Key Distribution

CS 470
Introduction to Applied Cryptography
Ali Aydin Selçuk

Public Key Cryptography

- Simple PKC solves key dist. problem against passive attackers (i.e., an attacker that just eavesdrops).
- Active attackers can send a fake public key & become a "man in the middle" (MitM).

Notation:
- \( \{M\}_X \): message M enc. with the pub. key of X
- \([M]_X \): message M signed with the prv. key of X

MitM Attack against RSA

Normal op:

MitM attack:

MitM Attack against DH

Normal op:

MitM attack:
Trusted Third Parties

- Solution against active attackers: "Trusted Third Parties" (TTPs)
- Symmetric key solution: KDC
  - Everyone registers with the KDC, shares a secret key.
  - When A & B want to communicate, they contact the KDC & obtain a session key.
- Public key solution: CA
  - Everyone registers with the CA, obtains a "certificate" for his/her public key.
  - Certificate: A document signed by the CA, including the ID and the public key of the subject.
  - People obtain each other’s certificates thru a repository, a webpage, or at the beginning of the protocol,
  - and use the certified public keys in the protocols.

KDC vs. CA

- KDC
  - faster (being based on symmetric keys)
  - has to be online
- CA
  - doesn’t have to be online
  - if crashes, doesn’t disable the network
  - much simpler
  - scales better
  - certificates are not disclosure-sensitive
  - a compromised CA can’t decrypt conversations
- KDCs are preferred for LANs, CAs for WANs (e.g., the Internet).

Key Distribution with KDC

A simple protocol:

\[ K_A^a, K_B^b : \text{Long-term secret keys of Alice, Bob.} \]
\[ K_A^a \{m\} : \text{Encryption of } m \text{ with } K_A. \]

KDC

A simple protocol:

- certificates are obtained in advance
- session key is transported with public key encryption

~ SSL key exchange:

\[ \text{cert} \leftarrow [\{ID, PK\}_C, \ldots]_{CA} \]
\[ k \leftarrow [k] \]
\[ k \text{ is the session key} \]

Key Distribution with CA
DH with Certificates

- STS Protocol: Authenticated DH protocol; basis for many real-life app’s.
- Certified PKs are used for signing the public DH parameters. A slightly simplified version:

  Alice \( x \) Bob
  
  \[ \text{cert}_B, y, [x,y]_B \]
  
  \[ \text{cert}_A, [x,y]_A \]

  where \( x = g^a \mod p \), \( y = g^b \mod p \), \( k = g^{ab} \mod p \).

- Feature: “Perfect forward secrecy”