Authentication Handshakes

- Secure communication almost always includes an initial authentication handshake.
  - Authenticate each other
  - Establish session keys
  - This process is not trivial; flaws in this process undermine secure communication

Authentication with Shared Secret

- Weaknesses
  - Authentication is not mutual; Trudy can convince Alice that she is Bob
  - Trudy can hijack the conversation after the initial exchange
  - If the shared key is derived from a password, Trudy can mount an off-line password guessing attack
  - Trudy may compromise Bob’s database and later impersonate Alice
Authentication with Shared Secret (Cont’d)

- A variation
  - Requires reversible cryptography
  - Other variations are possible
- Weaknesses
  - All the previous weaknesses remain
  - Trudy doesn’t have to see R to mount off-line password guessing if R has certain patterns (e.g., concatenated with a timestamp)
    * Trudy sends a message to Bob, pretending to be Alice

Authentication with Public Key

- Bob’s database is less risky
- Weaknesses
  - Authentication is not mutual; Trudy can convince Alice that she is Bob
  - Trudy can hijack the conversation after the initial exchange
  - Trudy can trick Alice into signing something
    * Use different private key for authentication

Authentication with Public Key (Cont’d)

A variation
Mutual Authentication

Alice
- I'm Alice
- \( R_1 \)
- \( f(K_{Alice-Bob}, R_1) \)
- \( R_1 \)
- \( f(K_{Alice-Bob}, R_1) \)

Bob

Mutual Authentication (Cont’d)

- **Reflection attack**

Trudy
- I'm Alice, \( R_2 \)
- \( R_2(f(K_{Alice-Bob}, R_1)) \)

Bob

- I'm Alice, \( R_2 \)
- \( R_2(f(K_{Alice-Bob}, R_1)) \)

Reflection Attacks (Con’d)

- **Lesson:** Don’t have Alice and Bob do exactly the same thing
  - Different keys
    - Totally different keys
    - \( K_{Alice-Bob} = K_{Bob-Alice} + 1 \)
  - Different Challenges
  - The initiator should be the first to prove its identity
    - Assumption: initiator is more likely to be the bad guy
Mutual Authentication (Cont’d)

• Password guessing

\[
\text{I'm Alice, } R_2, f(K_{Alice-Bob}, R_2), \ f(K_{Alice-Bob}, R_1) \\
\text{Bob}
\]

Mutual Authentication (Cont’d)

• Public keys

  – Authentication of public keys is a critical issue

\[
\text{I'm Alice, } R_2, \{R_1\}_{Bob} \\
\text{Bob, } \{R_1\}_{Alice}, R_2 \\
\text{Alice, } R_1
\]

Mutual Authentication (Cont’d)

• Mutual authentication with timestamps

  – Require synchronized clocks
  – Alice and Bob have to encrypt different timestamps

\[
\text{I'm Alice, } f(K_{Alice-Bob}, \text{timestamp}) \\
\text{Bob, } f(K_{Alice-Bob}, \text{timestamp}+1)
\]
Integrity/Encryption for Data

- Communication after mutual authentication should be cryptographically protected as well
  - Require a session key established during mutual authentication

Establishment of Session Keys

- Secret key based authentication
  - Assume the following authentication happened.
  - Can we use $K_{Alice-Bob}(R)$ as the session key?
  - Can we use $K_{Alice-Bob}(R+1)$ as the session key?
  - In general, modify $K_{Alice-Bob}$ and encrypt $R$. Use the result as the session key.

```
<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>I'm Alice</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td></td>
</tr>
<tr>
<td>$K_{Alice-Bob}(R)$</td>
<td></td>
</tr>
</tbody>
</table>
```

Establishment of Session Keys (Cont’d)

- Two-way public key based authentication
  - Alice chooses a random number $R$, encrypts it with Bob’s public key
    - Trudy may hijack the conversation
  - Alice encrypts and signs $R$
    - Trudy may save all the traffic, and decrypt all the encrypted traffic when she is able to compromise Bob
    - Less severe threat
Two-Way Public Key Based Authentication (Cont’d)

• A better approach
  – Alice chooses and encrypts \( R_1 \) with Bob’s public key
  – Bob chooses and encrypts \( R_2 \) with Alice’s public key
  – Session key is \( R_1 \oplus R_2 \)
  – Trudy will have to compromise both Alice and Bob

• An even better approach
  – Alice and Bob estabish the session key with Diffie-Hellman key exchange
  – Alice and Bob signs the quantity they send
  – Trudy can’t learn anything about the session key even if she compromises both Alice and Bob

Establishment of Session Keys (Cont’d)

• One-way public key based authentication
  – It’s only necessary to authenticate the server
    • Example: SSL
  – Encrypt \( R \) with Bob’s public key
  – Diffie-Hellman key exchange
    • Bob signs the D-H public key

Mediated Authentication (With KDC)

KDC operation (in principle)

- Some concerns
  – Trudy may claim to be Alice and talk to KDC
  – Trudy cannot get anything useful
  – Messages encrypted by Alice may get to Bob before KDC’s message
  – It may be difficult for KDC to connect to Bob
Mediated Authentication (With KDC)

- 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)
- Nonce: A number that is used only once
  - Deal with replay attacks

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

- The additional two messages assure Bob that the initiator has talked to KDC since Bob generates $N_B$.

1. Alice wants to talk to Bob, $K_{ab}(N_a)$
2. Generate $K_{ab}$, extract $N_B$
3. $N_C$, Alice wants Bob, $K_{ab}(N_a)$
4. $K_{ab}(N_a, "Bob", K_{ab}(N_a, Alice))$
5. where $ticket_{Bob} = K_{ab}(K_{ab}(Alice, N_a), N_B)$
6. $K_{ab}(N_B, N_C)$
7. $K_{ab}(N_B, N_C)$

Otway-Rees Protocol

- Only has five messages
- KDC checks if $N_C$ matches in both cipher-texts
  - Make sure that Bob is really Bob

2. Generate $K_{ab}$, Extract $N_C$
3. $K_{ab}(N_a, N_C, "Alice", "Bob")$
4. $N_C, K_{ab}(N_a, N_C), K_{ab}(N_B, K_{ab})$
5. $K_{ab}(N_B, N_C)$
6. $K_{ab}(N_B, N_C)$
7. $K_{ab}(anything\ recognizable)$