

# ***ePass2000 User Guide***

**Version 2.0**

Copyright © 1999-2004, Feitian Technologies Co., Ltd.

<http://www.FTsafe.com>

**Feitian Technologies Co., Ltd.** has made every attempt to ensure the information in this document is complete and accurate. Feitian Technologies is not responsible for any direct or indirect loss or damage from inaccuracies or omissions.

The information in this document is subject to change without notice

**Revision History:**

Date:	Version:	Change:
August 2002	1.00	Initial Release
April 2004	2.00	Second Edition

## Feitian Technologies Co., Ltd

### Developer's Agreement

All Products of Feitian Technologies including, but not limited to, evaluation copies, diskettes, CD-ROMs, hardware and documentation, and all future orders, are subject to the terms of this Agreement. If you do not agree with the terms herein, please return the evaluation package to us, postage and insurance prepaid, within seven days of their receipt, and we will reimburse you the cost of the Product, less freight and reasonable handling charges.

1. **Allowable Use** – You may use the Software with other programs for the sole purpose of protecting communications or encrypting files as described in the Developer's Guide with ePass2000's license. You may make archival copies of the Software.
2. **Prohibited Use** – The Software or ePass2000 hardware token or any other part of the Product may not be copied, reengineered, disassembled, de-compiled, revised, enhanced or otherwise modified, except as specifically allowed in item 1. You may not reverse engineer the Software or any part of the product or attempt to discover the Software's source code. You may not use the magnetic or optical media included with the Product for the purposes of transferring or storing data that was not either an original part of the Product, or a Feitian Technologies provided enhancement or upgrade to the Product. You may not place the Software on a server and make it publicly available.
3. **Warranty** – Feitian Technologies warrants that the ePass2000 tokens and Software storage media are substantially free from significant defects of workmanship or materials for a time period of twelve (12) months from the date of delivery of the Product to you.
4. **Breach of Warranty** – In the event of breach of this warranty, Feitian Technologies sole obligation is to replace or repair, at the discretion of Feitian Technologies, any Product free of charge. Any replaced Product becomes the property of Feitian Technologies.

Warranty claims must be made in writing to Feitian Technologies during the warranty period and within fourteen (14) days after the observation of the defect. All warranty claims must be accompanied by evidence of the defect that is deemed satisfactory by Feitian Technologies. Any Products that you return to Feitian Technologies, or a Feitian Technologies authorized distributor, must be sent with freight and insurance prepaid.

EXCEPT AS STATED ABOVE, THERE IS NO OTHER WARRANTY OR REPRESENTATION OF THE PRODUCT, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

5. **Limitation of Feitian Technologies' Liability** – Feitian Technologies' entire liability to you or any other party for any cause whatsoever, whether in contract or in tort, including negligence, shall not exceed the price you paid for the unit of the Product that caused the damages or are the subject of, or indirectly related to the cause of action. In no event shall

Feitian Technologies be liable for any damages caused by your failure to meet your obligations, nor for any loss of data, profit or savings, or any other consequential and incidental damages, even if Feitian Technologies has been advised of the possibility of damages, or for any claim by you based on any third-party claim.

6. **Termination** – This Agreement shall terminate if you fail to comply with the terms herein. Items 2, 3, 4 and 5 shall survive any termination of this Agreement.

## Contact Information

*World Wide Web:*

<http://www.FTsafes.com>

*Feitian Technologies Co., Ltd.*

Tel: +86-10-62304466

Fax: +86-10-62304477

Email: [world.sales@Ftsafes.com](mailto:world.sales@Ftsafes.com)

Address: Bldg. 7A/5F, 40 Xueyuan Road,  
Haidian District,  
Beijing 100083, CHINA

Please e-mail any comments, suggestions or questions regarding this document to us at:  
[world.sales@Ftsafes.com](mailto:world.sales@Ftsafes.com)

**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.

**USB**

This equipment is USB based.

**FCC certificate of approval**

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

**Microsoft® Windows® Logo Program**

This device past Windows HCT test carried out by Windows Hardware Quality Labs (WHQL) to determine whether a product meets the Windows Logo Program requirements.

**Check Point® OPSEC Partner**

ePass2000 has achieved OPSEC™ (Open Platform for Security) certification from Check Point Software Technologies Ltd. (NASDAQ: CHKP), the worldwide leader in securing the Internet. Through OPSEC certification, ePass2000 seamlessly integrates with Check Point's market-leading VPN-1®/FireWall-1® Next Generation™ software.

# Index

<b>Chapter 1</b>	<b>Introduction to ePass2000 .....</b>	<b>3</b>
1.1	What is ePass2000.....	3
1.2	Why Use ePass2000.....	3
1.3	Advantages of ePass2000.....	4
1.4	ePass2000 Hardware Features.....	5
1.5	ePass2000 System Architecture .....	6
<b>Chapter 2</b>	<b>Installation and Configuration .....</b>	<b>8</b>
2.1	Platform Requirements.....	8
2.2	ePass2000 System Prerequisites.....	8
2.3	Installation.....	8
2.3.1	ePass2000 SDK Installation .....	9
2.3.2	ePass2000 Runtime Library Installation .....	11
2.3.3	Uninstalling ePass2000 .....	12
2.4	ePass2000 Manager.....	13
2.4.1	Start ePass2000 Manager .....	13
2.4.2	Configuration Options.....	14
2.4.3	Certificate Management.....	18
<b>Chapter 3</b>	<b>Cryptography and PKI.....</b>	<b>22</b>
3.1	Cryptographic Technologies.....	22
3.2	Public Key Cryptography.....	22
3.3	Certificates .....	22
3.4	Public Key Infrastructure (PKI) .....	23
<b>Chapter 4</b>	<b>ePass2000 PKI Application Guide.....</b>	<b>25</b>
4.1	Configure Certificate Authority .....	25
4.1.1	Install Certificate Services .....	25
4.1.2	Install Root Certificate .....	31
4.2	Download and Install a Digital ID in ePass2000 .....	34
4.3	Configure SSL Encrypted Web .....	36
4.3.1	Configuring IIS .....	37
4.3.2	To Issue and Download a Certificate .....	45
4.3.3	Setting up an SSL Web Site.....	47
4.4	Encrypting E-Mail with ePass2000.....	51
4.4.1	To set security properties of an email account .....	56
4.4.2	Send a Message with a Digital Signature .....	59
4.4.3	Importing Another User's Certificate Into the Address Book.....	59
4.4.4	Send an encrypted Message .....	59
4.5	Smart Card logon with ePass2000 .....	61
4.5.1	Configure a CA to Distribute Certificates .....	61
4.5.2	Apply for a Smart Card Certificate .....	64
4.5.3	View Valid Smart Card Certificate.....	66

4.6	Integrating Microsoft VPN with ePass2000.....	66
4.6.1	VPN Server Configuration .....	66
4.6.2	Client Configuration.....	71
<b>Chapter 5</b>	<b>Development Guide.....</b>	<b>73</b>
5.1	ePass2000 API.....	73
5.2	ePass2000 PC/SC Interface.....	73
5.2.1	Data Enquiry Functions.....	74
5.2.2	Resource Management Functions .....	75
5.2.3	Resource Manager Handle Functions.....	75
5.2.4	Resource Manager Tool Function .....	75
5.2.5	Monitor Functions.....	75
5.2.6	Smart Card and Card Reader Access Functions.....	76
5.2.7	Direct Access Functions.....	76
5.3	ePass2000 MS CryptoAPI.....	76
5.3.1	CSP Module for ePass2000.....	78
5.3.2	Certificate Storage Space .....	78
5.4	The ePass2000 PKCS#11 Module .....	78
5.4.1	Supported PKCS#11 Object Class .....	79
5.4.2	PKCS#11 Mechanisms Supported by ePass2000.....	79
5.5	Functions of the ePass2000 PKCS#11 Library .....	80
<b>Appendix.....</b>	<b>.....</b>	<b>84</b>

## **Chapter 1                    Introduction to ePass2000**

The Internet has enabled unprecedented levels of communication interchange. Information can be exchanged more easily than ever before. But the Internet is a standards based public network. You can use it to communicate with almost anyone, but almost anyone can use the Internet to access your private network, or “snoop” your confidential communications. ePass2000 is a standards based network token that may be used to secure your communications on the Internet.

This guide is intended to explain ePass2000 application theory and security features. It is also intended to supply practical information for using ePass2000 to enhance your security infrastructure.

This chapter will cover the following topics:

- What is ePass2000
- Why Use ePass2000
- Advantages of ePass2000
- ePass2000 Hardware Features
- ePass2000 System Architecture

### **1.1    What is ePass2000**

ePass2000 is a new generation of data security product. It combines the security features of smart cards with the data transfer capability of a USB port. But it costs less than the traditional smart card devices. The ePass2000 token is small and lightweight, making it easy to carry on a key chain. It can directly communicate with a PC through a USB port without any card reader or additional power.

The driver for ePass2000 token fulfills the PC/SC standard. It works with all applications compliant with PC/SC standard.

### **1.2    Why Use ePass2000**

Traditional smart card devices are expensive because they require a card reader. They are not Plug-and-Play and are often inconvenient to use. The ePass2000 token overcomes these disadvantages.

The ePass2000 token communicates with the PC through a USB (Universal Serial Bus) port at a high rate of data transfer. It supports plug & play and power management, up to 127 devices can be attached to a single USB port. USB port provides much better compatibility and higher transfer speed than traditional serial port. The transfer speed may reach 115,200 baud, almost 12 times faster than the traditional 9600-baud rate.

The ePass2000 token is integrated with smart card technology. Most existing smart card based applications may work with ePass2000 without any modification or update. It provides a flexible

solution for smart card users.

In addition to the traditional smart card applications, ePass2000 also supports many PKI applications. Users may keep the private key and digital certificate in an ePass2000 token to ensure the security of the private key. It provides a secure container for digital certificates, often the weakest link for information security applications.

A special chip inside ePass2000 has enough muscle to run powerful RSA algorithms. This means that once the private key is generated or stored inside the token, it will never leave the tokens secure memory space. Even the token owner cannot read the private key. Tamper proof packaging makes it impossible to access the private key by brute force.

It is well known that the PC environment is not secure. System vulnerabilities, hackers, virus threaten the security of user's sensitive information (such as private keys and passwords). The security of information encryption applications, like digital signature, encrypted E-mail, and user authentication, face the same threats. In general, sensitive information is stored on the hard disk, where it may be stolen or corrupted. The ePass2000 token provides a secure storage space isolated from the PC environment. Personal security credentials are kept safely in the ePass2000 token, isolated from hackers, viruses and other threats.

ePass2000 is designed for secure storage of digital certificates, private keys and passwords, electronic signatures, encryption operations, hard disk data protection and remote or local authentication services.

### **1.3 Advantages of ePass2000**

The ePass2000 token is a portable network token that connects directly to the USB port of a computer. It is ideal for protecting user sensitive data and has the following advantages:

➤ **High Security Levels**

The on-board cryptographic function of the ePass2000 token, based on the RSA algorithm, is much more secure than a software-only solution. All the sensitive information is kept in the token's secure storage space. All signing and encryption operations are performed inside the ePass2000. The private key never leaves the token's secure storage space, effectively thwarting hackers who may try to copy the private key from the PC hard drive.

➤ **Ease of Use**

ePass2000 does not need any external devices. Just insert the token into the USB port of any desktop, laptop, keyboard or monitor. It is plug and play.

➤ **Low Cost**

The ePass2000 token can replace smart cards in many existing applications. Unlike smart cards, the ePass2000 token is directly connected to a USB port without any special reader. The user saves the cost of a special reader, and developers save the cost of supporting additional

hardware installation at users' computers.

➤ **Portable**

It is compact and light-weighted. The ePass2000 token can be easily carried to your key chain.

➤ **Seamless Integration**

We offer interfaces for two widely recognized industry standards: PKCS#11 and Microsoft CryptoAPI. The token can be integrated with any application compatible with either or both of these two standards. And ePass2000 is optimized to work with third party software solutions. In addition, ePass2000 has a large built-in secure memory space and can simultaneously accommodate several digital certificates, private keys, passwords and other personal security credentials. A single ePass2000 token may support multiple PKI applications.

➤ **High Reliability**

The ePass2000 token can safely store security credential for at least ten years.

## 1.4 ePass2000 Hardware Features

➤ **Hardware encryption**

The ePass2000 token supports the following algorithms:

- 1024-bit RSA algorithm, signing and verification
- DES and 3DES
- SHA-1 and MD5

The keys are secure because the essential algorithms are performed by hardware.

➤ **Hardware key pair generation**

The RSA key pair is generated in the ePass2000 hardware. It takes several seconds to generate a 1024 bit key pair. The big prime number used to generate keys is generated by a real random number generator on the chip.

➤ **Hardware random number generator**

The ePass2000 token uses a real random number generator to create the key pair and Message Authentication Code (MAC).

➤ **Multi-level access**

There are 16 security levels in the ePass2000 file system. The file system allows users to define one or more security rights for key management. Users can define complex security permission according to their requirements.

➤ **Secure storage space**

ePass2000 utilizes a processor that has in-chip storage for firmware and data. This design is very secure because data and low level instruction sets need never leave the token.

## 1.5 ePass2000 System Architecture

ePass2000 supports PC/SC standard. Developers may use the standard Microsoft Win32 PC/SC function set to manipulate the ePass2000 token.

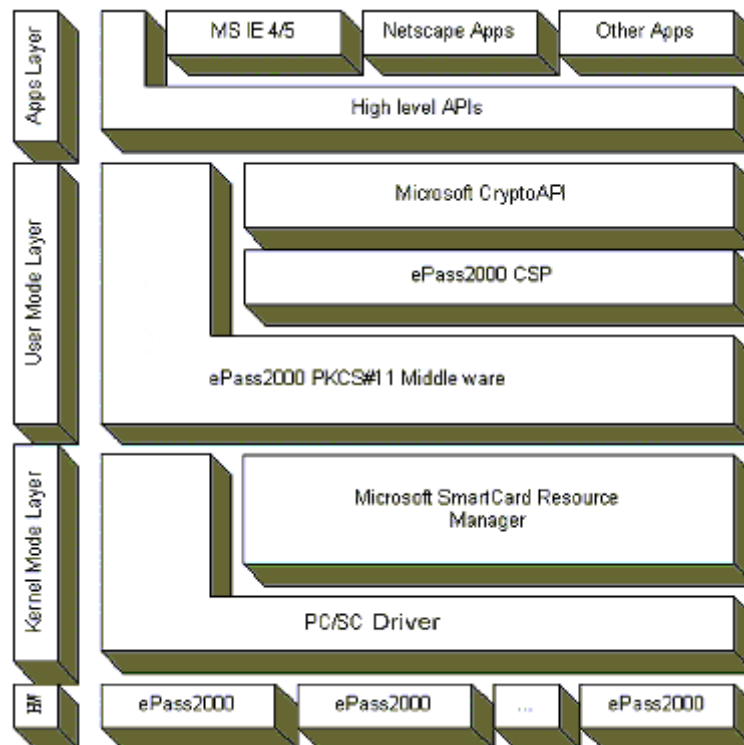


Figure 1-1 ePass2000 system architecture

The ePass2000 system architecture consists of 4 layers: Hardware, Kernel Driver, User Interface and Application Layers.

### ➤ **Hardware Layer**

The Hardware Layer is the bottom layer in the system. It consists of ePass2000 hardware circuit, firmware program and connection cable. It exchanges data with PC based on the standard USB communication protocol via a USB port.

### ➤ **Kernel Driver Layer**

It deals with the data interaction between the PC and Hardware Layer, and the token access requests from the upper layer application. It is the standard PC/SC drive interface. The upper layer applications may access to ePass2000 token via standard Win32 PC/SC function set.

### ➤ **User Interface Layer**

The interfaces on this layer are the PKCS#11 API and MS CryptoAPI interfaces. They are supported by the lower interfaces, compatible with the existing applications, and can be redeveloped. For example, some applications require the users to digitally sign the content they submit in browser with the ePass2000 token. Functions like this require the higher layer interface.

➤ **Application Layer**

Programs at the application layer include generally available applications. The interfaces provided by Feitian Technologies are based on industry standards, and will be familiar to most developers. Developers may integrate their application to the ePass2000 token using interfaces provided by Feitian Technologies.

## Chapter 2            Installation and Configuration

This chapter provides basic information on the installation and configuration of your ePass environment, including initializing your ePass2000 token. The following topics are covered:

- System Requirements
- Installing the ePass2000 SDK and/or Runtime Library (i.e. driver)
- Uninstalling ePass2000 Drivers and Libraries
- Using the ePass2000 Configuration Manager

### 2.1 Platform Requirements

At present, ePass2000 supports the following operating systems:

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

### 2.2 ePass2000 System Prerequisites

The requirements of ePass2000 runtime environment:

- An available USB port for ePass2000
- BIOS supports USB, enable USB in CMOS
- USB extension cable or USB Hub (Optional)
- The ePass2000 token

*Note: Please make sure you have permission to install devices and software.*

### 2.3 Installation

To start using ePass2000, you must first install and set up the ePass runtime environment. This software should be installed before the token is attached to the computer. For development machines, you will want to install the ePass2000 SDK, which includes all the files and tools needed to support ePass application development, along with the ePass2000 runtime libraries and driver. For end-user machines, you need only install the ePass2000 runtime library (i.e., driver) along with any applications you have developed using ePass.

*Note: Please install the ePass2000 software BEFORE you attach the token to the computer.*

### 2.3.1 ePass2000 SDK Installation

To install the ePass2000 SDK, start the SDK Setup Wizard, as shown in Figure 2-1.

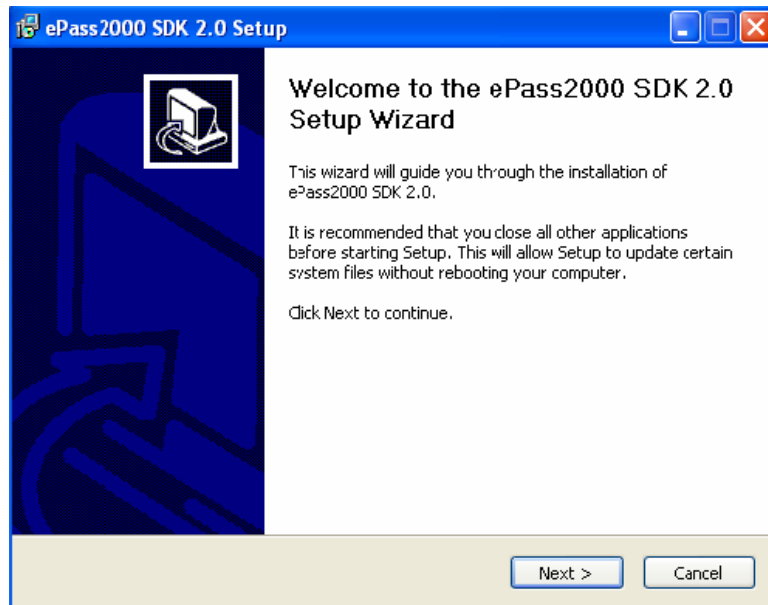


Figure 2-1 ePass2000 SDK Setup screen

Read the Software Developer's License Agreement carefully, click **I Agree** to continue.

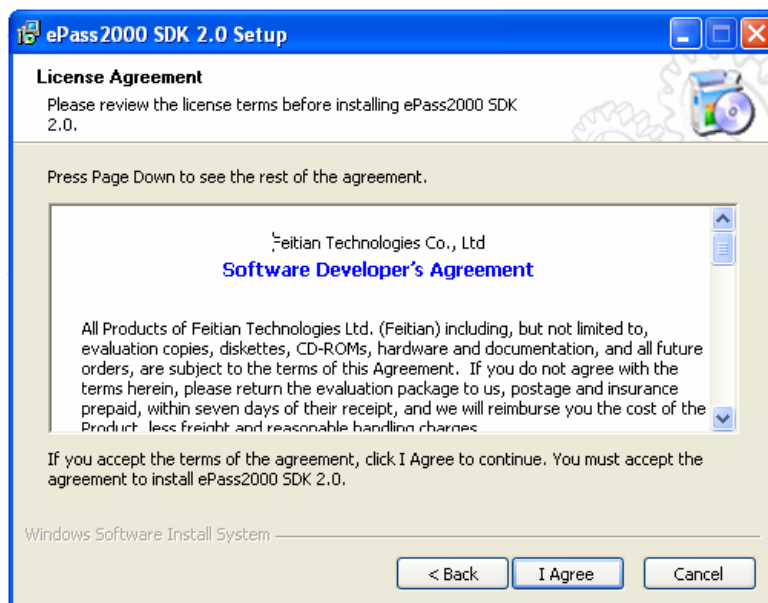


Figure 2-2 License agreement

If you would like the ePass2000 SDK to be installed somewhere other than the default location, input the **Destination Folder** path name next.

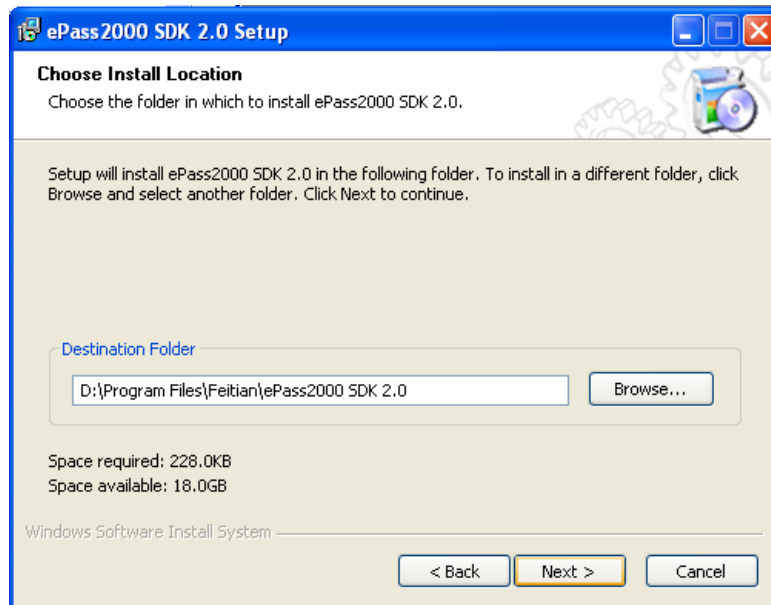


Figure 2-3 Input the installation path

The system will then display the installation options. Select the components you want to install.

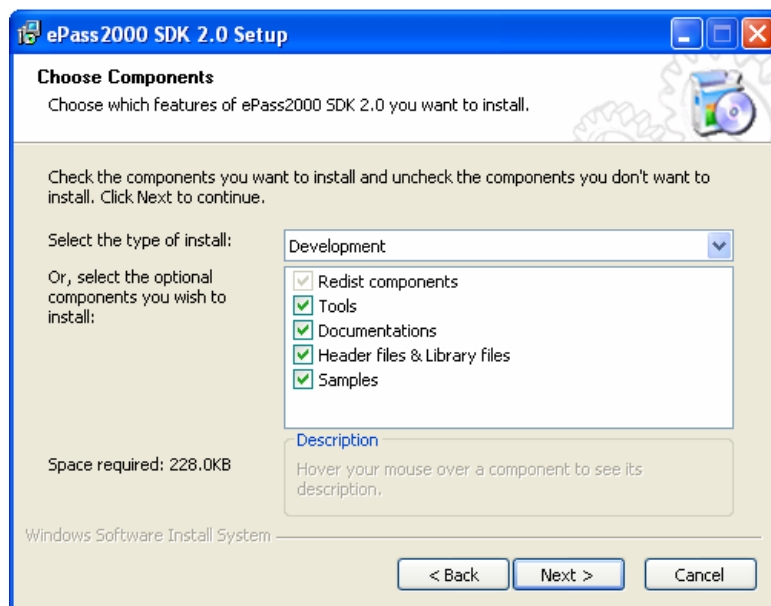


Figure 2-4 Select components to install

The selected ePass2000 SDK components are then installed. A successful dialog will be shown after installation.

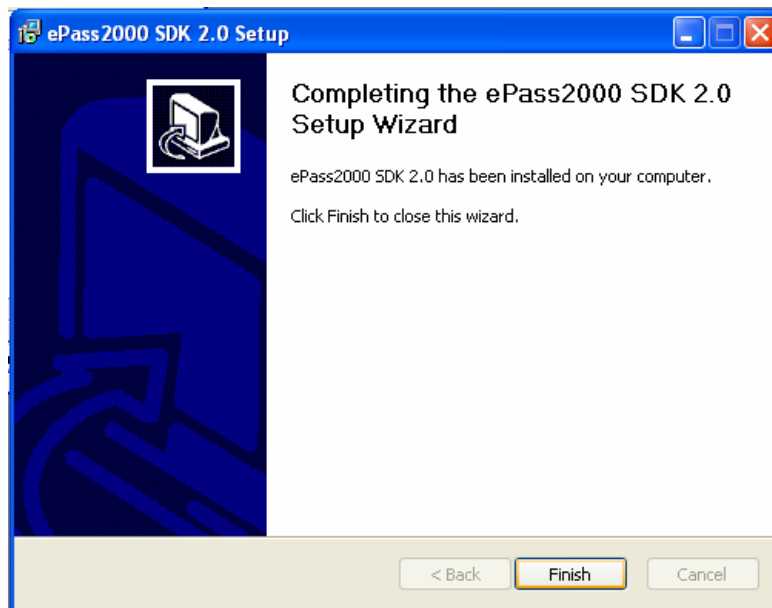


Fig 2-5 SDK installation complete

Click **Finish** to complete SDK installation.

*Note: The installation wizard may prompt you to restart the computer. Please wait for the computer to fully restart before inserting the token in the USB port.*

If the wizard detects Netscape Navigator during installation, it will automatically configure PKCS#11 middleware for Netscape.

After the installation is completed, insert the ePass2000 token to the USB port of your computer. The system will automatically detect that new hardware has been found and activate the already installed driver.

### 2.3.2 ePass2000 Runtime Library Installation

End-user machines need only install the ePass2000 runtime library (i.e., driver) along with their ePass-enabled applications, and need not install the complete SDK. To install the ePass2000 driver, start the InstallShield Wizard, as shown in Figure 2-6.

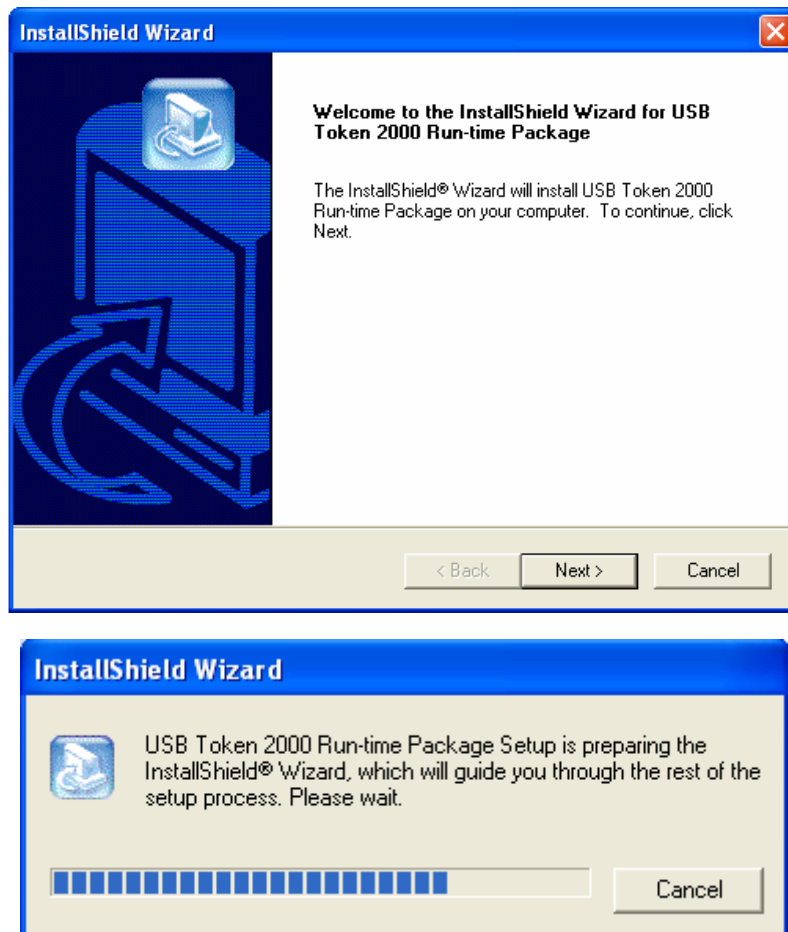


Fig 2-6 Install ePass2000 runtime package

*Note: The installation wizard may prompt you to restart the computer. Please wait for the computer to fully restart before inserting the token in the USB port.*

If the wizard detects Netscape Navigator during installation, it will automatically configure PKCS#11 middleware for Netscape.

After the installation is completed, insert the ePass2000 token to the USB port of your computer. The system will automatically detect that new hardware has been found and activate the already installed driver.

### 2.3.3 Uninstalling ePass2000

To remove the ePass2000 software from your machine, select **Start > Settings > Control Panel > Add/Remove Programs..** The Add/Remove Programs Properties dialog box is displayed. Select **ePass2000 Full Redist Package**, then the un-installation wizard will be started.

During uninstallation, some files may not be deleted because they are accessed by other systems or applications. The wizard will ask you whether you will try again to delete these files. If you have not installed anything that uses these files, you may choose to try again or simply delete them after restarting the computer. You can also check to see if some other programs are accessing ePass2000

token, close them, and then try to delete the files.

*Note: The installation wizard may prompt you to restart the computer. If the computer does not restart, do NOT re-install the ePass2000 drivers and runtime libraries, or insert ePass2000 token.*

## 2.4 ePass2000 Manager

After installing the ePass2000 software, the ePass2000 Manager is used to setup and configure the ePass2000 token. There are two versions of the ePass2000 Manager program -- one for developers, and the other for end-users. The two versions are identical, except that the end-user version lacks the initialization, User-PIN unblock and change SO PIN functions found in the developer's version.

### 2.4.1 Start ePass2000 Manager

With your ePass2000 token inserted, start the ePass2000 Manager program. The ePass2000 Manager will issue an error message if it cannot detect an ePass2000 token attached to the computer.

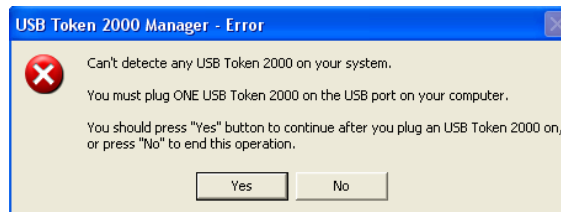


Figure 2-7 No token error message

*Notes: The ePass2000 Manager supports only one ePass2000 token at the same time. Please make sure only one ePass2000 token is connected to the PC.*

After inserting the ePass2000 token in your computer, click **Yes**.



Figure 2-8 Initial ePass2000 Manager screen

## 2.4.2 Configuration Options

To setup your ePass2000 token, click **Configure** on the main ePass2000 Manager menu. For developers, there are 5 options:



Figure 2-9 Configure ePass2000 options

*Note: The end-user version has only two options: Change user PIN, and Change token's name*

Each function is described in detail below.

### Initialization (developer version only)

The ePass2000 token must be initialized before it can be used. The initialization procedure will assign storage space for public and private data, and set the token name and PIN. Only the ePass2000 developer (Security Officer) may perform this function. To initialize the ePass2000 token, click **Initialize** or **Do it** on the main ePass2000 Configure menu.

**NOTE: THE INITIALIZATION PROCESS IS IRREVERSABLE.**

**ALL DATA ON THE TOKEN WILL BE ERASED DURING INTITIALIZATION.**

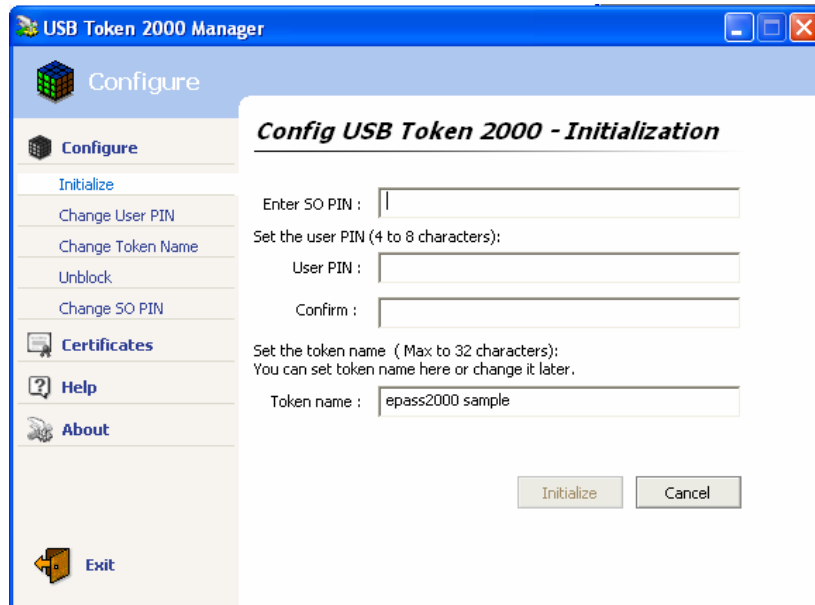


Figure 2-10 Token initialization

1. Enter both the SO (Security Officer) and user PIN. Only the SO can reset the SO PIN. The end-user only has rights to reset the user PIN.
2. Enter the token name. The token name may be used in some PKCS#11 based applications. You may change it later with the **Change token name** function.

Be sure that each field on this screen is filled in correctly. Click **Initialize** to perform the initialization operation.

*Note: The system will prompt you with an error message if the initialization process fails. If you receive any error messages, verify that each input field is within acceptable limits.*

The wizard will return to the main ePass2000 Manager menu when the initialization process finishes successfully (see figure 2-9).

### Change User PIN

To change the User PIN, click **Change user PIN** or **Do it** on the main ePass2000 Configure menu.



Figure 2-11 Change user PIN

Input the old user PIN first. Then input a new user PIN and click **OK**. The **OK** button will be grayed out if the user pin is not with-in the four to eight character limit.

*Note: The system will prompt you with an error message if the old user PIN is incorrect. A locked token can only be reset by the Security Officer*

### Change Token Name

To change the token name, click **Change Token Name** or **Do it** on the main ePass2000 Configure menu.

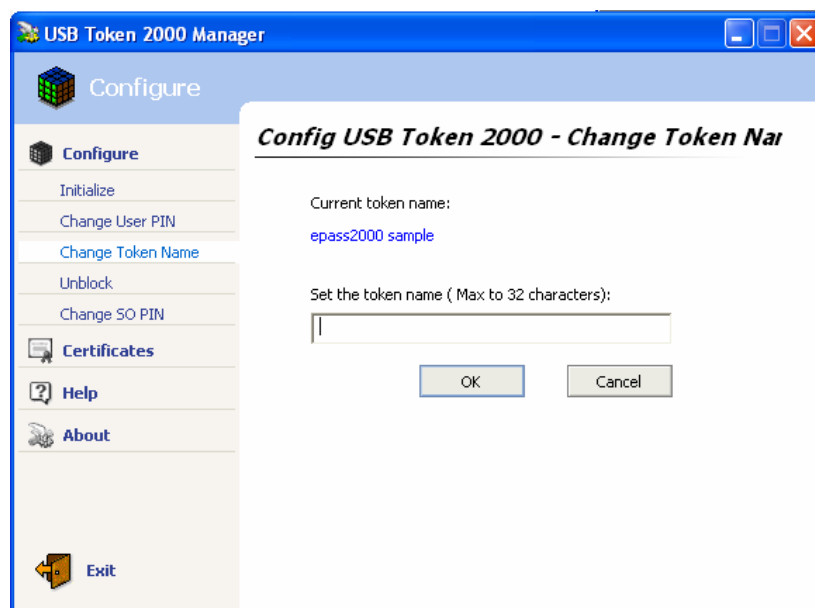


Figure 2-12 Change token name

You may view the current name of your token on this screen. Input the new name and click **OK**. The name of the token will be changed.

#### Unblock User PIN (developer version only)

The Security Officer will have to reset the user PIN if the user forgets the PIN. From the main ePass2000 Configure menu, click **Unblock user PIN** or the **Do it** button and then fill the fields in Figure 2-11:

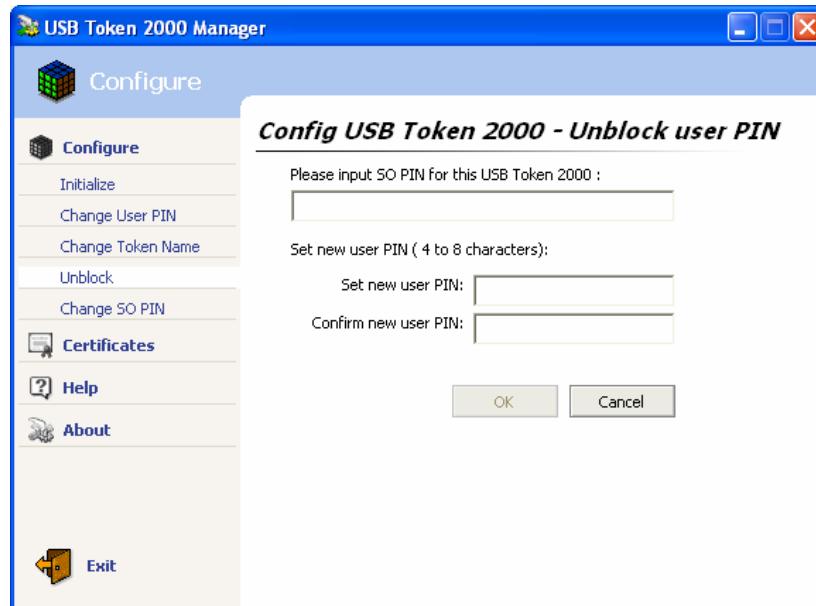


Figure 2-13 Unblock user PIN

Input the SO PIN first, then the new user PIN (4~8 characters). Click **OK** to reset the user PIN. Only the ePass2000 token issuer (Security Officer) has the rights to unblock a user PIN.

#### Change SO PIN (developer version only)

In the main ePass2000 Configure menu, click **Change SO PIN** or the **Do it...** button to set a new SO PIN:

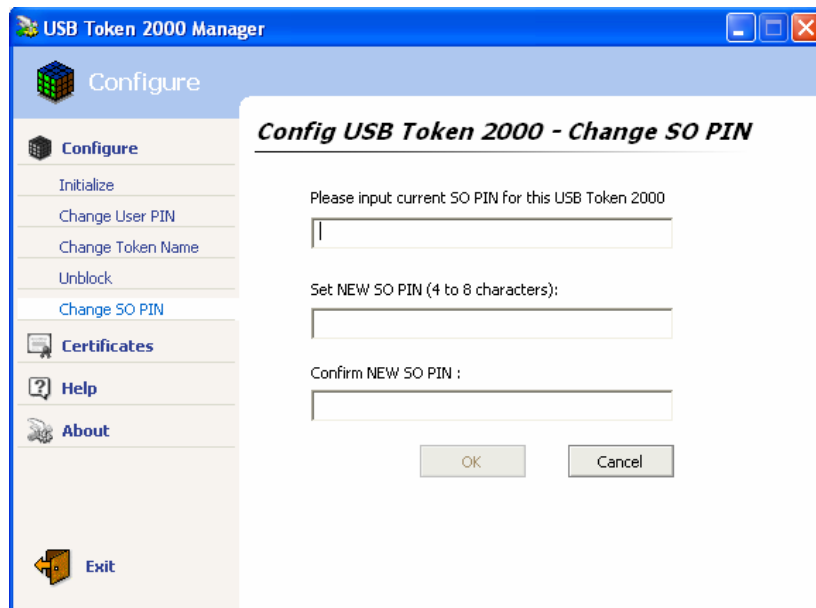


Figure 2-14 Change SO PIN

First input the existing SO PIN, then the new SO PIN. The OK button will be grayed out if the entered PIN(s) are not 4 to 8 characters in size.

### 2.4.3 Certificate Management

Many ePass2000 applications will involve the use of digital certificates, private and public keys. Private keys should always be securely stored on your ePass2000 token, as should certificates. ePass supports X.509 v3 certificates and the ePass2000 Certificate Manager allows you to import and manage a half dozen or so digital certificates on the token. Detailed instructions are provided in the sections below.

#### Login Certificate Manager

To login, select the **Certificates** option and input the user PIN:

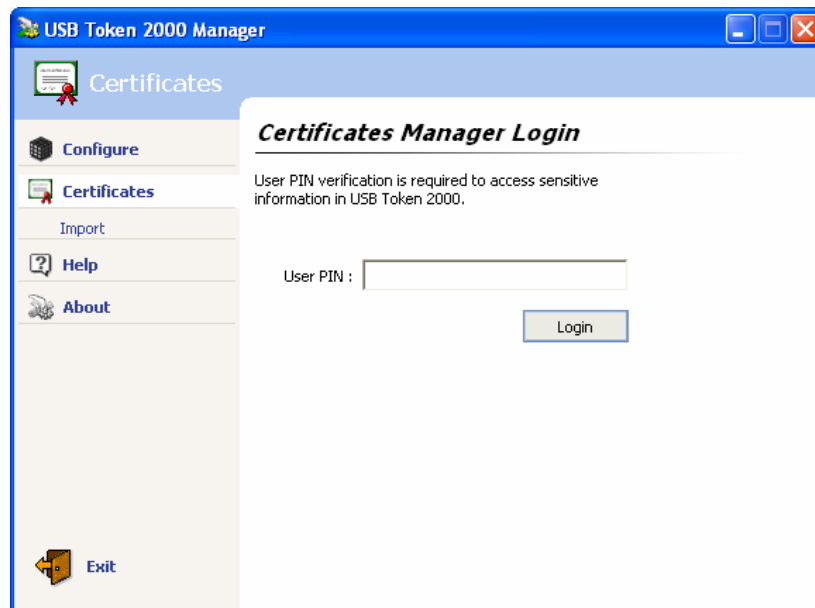


Figure 2-15 Certificate manager login

Click **Login** to view the Certificate Manager main window.



Figure 2-16 Certificate management

Certificate owner, issuing authority, and period of validity will be displayed for each certificate stored in the token.

*Note: An invalid certificate is still a qualified certificate. It may be marked invalid because it has not been activated, has expired or has been suspended. An invalid certificate cannot be used.*

### View Certificate Information

Double click the certificate, or click and select **View** to see detailed information regarding the certificate:

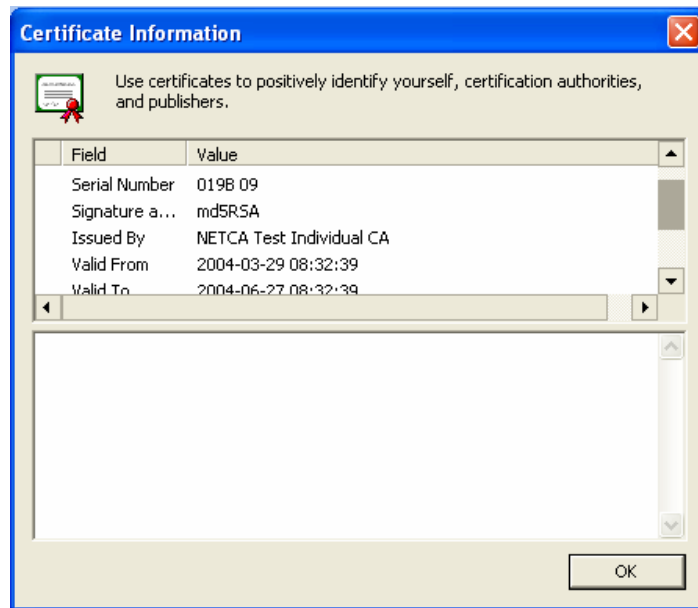


Figure 2-17 View a certificate

Certificate version, serial number, CA, valid dates, public key and other information will be displayed in this window. Click each item to view more detailed information in the text box. See Figure 2-17.

A red marker will be displayed beside the items that caused a certificate to become invalid. In the figure below, there is a red marker beside the **Valid To** item. This indicates that the certificate is invalid because it has expired.

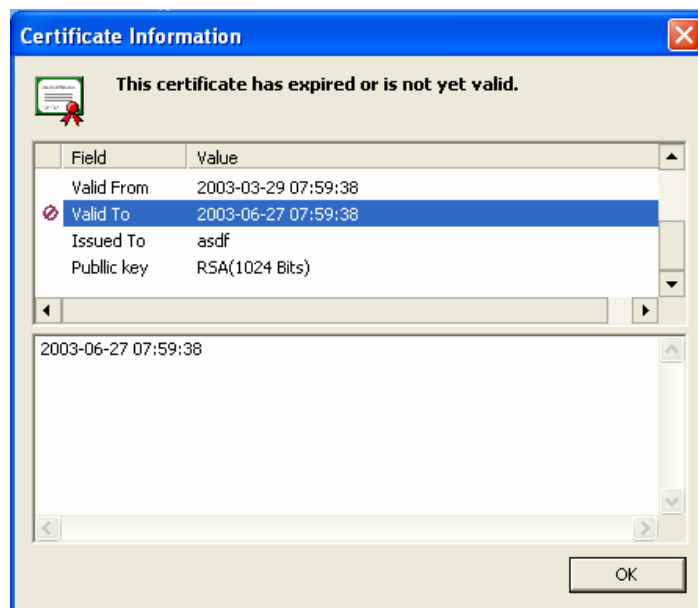


Figure 2-18 An invalid certificate

*Note: ePass2000 Manager only recognizes X.509 certificates. Be sure your CA is not using another certificate format before you delete the certificate.*

## Import Certificate File

You can import certificates to ePass2000 with the ePass2000 Manager. Select **Import** on certificate manager main screen to begin:

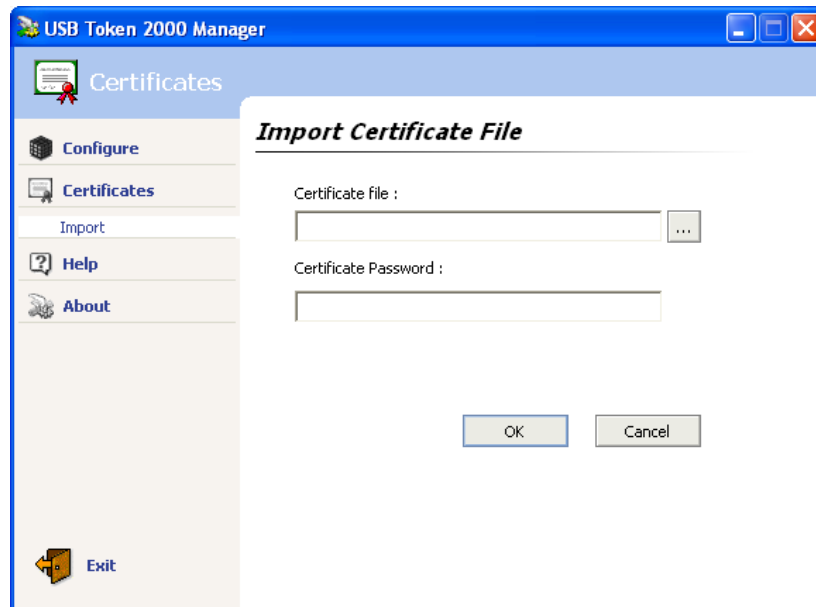


Figure 2-19 Import certificate

Type the path of the certificate file in the field shown above, or **click ...** to browse for it. Input the certificate password and click **OK** to import.

### Notes:

1. Any certificate exported from another PKI application should be based on X.509 v3 standard and include the private key. The exported certificate should be password protected to prevent loss of the private key.
2. Once a certificate with a private key is imported to the ePass2000 token, it cannot be exported. In this sense, no private keys can be found out of ePass2000. This requirement insures the integrity of the certificate data.

## Delete Certificate

You can delete certificates stored in ePass2000 token. Choose the targeted certificate from the certificate list in Certificate Manager. Click the **Delete** button to delete it. The occupied space will be released.

## **Chapter 3                      Cryptography and PKI**

Cryptography is the science of enciphering or deciphering messages in a secret code. It is a very broad discipline that has been transformed by the computer revolution begun in the 1960s. ePass2000 is based on technologies that grew out of basic research into cryptography.

The Internet is now becoming the default platform for business communications. The need then for security and authentication services is urgently needed by enterprises in all business sectors that rely on the Internet to transmit sensitive information. Cryptography provides the technologies protect your data as it travels the Internet, and verify the identity of the recipient.

### **3.1    Cryptographic Technologies**

Cryptography has lead to the development of technologies vital to the development of secure communications over the Internet. These new technologies fall into three domains:

Encryption is the transformation of data into a form that is impossible for unauthorized people to read. The purpose is to ensure privacy by keeping information hidden from anyone for whom it is not intended.

Decryption is the reverse of encryption; it is the transformation of encrypted data back into an intelligible form.

Authentication is when you verify certain information. You can verify the origin of a document, the identity of the sender, the identity of a computer, and so on.

### **3.2    Public Key Cryptography**

In public key cryptography, you use a key pair when you encrypt and decrypt messages. This key pair consists of a public key, which is published widely, and a private key, which is kept secretly and only known to the owner. The public key is used to encrypt messages and the private key is used to decrypt them.

For example, when Alice wishes to send a secret message to Bob, she uses Bob's public key to encrypt the message and sends it to Bob. Bob then uses his private key to decrypt the message and read it. Any eavesdropper cannot decrypt the message. Anyone can send an encrypted message to Bob, but only Bob can read it because only Bob knows his own private key.

### **3.3    Certificates**

Driver licenses and passports provide generally recognized proof of a person's identity. A certificate is an electronic file, which identifies a person and associates a person with a public key. Without the certificate, anyone can generate a key pair in your name and then use "your" public key,

pretending to be you. Public key cryptography uses certificates to address the problem of impersonation.

X.509 is the most widely used standard for defining digital certificates. An X.509 certificate contains information about its owner, issuer, validity period, certificate serial number, certificate issuer (CA) information and information to verify certificate integrity, such as electronic fingerprints.

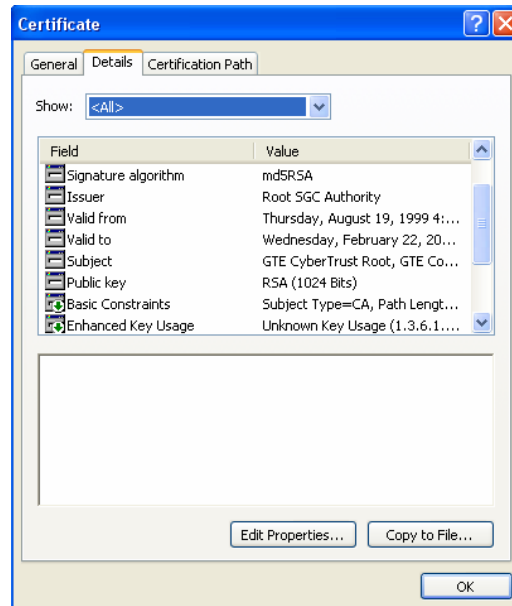


Figure 3-1 A sample of certificate

### 3.4 Public Key Infrastructure (PKI)

A PKI consists of standards and services that support applications of public key cryptography. The main issues to running a PKI include how keys should be managed, how users have their identities checked, and how a specific user's public key is made available to other users. A PKI consists of CAs and end entities to build a hierarchy of trust.

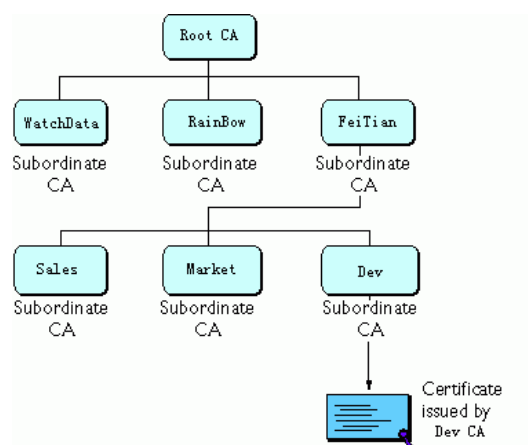


Figure 3-2 The structure of CA

Certificate authorities (CAs) are entities that validate identities and issue certificates. They can be either independent third parties or organizations running their own certificate-issuing server software (such as the Netscape Certificate Server).

The certificate issued by the CA binds a particular public key to the name of the entity the certificate identifies (such as the name of an employee). Certificates help prevent the use of fake public keys for impersonation. Only the public key certified by the certificate will work with the corresponding private key possessed by the entity identified by the certificate

## Chapter 4                      ePass2000 PKI Application Guide

ePass2000 is designed to be seamlessly integrated with existing PKI applications. PKI application developers can integrate ePass2000 simply by configuring the relevant server, with no programming required. This chapter will discuss following topics.

- Configure Certificate Authority
- Apply for Digital Certificate with ePass2000 token
- Configure SSL Encrypted Web
- Access SSL Encrypted Web with ePass2000 token
- Receive/Send Digital Signature and Encrypt E-mail with ePass2000
- Smart Card Logon with ePass2000
- Integration with Microsoft VPN

### 4.1 Configure Certificate Authority

The deployment and use of PKI requires a trustworthy institution (i.e., a Trust Center or a Certificate Authority) to distribute, revoke, and manage keys and certificates. The Windows environment offers one such mechanism to support PKI applications. In this section, we will use the CA of Windows 2000 Advanced Server as an example of how to configure a CA for use by ePass. After proper CA configuration, you can perform smart card logon, workstation lock, VPN remote access, SSL encrypted web access and other security operations easily with ePass.

#### 4.1.1 Install Certificate Services

Windows 2000 Advanced Server does not install certificate services by default because Windows2000 cannot change the name of computers after installing certificate services. To install certificate services in Windows2000 Server, choose **Windows Components** from **Add/Remove Programs**. A more detailed description is provided below

Install Certificate Service to Windows2000 Server.

1. From the **Start** menu, select **Settings > Control Panel**.
2. Double click **Add/Remove Programs** icon.

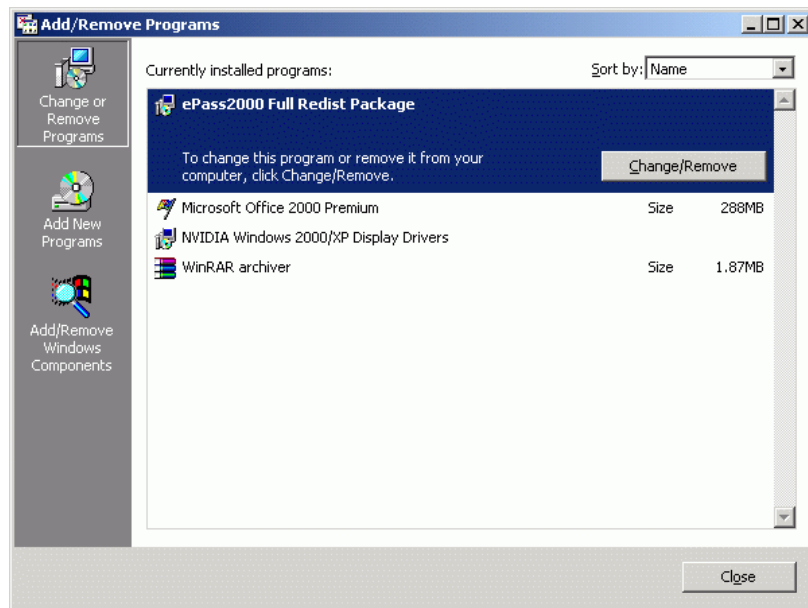


Figure 4-1 Add/Remove Programs

3. Click **Add/Remove Windows Components**. The Windows Component Wizard will start. Choose your targeted service or component

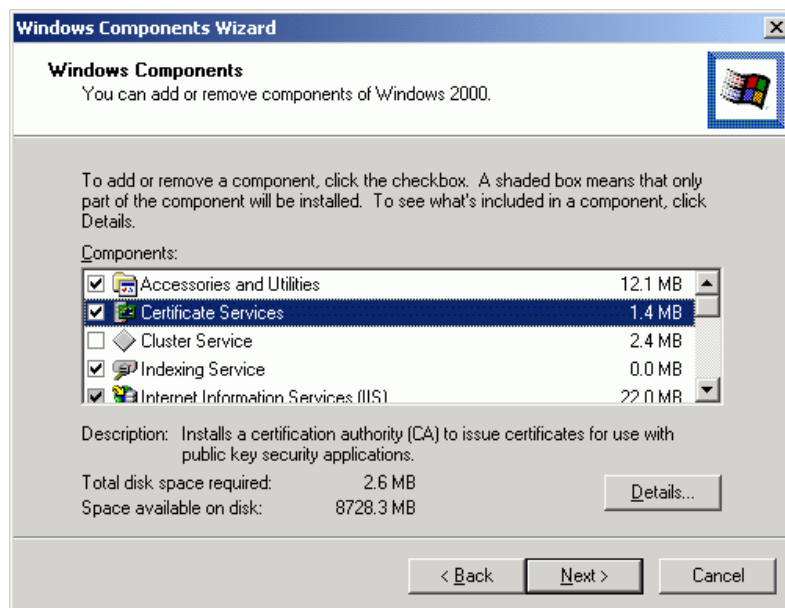


Figure 4-2 Windows Components Wizard

4. Select **Certificate Services** from the component list and follow the prompts. After installing Certificate Services, you cannot change its name, add it to other domain, or delete it from the current domain. Before you install Certificate Services, please make sure this is a properly configured and stable machine.
5. Click **Next** button to set Certification Authority Type. Choose the CA type you demand, then click **Next** to continue. The figure below lists possible CA types:

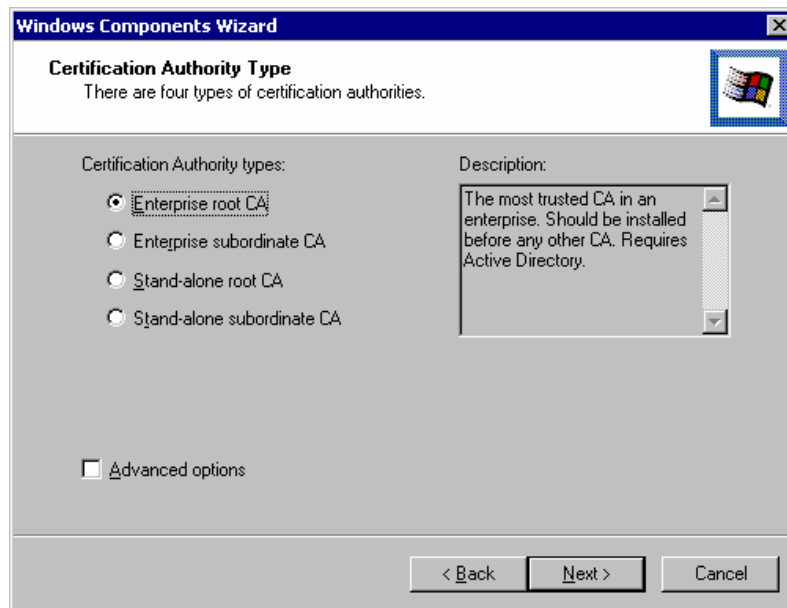


Figure 4-3 Select CA type

An Enterprise Root CA can issue certificates for purposes such as digital signatures, secure e-mail using S/MIME, authentication to a secure Web server using SSL, and logging on to a Windows 2000 domain using a smart card. It requires Active Directory. Assuming sufficient disk space, we generally recommend you install an Enterprise Root CA to issue certificates to Subordinate CAs. This enhances the security of certificates. If there is no CA inside the enterprise, you must install an Enterprise Root CA.

**Enterprise Subordinate CA:** If there is a Root CA already in the enterprise, you should install Enterprise Subordinate CA to issue the certificates to each entity in the enterprise. It requires Active Directory as well.

**Stand-Alone Root CA:** You should install a Stand-Alone Root CA if you will issue certificates outside of the enterprise network. For example, you can issue certificates to your customers so they can access your Web site from outside. A Stand-Alone CA is the root of a CA trust hierarchy.

*Note: No certificates can be issued for logging on to a Windows 2000 domain using smart cards. Do not install a Stand-Alone CA if you want to use smart card logon.*

A Stand-Alone Subordinate CA is one that operates with an existing CA trust hierarchy. This could be an external, commercial CA or a Stand-Alone Root CA. You should set up a Stand-Alone Subordinate CA when you will issue certificates to entities outside a corporation.

Here we install Enterprise CA.

The Certificate Services use the default encryption setting to provide the security policy. If you need to configure the advanced setting of CA (such as CSP, digital signature, the

Hash function for information integrity, the length of key size and key types, etc.), enable **Advanced options**. Then click **Next** to set Public/Private Key.

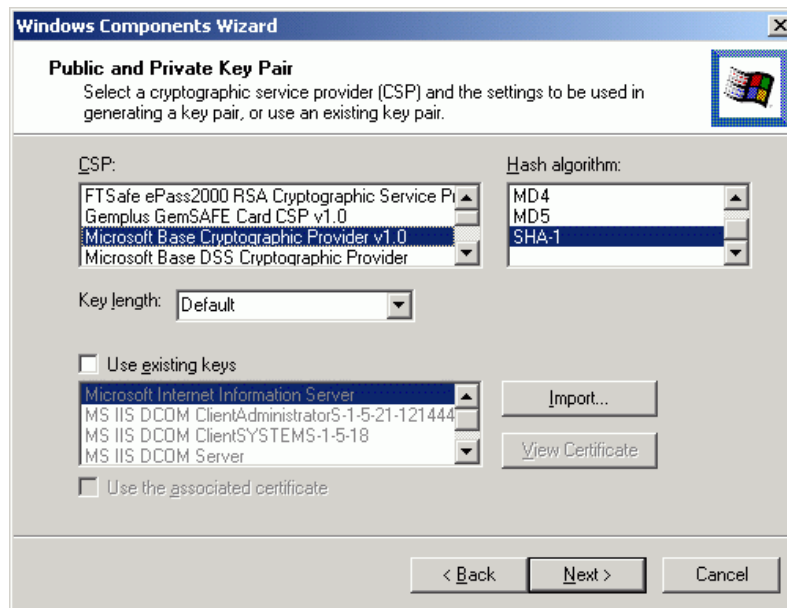


Figure 4-4 Key Pair Advanced Options

You can change the default encryption function. The listed encryption functions are supported by the software and hardware of your computer.

You can change the key length from the option list. In general, the longer the key, the more secure the data. We recommend that users select the longest key length possible. Some hardware will not support longer key lengths. If you want to use the existing keys in the system, choose **Use existing keys** and click the **Import** button.

Click **Next** to continue.

6. Give the appropriate information in the CA Identifying Information screen:

Input a name for the CA at **CA name** field. You will use this name later, when you identify the CA to the Certificate Services.

The CA name will identify the Enterprise CA to Active directory and any stand-alone CA. The expiration time for a root CA should be greater than that for any subordinate CA. The administrator should take into account the work required to manage the system when setting the expiration time.

**Windows Components Wizard**

**CA Identifying Information**  
Enter information to identify this CA

CA name: TestCA

Organization: ePass Management Department

Organizational unit: Ftsafe

City: Beijing

State or province: Country/region: CN

E-mail: Ftsafe@ftsaf.com

CA description: Test Certificate

Valid for: 2 Years Expires: 6/4/2004 1:53 PM

< Back Next > Cancel

Figure 4-5 CA Identifying Information

7. Specify the storage location of certificate database, certificate services setting information, certificate revocation list and certificate database log file.

**Windows Components Wizard**

**Data Storage Location**  
Specify the storage location for the configuration data, database and log

Certificate database: D:\WINNT\System32\CertLog Browse...

Certificate database log: D:\WINNT\System32\CertLog Browse...

☐ Store configuration information in a shared folder  
Shared folder: Browse...

☐ Preserve existing certificate database

< Back Next > Cancel

Figure 4-6 Certificate data storage locations

For Enterprise CAs, some setting and attribute information is kept on the domain controller.

If you did not set Certificate Services in the domain controller computer, choose the **shared folder** option and input a shared folder path to specify the storage place for the CA (users can specify a shared folder – this way a client outside of the domain can access the certificate revocation list).

Click **Next** to continue.

8. If you are installing a subordinate CA, the wizard prompts you for information about how you will request the certificate. Click **Browse** to locate an online CA, or select **Save the request to a file** if you make a request destined for a commercial CA or a CA that is not accessible from the network. (If you create a file, you must take the file to a CA for processing. The CA provides you with a certificate, which you install using the MMC snap-in.).

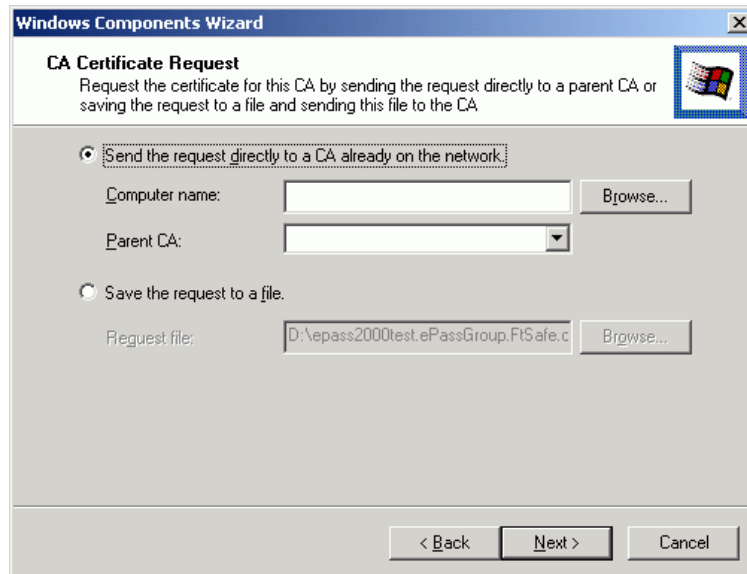


Figure 4-7 CA Certificate Request

Click **Next** to continue.

9. If IIS is running, a message will prompt you to stop the service. Click **OK** to stop IIS. You must stop IIS to install the Web components. If you do not have IIS installed, you will not see this message.

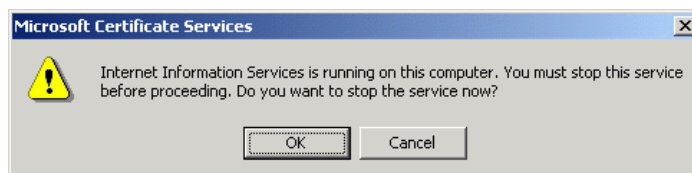


Figure 4-8 IIS warning message

10. The system begins to install the related components and programs

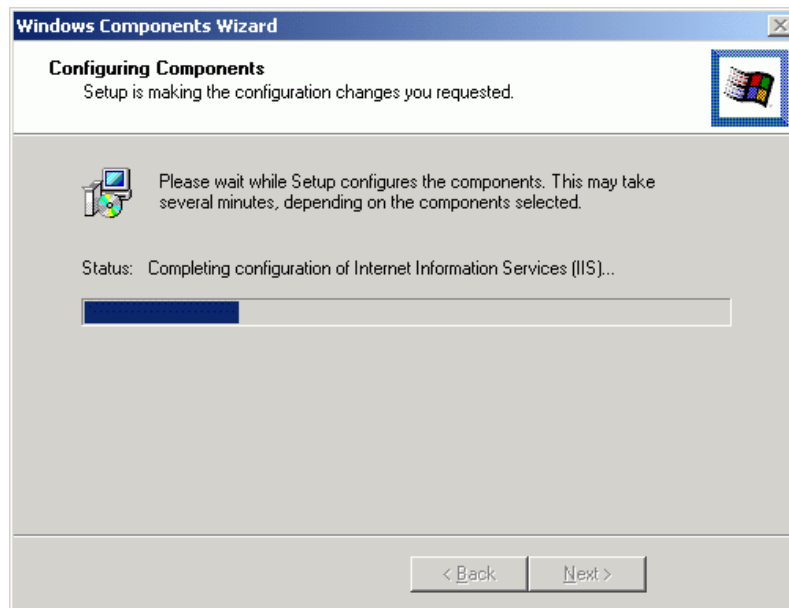


Figure 4-9 IIS certificate services install

11. The file folder **%SystemRoot%\system32\CerSrv\CertEnroll** is shared to allow the clients to access the information under this directory and check revocation list. If this folder is not shared, the client computer may not work properly.
12. Once Certificate Services has been successfully installed, you may start and manage Certificate Services from **Start Menu->Program->Management Tool->CA**.

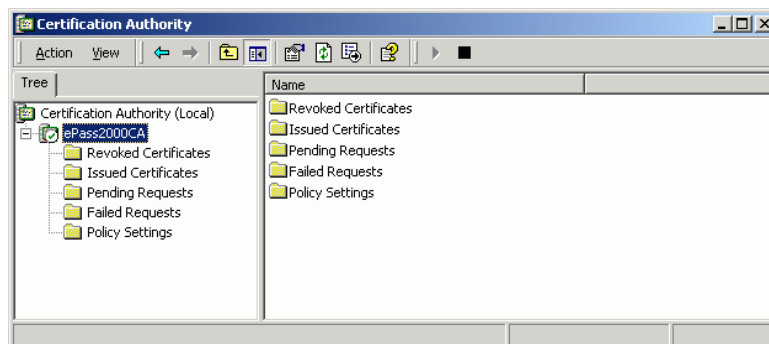


Figure 4-10 CA management tool

#### 4.1.2 Install Root Certificate

For Windows users, computers, and services, trust in a Certificate Authority is established when a copy of the root certificate is stored in the trusted root Certificate Authority's database. After installing the root certificate, you can issue the certificate to other entities to build a trust hierarchy. The following steps describe how to install root certificate:

1. Install the Enterprise Certificate Authority
2. By default, every CA that is hosted on a Windows 2000 server has Web pages available for users and administrators to request certificates. These Web pages are located at <http://servername/certsrv> where *servername* is the name of the Windows2000 server that hosts

the CA. In our example, the web page is <http://epass2000test/certsrv>

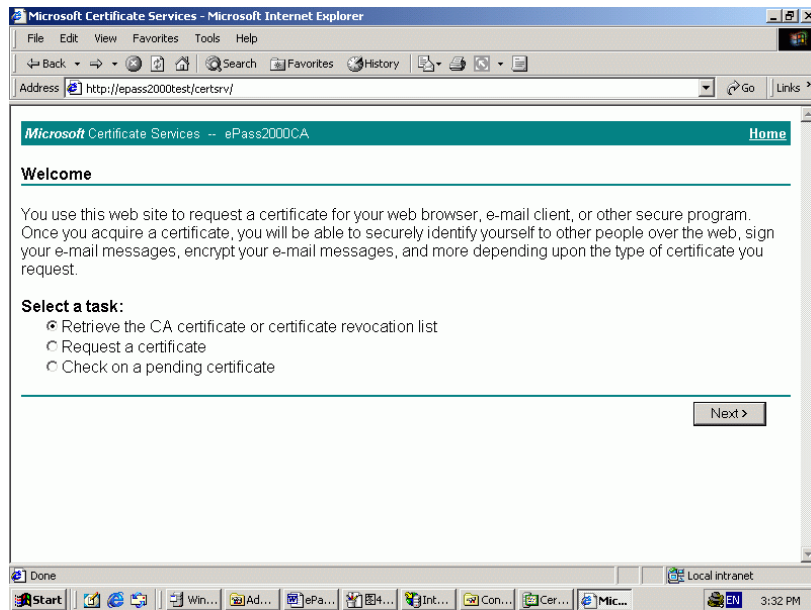


Figure 4-11 CA web site

3. Click **Retrieve the CA certificate or certificate revocation list**, and then click **Next**.

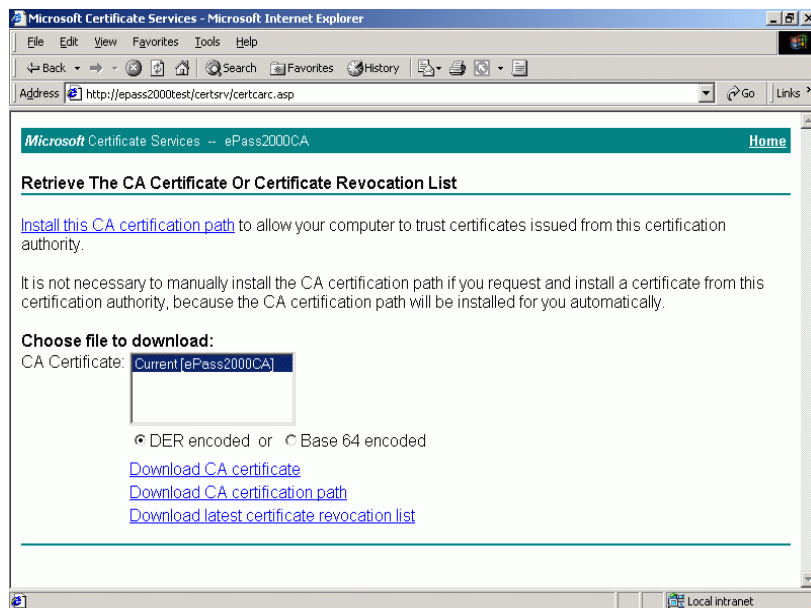


Figure 4-12 Download CA

4. If you want to trust **all** the certificates issued by this CA, click **Install this CA certification path**

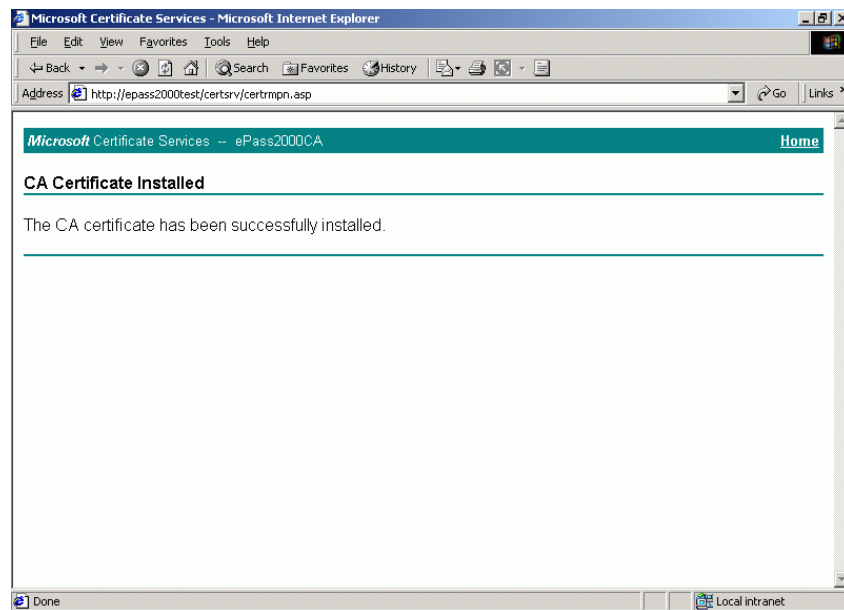


Figure 4-13 Automatically install CA certificate

5. If the Certificate Authority has been renewed, you will have the choice of which version of the CA certificate you want to download. Under **Choose file to download**, click the CA certificate you want to download, and then click **Download CA Certificate**.

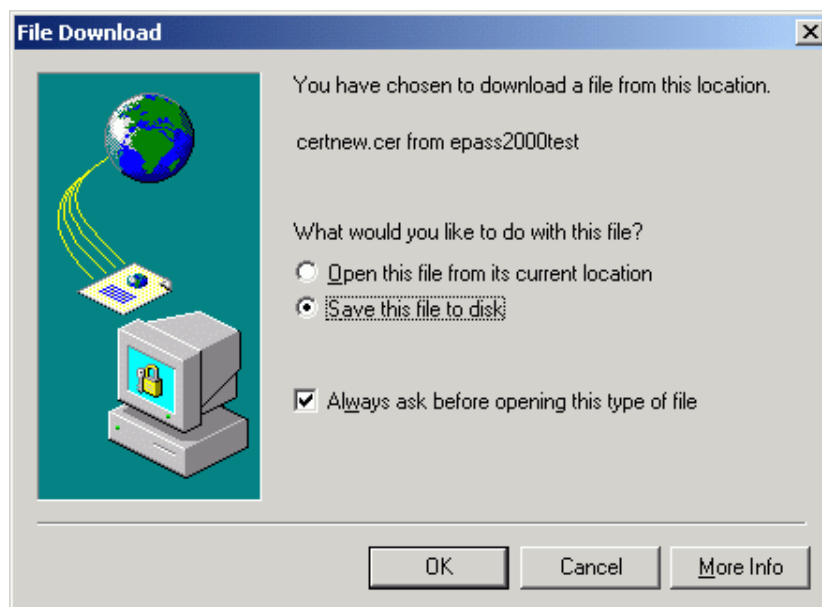


Figure 4-14 Save CA to hard disk

6. In **File Download**, click **Open this file from its current location** if want to review this file *certnew.cer*, and then click **OK**

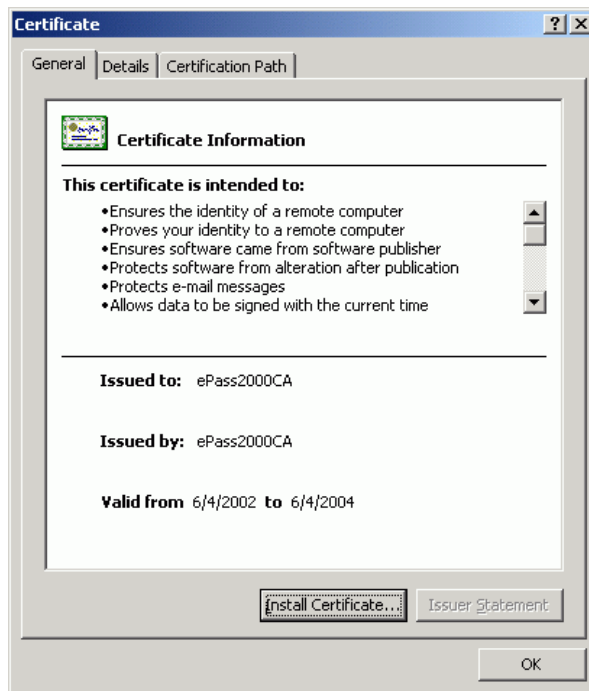


Figure 4-15 View certificate

1. When the **Certificate** dialog box appears, click **Install certificate ....**

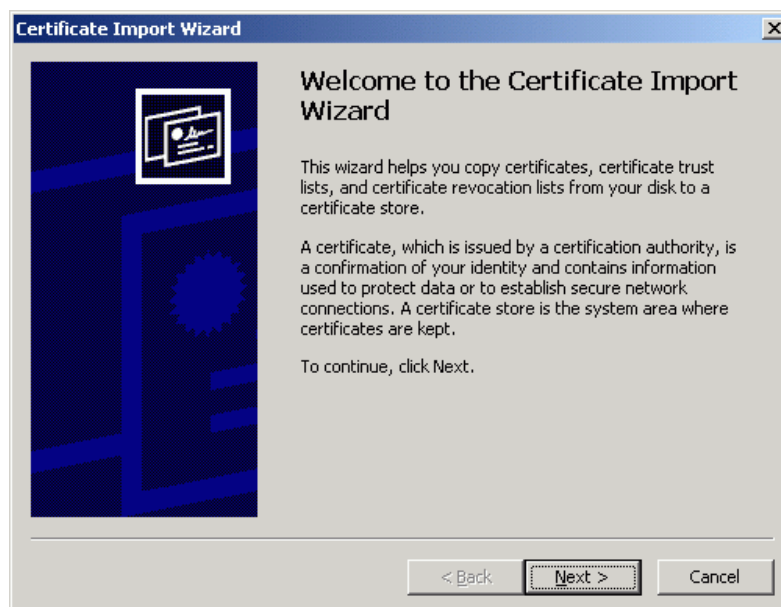


Figure 4-16 Import certificate wizard

In the Certificate Import wizard, follow the instruction to finish the installation.

## 4.2 Download and Install a Digital ID in ePass2000

Insert an ePass2000 token into the USB port. Open <http://YourWebServerName/certsrv/> on a web

browser.

1. Select **Request a certificate**, click **Next**
2. Select **Advanced request**, click **Next**.
3. Select **Submit a certificate request to this CA using a form**, and click **Next**.
4. Fill in the information fields. The user's e-mail address must be entered. Select **FTsafe ePass2000 RSA Cryptographic Service Provider** in CSP under Key Options. Click **Submit** to continue.

The screenshot shows a web browser window titled "Microsoft Certificate Services - Microsoft Internet Explorer". The address bar shows "http://192.168.0.112/certsrv/certreqma.asp". The page content is titled "Microsoft Certificate Services -- Test CA" and "Advanced Certificate Request". Under "Certificate Template:", there is a dropdown menu set to "User". Under "Key Options:", the "CSP:" dropdown is set to "FTsafe ePass2000 RSA Cryptographic Service Provider". The "Key Usage:" section has three radio buttons: "Exchange", "Signature", and "Both", with "Both" selected. The "Key Size:" is set to "1024", with "Min:1024" and "Max:1024" displayed. Below these are several checkboxes: "Create new key set" (selected), "Set the container name", "Use existing key set", "Enable strong private key protection", and "Mark keys as exportable".

Figure 4-17 Certificate request form

*Note: ePass2000 only supports 1024-bit keys.*

5. When prompted, type the user PIN associated with the ePass2000 token. Click **Login**.

The certificate request is pending the configuration of the CA server. Repeat section 4.2 to install the certificate to the token. The user can view the certificate with the Certificate Manager. At this point the user should click **OK** and see the secure web page.

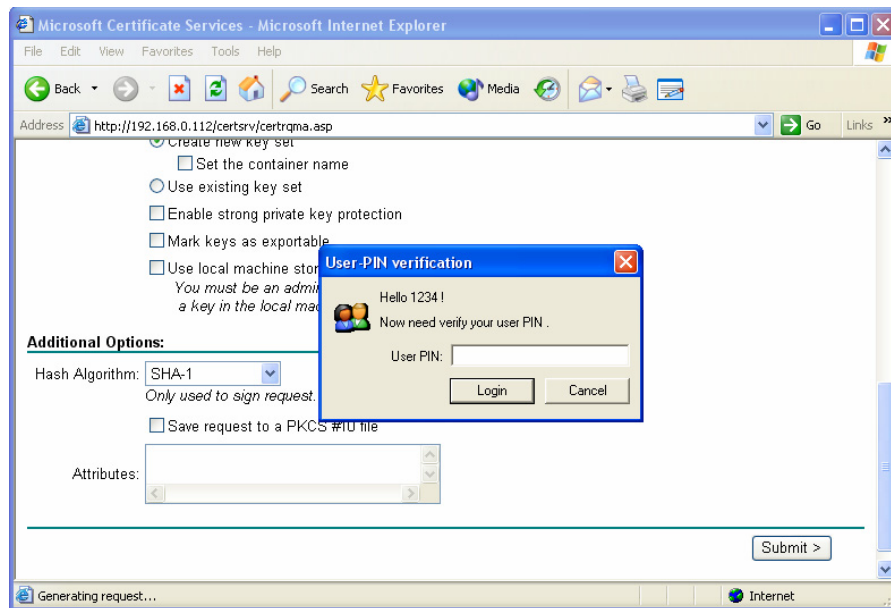


Figure 4-18 Enter the user PIN

Here we can view the new certificate we just requested in IE's Internet Options > Content > Certificates.

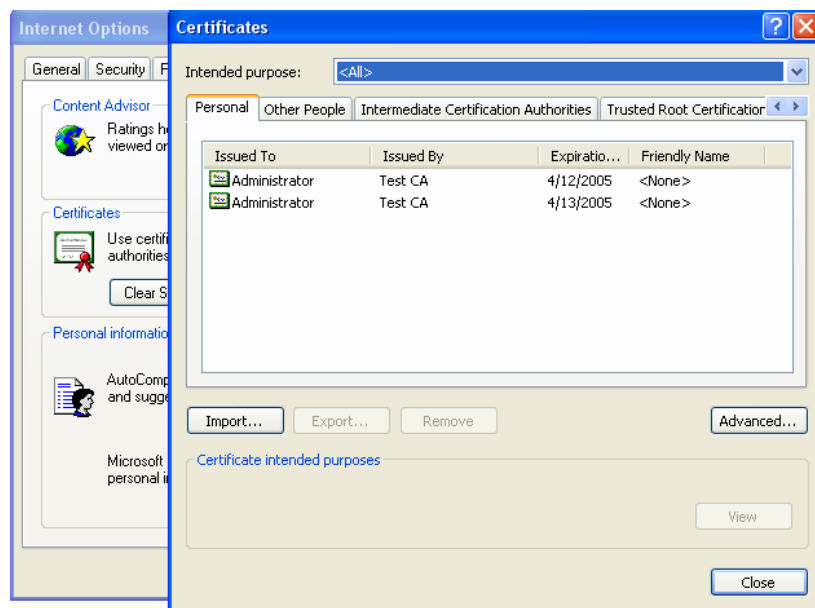


Figure 4-19 View certificates in IE

### 4.3 Configure SSL Encrypted Web

Internet Information Services (IIS) is installed on Windows2000 Server by default. It provides several Internet services, such as WWW, FTP, Gopher, etc. You can install IIS by using the **Add/Remove Programs** application in **Control Panel**. When the Windows Components wizard starts, select **IIS**. Then follow the on-screen instructions to install.

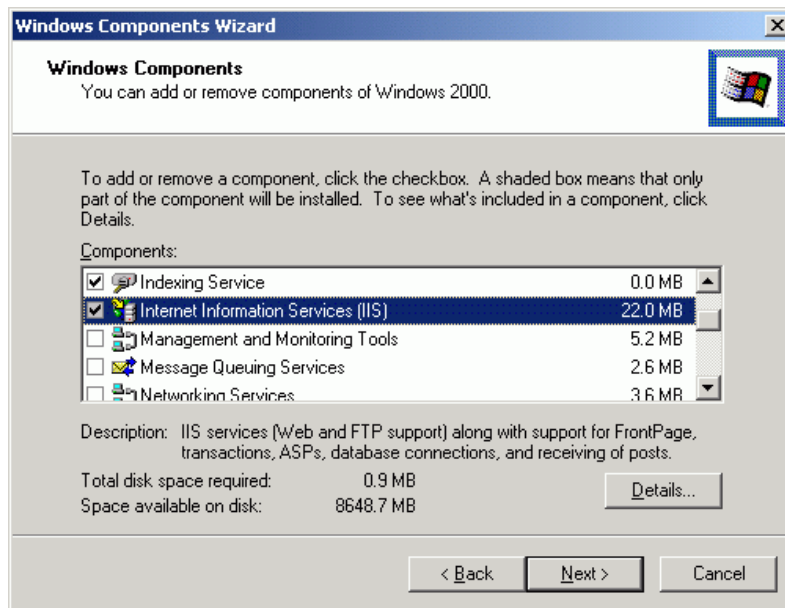


Figure 4-20 Install IIS

If IIS has already been installed and activated, you can run the IIS tool from **Start Menu -> Program -> Managing Tool -> Internet service managing tool.**

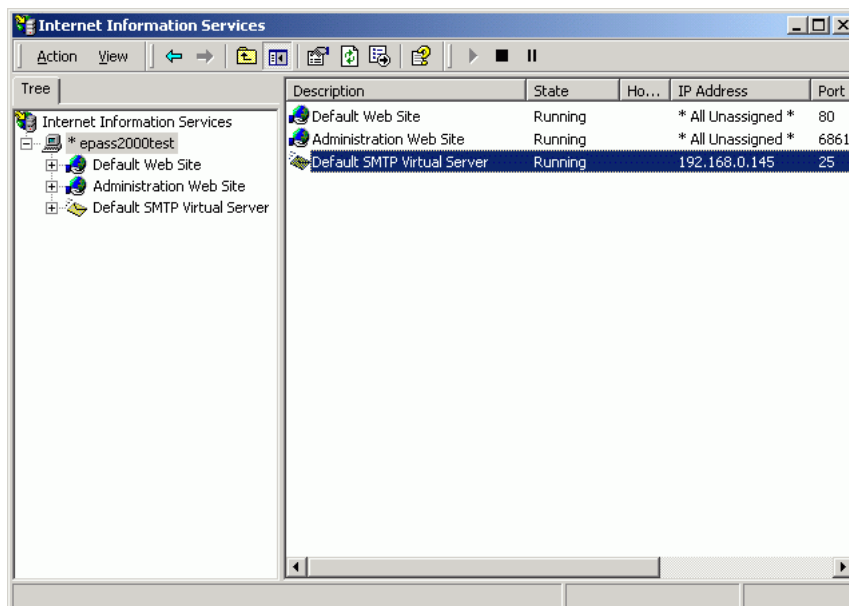


Figure 4-21 IIS management tool

#### 4.3.1 Configuring IIS

1. From **Start -> Programs -> Administrative Tools -> Internet Services Manager** to start IIS service managing tool.
2. Expand the domain node in the console. Then right click on it. Select **Properties** on the submenu as in Figure below.

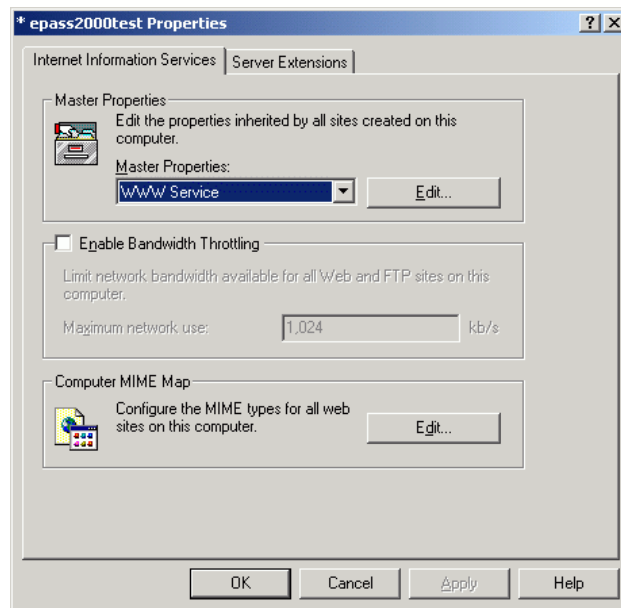


Figure 4-22 Configure IIS

3. Select **WWW Service** for Master Properties, then click **Edit**
4. The **WWW Service Master Properties for epass2000test** screen starts. Click the **Directory Security** tab.

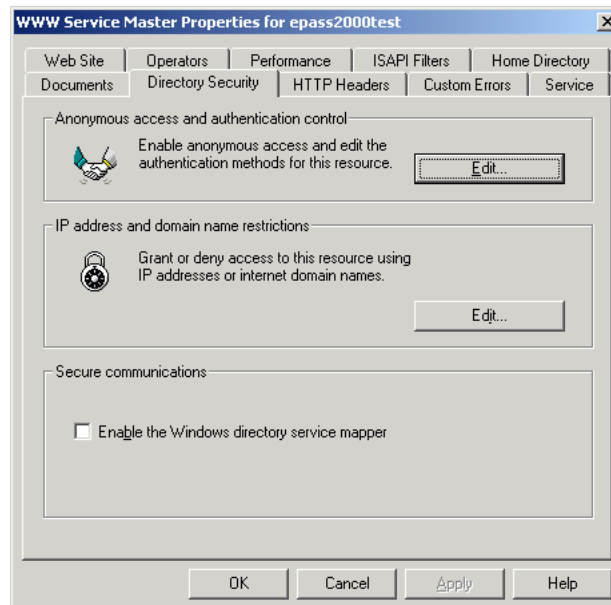


Figure 4-23 Directory security tab

5. Click **Enable the Windows directory service mapper** for Secure communications
6. Go back to the console, right click **Default Web Site**. Click **Properties**.

7. The **Default Web Site Properties** screen starts. Click the **Directory Security** tab.

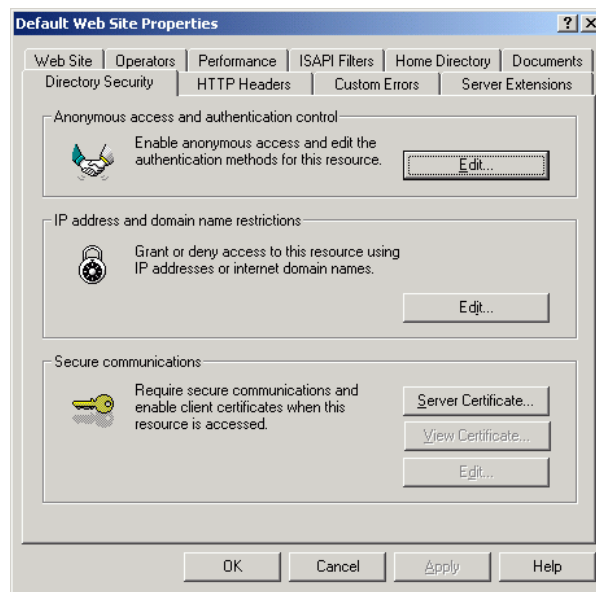


Figure 4-24 Set server certificate

*Note: The **Edit** button under **Secure communications** is unavailable until you request a Web server certificate*

8. Click the **Server Certificate** button. The **Web server Certificate Wizard** starts. Click **Next**



Figure 4-25 Wizard of server certificate

9. Select the **Create a New certificate** option, and click **Next**. You will see a different dialog box if IIS already has a certificate

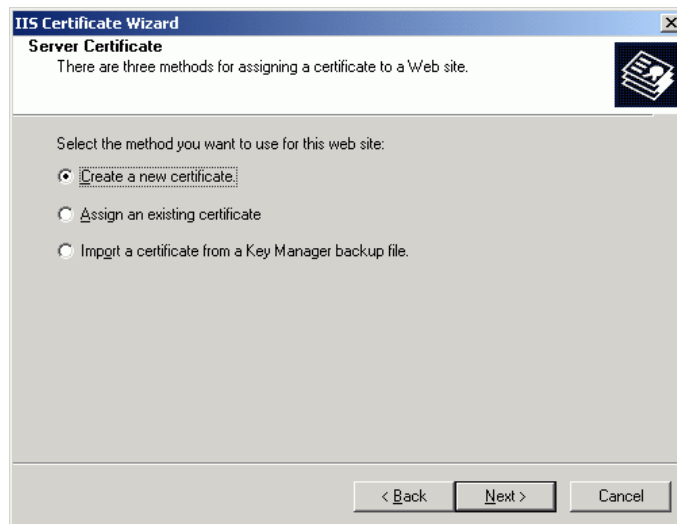


Figure 4-26 Create a new server certificate

10. Select the **Send the request immediately to an online Certificate Authority** option (This assumes that you have an enterprise CA in your domain that is configured to issue Web certificates). Click **Next**.
11. In the **Name and Security Settings** dialog box, set **Bit Length** to 1024. Click **Next**.

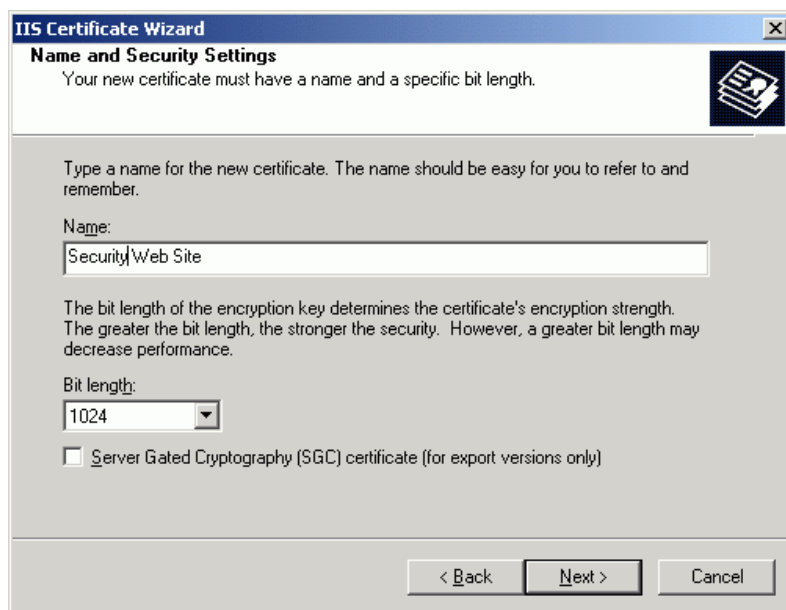
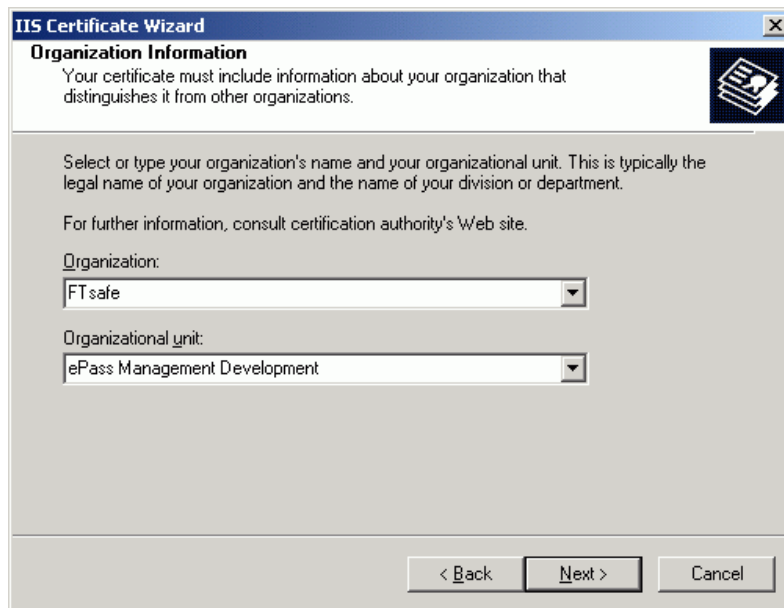


Figure 4-27 Name and security settings

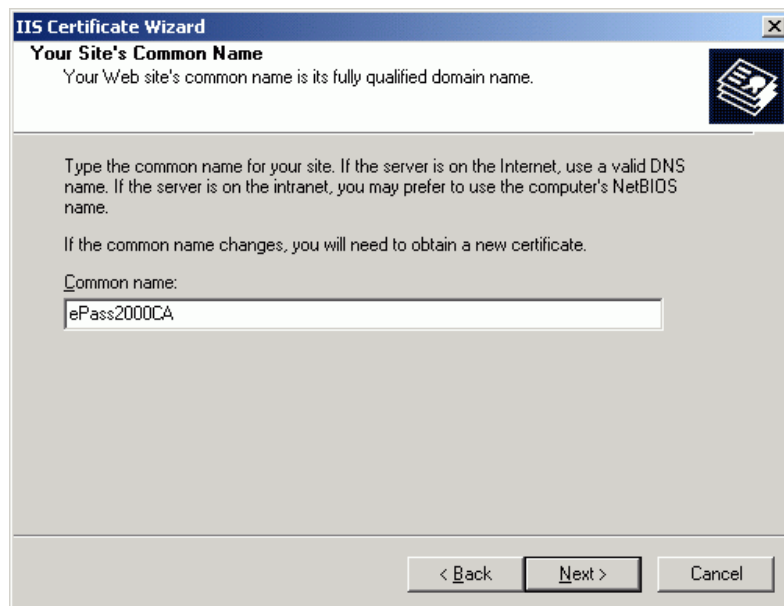
12. On the next page, enter your organization information, and click **Next**.



The screenshot shows the 'IIS Certificate Wizard' window with the 'Organization Information' tab selected. The window title is 'IIS Certificate Wizard'. The tab is 'Organization Information'. The text says: 'Your certificate must include information about your organization that distinguishes it from other organizations.' Below this, it says: 'Select or type your organization's name and your organizational unit. This is typically the legal name of your organization and the name of your division or department.' Then: 'For further information, consult certification authority's Web site.' There are two dropdown menus: 'Organization:' with 'FT safe' selected, and 'Organizational unit:' with 'ePass Management Development' selected. At the bottom are three buttons: '< Back', 'Next >', and 'Cancel'.

Figure 4-28 Organization information

13. Enter a valid name to your web site, click **Next**.



The screenshot shows the 'IIS Certificate Wizard' window with the 'Your Site's Common Name' tab selected. The window title is 'IIS Certificate Wizard'. The tab is 'Your Site's Common Name'. The text says: 'Your Web site's common name is its fully qualified domain name.' Below this, it says: 'Type the common name for your site. If the server is on the Internet, use a valid DNS name. If the server is on the intranet, you may prefer to use the computer's NetBIOS name.' Then: 'If the common name changes, you will need to obtain a new certificate.' There is a text input field labeled 'Common name:' with 'ePass2000CA' entered. At the bottom are three buttons: '< Back', 'Next >', and 'Cancel'.

Figure 4-29 Give a common name

14. Enter your location information, and click **Next**.

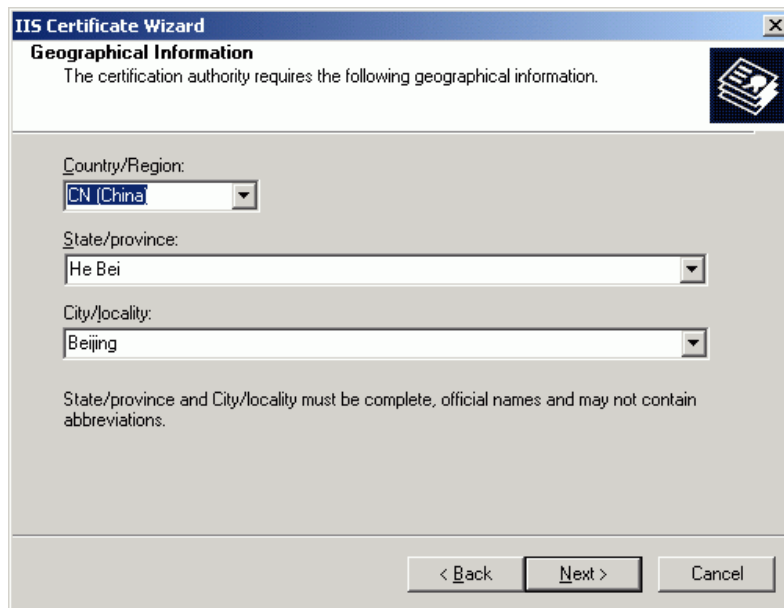


Figure 4-30 Geographical information

15. Save the file on your computer in a proper place. Click **Next**.

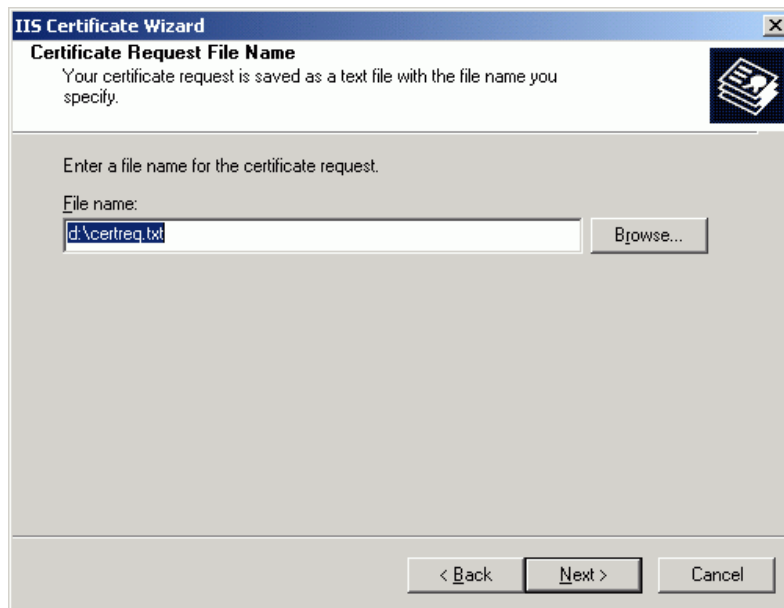


Figure 4-31 Location of request file

16. Check the information that you have entered, and then click **Next** to complete the process and create the certificate request

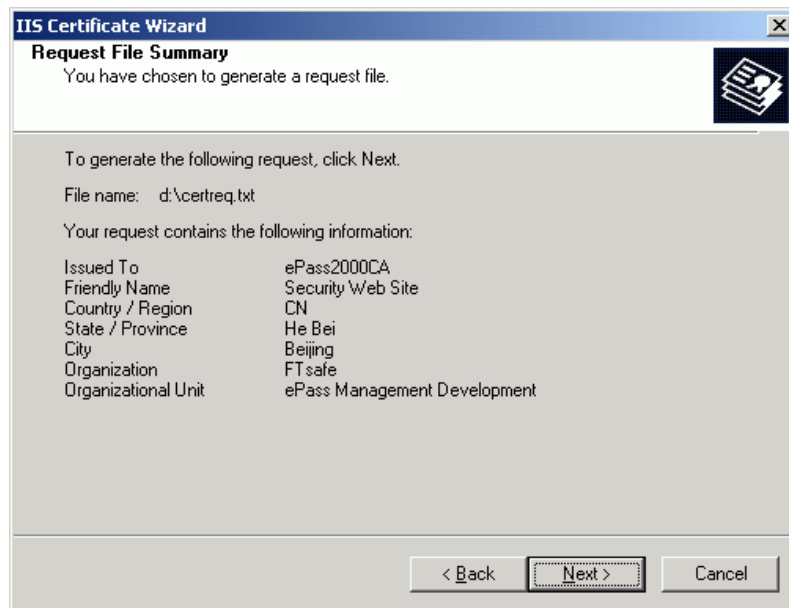


Figure 4-32 Review information of request

17. Open a browser and go to <http://YourWebServerName/certsrv/>. Select **Request a Certificate**. Click **Next**.

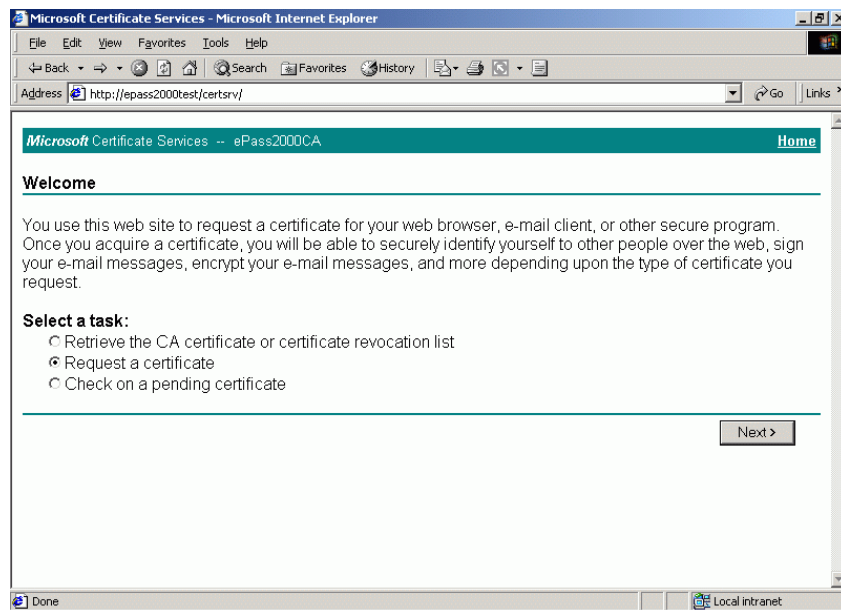


Figure 4-33 Send request from CA web site

18. Select **Advanced Request**. Click **Next**.

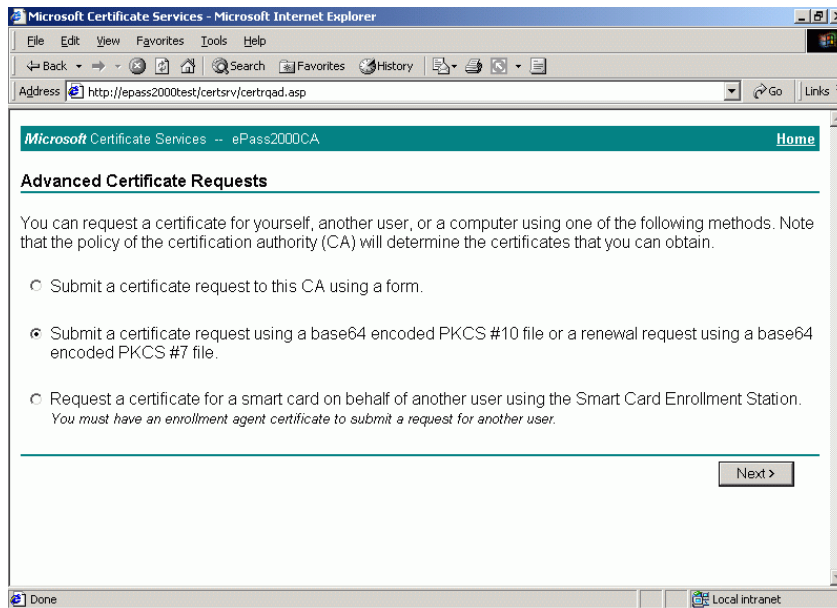


Figure 4-34 Advanced certificate request web page

19. Select **Submit a Certificate Request using a Base64 encoded PKCS#10 file or a renewal request using a Based64 encoded PKCS#7 file**, and click **Next**
20. Open the request file that you created in the former step, and copy the contents of the document and paste into the Web form's **Base64 Encoded Certificate Request** text box:

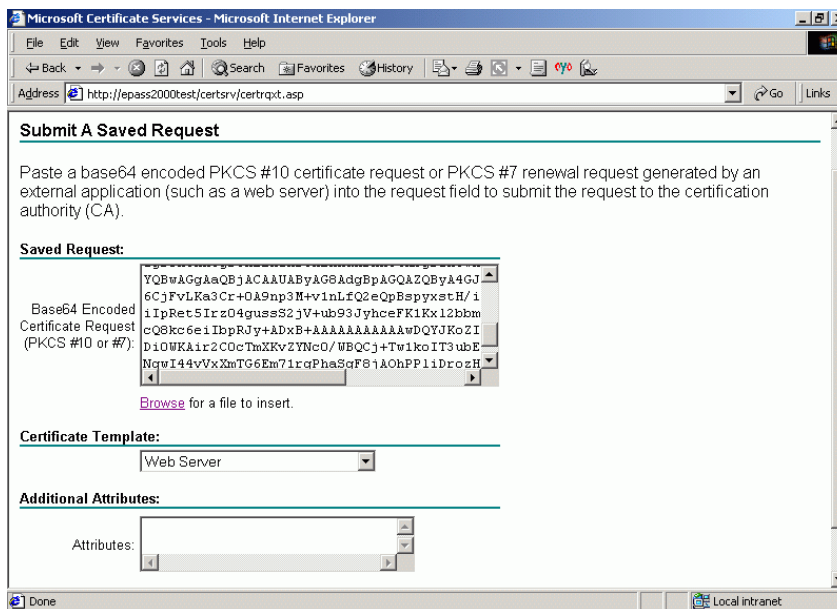


Figure 4-35 Paste request

21. The certificate request is pending. The applicant should wait for the Administrator to issue the certificate.

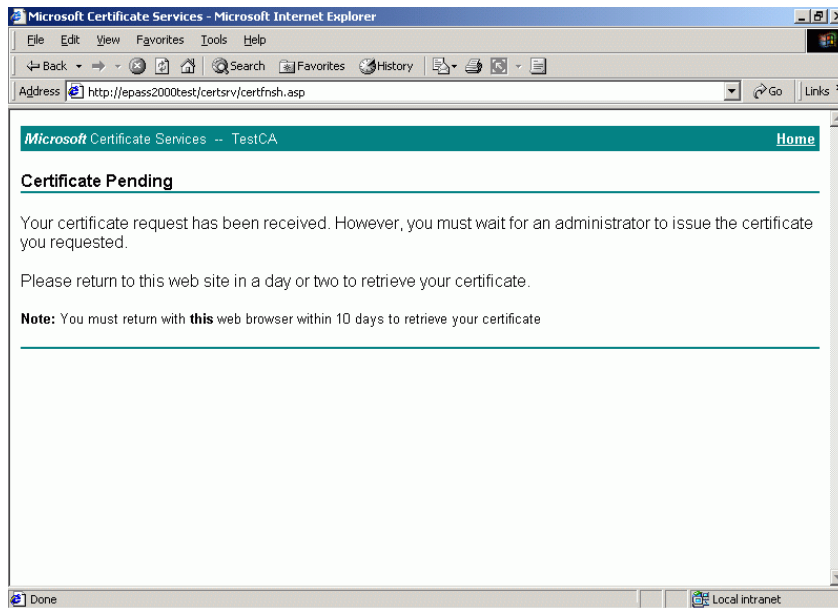


Figure 4-36 Certificate request pending

#### 4.3.2 To Issue and Download a Certificate

To issue a certificate in Certificate Server, follow these steps:

1. Click **Start -> Programs -> Administrative Tools**, and then click **Certificate Authority**.
2. Expand Certificate Authority. Click the Pending **Requests** folder.
3. The pending certificate requests appear in the right pane right-click the pending certificate request (submitted in the former procedure described in this chapter), select **All Tasks**, and then click **Issue**
4. After you have issued the certificate, go to the Certificate Servers Web Site <http://YourWebServerName/certsrv/> again. Select **Check on a pending certificate**, and then click **Next**

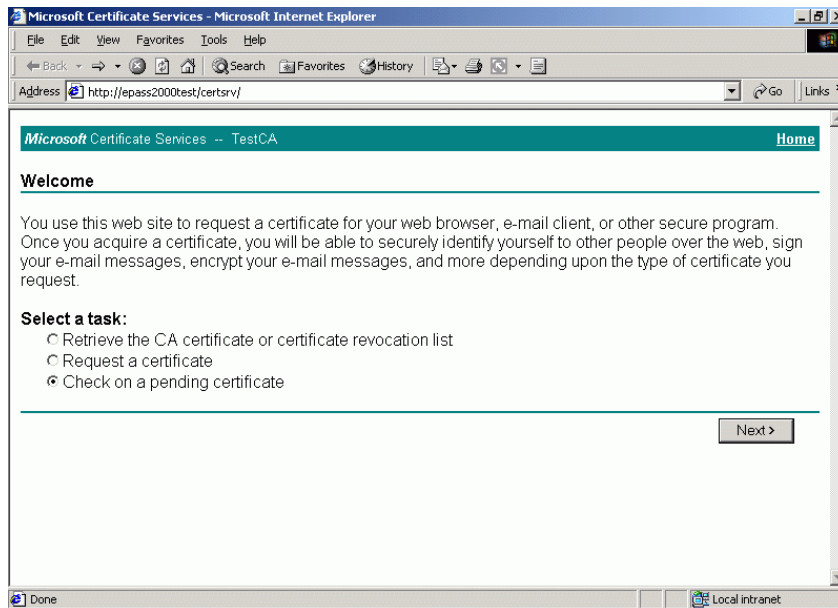


Figure 4-37 Check pending status

5. Select the pending certificate, and then click **Next** to go to the download page.

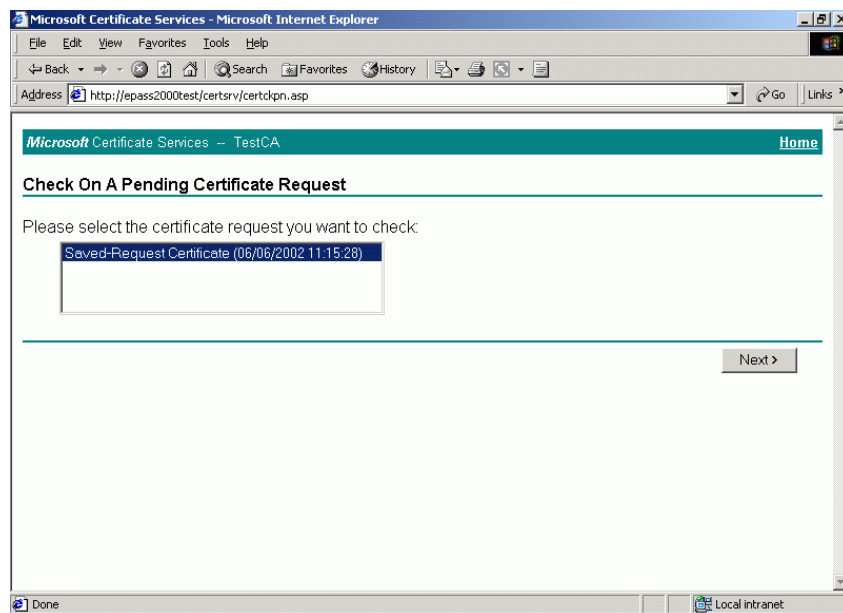


Figure 4-38 Certificate issued

6. On the download page, click the top hyperlink **Download CA Certificate**.

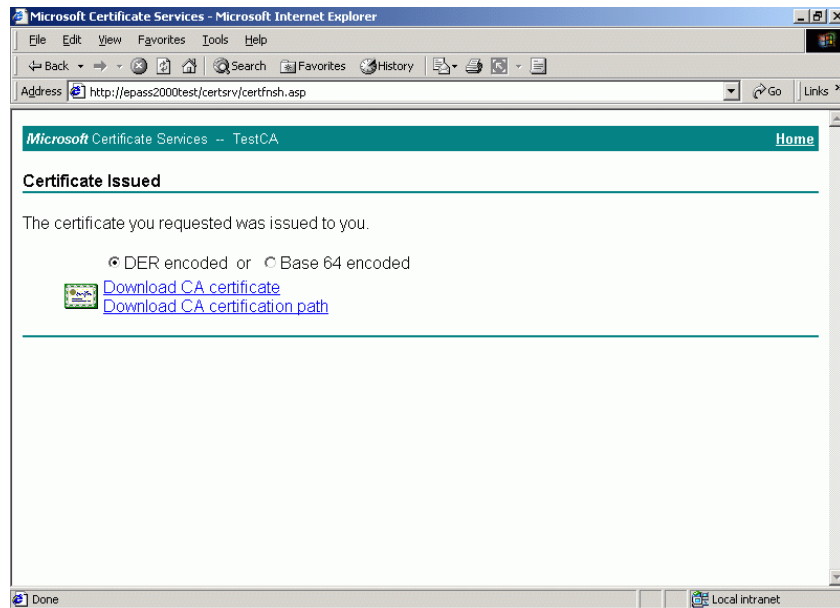


Figure 4-39 Download certificate

7. When prompted, select **Save this file to disk** and save the certificate to your desktop. Then finish installing this certificate.

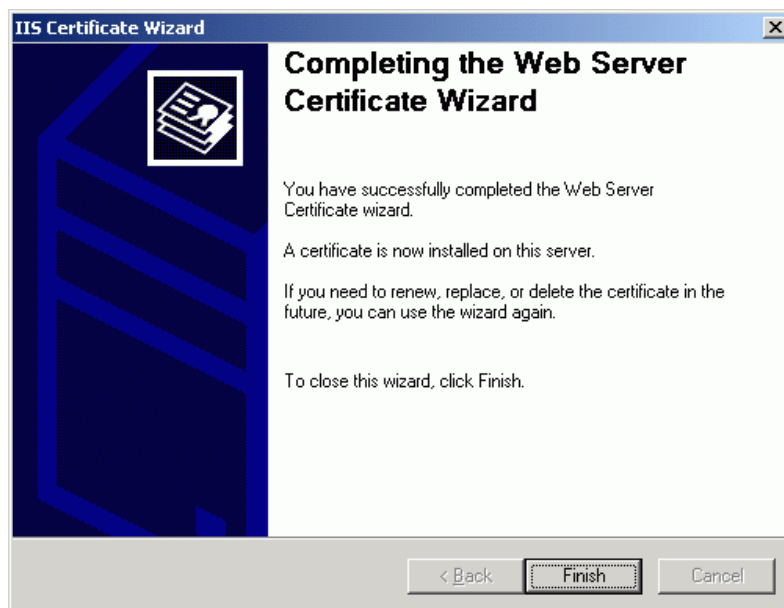


Figure 4-40 Completing server certificate wizard

If Certificate Server is set to **Always Issue the Certificate**, the certificate is issued immediately.

### 4.3.3 Setting up an SSL Web Site

You can configure your Web server's Secure Sockets Layer (SSL) security features to verify the integrity of your content, verify the identity of users, and encrypt network transmissions

In the Internet Information Services Manager, select the Web site that you want to protect with SSL

and open its property sheets. On the **Web Site** tab, under **Web Site Identification** assign SSL port to 443, the default port for secure communications.

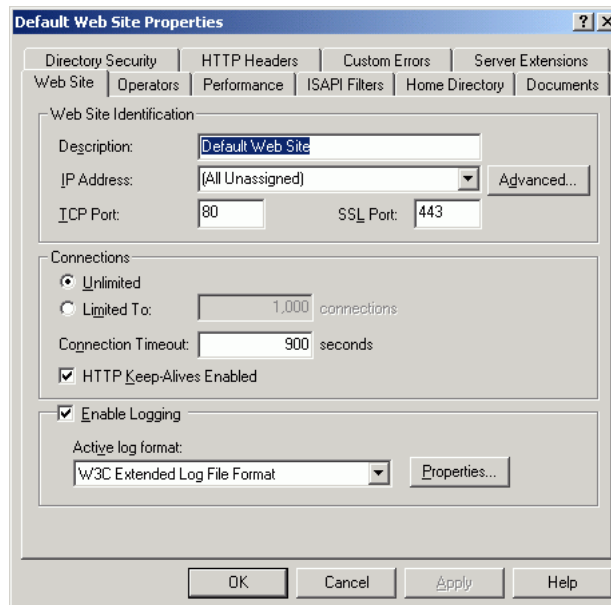


Figure 4-41 Set secure web site

Click the **Directory Security** tab. Under **Secure Communications**, the **Edit** and the **View Certificate** is now enabled. Now we can set up the proper IP address.

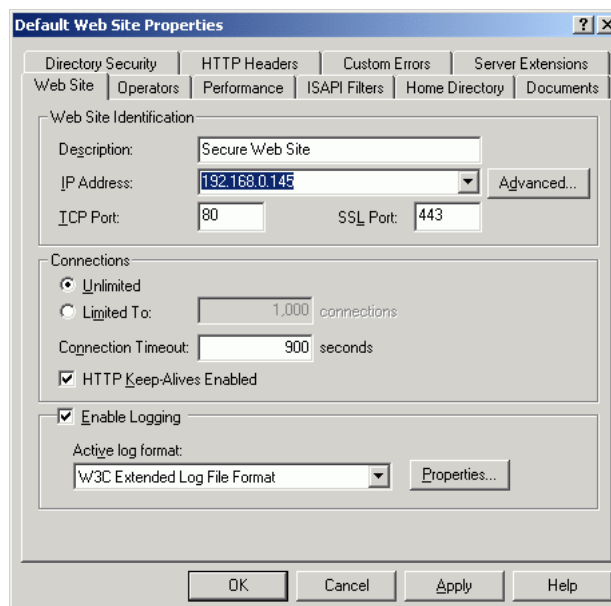


Figure 4-42 Setup IP

1. Click **Directory Security** tab:
2. Click **Edit** on the **Secure communications** dialog box.

3. Activate **Require secure channel (SSL)** and **Require client certificates**. Then click **OK**.

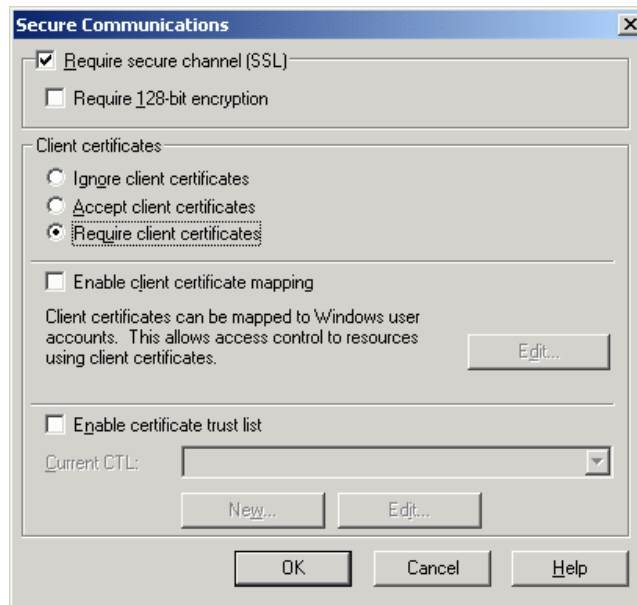


Figure 4-43 Configure secure communication

4. IIS expects you to load the web page with “https”. An error will be issued if you load the page with “http”.

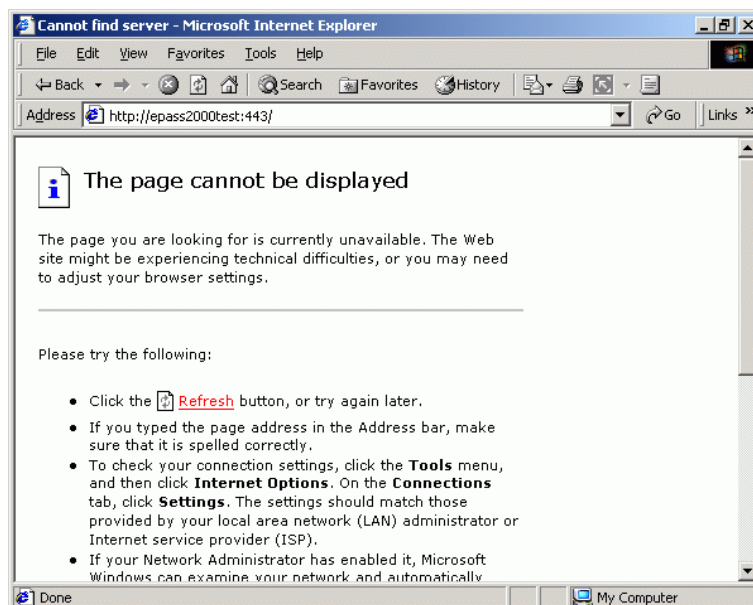


Figure 4-44 Load secure web page

If *https* was used to load the secure web page, you will be prompted with a warning message

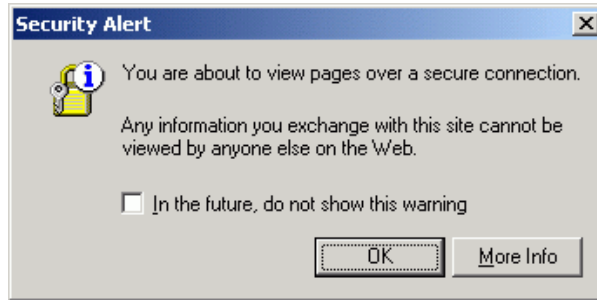


Figure 4-45 Security warning

Click **OK**. Select the client certificate from the window below and click **OK**. After the web server verifies the authentication, you can access the web page.

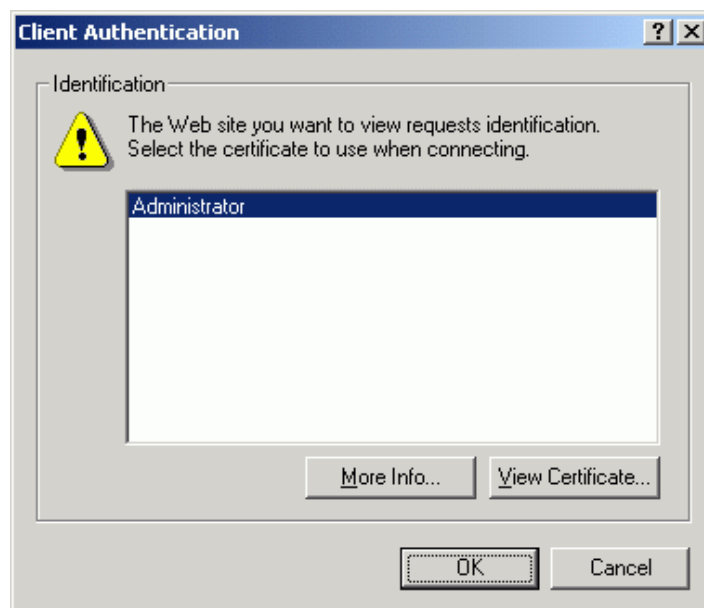


Figure 4-46 Authentication dialog box

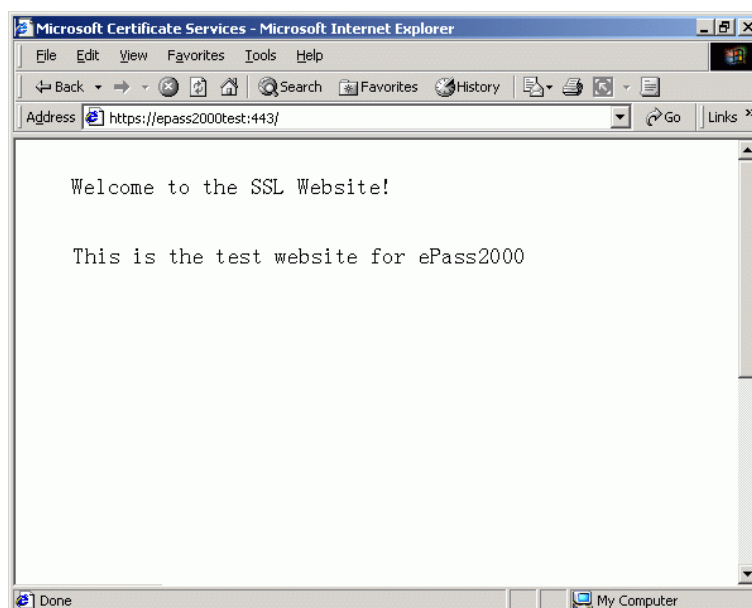


Figure 4-47 Access the SSL site

#### 4.4 Encrypting E-Mail with ePass2000

1. Open Outlook Express
2. Choose **Options** from the **Tools** menu.

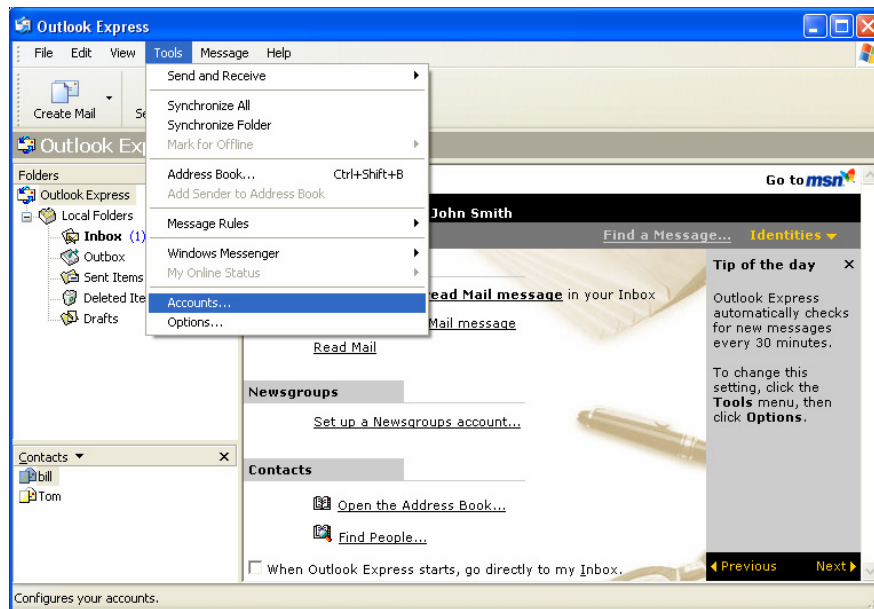


Figure 4-48 Start outlook express configuration

3. Click the **Security** tab:

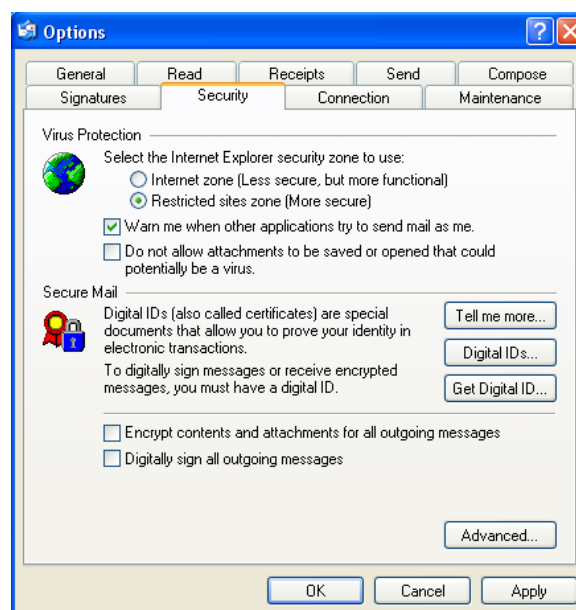


Figure4-49 Security tab

4. Click **Digital IDs** to get a list of web sites that can issue certificate to users.

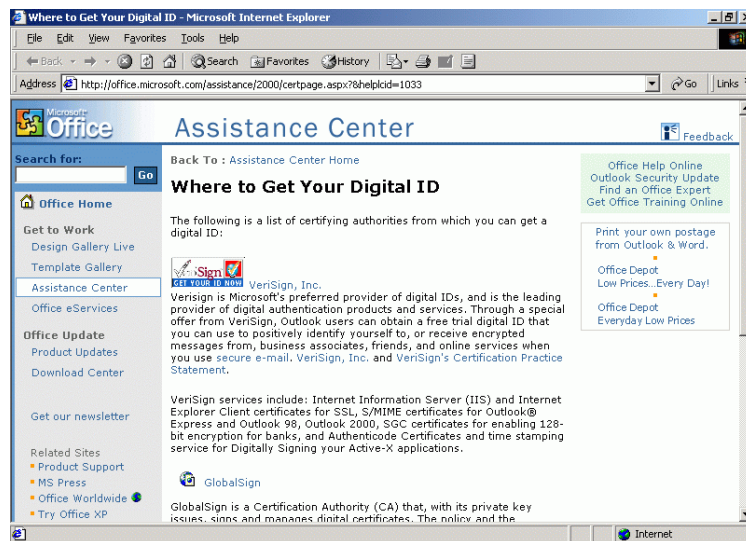


Figure 4-50 Digital ID center

In the following sections, we use the CA we configured earlier in section 4.4 to illustrate how to use ePass2000 to send an encrypted mail. First, we need download a certificate into ePass2000.

- 1) Access the CA's website, for example <http://YourWebServerName/certsrv/>.
- 2) Select **Request a certificate**

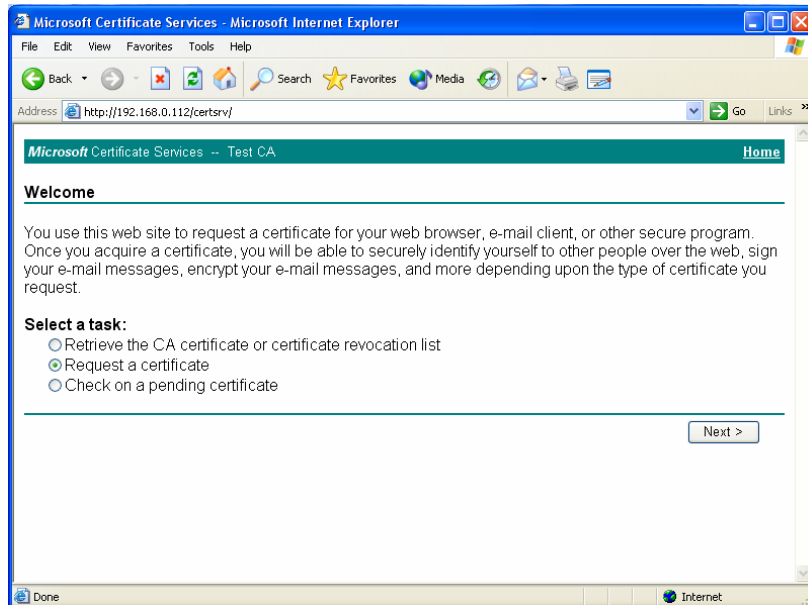


Figure 4-51 Request a certificate

- 3) Select **Advanced request**

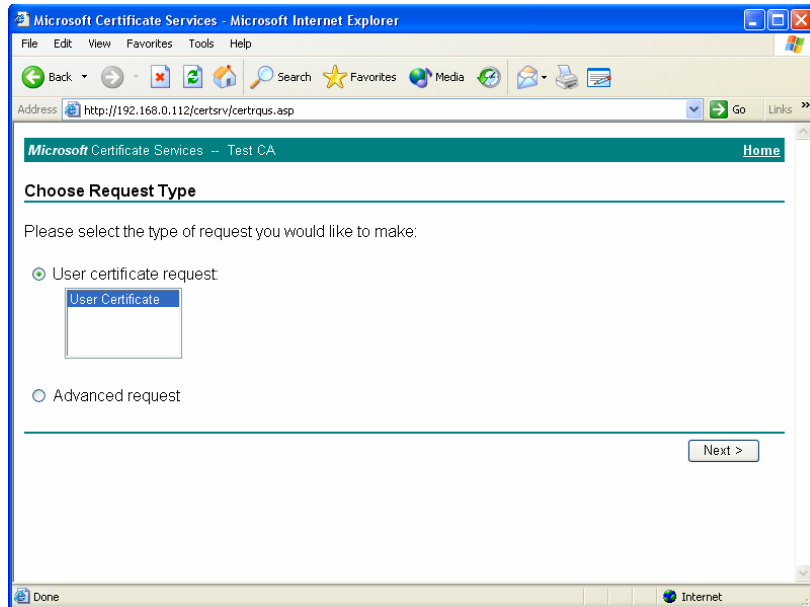


Figure 4-52 Advanced request

- 4) Select **Submit a certificate request to this CA using a form.**

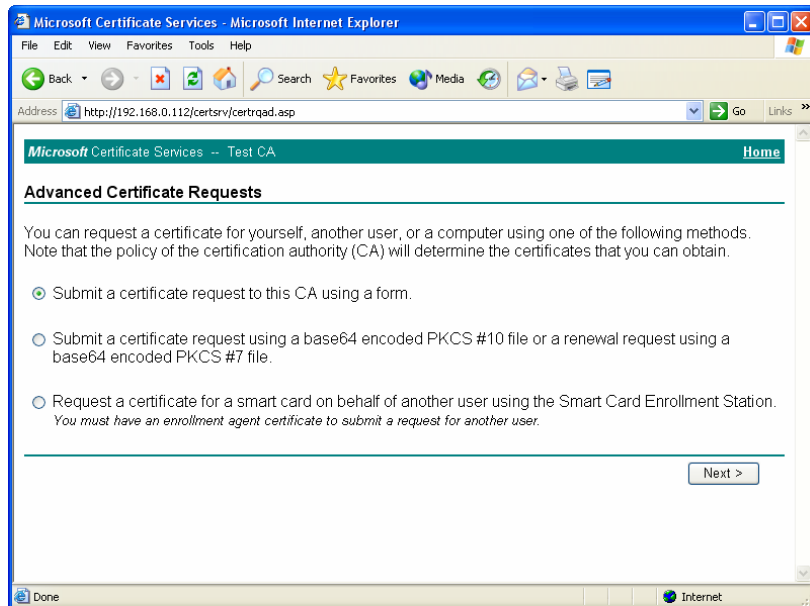


Figure 4-53 Online certificate request step 1

- 5) Fill the form and select **FTsafe ePass2000 RSA Cryptographic Service Provider** for the Cryptographic Service Provider Name.

The screenshot shows the 'Advanced Certificate Request' page in a Microsoft Internet Explorer window. The address bar shows 'http://192.168.0.112/certsrv/certrqma.asp'. The page has a green header with 'Microsoft Certificate Services -- Test CA' and a 'Home' link. The main content area is titled 'Advanced Certificate Request'. Under 'Certificate Template:', there is a dropdown menu set to 'User'. Under 'Key Options:', the 'CSP' is set to 'FTSafe ePass2000 RSA Cryptographic Service Provider'. 'Key Usage' has radio buttons for 'Exchange', 'Signature', and 'Both', with 'Both' selected. 'Key Size' is set to '1024' with a range from 'Min:1024' to 'Max:1024'. There are several checkboxes: 'Create new key set' (selected), 'Set the container name' (unchecked), 'Use existing key set' (unchecked), 'Enable strong private key protection' (unchecked), and 'Mark keys as exportable' (unchecked). The status bar at the bottom shows 'Done' and 'Internet'.

Figure 4-54 Request form

- 6) After entering the necessary information in the form, click **Accept**. When prompted with a new dialog box, type the user PIN associated with the ePass2000 (see Figure 4-44).

This screenshot shows the same 'Advanced Certificate Request' form as Figure 4-54, but with a 'User-PIN verification' dialog box overlaid. The dialog box has a title bar with a close button and contains the text 'Hello 1234 !' and 'Now need verify your user PIN .'. It has a 'User PIN:' text field and 'Login' and 'Cancel' buttons. In the background, the 'Additional Options:' section is visible, showing 'Hash Algorithm:' set to 'SHA-1' and a checkbox for 'Save request to a PKCS #10 file'. The status bar at the bottom shows 'Generating request...'.

Figure 4-55 Input user PIN

- 7) Click Login. A key pair is generated directly into the ePass token. This key pair may be used for all processes that require a public/private key pair.

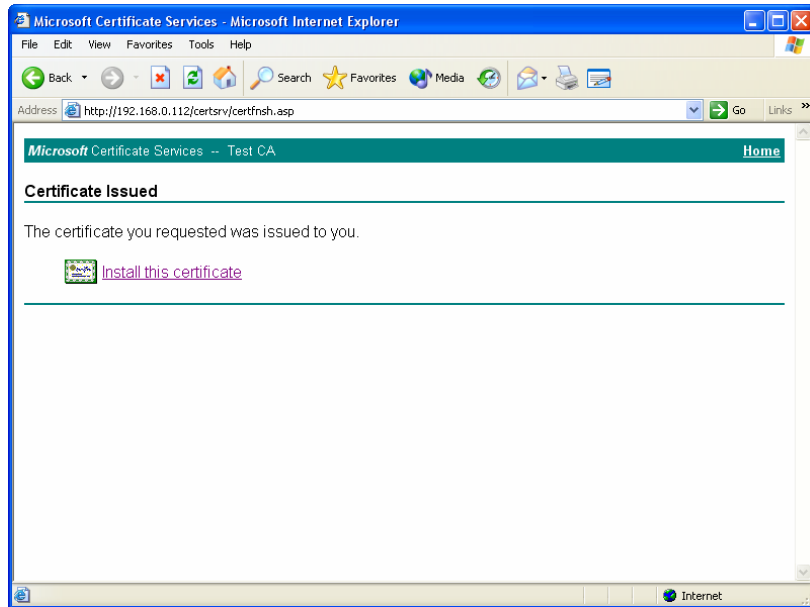


Figure 4-56 Certificate issued

*Notes:* 1024-bit keys need several minutes to generate. Internet Explorer may appear to “hang” during this time. Be patient and allow the key generation to continue.

- 8) Click Install. When prompted with a dialog box, type the user PIN associated with the ePass2000. The certificate is then installed to the ePass2000 token.

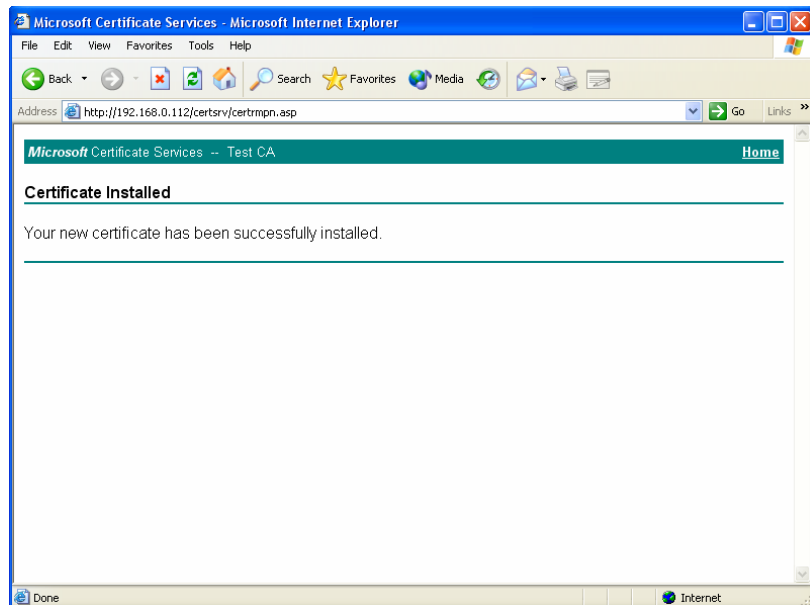


Figure 4-57 Install certificate

- 9) User can view their certificate from ePass2000 Certificate Manager

#### 4.4.1 To set security properties of an email account

Now that the user has a digital ID, secure email can be sent with Outlook Express. Before doing so, do the following:

1. Open Outlook Express
4. Choose **Accounts** from the **Tools** menu (see Figure 4-58):
5. Select **Mail** tab. Select the mail account which will use certificates and click **Properties**.

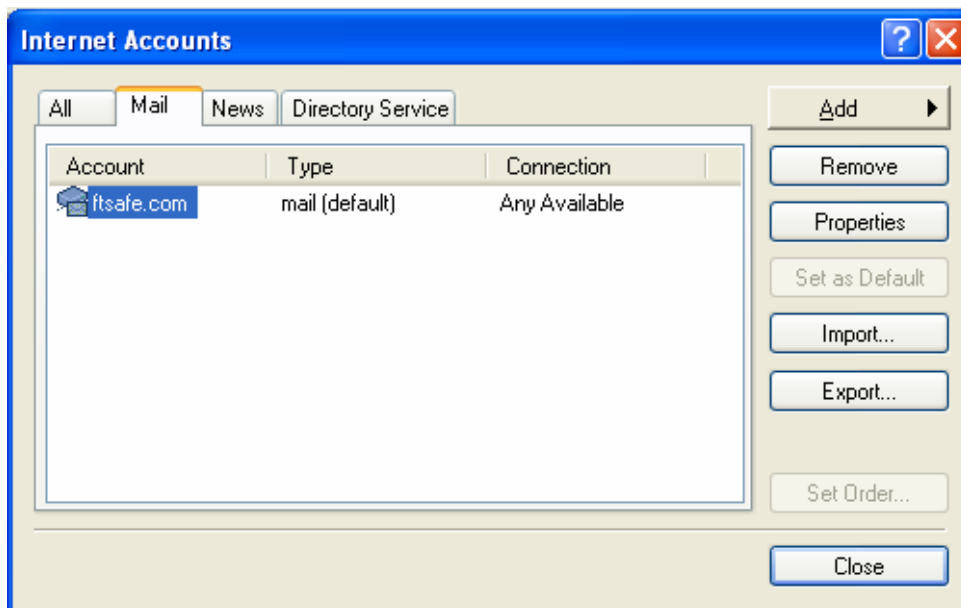


Figure 4-58 Setup property of email account

4. Click the **General** tab, to check if any information is not correct.

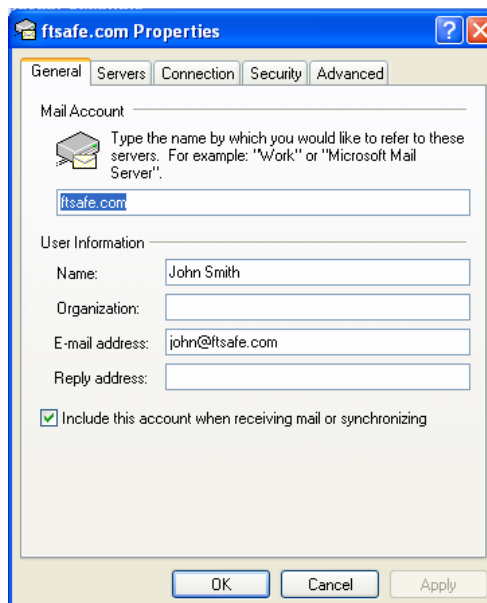


Figure 4-59 Check personal information

5. Click the **Security** tab.

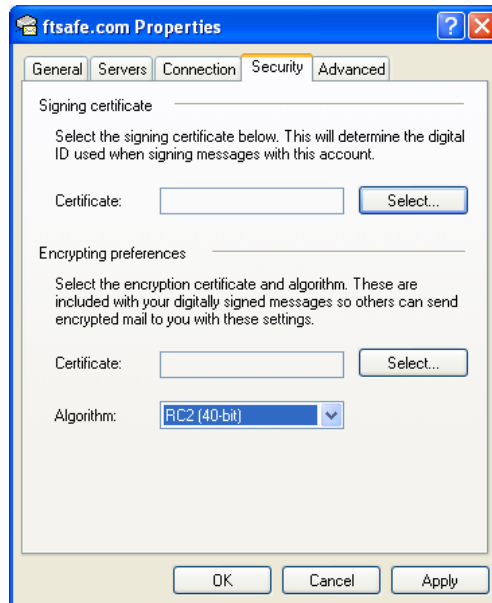


Figure 4-60 Security tab of account

6. Click the **Select...** button for **Signing certificate**.
7. Select the certificate you want to use for signing and click **OK**.

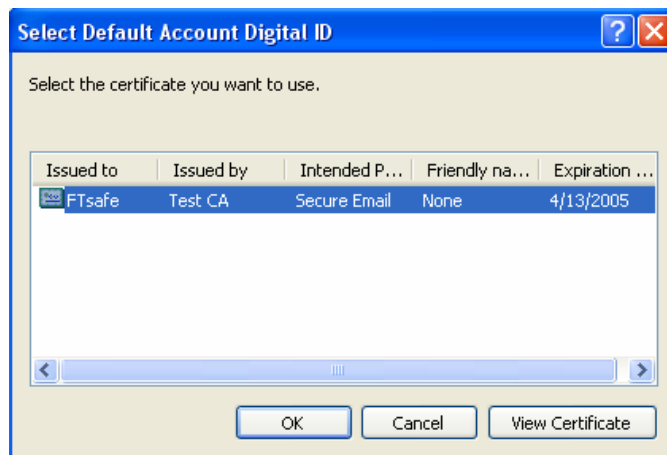


Figure 4-61 Select proper certificate

8. Click the **Select...** button for Encryption preferences.
9. Select the certificate you want to use for encryption and click **OK**.
10. Click **Apply**, then **OK**. Finally, click **Close**.

11. Choose **Options** from **Tools** menu, select **Security** tab

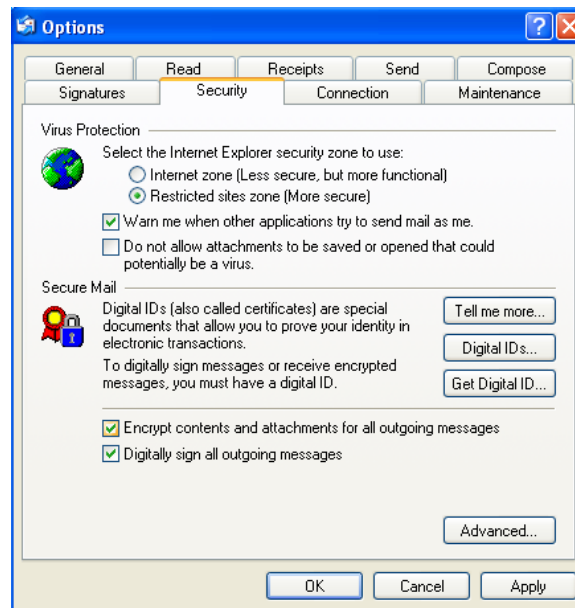


Figure 4-62 Secure email setup

12. If the user wants to send all emails digitally signed, click check box **Digitally sign all outgoing messages**. Also, if want to send all emails encrypted, click check box **Encrypt contents and attachments for all outgoing messages**.
13. Click **Advanced** button

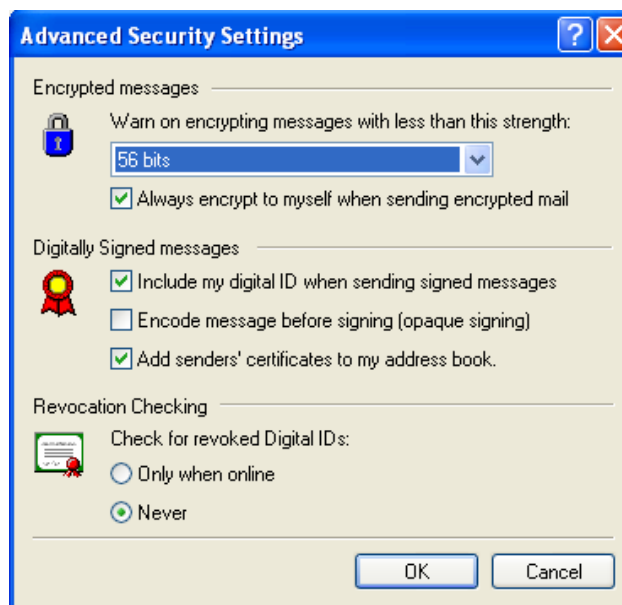



Figure 4-63 Advanced security settings

14. Click check box **Include my digital ID when sending signed messages**.

15. To encrypt your email, you must have the recipient's certificate and the associated public key. Activate **Add senders' certificates to my address book** check-box, the certificates will be automatically stored in your address book.

#### 4.4.2 Send a Message with a Digital Signature

1. Open Outlook Express
2. Create a new message.
3. Fill in the message in a normal way.
4. To add your digital signature to the message, choose **Digitally sign** from the Tools menu.  
  
You should see a red icon  on the right of the window, which indicates the message is signed.  
  
To automatically digitally sign messages, from the **Tools** main menu select **Options**. Click the **Security** tab and select the appropriate check boxes in the Secure mail area.

#### 4.4.3 Importing Another User's Certificate Into the Address Book

To send encrypted e-mail, the sender needs the certificates of the people to whom he wants to send messages. To obtain another user's certificate, do the following:

1. Have a signed-only e-mail message with a red seal icon from other person.
2. Double-click the message to view it.
3. From the **File** main menu, select **Properties**. Click **View Certificates...** on the **Security** tab.
4. Click **Add to Address Book** on View Certificates window. Click **OK** to automatically import the person's certificate.

#### 4.4.4 Send an encrypted Message

1. Create a new message
2. Select recipient from address book that has sent a signed email to you.

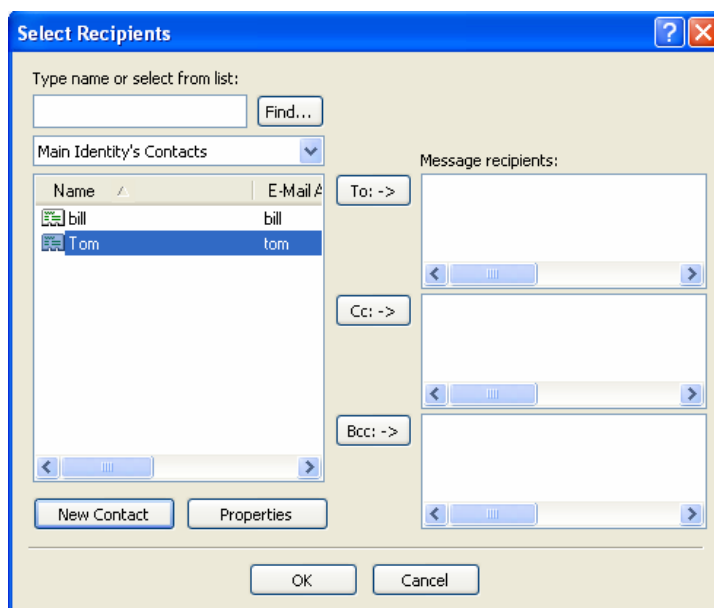


Figure 4-64 Select receiver

- Fill in the message as you normally would.
- Click **Encrypt** on the toolbar.

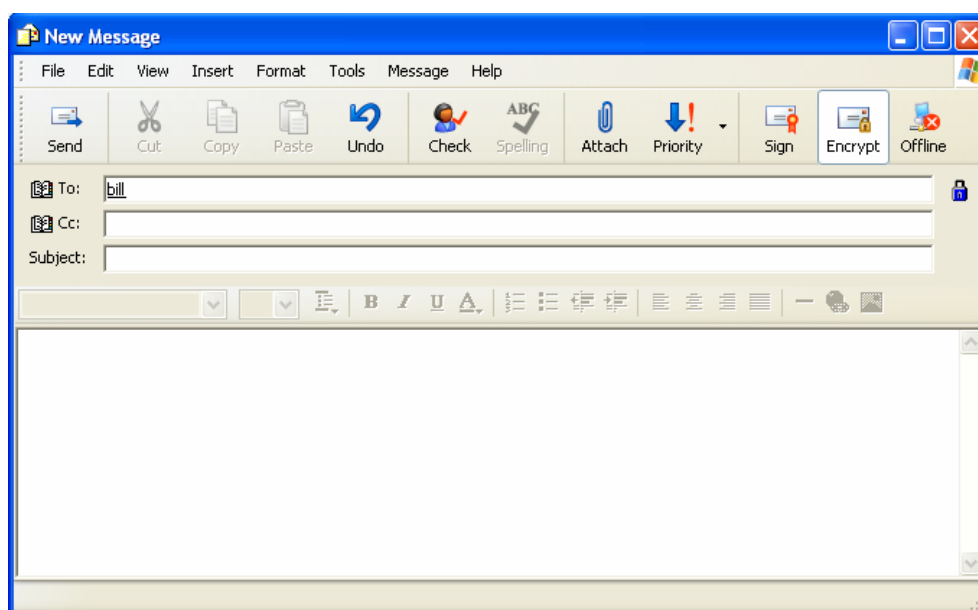


Figure 4-65 Encrypt email

5. Click **Send** to send this email.

## 4.5 Smart Card logon with ePass2000

Microsoft provides smart card logon as a built-in function in Windows 2000/2003/XP. Users can choose logon a domain with traditional username and password or with a smart card to accomplish the identity authentication process. Smart Card logon promotes security and ease-of-use to users, who need only remember the User PIN of the smart card.

### 4.5.1 Configure a CA to Distribute Certificates

In order to logon with ePass2000, one should setup a CA, which can distribute certificates to users. The certificates are stored in the smart card securely. The corresponding configuration processes are listed below.

1. Login the server that is used to issue certificates as the administrator, and open the **Certification Authority** as below.

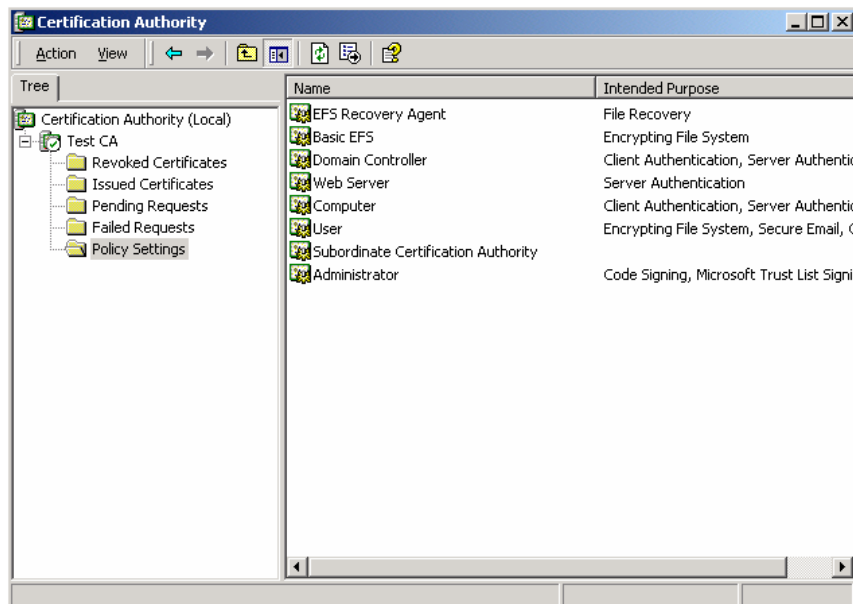


Figure 4-66 Configure CA

In the console, click **control tree** and select: **CA(computer name)-Policy Settings**. The available certificate templates will be listed in the windows on the right hand side.

2. In the **Action** menu click **New** and the sub-menu **certificates to issue**, the following dialog will show

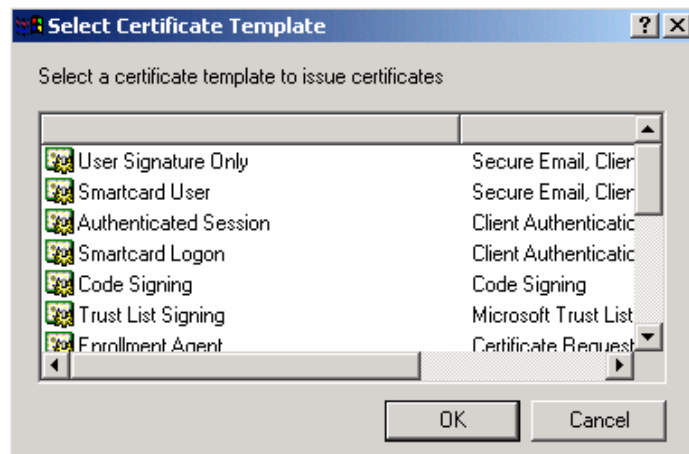


Figure 4-67 Select certificate template

6. Choose **Enrollment Agent** then press **OK**. Follow the previous steps; add **Smartcard Users** and **Smartcard Logon**, two templates, to **certificate to issue** dialogue. The picture will be displayed if finish configuration.

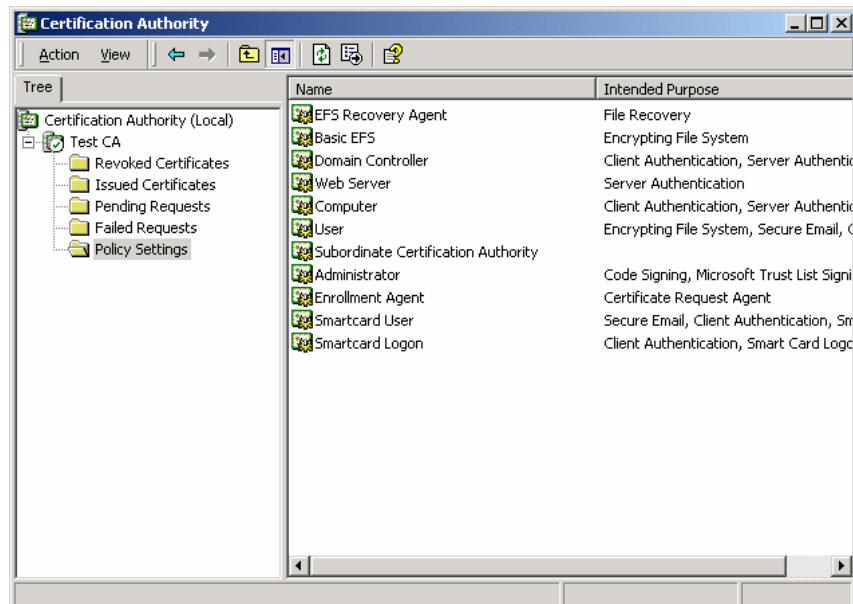


Figure 4-68 Certificate templates

In addition, **Enrollment Agent** certificates can be issued by different CA. The pre-requisite is that the CA, which issues register agent certificates, should be the trusted CA in the domain.

Having an **Enrollment Agent Certificate**, we can setup a website that distribute the certificates.

7. Login as the administrator, press **start** -> **run** , input **mmc** and press **enter**.
8. Click **Add/remove snap-in** on console's menu, and press add
9. In **Standalone snap-ins** dialog, double click **Certificates**. If you login as a user, the certificate will be loaded automatically. On the other hand, if login as the administrator, you should **My**

user account as follow

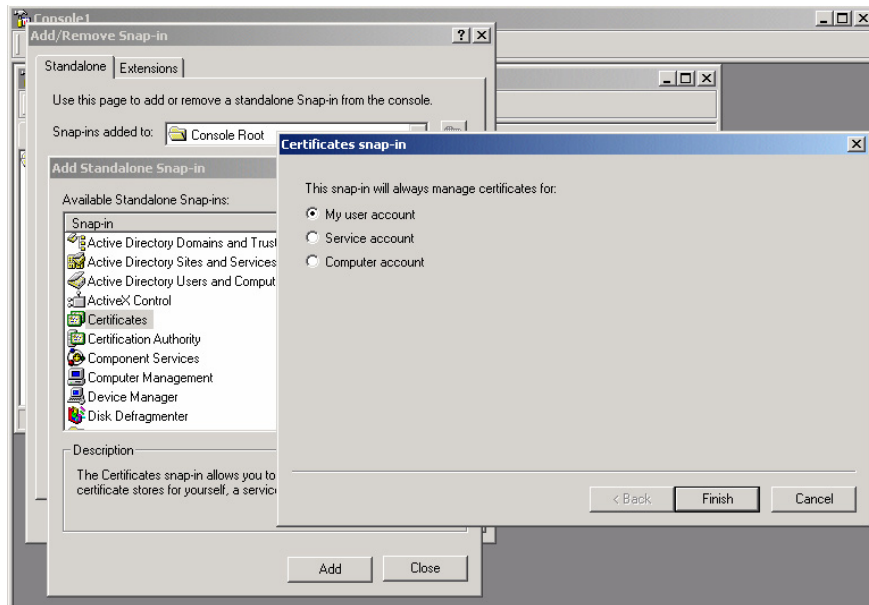


Figure 4-69 Certificate snap-in

8. Close **Snap-in** dialog. Double click **Certificate – Current user**, select **Personal**, then press the right button of the mouse, select **Apply for a new certificate in all missions** in popup menu

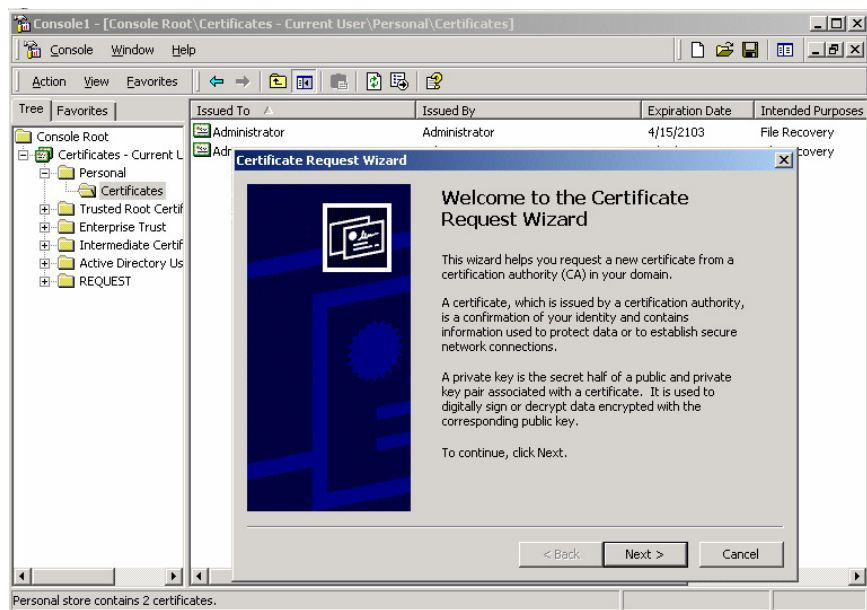


Figure 4-70 personal certificate application

9. Click **Next**, select **Enrollment agent** certificate template.

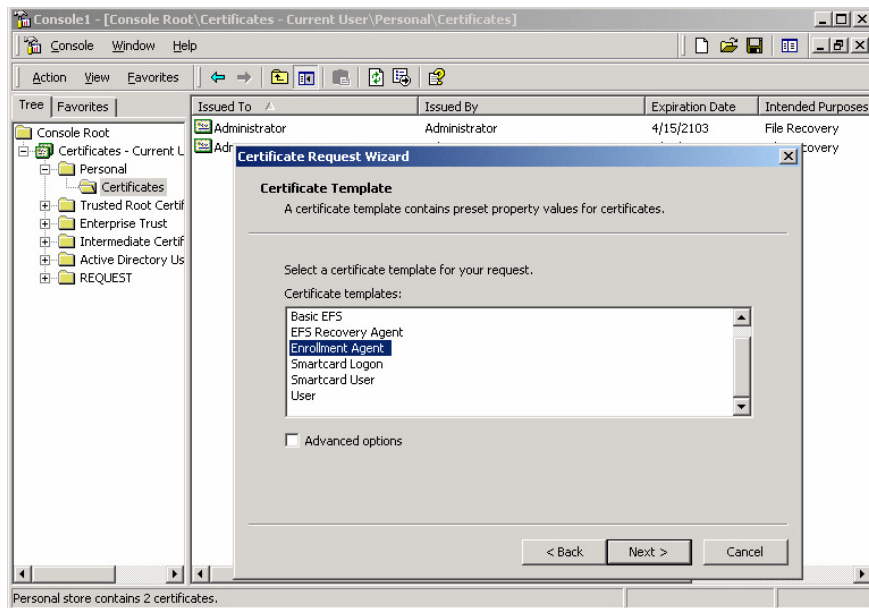


Figure 4-71 Certificate application wizard

10. Click **Next**, input the friendly name of the certificate to finish the application of **Enrollment agent certificate**.

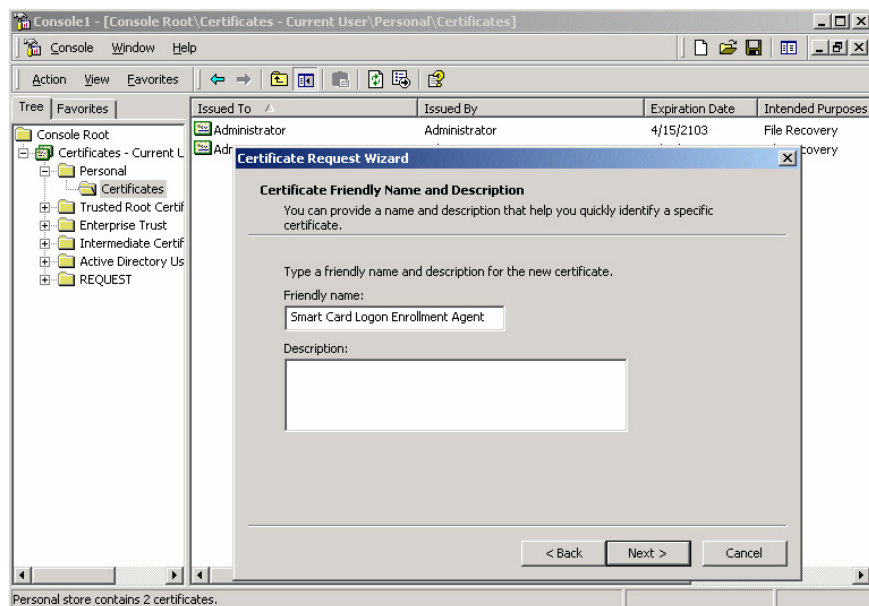


Figure 4-72 Certificate application wizard

The domain administrator should finish the steps above in order to setup a certificate distributor with an available **Enrollment agent certificate** before users can apply for certificates. This administrator should have permission to access the **Enrollment agent certificate** template. For more details about smart card logon and **Enrollment agent certificate**, see Windows online help.

#### 4.5.2 Apply for a Smart Card Certificate

Users can apply for a smart card logon certificate with ePass2000 by completing the following

steps.

1. Logon system as the administrator.
2. Launch an IE browser, and input the URL of the server that issue certificates.
3. Select **Request a certificate**, then click **Next**.
4. Select **Advanced request**, then click **Next**.
5. Plug in ePass2000.
6. Select **Request a certificate for a smart card on behalf of another user using a smart card enrollment station**, then click **Next**. (If it is the first time you apply for a certificate, the browser will download two ActiveX plug-in automatically).

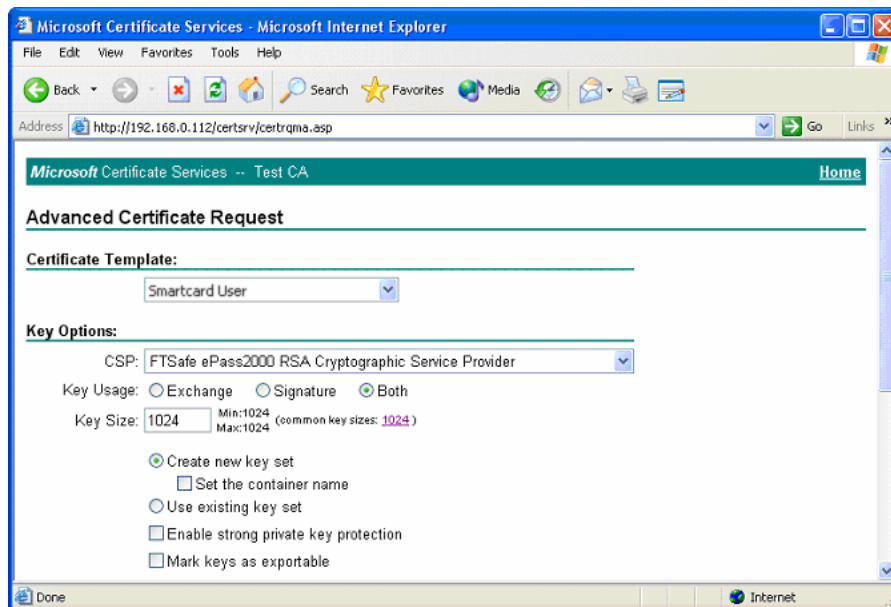


Figure 4-73 Smart card user apply for a certificate

7. Select certificate template as **Smart card User**
8. Select **FTSafe ePss200 RSA Cryptographic Service Provider** from **Cryptographic Service Provider**
9. Select the **Enrollment agent certificate** we applied for in **Administrator signing certificate**
10. Input the corresponding domain username in **Input Username** then click **Submit**.
11. A pop-up windows will display and input the User PIN to ePass2000 in it.

### 4.5.3 View Valid Smart Card Certificate

When finish downloading a certificate, users can view the certificates or apply for a new certificate. After obtaining a smart card certificate, users can logon automatically via ePass2000. In addition, users can use ePass2000 to lock their computer, such that when the user removes the ePass2000, the computer is locked automatically or the user is automatically logged off. In order to unlock the computer, the user need only plug in the ePass2000 and input a valid user PIN.

The operation to setup such these functions is as below:

1. Select **Administration Tools** in **Control Panel**, double click **Local Security Settings**.

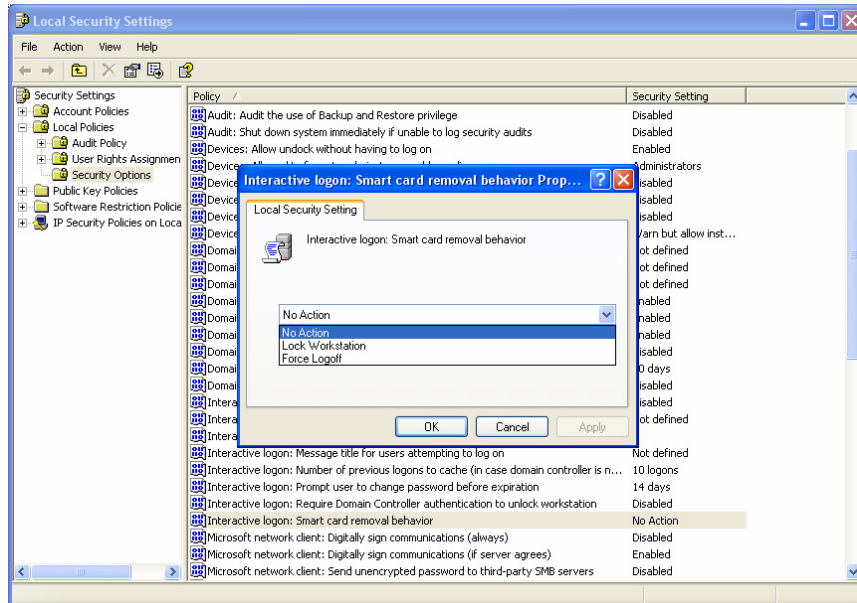


Figure 4-74 Local security settings

2. Double click **smart card removal behavior**, a pop up dialog will display and select the corresponding action.

## 4.6 Integrating Microsoft VPN with ePass2000

### 4.6.1 VPN Server Configuration

VPN stands for Virtual Private Network, which can establish secure channels on an insecure network, establishing a private encrypted transfer tunnel. Windows 2000 and XP provide built-in support for VPN. Like dial-up services to an ISP server, users can logon to the VPN server though a secure channel and secure data transfer is established.

When setting up a secure VPN channel, the client and the server need to authenticate each other, so that setup the secret key used to establish secure session and perform future encryption operation by such key. Windows 2000, XP and Server 2003 allow users to use smart card for logon. In the following chapters, the detail procedures on how to configure VPN routers and servers will be illustrated.

1. Select **Administration Tools** in **Control Panel**, double click **Routing and remote access**.

- Click the corresponding server in list tree on the left hand side, the **Routing and remote access setup wizard** dialog will display as below:

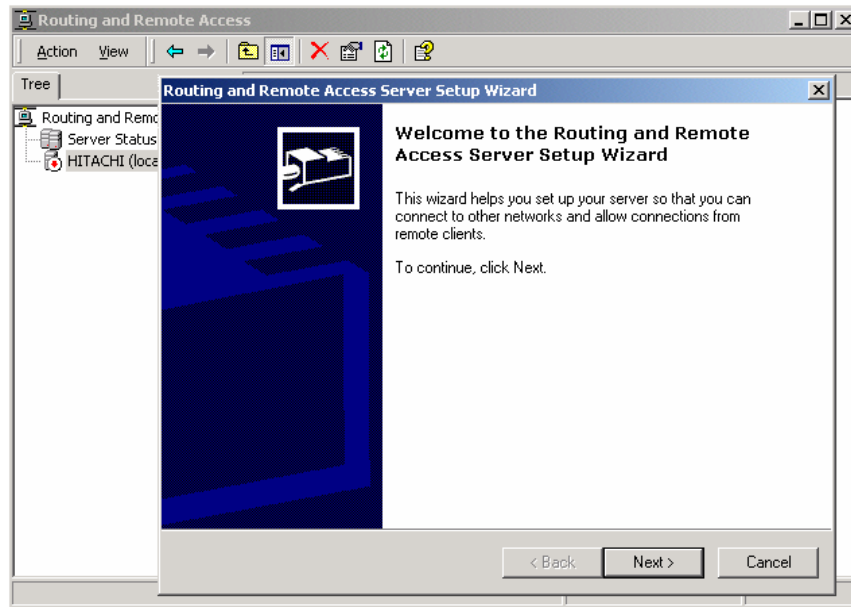


Figure 4-75 Configure Router

- Click **Next**, select **Virtual private network (VPN) server**

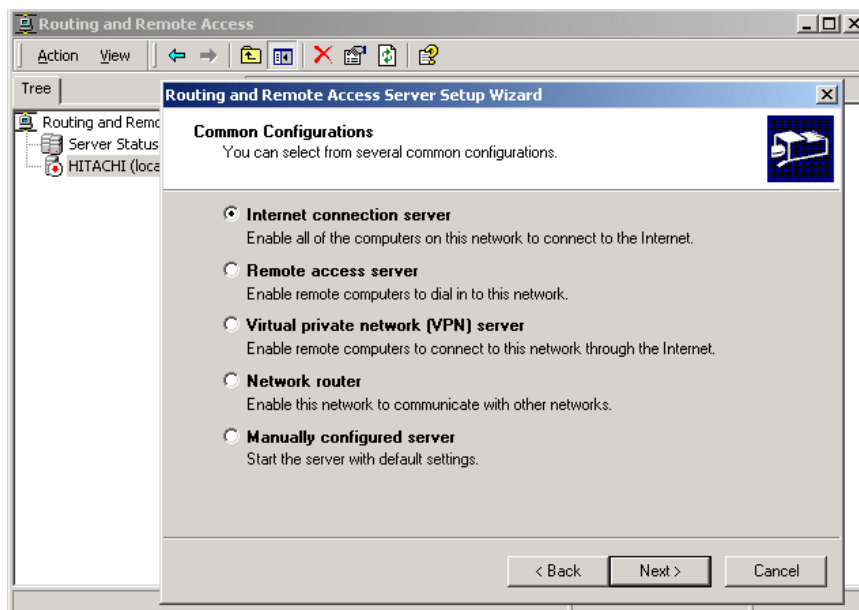


Figure 4-76 VPN server configurations

- Click **Next**, select the supported protocol.

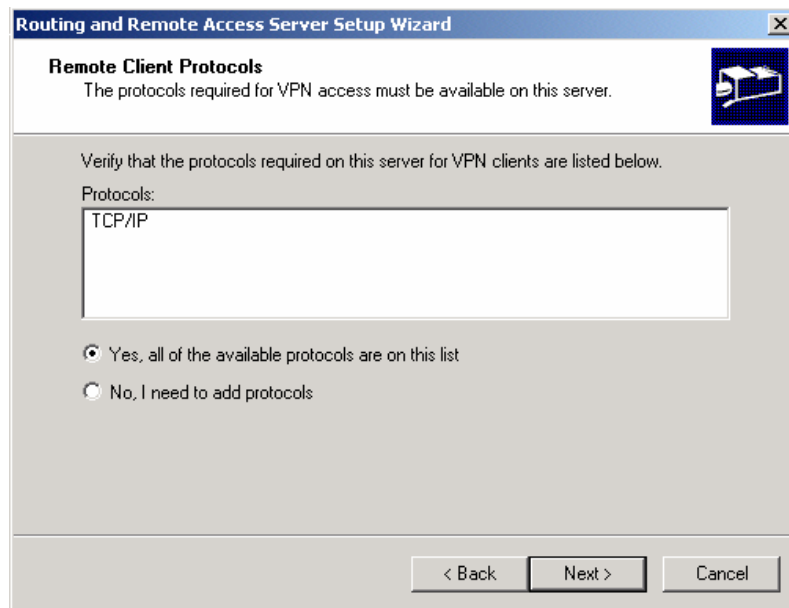


Figure 4-77 Select protocol

- Click **Next**, select network interface.

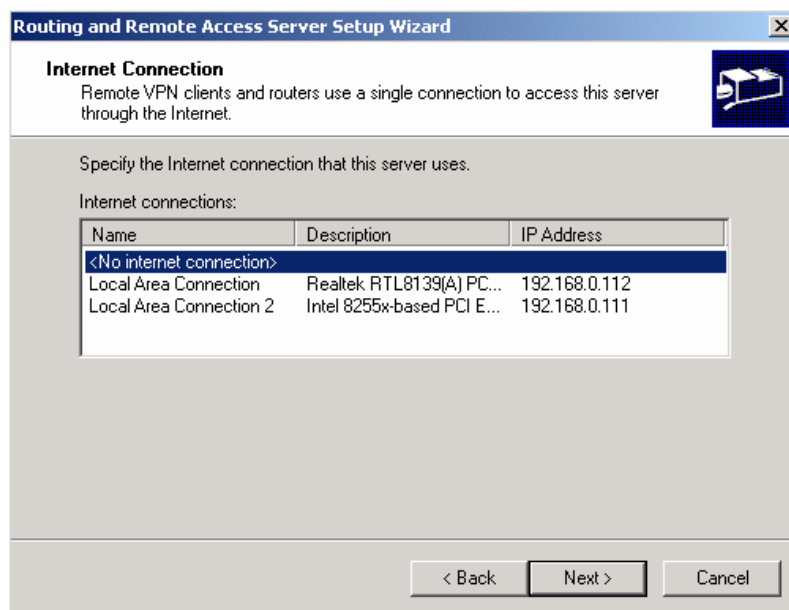


Figure 4-78 Select interface

- Click **Next**, select the **IP address assignment** for the VPN clients. The administrator can use DHCP or assign a fixed scope for the IP address pool.

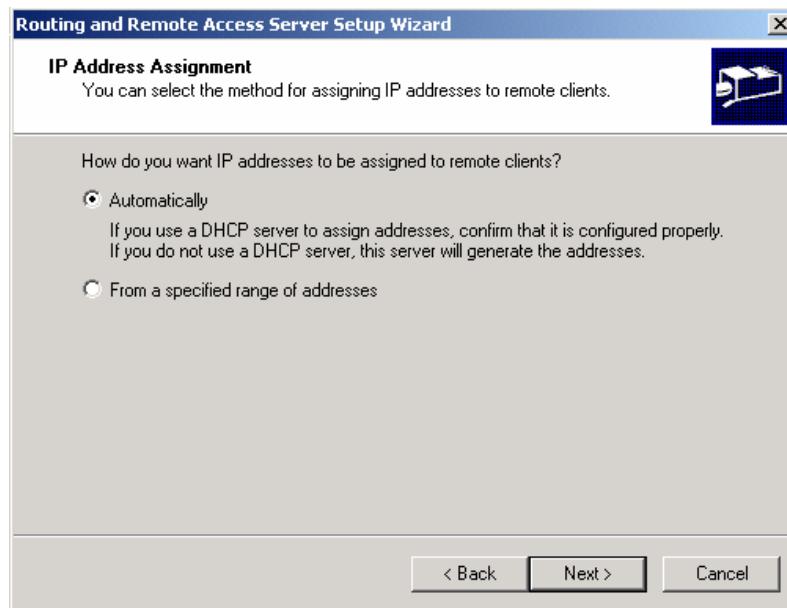


Figure 4-79 Configure IP address

7. Click **Next**, administrator may configure an authentication server RADIUS for VPN or configure it later.

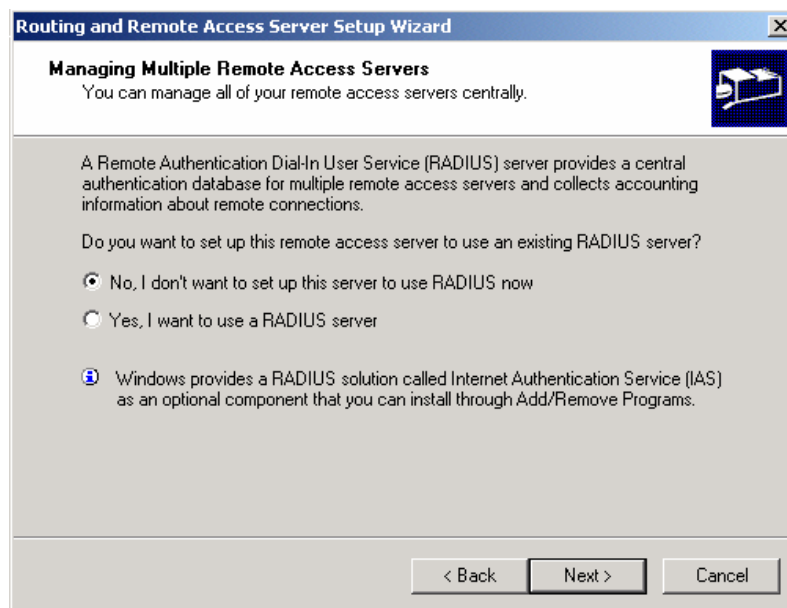


Figure 4-80 Configure authentication server

After completing the steps above, we have essentially setup a VPN dial-in server. Next, we will configure the VPN authentication using smart card logon.

8. In the left list tree in **Routing and remote access** dialog, right click the corresponding server and select the **Property**
9. Select **security** page and click **authentication** button as the following plot.

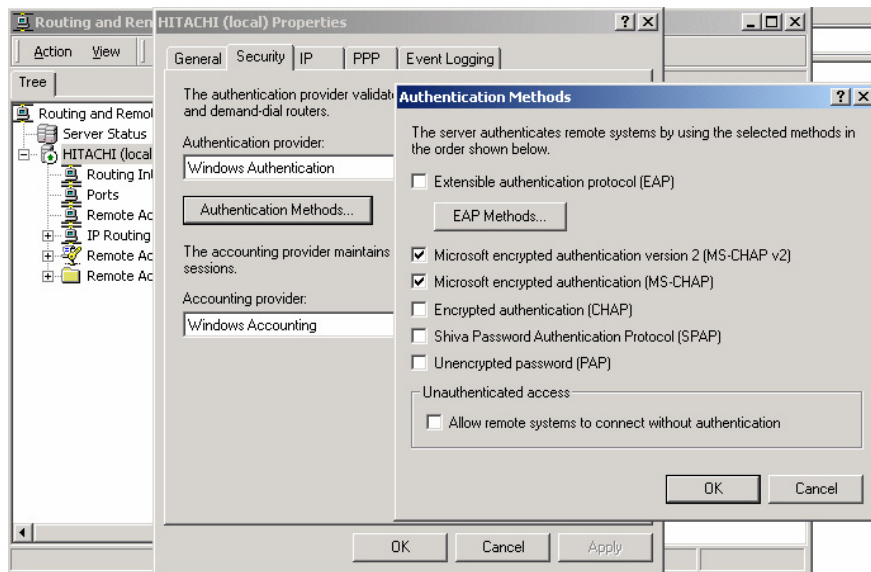


Figure 4-81 Configure server property

10. Select **Extensible Authentication Protocol (EAP)**. EAP is an improvement over traditional logons that only use a username and passwords. Smart card logon is one method of EAP.
11. Close **Authentication** dialog.
12. Close **Server Property** dialog.
13. In the left list tree in **Routing and remote access** dialog, select **Remote access policy** and double click on it.
14. Press **Edit Configure File** in the pop up dialog
15. Select **Identity Authentication** page in **Edit Dial-in Profile**.

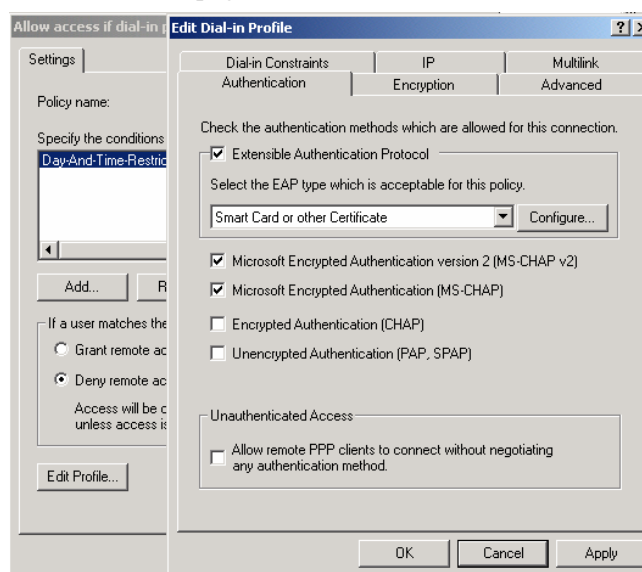


Figure 4-82 Edit dial-in configuration

16. Select **EAP**, then press **OK** to finish configuration.

The VPN dial-in server setup is now complete..

#### 4.6.2 Client Configuration

1. Select **dial-up network** in control panel
2. Double click **Create a new connection**, the **Network connection wizard** will be shown

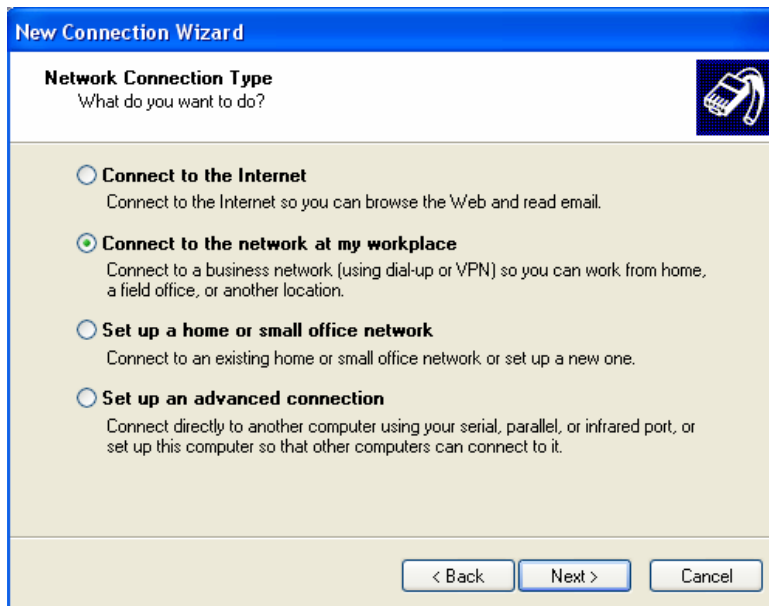


Figure 4-83 Configure network connection

3. Select **Virtual Private Network connection**, and click **Next**



Figure 4-84 Select connection type

4. Input the IP address or the host name of VPN server
5. Select **Use my smart card**

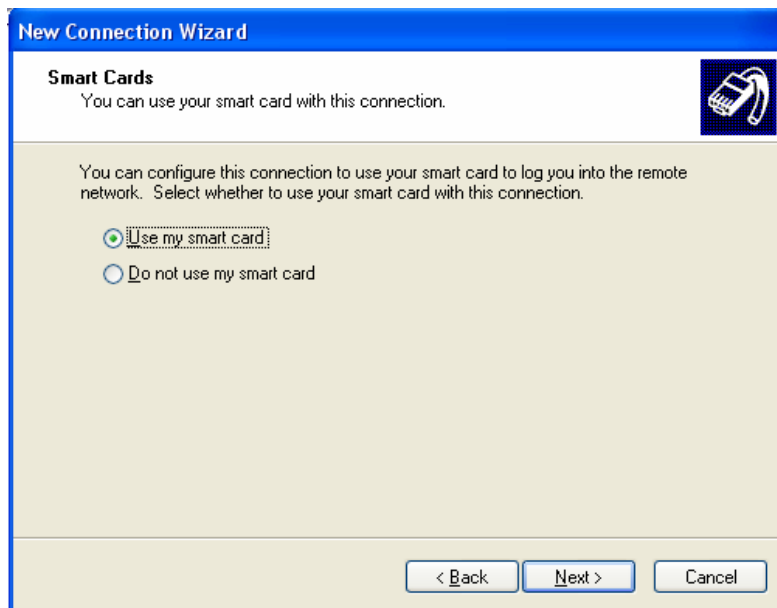


Figure 4-85 Select smart card logon

6. Finally, input the name of this connection.

The VPN client setup is complete. In order to logon using a secure VPN connection and your ePass2000, just double click on this connection and input the user PIN.

## Chapter 5                      Development Guide

This chapter will discuss issues about developing ePass2000 applications, including ePass2000 APIs and how to develop applications for different APIs.

- ePass2000 API
- PC/SC interface
- MS CryptoAPI
- PKCS#11 interface

### 5.1 ePass2000 API

ePass2000 supports two kinds of applications: PKI applications and smart card applications. ePass2000 offers PKCS#11 and CSP for Microsoft CryptoAPI 2.0 interfaces. They are compatible with the PKCS#11 standard from RSA or MS CryptoAPI from Microsoft. Many other software and hardware providers also support these two standards. The fact that ePass2000 conforms to these standards means that it may be integrated with other applications that also conform to these standards, without the need for further development. ePass2000 also offers a PC/SC interface for smart card applications.

ePass2000 PKI applications are based on the PC/SC interface. Developers may choose one or more interfaces to develop their own applications with the ePass2000 token.

### 5.2 ePass2000 PC/SC Interface

The smart card subsystem of Win32 platforms is based on PC/SC standard (for more information about PC/SC standard refer to <http://www.pcscworkgroup.com/>). Components include:

- Smart Card Resource Manager with Win32 API
- User Interface
- COM components for Smart Card Service

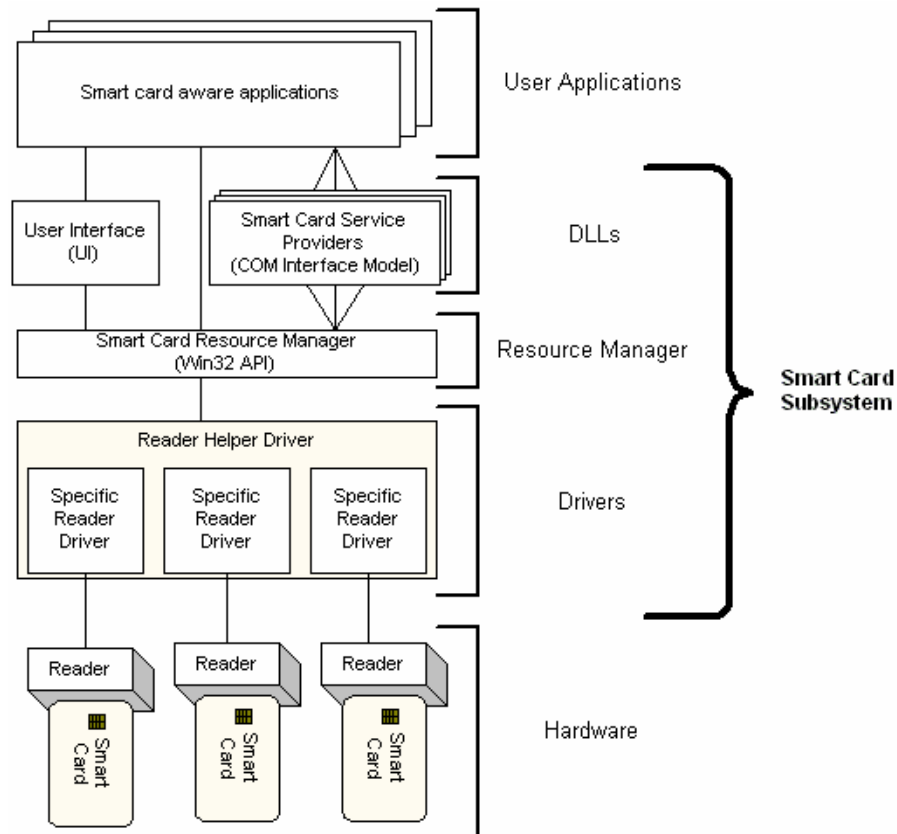


Figure 5-1 The structure of PC/SC system

The Win32 Smart Card Resource Manager abstracts all function calls from the application to the hardware driver. A smart card application need only communicate with the smart card subset of the Win32 API. The Win32 API takes care of communications to the hardware. Even if the smart card vendor changes or updates the smart card interface, there is no affect to the upper-layer smart card application.

The smart card resource manager is the abstract layer that interacts with smart card applications and smart cards. The smart card resource manager function set includes:

- Data Enquiry Functions
- Resource Management Functions
- Resource Manager Handle Functions
- Resource Manager Tool Function
- Monitor Functions
- Smart Card and Card Reader Access Functions
- Direct Access Functions

### 5.2.1 Data Enquiry Functions

These functions are for smart card data enquiry, such as card type list for specific system users, application service interfaces for specific smart cards, group lists of card readers in the system, and card reader lists in one specific group.

Function
SCardGetProviderId
SCardListCards
SCardListInterfaces
SCardListReaderGroups
SCardListReaders

### 5.2.2 Resource Management Functions

You can use specified resource context to update resource database settings.

Function
SCardAddReaderToGroup
SCardForgetCardType
SCardForgetReader
SCardForgetReaderGroup
SCardIntroduceCardType
SCardIntroduceReader
SCardIntroduceReaderGroup
SCardRemoveReaderFromGroup

### 5.2.3 Resource Manager Handle Functions

These functions are to establish and release operation handles for the enquiry and management functions of smart card resource manager.

Function
SCardEstablishContext
SCardReleaseContext

### 5.2.4 Resource Manager Tool Function

This function is to release the memory automatically assigned by the system function when marked with label SCARD\_AUTOALLOCATE.

Function
SCardFreeMemory

### 5.2.5 Monitor Functions

These functions are to trace the status of smart card and card reader. Most of them mark hardware status with an array SCARD\_READERSTATE.

Functions
SCardLocateCards
SCardGetStatusChange
SCardCancel

### 5.2.6 Smart Card and Card Reader Access Functions

These functions are for connection and access to the specified smart card device. They perform I/O operations to smart card using data block with control information. The control information always begins with SCARD\_IO\_REQUEST structure.

Function
SCardConnect
SCardReconnect
SCardDisconnect
SCardBeginTransaction
SCardStatus
SCardTransmit

### 5.2.7 Direct Access Functions

The Win32 smart card subsystem allows the applications to access smart cards that are not completely compatible with ISO7816 standard. So Win32 smart card functions allow applications to send control commands and data from the bottom layer directly to smart card readers. You must set an ID for every property you would like to control in order to use these functions. And Win32 smart card subset has already defined some property IDs.

Function
ScardControl
ScardGetAttrib
ScardSetAttrib

The smart card subsystem components have been configured when you install the system of Windows 2000 and XP. But you should install Microsoft smart card patch for Windows NT 4.0 and Windows 98 to support smart card applications.

Refer to the appropriate MSDN developer documents for more detailed information about the Win32 smart card function set.

## 5.3 ePass2000 MS CryptoAPI

Microsoft CryptoAPI is a set of Win32 data encryption and security APIs provided to developers. The CryptoAPI function set includes basic ASN.1 coding/decoding, hash, data encryption/decryption, digital certificate management and other important cryptology application

functions. Data encryption/decryption functions support both symmetric and asymmetric key algorithms. CryptoAPI is used by all Microsoft Win32 applications and data encryption interface from many other third-party vendors' applications, such as Internet Explorer and Outlook are developed on CryptoAPI.

Secure data transfer over a public network requires three factors: information hiding, identity authentication and integrity check. The CryptoAPI offers functions in addition to those listed above: standard ASN.1 coding and decoding, information decryption, digital certificate and certificate storage management, certificate credit list, certificate revocation list and certificate validity check.

CryptoAPI architecture consists of 5 components:

- Base Cryptography Functions: they fall into the following groups:
  - ◆ Services Provider Functions
  - ◆ Key Generation and Exchange Functions
  - ◆ CryptEncodeObject/CryptDecodeObject Functions
  - ◆ Data encryption/Decryption Functions
  - ◆ Hash and Digital Signature Functions
- Certificate Encode/Decode Functions
- Certificate Store Functions
- Simplified Message Structures
- Low-Level Message Structures

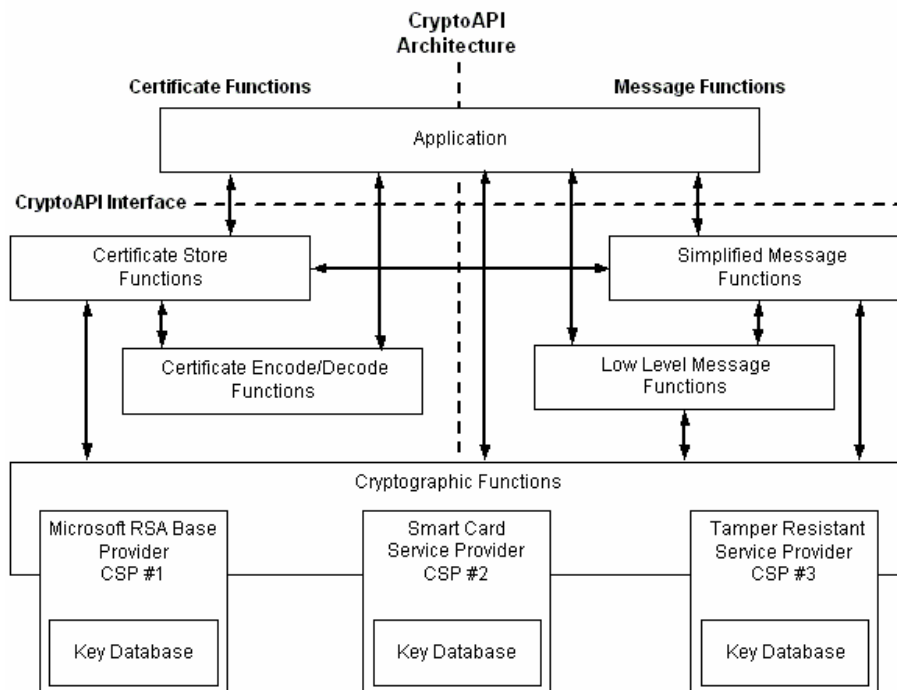


Figure 5-2 The structure of CryptoAPI

The prefix of each function in our CSP is defined in the following table.

Function Group	Prefix definition
Base Cryptography Functions	Crypt
Certificate Encode/Decode Functions	Crypt
Certificate Store Functions	Store
Simplified Message Structures	Message
Low-Level Message Structures	Msg

### 5.3.1 CSP Module for ePass2000

The standard CSP for ePass2000 is seamlessly integrated with the CryptoAPI application program. It can be compatible to the future CryptoAPI applications since it is compliant with Microsoft CSP standard coding style.

CSP for ePass2000 is PROV\_RSA\_FULL type of provider. It has the following features:

- Provide the secure storage for RSA key pair
- Provide different encryption algorithms and HASH functions
- Hardware RSA algorithm

ePass2000 CSP also supports the Microsoft extension standard for smart cards. So the ePass2000 token can be used with VPN and Windows 2000 smart card logon.

### 5.3.2 Certificate Storage Space

There is a physical storage space assigned to certificate in ePass2000 token, named “ePass2000”. When the application configures this physical name in **CertOpenStore** function, it can access the certificate inside. But the certificate cannot be deleted.

## 5.4 The ePass2000 PKCS#11 Module

The exponential growth of the Internet fueled the demand for applications to secure transactions and communications over the public networks. With the growth in cryptographic applications came a corresponding need for these new applications to interoperate and communicate with one another. RSA Laboratories created the Public Key Cryptography Standard (PKCS) with this purpose in mind.

The PKCS#11 standard is one of the PKCS standard families. The PKCS#11 standard (also known as “Cryptoki”) is designed to ensure compatibility and interoperability between Public Key Cryptography implementations developed by different software or token vendors. It defines a common programming interface for Cryptoki tokens, such as ePass2000, and other cryptographic hardware.

Before developing applications with the ePass2000 PKCS#11 module, you need to be familiar with the PKCS#11 standard. You may find the latest version of the standard at <http://www.rsasecurity.com/rsalabs/pkcs/pkcs-11/index.html>.

The ePass2000 Cryptoki module supports PKCS#11 version 2.11. The interface is provided as a set of C language functions in a Win32 dll. The following files are required to develop an ePass2000 PKCS#11 interface:

File	SDK Path
pkcs11.h	\Include (Provided by RSA)
pkcs11f.h	\Include (Provided by RSA)
pkcs11t.h	\Include (Provided by RSA)
ep2pk11.lib	\Lib
ep2pk11.dll	\Lib (should be under the system directory at runtime)

The ep2pk11.dll is the core dynamic library of ePass2000's PKCS#11 module. It contains implementation of the interface defined by the RSA PKCS#11 specification. You should include cryptoki.h header file in your project if you need to use this interface.

#### 5.4.1 Supported PKCS#11 Object Class

The ePass2000 PKCS#11 implementation creates the following PKCS#11 objects (as specified in version 2.11 of the standard):

Object Class	Description
CKO_DATA	For data structures defined by applications
CKO_SECRET_KEY	For symmetric keys
CKO_CERTIFICATE	For X.509 public certificates
CKO_PUBLIC_KEY	For RSA/ DSA public key.
CKO_PRIVATE_KEY	For RSA/DSA private key.

Applications may create and store inside epass2000 all of the objects listed in the table above. Or the developer may choose to allow the objects to be available at runtime. Developers should consider memory capacity limitations of ePass2000 as they select the objects for storage.

#### 5.4.2 PKCS#11 Mechanisms Supported by ePass2000

The following table illustrates all mechanisms implemented by ePass2000 PKCS#11 module:

Mechanisms
CKM_RSA_PKCS_KEY_PAIR_GEN
CKM_RSA_PKCS
CKM_DSA_KEY_PAIR_GEN
CKM_DSA
CKM_RC2_KEY_GEN

CKM_RC2_ECB
CKM_RC2_CBC
CKM_RC4_KEY_GEN
CKM_RC4_ECB
CKM_RC4_CBC
CKM_DES_KEY_GEN
CKM_DES_ECB
CKM_DES_CBC
CKM_DES_CBC_PAD
CKM_DES3_KEY_GEN
CKM_DES3_ECB
CKM_DES3_CBC
CKM_DES3_CBC_PAD
CKM_MD2
CKM_MD5
CKM_SHA_1

The following table shows key sizes supported by ePass2000 PKCS#11 library.

Mechanisms	Key Sizes
CKM_RSA_KEY_PAIR_GEN	1024 bits
CKM_DSA_KEY_PAIR_GEN	1024 bits
CKM_RC2_KEY_GEN	1-128 bytes
CKM_RC4_KEY_GEN	1-256 bytes
CKM_DES_KEY_GEN	8 bytes
CKM_DES3_KEY_GEN	24 bytes

## 5.5 Functions of the ePass2000 PKCS#11 Library

Developers and hardware vendors should familiarize themselves with the PKCS #11 functions because they will need to call these functions to integrate their applications to ePass2000 (See the RSA PKCS#11 specification at <http://www.rsasecurity.com>.)

The PKCS#11 specification is for a general case of Cryptoki hardware. Specific Cryptoki hardware implementation will vary depending on the characteristics of the hardware itself.

The ePass2000 PKCS#11 library also has some hardware specific variances from the standard:

- The C\_WaitForSlotEvent with block mode is not implemented.
- There are some functions defined in the PKCS #11 standard that are not implemented for the ePass2000 interface. If a non-implemented function is called, it will return a code, CKR\_FUNCTION\_NOT\_SUPPORT.

*Note: ePass2000 is the “token” referred to in the PKCS#11 standard.*

The PKCS #11 standard will refer to a “slot” for a card reader. Since the card reader function is integrated into ePass2000, this terminology does not apply.

The following table lists functions of PKCS#11 specification:

Name
<b>General Purpose Functions</b>
C_Initialize
C_Finalize
C_GetInfo
C_GetFunctionList
<b>Slot and Token Management Functions</b>
C_GetSlotList
C_GetSlotInfo
C_GetTokenInfo
C_WaitForSlotEvent
C_GetMechanismList
C_GetMechanismInfo
C_InitToken
C_InitPIN
C_SetPIN
<b>Session Management Functions</b>
C_OpenSession
C_CloseSession
C_CloseAllSessions
C_GetSessionInfo
C_GetOperationState *
C_SetOperationState *
C_Login
C_Logout
<b>Object Management Functions</b>
C_CreateObject
C_CopyObject *
C_DestroyObject
C_GetObjectSize *
C_GetAttributeValue
C_SetAttributeValue
C_FindObjectsInit
C_FindObjects
C_FindObjectsFinal
<b>Encryption Functions</b>

C_EncryptInit
C_Encrypt
C_EncryptUpdate
C_EncryptFinal
<b>Decrypt Functions</b>
C_DecryptInit
C_Decrypt
C_DecryptUpdate
C_DecryptFinal
<b>Message Digesting Functions</b>
C_DigestInit
C_Digest
C_DigestUpdate
C_DigestKey
C_DigestFinal
<b>Signing and MACing Functions</b>
C_SignInit
C_Sign
C_SignUpdate *
C_SignFinal *
C_SignRecoverInit *
C_SignRecover *
<b>Verifying Signatures and MAC Functions</b>
C_VerifyInit
C_Verify
C_VerifyUpdate *
C_VerifyFinal *
C_VerifyRecoverInit *
C_VerifyRecover *
<b>Dual-Function Cryptographic Functions *</b>
C_DigestEncryptUpdate
C_DecryptDigestUpdate
C_SignEncryptUpdate
C_DecryptVerifyUpdate
<b>Key Management Functions</b>
C_GenerateKey
C_GenerateKeyPair
C_WrapKey
C_UnwrapKey
C_DeriveKey
<b>Random Number Generation Functions</b>
C_SeedRandom
C_GenerateRandom

Parallel Function Management Functions
C_GetFunctionStatus
C_CancelFunction

*Note: The functions with star mark are not supported in this version.*

## Appendix

### ePass2000 Technical Specifications

<b>Supported Operating Systems</b>	Windows 98SE/ME/2000/XP/Server 2003; Mac OS 9 above; Linux
<b>Certifications and Standards</b>	PKCS#11 v2.10, MS CAPI, PC/SC, X.509 v3 Certificate Storage, SSL v3, IPSec, ISO 7816
<b>Memory Size (by Model)</b>	32k
<b>On-Board Security Algorithms</b>	RSA, DES, 3DES, DSA, MD5, SHA-1
<b>Chip Security Level</b>	Secured and Encrypted Data Storage
<b>Dimensions</b>	50 x 17 x 7 mm (1.97 x 0.67 x 0.28 inches)
<b>Weight</b>	6g
<b>Power Dissipation</b>	< 250 mW
<b>Operating Temperature</b>	0 C ~ 70 C (32 F to 156 F)
<b>Storage Temperature</b>	-40 C ~ 85 C (-40 F to 185 F)
<b>Humidity Rating</b>	0 to 100% without condensation
<b>Connector Type</b>	USB type A (Universal Serial Bus)
<b>Casing</b>	Hard Molded Plastic, Tamper Evident
<b>Memory Data Retention</b>	At least 10 years
<b>Memory Cell Rewrites</b>	At least 100,000 times