Cryptography

Key Management & Distribution
Public Key Infrastructure (PKI)

M S Vilkhu

02 Dec 2024(C1/C3/C5)

Topic to cover

- Key Management and Distribution, PKI

Topics

- Key Management and Distribution
- X.509
- PKI

Learning Objectives

After studying this chapter, you should be able to:

- Discuss the **concept of a key hierarchy**.
- Understand the **issues involved** in using asymmetric encryption to distribute symmetric keys.
- Present an overview of **approaches to public-key** distribution and analyze the risks involved in various approaches.
- List and explain the elements in an **X.509 certificate**.
- Present an overview of **public-key infrastructure** concepts.

Introduction

- The topics of **cryptographic key management** and **cryptographic key distribution** are **complex**, **involving cryptographic**, **protocol**, and **management** considerations.
- the issues involved and a broad survey of the various aspects of key management and distribution.
- For more information, the place to start is the three-volume NIST SP 800-57,
- NIST SP 800-57 **provides cryptographic key management guidance**. It consists of three parts.
- Part 1 provides **general guidance** and best practices for the management of cryptographic keying material.
- Part 2 provides guidance on policy and security planning requirements for U.S. government agencies.

Symmetric Key Distribution Using Symmetric Encryption

- For **symmetric encryption** to work, the two parties to an exchange must share the same key, and that key must be protected from access by others.
- Furthermore, **frequent key changes** are usually **desirable** to limit the amount of data compromised if an attacker learns the key.
- Therefore, the **strength of any cryptographic system** rests with the **key distribution technique,** a term that refers to the means of delivering a key to two parties who wish to exchange data **without allowing others** to see the key.
- For two parties A and B, key distribution can be achieved in a number of ways, as follows:

Symmetric Key Distribution Using Symmetric Encryption

- For two parties A and B, key distribution can be achieved in a number of ways, as follows:

1. A can select a key and **physically deliver** it to B.
2. A **third party can select** the key and **physically deliver it to A and B.**
3. If A and B have previously and **recently used a key**, one party can **transmit** the **new key to the other,** encrypted using the old key.
4. If A and B each has an **encrypted connection to a third party C**, C can deliver a **key on the encrypted links to A and B**.

Symmetric Key Distribution Using Symmetric Encryption

- For two parties A and B, key distribution can be achieved in a number of ways, as follows:

1. A can select a key and **physically deliver** it to B.
2. A **third party can select** the key and **physically deliver it to A and B.**
3. If A and B have previously and **recently used a key**, one party can **transmit** the **new key to the other,** encrypted using the **old key.**
4. If A and B each has an **encrypted connection to a third party C**, C can deliver a **key on the encrypted links to A and B**.

option 3 is a possibility for either link encryption or end-to-end encryption, but if an attacker ever succeeds in gaining access to one key, then all subsequent keys will be revealed. Furthermore, the **initial distribution of potentially millions** of keys still must be made

For **end-to-end encryption**, some variation on **option 4 has been widely adopted**. In this scheme, a **key distribution center** is responsible for distributing keys to pairs of users (hosts, processes, applications) as needed.

Each user must share a **unique key with the key distribution center** for purposes of **key distribution**

Symmetric Key Distribution Using Symmetric Encryption

- For two parties A and B, key distribution can be achieved in a number of ways, as follows:

1. A can select a key and **physically deliver** it to B.
2. A **third party can select** the key and **physically deliver it to A and B.**
3. If A and B have previously and **recently used a key**, one party can **transmit** the **new key to the other,** encrypted using the **old key.**
4. If A and B each has an **encrypted connection to a third party C**, C can deliver a **key on the encrypted links to A and B**.

Option 4. For **end-to-end encryption**, some variation on **option 4 has been widely adopted**. In this scheme, a **key distribution center** is responsible for distributing keys to pairs of users (hosts, processes, applications) as needed.

Each user must share a **unique key with the key distribution center** for purposes of **key distribution**

Key Distribution

- If **end-to-end encryption** is done at a **network or IP level**, key is needed for each **pair of hosts** on the network that wish to communicate.
- Thus, if there are N hosts, the number of required keys is **[N(N - 1)]/2**.
- If encryption is done at the **application level**, key is needed for every **pair of users** or processes
- Thus, a network may have **hundreds of hosts** but **thousands of users** and processes
- A network using **node-level encryption with 1000** nodes would conceivably need to distribute as many as **half a million keys**.
- If that same network supported **10,000 applications**, then as many as **50 million keys** may be required for application-level encryption.

Symmetric Key Distribution Using Symmetric Encryption

- The use of a **key distribution center** is based on the use of a **hierarchy of keys**. At a minimum, **two levels of keys are used** (Figure 14.2).
- **Communication** between end systems is **encrypted** using a **temporary key,** often referred to as a **session key -** the session key is used for the duration of a **logical connection**, and then **discarded**.
- Each **session key** is obtained from the **key distribution center** over the **same networking facilities** used for end-user communication.
- session keys are **transmitted in encrypted form**, using a **master key** that is shared by the key distribution center and an end system or user.
- Master key must be distributed in physical form

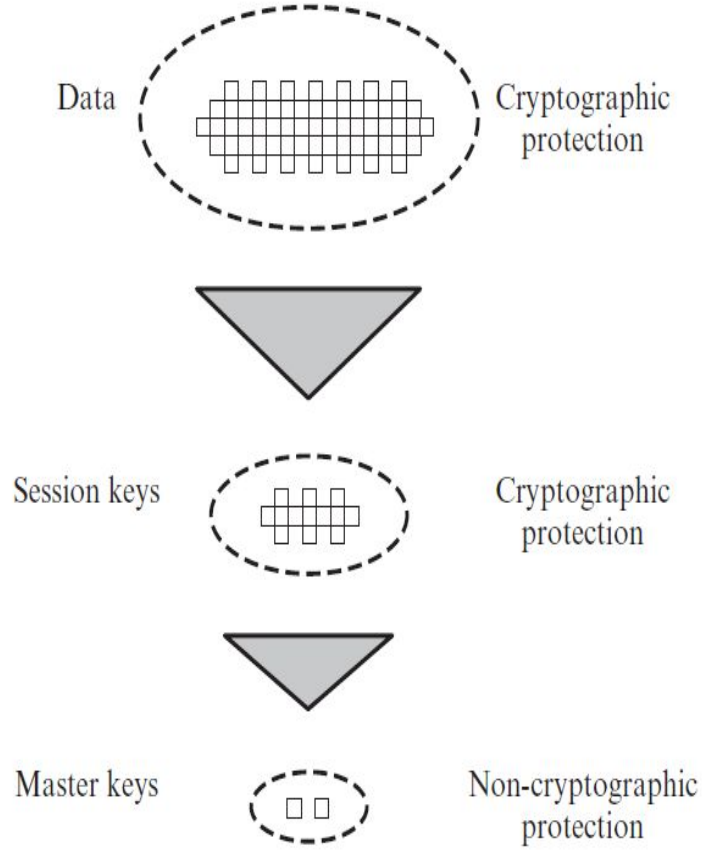

Figure 14.2 The Use of a Key Hierarchy

Key Distribution Scale of the problem

- In **key distribution**, the **scale of the problem** relates to the number of unique keys that must be managed as the number of participants or communication pairs increases. This scaling issue arises due to the need for secure communication between multiple entities

Key Distribution Scale of the problem

- **In key distribution, the scale of the problem**
- **Key Distribution Problem Scale**

1. **For Symmetric Key Cryptography**:
    1. Each pair of participants requires a unique shared key.
    2. If there are nnn participants, the number of unique keys K required is:

       $K=n\times(n-1)/2$

    3. This is a **quadratic scaling** problem ($O(n^2)$).

- **Example**:
- If 100 participants are in the network, the total keys needed are:

  $K=100\times99/\ 2=4950$

Key Distribution Scale of the problem

- **In key distribution, the scale of the problem**

2. For Public Key Cryptography:

1. Each participant needs only a single key pair (public and private keys).
2. For **n participants**, the number of keys required is: K=2×n
3. This is a **linear scaling** problem (O(n)).

- **Example**:
- For 100 participants, the total number of keys required is:
- K=2×100=200

Key Distribution Scale of the problem

- **In key distribution, the scale of the problem**
- **Challenges in Key Distribution**

1. **Symmetric Key Systems**:
    1. Management of large numbers of keys becomes impractical as **n** grows.
    2. Secure distribution of shared keys is a **major challenge**.
2. **Public Key Systems**:
    1. While **scaling is better**, **ensuring the authenticity of public keys** (via **certificates or trusted authorities**) is critical.

Key Distribution Scale of the problem

- **In key distribution, the scale of the problem**
- **Solution Approaches**
  - **Key Distribution Centers (KDCs):** Centralized systems for securely distributing symmetric keys.
  - **Public Key Infrastructure (PKI):** A framework for managing public and private key pairs efficiently.

Key Distribution Center (KDC)

Key Distribution Center (KDC):

- **Key Distribution Center (KDC): How It Works**
- A **Key Distribution Center (KDC)** is a trusted third party in symmetric key cryptography that facilitates secure communication by distributing shared keys to users. It eliminates the need for every pair of users to manage their own unique shared key, addressing the scalability issue in large networks.

Step-by-Step Process of KDC Operation

1. Initialization

- Each user in the system **registers with the KDC**.
- The KDC establishes a unique symmetric key, called the **Master Key**, with each user. This key is securely shared during the registration process.

Key Distribution Center (KDC):

- **Key Distribution Center (KDC): How It Works**

2. Request for Communication

- **Step 2.1**: A user (let's call them **Alice**) wants to communicate securely with another user (**Bob**).
- **Step 2.2**: Alice sends a request to the KDC, encrypted with her Master Key. The request includes:
  - Alice's identity.
  - Bob's identity.
  - A nonce (random value) for verification.

3. Key Generation

- The KDC generates a **Session Key**, a temporary key for Alice and Bob to use during their communication.

Key Distribution Center (KDC):

- **Key Distribution Center (KDC): How It Works**

4. Key Distribution

- The KDC sends two encrypted messages back to Alice:
  - **Message 1**: Contains the **Session Key** and Bob's identity, encrypted with **Alice's Master Key**.
  - **Message 2**: Contains the **Session Key** and Alice's identity, encrypted with **Bob's Master Key** (this is intended for Bob).

5. Key Forwarding

- Alice decrypts **Message 1** using her Master Key and retrieves the Session Key.
- Alice forwards **Message 2** to Bob.

Key Distribution Center (KDC):

Key Distribution Center (KDC): How It Works

6. Session Key Verification

- Bob decrypts **Message 2** using his Master Key and retrieves the Session Key.
- Now both Alice and Bob have the **same Session Key**.

7. Secure Communication

- Alice and Bob use the **Session Key** to encrypt and decrypt their messages, ensuring **confidentiality** and security during communication.

Key Distribution Center (KDC):

Summary of Steps

1. **Registration**: KDC shares Master Keys with users.
2. **Request**: Alice requests a session key to communicate with Bob.
3. **Session Key Creation**: KDC generates a new Session Key.
4. **Key Distribution**: KDC shares Session Key securely with Alice and Bob.
5. **Communication**: Alice and Bob use the Session Key for encrypted communication.

Benefits of Using KDC

1. **Reduced Key Management Complexity**: Each user only maintains one Master Key with the KDC.
2. **Scalability**: Facilitates secure communication for large networks without requiring $n \times (n-1)/2$ keys
3. **Dynamic Key Assignment**: New Session Keys can be generated for every communication, improving security.

Key Distribution Center (KDC):

Challenges

1. **Single Point of Failure**: If the KDC is compromised, all communications in the network are at risk.
2. **Overhead**: The KDC can become a **bottleneck** in **large networks** due to the high number of requests.

3. By acting as a trusted intermediary, the KDC ensures efficient and secure key distribution, enabling secure symmetric communication in complex systems.

X.509 Certificates

- ITU-T recommendation X.509 is part of the X.500 series of recommendations that define a **directory service.**
- The **directory** is, in effect, a server or distributed set of servers that **maintains a database of information about users**.
- The information includes a **mapping from user name to network address**, as well as other attributes and information about the users.
- X.509 **defines a framework** for the provision of **authentication services by the X.500 directory** to its users.
- The **directory** may serve as a **repository of public-key certificates**.
- Each certificate contains the **public key of a user** and is **signed with the private key of a trusted certification authority**.
- In addition, X.509 defines **alternative authentication protocols** based on the use of public-key certificates.

X.509 Certificates

- **X.509 Standard: An Overview**
- **X.509** is an internationally recognized standard for **Public Key Infrastructure (PKI)** that defines the format for public key certificates. It is widely used in securing internet communications, such as HTTPS, email encryption, and digital signatures.
- **Key Features of X.509**

1. **Certificate-Based Authentication**: Enables secure authentication using digital certificates.
2. **Public Key Infrastructure (PKI)**: Supports secure distribution and management of public keys.
3. **Hierarchical Trust Model**: Uses Certificate Authorities (CAs) to issue and validate certificates.

X.509 Certificates

Structure of an X.509 Certificate

An X.509 certificate contains the following fields:

1. **Version**: Indicates the version of the X.509 standard (commonly version 3).
2. **Serial Number**: A unique number assigned by the issuing Certificate Authority.
3. **Signature Algorithm**: The algorithm used by the CA to sign the certificate (e.g., SHA256 with RSA).
4. **Issuer**: The identity of the Certificate Authority that issued the certificate.
5. **Validity Period**: Defines the start and end dates for the certificate's validity.
6. **Subject**: The entity (e.g., a user, organization, or server) to which the certificate belongs.
7. **Subject Public Key Info**: Contains the public key and its algorithm.
8. **Extensions (Version 3 Only)**:
    1. **Key Usage**: Specifies the purpose of the certificate (e.g., signing, encryption).
    2. **Subject Alternative Name (SAN)**: Lists additional identities for the subject (e.g., multiple domain names).
    3. **CRL Distribution Points**: Provides information for accessing the Certificate Revocation List.

X.509 Certificates

Applications of X.509

1. **TLS/SSL Certificates**: Secures HTTPS connections by encrypting web traffic.
2. **Email Encryption**: Protects email communications using standards like S/MIME.
3. **Code Signing**: Verifies the authenticity and integrity of software.
4. **User Authentication**: Enables secure login using client certificates.
5. **VPN and Wireless Security**: Used for authentication in secure networks.

Purpose of X.509

- The primary purpose of **X.509** is to provide a standard framework for creating, managing, and verifying **digital certificates** that enable secure communication and authentication within a **Public Key Infrastructure (PKI)**.

The X.509 standard is a cornerstone of internet security, enabling robust encryption and trust mechanisms across diverse applications.

Public-Key Infrastructure

Public-Key Infrastructure

- **Public Key Infrastructure (PKI)**
- **Public Key Infrastructure (PKI)** is a framework of policies, processes, technologies, and standards **used to manage digital certificates** and **public-private key pairs**.
- PKI **enables secure electronic communication**, authentication, and data integrity across digital systems.

Public-Key Infrastructure

- RFC 4949 (Internet Security Glossary) defines **public-key infrastructure (PKI)** as the set of hardware, software, people, policies, and procedures needed to **create, manage, store, distribute, and revoke** digital certificates based on **asymmetric cryptography**.
- The **principal objective** for developing a PKI is to **enable secure, convenient, and efficient acquisition of public keys**.
- 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.
- This section describes the PKIX model.

Public-Key Infrastructure

- PKIX model.

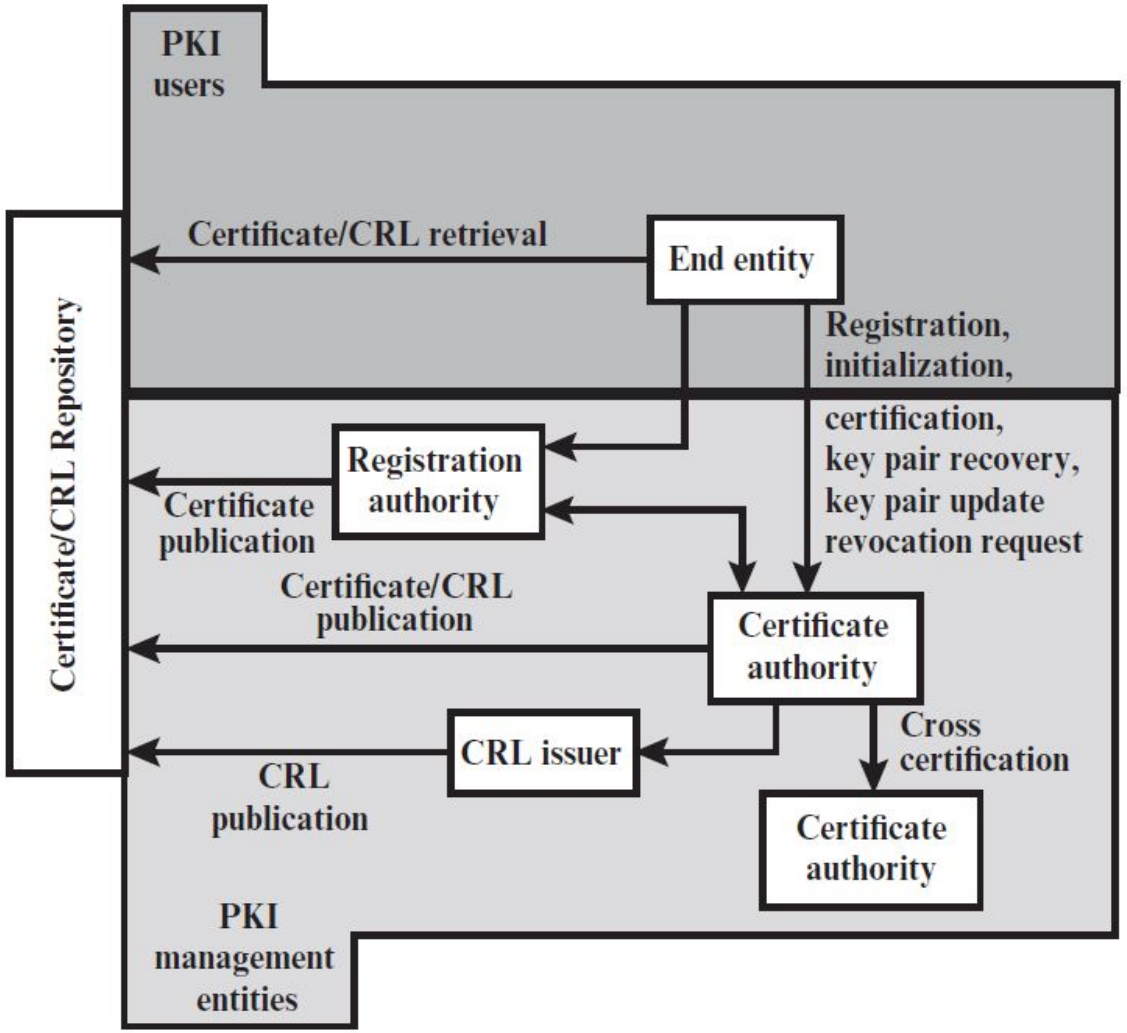

Figure 14.17 PKIX Architectural Model

Public-Key Infrastructure

- PKIX model.

End entity: 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 **issuer** of certificates and (usually) **certificate revocation lists** (CRLs). It may also support a variety of administrative functions, although these are often delegated to one or more Registration Authorities.

Registration authority (RA): 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.

CRL issuer: An optional component that a CA can delegate to publish CRLs.

Repository: A generic term used to denote any method for storing certificates and CRLs so that they can be retrieved by end entities.

Public-Key Infrastructure

Core Components of PKI

1. **Certificate Authority (CA)**:
    1. A trusted entity responsible for issuing, revoking, and managing digital certificates.
    2. Validates the identity of certificate requestors (users, servers, devices) before issuing certificates.
2. **Registration Authority (RA)**:
    1. Acts as an intermediary between users and the CA.
    2. Handles identity verification and forwards certificate requests to the CA.
3. **Digital Certificates**:
    1. Cryptographic files containing an entity's public key and identity information, signed by a CA.
    2. Common standard: X.509 certificates.
4. **Key Pairs**:
    1. **Public Key**: Shared openly and used for encryption or signature verification.
    2. **Private Key**: Kept secret by the owner and used for decryption or signing.
5. **Certificate Revocation List (CRL)** and **Online Certificate Status Protocol (OCSP)**:
    1. CRL: A list of revoked certificates no longer considered valid.
    2. OCSP: A protocol for real-time verification of certificate validity.
6. **PKI Repository**:
    1. A database or directory where public keys and certificates are stored for users to access.

Public-Key Infrastructure

- **Functions of PKI**

1. **Authentication**:
    1. Confirms the identity of users, systems, or devices using digital certificates.
    2. Example: HTTPS websites use SSL/TLS certificates to authenticate servers.
2. **Confidentiality**:
    1. Encrypts data to protect it from unauthorized access.
    2. Example: Emails secured with S/MIME.
3. **Integrity**:
    1. Ensures data has not been tampered with during transmission or storage.
    2. Example: Digital signatures validate file integrity.
4. **Non-Repudiation**:
    1. Provides proof that a specific user performed an action (e.g., sending a message or signing a document).
    2. Example: Digitally signed contracts.
5. **Key Management**:
    1. Manages the lifecycle of public-private key pairs, including generation, distribution, renewal, and revocation

Public-Key Infrastructure

- **Applications of PKI**

1. **Website Security**:
    1. Secures websites with SSL/TLS certificates, enabling HTTPS.
2. **Email Security**:
    1. Provides encryption and signing via S/MIME.
3. **Digital Signatures**:
    1. Used in e-documents to verify authenticity and integrity.
4. **Network Security**:
    1. Secures VPNs and enterprise network authentication (e.g., 802.1X).
5. **IoT Device Authentication**:
    1. Verifies identities of IoT devices to prevent unauthorized access.

Public-Key Infrastructure

- **Benefits of PKI**

1. **Scalability**: Suitable for managing security in large, distributed networks.
2. **Interoperability**: Standardized protocols enable integration across platforms.
3. **Enhanced Security**: Ensures strong encryption and trust mechanisms.
4. **Automation**: Streamlines key and certificate lifecycle management.

Thank You