In August 1999, a Swedish hacker circulated an e-mail message that provided a URL to a website that contained only a red box followed by "@hotmail.com." For 12 hours, this website enabled anyone to read or to send unauthorized e-mail messages simply by entering a username belonging to any one of 40 million users of Microsoft Hotmail, a free e-mail service.

You may be tempted to minimize the implications of this incident, attributing the security breach to the fact that Hotmail is a web-based service. You may be equally tempted to dismiss this incident's message, as others have, by viewing it only as an example of Microsoft's disregard for security or possibly its clever marketing. (See "Experts: Hotmail Hack Easy, Office Not Quite Fixed," Network World, Sept. 1, 1999. You can download this article at http://www.nwfusion.com/news/1999/0901hotmail.html. See also "Microsoft's Dastardly Plot," PC Magazine, Oct. 26, 1999. You can download this article at http://www.zdnet.com/zdnn/stories/comment/0,5859,2381921,00.html.)

But resist the temptation because the more important moral to the Hotmail story applies to you and to every other network administrator—regardless of the e-mail program you use.

Like all morals, this one teaches nothing new but instead reinforces what you already know: Information exchanges over the Internet are inherently insecure. For example, while arguably remote, the possibility exists for an unauthorized user to read e-mail intended only for you or to send bogus e-mail that appears to be from you.

The question is what can you do to eliminate or at least minimize this possibility? Novell's answer to this question is Novell Certificate Server 2.0.

**WHAT IS NOVELL CERTIFICATE SERVER 2.0?**

Novell Certificate Server 2.0 is a NetWare Loadable Module (NLM) called the Public Key Infrastructure (PKI) NLM. To help you manage and use Novell Certificate Server 2.0, Novell also includes a ConsoleOne snap-in module and Novell Certificate Console, which is a client utility for users.

The PKI NLM provides a set of services that integrate with Novell Directory Services (NDS) and interface with Novell International Cryptographic Infrastructure (NICI) to establish a PKI on your company's network. PKI is an intimidating industry acronym that refers in part to the technology, cryptographic framework, and services required to generate and store public-private key pairs. (For more information about PKIs and related concepts, see "Background Basics" on p. 8.)

A PKI provides a way to associate public-private key pairs with the entities (such as the servers or users) for whom these key pairs were generated. The resulting collection of information is called a digital certificate, which is the real star in any PKI production. In fact, the point of having a PKI is to mint (or create) digital certificates for servers and users.

As you may know, Novell Certificate Server 2.0 is not Novell's first PKI product. Novell Certificate Server 2.0 enhances two Novell products: PKI Services (PKIS) 1.0, which is included with NetWare 5.0, and PKIS 1.9, which is included with NDS 8.

Like Novell's previous PKI products, Novell Certificate Server 2.0 is available at no cost. Novell Certificate Server 2.0 is bundled with NetWare 5.1 and is fully integrated with the NetWare 5.1 installation program. In addition, Novell Certificate Server 2.0 is available as a separate product, which you can download from http://www.novell.com/download/#NDS.

**MINTING CERTS**

Novell Certificate Server 2.0 mints digital certificates that comply with the X.509 v3 certificate standard. X.509 v3 certificates include several fields of information, including the certificate owner's name and public key, but do not contain the...
Background Basics

A Public Key Infrastructure (PKI) is a set of services that enable server and user applications, such as e-mail programs, to use public-key cryptography and digital signatures to secure Internet communications. Public-key cryptography refers to an encryption method in which each party that is exchanging messages has a pair of keys: a public key and a private key. (A key is a long prime number that represents one of a large number of possible values. For more information about keys and their relative security, see “The Key to Security” sidebar, “Virtual Private Networks: Making a Public Network Private,” NetWare Connection, Feb. 1998, p. 12. You can download this sidebar from http://www.nwconnection.com/feb.98/vpn28/thekey28.html.)

Each party’s private key, as the name suggests, is kept secret. In contrast and as its name suggests, public keys can be made public. Anything you encrypt using one key, someone else can only decrypt using the other associated key. For example, if you wanted to send me a private message, you could encrypt the message using my public key, and I could then use my associated private key to decrypt the message. That way, the message would be confidential, that is, only the intended recipient (in this case, me) would be able to read it.

Encrypting and decrypting a message may ensure the confidentiality of a message, but it does little to ensure that the message comes from the person claiming to have sent it. In fact, in the faceless, online world of the Internet, how can you ever be sure that the person or server that you are communicating with is really the person or server you think it is?

AN ONLINE PASSPORT AUTHORITY

Enter the Certificate Authority (CA). A CA is a trusted third party that issues certificates in the same way a passport authority issues passports. Like physical passports, these certificates, or online passports, verify the identity of the user or server to which those passports belong.

Certificates contain a number of fields, including the name of the issuing CA, the name of the user or server the certificate belongs to, the public key of the user or server this certificate belongs to, the public key algorithm, and the signature algorithm. The public key algorithm is used to encrypt messages, and the signature algorithm is used to digitally sign messages.

A DIGITAL SIGNATURE

Applications create digital signatures by running a chunk of data through an algorithm to produce a large numerical value called a hash. This hash is then encrypted using the sender’s private key. A digital signature enables recipients to verify the integrity of a message, that is, to verify that the message has not been changed in transit.

For example, if I sent you a signed message, my e-mail program would run that message through an algorithm (specified in my certificate) to produce a hash. I would then encrypt that hash using my private key.

After receiving my signed message (which would include my certificate that specifies the signature algorithm I used), your e-mail program would complete the following tasks:

- Your e-mail program would run the signed message through the same signature algorithm to produce its own hash.
- Your e-mail program would use my public key (also contained in my certificate) to decrypt the encrypted hash that my e-mail program produced.

Your e-mail program would then have two values: The received hash (recently unencrypted) and the newly created hash. If the two hashes are the same, your e-mail program can be sure that the contents of the certificate have not been changed.
an internal CA and mint no-cost certificates. Whether you install Novell Certificate Server 2.0 as part of a NetWare 5.1 installation or you install the separate downloaded version, you will be prompted to create an Organizational CA during installation.

Of course, you can always respond “No” to the prompt and create the Organizational CA later using the ConsoleOne snap-in module for Novell Certificate Server 2.0. However, if you respond “Yes” when prompted to create an Organizational CA, the installation program automatically installs Novell Certificate Server 2.0 and, by default, creates an Organizational CA in the Security container of your company’s NDS tree. (See Figure 2 on p. 14.)

If you choose to create an Organizational CA when you install Novell Certificate Server 2.0, the installation program also asks you if you would like to create a server certificate. Of course, you do not have to create a server certificate at this point; you can later create as many certificates as you like using the ConsoleOne snap-in module for Novell Certificate Server 2.0. However, if you respond “Yes” to the prompt to create a server certificate, the installation program conveniently mints a server certificate for you.

If you install Novell Certificate Server 2.0 as part of a NetWare 5.1 installation and choose to create a server certificate, the installation program can also configure several Novell programs to use the certificate you mint during installation.

For example, depending upon your responses to prompts, the NetWare 5.1 installation program can configure Novell’s LDAP Services for NDS, NetWare Enterprise Web Server, and NetWare Management Portal to use the certificate you mint during installation.

**Not Just Any Server Will Do**

When you create the Organizational CA, you will be prompted to enter the name of a specific server that you want to function as this CA. Because the public key and, more important, because the private key is generated on the server you name as the Organizational CA, you should choose this server carefully. The private key generated on this server can sign and, therefore, legitimize all of the certificates you create in-house for servers and users. You should select a highly secure server to ensure the sanctity of the CA’s private key.

“Of course, you could pick a server out on the sidewalk,” jokes Kevin Ward, engineering manager of Novell PKI products, “but it wouldn’t be a good idea.” What you really want, Ward continues, is a server that “is housed behind a glass wall or that is in a protected room and that runs only software you know and trust.” The more secure the server, the less chance the security for the CA’s private key will be compromised.

**The CA’s Key Pair**

After you name a specific server, the ConsoleOne snap-in module for Novell Certificate Server 2.0 sends a message to the NICI layer on that server, asking NICI to generate a key pair for the internal CA. NICI is the modular infrastructure of network cryptographic services included with NetWare 5.0 and above. NICI provides all of the cryptographic services Novell Certificate Server 2.0 requires.

NICI complies with international import and export restrictions and, therefore, can provide the strongest cryptographic algorithm legally allowed in a particular country. Because Novell Certificate Server 2.0 is NICI-enabled, you can run the same version of Novell Certificate Server 2.0 anywhere in the world—despite the differences in cryptographic laws. (For more information about NICI, see “With NICI It’s All Holes Barred,” NetWare Connection, Dec. 1998, pp. 8–20. You can download this article from http://www.nwconnection.com/past.)

The NICI running on the server you specify generates the public-private key pair for the internal CA. NICI generates the strongest keys—that is, the largest values—supported by NICI on that server. For example, if the server is running the NICI version with the Domestic cryptographic engine (which is available to companies in the United States and Canada), NICI generates 2048-bit keys. If the server is running the NICI version with the Worldwide cryptographic engine (which is available to companies in other countries that allow the importation of cryptographic products), NICI generates 512-bit keys.

NICI also encrypts the CA’s private key, and Novell Certificate Server 2.0 then stores this key as a hidden attribute of the Organizational CA object. Novell Certificate Server 2.0 places the CA’s public key in an X.509 v3 certificate and publishes this certificate in NDS.

**WHO NEEDS CERTS?**

Generally speaking, servers and users use the certificates you mint using Novell Certificate Server 2.0 to secure Internet communications. However, to be more precise, server applications and user applications use certificates to secure Internet communications. In other words, certificates are useless unless applications can put those certificates to work.
The X Fields

Novell Certificate Server 2.0 issues X.509 v3 certificates and supports server certificates from any commercial Certificate Authority (CA) that also issues X.509 v3 certificates. X.509 is a standard from the International Telecommunication Union-Telecommunications Standardization Sector (ITU-T) and the International Standards Organization/International Electrotechnical Commission (ISO/IEC). First published in 1988 and extended in 1993, the current version (v3) was released in 1996.

X.509 v3 certificates contain several fields, the names and functions of which are described below. (See also Figure 1 on p. 10.)

<table>
<thead>
<tr>
<th>FIELD</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>This field simply shows the version of X.509 upon which the certificate is based.</td>
</tr>
<tr>
<td>Issuer</td>
<td>This field identifies the issuing CA.</td>
</tr>
<tr>
<td>Subject</td>
<td>This field provides the name of the certificate’s owner. (For more information about this field, see the FAQ at <a href="http://support.novell.com/beta/public/bpkis/linker.htm">http://support.novell.com/beta/public/bpkis/linker.htm</a>.)</td>
</tr>
<tr>
<td>Public Key</td>
<td>This field provides the certificate owner’s public key.</td>
</tr>
<tr>
<td>Validity Period</td>
<td>This field indicates the period of time during which this certificate is valid.</td>
</tr>
<tr>
<td>Public Key Algorithm</td>
<td>This field shows the algorithm from which this certificate owner’s public key was generated. Novell Certificate Server 2.0 supports the Rivest, Shamir, Adleman (RSA) public-key algorithm with key lengths of 512 bits for information exchanges that occur outside the United States and Canada and key lengths as high as 2,048 bits for information exchanges that occur within the United States and Canada.</td>
</tr>
<tr>
<td>Signature Algorithm</td>
<td>This field shows the algorithm that this certificate owner uses to digitally sign messages. Novell Certificate Server 2.0 supports several signature algorithms, including message digest 2 (MD2), MD5, and Secure Hash Algorithm (SHA)-1. Although you can choose which of these algorithms you want to use, Novell recommends that you use SHA-1.</td>
</tr>
<tr>
<td>Optional Attributes</td>
<td>These fields are optional and may or may not contain additional information that can be used to better control and manage certificates.</td>
</tr>
</tbody>
</table>

Within the X.509 v3 Optional Attributes field, Novell Certificate Server 2.0 inserts extensions called Novell security attributes. (For more information about these attributes, see “Novell Certificate Extension Attributes—Novell Security Attributes.” You can download this document from Novell’s developer web site at http://developer.novell.com/whitepapers.)

Novell’s security attributes are noncritical. That is, an application that receives a certificate containing Novell’s security attributes won’t reject that certificate even if the application doesn’t understand the syntax and semantics of these attributes.

Novell security attributes provide information about the quality of the public-key pair and the certificate’s relative level of security. These security attributes also provide a unique identifier for an enterprise. The purpose of these fields, according to Kevin Ward, engineering manager of Novell Public Key Infrastructure (PKI) products, is to provide an infrastructure for enabling e-commerce.

Fortunately, a number of applications can already use the certificates Novell Certificate Server 2.0 mints and manages, and Novell expects more applications will soon be able to use these certificates. To encourage third-party developers to write applications that can consume the server and user certificates that Novell Certificate Server 2.0 mints and manages, Novell has made a developer kit available at no cost. (You can download this developer kit from http://www.developer.novell.com/ndk/ncslib.htm.)

Server applications typically use server certificates to establish authenticated and encrypted Secure Sockets Layer (SSL) sessions over the Internet with other servers or with users. Many Novell products and applications can use certificates for this purpose and can directly access from NDS the server certificates Novell Certificate Server 2.0 mints. For example, the following products and applications can use these certificates:

- LDAP Services for NDS
- BorderManager Enterprise Edition
- NetWare 5.1 applications such as NetWare Enterprise Web Server, NetWare Management Portal, and NetWare News Server

Similarly, user applications use user certificates to secure communications with servers and other users. For example, e-mail programs use user certificates to secure e-mail, and browsers can use user certificates for user authentication and user-server mutual authentication. Mutual authentication is a procedure by which browsers can use user certificates to authenticate users to web servers, which in turn authenticate themselves to users’ browsers. This procedure occurs over an encrypted SSL session.

NDS supports the use of user certificates for authentication when you run Novell Modular Authentication Service (NMAS), an add-on product for NDS. NMAS, which will be released soon, enables you to use various methods to authenticate users to NDS. (Look for the upcoming article about NMAS in the February issue of NetWare Connection.) For example, users can log in via a smart card, a biometric reader (such as a fingerprint scanner), or an X.509 v3 certificate such as the certificates Novell Certificate Server 2.0 mints.
Using certificates minted by Novell Certificate Server 2.0, Novell tested SSL mutual authentication with two different configurations:

- Microsoft Internet Information Server (IIS) with Microsoft Internet Explorer 5
- NetWare Enterprise Web Server with Netscape Communicator 4.6 and 4.7

Exchanging secure e-mail is the premier use of user certificates. To use certificates to exchange secure e-mail, users begin by downloading their certificates and private keys into their e-mail client. Novell Certificate Server 2.0 supports several Novell and non-Novell e-mail clients, namely the following:

- GroupWise 5.5 Enhancement Pack client
- Outlook 98
- Outlook 2000
- Outlook Express
- Netscape Messenger 4.6 and 4.7

After users download their certificates into their e-mail client, they configure their e-mail clients to use those certificates to digitally sign e-mail messages or to encrypt e-mail messages using Secure Multipurpose Internet Mail Extensions (S/MIME). (For more information about digital signatures and encryption, see "Background Basics" on p. 8.)

PUTTING SERVER CERTIFICATES TO WORK

To enable server applications, such as LDAP Services for NDS, to use certificates to establish secure SSL sessions, you must complete just four steps:

1. Create server certificates (one or more per server).
2. Export from NDS the Organizational CA’s self-signed certificate.
3. Import into client browsers the Organizational CA’s self-signed certificate.
4. Configure applications to use server certificates.

Step 1: Create Server Certificates

Novell Certificate Server 2.0 enables you to create digital certificates (for an internal or external CA to sign) for any server running the following:

- NetWare 5 with Support Pack 1 or above and either NDS 7 or NDS 8. (To download NetWare 5 Support Pack 3—the most recent Support Pack—visit http://support.novell.com/misc/patlist.htm#nw.)
- Windows NT Server 4 with Service Pack 3 or above and NDS for NT. (To download Windows NT 4.0 Service Pack 6, visit http://www.microsoft.com/ntserver.)

To create a server certificate, you open the ConsoleOne snap-in module for Novell Certificate Server 2.0 and choose the container object that holds the server for which you want to create a certificate. You may then select either the standard or custom option.

If you select the standard option, Novell Certificate Server 2.0 creates the certificate and creates an associated NDS object for you using default parameters. Default parameters specify, among other things, that this certificate will be used for SSL and will be signed by the Organizational CA.

If you select the custom option, you can customize certificate parameters, specifying parameters such as the following:

- The life span of the certificate
- The purpose for which the certificate is created (that is, for encryption,
Staying in or Going Out?

Novell Certificate Server 2.0 enables you to create an internal Certificate Authority (CA) within your company's Novell Directory Services (NDS) tree. In addition, Novell Certificate Server 2.0 supports the use of external CAs, such as VeriSign. When it comes time to choose a CA, how do you decide when to stay in and when to go out?

THE AGE-OLD DILEMMA: COST VERSUS CONVENIENCE

Cost is the most obvious reason for choosing to use an internal CA that you set up using Novell Certificate Server 2.0: Novell Certificate Server 2.0 is provided at no cost, and you can mint an unlimited number of both user and server certificates also at no cost. In contrast, external CAs charge as much as U.S. $100 per user certificate per year and U.S. $400 per server certificate per year (not to mention the CA's server software, for which some CAs charge). Getting essentially the same security for free is an important point to consider.

When you use an internal CA, however, the public-key certificate that others must have to verify the identity of that CA is not published. What does that mean, you may well ask? Most external CAs' public-key certificates are made widely available by having been already installed in most browsers. See for yourself. If you are using Microsoft Internet Explorer 5.0, open Internet Options in the Tools Menu, and click the Content tab, where you'll find a field called Certificates. Click the Certificates button followed by the Trusted Root Certificate Authorities tab, and you will find preinstalled information about nearly 40 CAs.

When you use an internal CA, you have to install the public-key certificate for this CA on the browsers that will use server certificates or user certificates signed by the internal CA. To do so, you must export from NDS the Organizational CA's self-signed certificate and import that certificate into users' browsers. (This step is described in the main article under “Steps 2 and 3: Exporting and Importing the Organizational CA's Self-Signed Certificate” on this page.)

Avoiding this extra step is the primary advantage of using an external CA: When you use an external CA that has already published its public-key certificate in most browsers, servers and users will be able to use certificates issued by those CAs without you having publish that information yourself.

THE BOTTOM LINE

The bottom line is this: You pay for the convenience of an external CA. If convenience takes precedence in your company's environment, go out. For example, if users in your company need certificates to regularly send dozens of people outside of your organization information that should remain private, use the services of an external CA.

If frugality can take precedence for your company, stay in. For example, if you need to set up certificates for server applications that only a handful of remote, internal users will access or if your users need certificates to exchange private information only with other internal users, then, by all means, use the free services of an internal CA.

authentication, signing, or SSL)

- The internal or external CA that will sign this object

If your company's Organizational CA will sign the certificate, the NICI layer on the server generates the public-private key pair for this certificate and returns the public key to your console as a PKCS#10 CSR. Novell Certificate Server 2.0 automatically forwards this CSR to the Organizational CA, which signs the CSR and returns the resulting certificate. Novell Certificate Server 2.0 then installs this certificate in the Server Certificate object (which is also called the Key Material object). This entire process is automated and takes only a few seconds.

On the other hand, if an external CA will sign this certificate, Novell Certificate Server 2.0 returns the CSR in Base-64 code. You then paste this CSR into the form you are using to request a certificate from the external CA of your choice. When the external CA returns the certificate, you paste this certificate into the Server Certificate object.

For example, if you wanted VeriSign to sign this certificate, you would complete VeriSign's online request form for a certificate. During this process, you would cut and paste the CSR that Novell Certificate Server 2.0 generates into the VeriSign web form and then complete the web form.

VeriSign would then send you an e-mail message, indicating that you could pick up your certificate. You would access the web address specified in this e-mail message, download the certificate (in Base-64 code) as instructed, and open the ConsoleOne snap-in module for Novell Certificate Server 2.0. You would then paste this certificate into the Server Certificate object.

Whether signed by an internal or external CA, server certificates along with their public keys and associated private keys, which NICI encrypts, are stored securely in the Server Certificate object. NDS-enabled applications can then access these server certificates directly from NDS.

Novell Certificate Server 2.0 also provides a way for you to export signed server certificates into applications that do not support NDS. Applications that do not support NDS but that use server certificates typically create their own key pair and then formulate a PKCS#10 CSR. You manually cut the CSR from the non-NDS-enabled application and paste this CSR into the ConsoleOne snap-in module for Novell Certificate Server 2.0. You then submit the CSR to the Organizational CA for signing. After the CA signs the CSR and returns the certificate, you cut this certificate out of the ConsoleOne snap-in module and paste this certificate into the requesting application.

Steps 2 and 3: Export and Import the Organizational CA's Self-Signed Certificate

If some of your company's servers use certificates for SSL server authentication, browsers that will be using applications for which you have created server certificates need a way to verify the identity of the CA that issued those certificates. If an external CA signed the server certificate, the CA's public-key certificate is preinstalled on all major browsers, so the browsers already have what they need to verify this CA's identity. (For more information, see “Staying in or Going Out?”)
When an application that you have configured to use a server certificate is initialized, this application finds that its configuration information specifies that it should use a particular server certificate. As a result, the application accesses NDS, retrieves this server certificate, and uses NICI to read in the encrypted private key. Any time an application uses this private key (to encrypt or decrypt information during an SSL session, for example), the process occurs within the trusted computing framework NICI provides.

PUTTING USER CERTIFICATES TO WORK
The steps you take to put user certificates to work vary, depending on the purpose for which those user certificates will be used. However, because securing e-mail messages is the premier use of user certificates, those are the steps defined below.

Assuming you have created an Organizational CA, you can put user certificates to work securing e-mail messages by following these four steps:

1. Create user certificates.
2. Export users’ public key certificates and associated private keys from NDS.
3. Configure e-mail applications by importing users’ public-key certificates and associated private keys into e-mail clients.
4. Exchange public-key certificates.

Step 1: Create User Certificates
Novell Certificate Server 2.0 enables you to create user certificates for any NDS user object. As with server certificates, you create user certificates using the ConsoleOne snap-in module for Novell Certificate Server 2.0.

Using the “Properties of Multiple Objects” feature in ConsoleOne, you highlight the container object that holds the users for whom you want to create certificates and click Create. Incidentally, in case you haven’t checked out a lot of other PKI products, this ability to create user certificates in bulk is a rare feature.

Most other PKI products require you to generate the key pairs for user certificates. The NICI layer running on the server you select will generate the public-private key pairs for the user certificate(s) you plan to create.

Figure 4. After users select a certificate to export and click the Export button, Novell Certificate Server asks whether or not users want to export the private key with the certificate. If users want to exchange secure e-mail, they will answer “Yes.” Novell Certificate Server 2.0 then asks users to enter a password (to protect their private key) before displaying the certificate export parameters for final review.

However, if your company’s Organizational CA has signed the certificate, you must install a self-signed certificate into browsers that will be accessing the application that uses server certificates for SSL authentication. The self-signed certificate is created when you create the Organizational CA. The self-signed certificate is also stored (along with its associated and encrypted private key) within this CA’s NDS object.

The self-signed certificate is considered a trusted root. (Trusted root is a generic term that refers to any certificate used to validate other certificates.) Essentially, the Organizational CA’s self-signed certificate enables browsers to verify the identity of the server minting certificates.

You or users can install the self-signed certificate individually on client browsers that require access to applications using server certificates. You can also use ZEN -works to automatically distribute to workstations a browser with the self-signed certificate preinstalled.

To install the Organizational CA’s self-signed certificate on browsers, you can use the ConsoleOne snap-in module for Novell Certificate Server 2.0 to export the self-signed certificate to a file. You or users can then import that file to the users’ browsers.

Step 4: Configure Applications
You are now ready to configure applications to use server certificates to establish SSL sessions over the Internet. Some applications have ConsoleOne snap-in modules, which make configuring these applications to use certificates as easy as clicking on the appropriate certificate. For example, using the ConsoleOne snap-in module for Novell’s LDAP Services for NDS, you click a server object’s SSL certificate option. A list of all of the certificates you have created for this server appears, and you simply select the one you want this application to use.

When an application that you have configured to use a server certificate is initialized, this application finds that its configuration information specifies that it should use a particular server certificate. As a result, the application accesses NDS, retrieves this server certificate, and uses NICI to read in the encrypted private key. Any time an application uses this private key (to sec-
Steps 2 and 3: Export and Import Users' Certificates

Before a user can use the certificates that are stored in NDS, that user must export these certificates from NDS and import them into his or her e-mail client. To export the user certificates, a user logs in to NDS and launches Novell Certificate Console, the user utility that ships with Novell Certificate Server 2.0. Within this Java-based client utility, the user can view a list of all of the certificates stored within his or her NDS User object. From this list, the user clicks to highlight one of the certificates and then clicks the Export button. (See Figure 3 on p. 14.)

Clicking the Export button opens a wizard that walks the user through the steps required to export a certificate. During this export process, the user specifies certain parameters that determine how the certificate is exported. For example, the user must specify if his or her private keys should be exported. If the user plans to use certificates to encrypt and sign e-mail messages, that user will need his or her private keys. (See Figure 4 on p. 18.)

The wizard also prompts the user to enter a password of at least six characters. This password protects the user's private key as it is being transported across the wire to the user's client. When the wizard sequence is completed, the user's public-key certificate and associated private key and the Organizational CA's self-signed certificate (the trusted root certificate) are sent to the user's client in PKCS#12 format.

The user can now disconnect from NDS to import the PKCS#12 file into his or her e-mail application.

Step 4: Exchange Certificates

The user is now almost ready to exchange secure e-mail. There's only one thing missing. If you think about it, you can figure out the missing piece. If I encrypt a message using my private key, how will you decrypt the message I send? You need my public key, right? Users who plan to exchange secure e-mail must first exchange their public-key certificates, unencrypted but digitally signed to ensure the certificate is not changed in transit. (For more information about digital signatures, see “Background Basics” on p. 8.)

Of course, exchanging unencrypted public-key certificates is not ideal. Recognizing this, Novell is working on a product that will enable e-mail programs to access an LDAP P-compliant directory for the purpose of securely extracting a certificate.

USER CERTIFICATES IN ACTION

Armed with each other's public-key certificates and, of course, their own certificates, users can exchange encrypted, signed, or encrypted and signed e-mail messages.

A TOUGH DECISION (NOT)

Although several commercial PKI vendors offer certificate services, Novell Certificate Server 2.0 offers unique advantages. For example, few—if any—other PKI products enable you to create more than one user certificate at a time as Novell Certificate Server 2.0 does. Also, few if any other PKI products offer a single point of administration in a replicated environment.

In addition, some PKI products generate the public-private key pair for a user certificate in the untrusted, virus-prone realm of the client. Novell Certificate Server 2.0, in contrast, generates key pairs only on the server within its trusted NICI framework.

Finally, most PKI vendors charge for their services. However, Novell Certificate Server 2.0 is free, and the certificates you mint using your company's Organizational CA are also free.

You can opt to use another vendor's PKI product and to pay for each server certificate and each user certificate you use. Or you can use Novell Certificate Server 2.0 and create, sign, and use certificates for free. Talk about your tough decisions.

Linda Kennard works for Niche Associates, an agency that specializes in writing and editing technical documents.