Public Key Infrastructure (PKI) - Comprehensive Exam Notes

1. Introduction to PKI

What is PKI?

Public Key Infrastructure (PKI) is a comprehensive system that provides security for communications over insecure networks like the Internet. It combines personnel, policies, protocols, hardware, software, and cryptographic tools to enable secure communications.

Core Components of PKI:

- **Public-key cryptography**
- **Certification authorities (CAs)**
- **Digital certificates**
- **Registration authorities (RAs)**
- **Certificate repositories**
- **Revocation mechanisms**

---

2. Key Authentication Problem

The Challenge

- **Public Key Distribution**: How do you trust the source of a public key?
- **Potential for attacks**: Man-in-the-middle attacks, key substitution attacks
- **Trust establishment**: Need to verify authenticity of public keys

Solution Approach

- Replace direct trust in keys with trust in a trusted third party
- Use **Certification Authorities (CAs)** to vouch for key authenticity
- **Distribution of Trust**: Your trust in a key is replaced by trust in the CA

---

3. Certification Authorities (CAs)

Purpose

- Solve the problem of distributing public keys so recipients know they are valid
- Act as trusted third parties that vouch for key authenticity

- Keys are signed by the state, corporation, or someone you trust

How CAs Work

1. All users choose a CA and obtain the CA's public signature verification key through trusted means
2. **Critical requirement**: All users must trust the CA
3. CA signs user certificates to establish authenticity

CA Responsibilities

- Verify user identities before issuing certificates
- Sign certificates with their private key
- Maintain certificate repositories
- Handle certificate revocation

---

4. Digital Certificates

Certificate Creation Process

1. **User submission**: User submits their public key to the CA
2. **Data concatenation**: CA concatenates user name, user public key, expiry date, and other metadata
3. **Signature generation**: CA generates a signature on this data string using their private key
4. **Certificate creation**: The combination of data string and signature forms the **Public Key Certificate**
5. **Certificate distribution**: Certificate is sent back to the user

Certificate Verification

- Anyone with the CA's public key can verify a user's certificate
- Verification confirms the authenticity of the user's public key
- Enables trusted communication without prior key exchange

Certificate Storage

- Certificates are stored in **repositories** for easy access
- Certificate repositories may be separated from the CA that generates them
- **Important**: Certificates do NOT need to be stored securely (they're already signed)

---

5. Cross-Certification

The Problem

When multiple CAs exist, a user may not have a trusted copy of the CA's public key needed to verify another user's certificate.

Solution: Cross-Certificates

- One CA's public key is signed by another CA
- Creates a chain of trust between different CAs
- Enables interoperability between different PKI domains

Cross-Certification Example

Scenario: Alice trusts CA1, but needs to verify Bob's certificate signed by CA2

1. Alice obtains Bob's public key (signed by CA2's private key)
2. Alice obtains CA2's public key (signed by CA1's private key)
3. Alice first verifies CA2's cross-certificate using CA1's public key
4. Alice then verifies Bob's certificate using CA2's public key
5. **Result**: Alice can now trust Bob's public key

---

6. Registration Authorities (RAs)

Purpose

- Establish user identity before certificate issuance
- Can be co-implemented with the CA or separate entity
- Act as the front-end for certificate requests

RA Responsibilities

1. **Identity verification**: Establish the identity of certificate requesters
2. **Key ownership proof**: Verify user knows the private key corresponding to the public key being certified
3. **Key generation verification**: Ensure the key was generated "correctly"
4. **Request processing**: Handle certificate requests and forward to CA

---

7. Certificate Revocation

When Revocation is Needed

- **Key compromise**: Third party gains knowledge of the private key
- **Certificate misuse**: Certificate is being used inappropriately
- **Change in user status**: User leaves organization, role changes
- **CA compromise**: CA's private key is compromised

Revocation Process

1. **Identification**: Determine that a certificate should be revoked
2. **Notification**: CA must inform all users that the certificate is no longer valid
3. **Distribution**: Revocation information must be distributed to all relying parties

---

8. Certificate Revocation List (CRL)

What is a CRL?

- A list of serial numbers of all certificates revoked by a particular CA
- Signed by the CA concerned to ensure authenticity
- Updated regularly to reflect new revocations

How CRLs Work

1. **Creation**: CA maintains a list of revoked certificate serial numbers
2. **Signing**: CA signs the CRL with their private key
3. **Distribution**: CRL is made available to all users
4. **Verification**: Users must ensure they have the latest CRL before trusting certificates

CRL Analogy

Similar to the list of bad credit card numbers that used to be kept next to tellers in supermarkets.

CRL Limitations

- **Scalability**: Can become very large for CAs with many revoked certificates
- **Timeliness**: Updates may not be immediate
- **Availability**: Must be accessible when needed

---

9. Trusted Third Parties (TTPs)

Definition

CAs and RAs are examples of third parties that users must trust in some way. These entities are generically referred to as **Trusted Third Parties (TTPs)**.

TTP Characteristics

- Users must place trust in their integrity and competence
- May have knowledge of user secret keys in some network systems
- Critical components in the security chain
- Single points of failure if compromised

---

10. PKI Examples and Standards

X.509/PKIX

- **X.509** defines the structure for public key certificates
- **PKIX** (Public Key Infrastructure X.509) extends X.509 for Internet use
- Most widely used certificate format
- Defines certificate fields and validation rules

PGP (Pretty Good Privacy)

- Uses a "Web of Trust" model
- No centralized CA
- Users directly sign each other's keys
- Trust is established through networks of personal relationships

---

11. X.509 Certificate Structure

Certificate Fields

- **Version**: Determines the certificate format
- **Serial Number**: Unique identifier given by CA
- **Algorithm Identifier**: Refers to signature algorithm and parameters
- **Issuer**: Name of the CA
- **Subject**: Name of the certificate holder
- **Period of Validity**:
  - Not before date

    - Not after date
- **Subject's Public Key**:
    - Algorithm
    - Parameters
    - Public Key
- **Signature**: Digital signature by the CA

Certificate Hierarchy

- CAs are connected in a tree structure
- Each CA issues certificates for those beneath it
- Creates a hierarchical trust model

---

12. PKI Trust Models

1. Monopoly Model

- **Structure**: Only one universally trusted CA
- **Advantages**: Simple, clear trust path
- **Disadvantages**: Very hard to implement in practice, single point of failure
- **Usage**: Rarely used due to practical limitations

2. Oligarchy Model

- **Structure**: Multiple trusted CAs configured as trust anchors
- **Implementation**: Commonly used in web browsers
- **Advantages**: Redundancy, choice of CAs
- **Disadvantages**: Users must trust multiple CAs
- **Usage**: Default model for most web browsers

3. Delegated CAs (Hierarchical)

- **Structure**: Root CA can issue certificates to subordinate CAs
- **Features**:
    - CA vouches for other CAs' public keys
    - CA vouches for other CAs' trustworthiness
    - Creates a chain of certificates visible to users

- **Advantages**: Scalability, distributed trust
- **Disadvantages**: Complex trust chains, multiple points of failure

4. Anarchy Model (Web of Trust)

- **Structure**: No centralized CA
- **Implementation**: Used by PGP
- **Features**:
  - Each user configures their own trust anchors
  - Direct key signing between users
  - Trust based on personal relationships
- **Advantages**: No central authority, user control
- **Disadvantages**: Difficult to scale, complex trust decisions

---

13. Exam Tips and Key Concepts

Critical Concepts to Remember

1. **Trust Transfer**: PKI transfers trust from keys to CAs
2. **Certificate Verification**: Always requires CA's public key
3. **Cross-Certification**: Enables trust between different PKI domains
4. **Revocation**: Critical for maintaining security when keys are compromised
5. **Trust Models**: Each has different advantages and use cases

Common Exam Questions

1. **Explain the certificate verification process**
2. **Compare different PKI trust models**
3. **Describe how cross-certification works**
4. **Explain the role of CRLs in PKI**
5. **Identify components of X.509 certificates**

Security Considerations

- **Single Points of Failure**: CAs are critical components
- **Key Compromise**: Both user and CA key compromise scenarios
- **Trust Assumptions**: Users must trust CAs completely

- **Revocation Timeliness**: Delayed revocation can compromise security

---

14. Summary

PKI provides a comprehensive framework for secure communications by:

- Establishing trust in public keys through CAs
- Providing certificate-based authentication
- Enabling secure communication without prior key exchange
- Supporting multiple trust models for different environments
- Handling key lifecycle management including revocation

The success of PKI depends on proper implementation of all components: CAs, RAs, certificates, repositories, and revocation mechanisms. Understanding the trust relationships and security assumptions is crucial for both implementation and exam success.