CSE107: Intro to Modern Cryptography

https://cseweb.ucsd.edu/classes/sp22/cse107-a/

Emmanuel Thomé

May 19, 2022
Lecture 14b

Passwords and password-authenticated key exchange

Recap from Tuesday

Passwords and PAKE
Plan

Recap from Tuesday

Passwords and PAKE
Forward secrecy

Definition (Forward Secrecy)

Forward secrecy asks that exposure of $sk[B]$ does not allow recovery of session keys $K$ exchanged prior to the time of exposure.

FS is achieved using the DH key exchange inside the session key exchange protocol.

Forward secrecy is considered necessary in modern session key exchange, and is present in the TLS 1.3 protocol.

Session-key exchange protocols using DH for forward secrecy are often called authenticated DH key exchange protocols.
Protocol KE3

Let $G = \langle g \rangle$ be a cyclic group of order $m$ in which the CDH problem is hard.

Here $a, b \leftarrow \mathbb{Z}_m$ are chosen by $A, B$, respectively, and $g^a, g^b$ play the role of nonces.

$\text{Sig}_B(X)$ is $B$’s signature on $X$, computed under $sk[B]$ and verifiable under the $pk[B]$ that is in $\text{CERT}[B]$.

Let $L = g^{ab}$ be the DH key. Then session key is $K = H_1(A \Vert B \Vert g^a \Vert g^b \Vert L)$ and MAC key is $M = H_2(A \Vert B \Vert g^a \Vert g^b \Vert L)$ where $H_1, H_2$ are as before.
Protocol KE3

There is no public-key encryption used here, only signatures.

Compromise of $sk[B]$ only gives $E$ the ability to forge signatures. Even given $sk[B]$, it cannot recover the DH key $L = g^{ab}$ from a prior exchange, and thus cannot distinguish from random the session key $K = H_1(A||B||g^a||g^b||L)$.

Accordingly this provides forward secrecy.

This is roughly the core of the unilateral session-key exchange in the TLS 1.3 handshake.
Recap from Tuesday

Passwords and PAKE
A password is a human-memorizable key.

Attackers can form a set $D$ of possible passwords called a dictionary such that

- If the target password $pwd$ is in $D$, and also
- The attacker knows $pwd = f(pwd)$, the image of $pwd$ under some public function $f$,

then the target password $pwd$ can be found via:

For all $pwd' \in D$ do
  If $f(pwd') = pwd$ then return $pwd'$

This is called a dictionary, or brute-force, attack.
Password usage

Passwords are in widespread use for client authentication to Internet services and servers like gmail, Amazon, Internet banking, ...

Most of us have more passwords than we can remember.

Passwords are communicated over TLS. The main threat is dictionary attacks arising from the adversary obtaining the image $pwd = f(pwd)$ of the target password $pwd$ under some public function $f$.

Studies show that many users select poor passwords, meaning ones that fall into attacker dictionaries. And attackers get better and better at making dictionaries. So preventing dictionary attacks is important for security.
In 2016, the 25 most common passwords made up more than 10% of surveyed passwords, with the most common making up 4%.

### Top 25 most common passwords by year according to SplashData

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>password</td>
<td>password</td>
<td>123456</td>
<td>123456</td>
<td>123456</td>
<td>123456</td>
<td>123456</td>
<td>123456</td>
</tr>
<tr>
<td>2</td>
<td>123456</td>
<td>123456</td>
<td>password</td>
<td>password</td>
<td>password</td>
<td>password</td>
<td>password</td>
<td>password</td>
</tr>
<tr>
<td>3</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
<td>12345678</td>
</tr>
<tr>
<td>4</td>
<td>qwerty</td>
<td>abc123</td>
<td>qwerty</td>
<td>12345678</td>
<td>qwerty</td>
<td>12345678</td>
<td>qwerty</td>
<td>12345678</td>
</tr>
<tr>
<td>5</td>
<td>abc123</td>
<td>qwerty</td>
<td>abc123</td>
<td>qwerty</td>
<td>12345678</td>
<td>qwerty</td>
<td>12345678</td>
<td>qwerty</td>
</tr>
<tr>
<td>6</td>
<td>monkey</td>
<td>monkey</td>
<td>123456789</td>
<td>123456789</td>
<td>123456789</td>
<td>qwerty</td>
<td>123456789</td>
<td>111111</td>
</tr>
<tr>
<td>7</td>
<td>1234567</td>
<td>letmein</td>
<td>111111</td>
<td>111111</td>
<td>1234</td>
<td>football</td>
<td>1234567890</td>
<td>letmein</td>
</tr>
<tr>
<td>8</td>
<td>letmein</td>
<td>dragon</td>
<td>1234567</td>
<td>baseball</td>
<td>1234</td>
<td>1234567</td>
<td>1234567</td>
<td>sunshine</td>
</tr>
<tr>
<td>9</td>
<td>trustno1</td>
<td>111111</td>
<td>iloveyou</td>
<td>dragon</td>
<td>1234567</td>
<td>princess</td>
<td>football</td>
<td>qwerty</td>
</tr>
<tr>
<td>10</td>
<td>dragon</td>
<td>baseball</td>
<td>adobe123[a]</td>
<td>football</td>
<td>baseball</td>
<td>1234</td>
<td>iloveyou</td>
<td>iloveyou</td>
</tr>
<tr>
<td>11</td>
<td>baseball</td>
<td>iloveyou</td>
<td>123123</td>
<td>1234567</td>
<td>welcome</td>
<td>login</td>
<td>admin</td>
<td>princess</td>
</tr>
<tr>
<td>12</td>
<td>111111</td>
<td>trustno1</td>
<td>admin</td>
<td>monkey</td>
<td>1234567890</td>
<td>welcome</td>
<td>welcome</td>
<td>admin</td>
</tr>
<tr>
<td>13</td>
<td>iloveyou</td>
<td>1234567</td>
<td>1234567890</td>
<td>letmein</td>
<td>abc123</td>
<td>solo</td>
<td>monkey</td>
<td>welcome</td>
</tr>
<tr>
<td>14</td>
<td>master</td>
<td>sunshine</td>
<td>letmein</td>
<td>abc123</td>
<td>111111</td>
<td>abc123</td>
<td>login</td>
<td>666666</td>
</tr>
<tr>
<td>15</td>
<td>sunshine</td>
<td>master</td>
<td>photoshop[a]</td>
<td>111111</td>
<td>1qaz2wsx</td>
<td>admin</td>
<td>abc123</td>
<td>abc123</td>
</tr>
<tr>
<td>16</td>
<td>ashley</td>
<td>123123</td>
<td>1234</td>
<td>mustang</td>
<td>dragon</td>
<td>121212</td>
<td>starwars</td>
<td>football</td>
</tr>
<tr>
<td>17</td>
<td>bailey</td>
<td>welcome</td>
<td>monkey</td>
<td>access</td>
<td>master</td>
<td>flower</td>
<td>123123</td>
<td>123123</td>
</tr>
<tr>
<td>18</td>
<td>password</td>
<td>shadow</td>
<td>shadow</td>
<td>shadow</td>
<td>monkey</td>
<td>passw0rd</td>
<td>dragon</td>
<td>monkey</td>
</tr>
<tr>
<td>19</td>
<td>shadow</td>
<td>ashley</td>
<td>sunshine</td>
<td>master</td>
<td>letmein</td>
<td>dragon</td>
<td>passw0rd</td>
<td>654321</td>
</tr>
<tr>
<td>20</td>
<td>123123</td>
<td>football</td>
<td>12345</td>
<td>michael</td>
<td>login</td>
<td>sunshine</td>
<td>master</td>
<td>!@#$%^&amp;*</td>
</tr>
<tr>
<td>21</td>
<td>654321</td>
<td>jesus</td>
<td>password1</td>
<td>superman</td>
<td>princess</td>
<td>master</td>
<td>hello</td>
<td>charlie</td>
</tr>
<tr>
<td>22</td>
<td>superman</td>
<td>michael</td>
<td>princess</td>
<td>696969</td>
<td>qwertyuiop</td>
<td>hottie</td>
<td>freedom</td>
<td>aa123456</td>
</tr>
<tr>
<td>23</td>
<td>qazwsx</td>
<td>ninja</td>
<td>azerty</td>
<td>123123</td>
<td>solo</td>
<td>lovene</td>
<td>whatever</td>
<td>donald</td>
</tr>
<tr>
<td>24</td>
<td>michael</td>
<td>mustang</td>
<td>trustno1</td>
<td>batman</td>
<td>passw0rd</td>
<td>zaq1zaq1</td>
<td>qazwsx</td>
<td>password1</td>
</tr>
<tr>
<td>25</td>
<td>Football</td>
<td>password1</td>
<td>000000</td>
<td>trustno1</td>
<td>starwars</td>
<td>password1</td>
<td>trustno1</td>
<td>qwerty123</td>
</tr>
</tbody>
</table>
Over the years, recommendations about “password strength” have become ubiquitous.

- “your password must include uppercase and lowercase letters, digits, two punctuation symbols”, etc.
- and “you must change your password every 12 months”.
通过20年的努力，我们成功地训练了每个人使用对人类来说难以记住，但对计算机来说容易猜测的密码。
NIST SP800-63B, revised in 2020 (§5.1.1.2 Memorized Secret Verifiers).

- Verifiers *SHOULD NOT* impose other composition rules (e.g., requiring mixtures of different character types or prohibiting consecutively repeated characters) for memorized secrets.
- Verifiers *SHOULD NOT* require memorized secrets to be changed arbitrarily (e.g., periodically). However, verifiers *SHALL* force a change if there is evidence of compromise of the authenticator.
- Verifiers *SHOULD* permit claimants to use “paste” functionality when entering a memorized secret. This facilitates the use of password managers, which are widely used and in many cases increase the likelihood that users will choose stronger memorized secrets.

Password managers: lastpass, keepass, bitwarden, pass, …
A protocol for Password Authenticated Key Exchange (PAKE) assumes client $A$ has a password $pwd$ and server $B$ has either $pwd$ or its hash under a public hash function.

The parties interact to arrive at a common session key $K$ satisfying authenticity, secrecy, forward secrecy and also security against off-line dictionary attacks.

This means the protocol never reveals an image $pwd = f(pwd)$ of $pwd$ under a public function $f$. So even if the password is in the dictionary, the off-line dictionary attack is infeasible.

Roughly, one adversary interaction with one of the parties can eliminate at most one candidate password from the dictionary.

Authentication here is mutual, and no PKI / certificates are assumed.
Protocol KE4

Client $A$ has password pwd that is known to server $B$.

Let $L = g^{ab}$ be the DH key. Then the session key and MAC keys are $K = H_1(A || B || g^a || g^b || L || pwd)$ and $M = H_2(A || B || g^a || g^b || L || pwd)$, respectively.

Is this secure against dictionary attack?
Protocol KE4

A, g^a

B, g^b, MAC_M(1||A||B||g^a||g^b)

MAC_M(0||A||B||g^a||g^b)

A successful dictionary attack by adversary E is possible, as follows:

E has A, B, g^a, g^b and also L = g^{ab} = (g^b)^a. Let

f(pwd) = MAC_{H_2}(A||B||g^a||g^b||L||pwd)(A||B||g^a||g^b).

This f is a public function of the password, allowing E to mount the dictionary attack.
History and status of PAKE

The first protocols were by Bellovin and Merrit, 1992.
Definitions and proven-secure protocols begin with [BPR00].
Large literature.
A representative modern PAKE protocol is OPAQUE [JKX18].
CSE107: Intro to Modern Cryptography

https://cseweb.ucsd.edu/classes/sp22/cse107-a/

Emmanuel Thomé

May 19, 2022
Lecture 15a

Advanced primitives and protocols

Commitment schemes

Homomorphic encryption
A large body of work on cryptography for goals beyond secure communication.

Usually concerned with privacy in broader settings.

Encompasses computing on encrypted data, secure two- and multi-party computation protocols, zero-knowledge ...

We start with safes and commitment schemes ...
Plan

Commitment schemes

Homomorphic encryption
Safes and their properties

**Hiding:** Without key $K$, one cannot recover the content of locked safe $C$.

**Binding:** A single, locked safe $C$ cannot admit two keys $K_1, K_2$ that open it to reveal different content $M_1, M_2$. 

---

Alice knows the secret combination / key 92093. 

*Combination safe*
Commitment schemes

Commitment schemes are, at first cut, a cryptographic (mathematical, digital, ...) way to realize safes.

To be more accurate, a commitment scheme is a cryptographic primitive whose definition formalizes requirements called hiding and binding. A safe is a rough physical analogy, or metaphor, for a commitment scheme.

As with all metaphors, it has its limits, so try to understand commitment schemes via the definitions rather than solely via the metaphor.

Zen saying: The finger pointing at the moon is not the moon ...

Commitment schemes are used in many protocols, including zero-knowledge protocols.
Syntax of a Commitment Scheme

A commitment scheme $CS = (P, C, V)$ is a triple of algorithms that operate as follows:

- $\pi \leftarrow P$ — a trusted party runs the parameter generation algorithm $P$ to generate public parameters $\pi$
- $(K, C) \leftarrow C_\pi(M)$ — apply commitment algorithm $C$ to message $M$ to obtain a commitment $C$ to $M$ along with a decommitment (or opening) key $K$.
- $d \leftarrow V_\pi(C, M, K)$ — apply verification algorithm $V$ to commitment $C$, candidate message $M$ and key $K$ to obtain a decision $d \in \{0, 1\}$ as to whether $C$ is a commitment to $M$.

The correctness requirement is that, for all $\pi$ that may be output by $P$, and all messages $M$ from the underlying message space, we have $d = 1$ with probability 1 when $(K, C) \leftarrow C_\pi(M)$ and $d \leftarrow V_\pi(C, M, K)$. 
Hiding security

Let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be a commitment scheme.

**Game $\text{HIDE}_{\mathcal{CS}}$**

**procedure Initialize**

$\pi \leftarrow \mathcal{P}; \ b \leftarrow \{0, 1\}$

return $\pi$

**procedure LR($M_0$, $M_1$)**

$(K, C) \leftarrow \mathcal{C}_\pi(M_b)$

return $C$

**procedure Finalize($b'$)**

return $(b = b')$

**Definition (hiding-advantage)**

The hiding-advantage of an adversary $A$ is

$$\text{Adv}^{\text{HIDE}}_{\mathcal{CS}}(A) = 2 \cdot \text{Pr}[\text{HIDE}_{\mathcal{CS}}^A \Rightarrow \text{true}] - 1.$$ 

Hiding security asks that an adversary having $C$ but not $K$ should not learn even partial information about the message $M$. 
Binding security

Let $CS = (P, C, V)$ be a commitment scheme.

**Game $\text{BIND}_{CS}$**

**procedure Initialize**

$\pi \leftarrow P$

return $\pi$

**procedure Finalize** $(C, M_0, M_1, K_0, K_1)$

$v_0 \leftarrow V_\pi(C, M_0, K_0)$

$v_1 \leftarrow V_\pi(C, M_1, K_1)$

return $(v_0 = 1)$ and $(v_1 = 1)$ and $(M_0 \neq M_1)$

**Definition (binding-advantage)**

The binding-advantage of an adversary $A$ is

$$\text{Adv}_{CS}^{\text{BIND}}(A) = \Pr \left[ \text{BIND}_{CS}^A \Rightarrow \text{true} \right].$$

Binding security asks that an adversary be unable to create a commitment $C$ that it can open to two different messages.
Commitment from symmetric encryption?

Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an IND-CPA-secure symmetric encryption scheme and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

<table>
<thead>
<tr>
<th>$\text{Alg } \mathcal{P}$</th>
<th>$\text{Alg } C_\pi(M)$</th>
<th>$\text{Alg } V_\pi(C, M, K)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi \leftarrow \varepsilon$</td>
<td>$K \leftarrow $ \mathcal{K} \ ; \ C \leftarrow $ \mathcal{E}_K(M)$</td>
<td>if $\mathcal{D}_K(C) = M$ then return 1</td>
</tr>
<tr>
<td>return $\pi$</td>
<td>return $(K, C)$</td>
<td>else return 0</td>
</tr>
</tbody>
</table>

Q: Is this hiding?
Commitment from symmetric encryption?

Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an IND-CPA-secure symmetric encryption scheme and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

\[
\begin{align*}
\textbf{Alg P} & \quad \pi \leftarrow \varepsilon \quad \text{return } \pi \\
\textbf{Alg } C_{\pi}(M) & \quad K \leftarrow \mathcal{K} ; C \leftarrow \mathcal{E}_K(M) \quad \text{return } (K, C) \\
\textbf{Alg } V_{\pi}(C, M, K) & \quad \text{if } \mathcal{D}_K(C) = M \text{ then return } 1 \\
& \quad \text{else return } 0
\end{align*}
\]

\textbf{Q:} Is this hiding?

\textbf{YES,} since $\mathcal{SE}$ is IND-CPA.
Commitment from symmetric encryption?

Let \( SE = (K, E, D) \) be an IND-CPA-secure symmetric encryption scheme and let \( CS = (P, C, V) \) be the following commitment scheme:

\[
\text{Alg } P
\]

\[ \pi \leftarrow \varepsilon \]

\[ \text{return } \pi \]

\[
\text{Alg } C_{\pi}(M)
\]

\[ K \leftarrow^$ K ; C \leftarrow^$ E_K(M) \]

\[ \text{return } (K, C) \]

\[
\text{Alg } V_{\pi}(C, M, K)
\]

\[ \text{if } D_K(C) = M \text{ then return } 1 \]

\[ \text{else return } 0 \]

Q: Is this binding?
Commitment from symmetric encryption?

Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an IND-CPA-secure symmetric encryption scheme and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

\[
\text{Alg } \mathcal{P} \quad \pi \leftarrow \varepsilon \quad \text{return } \pi
\]
\[
\text{Alg } C_\pi(M) \quad K \leftarrow^\$ \mathcal{K} ; C \leftarrow^\$ \mathcal{E}_K(M) \quad \text{return } (K, C)
\]
\[
\text{Alg } V_\pi(C, M, K) \quad \text{if } \mathcal{D}_K(C) = M \text{ then return } 1 \quad \text{else return } 0
\]

Q: Is this binding?

Not necessarily. For schemes like CTR$ or CBC$, the following adversary will have high binding advantage:

\[
\text{adversary } A(\pi) \quad K_0, K_1 \leftarrow^\$ \{0, 1\}^k ; M_0 \leftarrow^\$ \{0, 1\}^L ; C \leftarrow^\$ \mathcal{E}_{K_0}(M_0) ; M_1 \leftarrow \mathcal{D}_{K_1}(C) \quad \text{return } (C, M_0, M_1, K_0, K_1)
\]
Commitment from symmetric encryption?

Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an IND-CPA-secure symmetric encryption scheme and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

\[
\begin{array}{l}
\text{Alg } \mathcal{P} \\
\pi \leftarrow \varepsilon \\
\text{return } \pi
\end{array} \quad \begin{array}{l}
\text{Alg } C_\pi(M) \\
K \leftarrow \$ \mathcal{K} ; C \leftarrow \$ \mathcal{E}_K(M) \\
\text{return } (K, C)
\end{array} \quad \begin{array}{l}
\text{Alg } V_\pi(C, M, K) \\
\text{if } \mathcal{D}_K(C) = M \text{ then return 1} \\
\text{else return 0}
\end{array}
\]

Q: Is this binding if we additionally assume $\mathcal{SE}$ is INT-CTXT-secure?
Commitment from symmetric encryption?

Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an IND-CPA-secure symmetric encryption scheme and let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

\[
\begin{align*}
\textbf{Alg} \; \mathcal{P} & : \pi \leftarrow \varepsilon \\
& \text{return } \pi \\
\textbf{Alg} \; \mathcal{C}_\pi (M) & : K \leftarrow \$ \mathcal{K} ; C \leftarrow \$ \mathcal{E}_K (M) \\
& \text{return } (K, C) \\
\textbf{Alg} \; \mathcal{V}_\pi (C, M, K) & : \text{if } \mathcal{D}_K (C) = M \text{ then return } 1 \\
& \text{else return } 0
\end{align*}
\]

Q: Is this binding if we additionally assume $\mathcal{SE}$ is INT-CTXT-secure?

The above attack may no longer work. But the answer to the above question is NO.

If $\mathcal{SE}$ is robust [ABN10, FLPQ13, FOR17] or committing [GLR17] then $\mathcal{CS}$ will be binding.
Commitment from hashing

Let $H: \{0, 1\}^* \rightarrow \{0, 1\}^{\ell}$ be a collision-resistant hash function and let $CS = (\mathcal{P}, C, V)$ be the following commitment scheme:

**Alg $\mathcal{P}$**
- $\pi \leftarrow \varepsilon$
- return $\pi$

**Alg $C^H_\pi(M)$**
- $C \leftarrow H(M)$
- $K \leftarrow M$
- return $(K, C)$

**Alg $V^H_\pi(C, M, K)$**
- If $(C = H(M)) \text{ and } (M = K)$
  - then return 1
- Else return 0

**Q**: Is this binding?
Commitment from hashing

Let $H : \{0, 1\}^* \to \{0, 1\}^\ell$ be a collision-resistant hash function and let $CS = (P, C, V)$ be the following commitment scheme:

<table>
<thead>
<tr>
<th>$\text{Alg } P$</th>
<th>$\text{Alg } C^H(M)$</th>
<th>$\text{Alg } V^H_\pi(C, M, K)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi \leftarrow \varepsilon$</td>
<td>$C \leftarrow H(M)$</td>
<td>If $((C = H(M))$ and $(M = K))$</td>
</tr>
<tr>
<td>return $\pi$</td>
<td>$K \leftarrow M$</td>
<td>then return 1</td>
</tr>
<tr>
<td></td>
<td>return $(K, C)$</td>
<td>Else return 0</td>
</tr>
</tbody>
</table>

Q: Is this binding?

YES, since $H$ is collision resistant.
Commitment from hashing

Let $H: \{0, 1\}^* \rightarrow \{0, 1\}^\ell$ be a collision-resistant hash function and let $CS = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be the following commitment scheme:

<table>
<thead>
<tr>
<th>Alg $\mathcal{P}$</th>
<th>Alg $C^H_\pi(M)$</th>
<th>Alg $V^H_\pi(C, M, K)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi \leftarrow \varepsilon$</td>
<td>$C \leftarrow H(M)$</td>
<td>If $((C = H(M))$ and $(M = K))$ then return 1</td>
</tr>
<tr>
<td>return $\pi$</td>
<td>$K \leftarrow M$</td>
<td>Else return 0</td>
</tr>
<tr>
<td></td>
<td>return $(K, C)$</td>
<td></td>
</tr>
</tbody>
</table>

Q: Is this hiding?
Commitment from hashing

Let $H: \{0, 1\}^* \rightarrow \{0, 1\}^\ell$ be a collision-resistant hash function and let $CS = (P, C, V)$ be the following commitment scheme:

<table>
<thead>
<tr>
<th>Alg $P$</th>
<th>Alg $C^H(M)$</th>
<th>Alg $V^H_{\pi}(C, M, K)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi \leftarrow \varepsilon$</td>
<td>$C \leftarrow H(M)$</td>
<td>If (($C = H(M)$) and ($M = K$)) then return 1</td>
</tr>
<tr>
<td>return $\pi$</td>
<td>$K \leftarrow M$</td>
<td>Else return 0</td>
</tr>
<tr>
<td></td>
<td>return $(K, C)$</td>
<td></td>
</tr>
</tbody>
</table>

**Q:** Is this hiding?

**NO,** since $C$ is deterministic. Specifically, the following adversary $A$ has $Adv^\text{HIDE}_{CS}(A) = 1$:

**adversary** $A(\pi)$

$C_1 \leftarrow LR(0, 1)$; $C_2 \leftarrow LR(1, 1)$

If ($C_1 = C_2$) then return 1 else return 0
Commitment from hashing

Let \( H : \{0, 1\}^* \rightarrow \{0, 1\}^\ell \) and let \( CS = (\mathcal{P}, C, \mathcal{V}) \) be the following commitment scheme:

\[
\textbf{Alg } \mathcal{P} \\
\pi \leftarrow \varepsilon \\
\text{return } \pi
\]

\[
\textbf{Alg } C^H_\pi(M) \\
K \leftarrow \{0, 1\}^\ell \\
C \leftarrow H(K \| M) \\
\text{return } (K, C)
\]

\[
\textbf{Alg } \mathcal{V}^H_\pi(C, M, K) \\
\text{If } (C = H(K \| M)) \text{ then return 1} \\
\text{Else return 0}
\]

This is binding if \( H \) is collision-resistant (CR). Note: \( \ell \) must be fixed!

One can give an example of CR \( H \) such that it is not hiding. But for “real" \( H \) such as SHA256 it seems to be hiding in the sense that no attacks are known.
Commitment from DL

Let $G = \langle g \rangle$ be a cyclic group whose order $m$ is prime. Let $H: \{0, 1\}^* \rightarrow \mathbb{Z}_m$ and let $CS = (P, C, V)$ be the following commitment scheme:

**Alg $P$**

$x \leftarrow^$ $\mathbb{Z}_m$

$h \leftarrow g^x$

return $h$

**Alg $C_h(M)$**

$K \leftarrow^$ $\mathbb{Z}_m$

$C \leftarrow g^{H(M)} h^K$

return $(K, C)$

**Alg $V_h(C, M, K)$**

If $(C = g^{H(M)} h^K)$ then return 1

Else return 0

This is binding if DL is hard in $G$ and $H$ is collision-resistant (CR).

This is unconditionally hiding, meaning $\text{Adv}_{CS}^{\text{HIDE}}(A) = 0$ for all $A$. 

Commitment from DL

Let \( G = \langle g \rangle \) be a cyclic group whose order \( m \) is prime. Let \( H : \{0, 1\}^* \rightarrow \mathbb{Z}_m \) and let \( CS = (P, C, V) \) be the following commitment scheme:

<table>
<thead>
<tr>
<th>( \text{Alg} \ P )</th>
<th>( \text{Alg} \ C_h^H(M) )</th>
<th>( \text{Alg} \ V_h^H(C, M, K) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x \leftarrow^$ \mathbb{Z}_m )</td>
<td>( K \leftarrow^$ \mathbb{Z}_m )</td>
<td>If ( C = g^{H(M)} h^K ) then return 1</td>
</tr>
<tr>
<td>( h \leftarrow g^x )</td>
<td>( C \leftarrow g^{H(M)} h^K )</td>
<td>Else return 0</td>
</tr>
<tr>
<td>return ( h )</td>
<td>return ( (K, C) )</td>
<td></td>
</tr>
</tbody>
</table>

The Pedersen commitment scheme [Pe91] is the special case where the message space is \( \mathbb{Z}_m \) and \( H(M) = M \).

The Pedersen scheme is **homomorphic**: If \( C_1 = g^{M_1} h^{K_1} \) is a commitment to \( M_1 \) and \( C_2 = g^{M_2} h^{K_2} \) is a commitment to \( M_2 \) then \( C_1 C_2 = g^M h^K \) is a commitment to \( M = (M_1 + M_2) \mod m \), with \( K = (K_1 + K_2) \mod m \).
Commitment from DL

Let $G = \langle g \rangle$ be a cyclic group whose order $m$ is prime. Let $H: \{0, 1\}^* \rightarrow \mathbb{Z}_m$ and let $CS = (P, C, V)$ be the following commitment scheme:

**Alg** $P$

$x \leftarrow^$ $\mathbb{Z}_m$

$h \leftarrow g^x$

return $h$

**Alg** $C_h(M)$

$K \leftarrow^$ $\mathbb{Z}_m$

$C \leftarrow g^{H(M)} h^K$

return $(K, C)$

**Alg** $V_h^H(C, M, K)$

If $(C = g^{H(M)} h^K)$ then return 1
Else return 0

The Pedersen commitment scheme [Pe91] is the special case where the message space is $\mathbb{Z}_m$ and $H(M) = M$.

The Pedersen scheme is *homomorphic*: If $C_1 = g^{M_1} h^{K_1}$ is a commitment to $M_1$ and $C_2 = g^{M_2} h^{K_2}$ is a commitment to $M_2$ then $C_1 C_2 = g^{M} h^{K}$ is a commitment to $M = (M_1 + M_2) \mod m$, with $K = (K_1 + K_2) \mod m$.

What is $x$ good for?
Let $G = \langle g \rangle$ be a cyclic group whose order $m$ is prime. Let $H: \{0, 1\}^* \rightarrow \mathbb{Z}_m$ and let $CS = (P, C, V)$ be the following commitment scheme:

<table>
<thead>
<tr>
<th>$\text{Alg } P$</th>
<th>$\text{Alg } C^H(M)$</th>
<th>$\text{Alg } V^H_H(C, M, K)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x \leftarrow^$ $\mathbb{Z}_m$</td>
<td>$K \leftarrow^$ $\mathbb{Z}_m$</td>
<td>If $(C = g^{H(M)}h^K)$ then return 1</td>
</tr>
<tr>
<td>$h \leftarrow g^x$</td>
<td>$C \leftarrow g^{H(M)}h^K$</td>
<td>Else return 0</td>
</tr>
<tr>
<td>return $h$</td>
<td>return $(K, C)$</td>
<td></td>
</tr>
</tbody>
</table>

What is $x$ good for?

Nobody knows $x$. A party that does know $x$ can easily forge commitments (win the binding game).

Exercise: given two messages $M$ and $M'$, show that an adversary that knows $x$ can compute $K$ and $K'$ such that $C$ is a commitment to both $(K, M)$ and $(K', M')$. 
Flipping a common coin

Alice and Bob are getting divorced. They want to flip a common, fair coin $c$ whose outcome decides which of them keeps the waffle maker.

The naive protocol is for $A$ to flip the coin $c$ and send it to $B$:

$\begin{align*}
A & \leftarrow \{0, 1\} \\
B & \\
\end{align*}$

But this allows $A$ to dictate the outcome. Unsurprisingly, she gets the waffle maker.
Flipping a common coin

Let $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$ be a commitment scheme and consider the following protocol to flip a common coin $c$:

![Protocol Diagram]

The hiding security of $\mathcal{CS}$ means that $B$ cannot dictate the outcome $c$.

The binding security of $\mathcal{CS}$ means that $A$ cannot dictate the outcome $c$. 
Plan

Commitment schemes

Homomorphic encryption
Let $\mathcal{ES} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme, either symmetric or asymmetric.

We write $ek$ for the encryption key and $dk$ for the decryption key. In the symmetric case, they are the same.

We define the homomorphic evaluation key $hk$ to be $ek$ in the asymmetric case and $\varepsilon$ in the symmetric case.

Let $\mathcal{FC}$ be a set (class) of functions. We write $\langle f \rangle$ for a description, for example as a circuit, of a function $f \in \mathcal{FC}$. 
Homomorphic encryption

$\mathcal{HE}$ is a homomorphic evaluation algorithm for $\mathcal{ES} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ and $\mathcal{FC}$ if for all functions $f \in \mathcal{FC}$ and all messages $M_1, \ldots, M_m$, where $m$ is the number of inputs of $f$, the following returns true with probability 1:

For $i = 1, \ldots, m$ do $C_i \leftarrow \mathcal{E}_{ek}(M_i)$
$C \leftarrow \mathcal{HE}_{hk}(\langle f \rangle, C_1, \ldots, C_m)$; $M \leftarrow \mathcal{D}_{dk}(C)$
Return $(M = f(M_1, \ldots, M_m))$

That is, $C$ is an encryption of $f(M_1, \ldots, M_m)$.

Encryption scheme $\mathcal{ES}$ is homomorphic for the class of functions $\mathcal{FC}$ if there is an efficient homomorphic evaluation algorithm $\mathcal{HE}$ as above.

A fully homomorphic encryption (FHE) scheme is one that is homomorphic for the class $\mathcal{FC}$ of all functions.
Q: Isn’t homomorphic evaluation always possible, via

\[
\text{Alg } \mathcal{HE}_{hk}(\langle f \rangle, C_1, \ldots, C_m)
\]
For \( i = 1, \ldots, m \) do \( M_i \leftarrow D_{dk}(C_i) \)
\( M \leftarrow f(M_1, \ldots, M_m) \); \( C \leftarrow E_{ek}(M) \); Return \( C \)

A: \( \mathcal{HE} \) is not given \( dk \). And the requirement that \( \mathcal{HE} \) is efficient means that it is infeasible for it to compute \( dk \) from \( hk \).
The primary security requirement for a homomorphic encryption scheme $\mathcal{E}S$ is simply IND-CPA.

**Sometimes** one wants the scheme to be function hiding (FH), which means that, on seeing $C \leftarrow \$ \mathcal{H}\mathcal{E}_{hk}(\langle f \rangle, \mathcal{E}_{ek}(M_1), \ldots, \mathcal{E}_{ek}(M_m))$, one does not learn $f$. A game-based definition follows.

**Sometimes** one wants that homomorphically evaluated ciphertexts are distributed just like real ones, meaning the following are indistinguishable:

- $C \leftarrow \$ \mathcal{H}\mathcal{E}_{hk}(\langle f \rangle, \mathcal{E}_{ek}(M_1), \ldots, \mathcal{E}_{ek}(M_m))$
- $C' \leftarrow \$ \mathcal{E}_{ek}(f(M_1, \ldots, M_m))$. 
Let $\mathcal{E}S = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme, either symmetric or asymmetric. We define its **extended key-generation algorithm** $\overline{\mathcal{K}}$ via:

If $\mathcal{E}S$ is symmetric:

- $\overline{\mathcal{K}}(\mathcal{K})$:
  - $K \leftarrow^\$ $\mathcal{K}$
  - $ek \leftarrow K$; $dk \leftarrow K$; $hk \leftarrow \varepsilon$
  - Return $(ek, dk, hk)$

If $\mathcal{E}S$ is asymmetric:

- $\overline{\mathcal{K}}((ek, dk))$:
  - $(ek, dk) \leftarrow^\$ $\mathcal{K}$
  - $hk \leftarrow ek$
  - Return $(ek, dk, hk)$

This yields a unified syntax for symmetric and asymmetric schemes.
Let $\mathcal{ES} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme with homomorphic evaluation algorithm $\mathcal{HE}$ for the class of functions $\mathcal{FC}$. Let $A$ be an adversary.

Game $\text{FH}_{\mathcal{ES}, \mathcal{HE}}$

**procedure Initialize**

$b \leftarrow \{0, 1\} ; i \leftarrow 0 ;$
$(ek, dk, hk) \leftarrow \mathcal{K} ;$
Return $hk$

**procedure Finalize**($b'$)

return $(b = b')$

**procedure Enc($M$)**

$i \leftarrow i + 1 ; M_i \leftarrow M ;$
$C_i \leftarrow \mathcal{E}_{ek}(M) ;$
Return $C_i$

**procedure LR($i_1, \ldots, i_m, f_0, f_1$)**

$C \leftarrow \mathcal{HE}_{hk}(\langle f_b \rangle, C_{i_1}, \ldots, C_{i_m}) ;$
Return $C$
Function hiding security

In game $\text{FH}_{\mathcal{E}S, \mathcal{H}E}$, any LR query $i_1, \ldots, i_m, f_0, f_1$ must satisfy the following conditions:

- $f_0, f_1 \in \text{FC}$
- $m$ is the number of inputs of both $f_0$ and $f_1$
- $1 \leq i_1, \ldots, i_m \leq i$
- $|f_0(M_{i_1}, \ldots, M_{i_m})| = |f_1(M_{i_1}, \ldots, M_{i_m})|$.

**Definition (fh-advantage)**

The fh-advantage of $A$ is

$$\text{Adv}^{\text{fh}}_{\mathcal{E}S, \mathcal{H}E}(A) = 2 \cdot \Pr \left[ \text{FH}^A_{\mathcal{E}S, \mathcal{H}E} \Rightarrow \text{true} \right] - 1.$$ 

We (informally) say that $(\mathcal{E}S, \mathcal{H}E)$ is FH-secure for FC if, as usual, any practical adversary $A$ has low fh-advantage.
Homomorphic encryption can’t be IND-CCA

If an encryption scheme $\mathcal{ES} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is homomorphic for a non-trivial class of functions $\mathcal{FC}$ then it cannot be IND-CCA-secure.

Why? Assume the adversary $A$ can find $M_0, M_1$ and $f \in \mathcal{FC}$ such that

1. $f(M_0) \neq M_0$ and $f(M_1) \neq M_1$
2. $f(M_0) \neq f(M_1)$

Then it can achieve $\text{Adv}^{\text{ind-cca}}_{\mathcal{ES}}(A) = 1$ via:

<table>
<thead>
<tr>
<th>adversary $A(hk)$ // $hk = ek$ (asymmetric) or $hk = \varepsilon$ (symmetric)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C \leftarrow LR(M_0, M_1)$ ; $C' \leftarrow HE_{hk}(\langle f \rangle, C)$ ; $M' \leftarrow \text{Dec}(C')$</td>
</tr>
<tr>
<td>If $(M' = f(M_1))$ then return 1 else return 0</td>
</tr>
</tbody>
</table>

Condition (1) ensures $C' \neq C$ so the $\text{Dec}$-query is valid. Then (2) ensures that $A$’s output is correct.
Possible usage of homomorphic encryption

Homomorphic encryption allows computing on encrypted data.

A picks keys $ek, dk, hk$, encrypts her data $M_1, \ldots, M_m$ under $ek$ to get $C_1, \ldots, C_m$.

A uploads the ciphertexts and $hk$ to in-the-cloud server $B$.

Later $A$ can send $\langle f \rangle$ to $B$, who computes and returns $C \leftarrow \mathcal{H}E_{hk}(\langle f \rangle, C_1, \ldots, C_m)$.

$A$ now recovers $M = f(M_1, \ldots, M_m) \leftarrow \mathcal{D}_{dk}(C)$. 