PKI NOTE

November 2002

PKI Basics - A Technical Perspective

This paper is the companion piece to the paper PKI Basics: a Business Perspective in the Forum's PKI Notes Series. Together the two papers provide a concise, vendor neutral introduction to the PKI technology from business and technical perspectives. The audience for this Technical Perspective includes both the businessperson looking for a high level description of the technology and the IT professional who is unfamiliar with PKI concepts. Our goal is to familiarize the reader with the terminology of PKI, the architectural components and how they interact, and the certificate life-cycle management concepts.

Introduction

The electronic information systems today are as complex as the business relationships they need to serve. The words 'Information Security' are now familiar at the highest levels of corporate structures. The security consultant is taking his place as an advisor along with the legal and accounting experts that are essential to conducting business today. Information security, when approached from a corporate perspective, is an enabler of traditional business goals in an electronic environment. Improved revenue through access to new markets, reduced costs through the efficiencies of extranet and internet delivery of information, compliance with government and industry regulations regarding the privacy of personal information, and reduced risk of liability are only a few examples of the business objectives that can be enabled by having a cogent security policy and security delivery infrastructure. The question today is not whether *to build a security infrastructure but rather* which *one to build.* [ROI]

One of the most crucial questions in any business transaction is the identity of the entity with which the transaction is being conducted. Historically, personal relationships, face-to-face contract signings, notaries, and third party counsel are used to help establish trust in this most important aspect of conducting our business. As the reliance on paper shifts to electronic transactions and documents, so must the reliance on traditional trust factors shift to electronic security measures to authenticate our electronic business partners, customers, and suppliers before engaging in the exchange of information, goods, and services. Similarly, the need for confidentiality and confidence in the integrity of exchanged information is critical. Extending this list of security services, there may be further need to establish the non-repudiation of agreements, and to digitally notarize and securely timestamp transactions. [BUS]

Acknowledgements

"PKI Basics - A Technical Perspective" is a deliverable from the PKI Forum's Business Working Group (BWG). Several member organizations and individuals have contributed by providing content, editorial assistance and editorial reviews.

Authors:

Shashi Kiran
Nortel

Patricia Lareau
PKI Forum

Steve Lloyd
PKI Forum

Table of Contents

PKI forum

As the world of commerce becomes increasingly dependent on the electronic storage, accessibility, and delivery of valuable information, the question of maintaining a level of trust in all those business processes, which is commensurate with the levels well established in the brick and mortar world, becomes critical. All of the security services mentioned above must be utilized to maximize the advantages of electronic commerce. PKI provides a well-conceived infrastructure to efficiently deliver these services in a cohesive manner. PKI is a long-term solution, as any infrastructure should be considered. Its return is through the ongoing progression of business applications it enables to conduct business electronically…safely. From the cryptographic underpinnings and building blocks through the architecture, certificate life cycle management, and deployment topics, this note is meant to give a vendor-neutral introductory explanation of the PKI technology at work.

The Cryptographic Building Blocks

To facilitate the architectural discussions that follow, this section describes, at a high level, the cryptographic underpinnings of the technology and why it provides such valuable elements with which to build a security infrastructure. Cryptography is fundamentally based on the use of keys that are used to encrypt and decrypt data[1]. There are two types of cryptography: 1) secret key or symmetric and 2) public key or asymmetric. Secret key cryptography is characterized by the fact that the same key used to encrypt the data is used to decrypt the data. Clearly, this key must be kept secret among the communicating parties; otherwise the communication can be intercepted and decrypted by others.

Until the mid 1970's, symmetric cryptography was the only form of cryptography available, so the same secret had to be known by all individuals participating in any application that provided a security service. Although this form of cryptography was computationally efficient, it suffered from the fact that it could not support certain security services, and it presented a difficult key management problem since the secret keys had to be distributed securely to the communicating parties. However, this all changed when Whitfield Diffie and Martin Hellman introduced the notion of public key cryptography with the publication of their "New Directions in Cryptography" paper [DH] in 1976. This represented a significant breakthrough in cryptography because it enabled services that could not previously have been entertained as well as making traditional security services more expedient.

Public key cryptography is based on the use of key pairs. When using a key pair, only one of the keys, referred to as the private key, must be kept secret and (usually) under the control of the owner. The other key, referred to as the public key, can be disseminated freely for use by any person who wishes to participate in security services with the person holding the private key. This is possible because the keys in the pair are mathematically related but it remains computationally infeasible to derive the private key from knowledge of the public key. In theory, any individual can send the holder of a private key a message encrypted using the corresponding public key and ONLY the holder of the private key can read the secure message (i.e. can decrypt it). Similarly, the holder of the private key can establish the integrity and origin of the data he sends to another party by digitally signing the data using his private key. Anyone who receives that data can use the associated public key to validate that it came from the holder of the private key and verify the integrity of the data has been maintained.

Why Sign?

In electronic commerce, the establishment of trust is key. Not only must we trust the identity of the business partner but we must also have utmost confidence in the transaction itself. Digitally signing a transaction can achieve both of these trust objectives. Public Key is a well-vetted, well-understood technology. The trust is built into the Infrastructure by design. No other signing solution can provide the pre-conditions for legal validity as consistently or comprehensively as a PKI. Electronic signatures alone cannot address all the 'real world' issues associated with signatures, but PKI digital signatures can mitigate those risks introduced by the electronic environment.

[1] Encryption is a mathematical transformation that takes plaintext information and makes it unintelligible (referred to as ciphertext). Decryption is the process that reverts the ciphertext back to plaintext.

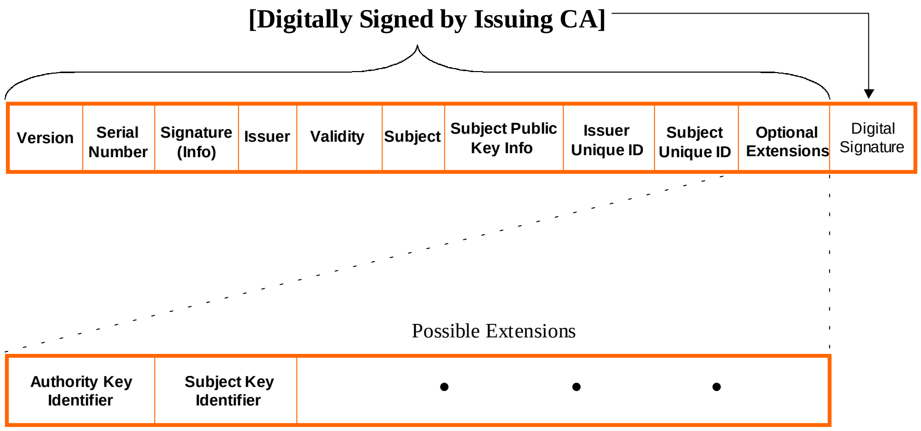
This entire concept was revolutionary. One of its initial uses was to facilitate the delivery of keys to be used in symmetric cryptographic functions. Prior to this, the delivery of secret keys was arduous to set up and could not even be accomplished if the persons involved did not know each other. It also reduces the number of keys that must be used within a system. To keep communications secure using symmetric cryptography, each person in the system must have a different key for each person with whom he communicates; in the system of n users, there are on the order of n^2 keys. Under a public key scheme, there only needs to be one key pair per person in the system, or n key pairs in the system. This is a valuable advantage.

One significant factor that has been hand-waved until this point in the discussion is trust. Since transactions can be no more secure than the system in which they occur, the most important element becomes establishing a way for correspondents to locate each other and have confidence that the public key they use truly belongs to the person (or machine) with whom/which they wish to communicate. A Public Key Infrastructure is designed to provide this trust. Using a data element called a ***digital certificate*** or ***public key certificate,*** which binds a public key to identifying information about its owner, the infrastructure is designed to create the binding, and manage it for the benefit of all within the community of use. Figure 2-1 illustrates the Version 3 public key certificate as defined in X.509.

*Using a data element called a **digital certificate** or **public key certificate,** which binds a public key to identifying information about its owner, the infrastructure is designed to create the binding, and manage it for the benefit of all within the community of use.*

Figure 2-1: Version 3 Public Key Certificate.

Data Fields and Extensions are defined in the X.509 standard.



Version	Serial Number	Signature (Info)	Issuer	Validity	Subject	Subject Public Key Info	Issuer Unique ID	Subject Unique ID	Optional Extensions	Digital Signature

Authority Key Identifier	Subject Key Identifier

Although PKI derives its name from Public Key Cryptography, some of the services it provides have their technical roots in techniques that are outside this branch of cryptography. PKI embodies the best of these well-understood techniques. PKI represents the integration of public key cryptography used for digital signatures and key management, and symmetric key cryptography used for encryption.

PKI represents the integration of public key cryptography used for digital signatures and key management, and symmetric key cryptography used for encryption.

As stated earlier, Whitfield Diffie and Martin Hellman first introduced the notion of public key cryptography in 1976 with the publication of "New Directions in Cryptography" [DH]. A great deal of progress has been made since then, including the development of public key cryptographic algorithms such as RSA [RSA], DSA [DSA], and the class of cryptographic algorithms based on Elliptic Curve Cryptography [ECC]. However, it is only within the last 10 years or so that technology has become available to manage the public/private key pairs. This managed solution is referred to as Public Key Infrastructure (PKI), and it provides the foundation for offering scaleable key and certificate life cycle management.

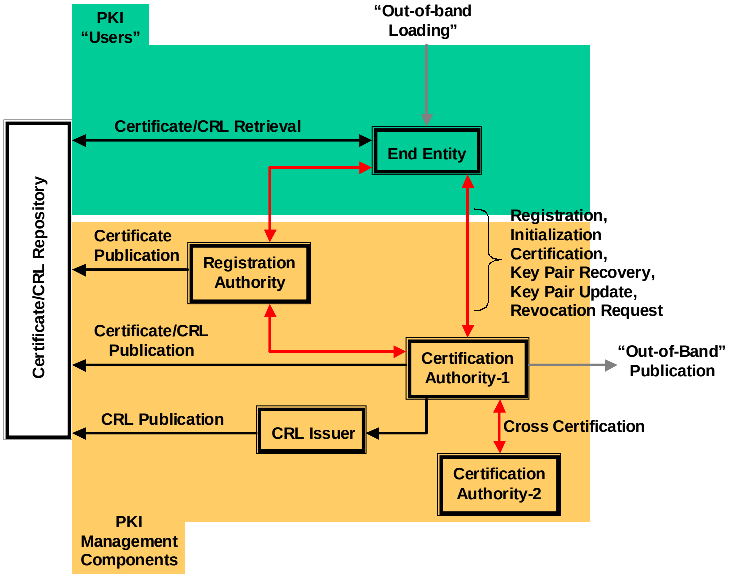
First and foremost, PKI is an authentication technology. Using a combination of secret key and public key cryptography, PKI enables a number of other security services including data confidentiality, data integrity, and key management. The very foundation or framework for PKI is defined in the ITU-T X.509 Recommendation [X.509]. The Internet Engineering Task Force (IETF) Public Key Infrastructure X.509 (PKIX) working group has been the driving force behind setting up a formal (and generic) model based on X.509 that is suitable for deploying a certificate-based architecture on the Internet. The purpose of this section is to describe this model, and to summarize the key management functions that are realized through this infrastructure[2].

First and foremost, PKI is an authentication technology. Using a combination of secret key and public key cryptography, PKI enables a number of other security services including data confidentiality, data integrity, and key management.

3.1 The PKIX Architecture Model

The basic PKIX architecture model has remained largely unchanged since it was first published in the original Internet Certificate and Certificate Revocation List (CRL) Profile [RFC2459]. The latest model is reflected in the most recent version of the Internet Certificate and CRL Profile [RFC3280]. Figure 3-1 illustrates our interpretation of this model, and Table 3-1 identifies the name and purpose of each component represented. These components are described in greater detail in the subsections that follow.

Figure 3-1: The PKIX Architecture Model


PKI "Users"
"Out-of-band Loading"
Certificate/CRL Retrieval
End Entity
Certificate/CRL Repository
Certificate Publication
Registration Authority
Registration, Initialization Certification, Key Pair Recovery, Key Pair Update, Revocation Request
Certificate/CRL Publication
Certification Authority-1
"Out-of-Band" Publication
CRL Publication
CRL Issuer
Cross Certification
Certification Authority-2
PKI Management Components

The CA is the very foundation of the PKI since it is the only component that can issue public key certificates. Public key certificates are digitally signed by the issuing CA (which effectively binds the subject name to the public key).

[2] Although this paper is technology-oriented, it should be noted that PKI is more than technology. This is reflected in numerous references where PKI is defined as "...the set of hardware, software, people, policies and procedures needed to create, manage, store, distribute and revoke public-key certificates...".

Table 3-1: PKIX Components

COMPONENT	PRIMARY ROLE
• End Entity	End Entity is a generic term used to denote end-users, devices (e.g., servers, routers), or any other entity that can be identified in the subject field of a public key certificate. End entities typically consume and/or support PKI-related services.
• Certification Authority (CA)	The CA is the issuer of certificates and (usually) CRLs. It may also support a variety of administrative functions, although these are often delegated to one or more Registration Authorities.
• Registration Authority (RA)	The RA is an optional component that can assume a number of administrative functions from the CA. The RA is often associated with the End Entity registration process, but can assist in a number of other areas as well.
• Repository	A repository is a generic term used to denote any method for storing certificates and CRLs so that they can be retrieved by End Entities.
• CRL Issuer	The CRL Issuer is an optional component that a CA can delegate to publish CRLs.

3.1.1 End Entities

End Entities are sometimes thought of as end-users. Although this is often the case, the term End Entity is meant to be much more generic. An End Entity can be an end-user, a device such as a router or a server, a process, or anything that can be identified in the subject name of a public key certificate. End Entities can also be thought of as consumers of the PKI-related services. There are even cases when a provider of PKI-related services is considered to be an End Entity. For example, a RA is considered to be an End Entity from the point of view of the CA.

End Entities that will be bound to certificates, such as servers and end users, must “enroll” into the PKI before they can participate as members of the PKI. This involves an initial registration step followed by initialization and certification as discussed in Section 3.2.

3.1.2 Certification Authority (CA)

Public keys are distributed in the form of public key certificates. The CA is the very foundation of the PKI since it is the only component that can issue public key certificates. Public key certificates are digitally signed by the issuing CA (which effectively binds the subject name to the public key). CAs are also responsible for issuing CRLs unless this has been delegated to a separate CRL Issuer.

CAs may also be involved in a number of administrative tasks such as end-user registration, but these are often delegated to the Registration Authority (RA). In implementation practice, CAs can also serve as the key backup and recovery facility although this function can also be delegated to a separate component.

CAs are often thought of as the “source of trust” in a PKI. The PKI Forum Note, CA Trust, endorses standardized frameworks for the establishment and auditing of the policies and procedures required for the operation of a PKI [CAT].Typically, End Entities are configured with one or more “trust anchors” which are then used as the starting point to validate a given certification path.[3] See the PKI Forum white paper on Certification Path Construction for additional information. [CPC]

Why Verify?

Did your bank ever process a check you had forgotten to sign? Banks overloaded with paper can’t hand check every signature on every check even though they should and wish they could. They accept the risk inherent in their processing. By not verifying an electronic signature the relying party is taking the same sort of risk. A Digital signature is designed to make verification available immediately if the relying party determines that the risk inherent in NOT doing so is significant enough. Digital Signatures allow the amount of risk in a process to be controlled by providing a mechanism by which to mitigate it.

[3] A *certification path* is a chain of certificates between any given certificate and its trust anchor (CA). Each certificate in the chain must be verifiable in order to validate the certificate at the end of the path; this functionality is critical to the usable PKI.

3.1.3 Registration Authority (RA)

A Registration Authority (RA) is an optional component that can be used to "offload" many of the administrative functions that a CA would have to assume in the absence of a RA. As stated earlier, the RA is normally associated with the End Entity registration process. This would include the verification of the identity of the End Entity attempting to register with the PKI. However, the RA can perform a number of other functions, including:

- Validating the attributes of the subject who is requesting the certificate
- Verifying that the subject has possession of the private key being registered (known as "Proof of possession")
- Generation of shared secrets to support the initialization and certification process
- Public/private key pair-generation
- Conducting interactions with the CA (or several CAs) as an intermediary of the End Entity, including key compromise notifications and key recovery requests
- Parameter validation of public keys presented for registration
    - *ANSI X9 standards provide guidance with algorithm-specific details* [ANS]

Judicious deployment of RAs can provide two primary advantages. First, RAs can help to reduce overall costs… Second, offloading the administrative functions from the CA allows an organization to operate their CA off-line, which reduces the window of opportunity to mount remote attacks against that CA.

Note that although the RA can offload many functions from the CA, the RA can never be the issuer of a public key certificate.

Judicious deployment of RAs can provide two primary advantages. First, RAs can help to reduce overall costs. This is especially true in large, geographically dispersed organizations that require their users to be physically present before certain PKI-related activities are permitted. A typical example would be end-user registration, but other PKI-related functions such as end-user initiated requests for certificate revocation or key pair recovery might also apply. There may also be other practical considerations, such as when an organization elects to outsource the CA service but retain control of the registration process. Second, offloading the administrative functions from the CA allows an organization to operate their CA off-line, which reduces the window of opportunity to mount remote attacks against that CA.

In the context of a PKI, a repository is a generic term used to denote any method for storing and retrieving PKI-related information such as public key certificates and CRLs.

3.1.4 Repositories

The term repository is often associated with a directory, but this is not necessarily the case. In the context of a PKI, a repository is a generic term used to denote any method for storing and retrieving PKI-related information such as public key certificates and CRLs. A repository can be an X.500-based directory with client access via the Lightweight Directory Access Protocol (LDAP), or it may be something much simpler such as retrieval of a flat file on a remote server via the File Transfer Protocol (FTP) or the Hyper Text Transfer Protocol (HTTP). The IETF PKIX working group has addressed several "operational protocols" to facilitate the distribution of public key certificates and CRLs, including LDAP, HTTP, and FTP.

It is also possible to offload certain functions from the client system to a trusted third party. For example, the Online Certificate Status Protocol [RFC2560] can be used to "ask" a trusted third party about the revocation status of one or more certificates. Arguably, this could also be viewed as a repository since the revocation status is derived and returned to the client system in response to a request for PKI-related information. The PKIX working group is also working on several protocols to offload the certification path construction and validation process from the client system (refer to Section 3.4).

In any case, the key here is that End Entities must have some mechanism to retrieve the necessary certificates and CRLs, or they must be able to request that this is done on their behalf.

3.1.5 Certificate Revocation List Issuers

The CRL Issuer is just as its name implies – it is the issuer of a CRL. Typically, the CA that issues a given set of certificates is also responsible for issuing revocation information associated with those certificates. However, it is possible for a CA to delegate that function to another entity. CRLs that are issued by another entity are referred to as indirect CRLs. Although the fact that this appears to be a new component in the PKIX architecture model, the notion of indirect CRLs has been standardized in the X.509 Recommendation for some time. This is simply now more explicit in the PKIX architecture model.

3.2 PKIX Management Functions

PKIX identifies a number of management functions that "potentially need to be supported by management protocols" [RFC3280]. Figure 3-1 illustrates the interaction between the various PKI components and it summarizes the types of management functions that might occur between these components. These particular management functions are discussed in more detail in the subsections that follow. Note that one or more of these functions may also occur off-line. Also note that additional functions may be supported as discussed in Section 3.2.8. The PKIX management protocols that might be used to realize these functions are introduced in Section 3.3.

3.2.1 Registration

End Entities must "enroll" into the PKI before they can take advantage of the PKI-enabled services. Registration is the first step in the End Entity enrollment process. This is usually characterized as the process whereby an End Entity first makes itself known to a CA [RFC3280]. This step is usually associated with the initial verification of the End Entity's identity. The rigor or "level of assurance" associated with the registration process will tend to vary based on the target environment, intended use of the certificate, and the associated policies. As noted above, the process of registration could be accomplished directly with the CA or through an intermediate RA. This process may also be accomplished on-line or off-line (or a combination of the two).

Once the identity of the End Entity is verified in accordance with the applicable policies, the End Entity is typically issued one or more shared secret(s) and other identifying information that will then be used for subsequent authentication as the enrollment process continues. The distribution of the shared secret(s) is typically performed out-of-band and may in fact be based on pre-existing shared secret(s).

3.2.2 Initialization

*I*nitial registration is followed by initialization. At a minimum, this involves initializing the associated trust anchor with the End Entity. Additional information such as applicable certificate policies may also be supplied.

In addition, this step is usually associated with initializing the End Entity with its associated key pair(s). Key pair generation involves the creation of the public/private key pair associated with an End Entity. Key pair generation can occur in advance of the End Entity enrollment process or it can take place in response to it. Key pairs can be generated by the End Entity client system, RA, CA or some other component such as a hardware security module. The location of the key pair generation is dictated by operational constraints and applicable policies. Often, the intended use of the keying material plays a critical role in determining where the key pairs should be generated.

It is possible that portions of this step may occur at different times. On the Internet, for example, browsers are initialized with the public keys of numerous root CAs that might be used as trust anchors. However, the end-user portion of initialization would not occur until an explicit certificate request is made. Further, end-users may import additional trust anchors over time.

Why doesn't the deployment always go smoothly?

The early adopters for any technology are pioneers. Remember when everyone had to have a database but no one knew how to integrate it into existing systems? Early adopters of PKI had no community of successful implementations to mimic, and no reference for how to plan and execute the deployment. The technology is not the largest stumbling block. Today, awareness of the importance of planning and project management has significantly improved because of those early deployments. The PKI Forum provides a community of interest to enhance this process through the sharing of knowledge. A growing portfolio of successful deployments, along with improved implementations of the technology, makes PKI an even smarter choice today.

3.2.3 Certification

Certification is the natural conclusion to the End Entity enrollment process. As its name implies, this step involves the issuance of the End Entity public key certificate by the CA. If the key pair is generated external to the CA, the public key component must be conveyed to the CA in a secure manner.

Once generated, the certificate is returned to the End Entity and/or published to a certificate repository.

Although we have presented registration, initialization and certification as separate management functions, note that two or more of these can be combined into a single protocol operation [RFC3280]. For example, this is the case with the PKIX Certificate Management Protocols [RFC2510], which is discussed briefly in Section 3.3.

Certification is the natural conclusion to the End Entity enrollment process. As its name implies, this step involves the issuance of the End Entity public key certificate by the CA. If the key pair is generated external to the CA, the public key component must be conveyed to the CA in a secure manner.

3.2.4 Key Pair Recovery

Key pairs can be used to support digital signature creation and verification, encryption and decryption, or both. When a key pair is used for encryption/decryption, it is important to provide a mechanism to recover the necessary decryption keys when "normal" access to the keying material is no longer possible, otherwise it will not be possible to recover the encrypted data.[4] Normal access to the decryption key can result from forgotten passwords/PINs, corrupted disk drives, damage to hardware tokens, et cetera. Key pair recovery allows End Entities to restore their encryption/ decryption key pair from an authorized key backup facility (typically, the CA that issued the End Entity's certificate).

It is also possible that an End Entity's association with an organization can change (for example, in the case of employee resignation, dismissal, or personal injury), and the organization has a legitimate need to recover data that has been encrypted by that End Entity. It is also possible that access to the keying material may be required in association with legitimate law enforcement requirements. Key pair recovery can be used to support both of these requirements as well.

3.2.5 Key Pair Update

Certificates are issued with fixed lifetimes (referred to as the "validity period" of the certificate). While these fixed lifetimes can be rather generous (say two to five years or so), the certificate will eventually expire. Key pair update may also be required as a result of certificate revocation as discussed in Section 3.2.6. Key pair update involves generation of a new key pair, and the issuance of a new public key certificate[5].

Key pair update can occur in advance of a given key pair's expiration. This will help to ensure that the End Entity is always in possession of a legitimate key pair. Although the PKIX working group recommends against the use of this feature on the Internet [RFC3280, Section 4.2.1.4], it is possible to establish different validity periods for the private and public keys that are used to digitally sign and verify. This would force a key pair update before the associated public key actually expires. It also provides a window of time where the non-expired public key certificate can be used to verify digital signatures that were created with the now expired private key. This will help to minimize irrelevant warning messages that would otherwise be displayed to the End Entity.

[4] Key recovery of keys used to produce signatures is prohibited by most security policies.

[5] Note that this is different from *certificate update* which involves the issuance of a new certificate, but it does not involve the generation of a new key pair (i.e., the old key pair continues to be used).

3.2.6 Revocation Request

As mentioned above, public key certificates are issued with fairly generous lifetimes. However, the circumstances that existed when the certificate was issued can change before the certificate would naturally expire. Reasons for revocation include private key compromise, change in affiliation, name change, et cetera (specific reason codes are defined in X.509).

Therefore, it is sometimes necessary to revoke a certificate before its expiration date. The Revocation Request allows an End Entity (or RA) to request revocation of a given certificate. Of course, out-of-band mechanisms may also be supported/required, and the End Entity may not be involved with the revocation process at all.

Certificate revocation information must be made available by the CA that issued that certificate or by the CRL Issuer to which the CA delegates this function. X.509 defines a method for publishing this information via Certificate Revocation Lists (CRLs). The frequency of publication and the type of CRLs used are a function of local policy. The publication and retrieval of CRLs is represented in Figure 3-1.

The PKIX working group has also introduced several protocols that are designed to provide certificate status information on-line. Work in this area continues as discussed further in Section 3.4.

Note that End Entities, or trusted third parties operating on their behalf, must check the revocation status of all certificates in a given certification path. This includes revocation information about End Entity certificates as well as intermediate CAs.

3.2.7 Cross-Certification

As illustrated in Figure 3-1, cross-certification occurs between CAs. A cross-certificate is a public key certificate that is issued by one CA to another CA. In other words, a cross-certificate is a public key certificate that contains the public key of a CA that has been digitally signed by another CA.

Many interpret cross-certification to mean "inter-domain" cross-certification. However, "intra-domain" cross-certification is also possible. This can be illustrated by using the Government of Canada (GOC) PKI as an example. Major departments within the GOC PKI cross-certify with the Canadian Central Facility, which acts as a bridge CA between these departments. As these departments all "belong" to the GOC PKI, this is "intra-domain" cross-certification. The Canadian Central Facility is also responsible for cross-certification with external PKI domains such as the US Federal Bridge CA. This is "inter-domain" cross-certification.

It should also be noted that cross-certification can be bi-directional or unidirectional. Bi-directional cross-certification typically occurs between peer CAs as described in the previous paragraph. Unidirectional cross-certification typically occurs in a hierarchical trust model where superior CAs issue cross-certificates to subordinate CAs, but the reverse is not true.

3.2.8 Additional Management Functions

The previous subsections have discussed the management functions specifically identified within the PKIX Internet Certificate and CRL Profile [RFC3280, Section 3.5]. However, there are additional management functions that might be required in certain environments, and the PKIX working group recognizes this in some of the

SSL

Everyone knows that the 'lock' displayed on the browser during an Internet session means the connection is secure. Or is it? SSL is a PKI–based protocol that can provide authentication, confidentiality, and data integrity when it is implemented properly. The purpose of SSL is to instill trust but without a trusted infrastructure behind the certificate delivered by the browser, trust is not achieved. Until recently only high assurance SSL certificates were issued by public Certificate Authorities. The requirements in this process assure that the identity associated with the certificate is the true and rightful owner (e.g. website). Today there are lower assurance certificates being issue that are not intended to provide authentication. Unfortunately our browsers cannot distinguish between the two. Commerce on the Internet will drive corrections to this dilemma because PKI underpinnings are critical to the Internet business model.

certificate management protocols that have been defined. The PKIX Certificate Management Protocols [RFC2510] probably provides the most complete set of management functions that might be required in a comprehensive PKI. Some of these additional management functions include:

- *CA key update announcement* – provides a mechanism for a CA to explicitly advertise CA key rollover information.
- *Certificate announcement* – provides a method for announcing the existence of a certificate when no other method (e.g., a repository) is available.
- *Revocation announcement* – provides a method to inform a particular End Entity that their certificate has been (or is going to be) revoked.
- *CRL announcement* – provides a method for a CA to advertise the issuance of new CRLs.
- *Certificate confirmation* – used by the End Entity to explicitly accept or reject the issued certificate.
- *Key Archive* – used to explicitly request private decryption key backup.

Key archive can be used to support the key recovery management operation discussed in Section 3.2.4. The concept of key archive can also be extended to support long-term storage of decryption keys as well as public key certificates used to verify digital signatures. This accomplishes several objectives. First, it allows End Entities to recover their key histories[6] over time. Second, long-term storage of public key certificates, and any associated revocation information, supports the ability to verify digital signatures that may have been created years in the past, well after the associated keys have expired.

The Revocation Request allows an End Entity (or RA) to request revocation of a given certificate… Certificate revocation information must be made available by the CA that issued that certificate or by the CRL Issuer to which the CA delegates this function. X.509 defines a method for publishing this information via Certificate Revocation Lists (CRLs).

3.3 PKIX Management Protocols

Management protocols can be used to support on-line protocol exchanges between various PKI components as illustrated in Figure 3-1 and described in Section 3.2. The IETF PKIX working group has developed two fairly comprehensive management protocols that can be used to support this component-level interaction. The first is the Certificate Management Protocols (CMP) based on RFC2510 and the associated Certificate Request Message Format (CRMF) based on RFC2511. The second management protocol is the Certificate Management Messages over the CMS (CMC) based on RFC2797.

Management protocols can be used to support on-line protocol exchanges between various PKI components as illustrated in Figure 3-1 and described in Section 3.2. The IETF PKIX working group has developed two fairly comprehensive management protocols that can be used to support this component-level interaction.

CMP is arguably the most comprehensive PKIX management protocol. All of the management functions discussed in Section 3.2 are explicitly identified as specific protocol exchanges (or as attributes within a specific protocol message). The basic certificate request format is defined in RFC2511. CMP is designed to be a flexible protocol able to accommodate a variety of technical, operational, and business models.

CMP is evolving based on multi-vendor interoperability experience co-sponsored by the PKI Forum and ICSA. A second draft of the CMP is under development (see http://www.ietf.org/html.charters/pkix-charter.html for the latest Internet Draft) and is expected to achieve RFC status later this year.

CMC is also a fairly comprehensive PKIX management protocol, although arguably some of the functions described in Section 3.2 above are not as explicit as they are in CMP. CMC is built upon the earlier work done with other standards such as CMS [RFC2630] and PKCS #10 [RFC2986]. CMC attempts to leverage existing implementations based on CMS and PKCS #10.

[6] *Key history* can be thought of as a local cache of private decryption keys necessary to decrypt data that was encrypted using public encryption keys that have since expired.

Note that other "management protocols" are available, but these tend to offer minimal subsets of the comprehensive management functions discussed in Section 3.2.

3.4 PKIX Certificate Discovery and Validation Protocols

As mentioned above, the PKIX working group is developing protocols that address the need to offload portions (potentially all) of the certificate discovery and/or validation process from the client system. The forerunner to these protocols is the Online Certificate Status Protocol (OCSP) as defined in RFC2560. OCSP is a very simple request/reply protocol that allows clients to ask an "OCSP responder" about the revocation status of one or more certificates. The OCSP responder returns digitally signed responses regarding the status of the certificates identified in the request. OCSP is designed to return real-time responses to client queries, and can provide an efficient method for returning certificate status on demand. However, OCSP offers limited functionality, and work on more comprehensive protocols has been underway for some time.

In August 2002, an IETF Internet Draft was published which identifies the requirements associated with delegated path discovery and delegated path validation [http://www.ietf.org/internet-drafts/draft-ietf-pkix-dpv-dpd-req-05.txt]. PKIX protocols that are designed to offload the path discovery and/or validation process from the client system should meet these requirements. Work on several protocols to meet these requirements is ongoing. Potential candidates include:

- OCSP Version 2 with Delegated Path Discovery and Delegated Path Validation,
- The Simple Certificate Validation Protocol (SCVP), and
- The Certificate Validation Protocol (CVP).

OCSP Version 2 is based on the earlier OCSP Extensions (or OCSP-X) work. When last published as an Internet Draft, it consisted of the OCSP Version 2 protocol, the Delegated Path Discovery protocol, and the Delegated Path Validation protocol. However, at the time of this writing, these Internet Drafts had expired, so it is unclear whether work on these protocols will continue.

SCVP was developed in response to the limitations in OCSP Version 1. SCVP allows the entire certification path construction and/or validation process to be offloaded from the client system. SCVP is currently an Internet Draft that can be retrieved from http://www.ietf.org/html.charters/pkix-charter.html.

CVP is yet a third protocol that was introduced only recently. As its name implies, CVP only addresses delegated path validation. This is yet another competing protocol, and the level of support this protocol will receive from members of the PKIX working group is unclear. CVP is also an Internet Draft that can be retrieved from http://www.ietf.org/html.charters/pkix-charter.html.

This is not the first time we have seen multiple, non-interoperable, protocols designed to accomplish the same function(s). For example, we witnessed a similar occurrence with CMP versus CMC. In general, these protocols tend to have their own advantages and disadvantages, and the PKIX working group membership is sometimes divided when it comes to selecting one protocol over another. Sometimes it is simply not possible to reach agreement within the IETF as to which protocol should be adopted when more than one protocol has been defined to achieve the same thing. When this occurs, the IETF tends to let the industry decide which protocol(s), if any, will dominate the industry. Unfortunately, this creates some degree of uncertainty within the vendor community at times, and it may force some vendors to implement more than one protocol in order to ensure interoperability with multiple vendor products.

We should also note that we may see XKMS play a role in this area in the future. XKMS (or "XML Key Management Specification") is an open standard under development

XKMS (or "XML Key Management Specification") is an open standard under development within the World Wide Web Consortium (W3C). The goal of XKMS is to simplify the integration of PKI security services and digital certificates into Internet applications requiring secure transactions. Developers can efficiently integrate authentication, encryption, and digital signature services, such as certificate processing and revocation status checking, by delegating all or part of the XML digital signature processing to XKMS-compliant Web services.

within the World Wide Web Consortium (W3C). The goal of XKMS is to simplify the integration of PKI security services and digital certificates into Internet applications requiring secure transactions. Developers can efficiently integrate authentication, encryption, and digital signature services, such as certificate processing and revocation status checking, by delegating all or part of the XML digital signature processing to XKMS-compliant Web services. XKMS also supports certain life cycle management functions such as the registration of public key information. Refer to http://www.w3c.org/2001/XKMS/2001/01/xkms-charter.html for additional information.

References

[ANS] ANS X9.30-1997 Digital Signature Algorithm

ANS X9.31-1998 Digital Signatures Using Reversible Public Key Cryptography (rDSA)

ANS X9.62-1999 Elliptic Curve Digital Signature Algorithm (ECDSA)

[BUS] PKI Basics - A Business Perspective, A PKI Forum Note, Patricia Lareau, April 2002, http://www.pkiforum.org/resources.html

[CAT] CA Trust, PKI Forum Note, Jeff Stapleton, July 2001, http://www.pkiforum.org/resources.html

[CPC) Understanding Certification Path Construction, PKI Forum White Paper, Steve Lloyd, September 2002, http://www.pkiforum.org/resources.html

[DH] Whitfield Diffie and Martin Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory 22 *(Pages 644-654), 1976*

[DSA] Federal Information Processing Standards Publication (FIPS) 186-2, "Digital Signature Standard", U.S. Department of Commerce, National Bureau of Standards, National Technical Information Service, Springfield, VA, 1994

[ECC] Neil Koblitz, Introduction to Number Theory and Cryptography, Springer, Second Edition, 1994 for description of the basic concepts of Elliptic Curve Cryptography.

[ROI] PKI and Financial Return on Investment, PKI Forum Note, Derek Brink, August 2002, http://www.pkiforum.org/resources.html

[RFC #] Content for all IETF PKIX RFCs may be accessed by number through the URL: http://www.rfc-editor.org/cgi-bin/rfcsearch.pl

[RSA] R. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM 21 (Pages 120-126), 1978

[X.509] ITU-T X.509 Recommendation, "Information Technology – Open Systems Interconnection – The Directory Public Key and Attribute Certificate Frameworks", June 2000 (Equivalent to ISO/IEC 9594-8, 2000)

Copyright Statement

About PKI Forum

The PKI Forum is an international, not-for-profit, multi-vendor and end-user alliance whose purpose is to accelerate the adoption and use of Public-Key Infrastructure (PKI).

The PKI Forum advocates industry cooperation and market awareness to enable organizations to understand and exploit the value of PKI in their e-business applications.

Web: http://www.pkiforum.org
e-Mail: info@pkiforum.org