COMP7120/8120 Cryptography and Data Security

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. \([\text{Alice’s public key is } 876234]_{\text{carol}}\)
  2. \([\text{Carol’s public key is } 676554]_{\text{Ted}} \& [\text{Alice’s public key is } 876234]_{\text{carol}}\)

• 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
Example

• CA - everyone knows CA’s public key.
  - CA is trusted.

• Alice wants to communicate to the real Bob
  - She sends a request to CA
  - Obtains a digital certificate from CA:
    
    Bob’s public key is 1902A12B2318871BF1
    Expiration: 1/1/2020
    [signed by CA]

Q: digital certificate vs digital signature?
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.

  Bob’s public key is 1902A1252318871BF1
  Expiration: 1/1/2020
  [signed by CA]

• 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 → issuer  Alice → subject

- Alice wants to find the Bob’s public key
  - Bob → target

- Anyone with a public key is a principal

- Alice is verifying a certificate (or a chain of certificates)
  - Alice → verifier

- Trust anchor → 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:
  - \([\text{Carol’s public key is } 676554]_{\text{Ted}} \& [\text{Alice’s public key is } 876234]_{\text{carol}}\)
  - 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
    - How do you trust a given list of trusted CAs?
  - It is hard to examine the set of trust anchors and determine whether someone 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
  - UofM CA certifies UofM students/faculty/staff

• 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
Bottom-Up with Name Constraints

How can $A/C/Y$ verify the certificate of $B/Y/Z/C$?

How can $B/Y/Z/C$ verify the certificate of $A/C/Y$?

Solution: Follow up-links until you encounter an ancestor of the target, then follow at most one cross-link, and then follow down-links from there.
PKI vs KDC

- Protocol design complexity
- Online and offline operations
- Security
- Scalability
- Efficiency