realtimepublishers.com™

The Shortcut Guide™ To

Managing Certificate Lifecycles

Kevin Behr

Copyright Statement

[**Editor's Note:** This eBook was downloaded from Realtime Nexus—The Digital Library. All leading technology guides from Realtimepublishers can be found at http://nexus.realtimepublishers.com.]

Chapter 4: Managing PKI Infrastructures

The previous chapters have discussed the components used in a Public Key Infrastructure (PKI). This chapter will address the key decisions that IT professionals need to make in planning and implementing the infrastructure that will best support their security needs—while addressing the demands of their budgets. The chapter will focus on

- Defining the requirements and challenges of the infrastructure
- Identifying the key components of the infrastructure
- Choosing the best means to provide the infrastructure—specifically, comparing outsourcing with building an in-house system

The Requirements for an Effective PKI

Why does an organization need a PKI? Although this topic has been examined in previous chapters, a brief review will help set the stage for the decisions you need to make in designing and building your own infrastructure.

PKIs help people secure information. Specifically, PKIs enable the exchange of information between entities in electronic form. There are a variety of specific functions for which PKIs are used.

One use is to encode a data stream. This could be all data exchanged on a TCP channel, such as used by the secure hypertext transfer protocol (HTTPS) or secure File Transfer Protocol (SFTP). It might be encoding the text within the message, transmitting other portions of the message in plain text so that other components, such as mail servers, can effectively deliver the encrypted payload without being aware of the encryption that protects the sensitive data. By encrypting the data, it can be sent securely through public channels such as the Internet. This enables electronic commerce and exchange of confidential information to be extended to nearly everyone, displacing the expense and added maintenance of private networks and Value-Added Networks.

Encryption keys can be used to secure data stored in databases and on file systems. These keys protect long-term assets and should be stored securely. If the keys are lost, the encrypted data becomes unusable. If the keys are compromised, the data is no longer secure. A PKI can implement technology and procedures that secure the keys that protect sensitive organizational data.

Another use is to confirm the identity of the sender. This is the process of using digital signatures. The data itself may not contain data that is sensitive. If electronic networks are to be used to conduct legal commerce, some form of non-repudiation of the parties involved in a transaction is required. If the parties met, they would sign paper documents, and their signatures would stand as evidence of their willingness to participation in the transaction. Digital certificates can be used as legally binding digital signatures.

More frequently, the need of identity confirmation is much more mundane. For instance, if remote access to the corporate network is required, the virtual private network (VPN) server might require a digital signature to validate the user. A smart card can contain such a signature and improve the security of VPN access.

PKI Requirements

These needs help define four key requirements for a PKI:

- The PKI must be able to deliver a public key that the client can use to encrypt data. The system must be able to validate the authenticity of this key.
- The system must be able to authenticate the validity of a key, so authorized administrators must be able to revoke a key. Keys need be revoked for a variety of reasons: the private key is compromised, the issuing entity is retired or no longer valid, or a new encryption algorithm or key size is implemented. Any of these reasons may require an existing certificate to be revoked.

> ✎ For the purposes of this discussion, a certificate is a digital document that contains a public key and metadata concerning the issuer and the Certification Authority (CA) that validates it (for instance, an X.509 certificate).

- If the system is to count on the certificate to provide proof of identity, it must provide non-repudiation. If a digital certificate is provided to a consumer, the PKI must reasonably guarantee that it was issued by the certificate holder. Thus, similar to a written signature on a paper document, it can serve as evidence that the communication was sent from the certificate's owner.
- For the PKI to be utilized by automated systems, there must be a centralized means of defining and enforcing policy within the system. For instance, defining encryption algorithms, enforcing key lengths, revoking keys issued by terminated employees, expiring keys after a defined period of time, and so on.

The Threats to Your PKI

To implement this type of system, there are two key attacks that must be defended against. First, the private key must be jealously guarded. With the private key, any encryption provided by the system is laid bare. Further, if the key is lost, all data encrypted using the public version of the key is rendered virtually useless. Thus, the key must be backed up securely.

The second threat comes from the "man-in-the-middle" attack. If a malefactor can intercept a legitimate certificate and substitute his or her own certificate in its place, the attacker now controls the private key that can decrypt the payload. The attacker can substitute his or herself as the legitimate recipient of the encrypted documents. To protect against this attack, the PKI must provide a mechanism to identify the source of the certificate.

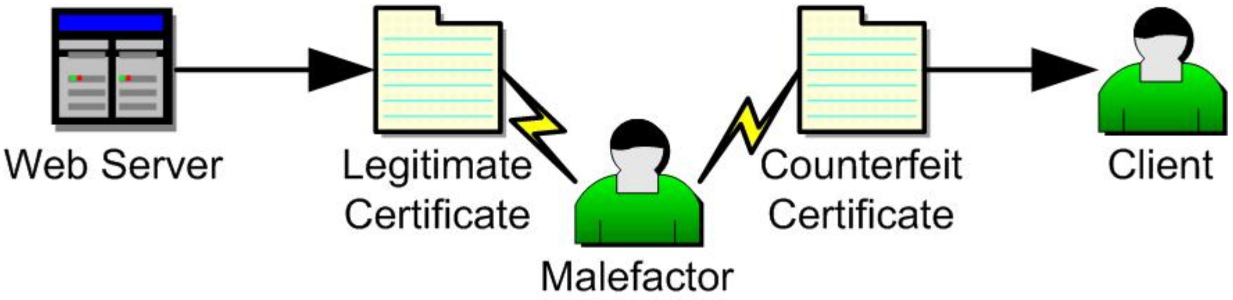

Figure 4.1: A man-in-the-middle attack.

Building a Network of Trust

Central to any PKI is developing a mechanism by which the legitimacy of a key can be confirmed. The simplest means is by direct trust. If a person or entity you know or can independently verify provides you with a public key, you trust that key directly. This could occur when a person hands you a CD with a public key file on it, or your employer provides you with a smart card.

The question becomes, how well does this system of direct trust meet the requirements of your need for a PKI? When you have independently authenticated the source of the key, that requirement is met. Of course, in the anonymity of the Internet, this could be much more difficult to confirm. Certificate revocation can also be a problem. The issuer can ask you to stop using the key, but they are wholly dependent on your compliance. If you issue keys with direct trust, and need to upgrade, changing the encryption algorithm or key length, you must contact each person who has a key and get them to exchange them. Non-repudiation also becomes more of an issue. As the issuer may not have tight control over the distribution system or control of the keys passed to clients, they may not stand by the issuance of the key. Also, policy becomes manual and cumbersome.

With all these disadvantages, one might think this system is seldom used. But, with a limited number of users and the right circumstances, it can be appropriate. For instance, sharing keys between the servers of business partners to share sensitive information is often accomplished through direct trust.

The Role of Certification Authorities

To better automate and manage the system of distributing and managing keys, a more formalized system has been developed. In this system, there is a centralized source—a Certification Authority (CA)—whose primary function is to validate certificates. The CA has its own signature that can be independently validated. When a user creates a certificate with his or her private key, the CA validates the public key and adds the CA's digital signature to the certificate. Thus, the CA stands as the validator of the certificate.

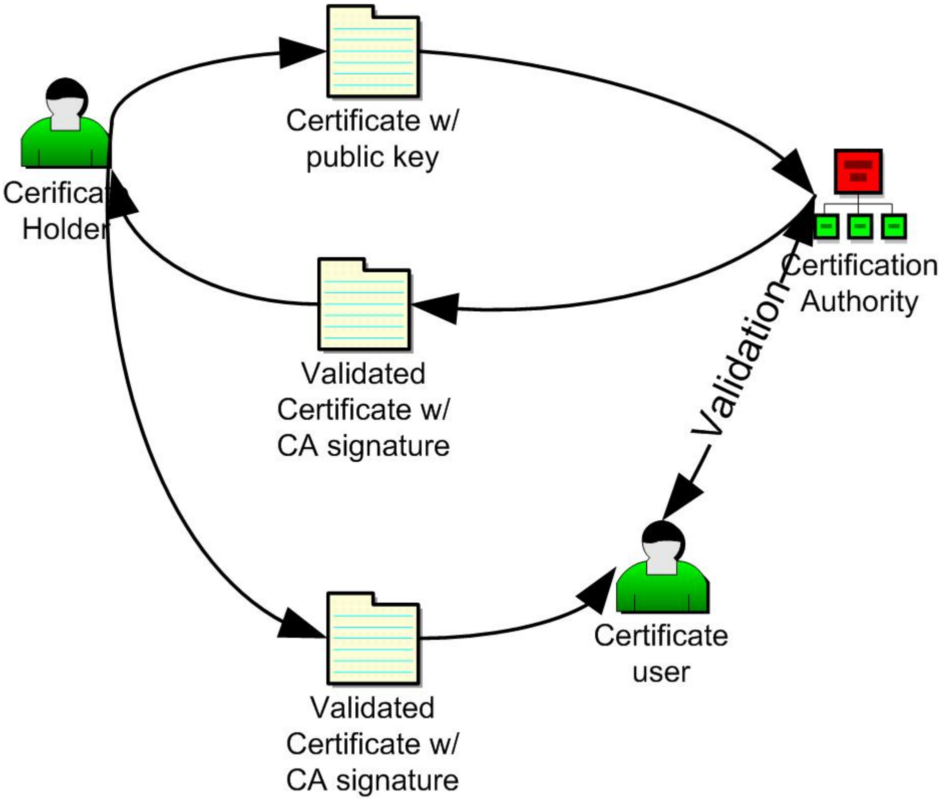

Figure 4.2: CAs validate certificates.

The CA acts as a trusted agency, so its function is typically fulfilled by a third-party organization. This setup promotes trust, particularly between independent organizations with divergent goals. A third-party CA best serves by remaining a neutral source, trusted by all parties.

The test of the CA system then becomes whether they can meet the requirements for your PKI. First, the CA can authenticate the certificates they sign. The CA signature on the certificate can contain a hash that can help determine any alterations to the certificate.

Certificate revocation is also readily handled. The certificate holder can inform the CA that the certificate is no longer in service. This can be an automatic expiration or an overt change or retirement of the certificate. When a user goes to validate the certificate, the user can be notified by the CA of the change in the status.

The certificate should stand as a non-repudiatable identification of the certificate holder, similar to a signature. The CA validates the identity of that holder when they sign the certificate. The holder then cannot deny that it is his or her certificate.

The matter of control is also addressed. Because use of the certificate can be validated with the CA and the status can be controlled by the holder, working with the CA, the policies relating to how the certificate is used are more easily managed.

CA Systems

It would be simple if there were but one CA. Everyone would register their keys with that one authority. The CA could guarantee uniqueness of names and easily track how certificates are used and the policy that is applied to them. They would have a common signature and a simplified system of authentication.

In practice, there are many competing CAs. Each one offers differing levels of security and service, as mentioned in the previous chapters. Such is the situation on the World Wide Web. This model is often referred to as the web of trust. In the web of trust, a user can receive certificates from a number of CAs. The user may also directly trust a certificate issued from another user who is known to the user, such as a business colleague. This is really an outgrowth of the direct trust system. It is used particularly within an organization.

The major drawback is that policy is difficult to enforce. If a person has certificates with multiple vendors that do not report with one another, that person must carefully manage the certificates they have with those vendors. Also, much of the burden of identifying the legitimacy of the certificate source falls onto the user.

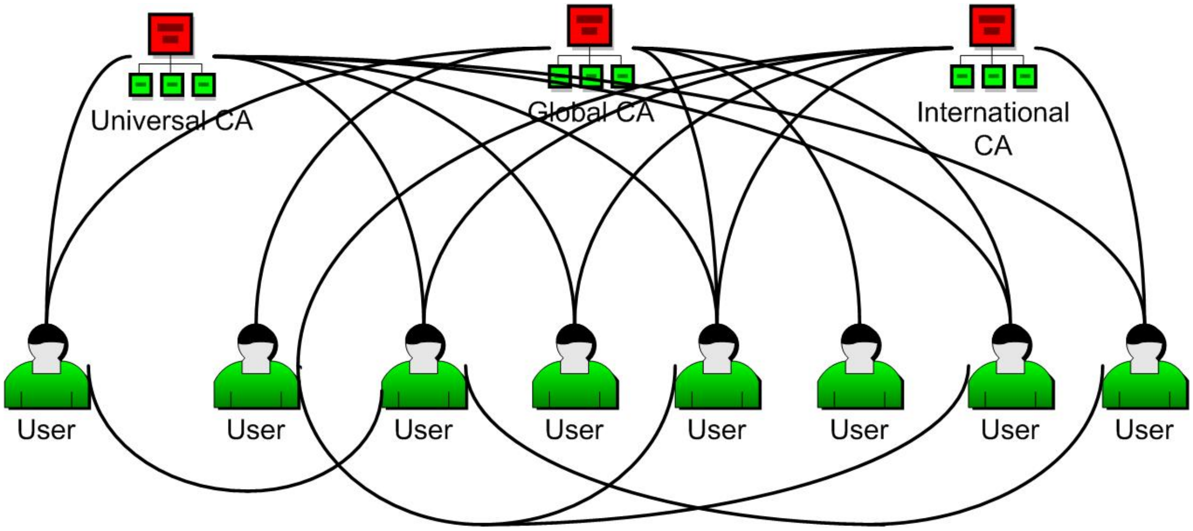

Figure 4.3: Web of trust.

It can be advantageous for a CA to create subordinate authorities. In this case, a root CA holds the credentials to validate multiple subordinate CAs. Each of the subordinate CAs can issue and validate individual certificates, and the subordinate CA can be validated against the root CA. For instance, an organization might contain subsidiaries. Each subsidiary might need to create and manage its own CA. The root CA for the organization can validate the certificates of the subordinate CAs and stand guard against someone creating a counterfeit CA within the hierarchy.

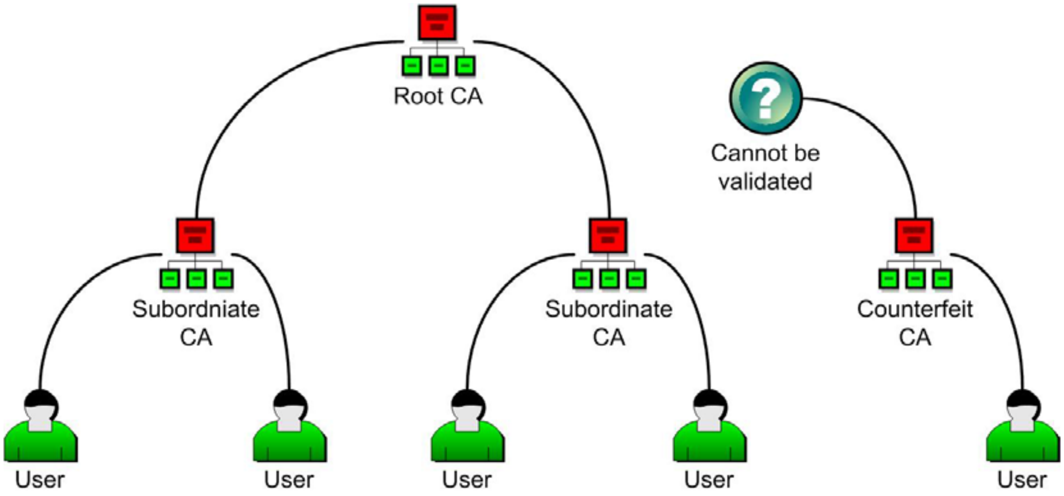

Figure 4.4: CA hierarchy.

Digital Certificates

Digital certificates are used to help standardize and automate the tasks of exchanging these public keys and authenticating signatures. A digital certificate contains a public encryption key and metadata that is used to identify, validate, and control use of the key. For instance, a typical certificate might contain:

- Owner's name
- Name of the CA that validates the certificate
- The public key used by the CA
- Owner's public key
- Period for which the key is valid
- Digital signature that can validate the key

There are several standards for controlling how certificates are constructed and authenticated:

- X.509 is the oldest and most commonly used standard in commercial applications. It provides for direct authentication from a single CA, web of trust, and hierarchical systems of trust. To accommodate these varieties, the certificate itself is extensible and more complex to use.
- OpenPGP (Pretty Good Privacy) was adopted from encryption software originally targeted for encrypting emails. It is lighter weight than the X.509 standard. It relies on direct authentication or the web of trust.
- Simple Public Key Infrastructure (SPKI) was developed to greatly simplify certificates for which only a single CA is required. These are ideal for use in intranet applications where validation of the source needs to be less rigorous. The infrastructure used to implement SPKI is referred to as Simple Distributed Security Infrastructure (SDSI). It is based on the work of Ron Rivest and is being developed by the Internet Engineering Task Force (IETF).

As has been previously mentioned, the X.509 standard is the most common standard for exchanging public keys. This standard is maintained by the IETF. They created a working group for maintaining this standard called PKIX. The charter for this group can be found on the Internet at http://www.ietf.org/html.charters/pkix-charter.html.

Elements in a PKI

As PKIs have grown, there are several common services and structures that have grown to meet their needs. Understanding these key elements can help one plan and administrate a PKI.

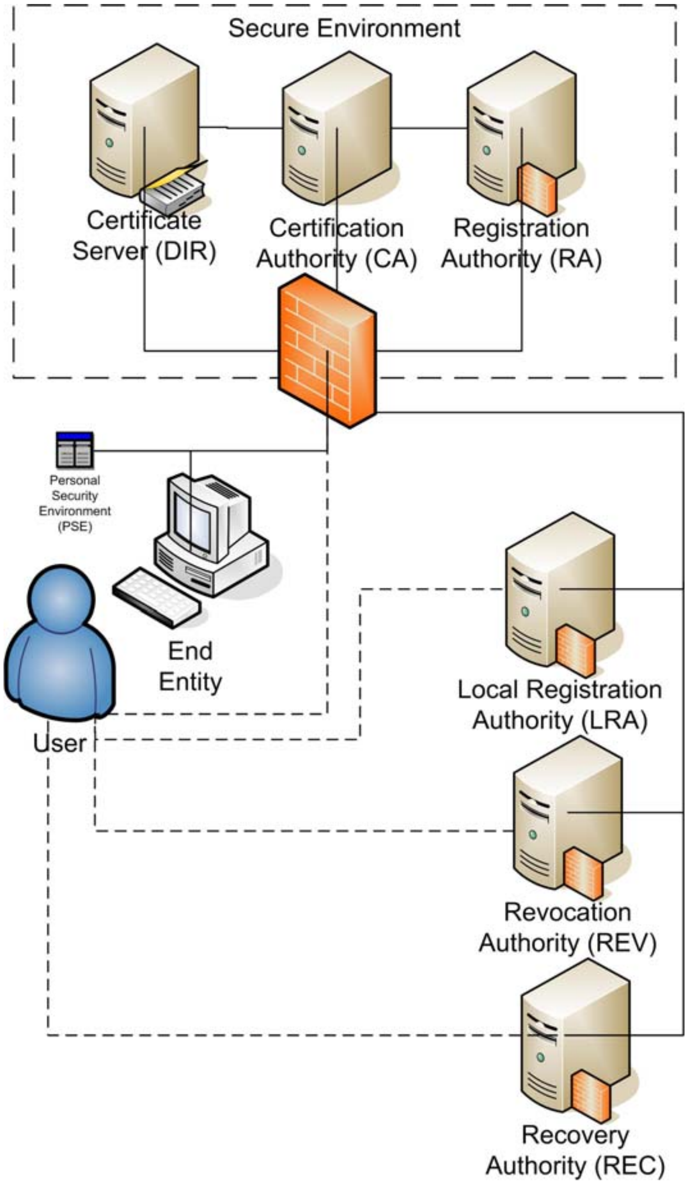
Secure Environment
Certificate Server (DIR)
Certification Authority (CA)
Registration Authority (RA)
Personal Security Environment (PSE)
End Entity
User
Local Registration Authority (LRA)
Revocation Authority (REV)
Recovery Authority (REC)

Figure 4.5: Components of a PKI.

CA

The CA component is responsible for generating digital certificates. A computer that wants to receive encrypted data generates a public key and submits that key to the CA computer, along with other required metadata, such as the duration of the certificate and the distinguished name of the owner. The CA computer compiles this information, adds it own public key, and signs the freshly minted digital certificate.

The CA system must be closely guarded. If its private key were lost, it could no longer create or validate new digital certificates. If the private key were compromised, a malefactor could forge certificates and violate the primary trust provided by the CA system.

Because of the need to protect this computer, it is typically kept in a high-security environment. This would include physical security, such as securely locked rooms, as well as regulations that enforce at least two people being present when the computer is directly accessed. The CA computer typically is not connected to the Internet directly.

Registration Authority

The registration authority (RA) allows users to apply for digital certificates. It collects the public key and metadata related to the certificate and typically handles corresponding with the applicant. Most RAs are public facing and require Internet access. Care must be exercised in the communications between the RA and the CA so that the CA security is never compromised. An RA may be a physical server or just an administrative service application.

RAs may be distributed outside the control of the CA. For instance, if an organization creates a PKI for their intranet, they may choose to distribute RAs in each satellite location. These distributed servers are referred to as local RAs.

Certificate Server

When the certificates are created, they need to be stored for future retrieval. This storage must be secure. The certificate server (or Directory—DIR) is the component that stores the certificates in a secure manner for future retrieval. Certificate servers are also used in support of the certificate revocation process.

Time Stamping Service

The time stamping service (TSS) is an optional but significant service. In order to provide non-repudiation, transaction must secure the time at which the certificate was used. Because the certificate holder maintains their policy, including the period of validity of the certificate, within the CA system, having a secure time stamp with the certificate helps to validate that the certificate holder was honoring the certificate at the time of use. If the trusted authority provides the time stamp (typically as a value-added service), it serves to indicate that the certificate was valid when in use.

Revocation Authority

There are times when a key must be revoked. This can be done as part of the RA or can be a service in and of itself. The revocation authority (REV) informs the CA that the certificate is no longer valid. The CA will then add the certificate to the revocation list and make the necessary changes with the DIR service so that the certificate is no longer issued.

Recovery Authority

If the PKI centralizes the secure storage of keys, the recovery authority (REC) provides a mechanism to obtain a copy of the key. Not all PKIs centralize the storage of keys. It is common for this service to be supplied in conjunction with one of the other aforementioned services, such as the RA or REV.

Personal Security Environment

The certificate holder needs a place to store their digital certificates, public keys, and private keys. This is known as the personal security environment (PSE). All the components that handle keys and certificates within the system also require a PSE for their secure information. A PSE can store data on a hardware security module, smart cards, or other security device. Typically, however, it is stored as a file on a computer. It is important to ensure that this information is well protected. As previously stated, loss of a private key invalidates the data encrypted with the accompanying public key. If the private key falls into nefarious hands, data encrypted with the public key is no longer secure and the trust is violated.

End Entities

Finally, the key contained in the certificate must be used to encrypt data. The PKI software applications that consume the digital certificate, find the public key, and use it to establish a secure connection with the server are termed end entities.

Certificate Management

Certificate management is the crucial function served by your PKI. The IETF provides guidelines for managing keys in RFC 2510 Certificate Management Protocols. It structures a number of critical elements that must be addressed by the PKI that bear consideration.

Initialization

Several elements of your PKI must be initialized before the system can be put to work. First, the CA must be initialized. A private key must be generated under controlled circumstances so that the key is secured and not copied or stolen. The process must be monitored by at least two administrators to ensure the integrity of the procedure.

Certificates must be created. Upon initialization of a new CA, the process begins as new requests are made. Over time, certificates will be revoked and replaced. The CA must store the new certificates within the DIR and prepare the publication and revocation lists used to validate the certificates.

The certificates will ultimately need to be exported to end entities so that they can use the public key therein contained to encrypt data. The CMP provides definition of the protocols used to regulate this process.

Publication of Certificates and Revocation Lists

Although the CA creates and manages the certificates, they are stored in the DIR. The CA must provide the DIR with copies of new certificates to store. If a certificate is revoked, the CA adds it to the revocation list, which is also stored by the DIR. The CMP lists several protocols for implementing this process.

Key Recovery

The CA can be used to create the private key as well as the public key used with a certificate. The advantage of this system is centralized management of the key pair. It requires the CA to maintain a secure database where the keys can be kept. If data is stored in an encrypted manner, recovery of the key can be crucial to restoration of the data. But there is a strong caveat to consider: If the database is compromised, all the data secured with the keys stored therein is at risk. The benefits should be carefully considered against the risks. If your organization chooses to centrally manage the key pairs, consider the following:

- The database is effectively an extended PSE. It must be kept in a secure environment and carefully guarded.
- There must be a clear, carefully followed protocol for recovering keys.
- Only keys used to store data that is kept in long-term storage needs to be secured. Keys used to temporarily encrypt emails or to provide digital signatures do not need this level of security.

Revocation

Once the certificates are in circulation, there are a variety of reasons that the may need to be revoked:

- If the private key is lost or compromised, the certificate should no longer be used to encrypt data. Either the data will not be decrypted (loss of private key) or it will not be secure (theft of private key).
- If the private key of a CA is compromised, the malefactor can coin counterfeit certificates. In this unlikely but devastating event, all the certificates managed and validated by the CA must be revoked and replaced with new certificates that are based on a new private key.
- If the metadata on a certificate changes, the hash values on the certificate change and thus the old certificates must be revoked and replaced with updated certificates. For instance, X.509 certificates contain the distinguished name of the server in which they are installed. If the distinguished name of the server is changed, the certificate must be updated. Doing so will result in the invalidation of the old certificates.
- If an organization needs to suspend a certificate for a short period of time—for example, to combat a Denial of Service (DoS) attack—it can be temporarily revoked and later reinstated.
- If a server or entity is retired or is no longer associated with the originating entity, his/her/its certificates are revoked. If an employee has a certificate for VPN access, that certificate is revoked when the employee terminates employment. If a server is replaced through a server consolidation, its certificates are no longer used and should be revoked.
- For security purposes, certificates come with expiration. Once they expire, they are revoked and, typically, a replacement certificate is made available.

Certificate revocation is performed by the CA. The CA maintains a revocation list that is stored in the DIR. If the certificate holder drives the revocation—because private keys have been lost or compromised—the certificate metadata changes or the certificate is to be retired; the holder must in form the CA. This can be done via the REV service.

Certificate Issuance

When an entity requests that a CA issue a certificate, the process is known as *enrollment*. The enrollment process is governed by the guideline established in the CMP. There are a variety of ways to implement enrollment.

An entity can generate a private and public key pair on his/her computer. The private key is secured in the PSE. The public key is packaged in some portable form, such as a CD, floppy disk, or USB key. It is transported in this form to the RA as the entity applies for the certificate. Validation of the entity is part and parcel with the application process and must adhere to the CA's Certification Practice Statement (CPS) for validating the user. The RA collects the rest of the requisite data (name of the certificate holder, expiration date, and so on), and submits the data to the CA. This process is governed by standards, such as PKCS#10 (see RFC 2511 for details). The CA creates a properly formatted certificate. A copy of the certificate is placed in the DIR for safekeeping and a copy is given back to the RA. The RA issues the copy of the certificate back to the requesting entity.

As most people do not venture in person to their CA, the process is often performed online. Similar to the previous process, the applicant fills out an application online and submits their public key (the generation of key pair and the process of submitting the key were detailed in previous chapters) to the RA. The RA uses the process outlined in its CPS to validate the applicant, then submits the key and metadata to the CA. A digital certificate is created, stored in the DIR, and returned through the RA to the applicant.

If the key pairs are centrally managed by the CA, the applicant begins by filling out an application. Once the application is validated, the RA forwards the request to the CA, which generates both the private and public key. The private key is secured and copied to a PSE (often in the form of a smart card) and a digital certificate is created. Both the PSE and digital certificate are returned to the applicant by the RA.

✎ A very common use of certificates is to encrypt data shared by network routers. Cisco Systems, a leading manufacturer of network routing equipment, has developed a protocol for enrolling and changing certificates, dubbed the Certificate Enrollment Protocol. CEP provides for decentralized key generation and online installation. This protocol is not compliant with the CMP protocols established by the IETF.

Certificate Servers

Although the CA creates and manages the certificates, the certificate server stores those certificates and makes them available. Because of the central role that the DIR component plays, it deserves special attention.

The job of the certificate server is to provide access to certificates and revocation lists, so it serves much the same function as a directory server. Directory servers provide information concerning the entities within an organization. They can provide authentication information for security, email address lists, and other information. Because this function is very similar to that required of the DIR service, certificate services are often combined with the corporate directory services.

✎ The PKI does not actually require or specify an organizational directory service. As this service is commonly used, many directory servers include certificate services as part of their functionality.

Directory Services

A directory service provides a database that can be accessed to identify the entities contained within an organization. Each entity gets an entry in the directory service database. The entity will have a set of attributes, such as its type (user, computer, printer, router, and so on), its name, and its network address. Each type of object in the directory will have a defined set of attributes. The definition of these attributes is called a *schema*.

To manage the objects, they are typically organized into a hierarchical structure commonly referred to as a Directory Information Tree. The DIT consists of container objects and leaf objects. Container objects are used to create departments, groups, or other organizational units (OU). The leaf objects represent actual entities, such as users, computers, routers, and so on. To help define objects within the tree and facilitate navigating to those objects, a namespace can be defined. The namespace holds the specification for the common objects that the DIT contains.

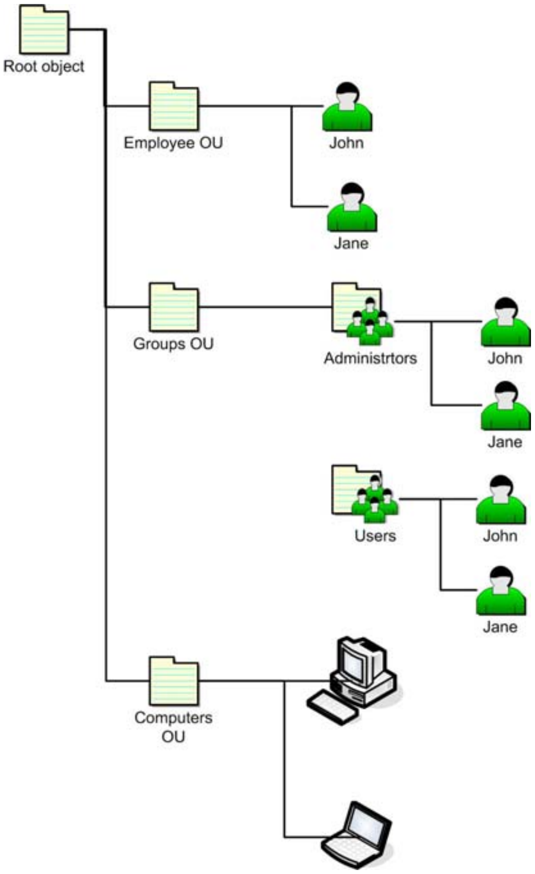

Figure 4.6: Example of a DIT.

X.500 Directory Specification

The ITU-T published a specification for directory services called X.500. It is a complex standard designed to cover a very wide range of uses. It has been organized into nine subsets of the standard. X.509, discussed ubiquitously through this guide, is one of those sub-standards. You can discover more about these standards at http://www.itu.int/home/index.html.

One of the key concepts is to develop a globally unique name of an object. By structuring the namespace correctly, this goal can be achieved. As one of the goals of the PKI is to identify a certificate holder, providing a unique name for the certificate holder is important. To provide a unique name so that a specific entity can be positively identified, the X.500 specification presents the concept of a DN.

A DN consists of a name that is broken down into components. The X.500 specification identifies five components that can be used to comprise a DN:

- Country (C)—The country in which the name is initiated
- Locality (L)—A localization, such as a city, region, and/or state to which the entity is affiliated
- Organization (O)—The organization within that locality to which the entity is affiliated
- OU—A group used to organize entities within the organization, such as Employees or Servers
- Common Name (CN)—A name for the entity that is unique within the OU in which it resides

Thus, if John Doe is in the Engineering department of MyOrganzation located in Anytown, USA, his DN within the directory could be expressed as

```
C=USA, L=Anytown, O=MyOrganization, OU=Engineering, CN=John Doe
```

> ✎ This is a navigational path. John Doe could easily be a member of an additional OU, such as Network Administrators. Thus, although there is only one John Doe, he can have many DNs that correctly identify him.

The X.500 specification has been implemented as several directory protocols. The following list identifies a few of the most common.

- Directory Access Protocol (DAP) is a very full-featured protocol. It allows the DIT to be navigated and objects within the tree to be modified. It can also include security protocols. Like many full-featured protocols, it is difficult to implement and use. DAP servers are often extended to build X.509 certificates and distribute them as required.
- Lightweight Directory Access Protocol (LDAP) was developed by the IETF to provide a directory protocol that was focused on the needs of Internet users. It is described in RFC 2251 through RFC 2267. The protocol is a simplified version of the full-scale DAP. LDAP has been implemented by several vendors to act as a directory service for their TCP/IP-based networks, providing authentication, location, and security services. LDAP servers are among the most popular of directory servers and are very frequently used to distribute certificates and post revocation lists.

When accessing a specific RFC by number, you can visit http://www.ieft.org/rfc/rfcXXXX.txt where XXXX is the number of the RFC. Thus, you can access RFC 2251 by browsing to http://www.ieft.org/rfc/rfc2251.txt.

- Domain Name System (DNS) is implemented as the standard means of mapping common text-based names to IP addresses. In the public Internet, it is used to ensure the uniqueness of common Internet names through the services of domain registrars. Because DNS is carefully monitored to prevent duplicate names, it becomes a standard source for developing a truly globally unique name space in which entities can be defined. Most TCP/IP networks implement DNS servers to resolve names within the network, so they are a very common network appliance.

  The IETF proposed a service extension to DNS that allows it to be used as a certificate server, termed DNSSec (see RFC 2532 and RFC 2931). The protocol allows a digital certificate to be distributed within a DNS message. This concept was not considered during the formation of the X.509 protocol specifications, so the process itself is not compatible with those protocols. This incompatibility has resulted in sparse adoption of the specification.
- Active Directory (AD) is Microsoft's proprietary directory service. It provides an LDAP-compatible interface and similar functionality to an LDAP service. Microsoft provides an LDAP-compatible interface to its directory to allow LDAP clients to access its information store. The primary limitation is that it requires the use of Microsoft Windows servers to implement and administrate. Microsoft Windows servers include a certificate service as part of their standard offering.
- Netware Directory Service (NDS) is the proprietary directory service developed by Novell. It provides similar functionality to that offered by AD. In conjunction with its implementation on Linux and open source platforms, Novell now offers eDirectory as an LDAP directory implementation.

Why All the Attention to Directory Services?

The X.500 specification is the basis for building digital certificates. The X.509 specification has always included a DN field used to identify the certificate holder. And because the primary functions of a DIR service are quite similar to those of a directory server, many directory server implementations include a DIR service as part of their standard offering.

Certificate Servers and Revocation Lists

One of the key functions of a PKI is management of a list of revoked certificates. The certificate server is an obvious choice to fulfill this function. When a certificate is revoked, the CA notifies the DIR of the revocation and the certificate is added to the revocation list. Each certificate has a serial number that serves as the unique identifier of the certificate. When asked to validate a certificate, the certificate server can quickly search the list. If it is located, a revocation notification can be issued. Accessing the list can be done in a variety of ways.

Checking Revocation Status Online

If the end entity can connect to the certificate server online, it can query the server to discover whether the certificate has been revoked. The certificate server searches the revocation list. It compiles a response, signs it, and returns it to the requestor.

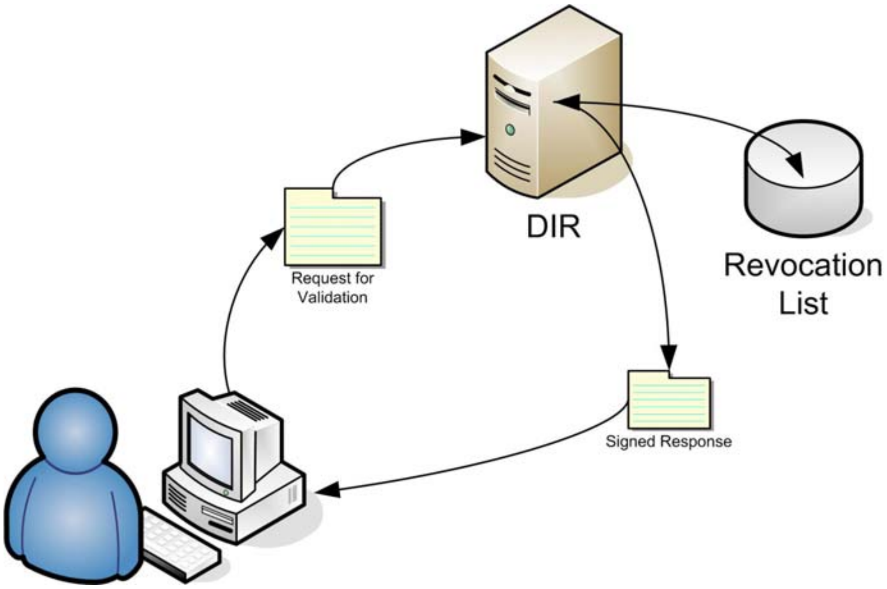

Figure 4.7: Online certificate status checking.

Although this system is conceptually simple, it includes costs that should be carefully considered. The user must have an available connection to the certificate server. As long as the certificate server is local and available, this is likely to be the case. But if the user is occasionally connected, or the DIR is separated through the Internet or a WAN line, this can be costly. Also, the activities performed by the DIR server to look up the specific certificate and then compose and sign a message consume server resources. During times of peak use, this can impact performance or even dictate the addition of servers. The chief advantage is that the status of the certificate can be ascertained with up-to-the-minute results.

The IETF developed the Online Certificate Status Protocol (OCSP) to facilitate online validation (see RFC 2560). When the server receives the request, it ascertains the status of the certificate, the reason for revocation (if any), and compiles and signs a response. The server does not give any information about itself beyond the signature. This led to the development of a more expansive protocol that provided more information.

The revised protocol is dubbed, ironically, the Simple Certificate Validation Protocol. The expanded protocol provides the following features.

- The end entity can submit the request to a server that queries the CA that issued the certificate. It can then follow the hierarchy chain to the root CA to determine whether the CA can be trusted. It can then return the result to the end entity.
- If the end entity does not trust the server, it can request the CA certificates be returned so that the end entity can validate the trustworthiness of the certificate validation.
- The protocol can be used to download a list of revoked certificates.

Using Revocation Lists

As checking, signing, and rechecking the same certificate over and over can create a great deal of overhead, it is common for the CA to create a periodic list of revoked certificates and sign the entire list at once. This Certification Revocation List (CRL) can be downloaded and will reduce the load on the CA server. An end entity can download the list and know which certificates are revoked.

Of course, if the end entity must download the entire list from every CA with which it deals, that would create an unmanageable system. To help control the deluge of information, the list can be partitioned to reduce the amount of data required. At first blush, this may not seem to reduce the amount of data downloading at all. The X.509 specification allows for the identification of a CRL Distribution Point. The CDP can be used to partition the CRL by a common factor, such as the URL of a distribution list. Thus, an end entity need only download the CRLs of entities for which it holds certificates. Once the CRL is downloaded, it can be used to identify revoked certificates from the point in the partition. The end entity can download the list as frequently as required, but does not need to connect to the server each time the certificate is used. Still, this results in a lot of information being downloaded to obtain the results of a single certificate.

The solution lies in creating a certificate revocation tree. The tree is a hash of the values of revoked certificates and their CAs. By combining them so that each possible combination appears in only a small hash, the amount of data can be significantly reduced for a single certificate query. The hash values allow only small amounts of data to be downloaded to provide confirmation of whether a given certificate has been revoked.

The X.509 specification provides detailed protocols for CRLs. The entries in the list are required to contain certain fields:

- The version number of the X.509 specification used to define the CRL
- The Object ID of the algorithm used to sign the CRL
- The X.500 DN of the CRL issuer
- The date the CRL was issued
- The date that the next CRL will be issued
- The serial number of the revoked certificates

When the X.509 specification was expanded to accommodate the X.509v3 specification, it incorporated the definition and use of additional fields. Standard extensions in this version of the CRL specification include:

- Authority key identifiers that provide a unique identifier for the issuer's key
- Issuer additional name fields, such as an IP address or email address
- CRL number is a unique identifier for this CRL

There are also additional fields created for each certificate

- Reason code for the revocation of the certificate
- Certificate issuer name
- Reason for a suspension—used if the certificate is suspended rather than revoked
- Invalidity date states the date after which the certificate is considered invalid

The Design and Implementation of a PKI

As you consider a PKI, you should organize a formal IT project to produce the results you require. This guide has introduced the wide variety of questions, opportunities, and risks involved in building a PKI. This section aspires to help you put that knowledge to practical use.

Requirements Analysis

The first step in designing your PKI system is to determine what you expect of that system. One of the first considerations is the applications that are supported by you PKI. For instance, if all you intent to support is SSL encryption of Internet traffic, centralized key generation and storage is probably not required. However, if you intend to use keys to encrypt data stored on a file system—data that you might need to recover—centralized key generation and storage becomes more significant. Carefully consider the applications that will leverage your PKI:

- Email encryption to protect sensitive information—Consider whether the email will be persisted in an encrypted manner or decoded and stored in plain text after transmission
- Secure Socket Layer communications through hypertext (HTTPS), file transfer (SFTP), or other protocols—Because of the ubiquitous use of HTTP-based protocols to help work around corporate firewalls, the use of encryption has spread and become a critical application for many Web-oriented services, including Web services—based applications.
- VPNs allow people to connect securely through the public Internet while encrypting the data passed through the TCP/IP channel.
- Protection of ERP systems, such as PeopleSoft, JD Edwards, and SAP—These systems provide encrypted connections to secure communications between systems.
- Single Sign-On (SSO) applications allow users to authenticate to a single network service and access all the services within that network—These services use certificates to validate users to the resources on the network.
- Digital signature for signing documents or signing applications and code to mark it as safe for use.
- Secure electronic transactions through credit cards—Credit card companies are distributing cards with smart chips. These chips contain digital certificates and are used to identify and authenticate the user.

What Is the Primary Purpose of the PKI for Your Organization?

Most organizations will want to establish a PKI to implement security for their employees, customers, and business partners. This may be as simple as building trust with consumers on the Internet and as complicated as meeting the complex requirements of governmental regulations. The driving requirements for this security will help answer many of the questions you will ask as you select systems and establish protocols. For this application, the cost of PKI is overhead, balanced against the cost of security breaches and meeting regulatory compliance.

Some organization may offer PKI as a value-added service. For instance, an Internet Service Provider (IPS) might want to offer SSL services to its clients who purchase Web hosting services. In this case, the balance of cost control and security for you direct customers must be carefully weighed.

Some organizations will want to offer PKI services for sale. In this case, PKI becomes a primary product and the focus of a profit center. Finding the unique blend of pricing and features to distinguish one's organization will serve to determine the services and protocols offered by the PKI.

Design

Once the requirements for the PKI have been established, a system can be designed. The first step at this point is to determine whether to develop the PKI in house or to outsource the PKI. Each side offer advantages and disadvantages.

In-House vs. Outsourced PKI

In-house development and deployment of the PKI can leverage existing resources. If staff already exists that can adequately manage the needs of the PKI and the physical security requirements are in place, it might be economical to develop and deploy with your standing resources. In-house systems can be more flexible and provide you with control of the operation. You are also not at the mercy of the security of an external organization. If your PKI outsource partner were to make changes, they could compel you to make costly modifications on their timeframe rather than your own.

However, many organizations do not have the expertise or additional resources to manage a secure PKI. If protocols are not carefully adhered to, the sanctity of the secured data can be at risk. If under-trained or inexperienced personnel are pressed into service, the results can be gaps in the PKI operation or security. Also, outsource firms are highly motivated to keep the PKI up-to-date and operating smoothly. Their livelihood is dependent on providing a secure operating environment for their customers. If you do not have the existing physical security and expertise, it is often less costly to seek the services of an outside firm.

Mixing and Matching Outsource and In-House

The trusted Root CA is used to validate the subordinate CAs under it. For many organizations, a third-party CA is used as the trusted Root CA for subordinate CAs created by the organization. That shifts the burden of creating a very costly environment to the CA.

If an organization can create their own subordinate CAs, they can be nimble and flexible in meeting their own security needs, while still being able to revoke certificates if a portion of the hierarchy is compromised. But remember, every subordinate CA you create must be secured in its own right. Failure to do so can create a major breach in security and permanently damage the trust you build with your infrastructure users.

Choosing a CA

If you provide externally facing certificates, such as those used by public Web servers, you will want to contract the services of a public CA. That authority will stand as an impartial third-party witness and attest your identity to your customers or business partners.

Previous chapters discussed how to evaluate a CA. Carefully consider their CPS and internal procedures. Query them concerning the services they offer and value-added services that you can leverage. Be certain to have your specific requirements in hand to use as a guideline for the services you need. Also, take time to contact existing customers and learn from their experiences the type of service that you are likely to enjoy. Consider the entire cost of the relationship, not just the price tag on a 2-year, level 3 certificate.

Choosing Components

Some portion of your PKI will be managed in-house. Work with your existing vendors to discover the services they offer that you might already own and can leverage for the build-out of your PKI. For instance, if you have an LDAP server operating within your network, it might have a certificate server extension. You might be able to use that service without adding a new physical server or learning a new operating system (OS). Many companies have Microsoft Windows servers. These servers can be used to create self-signed certificates that are quite suitable for internal use, such as development and test server, internally encrypted application, and the like. They are simple to use and incur little added expense.

Depending on your needs, you might want to investigate best-of-breed solutions. Companies that live or die based on their implementation of PKI solutions may have products that better serve your esoteric needs. Explore the Internet and check with others to learn which products supply the needs of organizations similar to yours. As with a public-facing CA, take time to read customer reviews or speak to customers concerning their individual experiences with a vendor. These conversations often lead to the best decisions.

Establish Practices

To keep your PKI secure, you need to establish proper practices among your operators. You need to develop your own policies and procedures. The recommendations and patterns established by the PKIX working group and other parties provide the framework for creating you own CPS. Those policies must be implemented as training materials and practices used by your operators every day to assure your PKI delivers as designed.

Pilots

When creating a new PKI, it is often best to go slow. Lessons learned as you begin to implement a solution can guide the next step. Even when enhancing or re-organizing a PKI, there is much to be said for taking small, metered steps that lead to the best solution.

Include training of personnel as part of your pilot. Ultimately, the ability to conduct the procedures defined by your PKI requirements is just as important as the servers and systems that operate it day-to-day.

A successful pilot has a set of measurable goals that define success. Implement these measures to discover whether the solution indeed provides the results you require. Take time to compile the lessons learned and use them to improve the process and the next phase of the PKI implementation.

Roll-Out and Monitoring

Once the design is set and the system is confirmed, the wide-spread deployment of the PKI is scheduled and implemented. Without question, regardless of the careful planning and testing, small hitches and obstacles will be encountered. Collecting these issues and documenting them, along with their solutions, will help you refine and improve your PKI system going forward.

Your system should include monitoring of critical systems. You need to audit use of the system and the activities of your personnel. You need to monitor the resources used by your internal systems. Some of these audits may be required for regulatory compliance. Others will help you predict the capacity of your system and proactively allow you to refine and enhance it.

Summary

This chapter has concluded this guide with a close look at the decision-making process involved when choosing between building an in-house PKI and third-party outsource management for your organization. This chapter covered the basics of a PKI and identified the key components of a PKI. It then built on this information by providing practical guidelines for PKI design and implementation.

I hope this guide has helped you to develop and improve your PKI. May your computing be safe a secure!

Download Additional eBooks from Realtime Nexus!