Handshake Protocols

Monday, October 18, 2010
Reading:  S&M Ch. 9; Kaufman, Perlman, & Speciner, Ch. 11;
Anderson, Ch 3

CS342 Computer Security
Department of Computer Science
Wellesley College

One-Way vs. Two-Way (Mutual) Authentication

One-Way Authentication
In one-way authentication Bob needs to be convinced that a
conversation request from Alice is really from Alice. (Bob is often
a remote server to which Alice is logging in from a local computer.)

Two-Way (Mutual) Authentication
Both Alice and Bob need to be convinced about the authenticity of
the other party.

Attackers
In both of these authentication situations, we do need to worry about
Eve and Mallory, who might try to impersonate Alice or Bob.

Game Plan
We’ll first study one-way authentication and associated attacks.
Then we’ll move on to mutual authentication.
**One-Way Authentication: Plaintext Passwords**

Alice sends password "in the clear" to Bob.

FTP, telnet, and HTTP passwords actually work this way!

*Attacks?*

---

**One-Way Authentication: Hashed Passwords**

Alice sends hash of password to Bob, who compares it to his database of hashed passwords for all users.

*Benefits?*

*Attacks?*
Password Interception

![Diagram]

Mallory can intercept password before it's hashed.

What are some ways she can do this?

Motivation for mutual authentication!

---

Man-in-the-Middle /Chess Grandmaster Attacks

Many password interception techniques are examples of man-in-the-middle (MITM) attacks: Mallory impersonates Alice to Bob and Bob to Alice.

![Diagram]

This is also known as the chess grandmaster attack: Mallory can play remote chess against grandmasters Alice (white) and Bob (black) at the same time and will win one game or draw both.
Replay Attack

Eve can record encrypted password and later replay it.

*How to foil replay attack?*

---

**Foiling Replay Attack**

- **Timestamps/Sequence Numbers/Nonces**
  - but, as we'll see in more detail later, these have problems:
    - timestamps require synchronization and might be guessable
    - consecutive sequence numbers are obviously related
    - nonces must be remembered by Bob.

- **Password Aging**
  - bad social properties; people try to circumvent.
  - how frequent to be effective deterrent?

- **One-time Passwords (OTP)**: Alice has a sequence of passwords and uses each only once (they're nonreusable!).
  - how to generate password sequence?
  - how to synchronize Alice and Bob?

- **Challenge-Response**: Alice responds to new challenge from Bob on each login.
One-Way Authentication: Encrypted Passwords

Alice sends password encrypted with key $K_{AB}$ she shares with Bob.

Note: $E(K_{AB}, \text{password})$ is often written $(\text{password})_{K_{AB}}$.

Benefits?

Attacks?

OTP: Tokens (e.g., RSA SecurID)

RSA SecurID token displays number determined from seed and time. User enters displayed number and PIN/password (two factors!), and info sent to central server (which also knows seed and time).

Problems?

Hardware tokens

Software tokens
**OTP: Iterated Hashing† (e.g., S/Key)**

One-way hash function $H$ is used to generate sequence of $n$ passwords from seed $p_0$: passwords $p_i = H(p_{i-1})$ are used in order $p_{n-1}, p_{n-2}, \ldots$

**Initialization**

<table>
<thead>
<tr>
<th>Alice</th>
<th>$&lt;\text{&quot;I'm Alice&quot;}, n, H^n(p_0)&gt;$</th>
<th>Bob</th>
<th>$&lt;n, H^n(p_0)&gt;$</th>
</tr>
</thead>
</table>

**Authentication**

<table>
<thead>
<tr>
<th>Alice</th>
<th>$&lt;\text{&quot;I'm Alice&quot;}$</th>
<th>Bob</th>
<th>$&lt;i, H^i(p_0)&gt;$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_0$</td>
<td>$i$</td>
<td>$p_{i-1} = H^{i-1}(p_0)$</td>
<td>$\text{Bob checks that } h_i = H(h_{i-1})$</td>
</tr>
<tr>
<td>$\text{&quot;I believe you're Alice&quot;}$</td>
<td>$&lt;i-1, H^{i-1}(p_0)&gt;$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

†Also known as Lamport's hash after its inventor, Leslie Lamport.

**Man-in-the-Middle (Small n) Attack on S/Key**

If Mallory can impersonate Bob, she can present a challenge $k \leq i$ and then impersonate Alice for $(k - i) + 1$ steps.

<table>
<thead>
<tr>
<th>Alice</th>
<th>$&lt;\text{&quot;I'm Alice&quot;}$</th>
<th>Mallory</th>
<th>$&lt;i, H^i(p_0)&gt;$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_0$</td>
<td>$k$</td>
<td>$p_{k-1} = H^{k-1}(p_0)$</td>
<td>$&lt;\text{&quot;I'm Alice&quot;}$</td>
</tr>
<tr>
<td>$\text{&quot;I believe you're Alice&quot;}$</td>
<td>$&lt;i-1, H^{i-1}(p_0)&gt;$</td>
<td>$i \geq k$</td>
<td>$p_{i-1} = H^{i-k}(p_{k-1}) = H^{i-1}(p_0)$</td>
</tr>
</tbody>
</table>

Handshake Protocols 12-12
One-Way Authentication: Challenge-Response

Alice responds to nonce challenge \( R \) from Bob by encrypting or hashing \( R \) with shared key \( K_{AB} \).

\[
\text{Alice} \quad \rightarrow \quad \text{"I'm Alice"} \quad \rightarrow \quad \text{Bob}
\]

\[
\text{Alice} \quad \rightarrow \quad R \quad \rightarrow \quad \text{E}(K_{AB}, R) \text{ or } H(\langle K_{AB}, R \rangle)
\]

MITM Attack on Challenge-Response

MIG-in-the-Middle attack, described by Anderson:

\[
\text{SAAF radar in Namibia} \quad \rightarrow \quad \text{R} \quad \rightarrow \quad \text{Cuban MIG over Namibia} \quad \rightarrow \quad \text{R} \quad \rightarrow \quad \text{Cuban radar in Angola} \quad \rightarrow \quad \text{R} \quad \rightarrow \quad \text{SAAF jet over Angola}
\]

\[
\text{SAAF radar in Namibia} \quad \rightarrow \quad \text{E}(K_{SAAF}, R) \quad \leftarrow \quad \text{Cuban MIG over Namibia} \quad \rightarrow \quad \text{E}(K_{SAAF}, R) \quad \leftarrow \quad \text{Cuban radar in Angola} \quad \rightarrow \quad \text{E}(K_{SAAF}, R) \quad \leftarrow \quad \text{SAAF jet over Angola}
\]
**Other Challenge-Response Attacks**

Violated assumptions can lead to many attacks:

**Impersonation:** Nothing authenticates Bob, so Mallory can impersonate Bob.

**Known Plaintext:** Plaintext R can help Eve figure out key $K_{AB}$.

**Chosen Plaintext:** As Bob, Mallory can get Alice to encrypt any $M$.

**Session Hijacking:** Mallory hijacks conversation after initial handshake.

**Server Database Attack:** Mallory may be able to find $K_{AB}$ (and thereby impersonate Alice) by attacking Bob's database.

**PredictableNonce:** If nonce is a sequence number or coarse-grained timestamp, Mallory can replace R by a "later" $R'$, obtain $E(K_{AB}, R')$ from Alice, and use this to impersonate Alice for later nonce $R'$. (This is impractical for fine-grained timestamp.)

*Moral:* Cryptography not enough by itself - must consider system in which it's used. (Radia Perlman's talk on *How to Build an Insecure System out of Perfectly Good Cryptography.*)

---

**Encrypted Challenge-Response**

```
Alice: E(K_{AB}, R)

Bob: "I'm Alice"
```

- Mallory can't impersonate Bob for unrepeatable nonce (e.g., R includes fine-grained timestamp) since she doesn't know $K_{AB}$ (unless she's MITM).
- If R includes fine-grained timestamp, Bob is authenticated too!
- Must use encryption rather than hash. Why?
- Solves chosen plaintext problem.
- As with regular challenge-response, this suffers from MITM attack, known-plaintext attack, session hijacking, server database attack, and predictable nonce attack. Predictable nonce attack even worse here, since Alice needn't be involved for Mallory to impersonate her by guessing "later" $R$.
**One-Message Encrypted Challenge-Response**

Alice $\left< \text{"I'm Alice"}, E(K_{AB}, \text{timestamp}) \right>$ Bob

More efficient, but:

- Bob is no longer authenticated
- if timestamp is course-grained, Eve can replay message to impersonate Alice (unless Bob keeps records of timestamps already used)
- if Mallory convinces Bob to set his clock back, can replay old messages.

---

**One-Way Authentication with Public Key Crypto**

Alice $E(K_{Apub}, R)$ Bob

Alice $R$ Bob $D(K_{Apriv}, R)$

- No shared secrets in Bob's database, so addresses server database attack.
- Still suffers from impersonation, man-in-the-middle, session hijacking, known plaintext, and predictable nonce attacks.
- By impersonating Bob, Mallory can get Alice to encrypt or sign arbitrary messages! How to address this?
- How can these protocols be modified so that Bob is authenticated?
Foiling Predictable Nonce & Known Plaintext

- Nonce is always sent encrypted, so it’s safe to use a predictable one.
- If nonce is unpredictable, foils known-plaintext attack.

Mutual Authentication

- Many one-way authentication problems due to unauthenticated Bob.
- In practice, mutual authentication very important:
  - are you ordering from Amazon or an Amazon impersonator?
  - are you giving your personal info to Citibank or a phisher?
  - are you sending embarrassing messages to friend or blackmailer?
Mutual Authentication as Two One-Ways

Mutual 1

Alice

"Alice"

<"Bob", R₁>

< E(K_{AB}, R₁), R₂ >

or

< H(K_{AB}, R₁), R₂ >

Bob

E(K_{AB}, R₂)

or

H(K_{AB}, R₂)

Vulnerable to same attacks for one-way protocols except Bob impersonation.

Attempted Optimization of Mutual 1

Alice

<"Alice", R₂>

<"Bob", R₁, E(K_{AB}, R₂)>

Bob

E(K_{AB}, R₁)

Chosen Plaintext Attack

Mallory

<"Alice", M>

<"Bob", R₁, E(K_{AB}, M)>

Bob

Reflection Attack

Mallory

<"Alice", R₂ (session 1)>

<"Bob", R₁, E(K_{AB}, R₂) (session 1)>

Bob

<"Alice", R₁ (session 2)>

<"Bob", R₁ (session 2)>

E(K_{AB}, R₁ (session 1))

Handshake Protocols 12-21

Moral: Beware changing or designing protocols! Best to stick with well-studied ones.
Asymmetry Foils Replay Attack

ISO SC27 Mutual Authentication Protocol (S&M)