Title: (ilab_blk.eps)
Creator: Adobe Illustrator(R) 8.0
Preview: This EPS picture was not saved with a

How do I get a Certificate stored on my hardware token?

A certificate is the signed-and-locked combination of your identity (your "Subject" information, in certificate-speak) and your public key. The certification authority signs your public key, which makes it useable in a wide variety of applications. So how do you get a certificate? There are four main steps:

1. Generate a public/private key pair.
2. Create a certificate signing request, and send that to a certification authority.
3. Ask the certification authority to sign your request, and provide any identification or authentication information needed.
4. Transfer the certificate back so that you can use it.

In this demonstration, we'll walk you through how you do this with a WWW browser, such as Netscape or Internet Explorer. Instead of storing your certificate in the browser's database, we will show you how to store it on a hardware token, such as an Rainbow Technologies iKey. Because these tools have had embedded PKI support, a lot of the steps above happen behind the scenes where you don't see them.

The hardware token has an encryption engine on-board. What this means to the user is that the private key is generated on the token, and never actually transferred out of the token. All public key encryption and decryption is performed on the token.

Generate a public/private key pair

If you're using a WWW browser, there is no special "make me a key pair" button. Instead, your browser will generate a key pair when a special chunk of HTML is sent to the browser. For example, Netscape Communicator will cause your token to make a key pair if you press the "submit" button generated by a form containing this code fragment: "`<KEYGEN NAME="name" KEYTYPE="type">`".

In the iLabs demonstration, you can go to `https://n1.pki.ilabs.interop.net/` to generate a new public/private key pair. When you request a key pair, your browser may ask you to select between the internal database and the token. Select the token. The hardware token may be password protected, so you may be prompted to enter a password.

Create a Certificate Signing Request (CSR) to send to a CA

The WWW browser-based certificate request is closely tied to the whole Certificate Signing Request (CSR) submission. After your token creates the private key and locks it away, it immediately creates a CSR (in a format called PKCS #10) and submits it to the WWW server named in the form. From there, the WWW server will submit the CSR to the Certification Authority (CA). This is often called *online enrollment.* Most of this is invisible to you. The only way you'll be able to tell that the CSR has been submitted is that the WWW server at the other end will (generally) put up a web page telling you what happened and asking you to check back later.

For example, in the iLabs demonstration, you'll see a screen giving you your request number and a URL to check back with. Why later? Because now you're waiting for the next step to be completed: the manager of the Certification Authority (CA) has to approve your request and issue your certificate.

Ask the CA to sign your request.

Your CA is not just a software package; it is a hardware system and a set of policies and procedures for how certificates are issued and managed. In some organizations, certificates will automatically be issued as soon as you make a request. For example, you might have your user information preloaded into an LDAP directory and as soon as a CSR arrives (typically accompanied by a password), the CA software can automatically approve it.

Some organizations will run their own CAs. Others will use one of the many public CAs available. Using a public CA is an excellent way to get started with PKI technologies, and may be a good long term alternative to running your own CA.

In our demonstration, however, we want you to see the process of approving a certificate. So you'll have to go to the CA you requested your certificate from and approve/issue the certificate. Each CA has a different syntax and rules, but the effect is the same: the CA will create a certificate and sign it. In the iLabs demonstration, ask one of the team members to help you approve and issue your certificate.

Although your WWW browser has suggested all the fields to go in the certificate, the CA may change the certificate to meet the security policy of the organization. For example, the CA administrator may add the CRL (Certificate Revocation List) Distribution Point to the certificate to indicate where the CRL for this certificate would be stored (if the certificate had to be revoked).

When the CA has signed the certificate, it may install it in a couple of places (such as an LDAP or WWW server), in addition to within the CA database itself.

Transfer the certificate back so that you can use it.

To use the newly-signed certificate, you have to get a copy and keep it near your private key. Since the certificate is (essentially) public information, it can be stored anywhere and retrieved via LDAP or http. In the most elegant version of certificate retrieval, a WWW server will deliver the certificate using a particular MIME type (such as `application/x-x509-user-cert`). The certificate will be wrapped up by the server in a format called PKCS #7 and automatically loaded into your hardware token.

In the iLabs demonstration, once your certificate has been signed by the CA, you can go back to `https://n1.pki.ilabs.interop.net/` to retrieve it.

In some applications, the certificate isn't automatically downloaded. In that case, usually you use a "file-based" storage (saving the PKCS #7 format to a file and then importing the file) or, often, simply cutting-and-pasting the certificate from one window to another.

In the iLabs demonstration, your certificate will also be displayed at the bottom of the page in a variety of formats so you could use cut-and-paste if the particular application required it.