novell

public key cryptography services

Preface

Novell Public Key Cryptography Services include several services that work together to allow you to protect confidential data transmissions over public communications channels such as the Internet.

- Novell International Cryptographic Infrastructure (NICI)
- Secure Authentication Services
- Novell PKI Services

This book describes how public key cryptography works, how to set up these services, and how to manage them.

chapter

1 Understanding Public Key Cryptography Services

Overview

This product includes public key cryptography services that allow you to protect confidential data transmissions over public communications channels such as the Internet.

Note If you are unfamiliar with public key cryptography concepts, see “Public Key Cryptography Basics” on page 11.

Public key cryptography presents unique challenges to network administrators. This product helps you meet these challenges in the following ways:

- **Provides public key cryptography services on your network**

  You can create an Organizational Certificate Authority within your NDS tree, use the services of an external certificate authority, or use a combination of both as your needs dictate.

- **Controls the costs associated with obtaining key pairs and managing public key certificates**

  You can create an organization-based certificate authority, generate unlimited key pairs, and issue unlimited public key certificates through the Organizational certificate authority at no charge.

- **Allows public keys and public key certificates to be openly available while also protecting them against tampering**

  Key pairs are stored in NDS and can therefore leverage NDS replication and access control features.

- **Allows private keys to be accessible to only the software routines that use them for signing and decrypting operations**

Private keys are encrypted by Novell International Crytographic Infrastructure (NICI) and made available only to the software routines using them for signing and decrypting operations.

- **Securely backs up private keys**

  Private keys are encrypted by NICI, stored in NDS, and backed up using standard NDS backup utilities.

Important The cryptography services available to you in this product depend on the country in which your network is located. Cryptography-enabled applications will not function if cryptography services are not fully installed. For example, the mass market exportable version of NICI is limited to 512-bit RSA keys for data encryption. The U.S. and Canadian version of NICI supports key sizes up to 2048 bits for all types of keys.

To ensure that you have the highest level of cryptography services available in your area, contact your Novell Authorized Reseller representative.

Product Components

The public key cryptography components available in this product include NICI, Novell Secure Authentication Service, and Novell Public Key InfrastructureServices (PKIS).

Novell International Cryptographic Infrastructure

Novell International Cryptographic Infrastructure (NICI) is the underlying cryptographic infrastructure that provides the cryptography for PKIS Services, Secure Authentication Service, and other applications. NICI is installed during the main product installation.

Novell Secure Authentication Service

Novell Secure Authentication Service includes support for the Secure Socket Layer (SSL) protocol. You can elect to install this component during the product installation.

When Novell Secure Authentication Service is installed, the Security Container object is created within NDS. This container holds security-related objects, including theOrganizational certificate authority object, for the NDS tree.

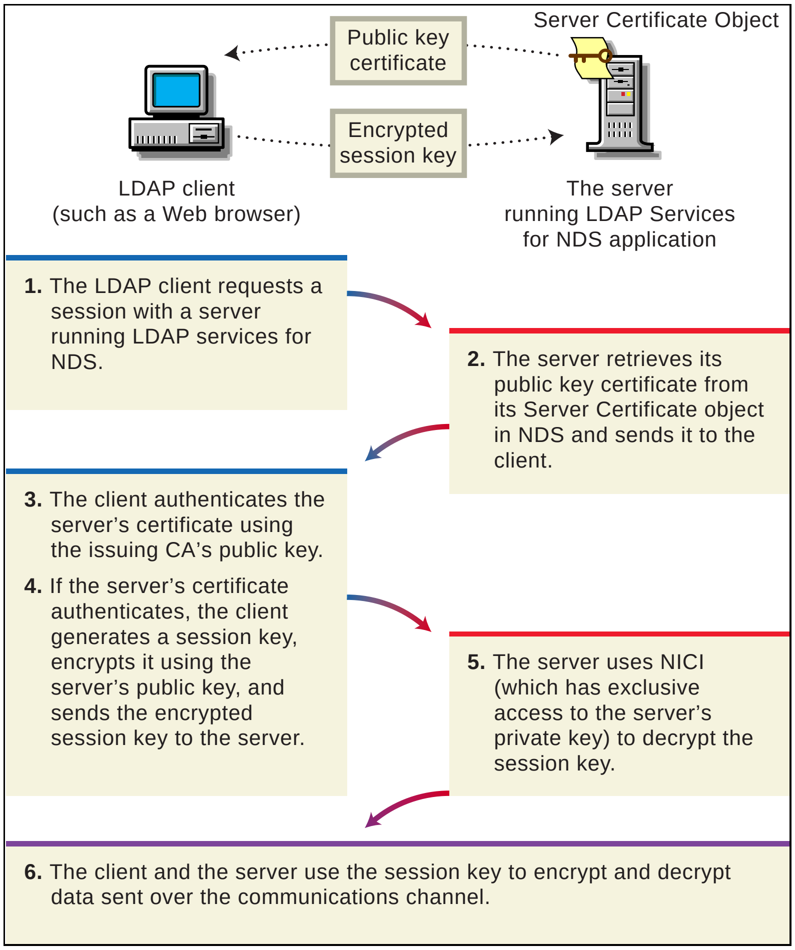
Secure Sockets Layer (SSL) is a protocol that establishes and maintains secure communications between SSL-enabled servers and clients across the Internet. Through a process called an *SSL handshake*, SSL allows a client and a server to establish a communication channel that prevents eavesdropping, tampering, and forgery.

The SSL handshake includes authentication. To implement the authentication and encryption processes, SSL uses the server's public/private key pair to exchange a session key. This session key is used for encrypting and decrypting all communications between the client and the server.

To prevent message tampering, SSL uses a message digest. To ensure message privacy, SSL provides for the creation and use of encrypted communication channels. To prevent message forgery, SSL allows the server and/or the client to authenticate each other when the secure connection is established.

Novell Secure Authentication Service and SSL are an integral part of LDAP Services, a cryptography-enabled application included in this product. LDAP requires only one-way, server-side authentication and requires that the authentication use public key cryptography. Therefore, LDAP Services must have a digital public key certificate in order to use SSL. You can create public key certificates for SSL through Novell PKI Services. The public key certificate includes a signature that is created either by the Organizational certificate authority or by an external certificate authority.

The following illustration shows the SSL authentication process when an LDAP client establishes a data transmission session with LDAP Services running on the server.



1. The LDAP client requests a session with a server running LDAP services for NDS.

2. The server retrieves its public key certificate from its Server Certificate object in NDS and sends it to the client.

3. The client authenticates the server's certificate using the issuing CA's public key.

4. If the server's certificate authenticates, the client generates a session key, encrypts it using the server's public key, and sends the encrypted session key to the server.

5. The server uses NICI (which has exclusive access to the server's private key) to decrypt the session key.

6. The client and the server use the session key to encrypt and decrypt data sent over the communications channel.

Novell PKI Services

Novell PKI Services consists of the PKI NLM (NetWare) or PKI_SERVER.DLL file (Windows NT) and a snap-in module to ConsoleOne, which is the administration point for PKI services. PKI Services allows you to request, manage, and store public key certificates and their associated key pairs in the NDS tree, and to establish an Organizational certificate authority that is specific to your NDS tree and your organization. You can elect to install Novell PKI Services as part of an integrated product installation, or you can install it separately using its own installation program.

Novell PKI Services derives all supported cryptography and signature algorithms, as well as supported key sizes, from NICI. Therefore, a single version of PKI Services can be used in installations throughout the world.

After installing Novell PKI Services, you will manage it through ConsoleOne. Initial setup tasks include the following:

- **Creating an Organizational certificate authority for your organization**

  During the product installation, you can elect to create an Organizational Certificate Authority (CA). You may also create or recreate the Organizational CA after the installation is completed.

  Th Organizational CA contains the public key, private key, certificate, certificate chain, and other configuration information for the Organizational certificate authority object.

  The private key is stored in the Organizational certificate authority object in NDS in encrypted form. Once a server is configured to provide the certificate authority service, it performs that service for the entire NDS tree. The Organizational certificate authority object resides in the Security container in NDS.

- **Creating a Server Certificate object for each cryptography-enabled application**

  During the product installation, you can create a Server Certificate object. You may create other Server Certificate objects after the installation is completed.

  The Server Certificate object contains the public key, private key, certificate, and certificate chain. The private key is stored in the Server Certificate object in encrypted form. This information is stored in NDS.

  A server can have many Server Certificate objects associated with it. Any cryptography-enabled applications running on a particular server that require keying material for their operation can be configured to use any one of the Server Certificate objects configured for that server. Multiple applications running on a given server can use the same Server Certificate object; however, a Server Certificate object cannont be used by applications on any other server.

  You can create Server Certificate objects only in the container where the server resides. If the Server Certificate object is moved, all Server

Certificate objects belonging to that server must be moved as well. You should not rename a Server Certificate object.

Note The key pair stored in the Server Certificate object is referenced by the name you enter when the key pair is created. The key pair name is not the name of the Server Certificate object. When configuring cryptography-enabled applications to use key pairs, you reference those keys by their key pair name, and not by the Server Certificate object name.

- **Requesting public key certificates from the Organizational certificate authority or from an external certificate authority.**

  A public key certificate is a digital message signed with a private key. It provides a cryptographic binding between the public key and a name.

  Public key certificates contain, at minimum, a public key, a subject name, an issuer name, a validity period, a serial number, and a certificate authority-generated signature. They may also contain specific extensions—for example, to further clarify the use of the certificate.

- **Creating a user certificate**

  A user certificate is a certificate hosted by a User object in NDS. Users have access to their own user certificates, which can be used for authentication, data encryption/decryption, digital signing, and secure email.

  Generally, only the network administrator has sufficient rights to create a user certificate. The user certificate is created from the Security tab of the user's property page.

  User certificates can only be signed by an NDS-based certificate authority. The private key is stored in the user's object in encrypted form. Any user can export any other user's public key certificate. However, the user who hosts the user certificate is the only person, including the network administrator, who has rights to export the private key.

  A User object can host multiple user certificates.

- **Creating a Trusted Root Container object**

  A Trusted Root Container object is an NDS object that contains Trusted Root Certificate objects.

  The Trusted root container object can reside in a number of different containers, but Novell recommends that you create the Trusted Root Container in the Security Container.

- **Creating a Trusted Root Certificate object**

  A Trusted Root Certificate object is an NDS object that contains a CA's certificate that is known to be authentic and valid. The Trusted Root Certificate can be exported and used as needed. Applications that are configured to use the trusted root certificate objects will consider a certificate valid if it has been signed by one of the CAs in the Trusted Root Container.

  The Trusted Root Certificate object must reside in a Trusted Root Container object.

For Additional Information

For instructions on installing and setting up public key cryptography components, see "Setting Up Public Key Cryptography Services" on page 19.

For information about administering public key cryptography services, see "Managing Public Key Cryptography Services" on page 29 and the Novell PKI Services online help within ConsoleOne.

For additional information about all of the Security Services products and technologies available from Novell, see the following web sites:

www.novell.com\security
www.novell.com\products\cryptography

Public Key Cryptography Basics

The content of most Internet communications, such as web page browsing or public chat forums, can be monitored by anyone equipped to do so. The content of other data transmissions, such as the exchange of credit card information for online purchases, needs to be kept private.

Public key cryptography is a widely used method for keeping data transmissions private and secure on the Internet. Specifically, public key cryptography is the system of using digital codes called "keys" to authenticate senders of messages and to encrypt message content.

Secure Transmissions

Data transmissions are private and secure when two things happen:

- **Authentication**—The data receiver knows that the data sender is exactly who or what it claims to be.

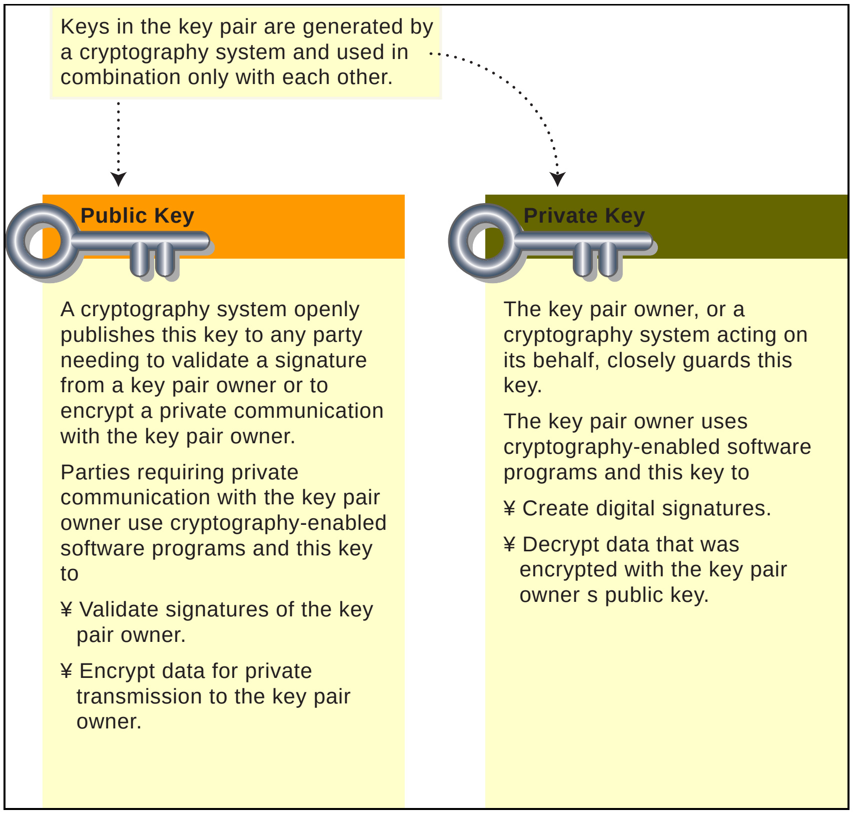
- **Encryption**—The data sent is encrypted so that it can be read only by the intended receiver.

Key Pairs

Authentication and encryption are both provided through the use of mathematically related pairs of digital codes or “keys.” One key in each pair is publicly distributed; the other is kept strictly private.

Each data transmitter, whether a person, a software program, or some other entity such as a bank or business, is issued a key pair by a public key cryptography system.

The basic principles and functions of each key in the key pair are summarized in the following illustration.



Keys in the key pair are generated by a cryptography system and used in combination only with each other.

Public Key

A cryptography system openly publishes this key to any party needing to validate a signature from a key pair owner or to encrypt a private communication with the key pair owner.

Parties requiring private communication with the key pair owner use cryptography-enabled software programs and this key to

¥ Validate signatures of the key pair owner.

¥ Encrypt data for private transmission to the key pair owner.

Private Key

The key pair owner, or a cryptography system acting on its behalf, closely guards this key.

The key pair owner uses cryptography-enabled software programs and this key to

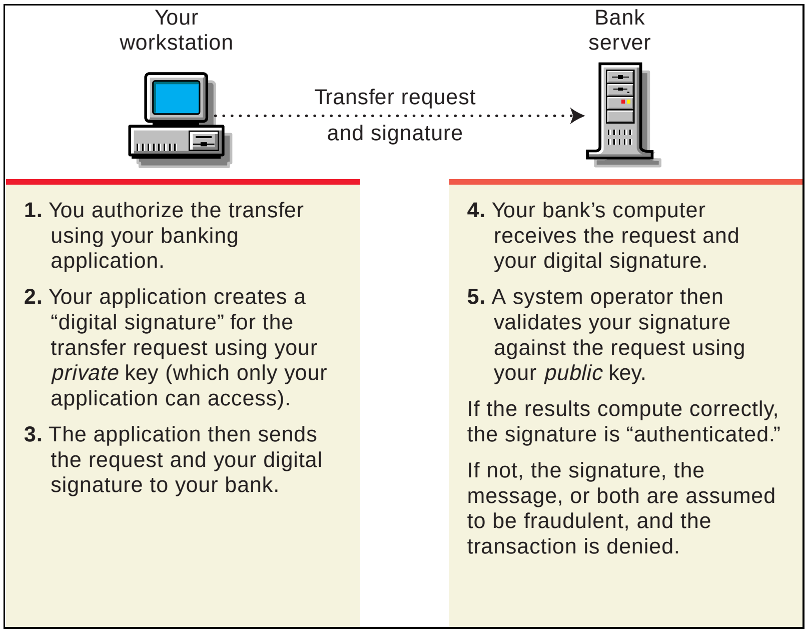
¥ Create digital signatures.

¥ Decrypt data that was encrypted with the key pair owner s public key.

Key Pairs and Authentication

Authentication means that the data receiver knows that the data sender is exactly who or what it claims to be.

Suppose that you want to authorize your bank to transfer funds from your account to another account. The bank needs proof that the message came from you and that it has not been altered during transit. The following illustrates the process that your online transaction would follow using public key cryptography.



1. You authorize the transfer using your banking application.
2. Your application creates a “digital signature” for the transfer request using your *private* key (which only your application can access).
3. The application then sends the request and your digital signature to your bank.
4. Your bank’s computer receives the request and your digital signature.
5. A system operator then validates your signature against the request using your *public* key.

If the results compute correctly, the signature is “authenticated.”

If not, the signature, the message, or both are assumed to be fraudulent, and the transaction is denied.

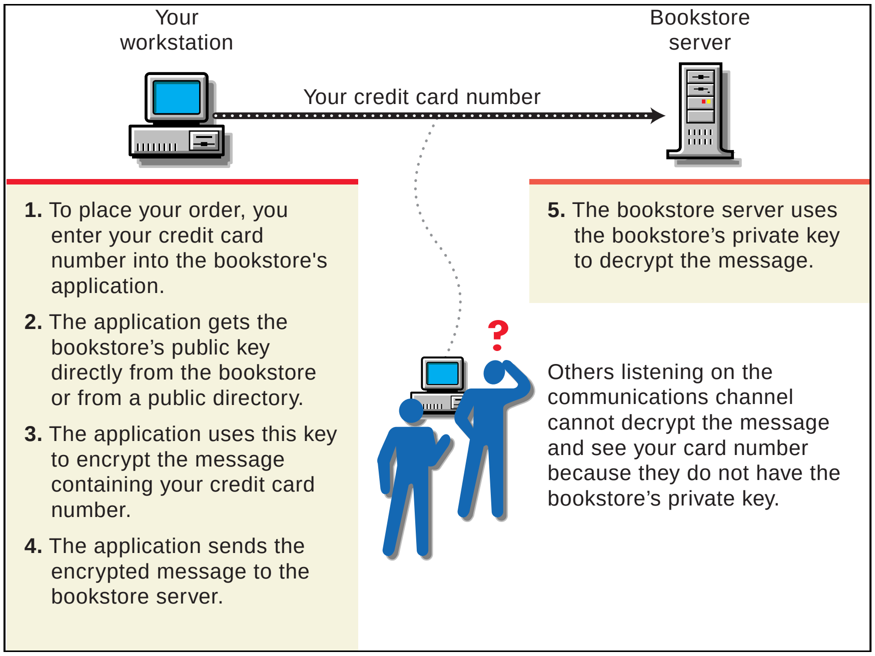
For information about digital signatures and their verification, see “Digital Signatures” on page 16 .

Key Pairs and Encryption

Encryption means that the data can be read only by the intended receiver.

Suppose you want to order a book from an Internet vendor and you need to use your credit card to pay for it. You don’t want your credit card number read by anyone other than the intended recipient.

The encryption process in the following illustration provides the mechanisms through which your credit card number can be safely transmitted.



1. To place your order, you enter your credit card number into the bookstore's application.

2. The application gets the bookstore’s public key directly from the bookstore or from a public directory.

3. The application uses this key to encrypt the message containing your credit card number.

4. The application sends the encrypted message to the bookstore server.

5. The bookstore server uses the bookstore’s private key to decrypt the message.

Others listening on the communications channel cannot decrypt the message and see your card number because they do not have the bookstore’s private key.

Establishing Trust

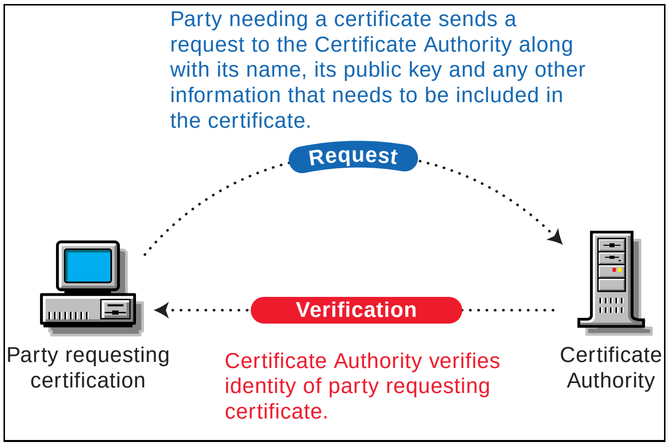
If a sender and receiver know and trust each other, they can simply exchange public keys and establish secure data transmission, including authentication and encryption. To do this, they would use each other’s public keys and their own private keys.

Under normal circumstances, however, parties needing secure data transmissions have no foundation for trusting the identity of each other. Each needs a third party, whom they both trust, to provide proof of their identity.

Certificate Authorities

A party needing to prove its identity in a public key cryptography environment enlists the services of a trusted third party known as a certificate authority.

The primary purpose of the certificate authority is to verify that a party is who or what it claims to be and then to issue a public key certificate for that party to use. The public key certificate verifies that the public key contained in the certificate belongs to the party named in the certificate.

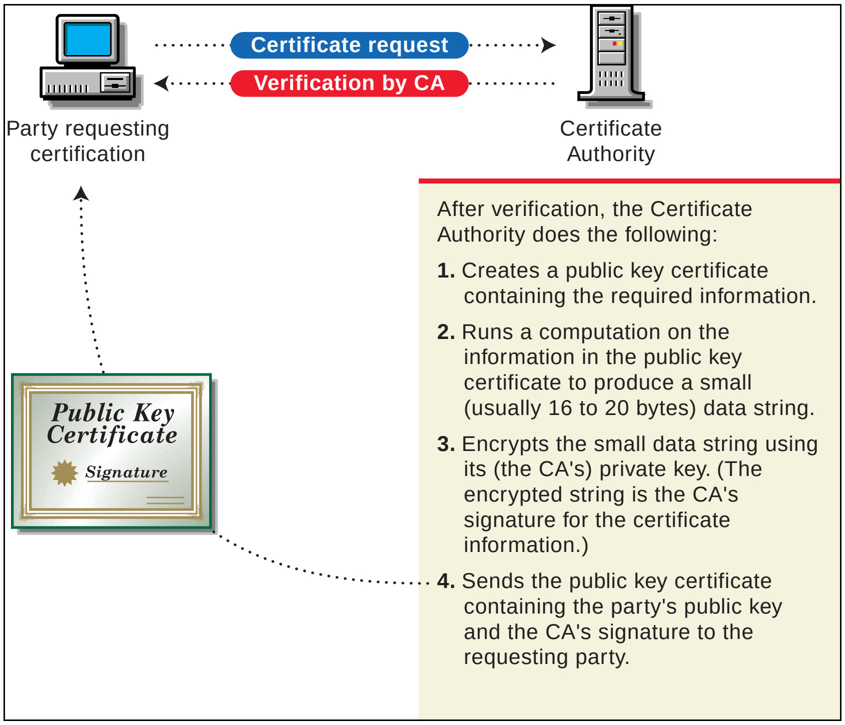

Party needing a certificate sends a request to the Certificate Authority along with its name, its public key and any other information that needs to be included in the certificate.
Request
Verification
Party requesting certification
Certificate Authority verifies identity of party requesting certificate.
Certificate Authority

Once the identity of the requesting party has been established to the satisfaction of the certificate authority, it issues an electronic “certificate” and applies its digital signature.

Digital Signatures

Just as a personal signature applied to a paper document indicates the authenticity of the document, a digital signature indicates the authenticity of electronic data.

To create a digital signature, the software used to create the signature links the data being signed with the private key of the signer. The following illustration shows the process that a CA follows to create its digital signature for a public key certificate.



After verification, the Certificate Authority does the following:

1. Creates a public key certificate containing the required information.

2. Runs a computation on the information in the public key certificate to produce a small (usually 16 to 20 bytes) data string.

3. Encrypts the small data string using its (the CA's) private key. (The encrypted string is the CA's signature for the certificate information.)

4. Sends the public key certificate containing the party's public key and the CA's signature to the requesting party.

A digital signature is uniquely linked to the signer and the data. No one else can duplicate the signature because no one else has the signer’s private key. In addition, the signer cannot deny having signed the data. This is known as *non-repudiation*.

When a certificate authority signs a public key certificate, it guarantees that it has verified the identify of the public key owner according to the certificate authority’s established and published policies.

After signed data (such as a public key certificate) is received, software verifies data authenticity by applying the same computation to the data that the signing software used originally. If the data is unaltered, both computations will produce identical results. It can then be safely assumed that neither the data nor the signature was modified in transit.

chapter

2 Setting Up Public Key Cryptography Services

This chapter explains how to

- Install the following components on all servers in the NDS tree that will run applications integrated with public key cryptography services:
    - NICI Cryptographic Modules appropriate to your locale. See “Installing NICI” on page 19.
    - Novell Secure Authentication Service including SSL. See “Installing Novell Secure Authentication Service” on page 21.
    - Novell PKI Services. See “Installing Novell PKI Services” on page 21.
- Set up Novell PKI Services for use on your network. See “Setting Up Novell PKI Services” on page 22.
- Ensure that the workstation that you will use to administer Novell PKI Services meets the requirements for using ConsoleOne. See “Workstation Requirements.”

Installing NICI

In order to run Novell PKI Services, Novell International Cryptography Infrastructure (NICI) must be installed on the server. PKIS 2.0 requires at least NICI 1.3.1. The exportable version of NICI 1.3.1 is included in the PKIS20.EXE self-extracting file or it can be downloaded from Novell’s web site. U.S. and Canadian customers who would like the domestic version of NICI 1.3.1 should contact Novell.

Note NICI is not part of the PKIS 2.0 installation process. It must be installed separately.

Important If you are running an older version of NICI that contains domestic-grade crypto (128 bit), you will need to obtain domestic-grade NICI 1.3.1 from Novell. Please contact Novell for the domestic version of NICI 1.3.1.

If you are running an older version of NICI that contains exportable-grade crypto (56 bit), you can upgrade to NICI 1.3.1 by simply installing NICI 1.3.1 on the server.

To determine what version of NICI you are running:

1. **Load NWCONFIG at the server console.**
2. **Select Product Options.**
3. **Select View/Configure/Remove Installed Products.**

   If you have NICU0 listed, this means your verions of NICI is domestic and needs to be deleted. I you have NICIW1 listed, this means your version of NICI is exportable and can be upgraded by installing NICI 1.3.1.

To install NICI:

1. **Download and extract PKIS20.EXE.**

   This will be extracted to C:\NOVELL\PKIS20. It will also create a directory at C:\NOVELL\PKIS20\NICI 1.3.1 that will contain the self-extracting file NICI-W1.EXE. This is the exportable version of NICI.

2. **Run NICI-W1.EXE (exportable) or NICI-U0.EXE (domestic) and extract its contents to a diskette.**
3. **Insert the diskette into the server's diskette drive.**
4. **Enter LOAD NWCONFIG at the server's console.**
5. **Highlight Product Options > Install a product not listed.**
6. **Press Enter.**

   You will be prompted to insert the first diskette of the product you wish to install.

7. **Press Enter.**

   The NICI installation will begin.

8. **Follow the on-screen instructions.**

Installing Novell Secure Authentication Service

Secure Authentication Service is installed automatically during the installation process. When you click Finish, the Security container is created, if it does not already exist, and the SAS Service object is created and placed in the Organization container.

Note You must have administrative rights to the root of the tree in order t create the Security container. You must have Create rights to the server's container in order to create the Secure Authentication Service object for the server.

Installing Novell PKI Services

To install PKIS 2.0 from a web download:

1. **Extract the files.**

   Locate the downloaded file (PKIS20.EXE) and double-click the file. By default, the files will be extracted to C:\NOVELL\PKIS20.

2. **Log in as the server administrator from the workstation from which you will be installing PKIS 2.0.**

3. **Run INSTALL.EXE from the extract directory.**

   This launches the installation wizard.

4. **Complete the product installation.**

To install PKIS 2.0 from a CD:

1. **Log in as the server administrator from the workstation from which you will be installing PKIS 2.0.**

2. **Run INSTALL.EXE from the CD's NOVELL\PKIS20 directory.**

   This launches the installation wizard.

3. **Complete the product installation.**

The installation installs PKIS 2.0 and the manageement utility on the selected server. In addition, the installation creates the Security container and server SAS object, if they did not already exist.

Setting Up Novell PKI Services

After Novell PKI Services is installed, you must set it up for use on your network by completing the following tasks:

- “Decide Which Type of Certificate Authority to Use” on page 22.
- “Create an Organizational Certificate Authority Object” on page 23.
- “Create Server Certificate Objects” on page 24 for applications whose secure transmissions you want to manage.
- “Configure Your Cryptography-Enabled Applications to Use Server Certificates” on page 26.

Decide Which Type of Certificate Authority to Use

You can manage public key certificates, Server Certificate objects, and their associated components and sign the public key certificates using either an Organizational certificate authority or an external certificate authority. During the Server Certificate object creation process, you will be asked which type of certificate authority will sign the Server Certificate object.

The Organizational certificate authority is specific to your organization and uses an organizational-specific public key for signing operations. The private key is created when you create the Organizational certificate authority.

An external certificate authority is managed by a third party outside of the NDS tree. An example of an external certificate authority is VeriSign*.

Both types of certificate authorities can be used simultaneously. Using one type of certificate authority does not preclude the use of the other.

Benefits of Using an Organizational Certificate Authority

- **Compatibility**. An Organizational certificate authority is compatible with other applications that share a common trusted root in NDS. These include BorderManager, LDAP Services, and future products using Novell security.
- **Cost savings**. An Organizational certificate authority lets you create an unlimited number of public key certificates at no cost; obtaining a single

public key certificate through an external certificate authority might cost hundreds of dollars.

- **Component of a complete and compatible solution**. By using the Organizational certificate authority, you can use the complete cryptographic system build into NDS without having to rely on any external services. In addition, Novell PKI Services is compatible with a wide range of Novell products.

- **Certificate attribute and content control**. An Organizational certificate authority is managed by the network administrator, who decides upon public key certificate attributes such as certificate life span, key size, and signature algorithm.

- **Simplified management**. The Organizational certificate authority performs a function similar to external certificate authorities but without the added cost and complexity.

Benefits of Using an External Certificate Authority

- **Liability**. An external certificate authority might offer some liability protection if, through the fault of the certificate authority, your private key was exposed or your public key certificate was misrepresented.

- **Availability**. An external certificate authority may be more widely available and compatible with applications outside of NDS.

Create an Organizational Certificate Authority Object

The PKIS 2.0 installation process can create the Organizational CA for you. You will be prompted to specify an Organizational CA name. When you click Finish, the Organizational CA will be created with the default parameters and placed in the Security container.

If you desire more control over the creation of the Organizational CA, you can create the Organizational CA manually.

To create the Organizational certificate authority

1. **Launch ConsoleOne.**

2. **Expand the NDS Tree in which you would like to create the Organizational certificate authority.**

   This reveals the Security Container.

3. **Right-click the Security Container and select New Object.**

4. **From the list box in the New Object dialog box, double-click NDSPKI:Certificate Authority.**

   This opens the Create an Organizational Certificate Authority Object dialog box and the corresponding wizard that creates the object. For information specific to the dialog box or any of the wizard pages, click Help.

Important During the creation process, you will be prompted to name the Organizational certificate authority object and to choose a server on which the certificate authority service will run.

Select a server that is physically secure, that will be available when needed to perform signing operations, that runs a protocol which is compatible with the other servers in your organization (i.e., IP, IPX, IP/IPX), and that only runs software that you trust. It is important that your server meet these conditions, because the Organizational certificate authority object is the centerpiece of your PKI system and if the server that contains the object is compromised, your entire PKI system could be compromised as well.

Create Server Certificate Objects

Server Certificate objects are created in the container that holds the server's NDS object. We recommend that you create a separate Server Certificate object for each cryptography-enabled application on the server.

The PKIS 2.0 installation process can create a Server Certificate object for you. You will be prompted to specify a Server Certificate object name. When you click Finish, the Server Certificate object will be created with the default parameters and placed in the server's container.

If you desire more control over the creation of the Server Certificate object, you can create the Server Certificate object manually.

To create additional Server Certificate objects

1. **Launch ConsoleOne.**

2. **Right-click the container object that contains the server that will run your cryptography-enabled applications; then choose Create.**

3. **From the list box in the New Objects dialog box, double-click NDSPKI:Key Material.**

   This opens the Create a Server Certificate dialog box and the corresponding wizard that creates the Server Certificate object. For information specific to the dialog box or any of the wizard pages, click Help.

Hints for Creating Server Certificates

During the Server Certificate object creation process, you will be prompted to name the key pair and to choose the server that the key pair will be associated with. The Server Certificate object is generated by Novell PKI Services and its name is based on the key pair name that you choose.

You will also be asked to specify whether the Server Certificate object will be signed by your organization's Organizational certificate authority or by an external certificate authority. For information about making this decision, see "Decide Which Type of Certificate Authority to Use" on page 22.

If you decide to use your organization's Organizational CA, the server that the Server Certificate object is associated with must be able to communicate with the server which hosts the Organizational CA, or it must be the same server. These servers must be running the same protocol (IP/IPX).

If you decide to use an external certificate authority to sign the certificate, the server that the Server Certificate object is associated with will generate a certificate signing request that you will need to submit to the external certificate authority. After the certificate is signed and returned to you, you will need to install it into the Server Certificate object, along with the trusted root for the external certificate authority. For information specific to any of the wizard pages, click Help.

Once you have created the Server Certificate object, you can configure your applications to use them. (See "Configure Your Cryptography-Enabled Applications to Use Server Certificates" on page 26.) Keys are referenced in the application's configuration by the key pair name that you entered when you created the Server Certificate object.

Configure Your Cryptography-Enabled Applications to Use Server Certificates

Once you have configured Novell PKI Services, you must configure your individual cryptography-enabled applications so that they can use the Server Certificate objects that you created. The configuration procedures will be unique to the individual applications, so we recommend that you consult the application's documentation for specific instructions.

Additional Components to Set Up

PKIS 2.0 includes some additional components that can be set up. These components are not required to make PKIS function properly, but they do provide important functionality.

Create User Certificates

To create user certificates

1. **Launch ConsoleOne.**

2. **Double click on the user object that will host the user certificate.**

3. **Click Create.**

   This opens a wizard that helps you create the user certificate. For specific information on the wizard pages, click Help.

Create a Trusted Root Container Object

Novell strongly recommends that you create all trusted root containers in the security container.

To create a trusted root container object

1. **Open ConsoleOne.**

2. **Right click the container you want to create the Trusted Root Container object in and click New > Object.**

   We recommend the Security container.

3. **From the list box in the New Object dialog box, double-click NDSPKI:Trusted Root.**

   This opens the New NDSPKI:Trusted Root wizard that helps you create the trusted root container. For specific information on the wizard pages, click Help.

Create Trusted Root Certificate Objects

A Trusted Root Certificate object can only reside in a Trusted Root Container object.

To create a Trusted Root Certificate object:

1. **Open ConsoleOne.**

2. **Open the Security container.**

3. **Right-click the Trusted Root Container object and click New > Object.**

4. **From the list box in the New Object dialog box, double-click NDSPKI:Trusted Root Certificate.**

   This opens the Create a Trusted Root Certificate Object wizard that helps you create the trusted root certificate object. For specific information on the wizard pages, click Help.

chapter

3 Managing Public Key Cryptography Services

As a system administrator, you will be required to perform several management tasks in maintaining the public key cryptography services. These tasks are performed within ConsoleOne. This chapter provided a brief overview of each task. For specific information on how to perform the tasks, see the ConsoleOne Help.

Certificate Authority Tasks

- Issuing a Certificate
- Viewing the Organizational Certificate Authority’s Properties

Server Certificate Object Tasks

- Importing a Public Key Certificate

User Certificate Tasks

- Viewing a User Certificate’s Properties
- Exporting a User Certificate

Trusted Root Certificate Object Tasks

- Viewing a Trusted Root Certificate Object’s Properties
- Replacing a Trusted Root Certificate

NDS Tasks

- Merging Two Trees That Have Security Containers

Certificate Authority Tasks

Issuing a Certificate

This task allows you to generate certificates for cyptography-enabled applications that do not recognize Server Certificate objects.

Your Organizational CA works the same way as an external CA, in that it has the ability to sign requests and issue certificates. You can issue certificates using your Organizational CA when a user sends a Certificate Signing Request (CSR) to you for signing. The user requesting the certificate can then take the issued certificate and import it directly into the cryptography-enabled application.

Viewing the Organizational CA’s Properties

ConsoleOne allows you to view the Organizational CA’s properties. Besides the NDS rights and properties that are viewable with any NDS object, you can also view properties specific to the Organizational CA, including the properties of the public key certificate and the self-signed certificate associated with it.

These properties provide you with the information you need to perform any task related to this object.

Server Certificate Object Tasks

Importing a Public Key Certificate

You import a public key certificate after you have issued a certificate signing request (CSR) and the Certificate Authority (CA) has returned a signed public key certificate to you. This task applies when you have created a Server Certificate object using the Custom option with the External CA signing option.

The External CA will return two certificates: a signed public key certificate, which verifies your identity, and a trusted root certificate, which verifies the CA’s identity. These certificates can then be imported and stored in the Server Certificate object. The cryptography-enabled application that is linked to this Server Certificate object can then use this information to perform secure transactions.

Export a Trusted Root Certificate

You export a trusted root certificate to a file so that a client (such as an Internet browser) can use it to verify the certificate chain sent by a cryptography-enabled application.

You can export a trusted root certificate in two file formats: DER encoded (.DER) and Base64 encoded (.B64). The .CRT extension can also be used for DER-encoded certificates.

If you export to the system clipboard, you can then paste the certificate directly into a cryptography-enabled application, if supported. The certificate exists on the clipboard in Base64 encoded format.

If you export to a file, you can specify either format. DER encoded format is the default format. It is the same as CRT format and can be used with applications that accept CRT formats.

Delete a Server Certificate Object

You should delete a Server Certificate object if you suspect that the private key has been compromised, if you no longer want to use the key pair, or if the trusted root in the Server Certificate object is no longer trusted.

Once the Server Certificate object is deleted, you will not be able to recover it. Before you delete this object, make sure that no cryptography-enabled applications still need to use it.

You can re-create a Server Certificate Object, but you will have to reconfigure any applications that referenced the old object.

View a Server Certificate Object's Properties

ConsoleOne allows you to view the Server Certificate object's properties. Besides the NDS rights and properties that are viewable with any NDS object, you can also view properties specific to the Server Certificate object, including the properties of the public key certificate and the trusted root certificate associated with it, if they exist.

These properties provide you with the information you need to perform any task related to this object.

User Certificate Tasks

View a User Certificate's Properties

ConsoleOne allows you to view the user certificate's properties. Besides the NDS rights and properties that are viewable with any NDS object, you can also view properties specific to the user certificate, including the issuer, the certificate status, the private key status and the validation period.

These properties provide you with the information you need to perform any task related to this object.

Export a User Certificate

This task allows a user or network administrator to export a user certificate for use in various secure transactions, such as e-mail.

The user Certificate can be exported with or without the private key. The network administrator or another user with sufficient rights can export a user certificate without the private key. But the only person who can export the user certficate with the private key is a person authenticated to the NDS tree as the user who hosts the User Certificate. No other user, even the network administrator, has rights to export a user's private key.

You can export a User Certificate from a user's property page under the Security tab.

Trusted Root Certificate Object Tasks

View a Trusted Root Certificate Object's Properties

ConsoleOne allows you to view the Trusted Root Certificate object's properties. Besides the NDS rights and properties that are viewable with any NDS object, you can also view properties specific to the Trusted Root Certificate object, including the issuer, the certificate status, and the validation period.

These properties provide you with the information you need to perform any task related to this object.

Replace a Trusted Root Certificate

This task allows you to replace a Trusted Root Certificate that is stored in the Trusted Root Certificate object. This task should be performed if the Trusted Root Certificate has expired.

You can replace a Trusted Root Certificate from the Trusted Root Certificate object's property page.

NDS Tasks

Merging Two Trees That Have Security Containers

In normal operation, there should never be a need to delete the Security container. However, if your NDS tree is to be merged with another NDS tree and both trees have a Security container, one of the Security containers must be deleted in order to successfully complete the merge.

Deleting the Security container in an NDS tree will have serious consequences for the Organizational CA and Server Certificate objects.

Issues Relating to the Organizational CA

Before you can delete the Security container, you must first delete the Organizational CA object. You cannot simply move the Organizational CA to a new container.

Deleting the Organizational CA object invalidates the Organizational CA and any public key certificates it issued. These public key certificates are stored in Server Certificate objects throughout the NDS tree. Public key certificates signed by the deleted Organizational CA remain valid for a short period of time. However, these public key certificates should be replaced by new public key certificates issued by a new Organizational CA or by an external CA.

Issues Relating to Server Certificate Objects

Server Certificate objects contain public key certificates used for authentication purposes by cryptography-enabled applications such as LDAP Services for NDS and Novell BorderManager. To authenticate to other cryptography-enabled applications, these applications must recognize who issued the service's public key certificate. This is enabled by installing the issuer's public key certificate into the other cryptography-enabled application. When installed in a cryptography-enabled application, such public key certificates are referred to as *trusted root certificates*.

When a service's public key certificate is replaced by another certificate that has a different issuer, the cryptography-enabled applications that the service needs to authenticate to will need to be updated. The public key certificate for the deleted Organizational CA must be deleted from the other cryptography-enabled application's list of trusted root certificates, and the public key certificate for the new issuer must be added.

Restoring or Re-creating the Security Container

After deleting a Security container, you can restore it from backup if needed. If no backups are available, an administrator with Supervisor rights at the [Root] of the NDS tree can re-create the Security container by installing Secure Authentication Services on a server in the NDS tree. Doing so creates the Security container under [Root].

The network administrator can then create a new Organizational CA object in the Security container using ConsoleOne. The network administrator must then replace the public key certificates signed by the previous Organizational CA and update any cryptography-enabled applications that have the previous Organizational CA's public key certificate in its list of trusted root certificates.