Public Key Infrastructures

Using PKC to solve network security problems
Distributing public keys

- P keys allow parties to share secrets over unprotected channels
  - Extremely useful in an open network:
    - Parties are not under a single manager
    - Symmetric keys cannot be shared beforehand

- How to distribute public keys?
  - Not a problem of secrecy (symmetric key)
  - A problem of legitimacy (identity binding)
Certification

- Public keys must be certified, i.e., an authenticated statement like “Public key PA belongs to user A” must be made by a trusted party.

- A Public Key Infrastructure defines:
  - The set of trusted parties or a mechanism to infer trust
  - An authentication/certification algorithm
Monopoly Model

- A central **Certification Authority (CA)** is:
  - universally trusted
  - its public key is known to all
- The central CA signs all public key certificates, or delegates its powers:
  - to lower level CAs: Certificate chaining
  - to registration authorities (RAs): single cert.
- This is a “flat” trust model.
Olygarchy

- The X.509 PKI is olygarchic.
- A number of root CAs is known in advance.
- User discretion is an afterthought; multiple points of failure.
- Certificate chaining is supported.
- Web browsers support olygarchic PKIs.
Certificate Revocation

- As the trusted parties multiply, so does the possibility of having to revoke trust
  - Private key of user compromised:
    - Revocation of user certificate
    - Publication of revoked certificates:
      - Certificate revocation lists, or CRLs.
  - Private key of trusted party compromised:
    - Update of CA’s public key
    - Re-certification of existing certificates?
    - Timestamping?
Anarchy model

- PGP: Each user is fully responsible for deciding its trust anchors (roots).
  - Practical for individual communication
    - Put your public key in your e-mail signature or website
    - Call user to verify PK fingerprint
  - Impractical for automated trust inference
    - How to decide that a certificate chain is trustworthy?
PGP: Details

- **PGP Identity** - Name and e-mail address associated with a key.
- **PGP Public key ring** - a local file/database of keys. Should have all keys that the user plans to correspond with, and any keys that have signed the user's public key.
- **PGP key server** - a networked repository for storing, retrieving, and searching for public keys. Key servers can use a few standardized protocols, among them LDAP, HTTP, and SMTP as public interfaces. A PGP key server is basically a centralized networked PGP public key ring.
- **Public key fingerprint** - A uniquely identifying string of numbers and characters used to identify public keys. This is the primary means for checking the authenticity of a key.
Constrained Naming PKIs

- **Assumptions:**
  - X.509 and other oligarchic PKIs cannot handle a very complex world without becoming very complex themselves.
  - Many certification needs are inherently local.
  - Local certification and local naming uniqueness can be maintained with minimal effort.
  - Global naming conventions exist (e.g.: DNS).
  - If public keys need global certification, then rely on relationships to infer trust.
Top-Down Constrained Naming

- Similar to oligarchic/monopoly model, but delegation takes place with domain name constraints:

```
/  
\  
.com  .edu  .uk
  \   /   /   /
.microsoft.com  .fsu.edu  .co.uk
      \         /         /
          .amazon.co.uk
```
Bottom-Up Constrained Naming

- Each organization creates an independent PKI and then link to others:
  - Top-down links: Parent certifies child
  - Bottom-up links: Child attests parent
  - Cross-links: A node certifies another node

- To certify a node N:
  1. Start from your trust anchor: if it is also an ancestor to N, just verify the delegation chain
  2. If (1) fails, query your trust anchor for a cross-link to an ancestor of N
  3. Else repeat using the parent of your trust anchor.
Example
Advantages of constrained naming PKIs

- Simple and flexible
- Locally deployable
- Compartmentalized trust
- Easy to replace keys at local levels
  - Lightweight and fast revocation
- Non-monopolistic, open architecture
- PKIX/X.509 (oligarchic) has recognized the advantages of constrained naming, and support it though the NameConstraints field.
Relative names

- Aliases, shorthand forms or non-global names that are locally understood:
  - Parent may refer to each child simply the part of the child’s name that extends of its own name
  - Child refers to parent simply as “parent”
  - Think of how file systems work
  - Cross links can use global names (absolute paths) or relative names
- SPKI certificates support relative names
Certificate revocation

- **CRLs:**
  - Signed, time-stamped list of all revoked certificates
  - Cost to generate and verify a CRL is proportional to the number of all revoked certificates

- **Δ CRLs:**
  - Publish only changes from a latest full CRL

- **OLRS (On-line Revocation Server)**
- **Affirmation of valid certificates**
Other issues

- Directories
  - A standardized mechanism for querying names is required for some PKIs (e.g. constrained names)
  - E.g.: DNS directory service
- Should a certification record be stored with the issuer or subject of the certification?
- Certificate chaining:
  - To certify Alice -- start with Alice’s name and go up (forward building) or with our trust anchor and down (reverse building)?
**X.509**

- The IETF chose to use X.500 naming standards for certificates
  - C=US, O=Sun, OU=Java, CN=java.sun.com
- Browsers know websites by DNS names, not X.500 names
  - Initial browser implementations did not check CN.
  - Today, DNS names are included either in CN or in SubjectAltName field
- **Rationale:** DNS does not support certificate lookup
X509 + PKIX Certificates

- Version
- SerialNumber
- Signature
- Issuer
- Validity
- Subject
- SubjectPublicKeyInfo
- IssuerUniqueIdentifier
- SubjectUniqueIdentifier
- AlgorithmIdentifier
- Encrypted
- Extensions
  - AuthorityKeyIdentifier
  - SubjectKeyIdentifier
  - KeyUsage
  - CertificatePolicies
  - PolicyMappings
  - NameConstraints
  - ...

PKIX Working Group (established 1995)

Goal: develop Internet standards needed to support an X.509-based PKI:

- RFC 2459, profiled X.509 version 3 certificates and version 2 CRLs for use in the Internet.
- Profiles for the use of Attribute Certificates (RFC XXXX [pending])
- LDAP v2 for certificate and CRL storage (RFC 2587)
- X.509 Public Key Infrastructure Qualified Certificates Profile (RFC 3039)
X.509

- Online Certificate Status Protocol (OCSP: RFC 2560)
- Certificate Management Request Format (CRMF: RFC 2511)
- Time-Stamp Protocol (RFC 3161)
- Certificate Management Messages over CMS (RFC 2797)
- Internet X.509 Public Key Infrastructure Time Stamp Protocols (RFC 3161)
- Use of FTP and HTTP for transport of PKI operations (RFC 2585)
Using capabilities for access control

- ACLs store permissions (read, write, execute, append, etc.) on the object
  - Easy to decide who has access to an object
  - Hard to revoke subjects
- Capabilities-based systems store capabilities on the subject
  - Hard to decide who has access to an object
  - Easy to revoke or add capabilities to a subject
- Role-based access control