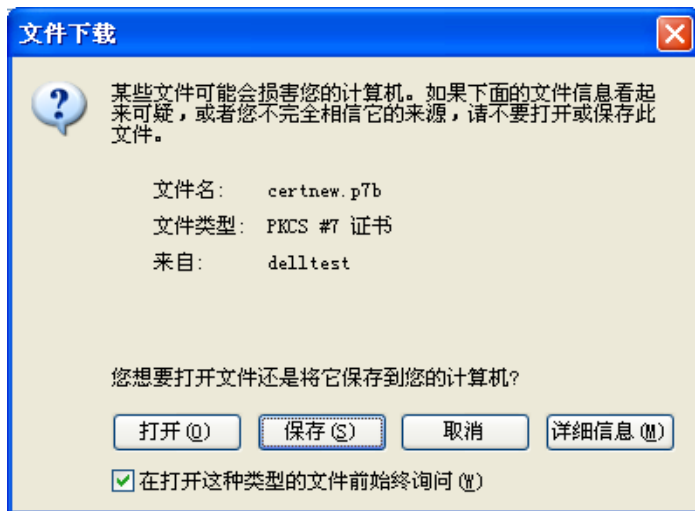


图 4-17 下载 CA 证书链

当按下“下载 CA 证书链”的链接后，系统会接着出现如图 4-17 的文件下载窗口。用户可以选择将此文件先存储在您的磁盘驱动器内，如图 4-18 所示。

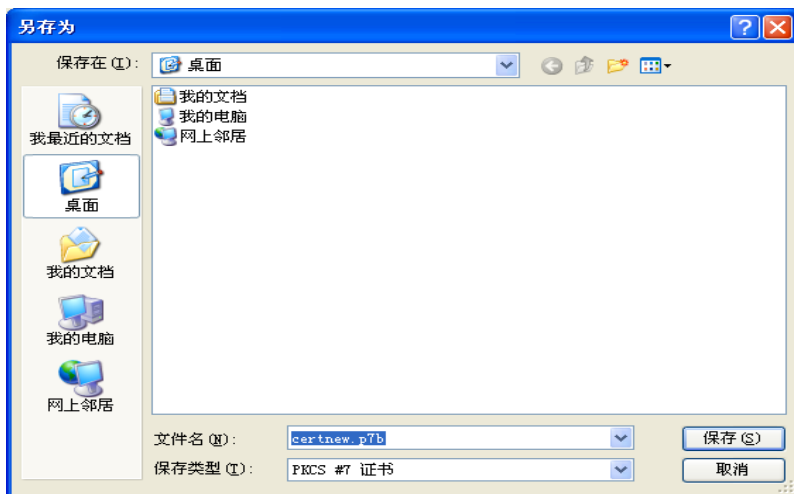


图 4-18 存储 CA 证书路径文件

当下载完成后，可以以手动的方式启动证书导入向导，将 CA 证书链

的相关信息导入到您的系统里。也可以采用同样的方式来下载该证书颁发机构的最新基证书吊销列表或增量证书吊销列表，用户只需要按下“下载最新的基 CRL”或“下载最新的增量 CRL”链接，即可打开文件下载列表，如图 4-19 所示。

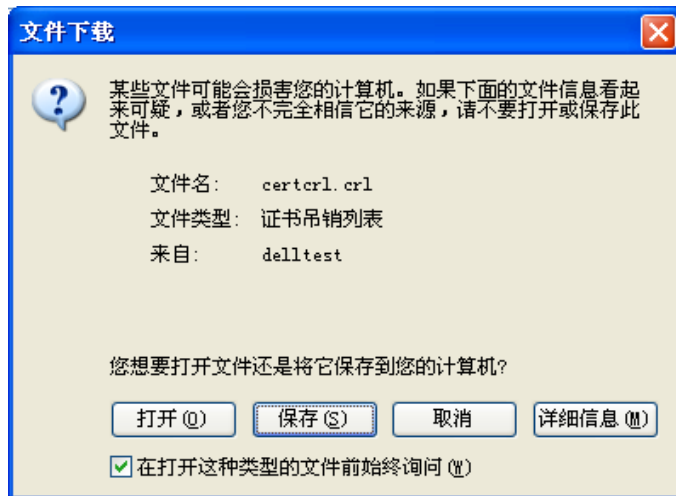


图 4-19 下载证书吊销列表

同样的，若要使用此证书吊销列表的话，可以在下载的文件上按下鼠标右键，并选择“安装 CRL”的选项。这时候，系统会启动证书导入向导，用户只需要按照向导的提示操作步骤，依序进行设置即可。

4.2 配置 SSL 加密站点

IIS 是 Microsoft Internet 信息服务器的简称 (Microsoft Internet Information Service)。IIS 为 Windows Server 2003 操作系统的一个服务之一，IIS 主要提供了 WWW、FTP、Gopher、以及其他国际互联网上的重要服务的主要服务器的功能。一般来说，在安装 Windows Server 2003 操作系统时，Windows Server 2003 操作系统的安装程序默认不会将 IIS 的相关组件安装到计算机上。不过也可以在安装 Windows Server 2003 操

作系统时，将 IIS 的组件安装的过程加上。假设目前还未安装 IIS 组件的话，用户可以由“配置服务器向导”。如图 4-20 所示。

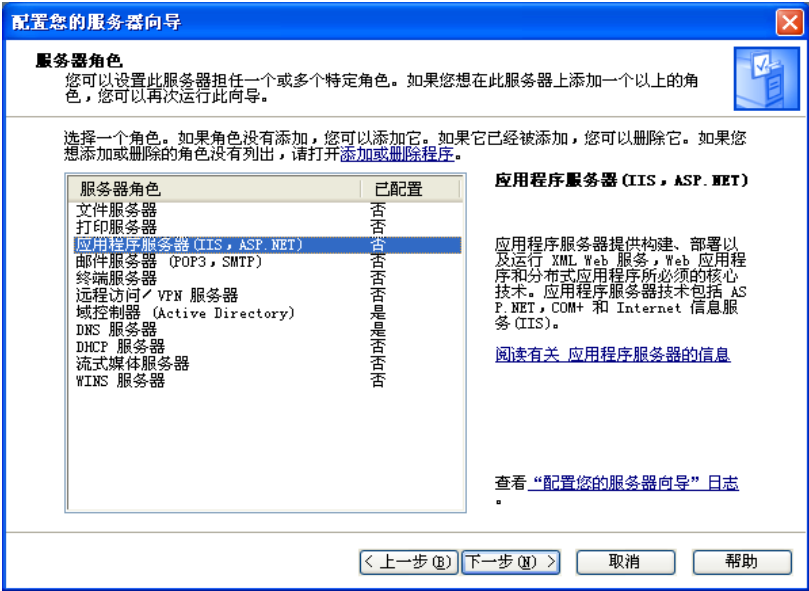


图 4-20 安装 IIS

假设用户已经在 Windows Server 2003 计算机上安装了 IIS, 而且目前 IIS 已经开始启动运行了。用户可以由“控制面板”→“管理工具”→“Internet 服务管理器”选项，来启动 IIS 服务管理工具，如图 4-21 所示。

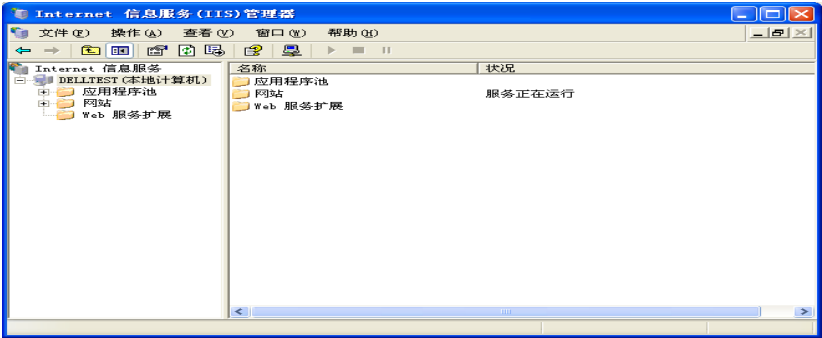


图 4-21 IIS 管理界面

因为 Windows Server 2003 上默认是 asp 服务是没有启动的，如图 4-22。

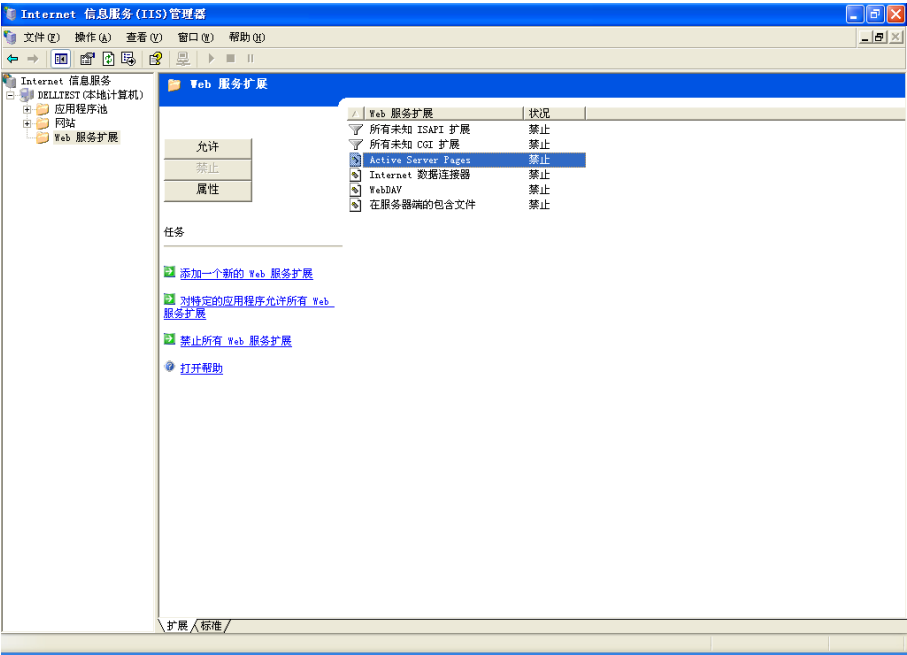


图 4-22 禁止 ASP 界面

选择 Active Server Page 后选择启动按钮来启动 asp 支持，启动后的界面如图 4-23



图 4-23 启动 ASP 界面

我们的目的是要设置 IIS 系统, 让用户 IIS 系统内的 Web 站点能够具有使用 SSL 安全性协议的能力及通 Internet Explorer 来申请证书。要设置 Web Server 的 SSL 使用能力, 必须打开 IIS 的主要目录安全对话框。要打开主要的 IIS 目录安全对话框, 请按照下面的过程进行操作:

1. 以系统管理员权限的账户登录 Windows Server 2003 计算机。
2. 依序打开“控制面板”→“管理工具”→“Internet 服务管理器”选项, 来启动 IIS 服务管理工具。
3. 展开 Internet 信息服务 (IIS) 管理器, 并在网站下的默认网站上点右键选择“属性”选项, 系统会打开“网站属性”设置窗口, 接着, 请选择“目录安全性”页面。如图 4-24 所示。



图 4-24 目录安全性设置

接着，请勾选“安全通信”部分的“启用 Windows 目录服务匹配器”的复选框选项。

如果在这里勾选“启用 Windows 目录服务匹配器”的选项，那么 IIS 将会要求 Active Directory 域控制器来负责处理证书与帐号的映射关系。请注意，只有在 IIS 主要属性里，才可以设置此选项。

如果使用 Windows 2003 Active Directory 域控制器的映射方式，用户就可以使用由在企业内部的证书颁发机构所发给的登录证书来连接上企业的 Web 站点。因为根据默认的状态，Windows Server 2003 会自动完成一对一的证书与用户账户的映射关系，所以用户目前就可以采用此映射关系来连接 Web 网站。

6. 我们将来看看如何设置一个单一的 Web 站点的安全功能。若用户不希望使用到 Windows 2003 Active Directory 域的映射功能（也就是用户在前一个操作步骤里没有勾选“启用 Windows 目录服务匹配器”的复选框选项），直接跳到这一小节来操作即可。

7. 在 IIS 里，可以同时设置管理多个 Internet 信息服务器（包括多部的 WWW Server、多部的 FTP Server、或是其他的国际互联网上的信息服务器），前面所说明的部分是针对整个 IIS 的安全性控管的设置（称为主要 IIS 目录安全设置），接下来，我们便要说明如何针对 IIS 内部的一个站点做安全性的设置与管理。
8. 接着，请再回到控制台，请在您想设置的 Internet 服务节点上（例如默认的 Web 站点），按下鼠标右键，并选择“属性”选项。

系统会打开该 internet 信息服务的属性设置窗口，请选择“目录安全性”的页面，如图 4-25 所示。



图 4-25 目录安全性页面

当要开始启用 IIS 功能时，必须先获取 Web 服务器证书，以提供基础的证书身份验证服务。

请读者注意到“安全通信”的部分，若用户还未获取并安装 Web 服务器证书，这时候“查看证书”按钮为不可用的状态。用户必须先安装服务器证书，才能继续设置安全通信的属性。要安装服务器

使用的证书，请按“服务器证书”按钮。

9. 当按下“服务器证书”按钮后，接着会出现 Web 服务器证书向导，指导用户进行服务器证书的安装过程，如图 4-26 所示。

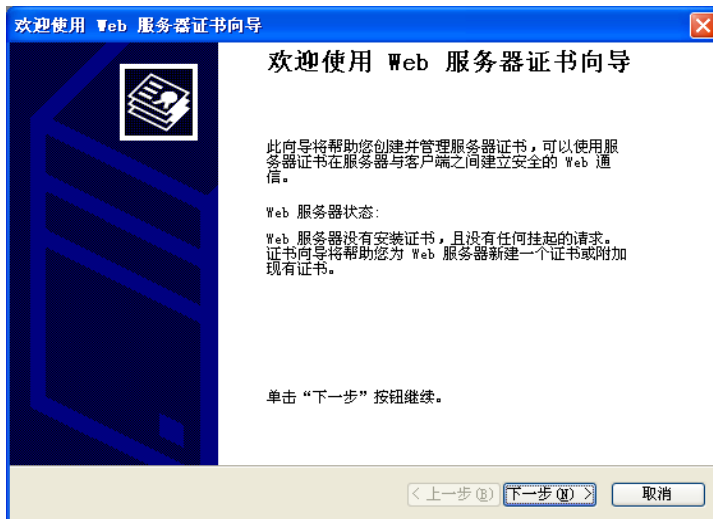


图 4-26 Web 服务器证书向导

10. 继续按“下一步”按钮，进行下一个步骤的服务器证书安装设置过程。接下来，系统会要求用户选择指定服务器证书的来源方式，如果尚未安装过服务器证书，这时候用户必须选择“新建证书”选项。若之前已获取过 Web 服务器证书，而且想要重新利用这些已有的证书，请选择“分配现有的证书”、“从密钥管理器备份文件导入证书”、“从 .pfx 文件中导入证书”、或者“将远程服务器站点的证书复制或移动到此站点”选项，将原有的 Web 服务器证书安装到 IIS 系统上。

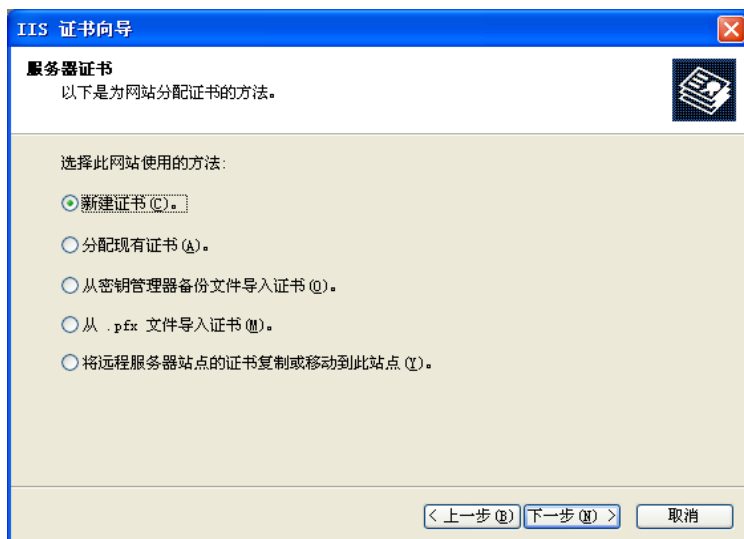


图 4-27 选择指定服务器证书的来源方式

如图 4-27 所示。以下的步骤假设用户选择“新建证书”选项。

11. 当您设置好上一个设置步骤后，请继续按“下一步”按钮进行下一个步骤的服务器证书安装设置过程。系统会要求您选择证书要求的时机，您可以按照您的需要来选择是否要先准备好证书要求，稍后再将此证书要求发送到证书颁发机构上，以获取适当的证书信息；或者立即将证书要求传递到您在稍后指定的证书颁发机构上，立即向证书颁发机构要求获取证书信息。

在这个步骤里，可以选择在线上直接连接证书颁发机构，直接获取证书信息（“立即发送一个请求到一个在线证书颁发机构”选项）；或是将证书要求储存成文件（选择“现在准备请求，但稍后发送”选项）再将此证书要求的文件发送到证书颁发机构上，以获取需要的证书。

12. 假设用户目前需要由企业外部商用性质的证书颁发机构获取所需要的证书的话，那用户可能需要使用文件方式的证书要求方式，产生要求证书的文件（一般是提供给该部商用证书颁发机构处理身份验证过程

使用的信息)，并由该部商用证书颁发机构确认核对后，再发行出给用户证书，这时候，用户就可以获取需要的证书。一般来说，联机获取的证书颁发机构通常会是本地的证书颁发机构，以及企业内部（域内）的证书颁发机构。

若证书服务器（证书颁发机构）目前处理证书的数量不是很多，或者需要立即操作 IIS 系统的 SSL 安全通信协议的设置时，用户可以选择“立即发送请求到一个在线证书颁发机构”选项，以便立即将稍后所设置的证书要求信息传递到适当的证书颁发机构上，以便获取适当的证书信息。在这里，我们选择“现在准备请求，但稍后发送”。

当设置好这一个设置步骤后，请继续按“下一步”按钮，进行下一个步骤的服务器证书安装设置过程。

13. 接下来，系统会出现“名称和安全性设置”的设置窗口。这时候系统会要求用户设置此证书的名称以及此证书安全设置项目。此时需要为请获取的服务器证书定义一个易于标识的证书名称，并设置此证书要使用的密钥长度。根据应用的需要，设置适当的密钥长度。并注意，若密钥长度设置太短，可能导致安全性的降低；若密钥长度设置太长，可能导致系统运算处理时间过长，导致系统效率不佳、或者软硬件系统无法配合等现象。一般来说大约 1024~2048 Bits 会是比较好的选择。

如图 4-28 所示。

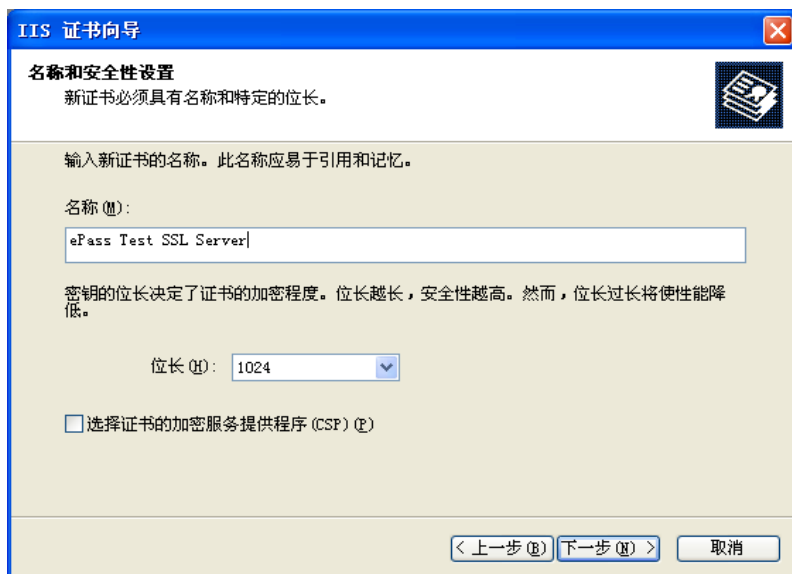


图 4-28 命名及安全设置

当完成此设置步骤后，按“下一步”按钮，进行下一个步骤的服务器证书安装设置过程。

14. 接下来，用户需要输入企业组织的一些相关信息，以便让系统将企业以及目前所处的单位等相关信息记录在想获取的证书信息内。如图 4-29 所示。输入完毕后，按“下一步”按钮继续下一个步骤的设置过程。

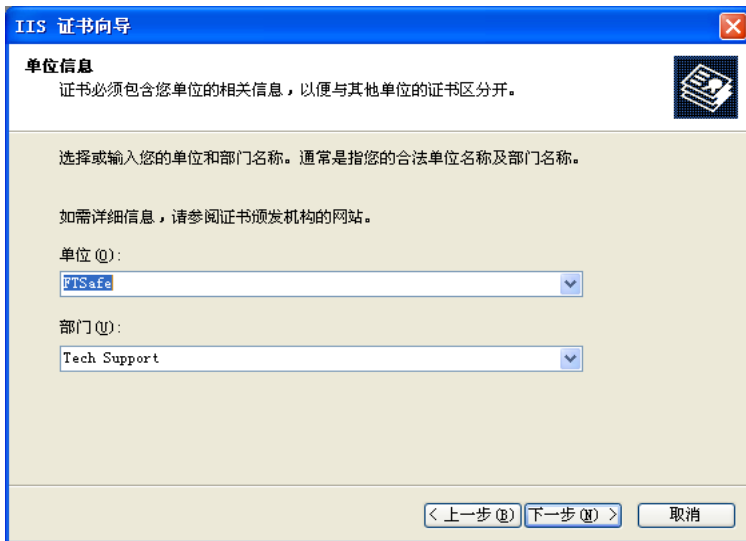


图 4-29 组织信息设置

15. 命名安装服务器证书的国际互联网服务器的标识公用名称。输入提供此 Web 服务器的 Windows Server 2003 计算机的完整资格名称 (也就是 DNS 名称)。若服务器是在企业内部运行的网络 (intranet)，用户可以输入提供此 Web 服务器的 Windows Server 2003 计算机的 NetBIOS 名称，如图 4-30 所示。

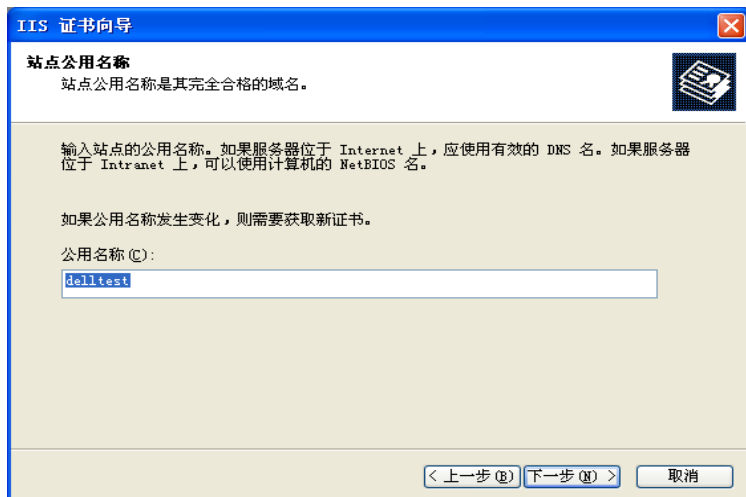


图 4-30 站点公用名称

当设置好这一个设置步骤后，继续按“下一步”按钮，进行下一个步骤的服务器证书安装设置过程。

16. 填入目前此 Web 服务器所在的地理位置信息，以便提供证书信息更详细的数据丰富性。如图 4-31 所示。



图 4-31 地理信息

当完成这一个设置步骤后，继续按“下一步”按钮，进行下一步步骤的服务器证书安装设置过程。

17. 当屏幕上出现“证书请求文件名”的设置窗口，用户可以在这里设置证书请求文件的文件名，并为其选择安装路径，如图 4-32 所示。

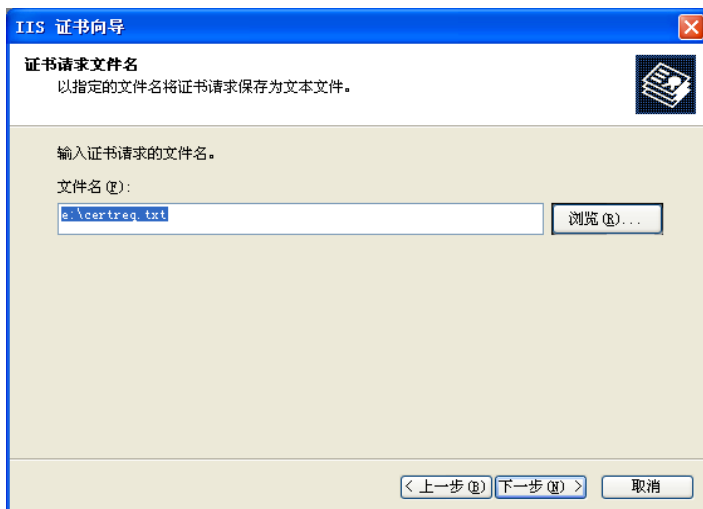


图 4-32 将证书请求存储成文件

18. 当设置完成后，系统会显示刚刚所设置的证书申请条件，用户可以检查看是否有错误，若无错误，可以继续按“下一步”按钮，进行下一个步骤的服务器证书安装设置过程。如图 3-33 所示。



图 4-33 请求文件摘要

19. 按“完成”按钮，这时计算机已经把证书请求文件存储下来了。现在，

就可以去证书颁发机构去获取证书了。

20. 打开 Internet Explorer 浏览器，连接证书服务器（这里以上面刚刚搭建的 CA 为例），进入证书颁发网页，选择“申请一个证书”选项，如图 4-34 所示。继续证书申请的下一个步骤。

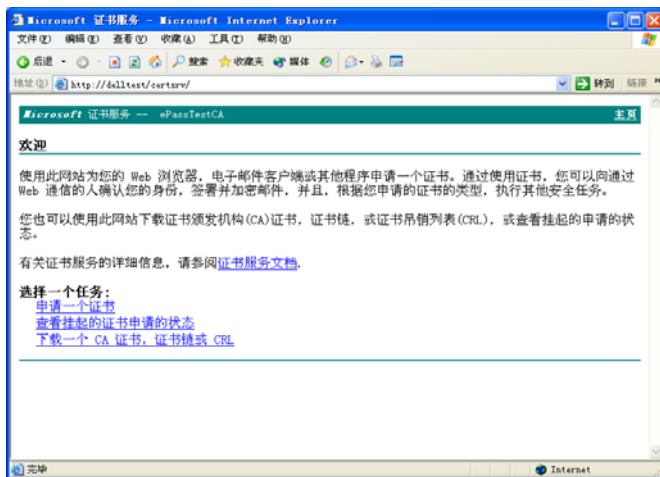


图 4-34 由 IE 获取证书

21. 接下来，进入选择证书申请类型页面，在这里我们选择“高级证书申请”选项，继续下一个过程。如图 4-35



图 4-35 证书高级申请

22. 如上图所示，这里要选择文件形式的证书获取方式，即利用刚刚得到的证书请求文件来申请证书。
23. 进入如图 4-36 所示的界面，用户需要将存储起来的证书请求文件的内容拷贝到“存储的请求文件”一栏中。然后按“提交”按钮。

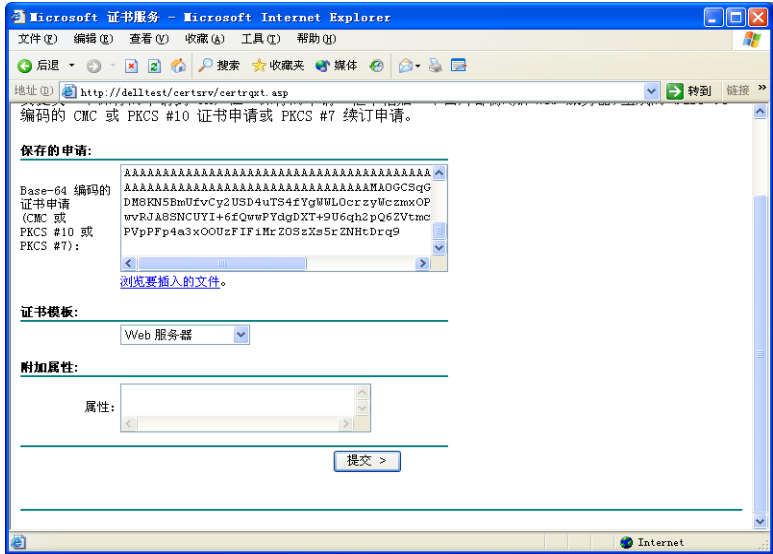


图 4-36 提供证书请求文件

24. 提交完证书请求文件后，会进入图 4-37 所示的页面中。虽然在前面安装证书颁发机构时选择的是企业根类型的证书颁发机构，但设置了不在线发放，所以，这里可以看到，请求的证书被挂起，要等待颁发机构确认身份并发行证书后才能去领取。



图 4-37 证书挂起

25. 等待证书颁发机构确认身份并通知用户去领取证书后，用户就可以再次进入颁发机构去领取证书了。打开颁发证书页面，选择检查挂起的证书选项。如图 4-38 所示。

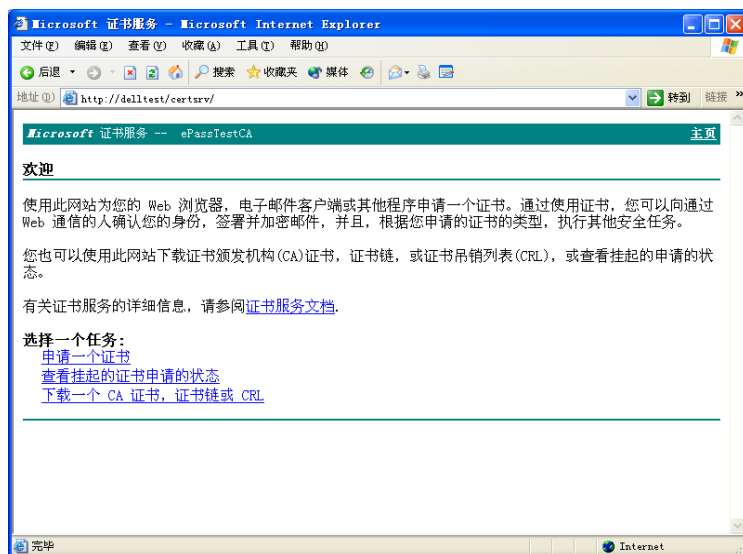


图 4-38 用 IE 获取被挂起的证书

接下来，继续领取证书操作的下一个步骤。

26. 选中与申请日期一致的证书申请请求，去领取证书，如图 4-39 所示。

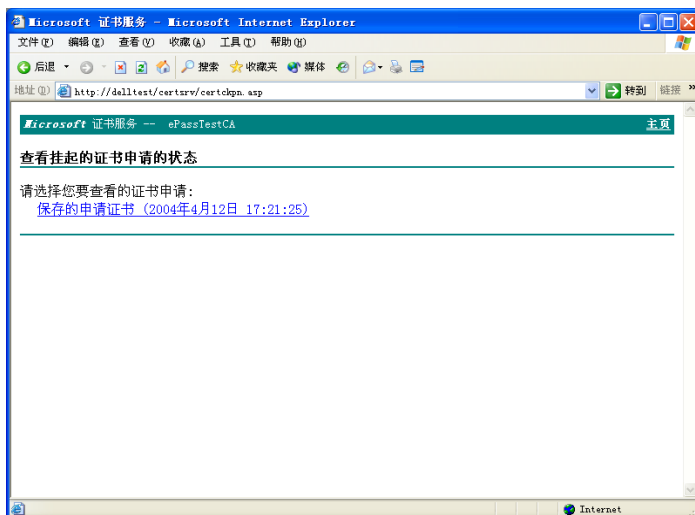


图 4-39 检查挂起的证书请求

27. 这时能看到，用户所申请的证书已经发行了，如图 4-40 所示。单击“下载 CA 证书”，就开始了证书下载过程。

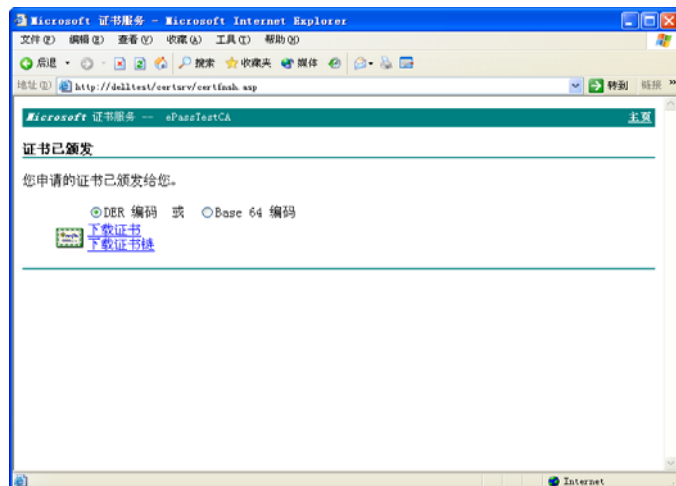


图 4-40 证书下载

28. 完成了证书下载，用户还必须启动证书安装向导来把证书安装在服务器上。有关如何打开证书安装向导请参照第 8、9 步骤。完成证书导入如图 4-41 所示。

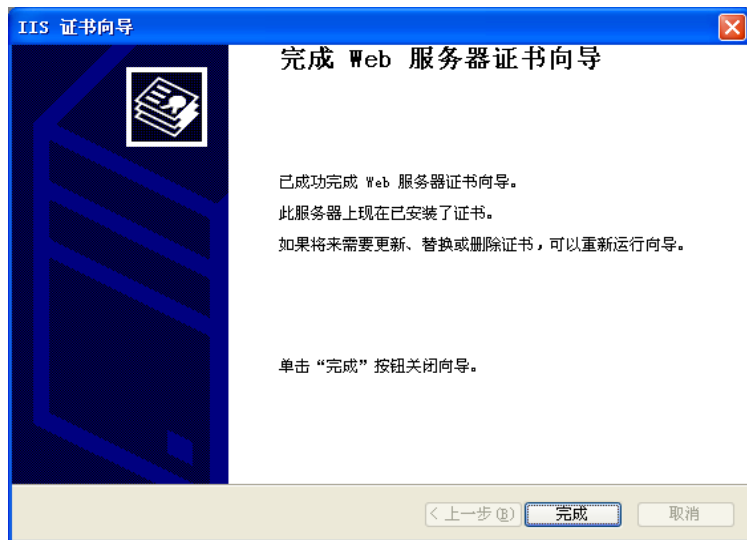


图 4-41 完成服务器证书导入工

如果采用直接链接上证书颁发机构的方式来获取证书，那么这时候向导会向用户所指定的证书颁发机构发出一个获取证书的要求信息，当该证书颁发机构身份验证通过时，就会发给用户一个证书，此证书会自动安装在用户 Web 服务器上。

在安装了服务器的证书后，接下来，用户就可以回到原来所打开的 Internet 服务器站点（Web 站点）的属性设置窗口上，这时 SSL Port 变为可填写状态。这里用户要为该 Web 站点填写一个安全通道端口（SSL Port），推荐填写默认值 443。如图 4-42 所示。

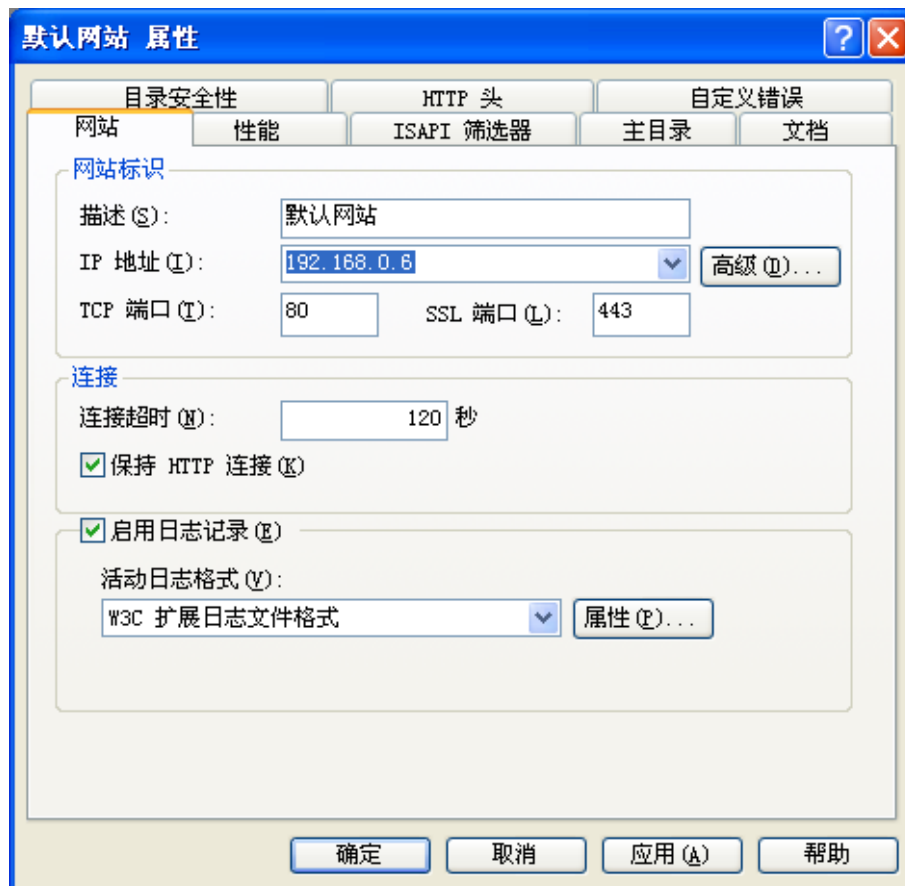


图 4-42 填写 SSL Port

现在展开“目录安全性”页面，用户可以看到在“安全通信”部分里的“查看证书”按钮已经呈现启用状态了，表示这时候就可以开始设置该国际互联网服务器的安全性协议使用设置了。如图 4-43 所示。



图 4-43 安装服务器证书后的服务器属性设置窗口

接下来，用户就可以开始进行此 Web 服务器使用 SSL 安全性协议的设置处理了。要设置此 Web 服务器使用的安全性协议功能的操作时，按照下列的过程进行设置：

1. 回到该 Internet 信息服务器的属性设置窗口，并选择“目录安全性”页面，如图 4-43 所示的画面。
2. 这时候，按下在“安全通信”部分里的“编辑”按钮，来进行该 Web 服务器的安全设置。当按下“编辑”按钮后，会出现安全通信编辑窗口。
3. 因为我们的目的是要完成设置安全 Web 站点，因此，勾选位于窗口上方的“要求安全通道（SSL）”的复选框。在客户证书中选择“要求客户端证书”选项，如图 4-44 所示。以下是关于这些选项的说明。

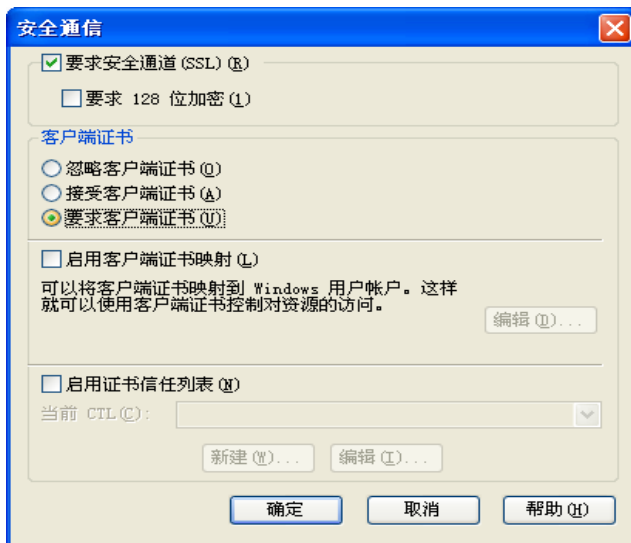


图 4-44 设置安全通信页面

- 要求安全通道（SSL）：一般来说，若没有启动此选项的话，Web 服务器默认都会以 HTTP 的通讯协议来提供 WWW 服务。但若启动了此选项后，IIS 系统就会强迫 WWW 客户端浏览器使用 SSL 的通讯协议（采用 SSL 安全协议）来使用 WWW 的服务。也就是当启用此选项后，系统就会关闭使用 http: 的连接，仅能使用 https: 连接来接上 Web 服务器（当服务器证书已经安装在您的国际互联网服务器上时，用户服务器就允许接受 https: 协议方式的联机了，若将该国际互联网服务器上的服务器证书删除，那么就无法使用 https 的方式来进行联机）。

换句话说，若勾选了这个选项，便是强迫终端用户一定要使用 SSL 的安全协议与服务器建立连接，以确保安全。

- 要求客户端证书：用户必须提供一个证书才能够获得访问权限，这种方式具有较高的安全。

当设置完成后，单击“确定”按钮。

这时，已经完成了安全 Web 站点的设置工作，并已经启用了安全通道，如果再通过 http:连接来连接该 Web 站点，会出现如下图所示的情况：

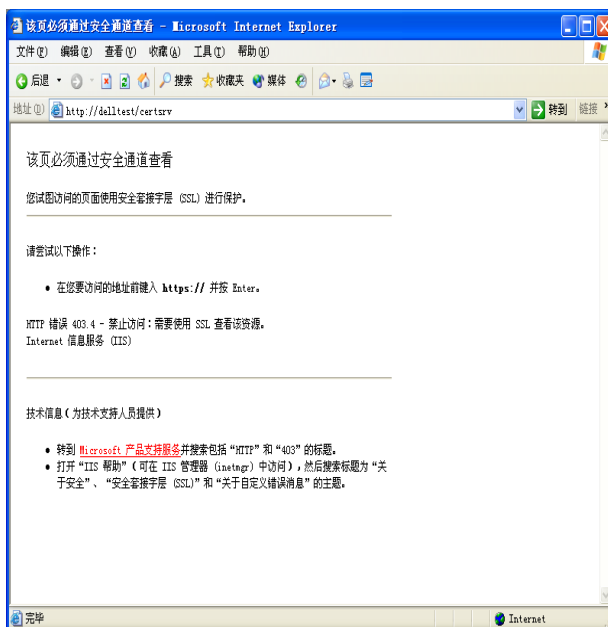


图 4-45 用 http:访问安全站点

系统提示必须要通过 https:连接来连接上要访问的站点。用户再通过 https:连接来连接上刚刚设置的安全 Web 站点。会看到系统有如下图所示的安全提示。

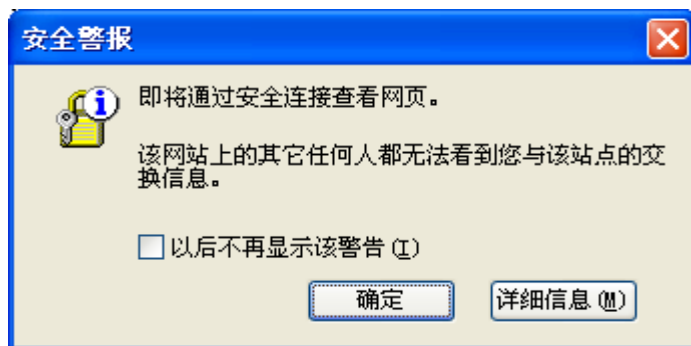


图 4-46 安全提示信息

单击“确定”按钮后，会有客户认证提示，要求选择用户要使用的证书。此时用户还没有申请客户认证证书，所以证书列表框为空。



图 4-47 客户证书选择

下面，我们就介绍如何利用 ePass2000 进行客户证书的申请。

4.3 使用 ePass2000 进行客户证书申请

确认插入了一支已经完成初始化的 ePass2000。然后通过 Internet Explorer 打开证书颁发机构的网页，选择申请证书。如图 4-48 所示。

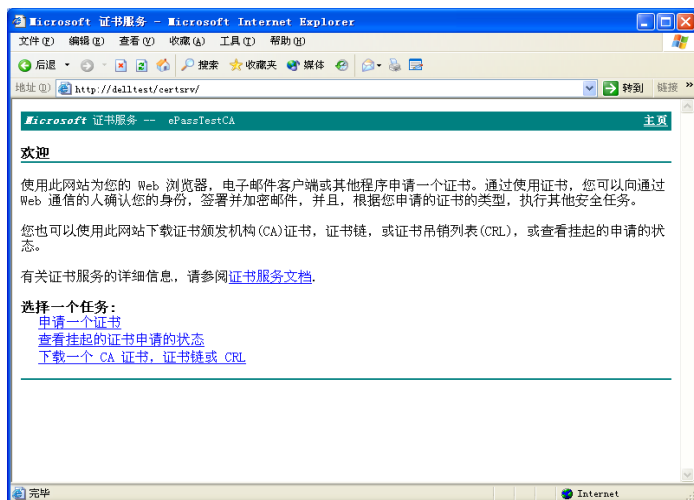


图 4-48 申请用户证书

选择申请证书的类型。选择“高级证书申请”选项，继续申请过程。在证书模板中选择“用户证书”或其它包含客户端验证的模板，在“CSP”（加密服务提供程序）选项中选择“FTsafe ePass2000 RSA Cryptographic Service Provider”。如图 4-49 所示。



图 4-49 用户信息

完成上述设置后，单击“提交”按钮，这时会看到选择 token 的提示

框，并且 ePass2000 已经被列入其中了。系统提示输入用户 PIN 码。如图 4-50 所示。



图 4-50 选择 token 并输入用户 PIN 码

输入正确的用户 PIN 码后，单击“登录”按钮。稍候，会看到证书挂起页面，需要等待颁发机构验证身份并颁发证书。如图 4-51 所示。



图 4-51 证书挂起

直到收到证书颁发机构的通知后，用户就可以去领取证书了，领取方法同第 24、25 步，只是在安装证书时，系统同样会让用户选择所需的 token 并要求输入正确的用户 PIN 码，在完成这些工作之后，只要单击“登录”按钮，系统就会自动将用户证书安装到这支 ePass2000 里了。用户可以通过

过 ePass2000 管理器来查看证书是否成功导入。

4.4 使用 ePass2000 访问 SSL 加密站点

现在，我们就可以用这支 ePass2000 来访问安全 Web 站点了。

首先，确定已插入这支已经导入证书的 ePass2000，然后，用 IE 浏览器通过 https:连接 (https://delltest:443) 到要访问的 Web 站点。此时，会看到安全提示对话框，单击“确定”按钮后，出现证书列表框供用户选择。现在，可以看到，用户证书已经列在列表框里了，选中证书，单击“确定”按钮。系统又出现了选择 token 的对话框，选中装有证书的这支 ePass2000 并输入正确的用户 PIN 码，单击“登录”按钮，现在，我们就能够看到这个安全 Web 站点的内容了。如图 4-52 所示。

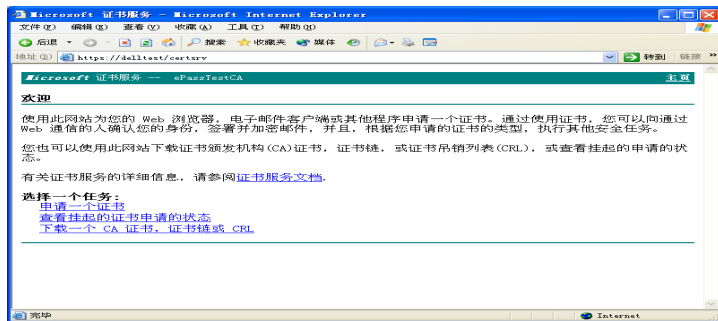


图 4-52 安全 Web 站点

4.5 使用 ePass2000 收发签名与加密邮件

在开始设置 Outlook Express 中收发签名与加密邮件之前，假设已经将 Outlook Express 设置好，可以连接上电子邮件服务器以及电子邮件帐号的相关设置，换句话说，用户已经可以使用 Outlook Express 以一般的方式发送/接收电子邮件。要设置 Outlook Express 的安全设置，必须先获取具有电子邮件安全处理能力的证书（在 Outlook Express 里称为“数字 ID”），当获取用户的数字标识后，用户才可以发送具有数字签名或者信息

加密的电子邮件。

4.5.1 获取数字证书

我们先进行获取数字标识的操作过程，来获取数字标识。由于电子邮件的应用是属于公开性的，因此，用户必须通过专门负责提供证书服务的企业，来获取适当的证书信息，以确保该证书的有效性。用户可以采用下列的操作步骤，连接上企业外部的证书颁发机构，并获取使用在 Outlook Express 内的证书。

1. 先以用户账户登录 Windows 系统。
2. 启动 Internet Explorer。
3. 在地址栏中输入<https://digitalid.verisign.com/>，如图 4-53

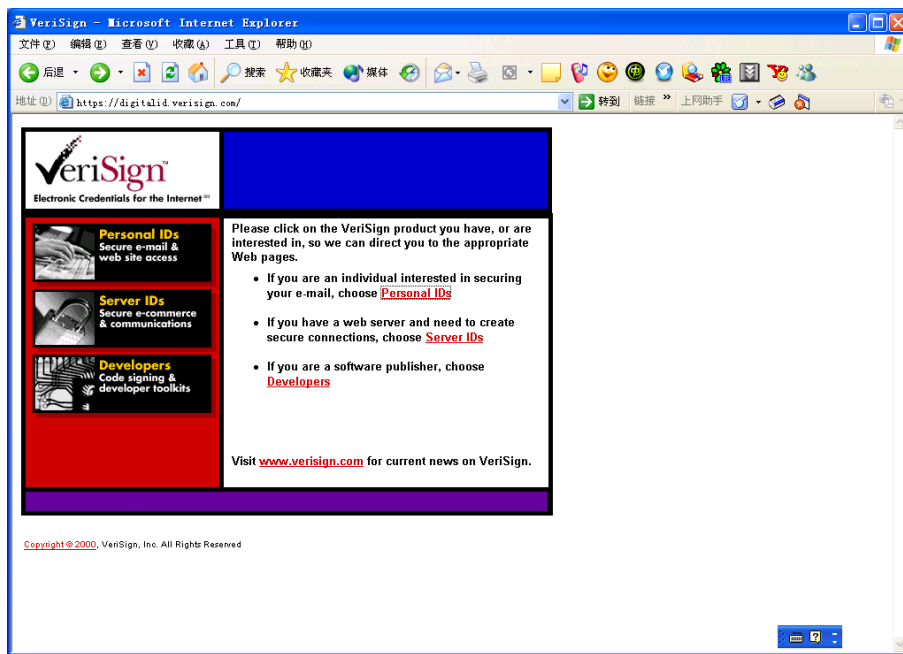


图 4-53 申请个人数字证书

4. 选择 Personal IDs 后进入图 4-54



图 4-54 申请免费数字证书

5. 选择 Free Digital ID Trail, 进入图 4-55。

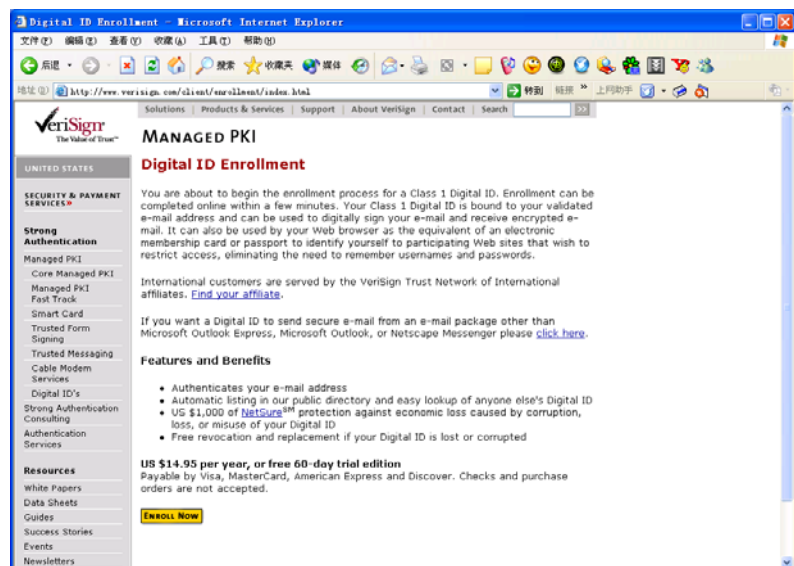


图 4-55 数字证书代理

6. 选择 Enroll Now 后进入证书申请页面图 4-56。

由于各网站所提供的这些提供证书服务的企业申请安全性证书(数字 ID)的方式各不相同,用户可以直接通过各 CA 网站的连接,连接到提供证书服务的企业,并获取专用的证书(数字 ID),然后用户才可以利用获取的数字 ID 来进行安全性邮件的一些设置。由这些提供证书服务线上登记获取数字 ID 时,若用户要求获取的证书(数字 ID)的用途是用于在安全性邮件的方面时,在登记获取数字 ID 时都会要求输入用户的 Email 帐号的地址,在这里填写的 Email 帐号即是该数字 ID 的授予对象,若用户有两个以上的 Email 帐号时,请注意填写要进行安全性邮件设置处理的 Email 帐号。

举个例子来说,假设要在 techsup@ftsafes.com 的 Email 帐号上设置使用安全性邮件的功能(在 Outlook Express 上设置),用户必须在向提供证书服务的网站填入您要获取证书(数字 ID)之签发对象的 Email 信箱位置。

以下继续由 Verisign 企业所提供的数字 ID 获取服务。

1. 在确认插入了一支已经初始化过的 ePass2000 后,我们通过 Microsoft Web 网站的连接来到 Verisign 的安全邮件申请页面(我们申请一个试用证书来说明)。来到入下图所示的界面。

Microsoft Class 1 Enrollment - Microsoft Internet Explorer

地址: <https://digitalid.verisign.com/client/class1EE.htm>

Verisign™ Class 1 Digital ID™ for Microsoft Internet Explorer

Step 1 of 4: Complete Enrollment Form

• Step 1: Complete Enrollment Form Step 3: Pick up Digital ID
Step 2: Check E-mail Step 4: Install Digital ID

Contents of Your Digital ID
Fill in all fields. Use only the English alphabet with no accented characters. This information is included in your Digital ID and is available to the public.

First Name: Nickname or middle initial allowed (example -- Jack B.)	<input type="text" value="techsup"/>
Last Name: (example -- Doe)	<input type="text" value="ftsafes"/>
Your E-mail Address: (example -- jbdoe@verisign.com)	<input type="text" value="techsup@ftsafes.com"/>

Challenge Phrase
This unique phrase protects you against unauthorized action on your Digital ID and should not be shared with anyone. Do not lose it! It is required to revoke, replace, renew or set preferences for your Digital ID.

Enter Challenge Phrase: Do not use any punctuation.	<input type="text" value="csaf.3kdr.1f"/>
---	---

图 4-56 线上登记获取数字 ID

2. 在此需要我们填写一些个人的信息资料,用户以看到由这个企业线上获取安全性电子邮件的数字 ID 时,用户必须填写 Email 信箱的位置,在这里请注意,用户填写的电子邮件位置必须是您稍后要在 Outlook Express 里设置使用安全性邮件的信箱位置。在“Cryptographic Service Provider Name”一项中,选择“FTsafe ePass2000 RSA Cryptographic Service Provider”。
3. 确定填写的一切信息无误后,请按页面最下边的“Accept”按钮,此时,会出现“选择令牌”的对话框。选定用户要安装证书的这支 ePass2000,系统会提示输入“用户密码”,如图 4-57 所示。



图 4-57 选择 token 并输入用户 PIN 码

3. 输入正确的用户 PIN 码后,单击“登录”按钮,稍候,会看到如图 4-58 所示的页面,提示用户去查看 Email 信箱。

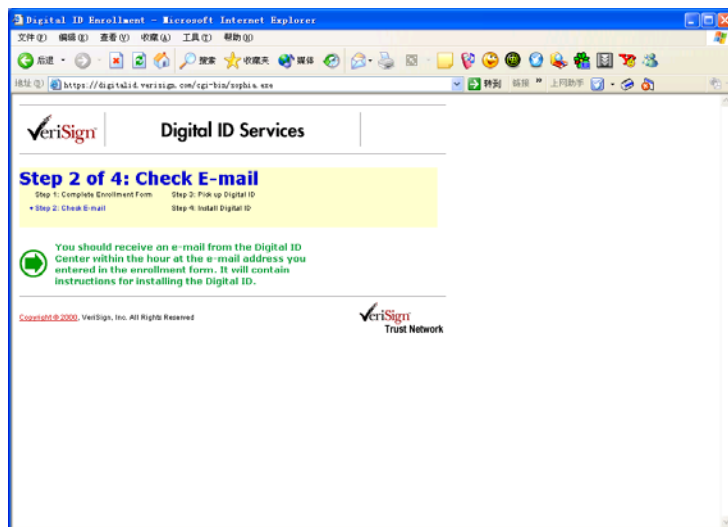


图 4-58 检查 Email 提示

4. 打开 Verisign 发的 Email, 可以看到用户提供的相关信息和一个 Internet 链接 <https://digitalid.verisign.com/enrollment/mspickup.htm>, 以及“PIN number”。在 Internet Explore 中打开这个链接, 来到“数字 ID 服务”的第三步, 如图 4-59 所示。



图 4-59 数字 ID 服务第三步

5. 将Email 中的“PIN number”填入到文本框中，然后按“Submit”按钮。来到“数字 ID 服务”的第四步—“安装数字证书”。如下图所示。

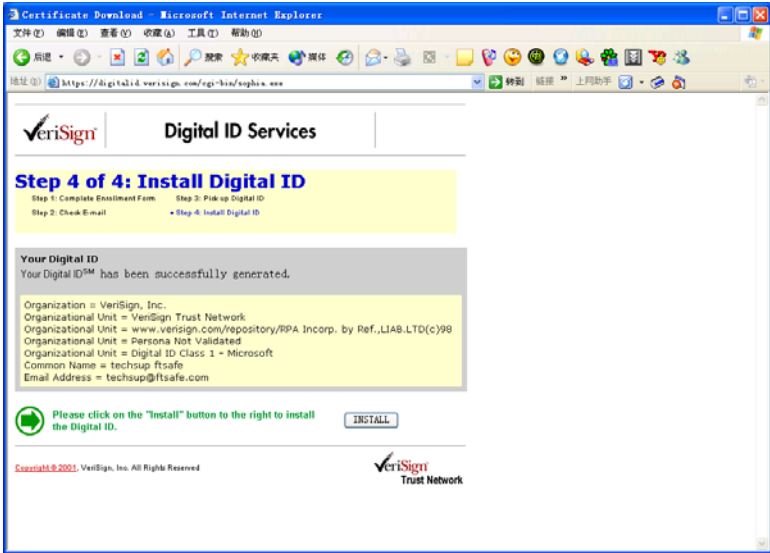


图 4-60 安装数字证书

按下“Install”按钮，此时，又会有“选择 token”对话框出现，仍然是选择要装入证书的 ePass2000，然后输入正确的用户密码，再单击“登录”按钮，稍候，Verisign 会提示证书已经成功安装。用户可以通过 ePass2000 管理工具来查看安装的证书。

以上是获取证书（数字 ID）操作过程，当获取证书（数字 ID）后，用户就可以开始设置 Outlook Express 中的 Email 帐号，让 Email 帐号能够具有安全性邮件的处理能力。

4.5.2 设置 Email 帐号的安全性

设置 Outlook Express 中的 Email 帐号中的安全性功能，按照下列的操作步骤依序进行操作：

1. 请先以用户帐户登录 Windows 系统。
2. 用户需先确定已经获取了使用在安全性邮件的数字 ID（证书）。用户可以由企业外专门提供证书服务的企业网站获取证书，也可以由 Windows 2003 证书服务器获取证书，要获取证书，请按照刚刚的说明方式，以此获取需要的数字 ID。
3. 启动 Outlook Express。
4. 接着，请由 Outlook Express 上方的选项中选择“工具”选项→“帐号”选项，如图 4-61 所示。

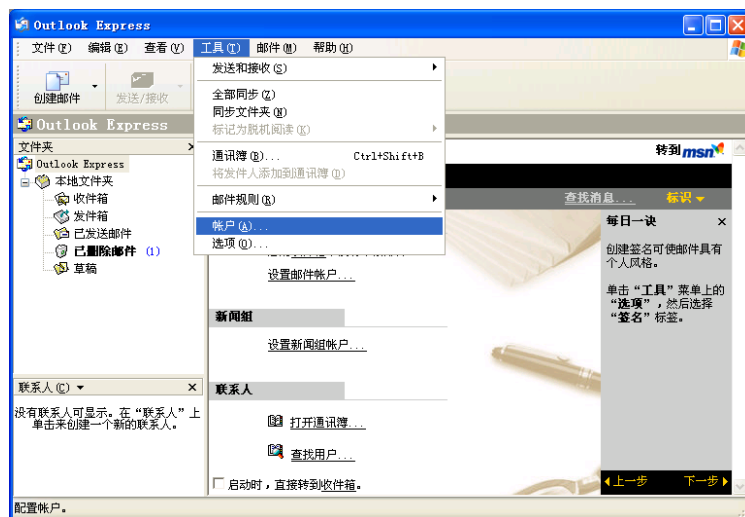


图 4-61 启动帐号设置

5. 当打开“Internet 帐号”窗口后，请点选“邮件”页面。我们假设用户已经设置好电子邮件信箱了，请选择想设置的电子邮件帐号，接着，请按下旁边的“属性”按钮。如图 4-62 所示。

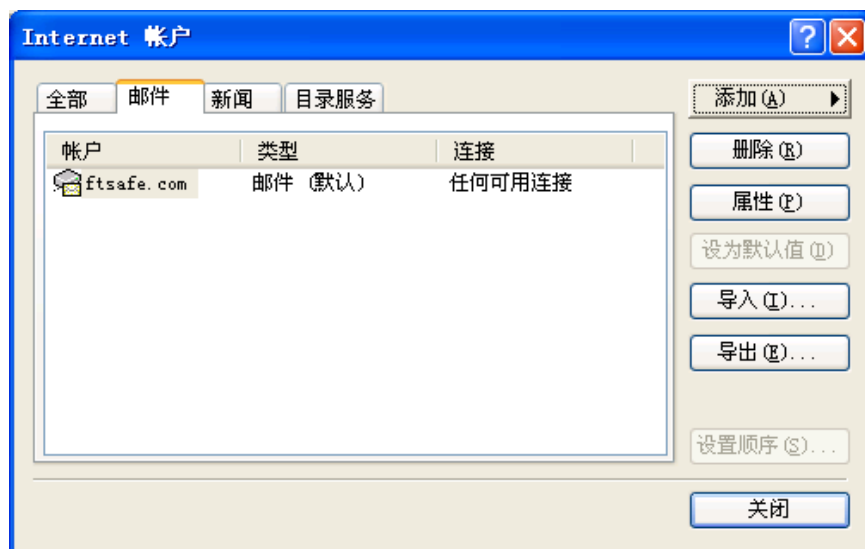


图 4-62 设置您电子邮件帐号的属性

6. 当打开此电子邮件帐号的属性设置窗口后，先选择“常规”页面（参见图 4-63），检查目前的 Email Address 是否有设置错误。



图 4-63 检查电子邮件地址的设置

7. 选择“安全”页面，以显示关于此电子邮件帐号的安全性相关设置，如图 4-64 所示。

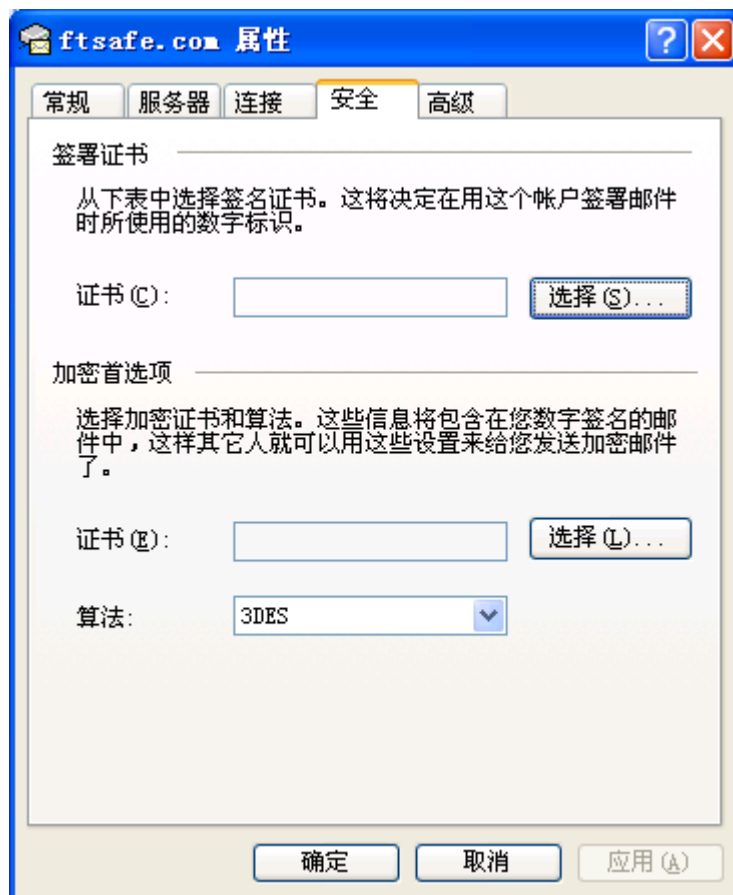


图 4-64 邮件帐号的安全设置

8. 若让此 Email 帐号能够具有数字签名的能力，在“签名标识”的部分里，按下“选择”按钮，并选择一个刚刚获取的证书（数字 ID）。若要让此 Email 帐号能够具有电子邮件加密的能力，在“加密首选项”的部分里，按下“选择”按钮，并选择一个刚刚获取的证书（数字 ID），以便让 Email 帐号具有处理电子邮件加密的功能，用户还可以选择想使用算法规则。
9. 当按下此按钮后，用户会看到下列的画面。Outlook Express 将只会使用用户信箱里所设置的证书来辨识 S/MIME 信件。此证书是记

录在 Email 信箱的证书的主题字段里的证书。这些证书都会显示在图 4-65 所示的选择窗口里，选择一个要使用的证书。

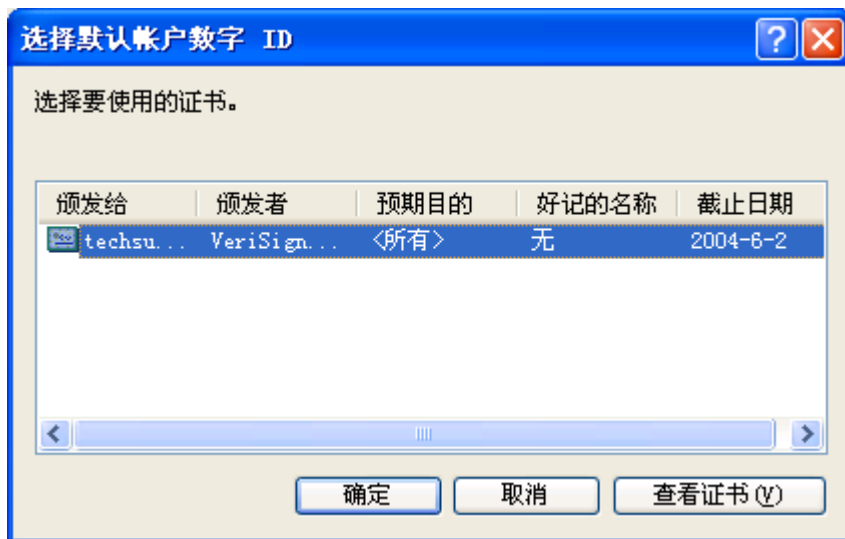


图 4-65 选择使用在 Outlook Express 的证书

用户可以按下“查看证书”来查看该证书的详细信息。

10. 按下“确定”完成设置，并回到 Outlook Express 的主界面。
11. 由下拉式菜单的“工具”选项里，选择“选项”设置。点选“安全”设置页面。这时候会显示关于安全设置的一些设置项目，如图 4-66 所示。

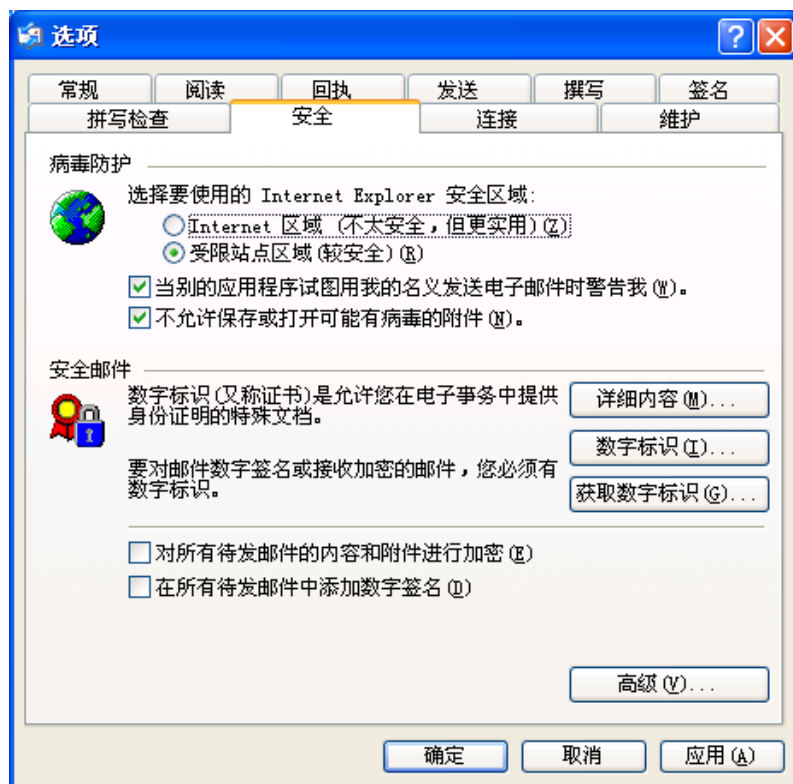


图 4-66 Outlook Express 整体安全设置

12. 如果想要让送出去的每一份电子邮件上都附加上数字签名，勾选“对所有待发邮件中添加数字签名”选项，如图 4-66 所示。用户也可以用稍后所说明的方法，在想要发送的电子邮件信息上加上数字签名。
13. 如果要将所发送出去的每一份电子邮件的属性都加密，请勾选“对所有待发邮件的内容和附件进行加密”的选项，如图 4-66 所示。用户也可以用稍后所说明的方式，对想要加密的个别信息进行内容和附加文件的加密设置。
14. 按下方的“高级”按钮，这时候会启动“高级安全设置”对话框，如图 4-67 所示。

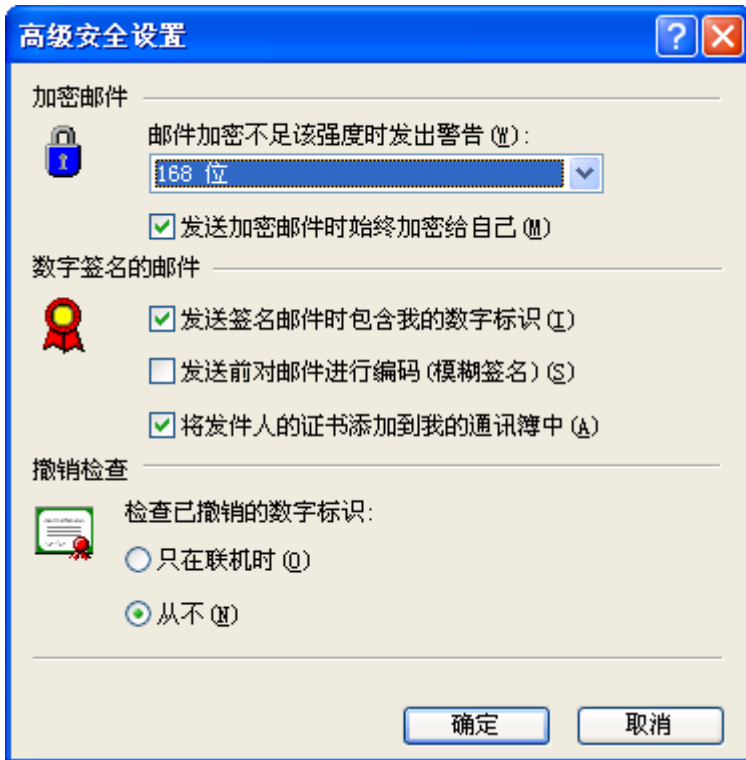


图 4-67 高级安全设置选项

15. 确定已经勾选了“发送加密邮件时始终加密给自己”的复选框选项（位于加密邮件部分下的选项）。
16. 确定也勾选了位于“数字签名的邮件”框下方的“发送签名邮件时包含我的数字标识”以及“将发件人的证书添加到我的通讯簿中”的选项。因为当发送加密型电子邮件时，或别人发送加密型电子邮件时，接受端的人都必须获取对方的密钥（存储在数字标识中）才可以读取到原始的信件内容，勾选此选项是确保能够正确获取对方解密所使用的密钥信息。
17. 另外，用户也可以根据需要，调整其它的设置，诸如密钥的长度的限制。

18. 到此时用户已经完成了 Outlook Express 的设置。当发送电子邮件信息时，邮件会自动进行加密，并加上数字签名信息。

4.5.3 使用 Outlook Express 发送附加数字签名的邮件

当设置好 Outlook Express 里的安全设置选项后，用户就可以开始发送具有安全性质的电子邮件。因为，Windows 2003 操作系统上的证书服务采用公钥的基础技术来建立的，因此，所有架构在 Windows 2003 公钥基础的许多应用程序都具有上述的安全性应用功能。在 Windows 操作系统内建提供的 Outlook Express 也提供了数字签名以及电子邮件加密的基本功能。

现在，我们就来看看如何在个别的邮件上加上证书所签上的数字签名。按照下列的步骤进行操作：

1. 以用户帐号登录 Windows 系统。
2. 启动 Outlook Express。
3. 按下 Outlook Express 上方的“新邮件”按钮，以便打开一个空的邮件写作窗口，开始编辑新的邮件信息。
4. 填上要发送的收件人地址，主题等相关字段的信息，并填写好该邮件的内容。
5. 若要在此邮件上加附数字签名信息，以证明此邮件的正确来源时，按“签名”按钮，
6. 按下“发送”按钮，将此信息发送出去。如果此信息仍然出现在发件箱里，您可以按“发送/接收”按钮，手动将邮件送到邮件服务器上。

4.5.4 获取收件人的公钥和证书

若要发送加密的电子邮件，用户必须先获取对方的公钥或者证书，再

利用对方的公钥对用户信件进行加密处理（也就是使用收件人的公钥来进行加密），这时候，只有此公钥映射的私钥（假设此私钥只有收件人持有）才能够对此加密过的信件进行解密的处理，因此，只有持有该私钥的人，才能够阅读信件属性（加密邮件）、或确认该信件的确切性（数字签名）。

要获取对方的公钥或者证书的话，必须要求电子邮件的收件人发送一封带有数字签名的信件给用户，并将带有数字签名信息的邮件中，将其内的证书（数字 ID）存储下来，这时候用户就可以保有对方的证书以及公钥的信息。

若要存储证书或公钥，请按照下列的步骤进行操作：

1. 先要求发件者以上一个小节的方式发送一份夹带有数字签名的电子邮件给您。
2. 启动 Outlook Express，接收对方送过来的电子邮件（夹带有数字签名的邮件），并打开送件人送给用户签名邮件。
3. 在送件人的“发件人”字段上，按下鼠标右键，并选择“添加到通讯簿”选项，按下“确定”按钮，将收件人）以及其公钥与证书存储到 Outlook Express 的通讯簿列表里（参见图 4-68）。这时候就完成了存储对方公钥与证书的操作过程。

4.5.5 使用 Outlook Express 发送属性加密的邮件

若要发送加密的邮件给对方时，要确定收件人已经使用上一个小节的方式获取对方的公钥或者加密证书等信息（证书包含了公钥信息）。在这里，假设收件人已经以上一个小节的方式获取对方的公钥证书并且已经存储在 Outlook Express 里的通讯簿列表里了。

要发送一封加密过的邮件，按照下列的步骤进行操作：

1. 按下 Outlook Express 上方的“新邮件”按钮，开始编辑新的邮件信息。

- 接着，在“收件者”的字段上，选择该加密邮件的收件者。注意，若 Outlook Express 通讯簿清单里的收件者有附带数字标识（证书）信息时，其通讯簿的清单上图标会有一个标志（红色的证书标志），您必须选择夹带有证书信息的收件者。如图 4-68 所示。



图 4-68 选择收件人

- 接着，填写电子邮件的主题等相关字段的信息，并填写好该邮件的内容。
- 按下“加密”按钮，要求将此邮件信息加密过，加密信息按钮图标如图 4-69 所示。

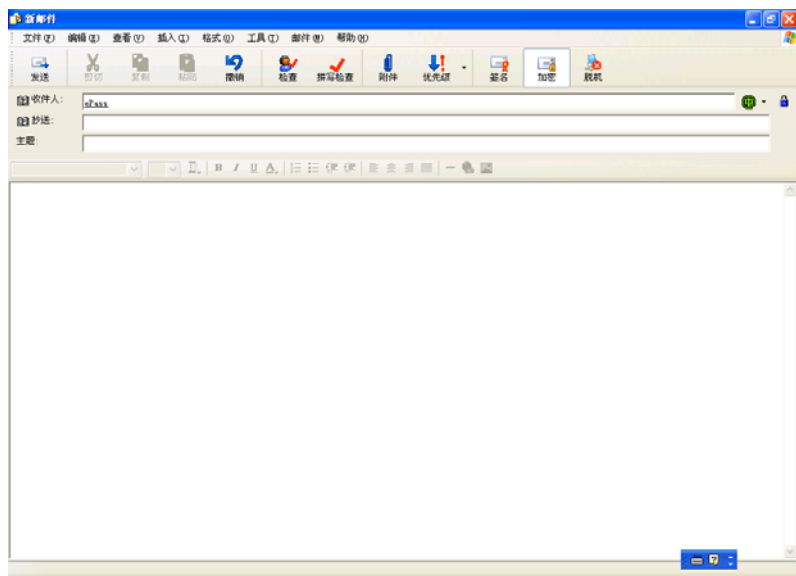


图 4-69 加密邮件

6. 按“发送”按钮，将邮件发送出去。

4.6 使用 ePass2000 进行 Windows 智能卡登录

Windows2000 操作系统以及微软后续的操作系统如 WindowsXP 都内置了对智能卡用户认证的支持，计算机用户可以选择使用传统的用户名、口令验证方式进行域用户身份验证，也可以使用智能卡来自动完成用户身份验证。智能卡用户身份验证的优势是更加安全和易于使用。用户只需记住智能卡的用户 PIN 码就可利用智能卡自动进行安全身份认证。

4.6.1 颁发智能卡证书管理

要在 Windows 的工作站上进行智能卡用户登录，首先工作站可以颁发智能卡证书给用户。智能卡证书是存储在用户智能卡内的数字证书。设置为用户颁发智能卡证书的具体步骤如下：

1. 使用管理员身份登录到用来颁发智能卡证书的证书颁发机构 (CA)，并打开证书颁发机构管理工具。如下图：

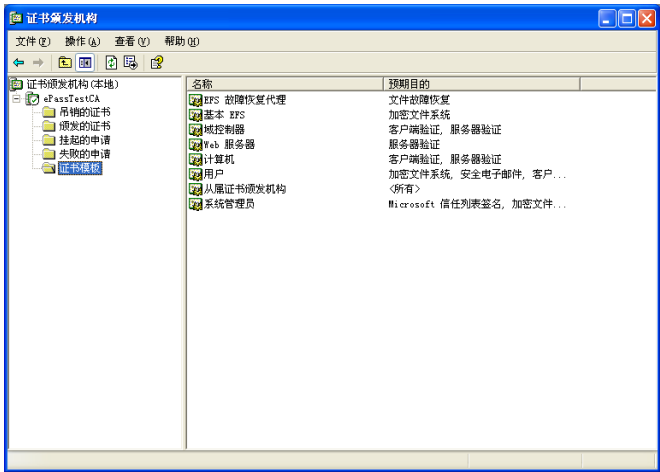


图 4-70 证书颁发机构管理工具

- 在控制台的证书颁发机构树图中选择：证书颁发机构（计算机名）—CA 名称—策略设置。右边的列表显示了当前可颁发证书的类型模板。
2. 在“操作”菜单上选择“新建”菜单的子菜单“要颁发证书”，接下来将显示下面的窗口。

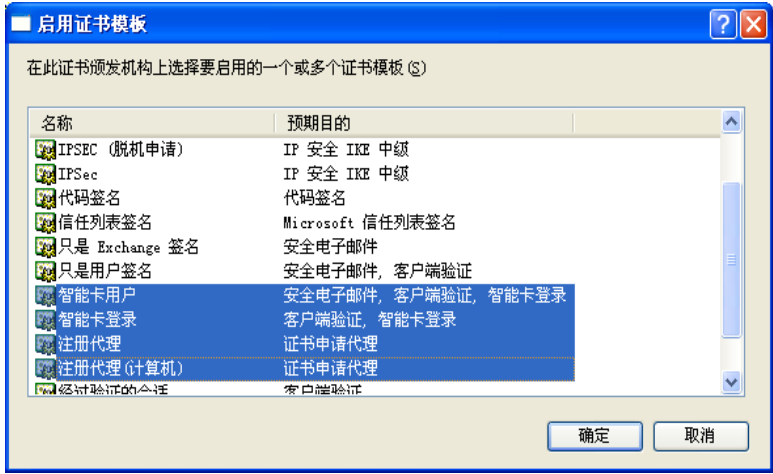


图 4-71 选择证书模板窗口

3. 选择“注册代理”，然后按确定。接下来重复前面步骤，将“智能卡用户”和“智能卡登录”两个证书模板类型也加入到证书模板中。完成后，如下图所示：

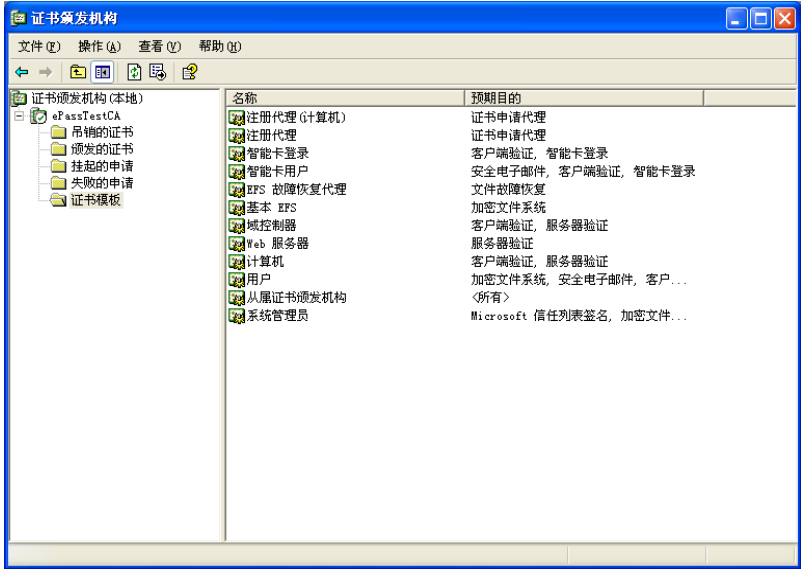


图 4-72 证书模板类型操作

另外，注册代理证书可以由不同于用来颁发智能卡证书的 CA 颁发，前提条件是颁发注册代理证书的 CA 必须是域内被信任的企业 CA。

有了注册代理证书就可以开始建立智能卡证书注册站点了。

4. 以管理员身份登录 Windows Server 2003，单击“开始”菜单并选择“运行”，键入“mmc”然后回车。
5. 在弹出的“控制台”菜单上，单击“添加、删除管理单元”，然后单击“添加”。
6. 在“独立管理单元”的对话框中，双击“证书”。如果以用户身份登录，证书管理单元将自动加载。如果作为管理员登录，则应当选择“我的用户帐户”。如下图所示：

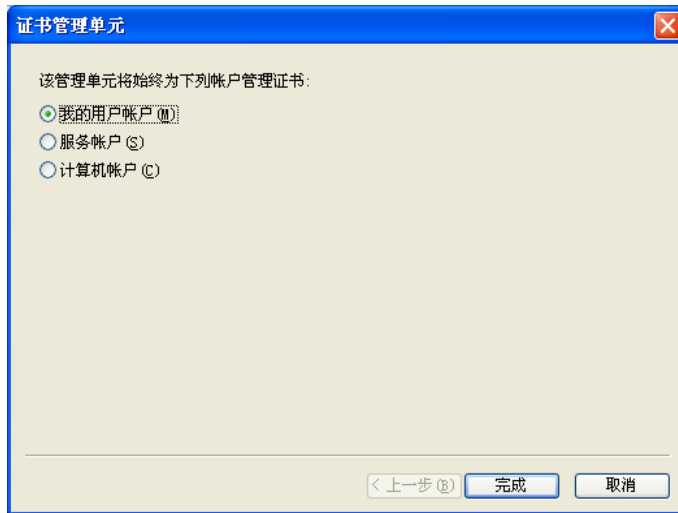


图 4-73 独立管理单元操作

7. 关闭添加管理单元对话框。双击“证书 – 当前用户”，在控制台树中选择“个人”。然后鼠标右键单击，在弹出菜单中选择“所有任务”下的“申请新证书”。如下图：

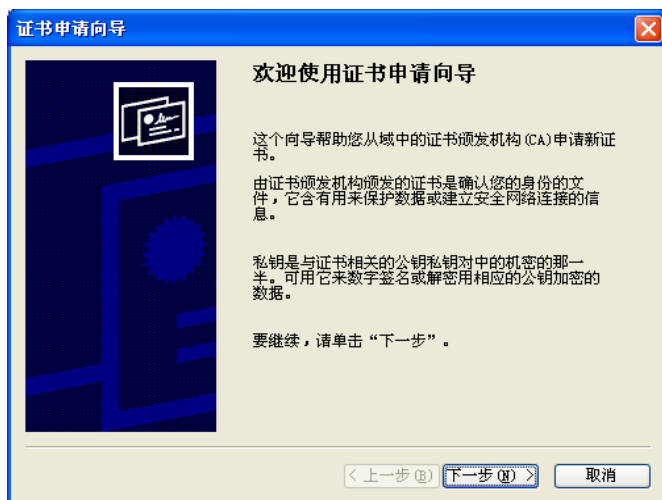


图 4-74 个人证书申请操作

8. 单击“下一步”，选择证书模板“注册代理”。如下图：

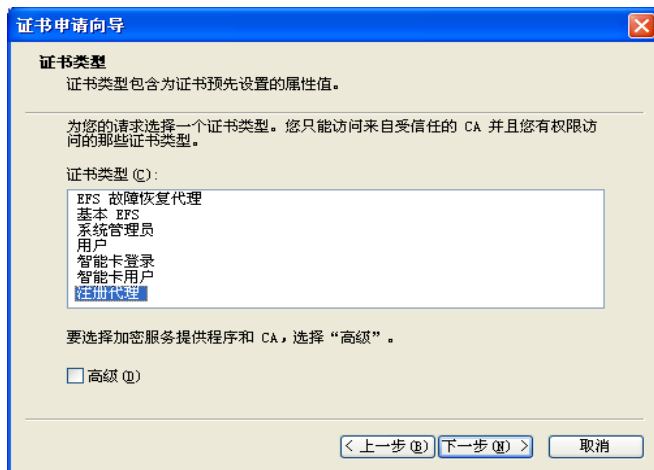


图 4-75 证书申请向导窗口之一

9. 单击“下一步”，在编辑框中输入证书的好记的名称。输入完毕后，继续单击“下一步”完成智能卡代理注册证书的申请。

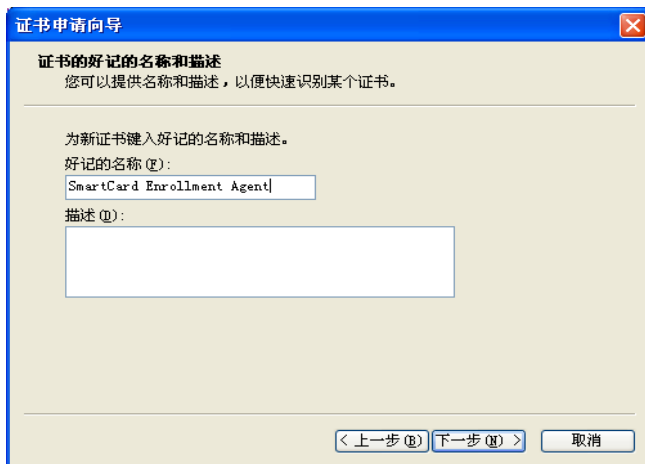


图 4-76 证书申请向导窗口之二

在为域内用户申请智能卡登录证书之前，智能卡管理员必须有可用的注册代理证书，用来代表域用户生成智能卡证书申请。这就是上述操作的目的。要执行这一操作，必须拥有访问注册代理证书模板的安全权限。有关注册代理证书和注册智能卡证书的详细信息，请参阅相关的 Windows 在线帮助。

4.6.2 申请智能卡证书

完成 4.6.1 的步骤之后，就可以正是进行智能卡证书的申请操作了。

1. 以管理员身份登录 Windows。
2. 打开 Internet Explorer，在地址栏中输入用来颁发智能卡证书的证书颁发中心的地址，然后按 Enter 键。
3. 在显示的网页中选择“申请一个证书”。
4. 在显示的网页中选择“高级证书申请”。
5. 插入 ePass2000（也可提前插入，但必须是初始化过的

ePass2000)。

6. 选择“通过使用智能卡证书注册站来为另一用户申请一个智能卡证书”，然后单击“下一步”。（如果是第一次申请证书，浏览器会自动下载两个 ActiveX 控件）。新的页面显示效果如下图所示：

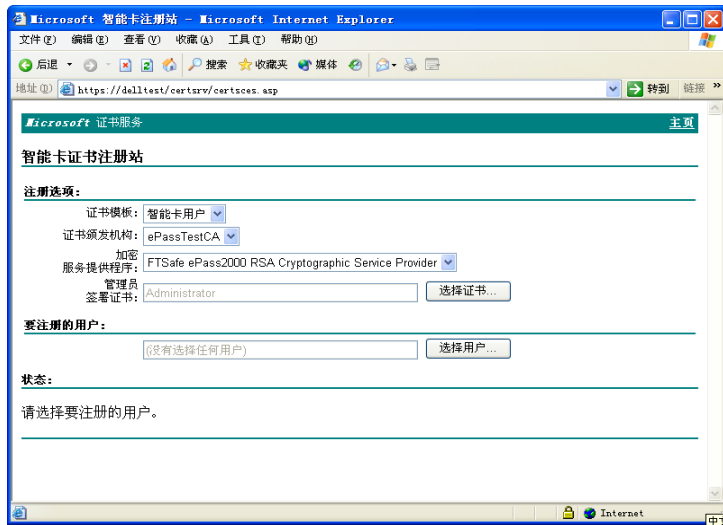


图 4-77 智能卡用户申请证书操作

7. 选择证书模板为“智能卡用户”。
8. 在“加密服务提供程序”中选择“FTSafe ePass2000 RSA Cryptographic Service Provider”。
9. 在“管理员签名证书”中选择我们先前申请的注册代理证书。
10. 在“要注册的用户”中，选择适当的域用户。
11. 在接下来弹出的验证用户 PIN 码对话框中输入正确的 ePass2000 用户 PIN 码，然后等待证书生成。

4.6.3 锁定工作站

当证书下载完成之后，可以选择查看证书或者申请新的智能卡证书。用户智能卡证书申请完成之后，就可以进行使用智能卡进行域用户登录了。用户还可以使用 ePass2000 进行工作站的锁定操作。也就是说，当用户从计算机上拔下 ePass2000 时，系统会锁定工作站，或者强制注销用户。要解除锁定，需要再次插入 ePass2000，并进行 PIN 码校验。设定锁定工作站的操作如下：

1. 打开控制面板中的“管理工具”，双击“域安全策略”图标，下图：

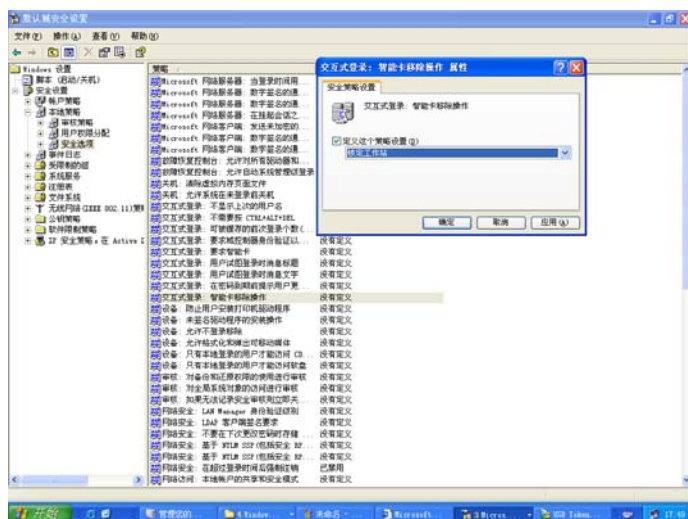


图 4-78 本地安全策略设置

2. 双击“智能卡移除操作”，弹出如上图所示的对话框，然后选择相应的智能卡移除事件相应操作。

4.7 使用 ePass2000 进行 VPN 远程登录

VPN 就是虚拟专用网，可以用来在不安全的网络环境中建立一条安全的网络信道，就好像专用信道一样安全。Windows Server 2003 操作系统内置了对 VPN 应用的支持。Windows 用户可以像使用拨号网络登录到 ISP

的服务器上一样，通过 VPN 连接到需要进行安全传输的网络接入服务器。

在建立 VPN 安全信道的时候，服务器和客户机需要相互进行认证操作，以此建立起安全会话密钥，并用会话密钥完成后续的信息加密操作。Windows 工作站允许用户在客户端使用智能卡进行登录用户身份验证。下面我们以前 Windows Server 2003 的 VPN 路由软件为例讲述 VPN 服务器的配置。

1. 首先打开控制面板里面的“管理您的服务器”来启动“配置您的服务器向导”。如图 4-79

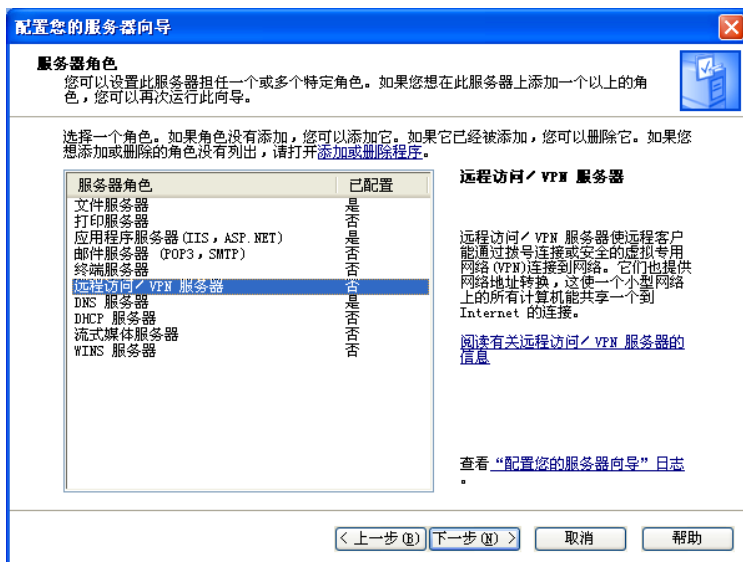


图 4-79 配置路由服务器

2. 单击下一步，并选择进行 VPN 服务的配置。

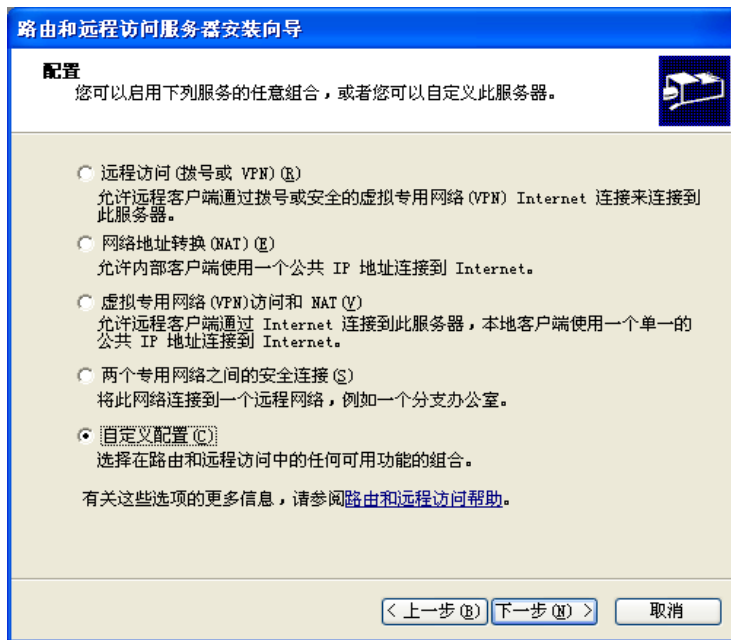


图 4-80 VPN 服务的配置

3. 根据您的情况选择适当的配置，单击“下一步”，选择支持的协议类型。
4. 单击“下一步”，选择网络连接接口。
5. 单击“下一步”，选择 VPN 登录客户端 IP 地址分配策略。可使用现有的 DHCP 服务器进行自动分配，也可以指定固定的 IP 地址范围作为分配地址池。
6. 单击“下一步”，可以为 VPN 配置验证服务器 RADIUS，可以先不配置。

完成上述步骤之后，VPN 接入服务器就安装完成了。接下来，我们需要为 VPN 服务器配置登录验证的方式，我们需要使用智能卡登录方式。

7. 在“路由和远程访问”控制台左边的树型结构中鼠标右键单击服务器名，并选择“属性”。
8. 在弹出的服务器属性对话框中选择“安全”属性页，并单击“验证”按钮。显示结果如下图所示：

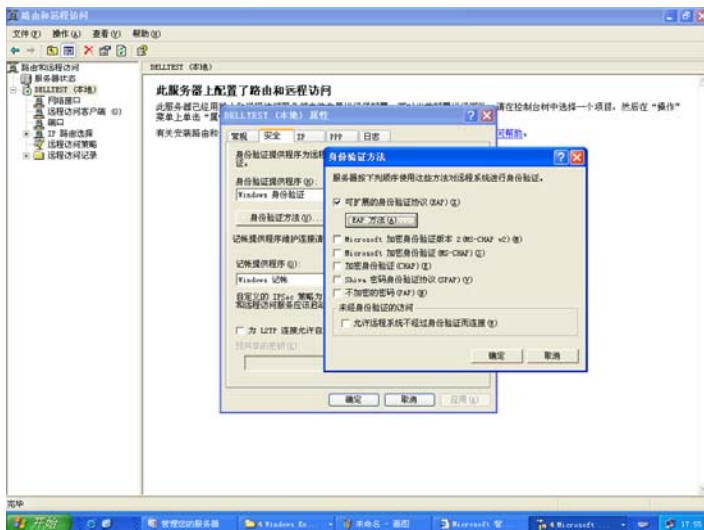


图 4-81 服务器属性设置

9. 选择扩展验证协议(EAP)。扩展验证协议是对传统用户名、口令验证的改进，智能卡用户验证属于扩展验证协议。
10. 关闭“验证方法”对话框。
11. 关闭“服务器属性”对话框。
12. 在“路由与远程访问”控制台窗口左边的树型结构中选择“远程访问策略”。
13. 双击窗口中列出配置项。
14. 在弹出的对话框中单击“编辑配置文件”按钮。
15. 在“编辑拨入配置文件”对话框中，单击“身份验证”属性页。如下图所示：

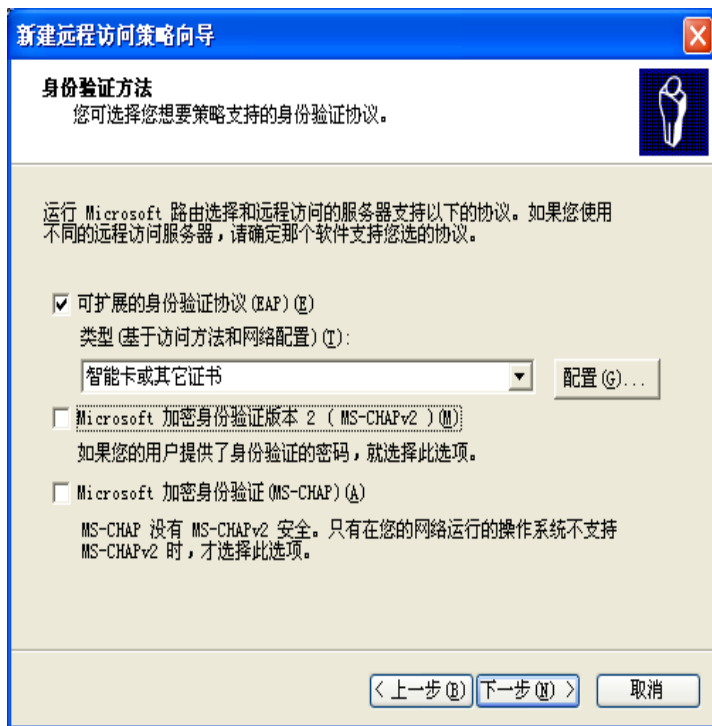


图 4-82 编辑拨入配置

16. 选择扩展验证协议。然后单击“确定”按钮完成配置。
至此，VPN 接入服务器的配置就完成了。

接下来，我们进行客户端软件的配置。

- (1) 打开控制面板中的“拨号网络”工具夹。
- (2) 双击“建立新的链接”图标，弹出“网络连接向导”对话框如下图所示：

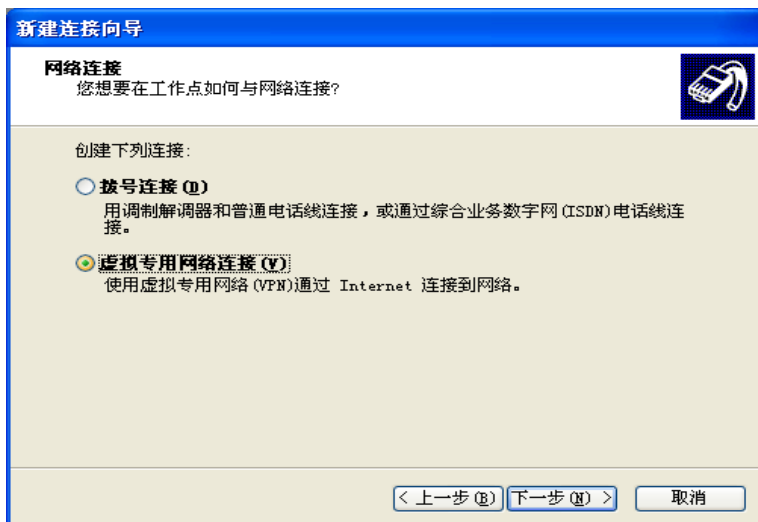


图 4-83 设置网络连接类型

- (3) 选择“虚拟专用网络连接”，并单击“下一步”。
- (4) 接下来填入要连接的 VPN 服务器的 IP 地址或域名。

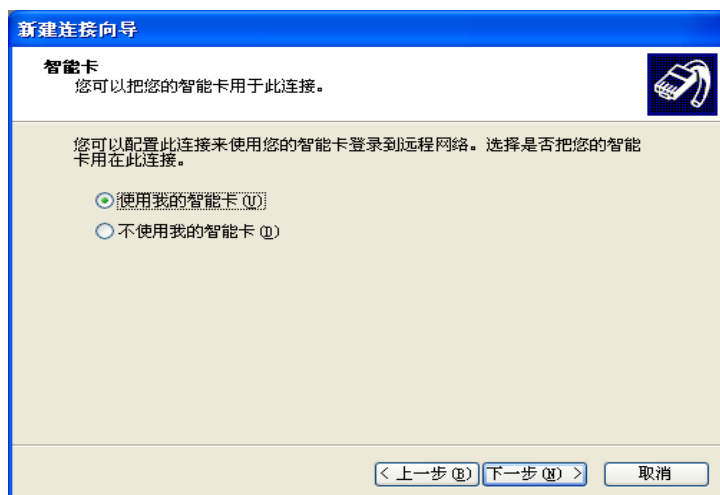


图 4-84 选择智能卡登录

- (5) 选择“使用我的智能卡”。

(6) 最后，输入新建的 VPN 连接的名称。

支持 VPN 客户端软件的配置就完成了。要进行 VPN 登录，双击新建立的 VPN 拨号图标，按提示输入 ePass2000 的用户 PIN 码就可以进行 VPN 连接访问了。

第五章 ePass2000 开发指南

本章讲述有关开发 ePass2000 应用程序的问题，涉及的内容包括 ePass2000 支持的各种开发接口和针对不同接口的开发方法。

- ePass2000 的应用开发接口
- 使用 PC/SC 接口开发 ePass2000 应用
- 使用 MS CryptoAPI 开发 ePass2000 应用
- 使用 PKCS#11 接口开发 ePass2000 应用

5.1 ePass2000 的应用开发接口

ePass2000 的应用程序开发可大体分为两类：一类是 PKI 应用的开发，另一类是智能卡应用的开发。对于针对 PKI 应用程序的开发的接口，ePass2000 提供了 PKCS#11 和 CSP for Microsoft CryptoAPI 2.0 两种主流的应用接口。这两个接口分别遵循 RSA 公司开发的 PKCS#11 标准和微软公司制定的 MS CryptoAPI 接口标准。由于这两个接口标准同时也被其他很多软硬件厂商所支持，因此，ePass2000 可以不需要二次开发就与符合这两种接口规范的应用集成使用。ePass2000 的另一类接口是针对智能卡应用的 PC/SC 接口。

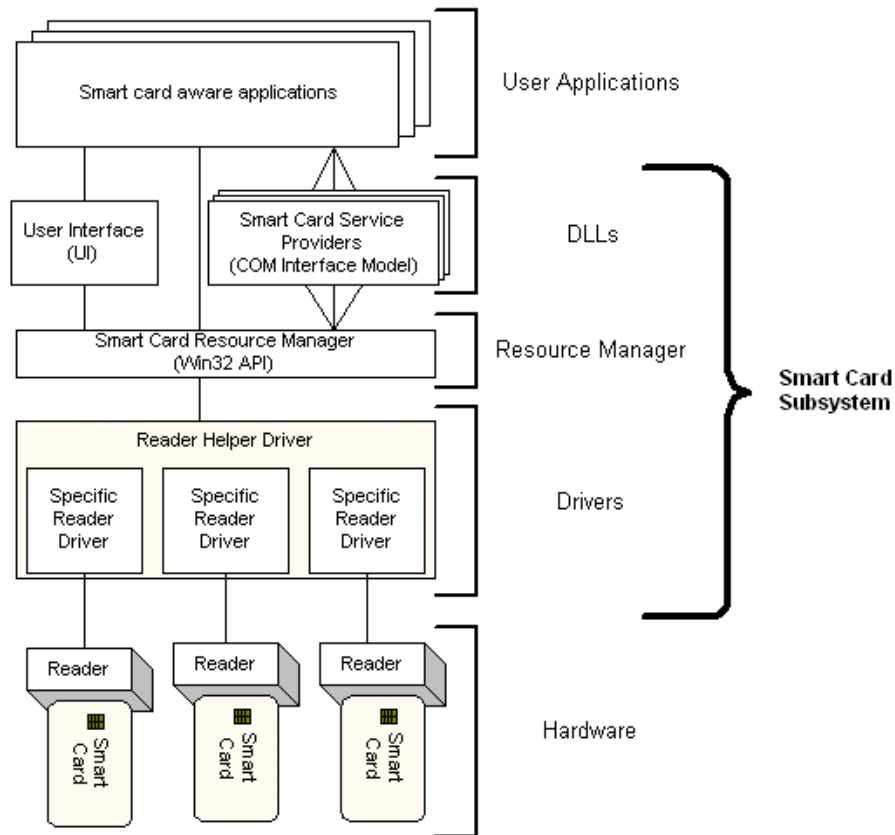
ePass2000 的 PKI 应用接口本身就是建立在 PC/SC 接口之上的。应用程序开发者可以针对自己项目的需求采用一种或多种接口对 ePass2000 进行开发。

5.2 使用 PC/SC 接口开发 ePass2000 应用

Win32 平台上的智能卡子系统是根据 PC/SC 标准设计的（标准参见 <http://www.pcscworkgroup.com/>）。其组成部分包括：

- 使用 Win32 系统编程接口的智能卡资源管理器
- 与智能卡资源管理器协作的用户界面
- 一组提供智能卡服务的 COM 组件

下图描述了 Win32 平台的智能卡子系统架构：



从上图可以看出，Win32 平台的智能卡系统架构实际上将智能卡厂商提供的接口与智能卡应用程序使用的接口隔离开了。也就是说，智能卡应用程序只需使用标准的 Win32 函数的智能卡子集就可以完成对智能卡设备的访问，而且对于不同智能卡厂商，编程的接口都是统一的。智能卡厂商提供的接口发生变化或更新，对上层的智能卡应用程序没有影响。

智能卡资源管理器就是起中间调节作用的抽象层。智能卡资源管理器函数集可分为下面几个部分：

5.2.1 智能卡数据库查询函数

这类函数用来对智能卡数据库进行查询。他们可以查询针对特定系统用户的智能卡类型列表，针对特定智能卡的应用服务接口，系统中的智能卡读卡器分组列表，以及某一分组下所有智能卡读卡器的列表。

在使用这些函数对数据库进行查询操作时，可以针对整个智能卡资源数据库进行查询，也可以指定匹配信息到资源管理器上下文中缩小搜索的范围。修改智能卡资源管理器上下文设置的函数是 **ScardEstablishContext**。对于有些信息，如果不指定特定的上下文信息的话，有可能因为安全的原因而无法访问。

函数	功能
SCardGetProviderId	获取特定智能卡的接口服务程序的标识（GUID）。
SCardListCards	获取特定系统用户可访问的智能卡类型列表。
SCardListInterfaces	获取特定智能卡的接口服务组件的唯一标识（GUID）。
SCardListReaderGroups	获取系统中智能卡分组信息的列表。
SCardListReaders	获取特定智能卡分组下的所有智能卡类型的列表。

5.2.2 智能卡数据库管理函数

这类函数用来管理系统的智能卡资源数据库，并使用指定的资源上下文更新资源数据库的设置。

函数	功能
SCardAddReaderToGroup	添加一个智能卡读卡器到特定的智能卡分组中。
SCardForgetCardType	从系统中删除一个智能卡类型信息。
ScardForgetReader	从系统中删除一个智能卡读卡器信息。
ScardForgetReaderGroup	从系统中删除一个智能卡读卡器分组的信息。
ScardIntroduceCardType	添加一个新的智能卡类型到系统中
ScardIntroduceReader	添加一个新的读卡器类型到系统中
SCardIntroduceReaderGroup	添加一个新的读卡器分组类型到系统中。
SCardRemoveReaderFromGroup	从指定的读卡器分组内删除一个读

	卡器类型。
--	-------

5.2.3 资源管理器句柄函数

这类函数用来建立和释放供智能卡资源管理器查询和管理函数使用的智能卡操作上下文句柄。

函数	功能
ScardEstablishContext	建立用来访问智能卡数据库的上下文句柄
ScardReleaseContext	关闭访问智能卡的上下文句柄

5.2.4 资源管理器工具函数

这个函数用来释放在指定 SCARD_AUTOALLOCATE 标志时，系统函数自动分配的内存区域。

函数	功能
ScardFreeMemory	释放指定 SCARD_AUTOALLOCATE 标志指定时系统函数分配的内存区域

5.2.5 智能卡监视函数

这类函数允许应用程序跟踪智能卡和读卡器当前的状态。这些函数大都使用 SCARD_READERSTATE 结构数组来标识硬件的状态。

函数	功能
SCardLocateCards	查找与指定的 ATR 串匹配的智能卡
SCardGetStatuesChange	阻塞等待智能卡和读卡器的状态发生变化
SCardCancel	取消阻塞或耗时的操作。

5.2.6 智能卡和读卡器访问函数

这类函数用来连接和访问一个指定的智能卡设备。使用包含控制信息的数据块对智能卡进行 I/O 操作。这类控制信息总是使用

SCARD_IO_REQUEST 结构开头。

函数	功能
ScardConnect	连接到一个智能卡
ScardReconnect	重新建立到一个智能卡的连接
ScardDisconnect	中止到一个智能卡的连接
ScardBegingTransaction	开始独占访问一个智能卡设备, 并挂起其他应用程序对智能卡设备的访问
ScardStatus	提供读卡器的当前状态
ScardTransmit	使用 T=0 或 T=1 协议对智能卡进行数据传输操作。

5.2.7 直接卡访问函数

Win32 平台的智能卡子系统允许应用程序访问不完全符合 ISO7816 标准的智能卡设备。为了满足这一要求, Win32 智能卡函数允许应用程序向读卡器直接发送底层控制命令和数据。要使用这些函数, 你必须为每个需要控制的属性设定一个标识。Win32 智能卡子集也定义了一些已有的属性标记。

函数	功能
ScardControl	提供对读卡器的直接访问控制
ScardGetAttrib	获得读卡器的属性
ScardSetAttrib	设置读卡器的属性

对于 Windows 2000 和 Windows XP 平台, 智能卡子系统的组件已经在系统安装的时候自动配置好了, 但是对于 Windows 98 系统来说, 则需要安装微软的智能卡补丁包来支持智能卡应用。

关于 Win32 的智能卡函数集的详细信息请参阅微软的 MSDN 开发者文档。

5.3 使用 MS CryptoAPI 开发 ePass2000 应用

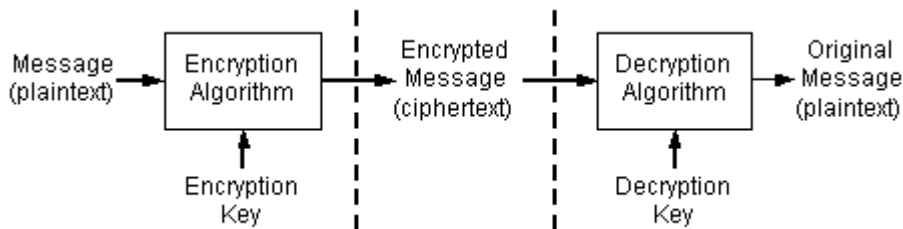
微软的 CryptoAPI 是 Win32 平台下为应用程序开发者提供的加密和安全的编程接口。CryptoAPI 函数集包含了基本的 ASN.1 编码、解码，散列，数据加密和解密，数字证书管理等重要的密码学应用功能。数据的加、解密支持对称算法和公开密钥两类算法。CryptoAPI 是所有微软的 Win32 应用程序及很多第三方厂商应用程序使用的加密接口，诸如 Internet Explorer 和 Outlook 等应用都是基于 CryptoAPI 开发的。

在不安全的网络上进行安全的数据传输涉及三个方面的要求：信息隐藏，身份鉴别和完整性检测。CryptoAPI 除了提供上述三个功能外还提供标准的 ASN.1 编码、解码，信息解密，数字证书和证书存储区的管理，证书信任列表，吊销列表和证书有效性检测等功能。

5.3.1 信息隐藏

信息隐藏的意义就是保证信息内容只能被特定的人获得。信息隐藏通常都要使用某种形式的密码学方法。数据加密算法能保证信息的安全隐藏和传输。数据加密算法将明文数据经过变化使其看上去像一组毫无规律的数据。在没有加密密钥的情况下，对于好的加密算法想从密文强制推导出明文是极其困难的。被加密的数据可以是任意 ASCII 码文本文件，数据库文件和其他任何需要进行安全传输的数据。在这里，“信息”这一术语用来表示任意一段数据，“明文”则指任意一段没有被加密的数据，“密文”则指任意一段被加密的数据。

被加密的数据可以通过不安全的介质或网络进行传输而不损害其安全性。之后，密文可被还原成明文，如下图所示：



数据加密和解密的概念非常简单，对数据加密的时候需要一个加密密钥，这里的密钥就相当于普通的门钥匙一样。解密的时候，需要使用一个解密密钥来解开数据。加密密钥可以相同也可以不同。

加密密钥必须小心保存，给其他用户时也必须通过安全的渠道传递。对解密密钥的访问权限必须小心控制，因为拥有解密密钥意味着可以解开所有用相应加密密钥加密的信息。

5.3.2 身份鉴别

安全通讯的前提是通讯的双方必须明确知道对方的身份。身份鉴别的任务就是鉴定一个用户或实体的真实身份。标识用户身份的文档通常被称为信任状或凭证。当兑现一张支票时，用户可以用身份证或驾照来表达自己的身份。身份证和驾照就是鉴别用户身份的有效凭证。护照是另外一个例子，海关官员通过护照来确定护照持有人的真实身份。海关官员信任发放护照的政府部门对持有人身份的鉴别。在上述两个例子中，用户身份的凭证存在于物理的文档中。

身份鉴别有时也用来判定接受到的数据就是被发送的数据。如果 A 向 B 发送了一段数据，B 需要鉴别这段数据就是 A 发出的，而不是其他人冒充 A 发出的。为了满足这类验证的要求，CryptoAPI 提供了数字签名和校验函数，来对信息进行鉴别。

因为在计算机网络上传输的数据与用户之间并没有物理的联系，因此对数据进行鉴别的凭证也必须能够在网络上进行传输。这种凭证必须由受信任的凭证发行机构发行。

数字证书也就是通常所说的证书就是这样一种凭证，是一种在计算机网络上进行身份验证的有效凭证。

数字证书是由一个被称为证书结构的被信任组织或实体颁发的凭证。它包含与证书对应的用户公钥以及其他一些记录证书主题和用户信息的数据。证书机构只有在验证了证书主题和证书对应的用户公钥的有效性后才会签发证书。

证书申请者和证书机构之间交互签发证书信息可以使用物理介质，比如软盘，进行传输。通常，这种信息交换都是在计算机网络上完成的。证书机构使用被信任的服务程序处理用户的请求和证书的签发工作。

5.3.3 完整性检测

任何通过不安全介质传输的信息都可能被意外或蓄意的修改。在现实世界中，印章就是用来提供和证明信息完整性的工具。例如一瓶阿司匹林使用不可还原的包装和完好的封印来证明它出厂后瓶里的药片没有变化。

同理，信息的接受者不但需要确定信息是由谁发送的，还需要确定自己收到的信息就是发送者发出的信息，而没有任何变化。要建立数据的完整性检测机制，不仅需要发送信息本身，还需要发送用来校验数据的信息，这一信息通常被称作哈希值。数据和验证信息都可以与数字签名一起发送来证明其完整性。

5.3.4 CSP 与加密处理

CryptoAPI 函数使用“加密服务提供程序”（CSPs）完成数据加密，解密以及密钥的存储管理。所有的 CSPs 都是相互独立的模块。理论上，CSPs 应该是独立于特定的应用程序的，也就是说所有的应用程序都可以使用任何一个 CSP。但是，实际上有些应用程序只能与特定的 CSP 协作。CSP 与应用程序之间的关系就类似于 Windows GDI 模型。CSP 就类似于图形硬件驱动程序。

密钥存储的安全性完全取决于 CSP 的具体实现而与操作系统没有关系。这就使得应用程序无需修改就可以运行于多种安全环境之下。

应用程序于加密模块之间的访问必须受到严格的控制。只有这样才能保证应用的安全性和移植性。下面是三条应用原则：

- 应用程序不能直接访问密钥的内容。因为所有的密钥都是在 CSP 内部产生的，应用程序通过不透明的句柄对密钥进行访问。这就避免了应用程序和其关联的动态链接库泄漏密钥或使用不好的随机数产生密钥的可能。
- 应用程序不能指定加密操作的细节。CSP 接口允许应用程序选择

进行加密或签名操作使用的算法类型，但是实际的操作完全由 CSP 内部进行控制。

- 应用程序不处理用户的信任凭证或其他身份鉴别数据。用户身份的鉴别是由 CSP 完成的。因此，对于未来可能出现的身份验证方式，例如指纹识别，应用程序无需修改其身份验证模型。

最简单的 CSP 提供形式是由一个 Win32 动态链接库文件（DLL）和一个签名文件组成。必须提供正确的签名文件 CSP 才能被 CryptoAPI 识别和使用。CryptoAPI 对 CSP 的签名要经常进行周期性的检查，以防止 CSP 程序被篡改。

有些 CSP 模块通过本地 RPC 调用或硬件驱动程序在独立的地址空间中实现其关键的加密操作功能。将密钥和相关的加密操作放到具有独立地址空间的服务程序或硬件内可以保证密钥数据不被应用程序地址空间内的程序随意修改。

应用程序依靠某一 CSP 的特殊特性是很不明智的做法。例如，Microsoft Base Cryptographic Provider 现在提供 40 位的会话密钥和 512 位的公钥。应用程序应避免以此假设存储密钥数据所需的空间大小，因为一旦使用一个不同的 CSP 这一大小就可能不适用。好的应用程序应能与不同的 CSP 协同工作。

5.3.5 CSP 上下文

应用程序调用的第一个 CryptoAPI 函数必定是 CryptAcquireContext。这个函数返回包含指定了特定密钥容器的 CSP 操作句柄。密钥容器的选择可以特别指定也可以使用登录用户缺省的容器。CryptAcquireContext 也可用来创建新的密钥容器。

CSP 模块本身是具有名称和类型的。例如，Windows 操作系统缺省安装的 CSP 程序的名字是：Microsoft Base Cryptographic Provider，其类型是 PROV_RSA_FULL。每个 CSP 的名称都必须是不同的，而类型可以相同。

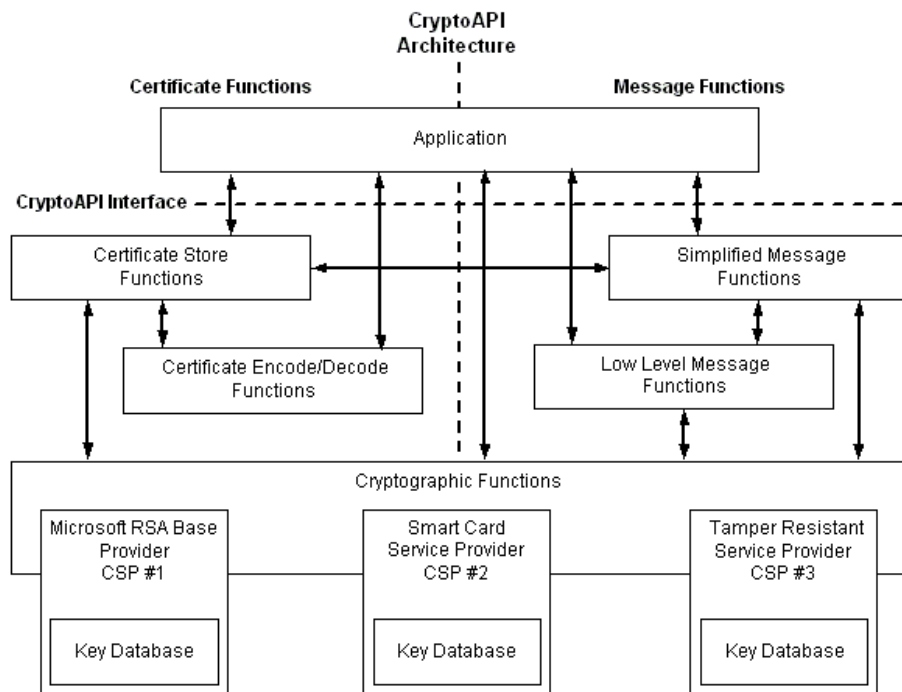
当应用程序调用 CryptAcquireContext 函数获取一个 CSP 操作句柄时，

可以指定 CSP 的类型和名字。如果指定了类型和名字则只有这两个属性匹配的 CSP 模块会被调用。调用成功后，函数返回 CSP 的操作句柄，之后应用程序就可以通过句柄访问 CSP 和 CSP 中的密钥容器了。

5.3.6 CryptoAPI 体系架构

CryptoAPI 体系架构由五个主要部分组成：

- **基本加密函数**
用来连接和建立 CSP 操作句柄的函数。这组函数允许应用程序通过指定名称和类型来选择特定类型的 CSP 模块。
密钥生成函数用来生成和保存加密密钥。其功能包括修改加密模式，初始化加密向量等加密特性。
密钥交换函数用来交换和传输密钥。
- **数字证书编码与解码函数**
这组函数用来加密和解密数据。其功能还包括数据散列计算的支持。
- **数字证书存储函数**
这组函数用来管理数字证书集合。
- **简化信息处理函数**
这组函数用来加密和解密消息及数据，对消息和数据进行签名，验证消息及数据签名的有效性。
- **底层信息处理函数**
这组函数是简化消息处理函数的实际实现函数。它提供了对消息进行各种操作的更为细致的控制。



每类函数的命名前缀都有约定，具体如下：

函数分类	前缀约定
基本加密函数	Crypt
数字证书编码与解码函数	Crypt
数字证书存储函数	Store
简化信息处理函数	Message
底层信息处理函数	Msg

5.3.7 ePass2000 的 CSP 模块

ePass2000 通过提供标准的 CSP 模块来实现与 CryptoAPI 应用程序的无缝集成。ePass2000 的 CSP 模块是遵从微软的 Crypto Service Provider 编程规范编写，可以兼容现有的和将来的 CryptoAPI 应用。

ePass2000 的 CSP 是一个 PROV_RSA_FULL 类型的 CSP，它具有以下

几个特点：

- 提供了安全的 RSA 密钥对存储容器
- 提供多种分组加密算法和哈希算法
- 硬件实现的 RSA 运算
- 便携的个人数字证书载体

除了上述特性之外，ePass2000 的 CSP 模块也针对微软的智能卡 CSP 扩展规范提供支持。使得 ePass2000 可以应用于微软的 VPN 远程客户端登录和 Windows 智能卡登录。

下面列出了 ePass2000 的 CSP 模块支持的函数。这些函数是 CSP SPI，CryptoAPI 应用程序不需要直接调用这些接口。

名称	描述
连接函数	
CPAcquireContext	这个函数为应用程序创建一个上下文。
CPGetProvParam	这个函数返回 CSP 相关的信息。
CPReleaseContext	这个函数释放 CPAcquireContext 创建的上下文。
CPSetProvParam	这个函数设置 CSP 的参数操作。
密钥生成和交换函数	
CPDeriveKey	这个函数从一个数据散列中生成一个会话密钥。它保证生成的密钥互不相同。
CPDestroyKey	这个函数释放一个密钥句柄。释放后，句柄将无效，密钥将无法再被访问。
CPDuplicateKey	这个函数创建密钥的一个拷贝。
CPExportKey	这个函数从 CSP 容器中导出密钥。
CPGenKey	这个函数用来生成密钥或密钥对。
CPGenRandom	这个函数使用随机数填充一个缓冲。
CPGetKeyParam	这个函数用来得到加密操作密钥的属性。
CPGetUserKey	这个函数用来获取 CSP 容器中的持久密钥对。
CPImportKey	这个函数从一个 blob 中导入密钥到 CSP 容器中。
CPSetKeyParam	这个函数设置密钥的属性
数据加密函数	

CPDecrypt	这个函数用来解密先前被加密的数据。
CPEncrypt	这个函数用来加密明文。
散列和数字签名函数	
CPCreateHash	这个函数初始化并散列输入数据。
CPDestroyHash	这个函数删除一个散列对象句柄。
CPDuplicateHash	这个函数创建一个散列对象的拷贝。
CPGetHashParam	这个函数获取散列对象的计算结果。
CPHashData	这个函数散列输入的数据。
CPHashSessionKey	这个函数散列一个会话密钥而不向应用程序暴露密钥的值。
CPSetHashParam	这个函数定制一个散列对象的属性。
CPSignHash	这个函数签名一个散列对象。
CPVerifySignature	这个函数校验一个数字签名。

微软的 CSP 规范中还定义了 OffloadModExpo 这个 CSP 函数，这个函数目前不被 ePass2000 的 CSP 模块支持。

CPAcquireContext 函数是所有 CSP 函数中最先被调用的函数。上层应用通过调用这个函数来指定操作哪一个密钥容器。每一个密钥容器中同时只能保存一对 RSA 密钥对，和任意多个会话密钥。RSA 密钥对是可以持久保存的对象，而会话密钥则只在运行时存在。如果应用程序需要访问密钥容器中的 RSA 私钥，则 ePass2000 的 CSP 模块会要求验证用户的 PIN 码。验证用户 PIN 码的方式是：用户在 CSP 模块弹出的“验证用户 PIN 码”对话框中输入用户 PIN，如果正确则 CSP 模块进行后续操作。如果应用程序希望避免弹出这个对话框则应当设置 CRYPT_SILENT 标志。

5.4 使用 PKCS#11 接口开发 ePass2000 应用

由于 Internet 全球范围内的爆炸式增长，应用程序对公众网领域中的安全事务和通讯的要求也日益迫切。而加密安全产品的迅猛增长也导致了

对应用程序之间交互性的需求。因此 RSA 公司创立了公开密钥加密标准 (PKCS)来满足这一要求。

PKCS#11 是 PKCS 系列标准中的一个。PKCS#11 标准（也称为“Cryptoki”）被设计用来解决不同厂商与开发者的公开密钥应用之间交互与兼容的问题。它定义了一个通用的编程接口模型 Cryptoki tokens，ePass2000 的 PKCS#11 接口符合 2.11 版本的 PKCS#11 规范。

在使用ePass2000 的PKCS#11 接口开发应用程序前，开发人员应当熟悉PKCS#11 标准。这个标准的文档可以在RSA公司的网站上自由下载，下载网址是：<http://www.rsasecurity.com/rsalabs/pkcs/pkcs-11/index.html>。

ePass2000 的 PKCS#11 接口以 Win32 动态链接库(DLL)的方式提供。开发者可以静态（使用 lib 文件）和动态链接两种方式进行访问。下面列出了 ePass2000 的 PKCS#11 接口相关的文件：

文件	SDK 路径
Pkcs11.h	\Include (由 RSA 公司提供)
Pkcs11f.h	\Include (由 RSA 公司提供)
Pkcs11t.h	\Include (由 RSA 公司提供)
Cryptoki.h	\Include (由 RSA 公司提供)
ep2pk11.lib	\Lib
ep2pk11.dll	\Lib (运行时应位于系统目录下)

ep2pk11.dll 是 ePass2000 的核心库文件，它实现了 RSA PKCS#11 标准中定义的所有接口函数。如果开发人员需要使用这个接口，必须在工程项目中包含 cryptoki.h 头文件。如果使用静态连接方式访问 ePass2000 的 PKCS#11 库，则还应当在项目中包含 ep2pk11.lib 这个文件。

5.4.1 ePass2000 支持的 PKCS#11 类对象

ePass2000 的 PKCS#11 模块支持创建和使用下列类型的对象：

类对象	描述
CKO_DATA	应用程序定义的对象。对象的数据结构可由应用程序任意定义，但是数据

	意义的解释由应用程序负责。
CKO_SECRET_KEY	对称加密算法使用的密钥。
CKO_CERTIFICATE	X.509 数字证书对象
CKO_PUBLIC_KEY	RSA 公钥对象
CKO_PRIVATE_KEY	RSA 私钥对象

所有的类对象都可根据生命期长短的不同分成两大类。一类是持久存储的类对象，这类对象被保存在 ePass2000 的安全存储区域当中，直到应用程序主动删除这些对象；另一类是会话对象，这类对象只存在于运行时建立的特定会话当中，一旦会话结束，这类对象也跟着被删除。决定类对象生命期的模板属性是 CKA_TOKEN，这是个布尔值，所有的类对象都有这一属性。开发人员应当根据 ePass2000 有限的存储空间来决定如何处理对象存储的策略。只有必须持续保存的关键对象才有必要在 ePass2000 的内部存储器进行保存。

PKCS#11 的类对象除了生命期长短有分别之外，在访问权限上也有限制。所有的类对象都可根据访问权限的不同分成两大类：一类是公开对象，这类对象是任何用户都可以访问的；另一类是私有对象，这一类对象只有身份被验证的用户才有权访问。决定对象的访问限制类型的模板属性是 CKA_PRIVATE。这是个布尔值，所有的类对象都有这一属性。应用程序可根据需要决定对象应为私有对象还是公开对象。需要注意的是，私有对象的存储区域和公开对象的存储区域都是有容量限制的，而且是相互独立的。应用程序必须衡量好这两种存储区域容量的比例分配。这一比例在 ePass2000 的初始时已经设置，不能更改了。

5.4.2 ePass2000 支持的加密算法

下表列出了所有 ePass2000 的 PKCS#11 模块支持的加密算法：

算法	加解密	签名校验	散列	密钥对生成	封装
CKM_RSA_PKCS_KEY_PAIR_GEN				√	
CKM_RSA_PKCS	√	√			√
CKM_DSA_KEY_PAIR_GEN				√	
CKM_DSA		√			
CKM_DH_PKCS_KEY_PAIR_GEN				√	

CKM_DH_PKCS_DERIVE					
CKM_RC2_KEY_GEN				√	
CKM_RC2_ECB	√				
CKM_RC2_CBC	√				
CKM_RC4_KEY_GEN				√	
CKM_RC4	√				
CKM_DES_KEY_GEN				√	
CKM_DES_ECB	√				√
CKM_DES_CBC	√				√
CKM_DES3_KEY_GEN				√	
CKM_DES3_ECB	√				√
CKM_DES3_CBC	√				√
CKM_MD2			√		
CKM_MD5			√		
CKM_SHA_1			√		

下表显示了 ePass2000 的 PKCS#11 库支持的密钥的长度：

算法	密钥长度
CKM_RSA_KEY_PAIR_GEN	1024bits
CKM_RC2_KEY_GEN	1-128bytes
CKM_RC4_KEY_GEN	1-256bytes
CKM_DES_KEY_GEN	8bytes
CKM_DES3_KEY_GEN	24bytes
CKM_DSA_KEY_PAIR_GEN	512-1024 bits
CKM_DH_PKCS_KEY_PAIR_GEN	128-2048bits
CKM_DH_PKCS_DERIVE	1-128bytes

5.4.3 ePass2000 支持的 PKCS#11 接口函数

PKCS#11 是针对 Cryptoki 硬件的通用模型定义，不同厂商的 PKCS#11 实现会有一些细节上的差别。

ePass2000 的 PKCS#11 接口模块也有一些不同于规范的地方：

- C_WaitForSlotEvents 函数没有完全实现，这个函数现在不支持阻塞调用方式，应用程序调用这个函数的时候，如果需要阻塞调用则必须自己提供线程处理的代码。
- 有一些 PKCS#11 标准中定义的函数没有被实现，但是它们也被导出了。只是当这些函数被调用的时候，应用程序将得到返回值 CKR_FUNCTION_NOT_SUPPORT。

注：ePass2000 就相当于 PKCS#11 标注中所指的“token”。

PKCS#11 标准中将读卡器称为“Slot”，但是由于 ePass2000 在使用的过程中并不需要读卡器，因此在 ePass2000 的实现中，slot 只是一个虚拟的设备，但对于应用程序来说，并没有什么差别。

下表列出了所有 PKCS#11 2.10 标准定义的接口函数：

名称	描述
一般功能函数	
C_Initialize	这个函数初始化库。在调用其它库函数前必须调用此函数。唯一的例外是 C_GetFunctionList 函数。
C_Finalize	当应用程序结束对库的访问时应调用此函数。
C_GetInfo	得到 cryptoki 库的信息。
C_GetFunctionList	得到库导出函数的指针列表
Slot 和 Token 管理函数	
C_GetSlotList	得到 slot 列表。
C_GetSlotInfo	获取 slot 的信息。
C_GetTokenInfo	获取 slot 中 token 的信息。
C_WaitForSlotEvent	等待 slot 事件的发生，如 token 被插入或移除。
C_GetMechanismList	获取库支持算法的列表。
C_GetMechanismInfo	获取算法详细信息。
C_InitToken	初始化 token。

C_InitPIN	初始化 USER PIN.
C_SetPIN	修改当前登录用户的 PIN 码。
会话管理函数	
C_OpenSession	在应用程序和 token 之间建立会话。
C_CloseSession	关闭会话。
C_CloseAllSessions	关闭应用程序打开的所有会话。
C_GetSessionInfo	获取会话的信息
C_GetOperationState	获取当前加密操作的状态
C_SetOperationState	使用从 C_GetOperationState 调用功能返回的状态恢复库的操作状态。.
C_Login	登录用户到 token。
C_Logout	登出用户。
对象管理函数	
C_CreateObject	创建新的 Cryptoki 对象。
C_CopyObject	创建对象的拷贝。
C_DestroyObject	删除一个对象。
C_GetObjectSize	获取对象的大小。
C_GetAttributeValue	获取对象一个或多个属性。
C_SetAttributeValue	修改对象的一个或多个属性。
C_FindObjectsInit	初始化一次对象查找操作。
C_FindObjects	继续一次对象查找操作。
C_FindObjectsFinal	结束一次对象查找操作。
加密函数	
C_EncryptInit	初始化一次加密操作
C_Encrypt	加密数据
C_EncryptUpdate	继续加密数据
C_EncryptFinal	结束数据加密操作。
解密函数	
C_DecryptInit	初始化一次解密操作。
C_Decrypt	解密输入数据。
C_DecryptUpdate	继续解密操作。
C_DecryptFinal	结束一次解密操作。

消息散列函数	
C_DigestInit	初始化一次散列操作。
C_Digest	散列输入的数据。
C_DigestUpdate	继续散列操作。
C_DigestKey	继续散列一个密钥。
C_DigestFinal	结束散列操作。
签名与消息鉴别函数	
C_SignInit	初始化一次签名操作
C_Sign	签名输入数据
C_SignUpdate	继续一次数据签名操作
C_SignFinal	结束数据签名操作
C_SignRecoverInit	初始化一次数据可恢复的签名操作。
C_SignRecover	继续签名操作。
校验签名和消息鉴别函数	
C_VerifyInit	初始化一次校验操作。
C_Verify	校验一个签名。
C_VerifyUpdate	继续校验签名。
C_VerifyFinal	结束一次校验操作。
C_VerifyRecoverInit	初始化一次数据可恢复校验操作。
C_VerifyRecover	校验数据可恢复的签名。
双功能加密函数	
C_DigestEncryptUpdate	继续一次散列并加密操作。
C_DecryptDigestUpdate	继续一次解密并散列操作。
C_SignEncryptUpdate	继续一次签名并加密操作。
C_DecryptVerifyUpdate	继续一次解密并校验操作。
密钥管理函数	
C_GenerateKey	生成密钥并创建新的密钥对象。
C_GenerateKeyPair	生成密钥对并创建新的公私钥对象。
C_DeriveKey	派生一个私钥或密钥。
C_WrapKey	包装一个私钥或密钥。
C_UnwrapKey	解包一个私钥或密钥。
随机数生成函数	

C_SeedRandom	加入随机种子到生成器中。
C_GenerateRandom	生成随机数。
并行功能管理函数	
C_GetFunctionStatus	这个函数已经废弃。
C_CancelFunction	这个函数已经废弃。

附录一 ePass2000 技术参数

支持的操作系统	Windows 98SE/ME/2000/XP/2003, MAC OS 8/9/10.x, Linux
证书和标准	PKCS # 11 v2.11, MS CAPI, PC/SC, X.509 v3 证书存储, SSL v3, IPSec, 兼容 ISO 7816
处理器	8 比特
存储空间	32K
内置安全算法	RSA, DES, 3DES, DSA, MD5, SHA-1
芯片安全水平	安全加密的数据存储
功率	< 250 mW
工作温度	0 ~ 70° C
存放温度	- 40 ~ 85° C
湿度	0 ~ 100%不结露
接口类型	A 类 USB
外壳	一次性, 防水, 硬塑料外壳
数据存储年限	至少 10 年
写次数	至少 10 万次