Public Key Infrastructure

- Definition and Description
- Functions
- Components
- Certificates

PKI Services

- Security Between Strangers
  - Encryption
  - Integrity
  - Non-repudiation
  - Key establishment

What Does A PKI Do?

- Accurately Deliver Stored, Authenticated Public Keys to Requestors
- It is an Infrastructure for Establishing and Maintaining an Address Book

PKI Definition #1

"A PKI is a pervasive security infrastructure whose services are implemented and delivered using public-key concepts and techniques".

-Adams/Lloyd

Text PKI Definition

- Certificates
- Repositories
- Method of evaluating certificates
- Revocation method
PKI Functions

- Create the Directory
- Update the Directory
- Register Users
- Deliver Keys on Request
- Keep the Directory Available
- Revoke Bad or Canceled Public Keys

Generic Parties to PKI

1. Client
2. Trusted Registrar
3. Database Administrator

  - Store
  - Issue
  - Verify

Generic Parties to PKI

1. Client
   - Roles
     - Public Key Bearer
       - Authenticated to PKI
       - Key granted by PKI
     - Public Key Requestor
       - Seeking another party's public key
       - This is the quintessential PKI user.

2. Trusted Registrar
   - Establishes absolute identify of Public Key Bearers
   - Binds the bearer's public key to their identity

Generic Parties to PKI

3. Certificate Authority
   - Acts as the Database Administrator
   - Stores Public Keys for Bearers
   - Establishes Certificates for Bearers
   - Provides Certificates to Requestors
   - Verifies Public Keys for Requestors

A Typical PKI Use

1. Client registers with an RA (once)
   - RA creates and distributes client identity to CAs
2. Client requests key from CA(s)
   - CA creates & issues key and certificate
3. Client uses the key and certificate to transmit sensitive data
PKI for Authorization

1. Client registers with an RA (once)
   - RA creates and distributes client identity to CAs
2. Client requests public key from CA(s)
   - CA creates & issues key and certificate
3. Client requests service from an application and presents the certificate
4. Application verifies the certificate (via the CA's key) and provides the service

Certificates

- Binds a public key to an identity
- Usual method is to use a data structure that contains at least following three elements:
  1. The public key
  2. The identity of the public key owner
  3. A signature over the above two fields

A Simple But Strong Certificate

<table>
<thead>
<tr>
<th>Alice</th>
<th>[Alice,PKₐ]ₚₖ⁻¹</th>
<th>Charlie</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Identity of the public key holder</td>
<td>The Encrypted Signature</td>
</tr>
</tbody>
</table>

In order to extract Alice's verified public key, the certificate holder must know Charlie's public key.

Certificate Standards

- X.509
- Simple Public Key Infrastructure (SPKI)
- Pretty Good Privacy (PGP)
- Attribute Certificates

X.509 Certificate

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Certificate Version (e.g. X.509_v3)</td>
</tr>
<tr>
<td>Serial Number</td>
<td>Unique Identifier for the Certificate</td>
</tr>
<tr>
<td>Signature</td>
<td>ID of the Algorithm Used to Sign the Certificate</td>
</tr>
<tr>
<td>Issuer</td>
<td>Unique Name of the Certificate Issuer</td>
</tr>
<tr>
<td>Validity</td>
<td>Time Period of Certificate Validity</td>
</tr>
<tr>
<td>Subject</td>
<td>Unique Name of the Certificate Owner</td>
</tr>
<tr>
<td>Subject Public Key Info</td>
<td>Public Key and Algorithm ID of the Owner</td>
</tr>
<tr>
<td>Issuer Unique ID</td>
<td>Optional Unique ID of the Certificate Issuer</td>
</tr>
<tr>
<td>Subject Unique ID</td>
<td>Optional Unique ID of the Certificate Owner</td>
</tr>
<tr>
<td>Extensions</td>
<td>Optional Extensions</td>
</tr>
</tbody>
</table>

Optional Extensions

1. Authority Key Identifier
   ID of the key that should be used to verify this certificate in case the CA has multiple signing keys
2. Subject Key Identifier
   ID of the public key contained in this certificate in case the owner has multiple public keys
3. Key Usage
   Bit string to identify or restrict usage of the key, e.g. signature only, encipher only, decipher only, etc.
4. Extended Key Usage
   Sequence of one or more OIDs that define specific usage of the public key in the certificate (see RFC 2459)
### Optional Extensions (cont)

<table>
<thead>
<tr>
<th>Extension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5. CRL Distribution Point</td>
<td>Location of the CRL partition where the CRL resides</td>
</tr>
<tr>
<td>6. Private Key Usage Period</td>
<td>Time window that the PRIVATE key associated with this certificate may be used (Valid field in certificate deals with the Public key validity period)</td>
</tr>
<tr>
<td>7. Certificate policies</td>
<td>One or more policy OIDs associated with the issuance and use of this certificate</td>
</tr>
<tr>
<td>8. Policy Mappings</td>
<td>One or more policy OIDs equivalencies between two CA domains (see RFC 2459)</td>
</tr>
<tr>
<td>9. Subject Alternative Name</td>
<td>Alternate owner name, e.g. email address, IP Addr</td>
</tr>
<tr>
<td>10. Issuer Alternative Name</td>
<td>Alternate issuer name, e.g. email address, IP Addr</td>
</tr>
<tr>
<td>11. Subject Directory Attributes</td>
<td>Used most to convey access/privilege information for the certificate owner, which is risky.</td>
</tr>
<tr>
<td>12. Basic constraints</td>
<td>Indicates whether this is a CA certificate (T or F)</td>
</tr>
<tr>
<td>13. Path Length Constraint</td>
<td>Indicates the maximum number of CA-certificates that may follow this certificate in a certificate chain.</td>
</tr>
<tr>
<td>14. Name Constraint</td>
<td>In CA-certificates, indicates required or excluded subtree names for certificate chain.</td>
</tr>
<tr>
<td>15. Policy Constraints</td>
<td>In CA-certificates, indicates required policy OIDs.</td>
</tr>
</tbody>
</table>

### Simple Public Key Infrastructure

- Develop an Authorization Certificate that is simple to:
  - Understand
  - Implement
  - Use
- Purpose
  - Convey Permissions
  - Delegate Permissions

### OpenPGP

- Trust Model: Web of Trust
- Excellent Internet model
- Not compatible with X.509
- Not suitable for centralized corporate use

### Certificate Revocation

1. Adds significant complexity in infrastructure
   - Certificates are fire and forget
   - Lists must be maintained
2. Adds complexity in implementation
   - It adds a second step in certificate verification
3. Time
   - Revocation will likely NEVER be instantaneous
   - If it is not instantaneous, how fast is fast enough?
# Certificate Revocation Options

1. Certificate Revocation Lists (CRLs)
2. Authority Revocation Lists
3. CRL Distribution Points (Partitioned CRLs)
4. Delta CRLs
5. Indirect CRLs
6. Enhanced CRL Distribution Points and Redirect CRLs
7. Certificate Revocation Trees (CRTs)

---

## Certificate Revocation Lists

- **Fields**
  - Version
  - Signature
  - Issuer
  - Date/time issued
  - Date/time expires
  - List of revoked certificates
  - Extensions

- **Extensions**
  - Reason
  - Certificate Issuer
  - Hold Instruction Code
  - Invalidity Date

## Authority Revocation Lists

- CRL for CA certificate revocation
  - Should be very rarely used

## CRL Distribution Points

- CRL for CA domains
  - Subdivides info about certificates issued by one CA into multiple CRLs

## Enhanced CRL Distribution Points

- Dynamic CRL partitioning
- Allows reassociation of certificate & CRL after the certificate is issued.
- Can be accomplished by having the originally associated CRL redirect inquiries

---

## Certificate Revocation Lists

- Signed data structure
- Contains a list of revoked certificates
- CR signer is usually the same entity that signed the certificate that the corresponding CRL revokes
- CRL is a signed list, not a database; cannot be updated
Delta CRLs

- Used to enhance timeliness without significantly impacting performance
- Allows incremental postings of CR information
- Each Delta version contains ALL changes to the base CRL, so when a new Delta is received, old Deltas can be discarded

Indirect CRLs

- Enables single CRL for multiple CAs
- No specification for how to accumulate CR info from CAs to the issuing CA

Certificate Revocation Trees

- Based on hash tree
- Dramatically reduces storage requirement
- Does not use CRL format

Other Revocation Option

- Short-lived certificate
- Problems with this?

Attribute Certificate

- Binds an attribute to a Distinguished Name
- The attribute could be an authorization
- To grant access, the attribute certificate is combined with an ID certificate

PKI Definition #2

- Infrastructure that has the following components, infrastructure, & services:
  1. Certification Authority
  2. Certificate Repository
  3. Certificate Revocation
  4. Key Backup and Recovery
  5. Automatic Key Update
  6. Key History Management
  7. Cross-certification
  8. Supports Non-repudiation
  9. Time stamping
  10. Client Software

Review

- Definition and Description
- Functions
- Components
- Certificates