Announcements

Homework 1
• Grade released
• Have 1-week “rebuttal period”
  • Submit re-grade request via GradeScope
Lecture 10

Protocols (Continued)

Chapters 9 and 11 in KPS

[lecture slides are adapted from previous slides by Prof. Gene Tsudik]
Recap: Key Distribution Center (KDC) aka Trusted Third Part (TTP)

- Alice and Bob need to share a key
- KDC shares different master key with *each* registered user (many users)
- Alice and Bob know their own master keys: $K_A$ and $K_B$

for communicating with KDC
Alice and Bob communicate using $K$ as a short-term (session) key for encryption and/or data integrity.

**Note:**
- $\text{Msg2}$ is not tied to $\text{Msg1}$
- $\text{Msg1}$ is possibly old
- $\text{Msg2}$ is possibly old and so is $\text{Msg3}$
- Bob and Alice don’t authenticate each other!
A Typical Key Distribution Scenario

1. Request, B, N₁
2. $E_{K_a}[K_s, \text{Request, } N₁, E_{K_b}(K_s, A)]$
3. $E_{K_b}[K_s, A]$
4. $E_{K_s}[A, N₂]$
5. $E_{K_s}[f(N₂)]$

Notes:
- Msg2 is tied to Msg1
- Msg2 is fresh/new
- Msg3 is possibly old *
- Msg1 is possibly old (KDC doesn’t authenticate Alice)
- Bob authenticates Alice
- Bob authenticates KDC
- Alice DOES NOT authenticate Bob

$E_K[X] = \text{Encryption of } X \text{ with } K$
Public Key Distribution

General schemes:
• Public announcement (e.g., in a newsgroup or email message)
  • Can be forged
• Publicly available directory
  • Can be tampered with
• Public-key certificates (PKCs) issued by trusted off-line Certification Authorities (CAs)
Certification Authorities

- Certification authority (CA): trusted, highly secure (physically and electronically) component
- Issues public key certificates; each binds a public key to a specific entity
- Each entity (user, host, etc.) registers its public key with CA.
  - Bob provides “proof of identity” to CA.
  - CA creates public key certificate binding Bob’s ID/name to this public key.
  - Certificate containing Bob’s public key is signed by CA:
    CA says: “this is Bob’s public key”
When Alice wants to get Bob's public key:
- Get Bob's certificate (from Bob or elsewhere)
- Using CA's public key verify the signature on Bob's certificate
- Check for expiration
- Check for revocation (we’ll talk about this later)
- Extract Bob’s public key
A Certificate Contains

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

- info about certificate issuer
- valid dates
- digital signature by issuer
A Sample Certificate (1/2)
A Sample Certificate (2/2)

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public Key Info</td>
<td></td>
</tr>
<tr>
<td>Algorithm</td>
<td>RSA Encryption (1.2.840.113549.1.1.1)</td>
</tr>
<tr>
<td>Parameters</td>
<td>none</td>
</tr>
<tr>
<td>Public Key</td>
<td>256 bytes : D7 D3 86 4F 23 D4 E6 E4 ...</td>
</tr>
<tr>
<td>Exponent</td>
<td>65537</td>
</tr>
<tr>
<td>Key Size</td>
<td>2048 bits</td>
</tr>
<tr>
<td>Key Usage</td>
<td>Any</td>
</tr>
<tr>
<td>Signature</td>
<td>256 bytes : 97 6B 72 86 AD 24 65 AD ...</td>
</tr>
<tr>
<td>Extension</td>
<td>Subject Key Identifier (2.5.29.14)</td>
</tr>
<tr>
<td>Critical</td>
<td>NO</td>
</tr>
<tr>
<td>Key ID</td>
<td>84 61 D1 1A 2F B1 EF 8E 4F F4 6F F0 8D 26 FC 91 58 77 9C A3</td>
</tr>
<tr>
<td>Extension</td>
<td>Authority Key Identifier (2.5.29.35)</td>
</tr>
<tr>
<td>Critical</td>
<td>NO</td>
</tr>
<tr>
<td>Key ID</td>
<td>DB D4 F7 BB 15 76 6C 3B 01 A5 23 59 C2 37 26 97 46 5D DC 46</td>
</tr>
<tr>
<td>Extension</td>
<td>Subject Alternative Name (2.5.29.17)</td>
</tr>
<tr>
<td>Critical</td>
<td>NO</td>
</tr>
<tr>
<td>DNS Name</td>
<td><a href="http://www.google.com">www.google.com</a></td>
</tr>
<tr>
<td>Fingerprints</td>
<td></td>
</tr>
<tr>
<td>SHA1</td>
<td>30 69 24 F3 14 57 D4 84 73 7F B2 BE B8 F5 92 A2 46 8E 9D 2E</td>
</tr>
<tr>
<td>MD5</td>
<td>20 CD 07 D1 A3 F4 96 95 2F 33 43 4D E6 F3 D0 1E</td>
</tr>
</tbody>
</table>

OK
Back to Protocols

\(\{X\}_K = \text{Encryption of } X \text{ with key } K\)

1. \(A \rightarrow T: \ A, B, N_A\)
2. \(T \rightarrow A: \ \{N_A, B, K, \{K, A\}_K\}_K\)
3. \(A \rightarrow B: \ \{K, A\}_K\)
4. \(B \rightarrow A: \ \{N_B\}_K\)
5. \(A \rightarrow B: \ \{N_B-1\}_K\)
Denning-Sacco Attack: suppose Eve recorded an old protocol session for which she somehow knows the session key $K'$:

1. $A \rightarrow T$: $A, B, N_A$
2. $T \rightarrow A$: $\{N_A, B, K', \{K', A\}_{KB}\}_{KA}$
3. $A \rightarrow B$: $\{K', A\}_{KB}$

-----------------------------------------------

At a later time:

3. $E \rightarrow B$: $\{K', A\}_{KB}$
4. $B \rightarrow E$: $\{N_B\}_{K'}$
5. $E \rightarrow B$: $\{N_B-1\}_{K'}$
Fixing the Attack

• Bob has no guarantees about freshness of the message in step 3.
• Eve exploits this to impersonate Alice to Bob - old session keys are useful.
• Can be fixed by adding timestamps:
  • Limits usefulness of old session keys
  • Eve’s attack becomes:

3: \text{E} \rightarrow \text{B}: \{K’, T’, A\}_{K_B}

attack is now thwarted because T’ is stale
PK-based Needham-Schroeder Protocol

1. \{A,B\}
2. \{PK_b,B\}_{SK_T}
3. \[[N_a, A]_{PK_b}\]
4. \{B,A\}_{SK_T}
5. \{PK_a,A\}_{SK_T}
6. \[[N_a, N_b]_{PK_a}\]
7. \[[N_b]_{PK_b}\]

- CERT_B = Message 2, CERT_A = Message 5
- PK_A: Alice’s public key, PK_B: Bob’s public key
- SK_T: TTP’s secret (private) key used for signing
- Everyone knows TTP’s public key PK_T
- [X]_K = Encryption of X with key K
Another Attack

• 1, 2, 4, 5: Delivery of public key
• Does not guarantee freshness of the public key

How to solve it?

• Timestamp in messages 2 and 5 or challenges in messages 1&2 and 4&5
• Public Key Certificate: assign expiration time/data to each certificate (messages 2 and 5)
PK-based Denning-Sacco Attack

1. A, B

2. Cert_A, Cert_B

3. Cert_A, Cert_B, \[ \{ K_{AB,T_A} \}_{SK_A} \]_{PK_B}

4. Secure communication with \( K_{AB} \)

3’. Cert_A, Cert_C, \[ \{ K_{AB,T_A} \}_{SK_A} \]_{PK_C}

4’. Secure communication with \( K_{AB} \)

Cert_A\={PK_A,A}_{SK_T}
Cert_B\={PK_B,B}_{SK_T}
Cert_C\={PK_C,C}_{SK_T}
Lowe’s Attack  
(Impersonation by Interleaving)

Original

3. \( A \rightarrow B: \ [N_a, A]_{PKb} \)
6. \( B \rightarrow A: \ [N_a, N_b]_{PKa} \)
7. \( A \rightarrow B: \ [N_b]_{PKb} \)

Fix

3. \( A \rightarrow B: \ [N_a, A]_{PKb} \)
6. \( B \rightarrow A: \ [B, N_a, N_b]_{PKa} \)
7. \( A \rightarrow B: \ [N_b]_{PKb} \)

Attack: E impersonates A

3. \( A \rightarrow E: \ [N_a, A]_{Pke} \)
3. \( E \rightarrow B: \ [N_a, A]_{PKb} \)
6. \( B \rightarrow E: \ [N_a, N_b]_{Pka} \)
6. \( E \rightarrow A: \ [N_a, N_b]_{Pka} \)
7. \( A \rightarrow E: \ [N_b]_{Pke} \)
7. \( E \rightarrow B: \ [N_b]_{PKb} \)