COMP7120/8120 Cryptography and Data Security

Trusted Intermediaries - Key Distribution Center
Trusted Key Servers

• How do a large number of users authenticate each other?
  - inefficient / impractical for every pair of users to negotiate a secret key or share passwords

• Alternative: everybody shares a key with (and authenticates to) a single trusted third-party
Trusted Intermediaries

• Problem: authentication for large networks

• Solution #1
  - Key Distribution Center (KDC)
    • Representative solution: Kerberos
  - Based on secret key cryptography

• Solution #2
  - Public Key Infrastructure (PKI)
  - Based on public key cryptography
Key Distribution Center (KDC)

- Shared keys between the *Key Distribution Center (KDC)* and users

Q: Can users establish the same key with KDC?
(Simplified) Example of Use

- Alice wishes to communicate securely with Bob; Alice has previously negotiated $K_{A-KDC}$ with the KDC, Bob has negotiated $K_{B-KDC}$

1. Alice requests from the KDC a session key to use with Bob
2. KDC generates session key $K_S$, sends to Alice, encrypted with $K_{A-KDC}$
3. KDC also sends $K_S$ to Bob, encrypted with $K_{B-KDC}$
   - Alice and Bob can then communicate using $K_S$
Assessment

• Simplifies mutual authentication / key negotiation, but...
  - secure against attacks?
    • if KDC cheats ...
  - robust to failures?
    • if KDC fails ...
  - efficient?
A Hierarchy of KDCs

• For an Internet, not practical to have a single KDC
  - instead, imagine one KDC per domain
• To communicate securely with user in your own domain, just contact your domain’s KDC
• To talk with user in another domain, your KDC needs to contact the other domain’s KDC
  - KDCs must be able to authenticate each other and communicate securely
Hierarchy... (cont’d)

Domain 1

A → K_{A-K1} → KDC-1 → B → K_{B-K1} → C

Domain 2

E → K_{E-K2} → KDC-2 → D → K_{D-K2}
Mediated Authentication (With KDC)

KDC operation (in principle)

Alice wants to talk to Bob

Alice

KDC operation (in principle)

KDC

Bob

• Some concerns
  - Trudy may claim to be Alice and talk to KDC
    • Trudy cannot get anything useful
  - Messages encrypted by Alice using $K_{AB}$ may arrive at Bob before KDC’s message $K_{Bob}{K_{AB}}$ arrive
Mediated Authentication (With KDC)

KDC operation (in practice)

Alice \[\xrightarrow{K_{\text{Alice}}\{K_{AB}\}, \ K_{\text{Bob}}\{K_{AB}\}}\] \(\xrightarrow{\text{Generate } K_{\text{AB}}}\) KDC \(\xrightarrow{\text{I’m Alice, here is my ticket: } K_{\text{Bob}}\{K_{AB}\}}\) Bob

- Must be followed by a mutual authentication exchange
  - To confirm that Alice and Bob have the same key
Needham-Schroeder Protocol

- Classic protocol for authentication with KDC
  - Many others have been modeled after it (e.g., Kerberos)

Alice wants to talk to Bob

KDC

Bob

Q: Why $N_1$, $N_2$, $N_3$?
Needham-Schroeder Protocol (Cont’d)

• A vulnerability
  - When Trudy gets a previous key used by Alice, Trudy may reuse a previous ticket issued to Bob for Alice
  - Essential reason
    • The ticket to Bob stays valid even if Alice changes her key
Expanded Needham-Schroeder Protocol

I want to talk to you

$K_{Bob}\{N_B\}$

Generate $K_{AB}$; extract $N_B$

$N_1$, Alice wants Bob, $K_{Bob}\{N_B\}$

$K_{Alice}\{N_1, "Bob", K_{AB}, ticket\ to\ Bob\}$, where $ticket\ to\ Bob = K_{Bob}\{K_{AB}, Alice, N_B\}$

$K_{AB}\{N_2\}$

ticket to Bob, $K_{AB}\{N_2\}$

$K_{AB}\{N_2-1, N_3\}$

$K_{AB}\{N3-1\}$