# Overview of Public Key Infrastructure (PKI)

## 1 Introduction

The section provides an overview of Public Key Infrastructure. It is presented at this point in the Concept of Operations as an aid to the reader because many of the terms and concepts of PKI will be used in subsequent sections.

## 2 Benefits

Electronic ordering systems for controlled substances and controlled substance prescription systems have the capability to (1) reduce the amount of paper, (2) speed transaction times, (3) lower costs per transactions, (4) improve accuracy of entries, (5) improve data archive and retrieval, and (6) improve overall system effectiveness and efficiency.

While these systems can provide the above benefits, they do not alone provide a sufficiently secure infrastructure to permit their employment in every environment.

## 3 Security

PKI technology adds the following security services to an electronic ordering system:

- **Confidentiality** - only authorized persons have access to data.
- **Authentication -** establishes who is sending/receiving data.
- **Integrity -** the data has not been altered in transmission.
- **Non-repudiation -** parties to a transaction cannot convincingly deny having participated in the transaction.

## 4 Fundamentals of Public Key Infrastructure

The sections below introduce the key concepts involved in cryptography and PKI. The reader already familiar with this information may skip this section and proceed to Section 4.

### 4.1 Terms and Definitions

- **Key** – aka cryptographic key, an input parameter that varies the transformation performed by a cryptographic algorithm.
- **Secret key** - a key used in a symmetric cryptographic transformation where the key is protected from being known by any system entity except those who are intended to know it.

- **Private key** – the non-publicly disclosable component of a pair of cryptographic keys used for asymmetric cryptography.

- **Public key** – the publicly–disclosable component of a pair of cryptographic keys used for asymmetric cryptography.

- **Encryption** – cryptographic transformation of data (plaintext) into a form (ciphertext) that conceals the data's original meaning to prevent it from being known or used.

- **Decryption** – cryptographic transformation of data (ciphertext) that restores encrypted data to its original state (plaintext).

- **Hash algorithm (or hash function)** – an algorithm that computes a value based on a data object (such as a message or file; usually of variable length; possibly very large), thereby mapping the data object to a smaller data object (the "hash result") which is usually a fixed-size value.

- **Message digest** – the fixed size result of hashing a message.

- **Secret key (conventional) cryptography** – a synonym for "symmetric cryptography."

- **Symmetric cryptography** – a branch of cryptography involving algorithms that use the same key for two different steps of the algorithm (such as encryption and decryption).

- **Asymmetric cryptography** – a modern branch of cryptography in which the algorithms employ a pair of keys (a public key and a private key) and use a different component of the pair for different steps of the algorithm.

- **Public key cryptography** – synonym for "asymmetric cryptography."

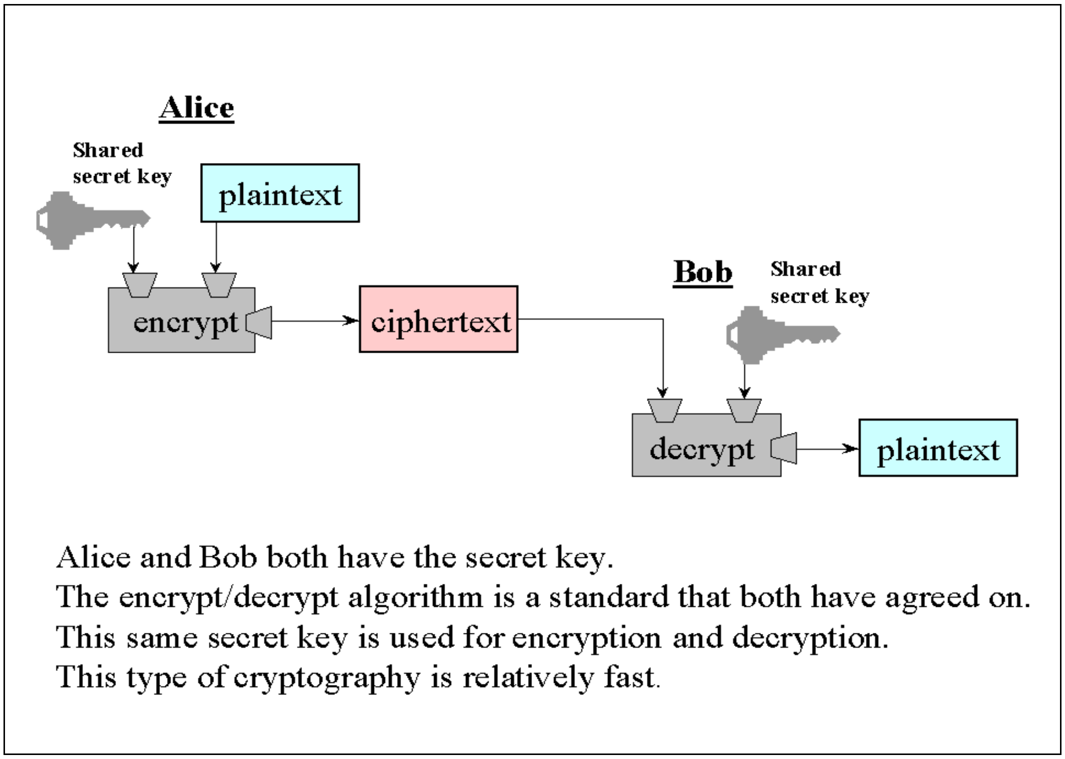
## 4.2 Public Key – The PK in PKI

- **Cryptography**

Cryptography deals with the transformation of ordinary text (plaintext) into a coded form (ciphertext) by encryption and the transformation of ciphertext into plaintext by decryption. Historically, before the advent of mechanical or electrical computers, the transformation was performed by hand and included, for example, the procedures of substitution and transposition. Whether performed by hand or by computer, these procedures, or transformations, are mathematical in nature. The transformation procedure is known as the cryptographic algorithm.

In a computer environment, the encryption and decryption algorithm uses a cryptographic key to perform these mathematical transformations. The key functions as an input parameter to vary the transformation of plaintext to ciphertext and vice versa.

When the cryptographic system uses a single key for both encryption and decryption, the key is known both as a symmetric and secret key. Exhibit 1 illustrates the symmetric key cryptography process.



Alice and Bob both have the secret key.
The encrypt/decrypt algorithm is a standard that both have agreed on.
This same secret key is used for encryption and decryption.
This type of cryptography is relatively fast.

**Exhibit 1. Symmetric Key Process**

A disadvantage of a symmetric key system is that as cryptographic systems increase in scope and complexity, that is, as the number of participants increase, it becomes increasingly difficult and prohibitively expensive to manage the safe distribution of the secret key or keys.

- **Public Key Cryptography**

Public key cryptography, known as asymmetric cryptography, is a modern branch of cryptography in which the cryptographic algorithms employ a pair of keys. Public key cryptography is distinct from traditional, symmetric key cryptography in which the same key is used for both encryption and decryption. The two keys are the public key and the private key, and either can encrypt or decrypt data. A user gives his or her public key to other users, keeping the private key to him or herself. Data encrypted with a public key can be decrypted only with the corresponding private key, and vice versa.

The asymmetric key system does not have the disadvantages of a symmetric key system because the public key is made widely available so that anyone can possess it. In this system only the private key needs to be kept private. Each entity can retrieve another entity's freely available public key, thus removing key distribution management complexity. Exhibit 2 shows the public key cryptography's use of the public and private keys.

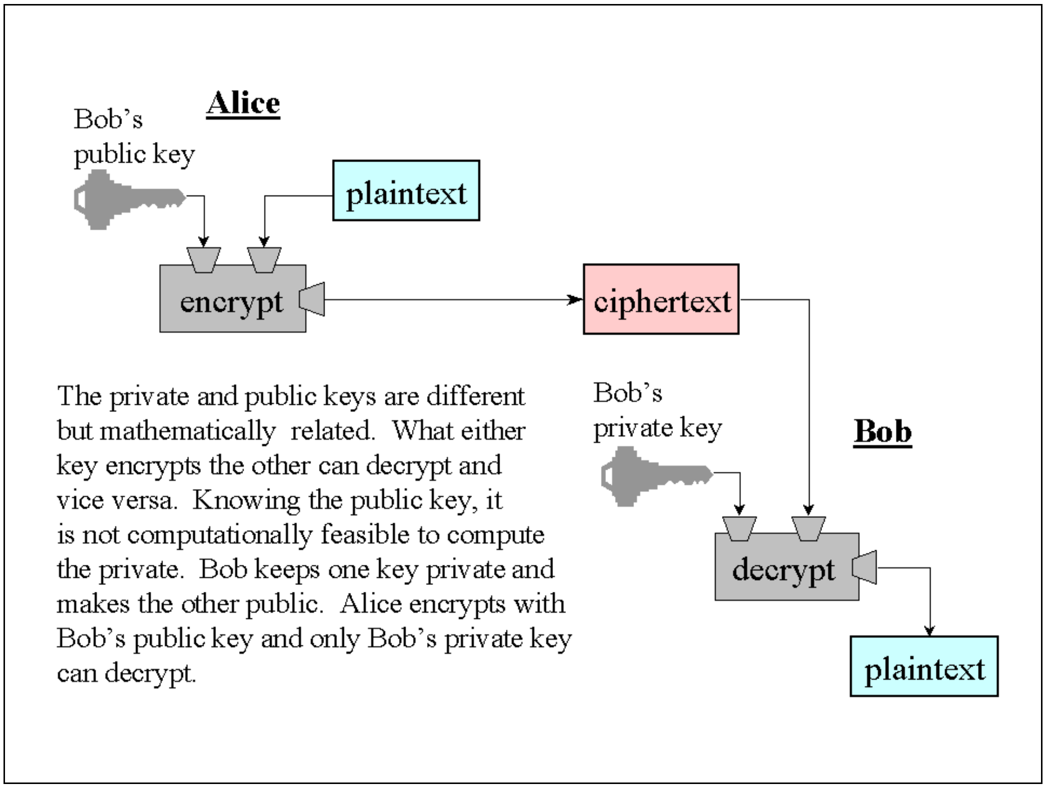
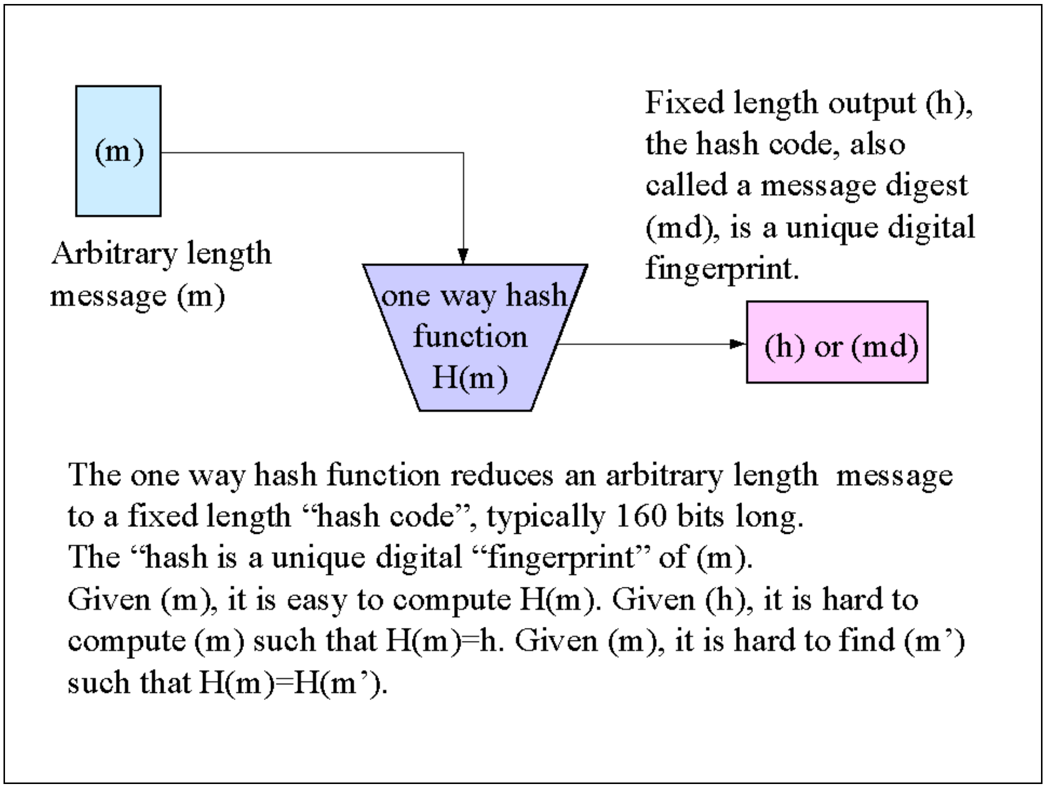

Alice
Bob's
public key
plaintext
encrypt
ciphertext
Bob's
private key
Bob
decrypt
plaintext
The private and public keys are different but mathematically related. What either key encrypts the other can decrypt and vice versa. Knowing the public key, it is not computationally feasible to compute the private. Bob keeps one key private and makes the other public. Alice encrypts with Bob's public key and only Bob's private key can decrypt.

**Exhibit 2. Asymmetric Key Process**

- **Hash function processes**

A cryptographic hash function is a function where it is computationally infeasible to find either (a) a data object (plaintext) that maps to a pre-specified hash result (the one-way property) or (b) two data objects (plaintext A and plaintext B) that map to the same hash result (the "collision-free" property).

Exhibit 3 illustrates the hash process used to generate a fixed size code from any size input message, in this case an arbitrary 160 bit code.


(m)
Arbitrary length message (m)
one way hash function H(m)
Fixed length output (h), the hash code, also called a message digest (md), is a unique digital fingerprint.
(h) or (md)

The one way hash function reduces an arbitrary length message to a fixed length “hash code”, typically 160 bits long.
The “hash is a unique digital “fingerprint” of (m).
Given (m), it is easy to compute H(m). Given (h), it is hard to compute (m) such that H(m)=h. Given (m), it is hard to find (m’) such that H(m)=H(m’).

**Exhibit 3. An Example of a Hash Function Process**

- **Digital Signature**

  A digital signature is a public key cryptography process in which a signer "signs" a message in such a way that anyone can verify that the message was signed by no one other than himself, and that the message has not been modified since he signed it.

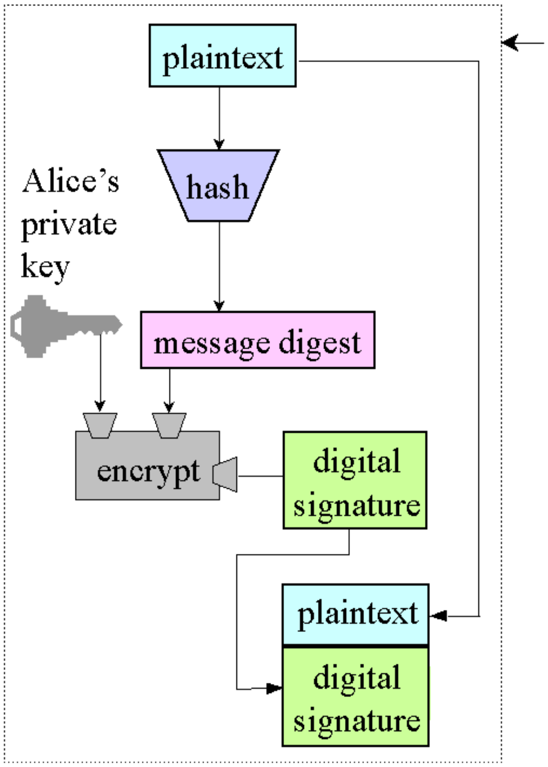
  The digital signature process results in a bit string that allows a recipient of a message to verify the identity of the signer of the message and the integrity of the message. Any one of several digital signature algorithms can generate the bit string. These algorithms have the generic characteristic that private information is used to make a signature and public information is used to verify signatures. A private key should be unique to its owner. If the owner of a private key uses it to encrypt a digital document, that encryption may be assumed to have the same meaning as a paper signature. That is to say, it is a “mark” on the document that only the owner could have made. In many algorithms, the owner does not sign an entire document but rather a digest of a document.

  A typical implementation of digital signature involves a message-digest, a private key for encrypting the message digest, and a public-key for decrypting the message digest. The digital signature procedure is as follows:

  - **The sender.** The software used by the sender computes; using a standard algorithm, a “message digest” from the message. The message digest is unique to the original message in that only the original, unmodified message could have produced the message digest. The sender then encrypts the message digest with his *private key*, yielding an encrypted message digest. He sends the message and the encrypted message digest to a recipient. The two parts together form the digitally signed message.

- **The recipient.** The recipient decrypts the received message digest with the signer's *public key*. The recipient then computes a message digest from the received message using the same algorithm as the signer. He then compares the decrypted received message digest to the computed message digest. If the two are the same, he accepts the message.

Exhibit 4 shows the creation of one type of digital signature.



**Digital Signature Creation:**
Alice "hashes" the plaintext document. Next, she encrypts the message digest with her private signing key. This encrypted digest is her digital signature, a "mark" on the document that only she could have made. The digital signature is then attached to the original plaintext document and sent to the intended recipient.

**Digital Signature Verification:**
The recipient, Bob, "hashes" the plaintext document. He then decrypts Alice's signature with her public key. He next compares the "hash" he obtained with the "hash" in Alice's signature. If they are identical, then the digital signature is verified.

**Exhibit 4. An Example of a Digital Signature Process**

The recipient knows that the signer has sent the message because only the sender's public key will work. However, it still remains that a particular public key be unquestionably associated with a particular individual or organization. Methods of developing trust in public keys are covered in the next section.

## 4.3 Infrastructure – The I in PKI

Components of the PKI infrastructure include:

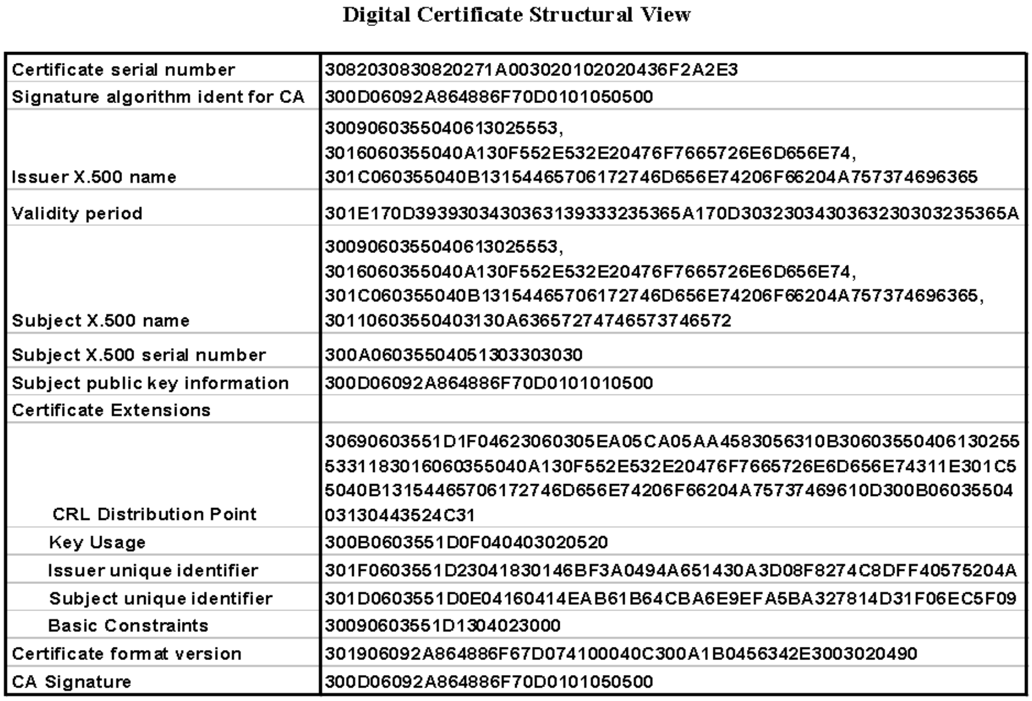
- **Certification Authority (CA)**

  A certification authority (CA) is an entity that creates and then "signs" a document or file containing the name of a user and his public key. Anyone can verify that the file was signed by no one other than the CA by using the public key of the CA. By trusting the CA, one can develop trust in a user’s public key.

  The trust in the certification authority's public key can be obtained recursively. One can have a certificate containing the certification authority's public key signed by a superior certification authority that he already trusts. Ultimately, one need only trust the public keys of a small number of top-level certification authorities. Through a chain of certificates, trust in a large number of users' signatures can be established.

  A broader application of digital certification includes not only name and public key but also other information. Such a combination, together with a signature, forms an extended certificate. The other information may include, for example, electronic-mail address, authorization to sign documents of a given value, or authorization to sign other certificates.

  A logical view of a sample digital certificate is shown in Exhibit 5.



**Digital Certificate Structural View**

| | |
|---|---|
| Certificate serial number | 3082030830820271A003020102020436F2A2E3 |
| Signature algorithm ident for CA | 300D06092A864886F70D0101050500 |
| Issuer X.500 name | 300906035504061302 5553,<br>3016060355040A130F552E532E20476F7665726E6D656E74,<br>301C060355040B13154465706172746D656E74206F66204A757374696365 |
| Validity period | 301E170D3939303430363139333235365A170D3032303430363230303235365A |
| Subject X.500 name | 3009060355040613025553,<br>3016060355040A130F552E532E20476F7665726E6D656E74,<br>301C060355040B13154465706172746D656E74206F66204A757374696365,<br>30110603550403130A63657274746573746572 |
| Subject X.500 serial number | 300A0603550405130330303030 |
| Subject public key information | 300D06092A864886F70D0101010500 |
| Certificate Extensions | |
| CRL Distribution Point | 30690603551D1F04623060305EA05CA05AA4583056310B300603550406130255533118301606035504 0A130F552E532E20476F7665726E6D656E74311E301C55040B13154465706172746D656E74206F66204A75737469610D300B0603550403130443524C31 |
| Key Usage | 300B0603551D0F040403020520 |
| Issuer unique identifier | 301F0603551D23041830146BF3A0494A651430A3D08F8274C8DFF40575204A |
| Subject unique identifier | 301D0603551D0E04160414EAB61B64CBA6E9EFA5BA327814D31F06EC5F09 |
| Basic Constraints | 30090603551D1304023000 |
| Certificate format version | 301906092A864886F67D074100040C300A1B0456342E3003020490 |
| CA Signature | 300D06092A864886F70D0101050500 |

**Exhibit 5. A Sample Digital Certificate**

- **Database**

  A data storage structure where the CA keeps information required for the internal operations of the CA.

- **Repository**

  A system for storing and distributing digital certificates and related information (including CPs, CRLs, and CPSs) to certificate users. The repository may be implemented as a trustworthy logically centralized database. It is often implemented as a remote server based on the Lightweight Directory Access Protocol (LDAP), an X.500 directory, or other directory.

- **Registration Authority**

  The registration authority (RA) is a PKI entity whose function can be separable from the CA. The RA assists the CA in the recording or verifying of information needed by the CA to issue public-key certificates, CRLs, or other certificate management functions.

- **Timestamp Server (TS) and Data Validation and Certification Servers (DVCS)**

  The TS signs a data string or file to establish that the data string or file existed at a particular point in time. A DVCS validates correctness of data and then signs it. TS and DVCS are optional PKI entities.

- **Archive**

  The archive provides long term storage of the certificates, and other valuable records for archival purposes.

## 4.4 Essential Documents of a PKI

The following documents serve as a basis for the detailed implementation, planning, development and direction of operations of a PKI, as well as a basis to establish the level of security and trust model necessary to support the application of the PKI processes.

- **Concept of Operations (CONOPS)**

  The CONOPS sets forth in high level, abstract terms the purpose of a PKI. Although there is no industry standard for this document, it serves to inform an organization's decision-makers about the fundamental concepts and applicability of a PKI. It may include the business rationale for the deployment of a PKI, and may contain a Memo of Understanding (MOU) for the parts of an organization establishing the PKI. It may also include applicable portions of the Certificate Policy (CP).

- **Certificate Policy (CP)**

  The certificate policy serves as a named set of rules that indicates the applicability of a certificate to a particular community and/or class of application having common security requirements. RFC 2527, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework," establishes a standard format for the development of a CP.

- **Certification Practice Statement (CPS)**

  This document, more specific than a CP, describes in greater detail how the CP will be implemented. It is written to comply with RFC 2527.

## 4.5 PKI Management Functions

PKI functions are performed in context of the structure of Exhibit 6.

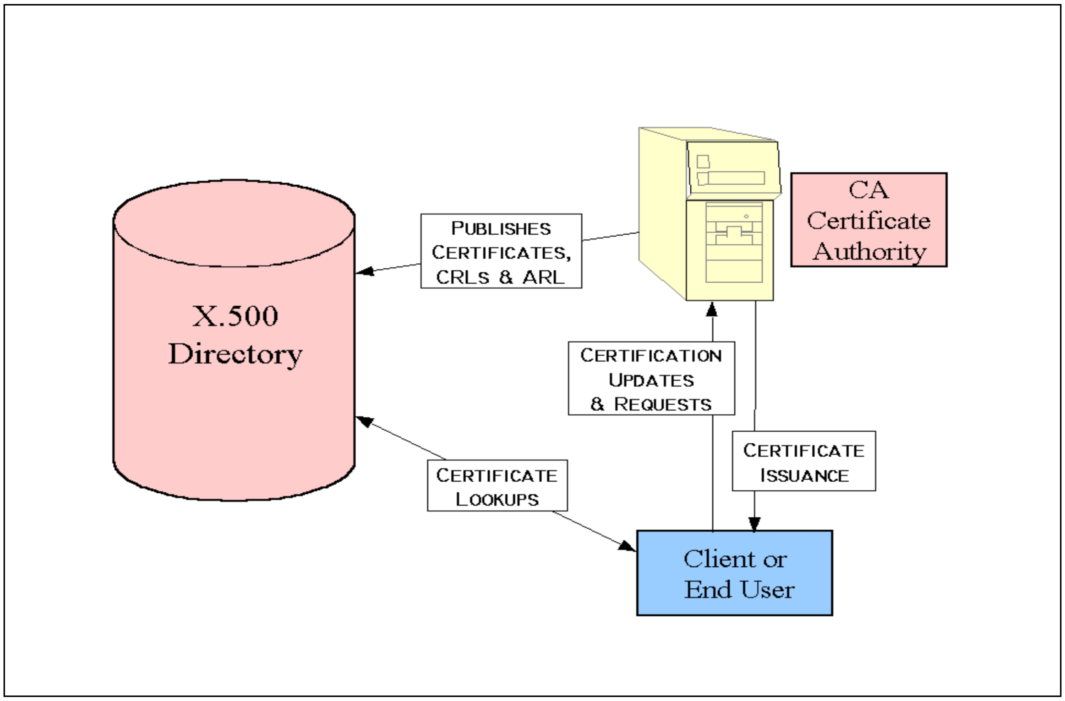


**Exhibit 6. PKI Functions Performed**

The following activities further identify management functions performed in a PKI.

- **Registration**

  The process whereby an applicant, who is the subject of a certificate, makes himself known to the CA, either directly or through a RA. The applicant's name, IP address, domain name, and/or other attributes are placed in the certificate. The CA/RA registers the new applicant by verifying the data provided by the applicant in compliance with the CPS.

- **Initialization**

  In the initialization phase the applicant receives the values to begin communicating with the CA or RA. These values could be the public key or Public Key Certificate (PKC) of the CA or the public/private key pair of the applicant. The initialization must be performed through a trusted channel.

- **Certification**

  Certification is the process wherein the CA issues a public-key certificate for a subject's public key and returns that public-key certificate to the subject and/or posts that public-key certificate to a repository.

- **Key generation**

  Depending on the CA's policy, the user's private/public key pair can either be generated by the user in his local environment, or generated by the CA. If generated by the CA, then the private key must be distributed in a secure manner to the user.

- **Key pair recovery**

  There is sometimes a business case for recovery of private signing keys, for example, the user may forget his password and therefore be unable to access his private key. Where this is the case, there are two classes of key recovery techniques: key escrow and key encapsulation, with each technique having it's own merits. The determination of preferred key recovery technique to be used is dependent upon the business organization's specific needs and requirements.

- **Key expiration**

  Key pairs expire at the end of their period of validation. Each expired key pair must be replaced by generating a new key pair and issuing a new public-key certificate.

- **Key compromise**

  The user's private key is subject to compromise. It is the responsibility of the user to maintain the security of this key since it is equivalent to a written signature. The private key should be considered compromised whenever it is stolen, duplicated, or whenever it's security status is in doubt. A compromised private key requires the generation of a new key pair and issuing a new public-key certificate.

- **Certificate expiration**

  The user's public-key certificate expires at the time of expiration of the public/private key pair. The expired certificate is replaced with a new public-key certificate when the user performs re-registration.

- **Cross-certification**

  Cross-certification is the process by which a public-key certificate is issued by one CA to another CA. The public-key certificate contains the public key associated with the signing CA. An end entity in one domain can establish a trust path with an end entity in another domain through a cross-certification process. For example, Alice trusts CA-1 and Bob trusts CA-2. If CA-1 and CA-2 cross-certify, Alice and Bob will have a trusted path.

- **Revocation**

  The revocation process utilizes Certification Revocation Lists (CRLs) in the following process description:

  > A public-key certificate has a validity period when it is issued. However, circumstances can require the CA to invalidate the public-key certificate before the end of the period; for example, due to a name change, termination of employment, or compromise of the private key. Therefore, in response to such events, the CA periodically issues a signed Certificate Revocation List of public-key certificates whose validity period may have not yet expired, but nevertheless are invalid for one reason or another. The CRLs are posted to the Repository where they are available to the users of the system. Additionally, CRLs can be distributed via un-trusted networks to other repositories, because their contents are protected from undetected alteration through the "hashing" process illustrated in Exhibit 3.

  If, for any reason, a user's certificate appears in a CRL, then the user's certificate is considered invalid by the system. The user will be unable to successfully accomplish transactions until a new private/public key pair and public-key certificate are obtained.