Key Distribution and Management
(Chapter 9.7; Chapter 15)

- Secret key distribution
- Public key distribution
- Secret key distribution using public key encryption

Trusted Intermediaries

Symmetric key problem
- How do two entities establish shared secret key over network?

Solution:
- Trusted key distribution center (KDC) acting as intermediary between entities

Public key problem
- When Alice obtains Bob’s public key (from web site, e-mail, diskette), how does she know it is Bob’s public key, not Trudy’s?

Solution:
- Trusted certification authority (CA)

Public Key Distribution

- General schemes:
  - Public announcement
    - Can be forged
  - Publicly available directory
    - Can be tampered
  - Public-key certification authorities (CAs)

Certification Authorities

- Certification authority (CA): binds public key to particular entity, E.

- E (person, router) registers its public key with CA.
  - E provides “proof of identity” to CA.
  - CA creates certificate binding E to its public key.
  - Certificate containing E’s public key digitally signed by CA: CA says “this is E’s public key”

- Certificate for Bob’s public key, signed by CA:
  - Bob’s public key
  - Digital signature (encrypt)
  - CA private key
  - Bob’s identifying information
  - CA
  - Bob's identifying information
  - Certificate
  - K_B
  - K_CA
  - K_{CA}
Public-key Certificates

- A certificate contains a public key and other information
  - Created by a certificate authority
  - Given to the participant with the matching private key

- A participant transmits its certificate to convey its key information
  - Other participants can verify that the certificate was created by the authority
    - All nodes are pre-configured with the public key of the certificate authority (CA)

Certification Authorities

- When Alice wants Bob’s public key:
  - gets Bob’s certificate (Bob or elsewhere).
  - apply CA’s public key to Bob’s certificate, get Bob’s public key

A Certificate Contains

- Serial number (unique to issuer)
- info about certificate owner, including algorithm and key value itself (not shown)

Exchange of Public-key Certificates

- A does: $C_A = E_{K_{ca, auth}}[T_1, ID_A, K_{e,a}]$
- B does: $D_{K_{e,a}}(C_A) = D_{K_{e,a}}(E_{K_{ca, auth}}[T_1, ID_A, K_{e,a}]) = (T_1, ID_A, K_{e,a})$
  - hence gets the trusted public key of A
SECRET KEY DISTRIBUTION

- A and B can establish a secret key by:
  - Manual delivery.
  - Selection and delivery by a trusted third party.
  - Using encrypted links to a third party to relay.
  - Using a previous key to encrypt the new key.

- Problem:
  - Need to scale up: need for each pair of hosts/applications...

KEY DISTRIBUTION CENTER (KDC)

- Responsible for distributing keys to pairs of users (hosts, processes, applications).
- Each user must share an unique key, the master key, with the KDC.
- Use the master key to communicate with KDC to get a temporary session key for establishing a secure "session" with another user.
- Master keys are distributed in some non-cryptographic ways.

Q: How does KDC allow Bob, Alice to determine shared symmetric secret key to communicate with each other?

Bob knows to use R1 to communicate with Alice.

Alice and Bob communicate: using R1 as one-time session key for shared symmetric encryption.
**Public Key** based Secret Key Distribution w/ Confidentiality and Authentication

1. \( E_{K_e,b}[N_1|ID_A] \)
2. \( E_{K_e,d}[N_1|N_2] \)
3. \( E_{K_e,b}[N_2] \)
4. \( E_{K_e,b}[E_{K_d,a}[K_s]] \)