CIS 6930/4930 Computer and Network Security

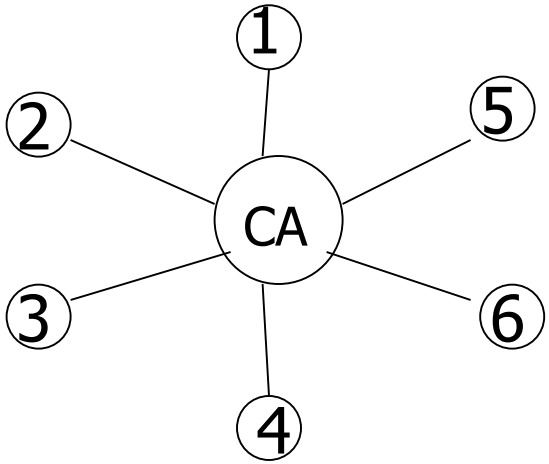
Topic 7.2 Public Key Infrastructure (PKI)

What Is PKI

- Informally, the infrastructure supporting the use of public key cryptography.
- A PKI consists of
  - Certificate Authority (CA)
  - Certificates
  - A repository for retrieving certificates
  - A method of revoking/updating certificates

Certification Authorities (CA)

- A CA is a trusted node that maintains the public keys for all nodes (Each node maintains its own private key)



If a new node is inserted in the network, only that new node and the CA need to be configured with the public key for that node

Certificates

- A CA is involved in authenticating users' public keys by generating certificates
- A certificate is a signed message vouching that a particular name goes with a particular public key
- Example:
  1. [Alice's public key is 876234]$_{carol}$
  2. [Carol's public key is 676554]$_{Ted}$ & [Alice's public key is 876234]$_{carol}$

- Knowing the CA's public key, users can verify the certificate and authenticate Alice's public key

Certificates

- Certificates can hold expiration date and time
- Alice keeps the same certificate as long as she has the same public key and the certificate does not expire
- Alice can append the certificate to her messages so that others know for sure her public key

CA Advantages

1. The CA does not need to be online. [Why?]

2. If a CA crashes, then nodes that already have their certificates can still operate.

3. Certificates are not security sensitive (in terms of confidentiality).
   - Can a compromised CA decrypt a conversation between two parties?
   - Can a compromised CA fool Alice into accepting an incorrect public key for Bob, and then impersonate Bob to Alice?

CA Problems

- What if Alice is given a certificate with an expiration time and then is revoked (fired) from the system?
  - Alice can still use her certificate till the expiration time expires.
  - What kind of harm can this do?
  - Alice can still exchange messages with Bob using her un-expired certificate.

- Solution:
  - Maintain a Certificate Revocation List (CRL) at the CA. A Certificate is valid if (1) it has a valid CA signature, (2) has not expired, and (3) is not listed in the CA's CRL list.

Terminology

- A CA signing a certificate for Alice's public key
  - CA $\rightarrow$ issuer Alice $\rightarrow$ subject
- Alice wants to find the Bob's public key
  - Bob $\rightarrow$ target
- Anyone with a public key is a principal
- Alice is verifying a certificate (or a chain of certificates)
  - Alice $\rightarrow$ verifier
- Trust anchor $\rightarrow$ A CA with a trusted public key

PKI Models

1. Monopoly model
2. Monopoly + RA
3. Delegated CAs
4. Oligarchy model
5. Anarchy model
6. Name constraints
7. Top-down with name constraints
8. Bottom-up with name constraints

Monopoly Model

- One CA universally trusted by everyone
- Everyone must get certificates from this CA
- The public key to this organization is the only PKI trust anchor and is embedded in all software and hardware

Problems

1. There is NO universally trusted organization
2. Monopoly control. CA could charge any fees.
3. Once deployed, it is hard to switch to a different CA
4. Entire world's security relies on this CA
5. Inconvenient.

PKI Models

1. Monopoly model
2. Monopoly + RA
3. Delegated CAs
4. Oligarchy model
5. Anarchy model
6. Name constraints
7. Top-down with name constraints
8. Bottom-up with name constraints

Monopoly + Registration Authorities (RA)

- RAs are affiliated with the single CA and are trusted by this CA.
- RAs check identities and provide the CA with relevant information (identity and public key information) to generate certificates.
- More convenient
- Still a monopoly. All the monopoly problems still hold.

PKI Models

1. Monopoly model
2. Monopoly + RA
3. Delegated CAs
4. Oligarchy model
5. Anarchy model
6. Name constraints
7. Top-down with name constraints
8. Bottom-up with name constraints

Delegated CAs

- The trust anchor (known CA) issues certificates to other CAs (delegated CAs) vouching for their trustworthiness as CAs.

- Users can obtain their certificates from delegated CAs instead of the trust anchor CA.

- Example:
  - [Carol's public key is 676554]$_{Ted}$ & [Alice's public key is 876234]$_{carol}$
  - Ted: trust anchor CA & Carol: delegated CA

PKI Models

1. Monopoly model
2. Monopoly + RA
3. Delegated CAs
4. Oligarchy model
5. Anarchy model
6. Name constraints
7. Top-down with name constraints
8. Bottom-up with name constraints

Oligarchy Model

- A few trusted CAs and a certificate issued by any one of them is accepted
- Competition between CAs is good
- Problems: Not as secure as the monopoly case
  - Need to protect more CAs (instead of only one)
  - Might be easier to trick a naïve user by inserting a bogus trust anchor in the list of trusted CAs
  - It is hard to examine the set of trust anchors and determine whether some has modified the set

PKI Models

1. Monopoly model
2. Monopoly + RA
3. Delegated CAs
4. Oligarchy model
5. Anarchy model
6. Name constraints
7. Top-down with name constraints
8. Bottom-up with name constraints

Anarchy Model (Web of Trust)

- Fully distributed approach. No CA or list of CA provided to the users. Anyone can sign certificates for anyone else.
- Each user is responsible for configuring some trust anchors .
- A database maintains these certificates.
- Unworkable on a large scale.

PKI Models

1. Monopoly model
2. Monopoly + RA
3. Delegated CAs
4. Oligarchy model
5. Anarchy model
6. Name constraints
7. Top-down with name constraints
8. Bottom-up with name constraints

Name Constraints

- A CA is responsible for certifying users in his domain only
  - USF CA certifies USF students
- Provides complete autonomy
- CAs need to be able to identify each other.
  - How?

PKI Models

1. Monopoly model
2. Monopoly + RA
3. Delegated CAs
4. Oligarchy model
5. Anarchy model
6. Name constraints
7. Top-down with name constraints
8. Bottom-up with name constraints

Top-Down with Name Constraints

- Everyone agrees on a root organization and the root CA delegates to other CA. (A centralized trust anchor (CA) + delegated CAs).
- To get a certificate, contact the root.
- You will be redirected to an appropriate delegated CA.
- Delegated CAs can only issue certificates for users in their domain.

PKI Models

1. Monopoly model
2. Monopoly + RA
3. Delegated CAs
4. Oligarchy model
5. Anarchy model
6. Name constraints
7. Top-down with name constraints
8. Bottom-up with name constraints

Bottom-Up with Name Constraints

- Each organization maintains its own CA, and CAs link to others.
  - A parent certifies its children and the children certify their parent
- The hierarchy is traversed in a bottom-up fashion.
  - In addition to up and down links, cross links are allowed

Advantages

1. Easy to navigate the hierarchy (similar to DNS).
2. No monopoly.
3. Replacing keys is reasonably easy.
4. Can be deployed in any organization without help from the rest of the world.
5. Authentication between users in the same organization does not need to go outside the organization.

Certificate Revocation

- Certificates for public keys (Campus IDs) might need to be revoked from the system
  - Someone is fired
  - Someone is graduated
  - Someone's certificate (card) is stolen

Certificate Revocation

- Certificates typically have an associated expiration time
  - Typically in the order of months (too long to wait if it needs to be revoked)

- Solutions:
  - Maintain a Certificate Revocation List (CRL)
  - A CRL is issued periodically by the CA and contains all the revoked certificates
  - Each transaction is checked against the CRL

CRLs

1. Why are CRLs issued periodically even if no certificates are revoked?

2. How frequent should CRLs be issued?

3. If a CRL is maintained, why associate an expiration time with certificates?

Delta CRL

- A Delta CRL includes lists changes from the last complete CRL

- Delta CRLs may be issued periodically (frequently) and full CRLs are issued less frequently

On-line Revocation Servers (OLRS)

- An OLRS is a system that can be queried over the network for the revocation status of individual certificates
- An OLRS maintains the full CRL list
- What if someone impersonates an OLRS?

  …..

  – Solution?

  …..

Good-lists vs. Bad-lists

- How about maintaining a list of valid certificates in the CRL instead of the revoked certificates?
- Is this more secure? Why?

- Problems:
  1. A good list is likely to be much larger than the bad list (worse performance)
  2. Organizations might not want to maintain its list of valid certificates public.

  Solution: The good-list can maintain only hashes of the valid certificates