IBM® SecureWay® X.509 Public Key Infrastructure
for Multiplatforms

Installation Guide

Version 1 Release 1

IBM® SecureWay® X.509 Public Key Infrastructure
for Multiplatforms

Installation Guide

Version 1 Release 1

Note

Before using this information and the product it supports, read the general information under “Appendix B. Notices” on page 65.

First Edition (October 1999)

This edition applies to SecureWay X.509 Public Key Infrastructure for Multiplatforms, program 5697-F93, version 1 release 1 modification 3, and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this book

This book provides information on the IBM® SecureWay® X.509 Public Key Infrastructure for Multiplatforms , hereafter referred to as PKIX. The topics discussed include:

- An overview of the product
- Installation and configuration procedures for both IBM AIX® and Microsoft® Windows® platforms

Who should read this book

The *Installation Guide* is for anyone responsible for performing installation and configuration tasks, and system administrators.

This book assumes that you have a basic understanding of e-business security implementations and terminology, computer network industry standards, and knowledge of Windows and AIX platforms.

How this book is organized

This book is organized as follows:

- **Chapter 1. About SecureWay X.509 Public Key Infrastructure for Multiplatforms** presents an overview of PKIX which discusses some of the major components and the architecture.
- **Chapter 2. Software requirements** lists the software requirements for both Windows and AIX environments.
- **Chapter 3. Installing the PKIX components** explains the installation steps required to install PKIX on Windows and AIX machines after downloading the install image.
- **Chapter 4. Setting up the LDAP server** describes how to configure the LDAP schema file to work with PKIX.
- **Chapter 5. Installing the PKIX APIs** overviews the PKIX Software Developer's Kit (SDK) and how to install it.
- **Chapter 6. Configuring the .ini files** describes the procedures required to define and configure the initialization files for the various PKIX components.
- **Chapter 7. Using the PKIX GUIs** discusses the GUIs that enable you request and manage certificates.
- **Chapter 8. Using the .ini file editor** explains how to use this tool to modify initialization files for the various PKIX components.
- **Chapter 9. Importing / exporting certificates for browsers** describes the steps required to export PKIX certificates and what you must do to import those certificates into a browser.
- **Appendix A. .Ini files** explains the initialization files, their format, and the various parameters and values you can specify.

A glossary of terms and an index are also provided.

Year 2000 readiness

These products are Year 2000 ready. When used in accordance with their associated documentation, they are capable of correctly processing, providing, and/or receiving date data within and between the twentieth and twenty-first centuries, provided that all products (for example, hardware, software, and firmware) used with the products properly exchange accurate date data with them.

Service and support

Contact IBM for service and support for all the products included in the IBM SecureWay FirstSecure offering. Some of these products may refer to non-IBM support. If you obtain these products as part of the FirstSecure offering, contact IBM for service and support.

Conventions

The following conventions are used in this book:

Convention	Meaning
`monospace`	Syntax, sample code
italic	Emphasis, book titles, variable data

Web information

Information about updates to IBM SecureWay FirstSecure products is available at the following Web address:

`http://www.ibm.com/software/security/firstsecure/library`

Related information

The *IBM SecureWay X.509 Public Key Infrastructure for Multiplatforms Programming Guide and Reference* contains programming information enabling you to develop applications using the APIs available with PKIX.

Chapter 1. About SecureWay X.509 Public Key Infrastructure for Multiplatforms

IBM SecureWay X.509 Public Key Infrastructure for Multiplatforms, hereafter referred to as PKIX, is an integrated security solution for the management of digital certificates. These standard certificates can be used in a wide range of Internet applications, providing a means to authenticate users and ensure trusted communications.

PKIX is based on the Internet Engineering Task Force's (IETF) Public Key Infrastructure (PKI) working group's specifications for interoperability of PKI offerings from separate vendors. The reference implementation for these standards, also known as "Jonah", is publicly available on the Web at http://www.mit.edu/pfl.

PKIX also uses the IBM KeyWorks toolkit, which implements the Common Data Security Architecture (CDSA) that has been accepted as a standard by The Open Group. CDSA is a standardized framework that can support multiple trust models, certificate formats, cryptographic algorithms, and certificate repositories. Because the reference implementation is based on a standardized framework, you can choose between different trust models, different key algorithms, and other features supported under this implementation.

PKIX includes support for the complete certificate life-cycle, including:

- Enrollment
- Certificate creation
- Certificate update
- Certificate and certificate revocation list (CRL) creation and publication
- Certificate revocation

PKIX components

PKIX software enables you to create, issue, manage, store, and revoke certificates based on public-key cryptography. Specifically, PKIX provides:

- Certificate authority (CA)
- Registration authority (RA)
- Support for a client system, also referred to as an end entity (EE)
- API Library
- Java™-based GUIs to support CA, RA, and EE functions

Descriptions of each of these components follow.

Certificate authority

The certificate authority (CA) is a server application trusted by one or more users to create, issue, and revoke certificates. Included with the CA is a GUI, which is discussed further in "Using the CA GUI" on page 33. The CA and its GUI are also referred to as the foreground CA.

A sample implementation of a background CA server and a remote CA administrator GUI is also provided. These entities enable the CA services to be controlled by one or more administrator GUIs, running in the same system or different systems.

Registration authority

The registration authority (RA) is a server application responsible for the following administrative tasks necessary in the registration of clients:

- Confirming the client's identity
- Validating that the client is entitled to have the attributes requested in a certificate
- Verifying that the client possesses the private key associated with the public key requested for a certificate

Included with the RA is a GUI that allows you to perform all of the above tasks. For more information about the GUI, see "Using the RA GUI" on page 35. The RA and its GUI are also referred to as the foreground RA.

A sample implementation of a background RA server and a remote RA administrator GUI enable the RA services to be controlled by one or more administrator GUIs. The administrator GUIs can run on the same system or different systems.

End entity

The end entity (EE) or client initiates certificate and revocation requests and maintains its own private keys and certificates issued by PKIX. Initially, the end entity must preregister with the RA over a secure channel (such as password-protected electronic mail) before requesting certificates. A GUI is provided that enables the client to request these services. The GUI is discussed further in "Using the EE GUI" on page 37.

API library

The PKIX APIs provide the functions and services available to the GUI. These C and C++ APIs, many of which also have a Java interface, enable programmers to develop applications that access the public key infrastructure. Included is a C++ class library that implements basic Abstract Syntax Notation One (ASN.1) types and enables the easy construction of high-level objects and their proper encoding and decoding.

These items, along with header files and libraries, are packaged in a Software Developer's Kit (SDK). Installation details for this optional component are described in "Chapter 5. Installing the PKIX APIs" on page 15.

The *IBM SecureWay X.509 Public Key Infrastructure for Multiplatforms Programming Guide and Reference* describes the APIs and presents information to enable programmers to develop PKIX functions beyond what is provided here.

Architecture overview

The previous section discussed the key components that get installed as part of the PKIX installation process. This section highlights some of the key components in the PKIX architecture.

Certificate Management Protocol

PKIX uses the Certificate Management Protocol (CMP) to control and manage its various components. Specifically, PKIX uses CMP for its RA to CA communications and for communicating with the EE. CMP provides management functions for the certificate and key life cycles. CMP defines message formats and describes the various transport protocols. CMP also specifies how message protection should be done, independent of the transport mechanism. Transmission control protocol (TCP)

is the primary transport mechanism used. An abstraction layer over sockets exists, enabling support for additional polling transports. For more information about CMP, see the IETF's RFC 2510 at http://www.ietf.org.

Object store

The object store is a disk-based repository for persistent objects (such as certificate and revocation requests) and is used to store transactions in progress and the state of those transactions. The state information in the object store gets updated each time a transaction goes from one entity to another. Each EE, RA, and CA has an object store. The objects contain a copy of the request message and control information. When stored, the objects are ASN.1 encoded. The object store is an indexed file and a new object (transaction) written to this file is assigned the lowest free index.

Objects in the object store are either active control objects or surrogates. A surrogate corresponds to a control object that has passed through the object store and is expected to return; the surrogate is a place where any state data regarding that object is saved. For example, when a client enrolls and requests the PKI to generate its keypair, the control object, CertRequest, moves from the client system to the RA. However, some state information (for example, an encryption key) must be retained by the client so that it can decode the fulfilled CertRequest when it returns. The client maintains a surrogate of the request once the active CertRequest has been forwarded to the RA for processing. The surrogate is used to decode the CertRequest when it returns, at which point the surrogate is automatically deleted.

Because objects in the object store are stored in ASN.1 encoding, retrieval and storage can be a relatively expensive operation. The object store caches modifications to objects and does not update the disk storage until either an object's state changes or the GUI modifies the object.

To minimize the ASN.1 parsing overhead, PKIX uses an object-cache layer above the object store that performs a write-through cache of object store objects. As a result, the object requires parsing only the first time PKIX references it after a server restart. The object cache layer also provides an additional per-object storage area which is not disk based. PKIX uses this area for storing transient, security-related information (such as the password under which a CMP preregistration record is protected). The object cache support also provides record locking of objects, protecting against simultaneous access by multiple threads.

BinBin

BinBin is a persistent storage object on the RA and CA servers used to store lists of trusted certificates and CA and RA certificates. In general, BinBin is used as a bin for storing any persistent binary data.

The structure of BinBin is as follows: the serial numbers (for CRLs and ARLs) followed by a Basic Encoding Rules (BER) encoded sequence of certificates.

BinBin is read upon server startup and cached into memory.

Scheduler database

The CA must issue certificate revocation lists (CRLs) on a regular basis. To ensure events occur when they need to, a scheduler is implemented in PKIX. The scheduler saves queued events to disk so that the events occur automatically and on time if the server is running.

Issued certificate list

In order to maintain a history of certificates that it issues, the CA uses an issued certificate list (ICL). The ICL retains a secure copy of each certificate issued, indexed by serial number. The index is stored as a separate file. This list is used to determine which certificates should be published on a CRL. The ICL is a persistent data store.

Smart card support

Smart cards are portable cryptographic devices used for storing certificates and keys, as well as performing cryptographic operations (specifically signing) without releasing the private key from the card. Not all users of PKIX will have access to smart card hardware, so in addition to support for smart card hardware, PKIX contains a virtual smart card (one implemented in software) that acts as if a real smart card is being used without requiring changes in the overall design.

The EE, RA, CA, and RA and CA administrator GUIs each have a smart card configured in the PKIX infrastructure. The end entity, in requesting a certificate, also stores a private key on its smart card. Once the certificate is approved, it is returned to the EE. Any certificates that are stored on the smart card for verification purposes can be associated with a private key by using the same key identifier as the private key.

The CA and RA each initialize their smart cards to have a private key and corresponding certificate that are self-signed. This allows RAs to sign requests and CAs to sign CRLs without exposing their private keys outside the smart card.

Smart card support in PKIX is implemented through two interfaces:

- CDSA
- PKCS#11

The CDSA framework is a common interface for cryptographic and data-storage operations endorsed by The Open Group. As a benefit to PKIX, using the CDSA framework expands the PKIX ability to provide plug-in services for cryptographic functions. CDSA provides a standard means to register a cryptographic provider. It also helps to manage the providers, thus ensuring that the provider used is secure.

CDSA provides its own set of APIs for cryptographic service providers, data libraries, trust policies, and certificate parsing libraries. The documentation for the CDSA APIs can be found at the Intel Web site and in the IBM KeyWorks toolkit. IBM KeyWorks is IBM's implementation of the CDSA standard. PKIX does not use the certificate parsing libraries; instead, PKIX contains its own set of flexible, reusable ASN.1 classes that provide the functions typically supplied by a certificate parsing library.

PKCS#11 version 2.01 is used in the virtual smart card implementation. A smart card must support PKCS#11 functions for key pair generation, data storage, and signing to be fully functional for a PKIX implementation. It is through this interface that Web browsers perform smart card operations and certificate management.

PKIX also provides support for predefined smart card key pairs. This feature allows users to select an existing key pair instead of generating a new one when certificates are created.

LDAP support

IBM PKIX uses the Lightweight Directory Access Protocol (LDAP) support to store digital certificates, and certificate revocation lists (CRLs). The CA delegates the authority to post information in the LDAP directory to the RA.

Details on configuring the LDAP schema are discussed in “Chapter 4. Setting up the LDAP server” on page 13.

Chapter 2. Software requirements

Your operating system must meet the software requirements discussed in the following sections. For the latest information, refer to the README that accompanies PKIX for any last-minute changes to the requirements and procedures in this guide.

During the installation process, the PKIX runtime environment, GUIs, and optional Software Developer's Kit (SDK) are installed. This chapter describes how to perform the installation process for the supported platforms. You can perform the installation steps from a GUI or from a command line interface.

Windows requirements

For a Windows environment, you must have:

- Any of the following:
  - Windows 95
  - Windows 98
  - Windows NT® 4.0, Service Pack 3 or later

    Note: Windows NT is required for the CA and RA.
- IBM KeyWorks 1.1.3
- Sun Java™ Foundation Classes (Swing) 1.0.3
- Sun Java™ Runtime Environment 1.1.6 (for the GUI and IniEditor)
- LDAP Client or IBM LDAP Server 2.1 or later

  Note: LDAP is required only if you post certificates and certificate revocation lists (CRLs).
- Microsoft Visual C++ Version 6.0 or later

  Note: Visual C++ is required only for compiling PKIX APIs.

AIX requirements

For an AIX environment, you must have the following:

- AIX 4.2.1 or later
- IBM KeyWorks 1.1.3
- IBM C++ Runtime Environment 3.6.4
- IBM Open Class Library Application Runtime Environment 1.6.6
- Sun Java Foundation Classes (Swing) 1.0.3
- LDAP Client or LDAP Server 2.1 or later

  Note: LDAP is required only if you post certificates and certificate revocation lists (CRLs).
- IBM Java Runtime Environment 1.1.6.7 (for the GUI and IniEditor)

Chapter 3. Installing the PKIX components

The PKIX installation process enables you to install the following components:

- PKIX Certificate Authority
- PKIX Registration Authority
- PKIX End Entity (EE)
- PKIX Documentation
- PKIX Software Developer's Kit (SDK)

The PKIX Runtime Environment is installed for each component except for the PKIX documentation.

The PKIX installation and configuration is designed to support a wide variety of configurations. It is possible to install all five components on the same system, install each separate component on a different system, or any configuration in between.

This chapter describes how to install the PKIX components. The PKIX documentation is independent of the other components. You can install the components in any order you like.

The installation procedures in this book describe the steps required after you have downloaded the IBM SecureWay Toolbox V2.0 install images onto your computer.

Installing on a Windows platform

The Windows installation program uses the InstallShield interface to guide you through the installation. To install any of the components, do the following:

1. Change to the directory containing the downloaded install images.
2. At the command line, type setup.exe.
3. Click **Next** when the **Welcome** window displays. The **Software License Agreement** window is displayed.
4. Click **Yes**. The **Choose Destination Location** window displays.

   Note: This window does not display during subsequent installs on the same machine.
5. The **Choose Destination Location** window lets you override the default installation path, `c:\pkix`.

   Note: If you choose a path other than the default, you must make those path changes in the .ini files later in the installation process.

   Browse lets you select from the local or remote drives and paths available.

   Click **Next** to accept the drive and path specified.
6. When the **Select Components** window displays, select a checkbox for each of the components listed in the **Components** field you want to install:
   - **PKIX Certificate Authority**
   - **PKIX Registration Authority**
   - **PKIX End Entity**
   - **PKIX Documentation**

- **PKIX Software Developer's Kit**

7. Click **Next**.
8. When the **Select Program Folder** window displays, specify the folder, if other than the default **IBM SecureWay X.509 Public Key Infrastructure** folder, in which the components are to be installed.
9. Click **Next** to accept the entry.
10. The **Installing** window displays. The **Installing** window is a progress indicator of files being copied to the target drive and folder.
11. When the **Install addins?** prompt displays, specify either **Yes** or **No**.

    If you specify **Yes**, the addins are automatically configured in the registry. If you specify **No**, you are notified that you must configure the addins using the **install_addins <***pathname***>** command (where <*pathname*> is the directory where the PKIX DLLs are installed) before you start any PKIX programs.

    Note: This prompt does not display if only PKIX documentation is installed.
12. Upon completion, the **Setup Complete** window displays. Click **Finish** to exit.

Upon completing the installation, you can begin configuring .ini files for the various components. This process in discussed in "Chapter 6. Configuring the .ini files" on page 17.

Command-line installation

You can also perform the installation process using a command-line interface. The command line statement has the following syntax:

```
<path>setup <parms>
```

where *path* is the drive and path information where setup.exe resides and *parms* is a required argument with the following options:

/SMS or –SMS
: Prevents a network connection and setup.exe from closing before the setup is complete.

/z or –z
: Prevents setup.exe from checking the available memory during initialization. This parameter is necessary when running a setup on a system with more than 256MB of memory. If omitted, setup.exe reports insufficient memory and exits.

/f2 or –f2:*<log>*
: Specifies an alternate path and name for the log file, where *log* is the fully-qualified path and file name information. If this parameter is omitted, the path defaults to C:\pkix\logs\UnPKIX.isu.

Installing on an AIX platform

Use the following procedure to install the various PKIX components on an AIX system:

1. Log in to the AIX system as **root**.
2. From the command line, enter the following command to start the SMIT interface:

   ```
   smit
   ```
3. Click **Software Installation and Maintenance**.
4. Click **Install and Update Software**.

5. Click **Install and Update From All Available Software**.
6. At **INPUT device/directory for software**, type the fully-qualified path to the downloaded install images in the **Entry Fields** area, and press Enter.
7. At **SOFTWARE to install**, select the **List** button to view a list of the filesets available for installation.

   You can choose to install:

Component	Installation Choices	Filesets
PKIX Runtime Environment	• SecureWay Public Key Application Development Kit • SecureWay Public Key Runtime Environment	• pkix.core
PKIX Certificate Authority	• SecureWay Public Key Certificate Authority Client Admin • SecureWay Public Key Certificate Authority	• pkix.ca
PKIX Registration Authority	• SecureWay Public Key Registration Authority Client Admin • SecureWay Public Key Registration Authority	• pkix.ra
PKIX End Entity	SecureWay Public Key End Entity	• pkix.ee
PKIX Documentation	SecureWay Public Key Documentation Adobe Acrobat Format	• pkix.doc

8. Click the filesets (listed above) that you want to install and press **Enter** to run the installation program.
9. After the software has been installed, exit the SMIT interface.

Repeat these installation procedures for each entity you install.

Upon completing the installations, you can begin configuring .ini files for the various components. This process in discussed in “Chapter 6. Configuring the .ini files” on page 17.

Chapter 4. Setting up the LDAP server

Lightweight Directory Access Protocol (LDAP) and the correct LDAP schema file are prerequisites for the RA. The LDAP support enables the RA server to post certificate revocation lists (CRLs) and other information about registered users and services. This section describes the steps you must perform to set up the LDAP server to run with PKIX. Any LDAP client support is permissible; however, the following sections describe IBM LDAP support requirements.

Note: This setup activity must be completed before issuing the **initsc** command on the RA as instructed in "Configuring the RA foreground server .ini file" on page 23.

Setting up LDAP 3.1

For IBM LDAP 3.1, perform the following steps:

1. On the LDAP server, using the Directory Management Tool (DMT), create an organization and country suffix entry appropriate to your organization:

   o=*<Organization1>* c=*<Country>*

2. Add the following entries to the LDAP server using the Directory Management Tool:

```
objectclass pkiUser
    requires
        objectClass
    allows
        userCertificate
        CN
        mail

objectclass pkiCA
    requires
        objectClass
    allows
        cACertificate,
        certificateRevocationList,
        authorityRevocationList,
        crossCertificatePair
        CN
        OU
        mail
```

   To add these entries, do the following:

3. Select **object classes–>add object class**.
4. Create the pkiUser object class.
5. Click **Optional attributes**.
6. Select an allowable attribute and click **Add**. Do this for each attribute you specify.
7. Click **OK**.
8. Perform step 3 through step 7, creating the object class for pkiCA and its allowable attributes.
9. If you want users to be able to access the attributes of the pkiCA and pkiUser objects without authentication to the LDAP server, change the attributes from critical to normal. Using the Directory Management Tool, select **attributes–> edit attributes–> IBM extensions** and change the security class to normal for the following attributes:

- userCertificate
- cACertificate
- certificateRevocationList
- authorityRevocationList
- crossCertificatePair

10. Shutdown and restart the LDAP server.

Setting up LDAP 2.1

For IBM LDAP 2.1, perform the following steps:

1. On the LDAP server, using a text editor, create the file slapd.oc.pkix in the /etc directory and include the entries that follow. This creates the necessary PKIX object classes:

```
objectclass pkiUser
    requires
        objectClass
    allows
        userCertificate
        CN
        mail

objectclass pkiCA
    requires
        objectClass
    allows
        cACertificate,
        certificateRevocationList,
        authorityRevocationList,
        crossCertificatePair
        CN
        OU
        mail
```

2. Using a text editor, add the following statement to the bottom of the slapd.oc.conf file in the /etc directory:

   ```
   include /etc/slapd.oc.pkix
   ```

3. Change all occurrences of “crossCertificationPair” to “crossCertificatePair”.
4. To read the entries in LDAP without authentication, change the critical entries in slapd.at.conf to normal for the following attributes:
   - userCertificate
   - cACertificate
   - certificateRevocationList
   - authorityRevocationList
   - crossCertificatePair
5. Using browser-based Web administration, create an organization and country suffix entry appropriate to your organization:

   o=*<Organization1>* c=*<Country>*

6. Shut down and restart the LDAP server.

Chapter 5. Installing the PKIX APIs

PKIX provides a set of C and C++ APIs that enable applications to create and manage the complete certificate life cycle. These APIs offer all the functionality offered through the Java GUI interface. A set of the JNH APIs also have Java wrappers so that they can be called from a Java program.

Also included with these APIs are header files, code samples, and libraries.

During the PKIX installation process, you can optionally install the APIs, which are available for Windows NT and AIX platforms.

The *IBM SecureWay X.509 Public Key Infrastructure for Multiplatforms Programming Guide and Reference* contains the documentation for the PKIX APIs along with code samples to assist you during application development.

Chapter 6. Configuring the .ini files

During installation, the CA, RA, and EE, along with the remote RA and CA administrator GUIs are installed, each with its own .ini file. The .ini file contains entries that specify the operational characteristics of each entity. Default values are specified in the .ini file and the values defined depend on the type of entity being installed.

Unlike the install process, you must configure these entities in a specific sequence. You must configure the CA first, followed by the RA, and then the EE.

An .ini file editor, IniEditor, is provided to assist you when making modifications to .ini files. Refer to “Chapter 8. Using the .ini file editor” on page 39 for details on IniEditor. Similarly, PKIX provides bootstrap wizards, which also configure .ini files. The bootstrap wizards are available for use only when an entity has not been enrolled or initialized.

This chapter guides you through steps required to configure the .ini files for the following entities:

- CA foreground server
- CA administrator GUI
- RA foreground server
- RA administrator GUI
- EE

Note: Successful operation of PKIX is largely dependent on the settings in the .ini files. Do not modify the settings unless you know what you are doing. The results are unpredictable if erroneous settings are specified.

A general description of the .ini file and its contents is presented in “.Ini file description” on page 47.

Configuring the CA foreground server .ini file

During the PKIX installation, the CA foreground server’s *.ini file template, caserver.ini, is installed with default values. The default caserver.ini file parameters are detailed in Table 2 on page 53. The default CA .ini file name is jonahca.ini.

Following installation, you must use the IniEditor to create the CA foreground server’s *.ini file, based on the caserver.ini template, and update the following parameters to customize the CA server for the PKIX system:

Note: Before making any .ini file changes, copy the appropriate .ini file template you require and then open the newly-created file with IniEditor, making changes as required on the copy.

1. From the command line, start IniEditor and open the .ini file:
   - For Windows, type `IniEditor` *`myfile.ini`*
   - For AIX, type `run_IniEditor` *`myfile.ini`*

   where *`myfile.ini`* is the fully-qualified path and file name of the .ini file.
2. Update [CertPolicy] and [CrossCertPolicy] PolicyName1= with the name of the policy.

3. In the [OIDs] section, create an entry for the name of the policy and its associated OID. For example, if the [CertPolicy] or [CrossCertPolicy] PolicyName1= value is set to *MyPolicy*, create an entry with the name *MyPolicy* in the [OIDs] section with the correct OID value set.
4. Update [CertPolicy] and [CrossCertPolicy] Policy1Org= with the name of the organization.
5. Update [CertPolicy] and [CrossCertPolicy] UserNoticeText1= with the legal statement for relying parties to read.
6. Update [CertPolicy] and [CrossCertPolicy] CPS1= with the URL where the company's policy statement can be found.
7. Select **Save** from the **IniObject** menu to save the modifications made to the CA server's new .ini file, then select **Exit** to exit IniEditor.
8. From the command line, type the following:

   ```
   initsc –e CA –i path
   ```

   where *path* is the fully-qualified path name to the .ini file. This command is used to initialize the *.ini file, set up the smart card for the CA server, and store the security officer and user PINs into the CA server's smart card.

   The syntax of the **initsc** command is as follows:

   ```
   initsc arg1 entity arg2 inifname
   ```

   arg1 Entity flag which can be specified as either –**e**, –**E**, **/e**, or **/E**.

   entity The entity whose smart card is to be initialized. Specify one of the following:

   CA Certificate authority

   CC CA administrator GUI

   RA Registration authority

   RC RA administrator GUI

   EE End entity

   Note: The *entity* can be specified as upper- or lower-case letters.

   arg2 The .ini file flag which can be specified as either –**i**, –**I**, **/i**, or **/I**.

   path The fully-qualified path name to the .ini file.
9. At the prompt, "**Enter Security Officer PIN:**", enter the password.
10. At the prompt, "**Enter User PIN:**", enter the password.

The next step is to start the CA foreground server where the bootstrap wizard will guide you through additional CA configuration steps. See "Starting the CA foreground server" for details.

Starting the CA foreground server

To start the CA foreground server, do the following:

1. Issue the following from the command line:
   - For Windows NT, type **jonahca** *inifname*
   - For AIX, type **run_ca** *inifname*

where *inifname* is the fully-qualified path name to the .ini file. If omitted, the default .ini file name is used. For Windows, the default is `c:\pkix\jonahca.ini`. For AIX, the default is `/usr/pkix/etc/jonahca.ini`.

2. Click **Next** when the **CA Initialization: Naming CA** window appears.

   Note: As as general rule, it is recommended that you take the default value for all settings.

3. Enter a **Distinguished Authority** name in the **Authority Name** field. The pull-down menu beside **Authority Name** contains the following entries:

   Distinguished Name (X500)

   Server Name

   DNS

   Click **Properties** to see the list of options available:

4. Select an entry from **Available Object Classes**. The entries listed are:

   Country

   Organization

   Organization Unit

   Common Name

5. Click **Add**.
6. Type the value in the **Value** field. For example, for Country enter **US**. This example shows the Country as the United States.
7. Repeat step 4 through step 6 for each object class listed.
8. Click **Calculate** after you have defined each object class. The entry containing all the values you defined displays in the **Distinguished Name** field.
9. Click **OK**.
10. Click **Next**.
11. Enter a valid host name and port number in the **Host Name** and **Port** fields respectively.
12. Click **Next**.
13. Enter the **Certificate Validity** information:

    Start Date
    The date the certificate goes into effect.

    Expiration Duration
    The period of time in which this certificate is valid. The value can be specified in months, days, or years.

    End Date
    This field shows the computed certificate expiration date.

14. Click **Next**.
15. Select **Key Type**:

    CA Generated Key

    EE Generated (Local) Key

    Imported (External) Key

    Imported Existing Credential Key

16. Select **EE Generated Key Information**:

    Algorithm
    Specify **rsaEncryption**.

Key Length
Specify either **512** or **1024**.

17. Select **Key Usage**:
    Signing Key Repudia. . .
    Key Enciphering
    Data Enciphering
    Key Agreement
    Enciphering
    Deciphering
18. Click **Next**.
19. Select the class from the **Certificate Policy Class Selection** field.
20. Type the certificate practices statement in the **Certificate Practices Statement** field.
21. Click **Next**.
22. Enter a value in the **Maximum Subtree Path Length** field.
23. Enter a base name in the **Base Name** field. This field can be left blank.
24. Click **Next**.
25. In the **Certificate Authority Password** field, enter the user PIN used to initialize the CA smart card.
26. Click **Next**.
27. Enter the path name in the **CA Information** field. The default is c:\pkix\data\ca.info.
28. Click **Next**.
29. When completed, a window indicating bootstrap completion is displayed.
30. Click **Finish** to exit.

Adding a CA administrator

If you are using the sample implementation of the background CA server and a separate remote CA administrator GUI, the next step of the CA initial configuration is to create a CA administrator using the **addCAAdm** command. If you are using the foreground CA server, you may skip this section and continue with “Configuring the RA foreground server .ini file” on page 23.

The **addCAAdm** command updates the *.ini file to change the [RemoteServer] NumAdmins= value to 1 and add the [RemoteServer] Admin1DN= entry with its correct value. This command also generates a PKCS12 file which is transferred through a secure channel (such as password-protected electronic mail) for use in initializing the CA’s administrator GUI.

The syntax of the **addCAAdm** command is as follows:

```
addCAAdm /mode [/A adminDN] [/F pkcs12-file] [/I inifname]
```

mode The *mode* is a required parameter and can be one of the following:

/N Add a new administrator to the server. The *adminDN* and *pkcs12-file* parameters must be specified.

/E Add an existing administrator to the server (for example, an administrator previously created at a different server). The *pkcs12-file* parameter must be specified.

/R Remove an administrator from a server. The *adminDN* parameter must be specified.

/C Add an administrator to the client smart card.

adminDN
The distinguished name (DN) of the administrator. If omitted, you are prompted for this information.

pkcs12-file
The file name and path of the PKCS12 file. If omitted, you are prompted for this information.

inifname
The file name and path of the .ini file. For the /N, /E, and /R options, *inifname* should match the .ini file with which the server is running, or the .ini file of the client GUI if the /C option is specified.

Notes:

1. The background CA server must be running before issuing the **addCAAdm** command.
2. **addCAAdm /N** must be run to add the first administrator for the CA server after the CA bootstrap wizard has run and **casrvr** has been started.
3. **addCAAdm /C** must be run at the CA client machine after ″initsc –E CC . . ″ and before the client CA GUI can connect to the server.

Starting the CA background server

Note: This section is necessary only if you are using the sample implementation of the background CA server and a remote administrator GUI.

Using the **casrvr** command, the CA server can be started now or later but it must be started before the initial startup of the CA administrator GUI. The administrator PIN (refrerred to as “user PIN” when using the initsc command) saved in the initialization of the CA server’s smart card is required information to start the CA server.

The syntax for the **casrvr** command is as follows:

```
casrvr  [/A adminport] [/I inifname] [/P sc_pin | /Q(uiet)] [/S(top)]
```

/A *adminport*
Indicates the port number that the server listens to for administrator connections. If *adminport* is not specified, the server listens to the port number specified on the [RemoteServer] AdminPort= parameter. If that entry is omitted, the default port number is 1831.

/I *inifname*
Specifies the name and path of the .ini file. If omitted, the default CA file name and path is used.

/P *sc_pin*
The PIN for CA server’s smart card. If omitted, the user is prompted for the PIN or alternatively the PIN is read quietly from stdin if the /Q parameter is specified.

/Q Specifies that the PIN for the CA server’s smart card is to be read from stdin.

/S Specifies to cleanly discontinue this CA server’s operation.

Configuration continues with the CA administrator GUI initial configuration. This process is discussed in the next section.

Configuring the CA administrator GUI .ini file

Note: This section is necessary only if you are using the sample implementation of the background CA server and a remote administrator GUI.

During the PKIX installation, the CA's administrator GUI *.ini file template, caadmin.ini, is installed with default values. The default caadmin.ini parameters are detailed in Table 3 on page 56. The default CA administrator GUI .ini file name is jonahcc.ini.

Following installation, you must use the IniEditor to create the CA's administrator GUI *.ini file, based on the caadmin.ini template, and update the following parameters to customize the CA administrator GUI for the PKIX system:

Note: Before making any .ini file changes, copy the appropriate .ini file template you require and then open the newly-created file with IniEditor, making changes as required on the copy.

1. From the command line, start IniEditor and open the .ini file:
   - For Windows, type `IniEditor` *`myfile.ini`*
   - For AIX, type `run_IniEditor` *`myfile.ini`*

   where *`myfile.ini`* is the fully-qualified path and file name of the .ini file.
2. Update [RemoteServer] Server1DN= with the distinguished name (DN) of the CA background server.
3. Update [RemoteServer] Server1IPAddress= with the correct IP address of the CA background server.

   The other parameters can be updated as required.
4. Select **Save** from the **IniObject** menu to save the modifications made to the CA administrator GUI's new .ini file, then select **Exit** to exit IniEditor.
5. From the command line, type the following:

   ```
   initsc –e CC –i path
   ```

   where *`path`* is the fully-qualified path name to the .ini file. This command is used to initialize the *.ini file, set up the smart card for the CA server, and store the security officer and user PINs into the CA administrator GUI's smart card.

   The syntax of the **initsc** command is as follows:

   ```
   initsc arg1 entity arg2 inifname
   ```

   arg1 Entity flag which can be specified as either –**e**, –**E**, **/e**, or **/E**.

   entity The entity whose smart card is to be initialized. Specify one of the following:

   CA Certificate authority

   CC CA administrator GUI

   RA Registration authority

   RC RA administrator GUI

   EE End entity

Note: The *entity* can be specified as upper- or lower-case letters.

arg2 The .ini file flag which can be specified as either –**i**, –**I**, **/i**, or **/I**.

path The fully-qualified path name to the .ini file.

6. At the prompt, "**Enter Security Officer PIN:**", enter the password.
7. At the prompt, "**Enter User PIN:**", enter the CA administrator password.

Before continuing, you must start the CA background server using the **casrvr** command. Refer to "Starting the CA background server" on page 21 for details.

The **StartCAAdmin** command starts the CA's administrator GUI. The PKCS12 file generated when adding the CA administrator is a required input. Refer to "Adding a CA administrator" on page 20 for details.

Upon successful completion of the **StartCAAdmin** command, the CA server and administrator GUI are configured and ready for the first RA to enroll. The only functions the CA can perform at this time are RA enrollment and display fingerprint. Configuration continues with the initial configuration of the RA server.

Configuring the RA foreground server .ini file

During the PKIX installation, the RA foreground server's *.ini file template, raserver.ini, is installed with default values. The default raserver.ini file parameters are detailed in Table 4 on page 57. The default RA .ini file name is jonahra.ini.

Following installation, you must use the IniEditor to create the RA foreground server's *.ini file, based on the raserver.ini template, and update the following parameters to customize the RA background server for the PKIX system:

Note: Before making any .ini file changes, copy the appropriate .ini file template you require and then open the newly-created file with IniEditor, making changes as required on the copy.

1. From the command line, start IniEditor and open the .ini file:
   - For Windows, type `IniEditor` *`myfile.ini`*
   - For AIX, type `run_IniEditor` *`myfile.ini`*

   where *`myfile.ini`* is the fully-qualified path and file name of the .ini file.
2. Update [CertPolicy] and [CrossCertPolicy] PolicyName1= with the name of the policy.
3. In the [OIDs] section, create an entry for the name of the policy and its associated OID. For example, if the [CertPolicy] PolicyName1= value is set to *MyPolicy*, create an entry with the name *MyPolicy* in the [OIDs] section with the correct OID value set.
4. Update [CertPolicy] and [CrossCertPolicy] Policy1Org= with the name of the organization.
5. Update [CertPolicy] and [CrossCertPolicy] UserNoticeText1= with the legal statement for relying parties to read.
6. Update [CertPolicy] and [CrossCertPolicy] CPS1= with the URL where the company's policy statement can be found.
7. Select **Save** from the **IniObject** menu to save the modfications made to the RA server's new .ini file, then select **Exit** to exit IniEditor.

Note: Before continuing this procedure, the LDAP schema for PKIX must be installed on the LDAP server. Refer to “Chapter 4. Setting up the LDAP server” on page 13 for instructions on how to install the LDAP schema. This procedure can be done at any time before this step, however, you must do it before issuing the **initsc** command on the RA.

Once you have defined the LDAP schema and restarted the LDAP server, do the following:

1. From the command line, type:

   ```
   initsc –e RA –i path
   ```

   where *path* is the fully-qualified path name to the .ini file. This command is used to initialize the *.ini file, set up the smart card for the RA background server, and store the security officer and user PINs into the RA server's smart card.

   The syntax of the **initsc** command is as follows:

   ```
   initsc arg1 entity arg2 inifname
   ```

   arg1 Entity flag which can be specified as either –**e**, –**E**, **/e**, or **/E**.

   entity The entity whose smart card is to be initialized. Specify one of the following:

   CA Certificate authority

   CC CA administrator GUI

   RA Registration authority

   RC RA administrator GUI

   EE End entity

   Note: The *entity* can be specified as upper- or lower-case letters.

   arg2 The .ini file flag which can be specified as either –**i**, –**I**, **/i**, or **/I**.

   path The fully-qualified path name to the .ini file.
2. At the prompt, “**Enter Security Officer PIN:**”, enter the password.
3. At the prompt, “**Enter User PIN:**”, enter the password.

The next step is to start the RA foreground server where the bootstrap wizard will guide you through additional RA configuration steps. See “Starting the RA foreground server” for details.

Starting the RA foreground server

To start the RA foreground server, do the following:

1. Issue the following from the command line:
   - For Windows NT, type **jonahra** *inifname*
   - For AIX, type **run_ra** *inifname*

   where *inifname* is the fully-qualified path name to the .ini file. If omitted, the default .ini file name is used. For Windows, the default is `c:\pkix\jonahra.ini`. For AIX, the default is `/usr/pkix/etc/jonahra.ini`.
2. Click **Next** when the **RA Initialization: Naming RA** window appears.

   Note: As as general rule, it is recommended that you take the default value for all settings.

3. Enter a **Distinguished Authority** name in the **Authority Name** field. The pull-down menu beside **Authority Name** contains the following entries:

   Distinguished Name (X500)

   Server Name

   DNS

   Click **Properties** to see the list of options available:
4. Select an entry from **Available Object Classes**. The entries listed are:

   Country

   Organization

   Organization Unit

   Common Name
5. Click **Add**.
6. Type the value in the **Value** field. For example, for Country enter **US**. This example shows the Country as the United States.
7. Repeat step 4 through step 6 for each object class listed.
8. Click **Calculate** after you have defined each object class. The entry containing all the values you defined displays in the **Distinguished Name** field.
9. Click **OK**.
10. Click **Next**.
11. Enter a valid host name and port number in the **Host Name** and **Port** fields respectively.
12. Click **Next**.
13. Enter the **Certificate Validity** information:

    Start Date
    The date the certificate goes into effect.

    Expiration Duration
    The period of time in which this certificate is valid. The value can be specified in months, days, or years.

    End Date
    This field shows the computed certificate expiration date.
14. Click **Next**.
15. Select **Key Type**:

    CA Generated Key

    EE Generated (Local) Key

    Imported (External) Key

    Imported Existing Credential Key
16. Select **EE Generated Key Information**:

    Algorithm
    Specify **rsaEncryption**.

    Key Length
    Specify either **512** or **1024**.
17. Select **Key Usage**:

    Signing Key Repudia. . .

    Key Enciphering

    Data Enciphering

    Key Agreement

Enciphering

Deciphering

18. Click **Next**.
19. Select the class from the **Certificate Policy Class Selection** field.
20. Type the certificate practices statement in the **Certificate Practices Statement** field.
21. Click **Next**.
22. Type the LDAP server name in the **Server Name** field.
23. Type the LDAP administrator's Distinguished Name in the **Authorization Name** field.

    Note: You must include the slash ("/") in the entry. For example, /CN=root.
24. Type the LDAP administrator's password in the **Authorization Password** field.
25. Click **Next**.
26. Type the user PIN for the RA's smart card in the **Registration Authority Password** field.
27. Click **Next**.
28. When completed, a window indicating bootstrap completion is displayed.
29. Click **Finish** to exit.

Enrolling the RA with the CA

The next step is enrolling the RA with the CA. To enroll the RA with the CA, do the following:

1. Issue the following from the command line at the CA:
   - For Windows NT, type **jonahca** *inifname*
   - For AIX, type **run_jonahca** *inifname*

   where *inifname* is the fully-qualified path name to the .ini file. If omitted, the default .ini file name is used. For Windows, the default is `c:\pkix\jonahca.ini`. For AIX, the default is `/usr/pkix/etc/jonahca.ini`.
2. Type the user PIN for the CA's smart card in the **Password:** field.
3. The CA GUI displays.
4. Issue the following from the command line at the RA:
   - For Windows NT, type **jonahra** *inifname*
   - For AIX, type **run_jonahra** *inifname*

   where *inifname* is the fully-qualified path name to the .ini file. If omitted, the default .ini file name is used. For Windows, the default is `c:\pkix\jonahra.ini`. For AIX, the default is `/usr/pkix/etc/jonahra.ini`.
5. Type the user PIN for the RA's smart card in the **Password:** field.
6. The RA GUI displays.
7. At the RA, select **Actions–>Enroll with CA**. The **Registration Authority Enrollment** window displays.
8. Select **File** from the **Information Location** field.
9. In the **Location Identifier** field, specify the information from the CA bootstrap. The default is c:\pkix\data\ca.info
10. Click **Next**.
11. To specify the fingerprint information in the **Issuer Fingerprint** field:
    a. At the CA, select **Actions–>Display Fingerprint**.

b. Select the fingerprint that corresponds with the **Serial Number** field.
c. Mark the fingerprint information displayed.
d. Copy the text using the Ctrl-c key sequence.
e. At the RA, paste the information using the Ctrl-v key sequence in the **Fingerprint** field under **Issuer Fingerprint**.

Note: This procedure assumes the RA and CA reside on the same machine.

f. Click **Next**.
g. Select the appropriate certificate and click **Finish**.

At the RA, an entry displays in the **Certificates** tab indicating “EnrollActive”.

12. At the CA, click the **Certificates** tab. An entry displays in the **Certificates** tab indicating “EnrollActive”.
13. At the CA, approve the enrollment by selecting the enrollment request, then clicking **Approve**.

    At the CA, the **Certificate Authority Input RA Fingerprint** window displays
14. In the **RA URL** field, specify the RA’s URL. The default is pkix://localhost:829.
15. At the CA, specify the RA’s fingerprint in the **Fingerprint** field. To specify the fingerprint information in the Fingerprint field:
    a. At the RA, select **Actions–>Display Fingerprint**.
    b. Select the fingerprint that corresponds with the **Serial Number** field.
    c. Mark the fingerprint information displayed.
    d. Copy the text using the Ctrl-c key sequence.
    e. At the CA, paste the information using the Ctrl-v key sequence in the **Fingerprint** field under **Issuer Fingerprint**.

    Note: This procedure assumes the RA and CA reside on the same machine.
16. Click **OK**.
17. The status at the CA indicates “EnrollSigned”. The status at the RA indicates “EnrollCA Approved”.

The RA is now enrolled with the CA.

Adding a RA administrator

If you are using the sample implementation of the background RA server and a separate remote RA administrator GUI, the next step of the RA initial configuration is to create an RA administrator using the **addRAAdm** command. If you are using the foreground RA server, you can skip this section and continue with “Configuring the end entity .ini file” on page 30.

The **addRAAdm** command updates RA server’s *.ini file, changing the [RemoteServer] NumAdmins= value to 1 and supplies the [RemoteServer] Admin1DN= entry with its correct value. This command also generates a PKCS12 file for use in initializing the RA’s administrator GUI, which is transferred through a secure channel (such as password-protected electronic mail).

The syntax of the **addRAAdm** command is as follows:

`addRAAdm /mode [/A adminDN] [/F pkcs12-file] [/I inifname]`

mode The *mode* is a required parameter and can be one of the following:

/N Add a new administrator to the server. The *adminDN* and *pkcs12-file* parameters must be specified.

/E Add an existing administrator to the server (for example, an administrator previously created at a different server). The *pkcs12-file* parameter must be specified.

/R Remove an administrator from a server. The *adminDN* parameter must be specified.

/C Add an administrator to the client's smart card.

adminDN
The distinguished name (DN) of the administrator. If omitted, you are prompted for this information.

pkcs12-file
The file name and path of the PKCS12 file. If omitted, you are prompted for this information.

inifname
The file name and path of the .ini file. For the /N, /E, and /R options, *inifname* should match the .ini file with which the server is running, or the .ini file of the client GUI if the /C option is specified.

Notes:

1. The background RA server must be running before issuing the **addRAAdm** command.
2. **addRAAdm (/N or /E)** must be run to add the first administrator for the RA server after the RA bootstrap wizard has run, and **rasrvr** has been started.
3. To run **addRAAdm /N**, the RA must be enrolled with the CA and the CA must be able to approve the certificate request created for the new RA administrator, that is, a CA GUI (foreground or remote) must be running.
4. **addRAAdm /C** must be run at the CA client after ″initsc -E RC. . .″ and before the client RA GUI can connect to the server

Starting the RA background server

Note: This section is necessary only if you are using the sample implementation of the background RA server and a remote RA administrator GUI.

Using the **rasrvr** command, the RA server can be started now or later but it must be started before the initial startup of the RA administrator GUI. The administrator PIN saved in the initialization of the RA server's smart card is required information to start the RA server.

The syntax for the **rasrvr** command is as follows:

```
rasrvr  [/A adminport] [/I inifname] [/P sc_pin | /Q(uiet)] [/S(top)]
```

/A *adminport*
Indicates the port number that the server listens to for administrator connections. If *adminport* is not specified, the server listens to the port number specified on the [RemoteServer] AdminPort= parameter. If that entry is omitted, the default port number is 1832.

/I *inifname*
Specifies the name and path of the .ini file. If omitted, the default RA file name and path is used.

/P *sc_pin*
The PIN for RA server's the smart card. If omitted, the user is prompted for the PIN or alternatively the PIN is read quietly from stdin if the /Q parameter is specified.

/Q Specifies that the PIN for the RA server's smart card is to be read from stdin.

/S Specifies to cleanly discontinue this RA server's operation.

Configuration continues with the RA administrator GUI initial configuration. This process is discussed in the next section.

Configuring the RA administrator GUI .ini file

Note: This section is necessary only if you are using the sample implementation of the background RA server and a remote RA administrator GUI.

During the PKIX installation, the RA's administrator GUI *.ini file template, raadmin.ini, is installed with default values. The default raadmin.ini parameters are detailed in Table 5 on page 61. The default RA administrator GUI .ini file name is jonahrc.ini.

Following installation, you must use the IniEditor to create the RA's administrator GUI *.ini file, based on the raadmin.ini template, and update the following parameters to customize the RA administrator GUI for the PKIX system:

Note: Before making any .ini file changes, copy the appropriate .ini file template you require and then open the newly-created file with IniEditor, making changes as required on the copy.

1. From the command line, start IniEditor and open the .ini file:
   - For Windows, type `IniEditor` *`myfile.ini`*
   - For AIX, type `run_IniEditor` *`myfile.ini`*

   where *`myfile.ini`* is the fully-qualified path and file name of the .ini file.
2. Update [RemoteServer] Server1DN= with the distinguished name (DN) of the RA background server.
3. Update [RemoteServer] Server1IPAddress= with the IP address of the RA background server.

   The other parameters can be updated as required.
4. Select **Save** from the **IniObject** menu to save the modifications made to the RA administrator GUI's new .ini file, then select **Exit** to exit IniEditor.
5. From the command line, type the following:

   ```
   initsc –e RC –i path
   ```

   where *`path`* is the fully-qualified path name to the .ini file. This command is used to initialize the *.ini file, set up the smart card for the RA administrator GUI, and store the security officer and user PINs into the RA administrator GUI's smart card.

   The syntax of the **initsc** command is as follows:

   ```
   initsc arg1 entity arg2 inifname
   ```

   arg1 Entity flag which can be specified as either –**e**, –**E**, **/e**, or **/E**.

entity The entity whose smart card is to be initialized. Specify one of the following:

CA Certificate authority

CC CA administrator GUI

RA Registration authority

RC RA administrator GUI

EE End entity

Note: The *entity* can be specified as upper- or lower-case letters.

arg2 The .ini file flag which can be specified as either –**i**, –**I**, **/i**, or **/I**.

path The fully-qualified path name to the .ini file.

6. At the prompt, "**Enter Security Officer PIN:**", enter the password.
7. At the prompt, "**Enter User PIN:**", enter the password.

Before continuing, you must start the RA background server using the **rasrvr** command. Refer to "Starting the RA background server" on page 28 for details.

The **StartRAAdmin** command is then used to start the RA's administrator GUI. The PKCS12 file, generated while adding the RA administrator, is a required input. Refer to "Adding a RA administrator" on page 27 for details. Upon successful completion of the **StartRAAdmin** command, the RA server and administrator GUI is configured and ready to accept requests from EEs and other CAs.

Configuring the end entity .ini file

During the PKIX installation, the EE's *.ini file template, eetemplate.ini, is installed with default values. The default eetemplate.ini file parameters are detailed in Table 6 on page 62. The default EE .ini file name is jonahee.ini.

Following installation, you must use the IniEditor to create the EE's *.ini file, based on the eetemplate.ini template, to customize the EE's *.ini file parameter values for the PKIX system:

Note: Before making any .ini file changes, copy the appropriate .ini file template you require and then open the newly-created file with IniEditor, making changes as required on the copy.

1. From the command line, start IniEditor and open the .ini file:
   - For Windows, type `IniEditor` *`myfile.ini`*
   - For AIX, type `run_IniEditor` *`myfile.ini`*

   where *`myfile.ini`* is the fully-qualified path and file name of the .ini file.
2. Update, as needed, any parameters values you require for customization.
3. Select **Save** from the **IniObject** menu to save the modifications made to the EE's new .ini file, then select **Exit** to exit IniEditor.
4. From the command line, type the following:

   ```
   initsc –e EE –i path
   ```

   where *`path`* is the fully-qualified path name to the .ini file. This command is used to initialize the *.ini file, set up the smart card for the EE, and store the security officer and user PINs into the EE's smart card.

The syntax of the **initsc** command is as follows:

`initsc` *`arg1 entity arg2 inifname`*

arg1 Entity flag which can be specified as either –**e**, –**E**, **/e**, or **/E**.

entity The entity whose smart card is to be initialized. Specify one of the following:

CA Certificate authority

CC CA administrator GUI

RA Registration authority

RC RA administrator GUI

EE End entity

Note: The *entity* can be specified as upper- or lower-case letters.

arg2 The .ini file flag which can be specified as either –**i**, –**I**, **/i**, or **/I**.

path The fully-qualified path name to the .ini file.

5. At the prompt, "**Enter Security Officer PIN:**", enter the password.
6. At the prompt, "**Enter User PIN:**", enter the password.

Starting the end entity

To start the end entity (EE), do the following:

1. Issue the following from the command line:
   - For Windows NT, type **jonahee** *inifname*
   - For AIX, type **run_ee** *inifname*

   where *inifname* is the fully-qualified path name to the .ini file. If omitted, the default .ini file name is used. For Windows, the default is `c:\pkix\jonahee.ini`. For AIX, the default is `/usr/pkix/etc/jonahee.ini`.
2. After the **End Entity Login** window displays, enter the user PIN in the **Password:** field.
3. Click **OK**.

   The EE GUI displays.

At this point the EE is ready to use.

Chapter 7. Using the PKIX GUIs

This chapter describes the Certificate Authority (CA), Registration Authority (RA), and End Entity (EE) GUIs, along with the CA and RA administrator GUIs, and the functions they provide.

Note: To save time when supplying information in input fields on the GUI panels, you can use the copy and paste feature. These features are helpful when supplying “fingerprint” information required on some of the GUI panels. To copy, press Ctrl-c. To paste, press Ctrl-v.

Using the CA GUI

The Certificate Authority (CA) GUI enables the administrator to authorize or refuse certificate or certificate revocation log (CRL) operations. This section describes the GUI panels that enable you to perform CA administrative tasks.

To start the CA GUI:

1. From the command line, type:
   - For Windows NT, **jonahca**. You can also double-click the icon.
   - For AIX, **run_ca**.
2. Enter the CA administrator password when prompted.

The CA GUI task bar presents the following options:

Jonah Options include:

- **CRL**—Create a certificate revocation list (CRL) for subsequent posting by the RA.
- **Exit**—Exit the GUI.

Edit Select **Certificate Requests** or **De-enrollment Requests**.

Certificate Requests. Options include:

- **Validate**—Validate certificate.
- **Approve**—Approve certificate.
- **Deny**—Deny certificate request.
- **Revert**—Revert to previous certificate state.
- **Save**—Save this certificate request.

De-enrollment Requests. Options include:

- **Delete**—Delete request for RA de-enrollment.

Actions

Options include:

- **Display Fingerprint**— Display unique certificate information.
- **De-enroll RA**— Request an RA no longer be enrolled.
- **Initialize**—Initialize the CA.
- **Store CA info**—Store information about this CA.
- **Create Cross-Certificate Request**—Create a cross-certificate request.

View Options include:

- **Windows Look**—View the window layout.

- **Motif Look**—View the motif.
- **Metal Look**—View the metal look.
- **Message Log**—View the CA message log. The log contains a maximum of 100 messages, beginning with the most recent entry.

Help Options include:

- **Package Reference**
- **About**

The GUI also has tab settings for certificates, revocations, and postings from which you can make selections.

Certificates tab settings

For certificates, the following tabs are presented:

General
Specify the subject name, the type of certificate, and the validity period.

Key Specify the key type and key usage.

Policy Specify the certificate policy class.

CA Constraints
Specify subtree path length and permitted subtrees.

Revocations tab settings

For revocations, the following tab is presented:

Revoke Data
View or specify information pertaining to a specific revocation request, such as requestor's name, reason for request, and distinguished name.

Postings tab settings

View the CA's pending CRL activity.

De-enrollments tab settings

For de-enrollments, the following tab is presented:

De-enrollment Data
View general information pertaining to de-enrollments.

Using the CA administrator GUI

The Certificate Authority (CA) administrator GUI enables the administrator to authorize or refuse certificate or certificate revocation log (CRL) operations from a remote system. Multiple administrators could be performing these operations by connecting to the same CA server. This section describes the GUI panels that enable you to perform CA administrative tasks from a remote system.

To start the CA administrator GUI:

1. From the command line, type **StartCAAdmin**. The command is the same for both Windows and AIX environments.
2. Enter the CA administrator keystore password. If a valid password is entered, the administrator information panel is displayed.
3. Select a certificate from the Certificate List Table.

4. Select the server distinguished name to which you want to connect from the server name pull-down menu or type the server distinguished name to connect to a new server. Selecting the server name populates the last known successful values for the server address and port for this server.
5. Click **OK** to connect to the CA server and start the administrative session.

Once the session begins, the GUI behaves as described in "Using the CA GUI" on page 33.

Using the RA GUI

The RA GUI enables you to monitor CA operations and control audit and archive settings. This section describes the GUI panels that enable you to perform RA administrative tasks.

To start the RA GUI:

1. From the command line, type:
   - For Windows NT, **jonahra**. You can also double-click the icon.
   - For AIX, **run_ra**.
2. Enter the RA administrator password when prompted.

The RA GUI task bar presents the following options:

Jonah **Exit** to exit the GUI.

Edit Select **Certificate Requests**, **Revoke Requests**, **Enrollment Requests**, or **De-enrollment Requests**.

Certificate Requests. Options include:

- **Validate**—Validate certificate.
- **Approve**—Approve certificate.
- **Deny**—Deny certificate request.
- **Revert**—Revert to previous certificate state.
- **Save**—Save this certificate request.

Revoke Requests. Options include:

- **Approve**—Approve revocation request.
- **Deny**—Deny revocation request.
- **New**—Create new revocation request.
- **Revert**—Revert to previous revocation state.
- **Save**—Save this revocation request.

Enrollment Requests. Options include:

- **Delete**—Delete request to enroll an RA.

De-enrollment Requests. Options include

- **Delete**—Delete request to de-enroll an RA.

Actions

Options include:

- **Enroll with CA**—Enroll the RA with a CA.
- **Display Fingerprint**—Display unique certificate information.
- **De-enroll from CA**—Request de-enrollment from a CA.
- **Preregister User**—Issue a preregistration request.

- **CA (Cross Cert)**—Request a cross certificate.

View Options include:

- **Windows Look**—View the window layout.
- **Motif Look**—View the motif.
- **Metal Look**—View the metal look.
- **Message Log**—View the RA message log. The log contains a maximum of 100 messages, beginning with the most recent entry.

Help Options include:

- **Package Reference**
- **About**

The GUI also has tab settings for certificates, revocations, and postings from which you can make selections.

Certificates tab settings

For certificates, the following tabs are presented:

General
Specify the subject name, the type of certificate, and the validity period.

Key Specify the key type and key usage.

Policy Specify the certificate policy class.

CA Constraints
Specify subtree path length and permitted subtrees.

Revocations tab settings

For revocations, the following tab is presented:

Revoke Data
View or specify information pertaining to a specific revocation request, such as requestor's name, reason for request, and distinguished name.

Postings tab settings

View the RA's pending certificate activity.

De-enrollments tab settings

For de-enrollments, the following tab is presented:

De-enrollment Data
View general information pertaining to de-enrollments.

Using the RA administrator GUI

The Registration Authority (CA) administrator GUI enables the administrator to monitor operations and control audit and archive settings from a remote system. Multiple administrators could be performing these operations by connecting to the same RA server. This section describes the GUI panels that enable you to perform RA administrative tasks from a remote system.

To start the RA administrator GUI:

1. From the command line, type **StartRAAdmin**. The command is the same for both Windows and AIX environments.

2. Enter the RA administrator keystore password. If a valid password is entered, the administrator information panel is displayed.
3. Select a certificate from the Certificate List Table.
4. Select the server distinguished name to which you want to connect from the server name pull-down menu or type the server distinguished name to connect to a new server. Selecting the server name populates the last known successful values for the server address and port for this server.
5. Click **OK** to connect to the RA server and start the administrative session.

Once the session begins, the GUI behaves as described in "Using the RA GUI" on page 35.

Using the EE GUI

The EE GUI enables you to initiate certificate and revocation requests, and maintain the certificates issued by PKIX.

To start the EE GUI:

1. From the command line, type:
   - For Windows NT, **jonahee**. You can also double-click the icon.
   - For AIX, **run_ee**.
2. Enter the EE administrator password when prompted.

The EE GUI task bar presents the following options:

Jonah **Exit** to exit the GUI.

Edit Select either **Certificate Requests** or **Revoke Requests**.

Certificate Requests. Options include:

- **Submit**—Submit certificate request.
- **Delete**—Delete the certificate request.
- **Revert**—Revert to previous certificate state.
- **Save**—Save this certificate request.

Revoke Requests. Options include:

- **Submit**—Submit revocation request.
- **Delete**—Delete revocation request.
- **Cancel**—Cancel revocation request.
- **Revert**—Revert to previous revocation state.
- **Save**—Save this revocation request.

Actions

Options include:

- **Create Certificate Request**—Create new certificate request.
- **Revoke**—Issue certificate revocation request.
- **Import**—Import a certificate request.
- **Export**—Export a certificate request.

View Options include:

- **Windows Look**—View the window layout.
- **Motif Look**—View the motif.
- **Metal Look**—View the metal look.

- **Message Log**—View the EE message log. The log contains a maximum of 100 messages, beginning with the most recent entry.

Help Options include:

- **Package Reference**
- **About**

Certificates Data tab settings

For certificates, the following tabs are presented:

General
Specify the subject name, the type of certificate, and the validity period.

Key Specify the key type and key usage.

Policy Specify the certificate policy class.

CA Constraints
Specify subtree path length and permitted subtrees.

Extensions
From this tab, you can add, modify, or remove certificate extensions.

Keystore Data tab settings

For keystore data, the following tab is presented:

Information
Lists the certificates that reside on the virtual smart card and shows general information about the certificates such as subject, issuer name, validity period.

Chapter 8. Using the .ini file editor

This section provides instructions on how to invoke and use the .ini file editor, IniEditor.

Overview

The IniEditor allows you to add, modify, and delete parameters and sections in the .ini file. The IniEditor displays each parameter and value pair in an editing field so you can easily locate and edit what you need.

Refer to “Chapter 6. Configuring the .ini files” on page 17 for detailed information on the .ini files.

Starting the IniEditor

You must have the Java™ Runtime Environment installed on your system in order to start the IniEditor.

You can start the IniEditor at the command prompt with or without an .ini file name specified.

- Starting IniEditor with an .ini file:
  - Windows:

    ```
    IniEditor myfile.ini
    ```
  - AIX:

    ```
    run_IniEditor myfile.ini
    ```
- Starting IniEditor without an .ini file name:
  - Windows:

    ```
    IniEditor
    ```
  - AIX:

    ```
    run_IniEditor
    ```

The following .ini file templates are installed when you perform the installation procedures described in “Chapter 3. Installing the PKIX components” on page 9.

CA Server	caserver.ini
CA Admin	caadmin.ini
RA Server	raserver.ini
RA Admin	raadmin.ini
End Entity	eetemplate.ini

Before making any .ini file changes, copy the appropriate .ini file template you require and then open the newly-created file with IniEditor, making changes as required on the copy.

Viewing the IniEditor

Once loaded, the .ini file appears on the left-hand side of your screen in a tree structure. This structure contains the .ini file's sections and keys. You select a section by clicking on it. To expand the section, click on the plus (**+**) sign. If a section contains keys, the parameter and value pair appear on the right-hand side of your screen in an editing field.

Editing parameters

To change a parameter's value, type over the text in the editing field. To undo the editing, select the **Undo** option under the **Edit** menu.

A parameter value can be deleted using the **Delete** option under the **Edit** menu.

A parameter and a section can be removed by selecting them in the tree structure and then selecting **Remove** under the **Object** menu.

Adding a new section

Add a new section to the .ini file by selecting the **New Section** menu item under the **Object** menu, and then specify the section's name in the **Add a New Section** dialog box.

To add a parameter or value to a new section, see ″Adding a new parameter″.

Adding a new parameter

Add a new parameter by selecting the **New Parameter** menu item under the **Object** menu.

The **Create a New Parameter** dialog box appears. This dialog box allows you to select the section to which the new parameter is added. The currently-selected section in the tree structure is the default. You are prompted to specify the parameter name and value.

Selecting **OK** causes the new parameter to appear in the tree structure under the selected section, and on the right-hand side of the window in an editing field.

Saving the .ini file

The **File** menu allows you to save the current .ini file or exit the program. If the .ini file has been changed, you are prompted to save the changes.

Chapter 9. Importing / exporting certificates for browsers

This chapter explains the various setup activities you must perform to enable PKIX certificates to interoperate with browsers. The discussion addresses setup for Netscape Navigator 4.51 and Microsoft Internet Explorer 5.

The following tasks are discussed:

- Importing a CA certificate from a Web server
- Reading a certificate from a virtual smart card
- Exporting an EE certificate from a virtual smart card
- Importing an EE certificate into a browser

This chapter assumes you have already exported a PKCS #12 certificate at the EE using the EE GUI. Refer to “Using the EE GUI” on page 37 for details.

At the end of this chapter, a procedure is given to enable Netscape Navigator to retrieve a certificate from LDAP based on electronic mail userids.

Setting up to use Netscape Navigator

This section addresses the setup activities you must perform when using the Netscape Navigator browser.

Importing a CA certificate from a Web server

There are several things that are required in order to import a CA certificate into Netscape Navigator from a Web server. First, the Web server must be set up to accept DER-encoded certificates. The next section gives an example of how to do this. Depending on how the Web server was installed, the directory may be different, but the name of the file you must edit and the entry you make should be the same as described in the following steps:

1. Using a text editor, open the file mime.types.

   Note: The mime.types file resides in the same directory in which the Web server code was installed.

2. Add the following statement to the bottom of the mime.types file:

   ```
   application/x-x509-ca-cert        der
   ```

3. Save the updated mime.types file and exit the text editor.
4. Shutdown and restart the Web server.
5. At the LDAP server, issue the **ldapsearch** command to retrieve the CA certificate information.

   Using the following command as an example, edit the command to reflect the logon and password of the LDAP server and the distinguished name of the certificate being retrieved:

   ```
   ldapsearch -D "cn=root" -w password -t -B -b "cn=CA830, ou=Jonah, o=IBM, c=US" "objectclass=*"
   ```

   The following is an example of the output from the ldapsearch command:

   ```
   CN=CA830,OU=Jonah,O=IBM,C=US
   objectclass=ldapsearch-objectclass-a00208
   objectclass=ldapsearch-objectclass-b00208
   cacertificate=ldapsearch-cacertificate-a00208
   cn=ldapsearch-cn-a00208
   certificaterevocationlist=ldapsearch-certificaterevocationlist-a00208
   authorityrevocationlist=ldapsearch-authorityrevocationlist-a00208
   ```

The statement that begins with cacertificate= specifies the name of the file that the ldapsearch command created with the certificate information.

6. Copy the file created by the ldapsearch command to the Web server:

   Note: The fully-qualified path information depends on how the Web server was installed.

   ```
   copy ldapsearch-cacertificate-a00208 j:\apache\htdocs\cacert.der
   ```

7. In the browser input area, load the CA certificate into Netscape Navigator. The location where the CA certificate is loaded is the actual URL of the Web server followed by the name of the file that was created and copied to the Web server in step 6. The following is a sample entry for the file created in step 6:

   ```
   http://webserver.address/cacert.der
   ```

   After the link is made, a **New Certificate Authority** window appears. Import the certificate as follows:

8. Click **Next** twice. The window has a **More info** button for viewing certificate information.
9. Click **Next** again.
10. Select all check boxes.
11. Click **Next** twice.
12. Type a nickname for the certificate.
13. Click **Finish**.

At this point, the browser now has the CA in its list of certificate signers.

You can verify the certificate by doing the following:

1. Click the **Security** icon in the Netscape Navigator window. A **Security Info** dialog appears.
2. Click **Signers**.
3. Find your certificate's nickname in the list and select it.
4. Click **Verify**.

Reading a certificate from a virtual smart card

The next step is to set up the browser to enable it to read certificates from a virtual smart card. Perform the following steps:

1. Click the **Security** icon in the Netscape Navigator window. A **Security Info** dialog appears.
2. Click **Cryptographic Modules**.
3. Click **Add**.
4. Type a nickname for the security module, for example "PKIX PKCS11 Library", in the **Security Module Name** field.
5. Type the name of the PKIX PKCS #11 library, including the fully-qualified path information. For Windows NT, the default is c:\pkix\pkcs11.dll. For AIX, the default is /usr/pkix/lib/libpkcs11.a.
6. Click **OK**.
7. Type the password for the virtual smart card. The name of the security module is added to the list.
8. Click **Yours**. A list of all the certificates in the system, including the certificates on the virtual smart card, is displayed.

Exporting an EE certificate from a virtual smart card

The next sequence of steps explains how to export an end entity certificate from a virtual smart card. The exported certificate is written to a file in PKCS #12 format.

To export the certificate, do the following:

1. Perform the steps listed in “Reading a certificate from a virtual smart card” on page 42.
2. Click the **Security** icon in the Netscape Navigator window. A **Security Info** dialog appears.
3. If prompted, specify the password for the virtual smart card.
4. Click **Yours** in the **Security Info** window.
5. Click on a certificate.
6. Click **Export**.
7. Type the password for the virtual smart card.
8. Type the password for encrypting the PKCS #12 file in the **Password Entry** dialog.
9. Type the password again to verify it.
10. Type the name with which the PKCS #12 file will be stored.
11. Click **OK**.

Importing an EE certificate into a browser

The next sequence of steps explains how to import a certificate, stored in a file in PKCS #12 format, into the browser.

To import the certificate, do the following:

1. Click the **Security** icon in the Netscape Navigator window.
2. Click **Yours** in the **Security Info** window.
3. Click **Import a Certificate...**.
4. If you have installed virtual smart card support, you are prompted to specify where the imported certificate is to be stored.
5. Find the PKCS #12 file that you want to import.
6. Click **Open**
7. Specify the password used when the file was exported.
8. Click **OK**.

At this point, the exported PKCS #12 file is now imported into the browser.

Setting up to use Internet Explorer

This section addresses the setup activities you must perform when using the Internet Explorer browser.

Importing a CA certificate from a Web server

There are several things that are required in order to import a CA certificate into Internet Explorer from a Web server. First, the Web server must be set up to accept DER-encoded certificates. The next section gives an example of how to do this. Depending on how the Web server was installed, the directory may be different, but the name of the file you must edit and the entry you make should be the same as described in the following steps:

1. Using a text editor, open the file mime.types.

Note: The mime.types file resides in the same directory in which the server code was installed.

2. Add the following statement to the bottom of the mime.types file:

   ```
   application/x-x509-ca-cert        der
   ```

3. Save the updated mime.types file and exit the text editor.
4. Shutdown and restart the Web server.
5. At the LDAP server, issue the **ldapsearch** command to retrieve the CA certificate information.

   Using the following command as an example, edit the command to reflect the logon and password of the LDAP server and the distinguished name of the certificate being retrieved:

   ```
   ldapsearch -D "cn=root" -w password -t -B -b "cn=CA830, ou=Jonah, o=IBM, c=US" "objectclass=*"
   ```

   The following is an example of the output from the ldapsearch command:

   ```
   CN=CA830,OU=Jonah,O=IBM,C=US
   objectclass=ldapsearch-objectclass-a00208
   objectclass=ldapsearch-objectclass-b00208
   cacertificate=ldapsearch-cacertificate-a00208
   cn=ldapsearch-cn-a00208
   certificaterevocationlist=ldapsearch-certificaterevocationlist-a00208
   authorityrevocationlist=ldapsearch-authorityrevocationlist-a00208
   ```

   The statement that begins with cacertificate= specifies the name of the file that the ldapsearch command created with the certificate information.
6. Copy the file created by the ldapsearch command to the Web server:

   Note: The fully-qualified path information depends on how the Web server was installed.

   ```
   copy ldapsearch-cacertificate-a00208 j:\apache\htdocs\cacert.der
   ```

7. In the browser input area, load the CA certificate into Internet Explorer. The location where the CA certificate is loaded is the actual URL of the Web server followed by the name of the file that was created and copied to the Web server in step 6. The following is a sample entry for the file created in step 6:

   ```
   http://webserver.address/cacert.der
   ```

   After the link is made, a **File Download** window appears. Import the certificate as follows:
8. Select **Open this file from its current location**.
9. Click **OK**. The **Certificate** window displays.
10. Click **Install Certificate...**. The **Certificate Manager Import Wizard** window displays.
11. Click **Next**.
12. Click **Next**.
13. Click **Finish**. The **Root Certificate Store** window displays.
14. Click **Yes**. A window indicating the import was successful displays.
15. Click **OK**.
16. Click **OK**.

At this point, the browser now has the CA in its list of certificate signers.

You can verify the certificate by doing the following:

1. In the browser input area, load the CA certificate into Internet Explorer. The location where the CA certificate is loaded is the actual URL of the Web server

followed by the name of the file that was created and copied to the Web server in step 6 on page 44. The following is a sample entry for the file created in step 6 on page 44:

```
http://webserver.address/cacert.der
```

2. Select **Open this file from its current location**.
3. Click **OK**. The **Certificate** window displays.
4. Click the **Certification path** tab. The **Certificate status** should indicate “This certificate is OK”.

Importing an EE certificate into a browser

The next sequence of steps explains how to import a certificate, stored in a file in PKCS #12 format, into the browser.

To import the certificate, do the following:

1. From the **Tools** menu, select **Internet Options**.
2. Click the **Contents** tab.
3. Click **Certificates...**.
4. Click **Import...**. The **Certificate Manager Import Wizard** window displays.
5. Click **Next**.
6. Find the PKCS #12 file that you want to import.
7. Click **Next**.
8. Specify the password used when the file was exported and click **Next**.
9. Click **Next**.
10. Click **Finish**. A window displays indicating the import was successful.

At this point, the exported PKCS #12 file is now imported into the browser.

Retrieving certificates through Netscape Navigator

This section describes how to add an mail attribute to a PKIX certificate entry in LDAP. This enables Netscape Navigator to retrieve the certificate from LDAP.

Perform the following steps at the LDAP server:

1. Using a text editor, create a file and enter the following statements:

```
dn: cn=cnname1,ou=ouname,o=oname,c=cname
changetype: modify
add: mail
mail: emailaddr

dn: cn=cnname2,ou=ouname,o=oname,c=cname
changetype: modify
add: mail
mail: emailaddr
```

   The example shows two entries. Multiple entries must have a blank line between them.
2. Save the file after all entries are made. You will specify this file in the next step.
3. From the command line, type:

```
ldapmodify -D "cn=root" -w password -h hostname -f filename
```

Appendix A. .Ini files

This appendix contains a description of the .ini files and their parameters. The .ini file templates along with the default values are also provided.

.Ini file description

An .ini file is divided into sections, each section beginning with a section header enclosed in brackets, for example **[section]**. Within a [section], there can be one or more statements of the form, “parameter=value”. The parameters are used to define values, which can be Boolean, integer, or strings. Boolean values use “T” or “F” to represent a true or false condition respectively. All strings are null-terminated strings.

“Interval”, a special case of string, has the format [*N* d], [*N* h], [*N* m], or [*N* s] where “*N*” is an integer and “d”, “h”, “m”, and “s” are literal characters that specify the units of the preceding integer and represent days, hours, minutes, and seconds respectively. Any blank space is ignored. Any unit specifiers that are present must occur in the correct sequence. The default unit is hours.

“IP Address”, another special case of string, has the format *w.x.y.z*, where each *w*, *x*, *y*, and *z* may be an integer between 0 and 255, inclusive.

For an object identifier (OID), the format consists of a sequence of non-negative integers separated by periods. The first number in the sequence should always be 0, 1, or 2.

.Ini file sections and parameters

Table 1 identifies all the sections and parameters that are part of the .ini files in the C2 release. The “Format” column identifies the expected format for the value of the parameter.

A parameter is defined as optional if there is a default value; otherwise, it is required. The “Config?” column indicates whether the parameter is configurable and any limitations on configuration.

Table 1. PKIX .ini file sections and parameters

Parameter	Use	Format	Req/Opt	Config?
[OIDs] (Common defaults)	Standards-based mechanism for uniquely identifying items			
C=2.5.4.6		OID	Required	No
O=2.5.4.10		OID	Required	No
OU=2.5.4.11		OID	Required	No
CN=2.5.4.3		OID	Required	No
L=2.5.4.7		OID	Required	No
S=2.5.4.8		OID	Required	No
T=2.5.4.12		OID	Required	No
id-dsa=1.2.840.10040.4.1		OID	Required	No

Table 1. PKIX .ini file sections and parameters (continued)

Parameter	Use	Format	Req/Opt	Config?
id-dsa-with-shal= 1.2.840.10040.4.3		OID	Required	No
rsaEncryption= 1.2.840.113549.1.1.1		OID	Required	No
sha-1WithRSAEncryption= 1.2.840.113549.1.1.5		OID	Required	No
sha1=1.3.14.3.2.26		OID	Required	No
hmac-sha1=1.3.6.1.5.5.8.1.2		OID	Required	No
pkcs7-data=1.2.840.113549.1.7.1		OID	Required	No
pkcs12-certbag= 1.2.840.113549.1.12.10.1.3		OID	Required	No
pkcs12-keybag= 1.2.840.113549.1.12.10.1.1		OID	Required	No
X509-Certificate= 1.2.840.113549.1.9.22.1		OID	Required	No
PasswordBasedMAC= 1.2.840.113533.7.66.13		OID	Required	No
MyPolicy=	Example of an OID entry for PolicyName1.	OID	Note 1	Yes
MyLitePolicy=	Example of an OID entry for PolicyName2.	OID	Note 1	Yes
[AsymmetricKeyAlgs]				
DSA=id-dsa		string	Required	No
RSA=rsaEncryption		string	Required	No
[AsymmetricEncAlgs]				
DSA=id-dsa		string	Required	No
[AsymmetricSigAlgs]				
DSAwithSHA1= id-dsa-with-sha1		string	Required	No
RSAwithSHA1= sha-1WithRSAEncryption		string	Required	No
[ObjectStore]				
Name=	File name without extension to be used for object store, BinBin, and scheduler.	string	Required	Yes
Path=	Fully-qualifed path where files will be placed.	string	Required	Yes, but only before init.
[CertPolicy] also known as **[Issuer*x*CertPolicy]** for RA	Each signature algorithm must have its OID declared in the OIDs section. Each policy name must have a corresponding OID in the OIDs section.			Note 3
SigAlg1=sha-1WithRSAEncryption	Name of signature algorithm. Must have corresponding entry in OIDs section.	string	Required	Yes

Table 1. PKIX .ini file sections and parameters (continued)

Parameter	Use	Format	Req/Opt	Config?
StartTimeSpecifiable=T	Can the requester (EE to RA or RA to CA) specify the start time?	T or F	Optional	Yes
MaxLifetime=8784h	What is the maximum lifetime of a certificate?	interval	Optional	Yes
LifeTimeDef=180d	Default certificate lifetime.	interval	Optional	Yes
KeySpecifiable=T	Can the requester (EE or RA) specify the subject's public key?	T or F	Optional	Yes
KeyUsageSupported=T	Is key usage supported?	T or F	Optional	Yes
KeyUsageRequired=T	Is key usage required?	T or F	Optional	Yes
PolicyCritical=F	Should the policy be critical?	T or F	Optional	Yes
PolicyRequired=F	Is a policy required?	T or F	Optional	No
PolicyName1=	What is the name of the primary policy? Must have corresponding OID in OIDs section.	string	Required	Yes
Policy1Org=	What organization requires this policy?	string	Required	Yes
Policy1Notice1=3		integer	Optional	Yes
Policy1Notice2=17		integer	Optional	Yes
UserNoticeText1=	Legal statement or disclaimer for users to read.	string	Optional	Yes
CPS1=	URL of where policy1's statement can be read.	string	Required	Yes
PolicyName2=	Secondary policy name. If present, must have corresponding OID in OIDs section.	string	Optional	Yes
CPS2=	URL of where policy2's statement can be read.	string	If PolicyName2 present, required	Yes
TimeBetweenCRLs=1d	Default time between scheduled CRL publication.	interval	Optional	Yes
CRLDuration=2d	CRL lifetime.	interval	Optional	Yes
EERevokeRequests=ANY	Can an EE request the revocation of a certificate? ANY: EE can request the revocation of any certificate; SELF: EE can request the revocation of a certificate it requested; NONE: EE cannot request the revocation of certificates.	string='ANY' or 'SELF' or 'NONE'	Optional	Yes
[CrossCertPolicy] also known as **[Issuer*x*CrossCertPolicy]** for RA	Each signature algorithm must have its OID declared in the OIDs section. Each policy name must have a corresponding OID in the OIDs section.			Note 3
SigAlg1=sha-1WithRSAEncryption	Name of signature algorithm. Must have corresponding entry in OIDs section.	string	Required	Yes
StartTimeSpecifiable=T	Can the requester (EE to RA or RA to CA) specify the start time?	T or F	Optional	Yes

Table 1. PKIX .ini file sections and parameters (continued)

Parameter	Use	Format	Req/Opt	Config?
MaxLifetime=8784h	What is the maximum lifetime of a certificate?	interval	Optional	Yes
LifeTimeDef=180d	Default certificate lifetime.	interval	Optional	Yes
KeySpecifiable=T	Can the requester (EE or RA) specify the subject's public key?	T or F	Optional	Yes
KeyUsageSupported=T	Is key usage supported?	T or F	Optional	Yes
KeyUsageRequired=T	Is key usage required?	T or F	Optional	Yes
PolicyCritical=F	Should the policy be critical?	T or F	Optional	Yes
PolicyRequired=F	Is a policy required?	T or F	Optional	Yes
PolicyName1=	What is the name of the primary policy? Must have corresponding OID in OIDs section.	string	Required	Yes
Policy1Org=	What organization requires this policy?	string	Required	Yes
Policy1Notice1=3		integer	Optional	Yes
Policy1Notice2=17		integer	Optional	Yes
UserNoticeText1=	Legal statement or disclaimer for users to read.	string	Optional	Yes
CPS1=	URL of where policy1's statement can be read.	string	Required	Yes
PolicyName2=	Secondary policy name. If present, must have corresponding OID in OIDs section.	string	Optional	Yes
CPS2=	URL of where policy2's statement can be read.	string	If PolicyName2 present, required	Yes
TimeBetweenCRLs=1d	Default time between scheduled CRL publication.	interval	Optional	Yes
CRLDuration=2d	CRL lifetime.	interval	Optional	Yes
EERevokeRequests=ANY	Can an EE request the revocation of a certificate? ANY: EE can request the revocation of any certificate; SELF: EE can request the revocation of a certificate it requested; NONE: EE cannot request the revocation of certificates.	string='ANY' or 'SELF' or 'NONE'	Optional	Yes
[General]				
MyName=	Distinguished name of the entity.	string	Required	Note 2
RA1=	Distinguished name of the first RA for this CA.	string	Required	Note 2
DefaultRA=	Which RA is the default RA?	integer	Required	Note 4
PreferredCryptoProvider= dda0c1e0-7b73-11d0 -8e0c-0004ac602b18	GUID (globally unique identifier) for the crypto provider.	string	Required	No
Issuer1=	Distinguished name of this RA's CA.	string	Required	Note 2
Issuer1URL1=	URL for this RA's CA.	string	Required	Note 2

Table 1. PKIX .ini file sections and parameters (continued)

Parameter	Use	Format	Req/Opt	Config?
CertperDP=0	Certificates per distribution point.	integer	Optional	Yes
CRLDistName=	Distribution point name to be put into certificate.	string ending with '%d'	Optional	Yes
TempPath=	Path for storage of temporary objects.	string	Required	Yes
PathToDLLs=	Path where PKIX *.dlls (on Windows NT) or *.a (on AIX) files are installed.	string	Required	Yes
[Transport]				
TCPPort=	TCP Port to listen on.	integer	Optional	Yes
TCPHost=	TCP/IP host name.	string	Optional	Yes
PollInterval=30s	Polling Interval.	interval	Optional	Yes
RetryInterval=1m		interval	Optional	Yes
MaxErrorCount=20		integer	Optional	Yes
[Keystore]				
CurKeyStore=VSC	Keystore in use. For the EE specify either VSC or SmartCard.	string=VSC or SmartCard	Required	No
[VSC]	Required section if CurKeyStore=VSC.			
InitialSOpw=SOPIN	Initial password for the virtual smart card.	string	Required	Yes, but only before init
TokenDir=	Fully-qualified path and file name for the virtual smart card.	string	Required	Yes, but only before init
Model=PKCS11_STORAGE _MODEL	Type of storage used.	string	Required	No
Guid=7f529c80-c942-11d1-8fb0-0004ac61389a	Global unique identifier.	string	Required	No
[SmartCard]	Required section if CurKeyStore=SmartCard.			
Model=PKCS11_STORAGE _MODEL	Type of storage used.	string	Required	No
Guid=EA8A7C01-13BC-11D3-B150 -002035C00173	Global unique identifier.	string	Required	No
LibraryName=	The fully-qualified path in which the smart card support DLL resides.	string	Required	Yes
[URLs]	Used by Keyworks to identify Virtual SmartCard module.			
/C%EQ%us/O%EQ% MyCompany/OU%EQ %MyOrganization/CN% EQ%MyCompanyName=	This is an example of the entry for the URL for this server. The entry needs to be updated to reflect the company name, organization, and common name for the registration authority.	string	Required	Note 2

Table 1. PKIX .ini file sections and parameters (continued)

Parameter	Use	Format	Req/Opt	Config?
[TrustPolicy]	Used to determine how to verify PKIX certificates.			
UseCRLs=T	Should CRLs and ARLs be used as part of validation process?	T or F	Optional	Yes
AllowExpiredCRLs=F	Are expired CRLs valid?	T or F	Optional	Yes
AllowFutureCRLs=F	Are CRLs with future dates valid?	T or F	Optional	Yes
AllowExpiredCertificates=F	Are expired certificates valid?	T or F	Optional	Yes
AllowFutureCertificates=F	Are future certificates valid?	T or F	Optional	Yes
ApplyNameConstraintsTo EEOnly=F	Apply name constraints to every certificate in chain or only the last?	T or F	Optional	Yes
AllowCRLSearchToFail=F	Is it an error to find no CRLs or ARLs for a given issuer?	T or F	Optional	Yes
MaximumChainSearchDepth=15	Maximum chain depth allowed during recursive breadth-first chain construction.	integer	Optional	Yes
[LDAP]				
NumServers=	Number of LDAP servers.	integer	Required	Yes
Server1=	LDAP domain name and port.	string	Required	Note 2
AuthName1=	LDAP Authority's distinguished name.	string	Required	Note 2
AuthPwd1=	Required password to write entries in LDAP.	string	Required	Note 2
PostInterval=5m	Interval between checks for CRL that needs posting.	interval	Optional	Yes
[RemoteServer]				
MaxSessions=16	Tuning parameter.	integer	Optional	Yes
EncryptionPolicy=F	Is an encryption policy used between a background server and its remote admin if both are on the same system?	T or F	Optional	Yes
NumAdmins=0	Number of administrators certified.	integer	Required if using background servers	Yes
Admin1DN=	Distinguished name of Administrator 1.	string	Required if using background servers	Yes
NumServers=1	Number of servers for this client.	integer	Required if using background servers	Yes
Server1DN=	Server 1's distinguished name.	string	Required if using background servers	Yes
Server1AdminPort=	Server 1's TCP port.	integer	Required if using background servers	Yes

Table 1. PKIX .ini file sections and parameters (continued)

Parameter	Use	Format	Req/Opt	Config?
Server1IPAddress=	Server 1's IP address.	IP address	Required if using background servers	Yes

Notes:

1. The examples shown in the [OIDs] section are not specifically required but there must be an OID entry corresponding to the PolicyName1 parameter. For example, if PolicyName1=MyName, there must be a MyName= parameter in the [OIDs] section with the appropriate value set.
2. This parameter is inserted into the .ini file during initialization automatically. Do not insert this parameter manually.
3. Although the [CertPolicy] and [CrossCertPolicy] parameters have the same default values and therefore use the same implementation for certification and cross-certification, you can customize the implementation so that the two are different.
4. This value must be 1 initially. If additional RAs are enrolled, it may change to another digit which matches an RA*x*= entry.

CA .ini file template

During the PKIX installation, the CA's *.ini file template, caserver.ini, is installed with default values. Use this file as a template for creating the .ini file for the certificate authority.

In addition to changes you make using the IniEditor, some parameters might be changed by the bootstrap wizard or through the RA enrollment wizard. These fields are identified in the "Comments" column.

Table 2. Certificate authority .ini file template

Parameter	Default	Comments
[OIDs]		
C=	2.5.4.6	
O=	2.5.4.10	
OU=	2.5.4.11	
CN=	2.5.4.3	
L=	2.5.4.7	
S=	2.5.4.8	
T=	2.5.4.12	
id-dsa=	1.2.840.10040.4.1	
id-dsa-with-sha1=	1.2.840.10040.4.3	
rsaEncryption=	1.2.840.113549.1.1.1	
sha-1WithRSAEncryption=	1.2.840.113549.1.1.5	
sha1=	1.3.14.3.2.26	
hmac-sha1=	1.3.6.1.5.5.8.1.2	
pkcs7-data=	1.2.840.113549.1.7.1	
pkcs12-certbag=	1.2.840.113549.1.12.10.1.3	

Table 2. Certificate authority .ini file template (continued)

Parameter	Default	Comments
pkcs12-keybag=	1.2.840.113549.1.12.10.1.1	
X509-Certificate=	1.2.840.113549.1.9.22.1	
PasswordBasedMAC=	1.2.840.113533.7.66.13	
MyPolicy=	1.34.67.7	Example entry. Parameter name must match value in [CertPolicy] PolicyName1.
[AsymmetricKeyAlgs]		
DSA=	id-dsa	
RSA=	id-rsaEncryption	
[AsymmetricEncAlgs]		
DSA=	id-dsa	
[AsymmetricSigAlgs]		
DSAwithSHA1=	id-dsa-with-sha1	
RSAwithSHA1=	sha-1WithRSAEncryption	
[ObjectStore]		
Name=	jonahca	
Path=	Windows: c:\pkix\data\ AIX: /usr/pkix/data/	
[CertPolicy]		
SigAlg1=	sha-1WithRSAEncryption	
StartTimeSpecifiable=	T	
MaxLifetime=	8784h	
LifeTimeDef=	180d	
KeySpecifiable=	T	
KeyUsageSupported=	T	
KeyUsageRequired=	F	
PolicyCritical=	F	
PolicyRequired=	F	
PolicyName1=	MyPolicy	
Policy1Org=	MyOrganization	
Policy1Notice1=	3	
Policy1Notice2=	17	
UserNoticeText1=	This is some very ...	
CPS1=	http://www.mycompany.com/cps.html	
TimeBetweenCRLs=	1d	
CRLDuration=	2d	
[CrossCertPolicy]		
SigAlg1=	sha-1WithRSAEncryption	
StartTimeSpecifiable=	T	
MaxLifetime=	8784h	

Table 2. Certificate authority .ini file template (continued)

Parameter	Default	Comments
LifeTimeDef=	180d	
KeySpecifiable=	T	
KeyUsageSupported=	T	
KeyUsageRequired=	F	
PolicyCritical=	F	
PolicyRequired=	F	
PolicyName1=	MyPolicy	
Policy1Org=	MyOrganization	
Policy1Notice1=	3	
Policy1Notice2=	17	
UserNoticeText1=	This is some very ...	
CPS1=	http://www.mycompany.com/cps.html	
TimeBetweenCRLs=	1d	
CRLDuration=	2d	
[General]		
MyName=	*Must not be present for initial configuration.*	Bootstrap wizard sets value.
RA1=	*Must not be present for initial configuration.*	Enrollment wizard sets value.
DefaultRA=	1	
PreferredCryptoProvider=	dda0c1e0-7b73-11d0-8e0c-0004ac602b18	Should not be changed.
CertperDP=	0	Optional parameter
CRLDistName=	MyCRLDistName%d	Optional parameter
TempPath=	Windows: c:\pkix\data\ AIX: /usr/pkix/data/	
PathToDLLs	Windows: c:\pkix\ AIX: /usr/pkix/lib/	
[Transport]		
TCPPort=	1830	Bootstrap wizard may reset value.
TCPHost=	localhost	Bootstrap wizard may reset value.
PollInterval=	30s	
[Keystore]		
CurKeyStore=	VSC	
[VSC]		This section is required if **[Keystore]** CurKeyStore=VSC.
InitialSOpw=	SOPIN	
TokenDir=	Windows: c:\pkix\data\catoken.fil AIX: /usr/pkix/data/catoken.fil	
Model=	PKCS11_STORAGE_MODEL	

Table 2. Certificate authority .ini file template (continued)

Parameter	Default	Comments
Guid=	7f529c80-c942-11d1-8fb0-0004ac61389a	
[URLs]		
/C%EQ%us/O%EQ%MyCompany /OU%EQ%MyOrganization/CN% EQ%MyRegistrationAuthority=	*Must not be present for initial configuration.*	Bootstrap wizard sets value.
[TrustPolicy]		
UseCRLs=	T	
AllowExpiredCRLs=	F	
AllowFutureCRLs=	F	
AllowExpiredCertificates=	F	
AllowFutureCertificates=	F	
ApplyNameConstraintsToEEOnly=	F	
AllowCRLSearchToFail=	F	
MaximumChainSearchDepth=	15	
[RemoteServer]		
MaxSessions=	16	
EncryptionPolicy=	F	
NumAdmins=	0	Updated by addCAAdm.exe
Admin1DN=	*Not initialized*	Added by addCAAdm.exe

CA administrator GUI .ini file

During the PKIX installation, the CA's administrator GUI *.ini file template, caadmin.ini, is installed with default values. Use this file as a template for creating the .ini file for the CA administrator GUI.

Table 3. Certificate authority administrator GUI .ini file template

Parameter	Default	Comments
[OIDs]		
C=	2.5.4.6	
O=	2.5.4.10	
OU=	2.5.4.11	
CN=	2.5.4.3	
L=	2.5.4.7	
S=	2.5.4.8	
T=	2.5.4.12	
id-dsa=	1.2.840.10040.4.1	
id-dsa-with-sha1=	1.2.840.10040.4.3	
rsaEncryption=	1.2.840.113549.1.1.1	
sha-1WithRSAEncryption=	1.2.840.113549.1.1.5	
sha1=	1.3.14.3.2.26	
hmac-sha1=	1.3.6.1.5.5.8.1.2	

Table 3. Certificate authority administrator GUI .ini file template (continued)

Parameter	Default	Comments
pkcs7-data=	1.2.840.113549.1.7.1	
pkcs12-certbag=	1.2.840.113549.1.12.10.1.3	
pkcs12-keybag=	1.2.840.113549.1.12.10.1.1	
X509-Certificate=	1.2.840.113549.1.9.22.1	
PasswordBasedMAC=	1.2.840.113533.7.66.13	
MyPolicy=	1.34.67.7	Example entry. Parameter name must match value in [CertPolicy] PolicyName1.
[General]		
PathToDLLs=	Windows: c:\pkix\ AIX: /usr/pkix/lib/	
[Keystore]		
CurKeyStore=	VSC	
[VSC]		
InitialSOpw=	SOPIN	
TokenDir=	Windows: c:\pkix\data\catoken.fil AIX: /usr/pkix/data/catoken.fil	
Model=	PKCS11_STORAGE_MODEL	
Guid=	7f529c80-c942-11d1-8fb0-0004ac61389a	
[RemoteServer]		
NumServers=	1	
Server1DN=	*Not initialized*	
Server1AdminPort=	1831	
Server1IPAddress=	127.0.0.1	
EncryptionPolicy=	F	

RA .ini file template

During the PKIX installation, the RA's *.ini file template, raserver.ini, is installed with default values. Use this file as a template for creating the .ini file for the registration authority.

In addition to changes made through the IniEditor, some parameters might be changed by the bootstrap wizard or through the RA enrollment wizard. These fields are identified in the "Comments" column.

Table 4. Registration authority .ini file template

Parameter	Default	Comments
[OIDs]		
C=	2.5.4.6	
O=	2.5.4.10	
OU=	2.5.4.11	

Table 4. Registration authority .ini file template (continued)

Parameter	Default	Comments
CN=	2.5.4.3	
L=	2.5.4.7	
S=	2.5.4.8	
T=	2.5.4.12	
id-dsa=	1.2.840.10040.4.1	
id-dsa-with-sha1=	1.2.840.10040.4.3	
rsaEncryption=	1.2.840.113549.1.1.1	
sha-1WithRSAEncryption=	1.2.840.113549.1.1.5	
sha1=	1.3.14.3.2.26	
hmac-sha1=	1.3.6.1.5.5.8.1.2	
pkcs7-data=	1.2.840.113549.1.7.1	
pkcs12-certbag=	1.2.840.113549.1.12.10.1.3	
pkcs12-keybag=	1.2.840.113549.1.12.10.1.1	
X509-Certificate=	1.2.840.113549.1.9.22.1	
PasswordBasedMAC=	1.2.840.113533.7.66.13	
MyPolicy=	1.34.67.7	Example entry. Parameter name must match value in [CertPolicy]PolicyName.
[AsymmetricKeyAlgs]		
DSA=	id-dsa	
RSA=	rsaEncryption	
[AsymmetricEncAlgs]		
DSA=	id-dsa	
[AsymmetricSigAlgs]		
DSAwithSHA1=	id-dsa-with-sha1	
RSAwithSHA1=	sha-1WithRSAEncryption	
[ObjectStore]		
Name=	jonahra	
Path=	Windows: c:\pkix\ data\ AIX: /usr/pkix/data/	
[CertPolicy] also known as **[Issuer*x*CertPolicy]**		
SigAlg1=	sha-1WithRSAEncryption	
StartTimeSpecifiable=	T	
MaxLifetime=	8784h	Only used during bootstrap
LifeTimeDef=	180d	Only used during bootstrap
KeySpecifiable=	T	
KeyUsageSupported=	T	
KeyUsageRequired=	F	
PolicyCritical=	F	

Table 4. Registration authority .ini file template (continued)

Parameter	Default	Comments
PolicyRequired=	F	
PolicyName1=	MyPolicy	
Policy1Org=MyOrganization	MyOrganization	
Policy1Notice1=	3	
Policy1Notice2=	17	
UserNoticeText1=	This is some very ...	
CPS1=	http://www.mycompany.com/cps.html	
EERevokeRequests	Any	
[CrossCertPolicy] also known as **[Issuer*x*CrossCertPolicy]**		
SigAlg1=	sha-1WithRSAEncryption	
StartTimeSpecifiable=	T	
MaxLifetime=	8784h	Only used during bootstrap
LifeTimeDef=	180d	Only used during bootstrap
KeySpecifiable=	T	
KeyUsageSupported=	T	
KeyUsageRequired=	F	
PolicyCritical=	F	
PolicyRequired=	F	
PolicyName1=	MyPolicy	
Policy1Org=MyOrganization	MyOrganization	
Policy1Notice1=	3	
Policy1Notice2=	17	
UserNoticeText1=	This is some very ...	
CPS1=	http://www.mycompany.com/cps.html	
EERevokeRequests	Any	
[General]		
MyName=	*Must not be present for initial configuration.*	Set by bootstrap wizard.
Issuer1=	*Must not be present for initial configuration.*	Set by bootstrap wizard.
Issuer1URL1=	*Must not be present for initial configuration.*	Set by bootstrap wizard.
TempPath=	Windows: c:\pkix\data\ AIX: /usr/pkix/data/	
PathToDLLs	Windows: c:\pkix\ AIX: /usr/pkix/lib/	
[Transport]		
TCPPort=	1830	Bootstrap wizard might reset value.
TCPHost=	localhost	Bootstrap wizard might reset value.

Table 4. Registration authority .ini file template (continued)

Parameter	Default	Comments
PollInterval=	30s	
[Keystore]		
CurKeyStore=	VSC	
[VSC]		This section is required if **[Keystore]** CurKeyStore= VSC.
InitialSOpw=	SOPIN	
TokenDir=	Windows: c:\pkix\data\ratoken.fil AIX: /usr/pkix/data/ratoken.fil	
Model=	PKCS11_STORAGE_MODEL	
Guid=	7f529c80-c942-11d1-8fb0-0004ac61389a	
[TrustPolicy]		
UseCRLs=	T	
AllowExpiredCRLs=	F	
AllowFutureCRLs=	F	
AllowExpiredCertificates=	F	
AllowFutureCertificates=	F	
ApplyNameConstraintsToEEOnly=	F	
AllowCRLSearchToFail=	F	
MaximumChainSearchDepth=	15	
[LDAP]		
NumServers=	1	
Server1=	*Must not be present for initial configuration.*	Set by bootstrap wizard.
AuthName1=	*Must not be present for initial configuration.*	Set by bootstrap wizard.
AuthPwd1=	*Must not be present for initial configuration.*	Set by bootstrap wizard.
PostInterval=	5m	
[RemoteServer]		
MaxSessions=	16	
EncryptionPolicy=	F	
NumAdmins=	0	Updated by addRAAdm.exe
Admin1DN=	*Not initialized*	Added by addRAAdm.exe

RA administrator GUI .ini file

During the PKIX installation, the RA's administrator GUI *.ini file template, raadmin.ini, is installed with default values. Use this file as a template for creating the .ini file for the RA administrator GUI.

Table 5. Registration authority administrator GUI .ini file template

Parameter	Default	Comments
[OIDs]		
C=	2.5.4.6	
O=	2.5.4.10	
OU=	2.5.4.11	
CN=	2.5.4.3	
L=	2.5.4.7	
S=	2.5.4.8	
T=	2.5.4.12	
id-dsa=	1.2.840.10040.4.1	
id-dsa-with-sha1=	1.2.840.10040.4.3	
rsaEncryption=	1.2.840.113549.1.1.1	
sha-1WithRSAEncryption=	1.2.840.113549.1.1.5	
sha1=	1.3.14.3.2.26	
hmac-sha1=	1.3.6.1.5.5.8.1.2	
pkcs7-data=	1.2.840.113549.1.7.1	
pkcs12-certbag=	1.2.840.113549.1.12.10.1.3	
pkcs12-keybag=	1.2.840.113549.1.12.10.1.1	
X509-Certificate=	1.2.840.113549.1.9.22.1	
PasswordBasedMAC=	1.2.840.113533.7.66.13	
MyPolicy=	1.34.67.7	Example entry. Parameter name must match value in [CertPolicy]PolicyName.
[General]		
PathToDLLs=	Windows: c:\pkix\ AIX: /usr/pkix/lib/	
[Keystore]		
CurKeyStore=	VSC	
[VSC]		This section is required if **[Keystore]** CurKeyStore= VSC.
InitialSOpw=	SOPIN	
TokenDir=	Windows: c:\pkix\data\ratoken.fil AIX: /usr/pkix/data/ratoken.fil	
Model=	PKCS11_STORAGE_MODEL	
Guid=	7f529c80-c942-11d1-8fb0-0004ac61389a	
[RemoteServer]		
NumServers=	1	
Server1DN=	*Not initialized*	
Server1AdminPort=	1832	
Server1IPAddress=	127.0.0.1	

Table 5. Registration authority administrator GUI .ini file template (continued)

Parameter	Default	Comments
EncryptionPolicy=	F	

EE .ini file template

During the PKIX installation, the EE's *.ini file template, eetemplate.ini, is installed with default values. Use this file as a template for creating the .ini file for the end entity.

Table 6. End entity .ini file template

Parameter	Default	Comments
[OIDs]		
C=	2.5.4.6	
O=	2.5.4.10	
OU=	2.5.4.11	
CN=	2.5.4.3	
L=	2.5.4.7	
S=	2.5.4.8	
T=	2.5.4.12	
id-dsa=	1.2.840.10040.4.1	
id-dsa-with-sha1=	1.2.840.10040.4.3	
rsaEncryption=	1.2.840.113549.1.1.1	
sha-1WithRSAEncryption=	1.2.840.113549.1.1.5	
sha1=	1.3.14.3.2.26	
hmac-sha1=	1.3.6.1.5.5.8.1.2	
pkcs7-data=	1.2.840.113549.1.7.1	
pkcs12-certbag=	1.2.840.113549.1.12.10.1.3	
pkcs12-keybag=	1.2.840.113549.1.12.10.1.1	
X509-Certificate=	1.2.840.113549.1.9.22.1	
PasswordBasedMAC=	1.2.840.113533.7.66.13	
MyPolicy=	1.34.67.7	Example entry. Parameter name must match value in [CertPolicy]PolicyName.
[ObjectStore]		
Name=	jonahee	
Path=	Windows: c:\pkix\data AIX: /usr/pkix/data/	
TempPath=	Windows: c:\pkix\data\ AIX: /usr/pkix/data/	
[General]		
TempPath=	Windows: c:\pkix\data\ AIX: /usr/pkix/data/	

Table 6. End entity .ini file template (continued)

Parameter	Default	Comments
PathToDLLs	Windows: c:\pkix\ AIX: /usr/pkix/lib/	
[Transport]		
RetryInterval=	1m	
MaxErrorCount=	20	
[Keystore]		
CurKeyStore=	VSC	
[VSC]		This section is required if **[Keystore]** CurKeyStore= VSC.
InitialSOpw=	SOPIN	
TokenDir=	Windows: c:\pkix\data\eetoken.fil AIX: /usr/pkix/data/eetoken.fil	
Model=	PKCS11_STORAGE_MODEL	
Guid=	7f529c80-c942-11d1-8fb0-0004ac61389a	
[SmartCard]		
Model=	PKCS11_STORAGE_MODEL	
Guid=	EA8A7C01-12BC-11D3-B150-002035C00173	
LibraryName=		
[TrustPolicy]		
UseCRLs=	T	
AllowExpiredCRLs=	F	
AllowFutureCRLs=	F	
AllowExpiredCertificates=	F	
AllowFutureCertificates=	F	
ApplyNameConstraintsToEEOnly=	F	
AllowCRLSearchToFail=	F	
MaximumChainSearchDepth=	15	

Appendix B. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs

and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department LZKS
11400 Burnet Road
Austin, TX 78758
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX
IBM
SecureWay

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

This program contains Standard Template Library (STL) software from Hewlett-Packard Company. Copyright (c) 1994.

- Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Hewlett-Packard Company makes no representations about the suitability of this software for any purpose. It is provided “as is” without express or implied warranty.

This program contains Standard Template Library (STL) software from Silicon Graphics Computer Systems, Inc. Copyright (c) 1996–1999.

- Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Silicon Graphics makes no representations about the suitability of this software for any purpose. It is provided “as is” without express or implied warranty.

Other company, product, and service names may be trademarks or service marks of others.

Glossary

This glossary defines the terms and abbreviations in this book that may be new or unfamiliar and terms that may be of interest. It includes terms and definitions from:

- The IBM Dictionary of Computing, New York: McGraw-Hill, 1994.
- The American National Standard Dictionary for Information Systems, ANSI X3.172–1990, American National Standards Institute (ANSI), 1990.
- The Answers to Frequently Asked Questions, Version 3.0, California: RSA Data Security, Inc., 1998.

Numbers

4758 PCI Cryptographic Coprocessor. A programmable, tamper-resistant cryptographic PCI-bus card offering high performance DES and RSA cryptographic processing. The cryptographic processes occur within a secure enclosure on the card. The card meets the stringent requirements of the FIPS PUB 140-1 level 4 standard. Software can run within the secure enclosure. For example, credit card transaction processing can use the SET standard.

A

Abstract Syntax Notation One (ASN.1). An ITU notation that is used to define the syntax of information data. It defines a number of simple data types and specifies a notation for identifying these types and for specifying values of these types. These notations can be applied whenever it is necessary to define the abstract syntax of information without curbing how the information is encoded for transmission.

access control list (ACL). A mechanism for limiting the use of a specific resource to authorized users.

ACL. Access control list.

action history. Accumulated events in the life cycle of a credential.

American National Standard Code for Information Interchange (ASCII). The standard code that is used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set uses a coded character set that consists of 7-bit coded characters (8 bits including a bit for parity checking). The character set consists of control characters and graphic characters.

American National Standards Institute (ANSI). An organization that establishes the procedures by which accredited organizations create and maintain voluntary industry standards in the United States. It consists of producers, consumers, and general interest groups.

ANSI. American National Standards Institute.

API. Application program interface.

applet. A computer program that is written in Java and runs inside a Java-compatible Web browser. Also known as a Java applet.

application program interface (API). In PKIX, a functional interface that allows an application program that is written in a high-level language to use specific PKIX functions.

ASCII. American National Standard Code for Information Interchange.

ASN.1. Abstract Syntax Notation One.

asymmetric cryptography. Cryptography that uses different, asymmetric keys for encryption and decryption. Each user receives a pair of keys: a public key accessible to all, and a private key known only to the user. A secure transaction can occur when the public key and the corresponding private key match, enabling the decryption of the transaction. This is also known as key pair cryptography. *Contrast with* symmetric cryptography.

asynchronous communication. A mode of communication that does not require the sender and recipient to be present simultaneously.

audit trail. Data, in the form of a logical path, that links a sequence of events. An audit trail enables tracing of transactions or the history of a given activity.

authentication. The process of reliably determining the identity of a communicating party.

authorization. Permission to access a resource.

B

base64 encoding. A common means of conveying binary data with MIME.

Basic Encoding Rules (BER). The rules specified in ISO 8825 for encoding data units described in abstract syntax notation 1 (ASN.1). The rules specify the encoding technique, not the abstract syntax.

BER. Basic Encoding Rules.

business process objects. A set of code used to accomplish a specific registration operation, such as checking the status of an enrollment request or verifying that a public key was sent.

business process template. A set of business process objects that are run in a specified order.

browser. *See* Web browser.

browser certificate. A digital certificate is also known as a client-side certificate. It is issued by a CA through an SSL-enabled Web server. Keys in an encrypted file enable the holder of the certificate to encrypt, decrypt, and sign data. Typically, the Web browser stores these keys. Some applications permit storage of the keys on smart cards or other media. *See also* digital certificate.

bytecode. Machine-independent code that is generated by the Java compiler and run by the Java interpreter.

C

CA. Certificate authority.

CA certificate. A certificate your Web browser accepts, at your request, from a CA it does not recognize. The browser can then use this certificate to authenticate communications with servers that hold certificates issued by that CA.

CAST-64. A block cipher algorithm that uses a 64-bit block size and a 6-bit key. It was designed by Carlisle Adams and Stafford Tavares.

CCA. IBM Common Cryptographic Architecture.

CDSA. Common Data Security Architecture.

Common Data Security Architecture (CDSA). An initiative to define a comprehensive approach to security service and security management for computer-based security applications. It was designed by Intel, to make computer platforms more secure for applications.

certificate authority (CA). The software responsible for following an organization's security policies and assigning secure electronic identities in the form of certificates. The CA processes requests from RAs to issue, renew, and revoke certificates. The CA interacts with the RA to publish certificates and CRLs in the Directory. *See also* digital certificate.

certificate extension. An optional feature of the X.509v3 certificate format that provides for the inclusion of additional fields in the certificate. There are standard extensions and user-defined extensions. Standard extensions exist for various purposes, including key and policy information, subject and issuer attributes, and certification path constraints.

certificate policy. A named set of rules that indicates the applicability of a certificate to a particular class of applications that have common security requirements. For example, a certificate policy might indicate whether a particular certification type allows a user to conduct transactions for goods within a given price range.

certificate profile. A set of characteristics that define the type of certificate wanted (such as SSL certificates or IPSec certificates). The profile aids in managing certificate specification and registration. The issuer can change the names of the profiles and specify characteristics of the desired certificate, such as the validity period, key usage, DN constraints, and so forth.

certificate revocation list (CRL). A digitally signed, time-stamped list of certificates that the certificate authority has revoked. The certificates in this list should be considered unacceptable. *See also* digital certificate.

certification. The process during which a trusted third party issues an electronic credential that vouches for an individual, business, or organizational identity.

CGI. Common Gateway Interface.

chain validation. The validation of all CA signatures in the trust hierarchy through which a given certificate was issued. For example, if a CA was issued its signing certificate by another CA, both signatures are validated during validation of the certificate that the user presents.

class. In object-oriented design or programming, a group of objects that share a common definition and therefore share common properties, operations, and behavior.

cleartext. Data that is not encrypted. *Synonym for* plaintext.

client. (1) A functional unit that receives shared services from a server. (2) A computer or program that requests a service of another computer or program.

client/server. A model in distributed processing in which a program at one site sends a request to a program at another site and waits for a response. The requesting program is called a client; the answering one is called a server.

code signing. A technique for signing executable programs with digital signatures. Code signing is designed to improve the reliability of software that is distributed over the Internet.

Common Cryptographic Architecture (CCA). IBM software that enables a consistent approach to cryptography on major IBM computing platforms. It supports application software that is written in a variety of programming languages. Application software can call on CCA services to perform a broad range of cryptographic functions, including DES and RSA encryption.

Common Gateway Interface (CGI). Standard method of transmitting information between Web pages and Web servers.

confidentiality. The property of not being divulged to unauthorized parties.

credential. Confidential information used to prove one's identity in an authentication exchange. In environments for network computing, the most common type of credential is a certificate that a CA has created and signed.

CRL. Certificate revocation list.

CRL publication interval. Set in the CA configuration file, the interval of time between periodic publications of the CRL to the Directory.

cross-certification. A trust model whereby one CA issues to another CA a certificate that contains the public key associated with its private signature key. A cross-certified certificate allows client systems or end entities in one administrative domain to communicate securely with client systems or end entities in another domain.

cryptographic. Pertaining to the transformation of data to conceal its meaning.

cryptography. In computer security, the principles, means, and methods for encrypting plaintext and decrypting encrypted text.

D

daemon. A program that carries out tasks in the background. It is implicitly called when a condition occurs that requires its help. A user need not be aware of a daemon, because the system usually spawns it automatically. A daemon might live forever or the system might regenerate it at intervals.

The term (pronounced *demon*) comes from mythology. Later, it was rationalized as the acronym DAEMON: Disk And Execution MONitor.

Data Encryption Standard (DES). An encryption block cipher, defined and endorsed by the U.S. government in 1977 as an official standard. IBM developed it originally. DES has been extensively studied since its publication and is a well-known and widely used cryptographic system.

DES is a symmetric cryptographic system. When it is used for communication, both the sender and receiver must know the same secret key. This key is used to encrypt and decrypt the message. DES can also be used for single-user encryption, such as to store files on a hard disk in encrypted form. DES has a 64-bit block size and uses a 56-bit key during encryption. It is was originally designed for implementation in hardware. NIST has recertified DES as an official U.S. government encryption standard every five years.

Data Storage Library (DL). A module that provides access to persistent data stores of certificates, CRLs, keys, policies, and other security-related objects.

decrypt. To undo the encryption process.

DEK. Document encrypting key.

DER. Distinguished Encoding Rules.

DES. Data Encryption Standard.

Diffie-Hellman. A method of establishing a shared key over an insecure medium, named after the inventors (Diffie and Hellman).

digital certificate. An electronic credential that is issued by a trusted third party to a person or entity. Each certificate is signed with the private key of the CA. It vouches for an individual, business, or organizational identity.

Depending on the role of the CA, the certificate can attest to the authority of the bearer to conduct e-business over the Internet. In a sense, a digital certificate performs a similar role to a driver's license or a medical diploma. It certifies that the bearer of the corresponding private key has authority to conduct certain e-business activities.

A certificate contains information about the entity it certifies, whether person, machine, or computer program. It includes the certified public key of that entity.

digital certification. *See* certification.

digital signature. A coded message added to a document or data that guarantees the identity of the sender.

A digital signature can provide a greater level of security than a physical signature. The reason for this is that a digital signature is not an encrypted name or series of simple identification codes. Instead, it is an encrypted summary of the message that is being signed. Thus, affixing a digital signature to a message provides solid identification of the sender. (Only the sender's key can create the signature.) It also fixes the content of the message that is being signed (the encrypted message summary must match the message content or the signature is not valid). Thus, a digital signature cannot be copied from one message and applied to another because the summary, or hash, would not match. Any alterations to the signed message would also invalidate the signature.

Digital Signature Algorithm (DSA). A public key algorithm that is used as part of the Digital Signature Standard. It cannot be used for encryption, only for digital signatures.

Directory. A hierarchical structure intended as a global repository for information related to communications (such as e-mail or cryptographic exchanges). The Directory stores specific items that are essential to the PKI structure, including public keys, certificates, and certificate revocation lists.

Data in the Directory is organized hierarchically in the form of a tree, with the root at the top of the tree. Often, higher level organizations represent individual countries, governments, or companies. Users and devices are typically represented as leaves of each tree. These users, organizations, localities, countries, and devices each have their own entry. Each entry consists of typed attributes. These provide information about the object that the entry represents.

Each entry in the Directory is bound with an associated distinguished name (DN). This is unique when the entry includes an attribute that is known to be unique to the real world object. Consider the following example DN. In it, the country (C) is US, the organization (O) is IBM, the organizational unit (OU) is Trust, and the common name (CN) is CA1.

```
C=US/O=vnet/OU=Trust/CN=CA1
```

Distinguished Encoding Rules (DER). Provides constraints on the BER. DER selects just one type of encoding from those that the encoding rules allow, eliminating all of the sender's options.

distinguished name (DN). The unique name of a data entry that is stored in the Directory. The DN uniquely identifies the position of an entry in the hierarchical structure of the Directory.

DL. Data Storage Library.

DN. Distinguished name.

document encrypting key (DEK). Typically, a symmetric encryption/decryption key, such as DES.

domain. *See* security domain *and* registration domain.

DSA. Digital Signature Algorithm.

E

e-business. Business transactions over networks and through computers. It includes buying and selling goods and services. It also includes transferring funds through digital communications.

e-commerce. Business-to-business transactions. It includes buying and selling goods and services (with customers, suppliers, vendors, and others) on the Internet. It is a primary element of e-business.

end-entity. The subject of a certificate that is not a CA.

encrypt. To scramble information so that only someone who has the appropriate decryption code can obtain the original information through decryption.

encryption/decryption. Using the public key of the intended recipient to encipher data for that person, who then uses the private key of the pair to decipher the data.

enrollment attribute. An enrollment variable that is contained in an enrollment form. Its value reflects the information that is captured during the enrollment. The value of the enrollment attribute remains the same throughout the lifetime of the credential.

enrollment variable. *See* enrollment attribute.

extranet. A derivative of the Internet that uses similar technology. Companies are beginning to apply Web publishing, electronic commerce, message transmission, and groupware to multiple communities of customers, partners, and internal staff.

F

File Transfer Protocol (FTP). An Internet client/server protocol for use in transferring files between computers.

firewall. A gateway between networks that restricts the flow of information between networks. Typically, the purpose of a firewall is to protect internal networks from unauthorized use from the outside.

FTP. File Transfer Protocol.

G

gateway. A functional unit that allows incompatible networks or applications to communicate with each other.

H

HTML. Hypertext Markup Language.

HTTP. Hypertext Transaction Protocol.

HTTP server. A server that handles Web-based communications with browsers and other programs in a network.

hypertext. Text that contains words, phrases, or graphics that the reader can click with the mouse to retrieve and display another document. These words, phrases, or graphics are known as hyperlinks. Retrieving them is known as linking to them.

Hypertext Markup Language (HTML). A markup language for coding Web pages. It is based on SGML.

Hypertext Transaction Protocol (HTTP). An Internet client/server protocol for transferring hypertext files across the Web.

I

ICL. Issued certificate list.

IETF (Internet Engineering Task Force). A group that focuses on engineering and developing protocols for the Internet. It represents an international community of network designers, operators, vendors, and researchers. The IETF is concerned with the development of the Internet architecture and the smooth use of the Internet.

integrity. A system protects the integrity of data if it prevents unauthorized modification (as opposed to protecting the confidentiality of data, which prevents unauthorized disclosure).

integrity checking. The checking of audit records that result from transactions with external components.

internal structure. *See* schema.

International Standards Organization (ISO). An international organization tasked with developing and publishing standards for everything from wine glasses to computer network protocols.

International Telecommunication Union (ITU). An international organization within which governments and the private sector coordinate global telecommunication networks and services. It is the leading publisher of telecommunication technology, regulatory, and standards information.

Internet. A worldwide collection of networks that provide electronic connection between computers. This enables them to communicate with each other via software devices such as electronic mail or Web browsers. For example, some universities are on a network that in turn links with other similar networks to form the Internet.

intranet. A network within an enterprise that usually resides behind firewalls. It is a derivative of the Internet and uses similar technology. Technically, intranet is a mere extension of the Internet. HTML and HTTP are some of the commonalties.

IPSec. An Internet Protocol Security standard, developed by the IETF. IPSec is a network layer protocol, designed to provide cryptographic security services that flexibly support combinations of authentication, integrity, access control, and confidentiality. Because of its strong authentication features, it has been adopted by many VPN product vendors as the protocol for establishing secure point-to-point connections over the Internet.

ISO. International Standards Organization.

issued certificate list (ICL). A complete list of the certificates that have been issued and their current status. Certificates are indexed by serial number and state. This list is maintained by the CA and stored in the CA database.

ITU. International Telecommunication Union.

J

Java. A set of network-aware, non-platform-specific computer technologies developed by Sun Microsystems, Incorporated. The Java environment consists of the Java OS, the virtual machines for various platforms, the object-oriented Java programming language, and several class libraries.

Java applet. *See* applet. *Contrast with* Java application.

Java application. A stand-alone program that is written in the Java language. It runs outside the context of a Web browser.

Java class. A unit of Java program code.

Java language. A programming language, developed by Sun Microsystems, designed specifically for use in applet and agent applications.

Java Virtual Machine (JVM). The part of the Java run-time environment responsible for interpreting bytecodes.

K

key. A quantity used in cryptography to encipher or decipher information.

key pair. Corresponding keys that are used in asymmetric cryptography. One key is used to encrypt and the other to decrypt.

L

LDAP. Lightweight Directory Access Protocol.

Lightweight Directory Access Protocol (LDAP). A protocol used to access the Directory.

M

MAC. Message authentication code.

MIME (Multipurpose Internet Mail Extensions). A freely available set of specifications that allows the interchange of text in languages with different character sets. it also allows multimedia e-mail among many different computer systems that use Internet mail

standards. For example, the e-mail messages may contain character sets other than US-ASCII, enriched text, images, and sounds.

modulus. In the RSA public key cryptographic system, the product (n) of two large primes: p and q. The best size for an RSA modulus depends on one's security needs. The larger the modulus, the greater the security. The current RSA Laboratories–recommended key sizes depend on the planned use for the key: 768 bits for personal use, 1024 bits for corporate use, and 2048 bits for extremely valuable keys like the key pair of a CA. A 768-bit key is expected to be secure until at least the year 2004.

N

National Language Support (NLS). Support within a product for differences in locales, including language, currency, date and time format, and numeric presentation.

National Security Agency (NSA). The official security body of the U.S. government.

NIST. National Institute of Standards and Technology, formerly known as NBS (National Bureau of Standards). It promotes open standards and interoperability in computer-based industries.

NLS. National language support.

nonce. A string that is sent down from a server or application, requesting user authorization. The user that is asked for authentication signs the nonce with a private key. The user's public key and the signed nonce are sent back to the server or application that requested authentication. The server then attempts to decipher the signed nonce with the user's public key. If the deciphered nonce is the same as the original nonce that was sent, the user is authenticated.

non-repudiation. The use of a digital private key to prevent the signer of a document from falsely denying having signed it.

NSA. National Security Agency.

O

object. In object-oriented design or programming, an abstraction encapsulating data and the operations associated with that data. *See also* class.

object identifier (OID). An administratively assigned data value of the type defined in abstract syntax notation 1 (ASN.1).

object type. The kind of object that can be stored in the Directory. For example, an organization, meeting room, device, person, program, or process.

ODBC. Open Database Connectivity.

Open Database Connectivity (ODBC). A standard for accessing different database systems.

Open Systems Interconnect (OSI). The name of the computer networking standards that the ISO approved.

OSI. Open Systems Interconnect.

P

PC card. Similar to a smart card, and sometimes called a PCMCIA card. This card is somewhat larger than a smart card and usually has a greater capacity.

PEM. Privacy-enhanced mail.

PKCS. Public Key Cryptography Standards.

PKCS #1. *See* Public Key Cryptography Standards.

PKCS #7. *See* Public Key Cryptography Standards.

PKCS #10. *See* Public Key Cryptography Standards.

PKCS #11. *See* Public Key Cryptography Standards.

PKCS #12. *See* Public Key Cryptography Standards.

PKI. Public key infrastructure.

PKIX. An X.509v3-based PKI.

PKIX certificate management protocol (CMP). A protocol that enables connections with PKIX-compliant applications. PKIX CMP uses TCP/IP as its primary transport mechanism, but an abstraction layer over sockets exists. This enables support for additional polling transports.

PKIX CMP. PKIX certificate management protocol.

plaintext. Unencrypted data. *Synonym for* cleartext.

policy exit. In a registration application, an organization-defined program that is called by the application. The rules specified in a policy exit apply the organization's business and security preferences to the enrollment process.

privacy. Protection from the unauthorized disclosure of data.

privacy-enhanced mail (PEM). The Internet privacy-enhanced mail standard, that the Internet Architect Board (IAB) adopted to provide secure electronic mail over the Internet. The PEM protocols provide for encryption, authentication, message integrity, and key management.

private key. The key in a public/private key pair that is available only to its owner. It enables the owner to receive a private transaction or make a digital signature.

Data signed with a private key can be verified only with the corresponding public key. *Contrast with* public key. *See also* public/private key pair.

protocol. An agreed-on convention for inter-computer communication.

proxy server. An intermediary between the computer that is requesting access (computer A) and the computer that is being accessed (computer B). Thus, if an end user makes a request for a resource from computer A, this request is directed to a proxy server. The proxy server makes the request, gets the response from computer B, and then forwards the response to the end user. Proxy servers are useful for accessing World Wide Web resources from inside a firewall.

public key. The key in a public/private key pair that is made available to others. It enables them to direct a transaction to the owner of the key or verify a digital signature. Data encrypted with the public key can be decrypted only with the corresponding private key. *Contrast with* private key. *See also* public/private key pair.

Public Key Cryptography Standards (PKCS). Informal inter-vendor standards developed in 1991 by RSA Laboratories with representatives from various computer vendors. These standards cover RSA encryption, the Diffie-Hellman agreement, password-based encryption, extended-certificate syntax, cryptographic message syntax, private-key information syntax, and certification syntax.

- PKCS #1 describes a method for encrypting data by using the RSA public key cryptosystem. Its intended use is in the construction of digital signatures and digital envelopes.
- PKCS #7 specifies a general format for cryptographic messages.
- PKCS #10 specifies a standard syntax for certification requests.
- PKCS #11 defines a technology-independent programming interface for cryptographic devices such as smart cards.
- PKCS #12 specifies a portable format for storing or transporting a user's private keys, certificates, miscellaneous secrets, and so forth.

public key infrastructure (PKI). A standard for security software that is based on public key cryptography. The PKI is a system of digital certificates, certificate authorities, registration authorities, certificate management services, and distributed directory services. It is used to verify the identity and authority of each party involved in any transaction over the Internet. These transactions might involve operations where identity verification is required. For example, they might confirm the origin of proposal bids, authors of e-mail messages, or financial transactions.

The PKI achieves this by making the public encryption keys and certificates of users available for authentication by a valid individual or organization. It provides online directories that contain the public encryption keys and certificates that are used in verifying digital certificates, credentials, and digital signatures.

The PKI provides a means for swift and efficient responses to verification queries and requests for public encryption keys. It also identifies potential security threats to the system and maintains resources to deal with security breaches. Lastly, the PKI provides a digital timestamping service for important business transactions.

public/private key pair. A public/private key pair is part of the concept of key pair cryptography (introduced in 1976 by Diffie and Hellman to solve the key management problem). In their concept, each person obtains a pair of keys, one called the public key and the other called the private key. Each person's public key is made public while the private key is kept secret. The sender and receiver do not need to share secret information: all communications involve only public keys, and no private key is ever transmitted or shared. It is no longer necessary to trust some communications channel to be secure against eavesdropping or betrayal. The only requirement is that public keys must be associated with their users in a trusted (authenticated) manner (for instance, in a trusted directory). Anyone can send a confidential message by using public information. However, the message can be decrypted only with a private key, which is in the sole possession of the intended recipient. Furthermore, key pair cryptography can be used not only for privacy (encryption), but also for authentication (digital signatures).

R

RA. Registration authority.

RA administrator. A user who has been authorized to access the RA Desktop, to administer certificates and requests for certificates.

RC2. A variable key-size block cipher, designed by Ron Rivest for RSA Data Security. *RC* stands for *Ron's Code* or *Rivest's Cipher*. It is faster than DES and is designed as a drop-in replacement for DES. It can be made more secure or less secure against exhaustive key search than DES by using appropriate key sizes. It has a block size of 64 bits and is about two to three times faster than DES in software. RC2 can be used in the same modes as DES.

An agreement between the Software Publishers Association (SPA) and the United States government gives RC2 special status. This makes the export approval process simpler and quicker than the usual cryptographic export process. However, to qualify for quick export approval a product must limit the RC2 key

size to 40 bits with some exceptions. An additional string can be used to thwart attackers who try to precompute a large look-up table of possible encryptions.

registration authority (RA). The software that administers digital certificates to ensure that an organization's business policies are applied from the initial receipt of an enrollment request through certificate revocation.

registration database. Contains information about certificate requests and issued certificates. The database stores enrollment data and all changes to the certificate data throughout its life cycle. The database can be updated by RA processes and policy exits, or by RA administrators.

registration domain. A set of resources, policies, and configuration options related to specific certificate registration processes. The domain name is a subset of the URL that is used to run the registration application.

repudiate. To reject as untrue; for example, to deny that you sent a specific message or submitted a specific request.

request ID. A 24- to 32-character ASCII value that uniquely identifies a certificate request to the RA. This value can be used on the certificate request transaction to retrieve the status of the request or the certificate that is associated with it.

RSA. A public key cryptographic algorithm that is named for its inventors (Rivest, Shamir, and Adelman). It is used for encryption and digital signatures.

S

schema. As relates to the Directory, the internal structure that defines the relationships between different object types.

Secure Electronic Transaction (SET). An industry standard that facilitates secure credit card or debit card payment over untrusted networks. The standard incorporates authentication of cardholders, merchants, and card-issuing banks because it calls for the issuance of certificates.

Secure Sockets Layer (SSL). An IETF standard communications protocol with built-in security services that are as transparent as possible to the end user. It provides a digitally secure communications channel.

An SSL-capable server usually accepts SSL connection requests on a different port than requests for standard HTTP requests. SSL creates a session during which the exchange signals to set up communications between two modems need to occur only once. After that, communication is encrypted. Message integrity checking continues until the SSL session expires.

security domain. A group (a company, work group or team, educational or governmental) whose certificates have been certified by the same CA. Users with certificates that are signed by a CA can trust the identity of another user that has a certificate signed by the same CA.

server. (1) In a network, a data station that provides functions to other stations; for example, a file server. (2) In TCP/IP, a system in a network that handles the requests of a system at another site, called a client/server.

server certificate. A digital certificate, issued by a CA to enable a Web server to conduct SSL-based transactions. When a browser connects to the server by using the SSL protocol, the server sends the browser its public key. This enables authentication of the identity of the server. It also enables encrypted information to be sent to the server. *See also* CA certificate, digital certificate, *and* browser certificate.

servlet. A server-side program that gives Java-enabled servers additional functionality.

SET. Secure Electronic Transaction.

SGML. Standard Generalized Markup Language.

SHA-1 (Secure Hash Algorithm). An algorithm that was designed by NIST and NSA for use with the Digital Signature Standard. The standard is the Secure Hash Standard; SHA is the algorithm that the standard uses. SHA produces a 160-bit hash.

sign. To use your private key to generate a signature. The signature is a means of proving that you are responsible for and approve of the message you are signing.

signing/verifying. To sign is to use a private digital key to generate a signature. To verify is to use the corresponding public key to verify the signature.

Simple Mail Transfer Protocol (SMTP). A protocol that transfers electronic mail over the Internet.

site certificate. Similar to a CA certificate, but valid only for a specific Web site. *See also* CA certificate.

smart card. A piece of hardware, typically the size of a credit card, for storing a user's digital keys. A smart card can be password-protected.

S/MIME. A standard that supports the signing and encryption of e-mail transmitted across the Internet. *See* MIME.

SMTP. Simple Mail Transfer Protocol.

SSL. Secure Sockets Layer.

Standard Generalized Markup Language (SGML). A standard for describing markup languages. HTML is based on SGML.

symmetric cryptography. Cryptography that uses the same key for both encryption and decryption. Its security rests in the key — revealing the key means that anyone could encipher and decipher messages. The communication remains secret only as long as the key remains secret. *Contrast with* asymmetric cryptography.

symmetric key. A key that can be used for both encryption and decryption. *See also* symmetric cryptography.

T

target. A designated or selected data source.

TCP/IP. Transmission Control Protocol/Internet Protocol.

top CA. The CA at the top of a PKI CA hierarchy.

TP. Trust Policy.

Transmission Control Protocol/Internet Protocol (TCP/IP). A set of communication protocols that support peer-to-peer connectivity functions for local and wide area networks.

triple DES. A symmetric algorithm that encrypts the plaintext three times. Although many ways exist to do this, the most secure form of multiple encryption is triple-DES with three distinct keys.

trust domain. A set of entities whose certificates have been certified by the same CA.

trusted computer base (TCB). The software and hardware elements that collectively enforce an organization's computer security policy. Any element or part of an element that can effect security policy enforcement is security-relevant and part of the TCB. The TCB is an object that is bounded by the security perimeter. The mechanisms that carry out the security policy must be non-circumventable, and must prevent programs from gaining access to system privileges to which they are not authorized.

trust model. A structuring convention that governs how certificate authorities certify other certificate authorities.

tunnel. In VPN technology, an on-demand virtual point-to-point connection made through the Internet. While connected, remote users can use the tunnel to exchange secure, encrypted, and encapsulated information with servers on the corporate private network.

type. *See* object type.

U

Unicode. A 16-bit character set that is defined by ISO 10646. The Unicode character encoding standard is an international character code for information processing. The Unicode standard encompasses the principal scripts of the world and provides the foundation for the internationalization and localization of software. All source code in the Java programming environment is written in Unicode.

Uniform Resource Locator (URL). A scheme for addressing resources on the Internet. The URL specifies the protocol, host name or IP address. It also includes the port number, path, and resource details needed to access a resource from a particular machine.

URL. Uniform Resource Locator.

user authentication. The process of validating that the originator of a message is the identifiable and legitimate owner of the message. It also validates that you are communicating with the end user or system you expected to.

UTF-8. A transformation format. It enables information processing systems that handle only 8-bit character sets to convert 16-bit Unicode to an 8-bit equivalent and back again without loss of information.

V

VPN. Virtual Private Network.

Virtual Private Network (VPN). A private data network that uses the Internet rather than phone lines to establish remote connections. Because users access corporate network resources through an Internet Service Provider (ISP) rather than a telephone company, organizations can significantly reduce remote access costs. A VPN also enhances the security of data exchanges. In traditional firewall technology, message content can be encrypted, but the source and destination addresses are not. In VPN technology, users can establish a tunnel connection in which the entire information packet (content and header) is encrypted and encapsulated.

W

Web browser. Client software that runs on a desktop PC and enables the user to browse the World Wide Web or local HTML pages. It is a retrieval tool that provides universal access to the large collection of hypermedia material available in the Web and Internet. Some browsers can display text and graphics, and some can display only text. Most browsers can handle the major forms of Internet communication, such as FTP transactions.

Web server. A server program that responds to requests for information resources from browser programs. *See also* server.

World Wide Web (WWW). That part of the Internet where a network of connections is established between computers that contain hypermedia materials. These materials provide information and can provide links to other materials in the WWW and Internet. WWW resources are accessed through a Web browser program.

X

X.500. A standard for putting into effect a multipurpose, distributed and replicated directory service by interconnecting computer systems. Jointly defined by the International Telecommunications Union (ITU), formerly known as CCITT, and the International Organization for Standardization and International Electro-Chemical Commission (ISO/IEC).

X.509 certificate. A widely-accepted certificate standard designed to support secure management and distribution of digitally signed certificates across secure Internet networks. The X.509 certificate defines data structures that accommodate procedures for distributing public keys that are digitally signed by trusted third parties.

X.509 Version 3 certificate. The X.509v3 certificate has extended data structures for storing and retrieving certificate application information, certificate distribution information, certificate revocation information, policy information, and digital signatures.

X.509v3 processes create time-stamped CRLs for all certificates. Each time a certificate is used, X.509v3 capabilities allow the application to check the validity of the certificate. It also allows the application to determine whether the certificate is on the CRL. X.509v3 CRLs can be constructed for a specific validity period. They can also be based on other circumstances that might invalidate a certificate. For example, if an employee leaves an organization, their certificate would be put on the CRL.

Index

Special Characters

A

B

C

D

E

F

I

J

K

L

M

O

P

R

S

T

V

W

Y

Program Number: 5697–F93

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.