InteropNet Labs

What is a Certificate?

A certificate can have many forms, but at the most basic level, a certificate is an **identity** combined with a **public key**, signed by a **certification authority.** Once you have a certificate, you can then use it in a number of different ways. The most common operations are **authentication** and **encryption**. Both encryption and authentication depend on the concept of **public/private key cryptosystems.** If you understand these six things (along with what it means to sign), you'll know 80% of what there is to know about PKI (public key infrastructure).

Subject Identity, Public Keys, and Certification Authorities

The **identity** in a certificate (also called the "Subject" of the certificate) says who (or what) the certificate was issued to. It can have a variety of syntaxes, but usually contains a *distinguished name* with attributes like your "common name" (CN), "organization" (O), and "organizational unit" (OU) such as "CN=Joel Snyder, OU=users, O=pki.ilabs.interop.net." Often, the subject of a certificate has additional entries, such as an email address (jms@opus1.com) or a fully-qualified domain name (w1.pki.ilabs.interop.net).

The **public key** is part of a private/public key pair, generally one created for use with the RSA public key cryptosystem. The private key is not part of the certificate and need only be stored; never transferred or viewed (including by the certification authority); in fact, good security practices suggest that the private key never leave a protected key store (such as a smart card). The relationship between the subject of the certificate and the public/private keys is very direct: the keys belong to the subject. Anyone wishing to communicate with the subject of the certificate can use the public key either as part of an authentication or encryption algorithm.

A **certification authority** is a trusted third party which "signs" certificates. When a certificate has been signed, it gains some cryptographic properties. It cannot be tampered with, because the signature algorithm makes such tampering impossible without detection. This means that the identity and the public key are inalterable.

In addition, the certification authority signature guarantees the real identity of the subject of the certificate. For example, a certification authority (or CA) may require the subject to present some picture identification before the certification authority will sign the certificate.

Once the **certification authority** has signed the certificate, which seals the certificate against tampering, anyone can use the **public key** to either **authenticate** the subject's **identity** or to **encrypt** data sent to the subject.

A Sample Certificate

We have included a sample certificate below, slightly edited to shorten it. Notice the "`Issuer`" field (the certification authority issuing the certificate), the "`Subject`" field (the identity of the holder of the certificate), the "`Public Key`" field (the public key of the subject), and the "`Signature`" field (the signature of the CA attesting to the identity of the subject).

```
Certificate:
    Data:
     Version:  v3
     Serial Number: 0x7
     Signature Algorithm: MD5withRSA - 1.2.840.113549.1.1.4
     Issuer: CN=Certificate Manager,OU=iPlanet,O=pki.ilabs.interop.net
         Validity:
             Not Before: Saturday, April 14, 2001 6:47:23 PM PST
             Not  After: Sunday, April 14, 2002 6:47:23 PM PM PST
     Subject: CN=Joel Snyder,OU=users,O=pki.ilabs.interop.net
     Subject Public Key Info:
        Algorithm: RSA - 1.2.840.113549.1.1.1
        Public Key:
             Exponent: 65537
             Public Key Modulus: (1024 bits) :
                 <long ugly hex string deleted>

     Extensions:
         Identifier: Key Usage: - 2.5.29.15
             Critical: yes
             Key Usage:
                 Digital Signature
                 Non Repudiation
                 Key Encipherment
         Identifier: Subject Alternative Name - 2.5.29.17
                    Critical: no
                    Value:
                        [RFC822Name: jms@opus1.com]
 Signature:
     Algorithm: MD5withRSA - 1.2.840.113549.1.1.4
     Signature:
         <long ugly text string deleted>
```

Authentication and Encryption

Certificates can be used for many different things, but one of the main reasons to use a certificate is to **authenticate** the holder of a certificate. By using public/private key cryptography and one of a number of algorithms, it is possible to "prove" your identity. The details are out of the scope of this white paper,[1] but please see our "References and Resources" white paper if you want to learn more.

Certificates can also be used for **encryption**, although they are rarely used for encrypting more than a few bytes of data.

[1] As an example, if I knew your public key (because I had read it out of a certificate), I could send you a random number and challenge you to encrypt it with your private key. Using your public key (which I took from your certificate), I could decrypt the encrypted challenge. If the assumption that *only* you have your private key is true, then you have authenticated yourself to me. (There are additional details, of course, but this is the gist of the way it is done). Only you could have made an encryption of the challenge using your private key, which then decrypts with your public key.