This Lecture:

- Password-Based Authentication (Continued)
- Putting it Together: Secure Authentication
- Case Study: Group Chat
- Single Sign-On (SSO) Authentication
One-Sided Secure Authentication

\[(V_{K_b}, S_{K_b}) \leftarrow \text{Gen}(1^\lambda)\]

\[
\begin{align*}
\text{User} & \quad g^a & \quad \text{Server} \\
\downarrow & \quad g^a & \quad \downarrow \\
g^{ab} & \quad \text{Diffie-Hellman Key Exchange} & \quad g^{ab} \\
\downarrow \text{Hash} & \quad \text{Hash} & \\
K & \quad (K_1, K_2) & \\
(K_1, K_2) & \quad \text{Authenticated Encryption} & \\
& \quad (\text{Encrypt-then-MAC}) & \\
\end{align*}
\]
Certificate Chain

Root CA

\((\text{VK}, \text{SK}) \leftarrow \text{Gen}(1^\lambda)\)

\(\text{Cert}_1: 6 \leftarrow \text{Sign}_{\text{sk}}(\text{VK}_1)\)

Intermediate CA1

\(\text{Cert}_2 = (\text{VK}_1, 6 \leftarrow \text{Sign}_{\text{sk}}(\text{VK}_1))\)

\(\text{VK}_2, 6_1 \leftarrow \text{Sign}_{\text{sk}_1}(\text{VK}_2)\)

Intermediate CA2

\(\text{Cert}_{\text{Bob}} = (\text{VK}_1, 6 \leftarrow \text{Sign}_{\text{sk}}(\text{VK}_1))\)

\(\text{VK}_2, 6_1 \leftarrow \text{Sign}_{\text{sk}_1}(\text{VK}_2)\)

\(\text{VK}_B, 6_2 \leftarrow \text{Sign}_{\text{sk}_2}(\text{VK}_B)\)

\((\text{VK}_B, \text{SK}_B) \leftarrow \text{Gen}(1^\lambda)\)

Short-lived certificates?
Password-Based Authentication

ID, password

\[ h = H(\text{password}) \]

Login

Store (ID, h)

Check

Attacks?
Online Dictionary Attack

User

ID, password

h = H(password)

Server

Signup

ID, h

Store (ID, h)

Login

ID, h'

h' = H(pwd1)
Offline Dictionary Attack

User

ID, password

$h = H(password)$

Server

Signup

ID, $h$

Store $(ID, h)$

$h' = H(pwd')$

(preprocessing)
Salting

**User**

- ID, password

**Server**

- Salt \(\gets \{0, 1\}\)

1. **Signup**
   - User sends ID
   - Server generates random salt and sends it back
   - User computes hash function: \(h = H(\text{password} || \text{salt})\)
   - Server stores (ID, Salt, h)

2. **Login?**
   - User enters ID
   - Server verifies if ID exists
   - User enters password
   - Server authenticates by checking if the computed hash matches the stored hash: \(h' = H(\text{pwd'} || \text{salt})\)

**Why does it help?**
Salt & Pepper

User

ID, password

Server

Signup

ID

Salt \leftarrow \{0, 1\}^{128}

salt

h = H(password || salt)

pepper \leftarrow \{0, 1\}^{13}

h^* = H(h || pepper)

Store (ID, Salt, h^*)

Login?

Why does it help?
Slow Hash Functions

- Computation-heavy hash function
  - Compose SHA256 in a certain way.

Application-Specific Integrated Circuit (ASIC) → blockchain mining

- Memory-hard hash functions
  - Scrypt
Two-Factor Authentication (2FA)

User

ID, password

Server

ID

Signup

Salt ← {0, 1}^s

h = H(password || salt)

h

1. SMS
2. app-generated code

h^* = H(h || pepper)

Store (ID, salt, h^*)

How would you design it?
Putting it Together: Secure Authentication

Server

Public (X.509 certificate)

\( (VK_s, SK_s) \leftarrow \text{Gen}(1^\lambda) \)

Alice

\( (VK_A, SK_A) \leftarrow \text{Gen}(1^\lambda) \)

Cert_A

Bob

\( (VK_B, SK_B) \leftarrow \text{Gen}(1^\lambda) \)

Cert_B

Authenticated Key Exchange

Login

Signup

Login

Signup
One-Sided Secure Authentication

\( (VK_s, SK_s) \leftarrow \text{Gen}(1^\lambda) \)

\[ g^a \]

\[ g^b, b_s \]

\[ g^{ab} \]

\[ \text{Hash} \]

\[ K \]

\[ (K_1, K_2) \]

\[ (VK_A, SK_A) \leftarrow \text{Gen}(1^\lambda) \]

\[ VK_A \]

\[ \text{Cert}_A \]
Two-Sided Authenticated Key Exchange

\[(V_{KA}, S_{KA}) \leftarrow \text{Gen}(1^n); \quad \text{cert}_A\]

Alice

\[6_A \leftarrow \text{Sign}_{SK_A}(g^a)\]

diffie-hellman key exchange

\[\begin{align*}
g^a, 6_A & \overset{\text{cert}_A}{\rightarrow} \text{Diffie-Hellman Key Exchange} \\
g^a, 6_A & \leftarrow \text{Vrfy}_{VK_B}(g^a, 6_A) \overset{?}{=} 1
\end{align*}\]

Bob

\[6_B \leftarrow \text{Sign}_{SK_B}(g^b)\]

\[\begin{align*}
g^{ab} & \overset{\text{cert}_B}{\leftarrow} \\
\text{Vrfy}_{VK_B}(g^b, 6_B) \overset{?}{=} 1 & \quad g^{ab} \\
\downarrow \text{Hash} & \\
K & \\
(k_1, k_2) & \rightarrow \text{Authenticated Encryption}
\]

\[(\text{Encrypt-then-MAC})\]
Secure Messaging

Server

public (X.509 certificate)

(VKs, SKs) \leftarrow \text{Gen}(1^n)

Alice

Bob

How would you design it?
Group Chat?

Server

public (X.509 certificate)

(Vks, SKs) ← Gen(1^n)

\[ A \]
\[ 7 \]

\[ A \]
\[ 7 \]

Signal, Login

Alice

Bob

Charlie

How would you design it?
Figure 3. Schematic depiction of Signal’s traffic, generated for a message $m$ from sender $A$ to receivers $B$ and $C$ in group $gr$ with $G_{gr} = \{A, B, C\}$. Transport layer protection is not in the analysis scope (gray).
Figure 5. Schematic depiction of traffic, generated for a message \( m \) from sender \( A \) to receivers \( B, C \) in group \( gr \) with \( \mathcal{G}_{gr} = \{A, B, C\} \) in WhatsApp.
Single Sign-On (SSO) Authentication

User

Password-Based Authentication

Server

Request "token"

"token" (Signature / MAC)

Service Provider

"token"

- OAuth / OpenID: Sign-in with Google/Apple/...
- Kerberos: enterprises