

# Topic A: Public key cryptography

This topic covers the following CompTIA Security+ objectives for exam SY0-301.

#	Objective
6.3	<b>Explain the core concepts of public key infrastructure</b> <ul style="list-style-type: none"><li>• Certificate authorities and digital certificates<ul style="list-style-type: none"><li>– CA</li><li>– CRLs</li></ul></li><li>• PKI</li><li>• Recovery agent</li><li>• Public key</li><li>• Private key</li><li>• Registration</li><li>• Key escrow</li><li>• Trust models</li></ul>

## Securing public-key management systems

### Explanation

Attacks on public-key systems typically target key-management systems rather than attempting to crack public-key encryptions. Therefore, these systems must be well protected. If a key-management system is compromised, the hacker can use the stolen keys to forge certificates and impersonate someone else. All trust associations are compromised as well.

The term *key life cycle* describes the stages a key goes through during its “life”: generation, distribution, storage, backup, and destruction. *Encryption key management* describes the systems used to manage those keys throughout their life cycle. If any phase of a key’s life is not managed properly, the entire security system can be compromised.

## Centralized and decentralized management

PKI (public key infrastructure) offers two broad models for generating and administering public keys: centralized and decentralized management.

*Centralized key-management systems* place all authority for key administration with a top-level entity. This could be a *certificate authority (CA)* within an organization or a trusted third-party entity. This model gives the administrator system-wide control over each aspect of key management. This model typically appears in scenarios where a hierarchical or single-authority *trust model* is implemented, as with X.509 certificates.

*Decentralized key-management systems* place responsibility for key management with the individual. The key and certificate are stored locally on the user’s system or some other device, and the user controls all key-management functions. Decentralized systems do not provide all the functionality of centralized systems. For example, if a user loses or damages the private key, there is no way to recover the private key or the encrypted information. This model typically appears in scenarios where the Web of Trust model is implemented, as with PGP certificates.

The decision to use centralized or decentralized systems depends on the size of the public-key infrastructure. If the number of keys that users retain on their key rings is limited, and the users are educated to properly protect their private keys, decentralized management works well. However, for a large organization, where thousands of keys might be generated, centralized management transfers the burden of private-key security from the end-users to a trained individual or team.

## **Setup and initialization**

The three main phases of the key life-cycle management process are: setup or initialization; administration of issued keys and certificates; and certificate cancellation and key history.

The setup or initialization process consists of:

- 1 Registration
- 2 Key pair generation
- 3 Certificate generation
- 4 Certificate dissemination

### **Registration**

The registration process starts when a user approaches the CA with a specific request for a certificate. After verifying the identity and credentials of the user, the CA registers the user. Depending on the certificate practice statement, certificate policy, and privileges associated with a given certificate, the identity verification process might require a physical appearance at the CA or submission of documented proof of identity.

### **Key pair generation**

Key pair generation involves creating matching private and public keys by using the same passphrase and different algorithms. Especially within the context of keys being used for non-repudiation services, the owner of the private key is entrusted with generating and storing such keys. In other scenarios, performance, usage, legalities, and algorithm specifications are the factors affecting the choice of location.

Multiple key pairs are often generated to perform different roles to support distinct services. A key pair can also be restricted by policy to certain roles based on usage factors such as type, quantity, category, service, and protocol. For instance, a certificate can be restricted to a particular function, such as signing or encryption. Multiple key pairs allow the CA to issue multiple certificates to the user for distinct functions.

### **Certificate generation**

The responsibility of creating certificates lies with the CA, regardless of where the key pair is generated. A certificate binds an entity's unique *distinguished name* (DN) and other identifying attributes to its public key. The entity DN can refer to an individual, an organization or organizational unit, or a resource (for example, a Web server or site).

The certificate policy governs the creation and issuance of certificates. The public key needs to be transmitted securely to the CA if it was generated elsewhere by a party other than the CA.

Requests for keys and certificates require secure transmission modes. The Internet Engineering Task Force (IETF) defines management and request-

message-format protocols for the purpose of transmitting public keys and certificates between key owners and CAs. Alternatives such as the Public Key Cryptography Standard also exist.

### **Certificate dissemination**

*Dissemination* involves securely making the certificate information available to a requester without too much difficulty. This is done through several techniques, including out-of-band and in-band distribution, publication, and centralized repositories with controlled access. Each method has its own benefits and drawbacks.

Depending on the client-side software, certificate usage, privacy, and operational considerations, the information requirements and dissemination methods vary. Several protocols are available that facilitate the secure dissemination of certificates and revocation information. Enterprise domains widely use LDAP (Lightweight Directory Access Protocol) repositories with appropriate security controls, along with in-band distribution through S/MIME-based (Secure Multimedia Mail Extensions) e-mail. This hybrid approach maximizes the benefits. Even within the repository model, several configurations—such as direct access, interdomain replication, guard mechanism, border, and shared repositories—are possible and often used.

### **Administration of issued keys and certificates**

The issued keys and certificates need to be administered properly after the initialization phase. The administrative phase involves the following:

- Key storage
- Certificate retrieval and validation
- Key archiving or escrow
- Key recovery

#### **Key storage**

After the key pair has been generated, the private key must be safely stored to protect it from being compromised, lost, or damaged. There are several key-storage methods, generally categorized as hardware or software storage.

*Hardware storage* refers to storing the private key on a hardware storage medium, such as a smart card, memory stick, USB device, PCMCIA card, or other such device. These devices can be physically carried on the person, enforce encryption of the private key, and often provide the added benefit of on-board encryption and decryption processing. The disadvantage of this method is that the storage medium can be easily lost or stolen.

*Software storage* refers to storing the private key in a computer file on the hard drive. The owner encrypts the private key by using a password or passphrase and stores the encrypted key in a restricted file. The user can enable auditing to track access to this file. Software storage is not considered reliable because if the file is restored to a different medium (such as a floppy disk or FAT drive), the encryption is removed.

#### **Certificate retrieval and validation**

As the name implies, *certificate retrieval* involves access to certificates for general signature verification and for encryption purposes. Retrieval is necessary as part of the normal encryption process for key management between the sender and the receiver. During verification, the certificate containing the

public key of a signed private key is retrieved and sent along with the signature or is made available on demand. It's imperative to have an easy and simple mechanism to retrieve certificates; otherwise, the complexity makes the system unusable.

Validation is performed to ensure that a certificate is issued by a trusted CA in accordance with appropriate policy restrictions and to ascertain the certificate's integrity and validity (whether it's expired or has been revoked) before its actual usage. In most cases, all of this is achieved transparently by the client software before cryptographic operations using the certificate are carried out.

**Note:** Attempts to use revoked certificates are a likely sign of an attempted break-in.

### **Key archiving and escrow**

*Key archiving* is the storage of keys and certificates for an extended period of time. It's an essential element of business continuity and disaster recovery planning, and it's the only solution that addresses lost keys and recovery of encrypted data. When used with additional services such as time stamping and notarization, a key-archive service meets audit requirements and handles the resolution of disputes.

Key archiving is typically undertaken by an organization's CA, a trusted third party, or, in some cases, the end entity (the user or computer that owns the key). Relying on the key's owner to manage archiving is generally unreliable due to the complexities involved. All private keys (current, expired, and revoked), with the exception of keys used for non-repudiation, are backed up to a key-archival server. The server requires strong physical security and at least the same security as the key-generating system.

*Key escrow* is a form of key archiving that allows third-party access without the cooperation of the subject (such as for law enforcement or other government agencies). Copies of the private keys are stored in an off-site repository called a *key escrow agency*. In 1995, the U.S. government required that all parties keep copies of the key pairs with a key escrow agency. Almost immediately, the government was questioned about its intentions for requiring key escrows. Eventually, the government dropped the requirement.

Key escrow has severe implications for individual privacy because control of the private keys is passed to a third party.

### **Key recovery**

Key recovery complements the key backup/escrow process. The recovery of lost, damaged, or archived keys allows access to encrypted messages and prevents permanent loss of business-critical information. This process is automated to minimize user intervention and errors.

Many archive systems use the *M of N Control* technique to ensure that no single administrator can abuse the recovery process. This access-control mechanism creates a PIN during the archive process and splits the number into two or more parts ( $N$  is the number of parts). Each part is given to a separate key-recovery agent (a person authorized to retrieve a user's private key). The recovery system can reconstruct the PIN only if  $M$  number of agents provide their individual PINs. For *M of N Control* to work,  $N$  must be greater than 1, and  $M$  must be less than or equal to  $N$  ( $N > 1$  and  $M \leq N$ ).

## Certificate cancellation and key history

The final phase in the process of key and certificate life-cycle management deals with cancellation procedures. This phase includes:

- Certificate expiration
- Certificate renewal
- Certificate revocation
- Certificate suspension
- Key destruction

### Certificate expiration

Certificate expiration occurs when the validity period of a certificate expires. Every certificate has a fixed lifetime, and expiration is a normal occurrence. Upon expiration, a certificate can be renewed if the keys are still valid and remain uncompromised, or are destroyed.

**Note:** Most applications will reject a certificate if it's in an expired state.

### Certificate renewal

Certificate renewal (also called a *certificate update*) is the process of issuing a new certificate with a new validity period. All that's required is that the certificate owner use the old key to sign a request for a new certificate. To facilitate smooth transition and prevent service interruption, the renewal should be initiated when a certificate approaches three-quarters of its intended lifetime (or 30 days before expiration).

Many certificate authorities merely repackage the old public key with the new certificate. This is a bad practice because the longer you keep the same key pair, the more insecure it will become over time. Ideally, a new key should be generated with each renewal.

### Certificate revocation

Certificate revocation implies the cancellation of a certificate before its expiration date. Certificate owners and PKI administrators (with the approval of the certificate owner) can revoke a key for any number of reasons; for instance, a company changes its ISP or moves to a new address, a contact leaves the company, or a private key is compromised or damaged.

The cancellation process is much easier than properly publishing and maintaining the revocation information after the fact. There are several ways in which the notification is accomplished. The primary method is through *certificate revocation lists* (CRLs). Essentially, CRLs are data structures containing revoked certificates. To maintain integrity and authenticity, CRLs are signed. Other methods include CRL distribution points, *certificate revocation trees* (CRTs), and Redirect/Referral CRLs.

Performance, timeliness, and scalability are some of the main factors that influence the revocation mechanisms. Instant-access methods through *Online Certificate Status Protocol* (OCSP) are also available. However, there is no guarantee that the real-time service is indeed providing an up-to-the-moment status. It's possible that the service might respond based on poorly updated databases. Additionally, many application implementations do not constantly check CRLs.

There are also exceptions for which such notification is deemed unnecessary. Two such exceptions involve short certificate lifetimes and single-entity approvals. In the former case, the accepted revocation delay might be more than the certificate lifetime, so the certificate might not require revocation at all. In the latter case, because requests are always approved by a single entity, it might not be necessary to publish the revocation separately.

The delay associated with the revocation requirement and subsequent notification is called *revocation delay*. Revocation delay must be clearly defined in the certificate policy because it determines how frequently or quickly the revocation information is broadcast and used for verification.

### **Certificate suspension**

If a certificate is not used for a period of time, the CA will eventually revoke it. To prevent this from taking place, a certificate owner will *suspend* the certificate, temporarily revoking it. Often, this option is used if an employee is on an extended leave of absence or a website is taken offline for renovations.

The suspension is published in the CRL or OCSP response with a status of “Certification Hold.” At the appropriate time, the suspension can be undone.

### **Key destruction**

CAs typically destroy certificates and any keys associated with them when certificates expire or get revoked. Another significant event warranting key destruction occurs before a certificate server or key archival server is sold or recycled. Key destruction is usually accomplished by overwriting the key data. One common method is *zeroization*, which overwrites the data with zeros.

## **Administrative responsibilities**

Setting up an enterprise PKI is a time-consuming and extremely complex task with enormous demands on financial, human, hardware, and software resources. It’s very important to understand the concepts, processes, and products involved, and to ask pertinent questions right from the beginning. In addition to basic support, training, and documentation issues, the areas that need to be explored in detail include the following:

- Support for standards, protocols, and third-party applications
- Issues related to cross-certification, interoperability, and trust models
- Multiple key pairs and key-pair uses
- Methods to PKI-enable applications and client-side software availability
- Impact on end-users for key backup, key or certificate update, and non-repudiation services
- Performance, scalability, and flexibility issues regarding distribution, retrieval, and revocation systems
- Physical access control to facilities

The security awareness in the IT industry has grown considerably, and the business community is beginning to understand the seriousness of security implications and the benefits of PKI. With the growth in e-commerce, PKI deployments are expected to continue to grow significantly over the next couple of years, despite questions about standards, policies, products, legalities, return on investment, and the technology itself.

Do it!

## A-1: Understanding the life cycle and management of certificates – Model Answers on the VLC

### Exercise

1 What is the key life cycle?

2 What is a centralized key-management system?

3 Match each phase of key management below with its definition:

Certificate generation

Certificate renewal

Certificate revocation

Certificate validation

Key archival

Key escrow

Key pair generation

Key recovery

Key storage

Registration

A browser requests a signature verification for a certificate.

A certificate is canceled before its expiration date.

A certificate is reissued with a new validity period.

A key is retrieved from archive due to loss or damage of the original.

Matching private and public keys are created.

A private key is safely stored on a hardware or software medium.

The CA binds the requestor's identifying attributes to its public key.

The key is stored for an extended period of time.

The key is stored in an off-site repository for third-party access.

The user approaches the CA with a specific request for a certificate.