18-330 Cryptography Notes: Hashes and Authentication

Note: This is provided as a resource and is not meant to include all material from lectures or recitations. The proofs shown, however, are good models for your homework and exams.

1 Hash Functions

Definition 1. A hash function is any deterministic function that maps arbitrary-length inputs to fixed-length outputs.

Definition 2. A cryptographic hash function (CHF) must provide at least one of the following (in order of strongest to weakest):

1. Random oracle
2. Collision resistance
3. Second pre-image resistance
4. Pre-image resistance (sometimes known as one-way)

Definition 3. Let \( h : \{0,1\}^* \rightarrow \{0,1\}^L \). A collision for \( h \) is a pair \( m_0, m_1 \in \{0,1\}^* \) such that \( h(m_0) = h(m_1) \) and \( m_0 \neq m_1 \).

1.1 Pre-Image Resistance

Definition 4. Let \( h : \{0,1\}^* \rightarrow \{0,1\}^L \). The pre-image resistance game is defined as follows:

1. The Challenger samples \( x \) from \( \{0,1\}^* \) uniformly at random.
2. The Challenger sends \( h(x) \) to the Adversary.
3. The Adversary runs some logic to output \( x' \in \{0,1\}^* \).

Definition 5. Let \( h : \{0,1\}^* \rightarrow \{0,1\}^L \), and let \( A \) be an efficient adversary. The pre-image resistance advantage is defined as:

\[
\text{Adv}_{\text{Pre}}[A,h,q] := Pr[h(x) = h(x')]
\]

Definition 6. Let \( h : \{0,1\}^* \rightarrow \{0,1\}^L \). \( h \) is pre-image resistant if for all efficient adversaries \( A \):

\[
\text{Adv}_{\text{Pre}}[A,h,q] < \epsilon
\]
1.2 Second Pre-Image Resistance

Definition 7. Let $h : \{0,1\}^* \rightarrow \{0,1\}^L$. The second pre-image resistance game is defined as follows:

1. The Challenger samples $x$ from $\{0,1\}^*$ uniformly at random.
2. The Challenger sends $(x, h(x))$ to the Adversary.
3. The Adversary runs some logic to output $x' \in \{0,1\}^*$.

Definition 8. Let $h : \{0,1\}^* \rightarrow \{0,1\}^L$, and let $A$ be an efficient adversary. The second pre-image resistance advantage is defined as:

$$\text{Adv}_{2\text{Pre}}[A, h, q] := \Pr[h(x) = h(x') \land x \neq x]$$

Definition 9. Let $h : \{0,1\}^* \rightarrow \{0,1\}^L$. $h$ is second pre-image resistant if for all efficient adversaries $A$:

$$\text{Adv}_{2\text{Pre}}[A, h, q] < \epsilon$$

1.3 Collision Resistance

Definition 10. A function $h$ is collision resistant if for all efficient algorithms $A$:

$$\text{Adv}_{\text{CR}}[A, h] = \Pr[A \text{ outputs collision for } h] < \epsilon$$

2 Merkle-Damgard Construction

Definition 11. Let $h$ be a one-way compression function. The Merkle-Damgard hash construction $H$ is roughly as follows (some details are implementation-defined):

<table>
<thead>
<tr>
<th>Algorithm 1: Merkle-Damgard construction $H$ (for fixed IV, blockSize, and h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 $state \leftarrow IV$</td>
</tr>
<tr>
<td>2 $m \leftarrow \text{input to the algorithm}$</td>
</tr>
<tr>
<td>3 $m' \leftarrow \text{pad}(m), \text{ where }</td>
</tr>
<tr>
<td>4 for $i \in [0, \text{numBlocks})$ do</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6 end</td>
</tr>
<tr>
<td>7 return $state$</td>
</tr>
</tbody>
</table>

Typically, the padding consists of a 1 bit, followed by 0 bits, and then 8 bytes that encode the length of the original message $m$. If $h$ is collision resistant, then so is $H$. 

3 Password Salts

**Enrollment:** store \(s\text{alt}||h(\text{password}||\text{salt})\)

**Verification:** extract \(\text{salt}\) and \(h(\text{password}||\text{salt})\) from stored file, check \(h(\text{input}||\text{salt}) == h(\text{password}||\text{salt})\)

Salts are unique to each user/password. They prevent brute forcing with pre-computed hashes.

4 Message Authentication Codes

MACs are used for message integrity. Intuitively speaking, the corruption of messages should be detectable.

A **Cyclic Redundancy Check** (CRC) is only a sanity check for detecting random errors, not malicious attacks.

**Definition 12. A Message Authentication Code (MAC) \(\text{MAC} = (S,V)\) defined over \((K,M,T)\) is a pair of algorithms:**

1. **Sign:** \(S(k,m)\) outputs \(t \in T\)
2. **Verify:** \(V(k,m,t)\) outputs ‘yes’ or ‘no’

**Correctness:** \(V(k,m,S(k,m)) = ‘yes’\)

4.1 Secure MAC Adversarial Game

**Definition 13.** Let \(I = (S,V)\) be a MAC defined over \((K,M,T)\). The **Secure MAC game** is defined as follows:

1. The Challenger generates a key \(k = \text{KeyGen}(l)\)
2. The Adversary selects \(m_1,..,m_q \in M\) and sends them to the Challenger.
3. The Challenger replies with \(S(k,m_1),..,S(k,m_q)\).
4. The Adversary runs some logic to select \(m\) and \(t\), and then sends \(m,t\) to the Challenger.
5. The Challenger checks: If \(m \in \{m_1,..,m_q\}\), output ‘no’.
6. The Challenger outputs \(V(k,m,t) \in \{yes,no\}\).

4.2 Secure MAC Advantage

Let \(I = (S,V)\) be a MAC defined over \((K,M,T)\), and let \(A\) be an adversary. We define \(A\)’s MAC advantage as:

\[
\text{Adv}_{\text{MAC}}[A,I] = \Pr[\text{Challenger outputs ‘yes’}]
\]
4.3 Secure MAC

Let \( I = (S, V) \) be a MAC defined over \((\mathcal{K}, \mathcal{M}, T)\). We say that \( I \) is a secure MAC if for all efficient adversaries \( A \):

\[
\text{Adv}_{MAC}[A, I] < \epsilon
\]

4.4 HMACs

Below is the HMAC algorithm. The differences from the Merkle-Damgard construction are highlighted.

\begin{algorithm}
\begin{algorithmic}
\State \( \text{state} \leftarrow IV \)
\State \( m \leftarrow \text{input to the algorithm} \)
\State \( m' \leftarrow (k \oplus \text{ipad})|| \text{pad}(m), \) where \( |m'| = \text{blockSize} \times \text{numBlocks} \)
\For{\( i \in [0, \text{numBlocks}) \)}
\State \( \text{state} \leftarrow h(\text{state}, m'[i]) \)
\EndFor
\State \( \text{pad}_\text{final} \leftarrow h(IV, k \oplus \text{opad}) \)
\State \( \text{return } h(\text{pad}_\text{final}, \text{state}) \)
\end{algorithmic}
\end{algorithm}

\[ S(k, m) = H((k \oplus \text{opad})||H((k \oplus \text{ipad})||m)) \]

5 Authenticated Encryption

5.1 Adversarial Game for Ciphertext Integrity

\textbf{Definition 14.} Let \( \mathcal{I} = (\text{KeyGen}, E, D) \) be a cipher. The \textbf{ciphertext integrity game} is defined as follows:

1. The experiment takes as input bit \( b \in \{0,1\} \), chosen uniformly at random.
2. The Challenger runs \( k \leftarrow \text{KeyGen}(\lambda) \) for security parameter \( \lambda \).
3. The Adversary runs some logic and selects a message \( m_i \in \mathcal{M} \) to send to the Challenger.
4. The Challenger responds with \( c_i = E(k, m_i) \).
5. Repeats steps 2 through 3 some polynomial \( q \) number of times.
6. The Adversary sends \( c \) to the Challenger.
7. The Challenger outputs \( b = 1 \) if \( D(k, c) \neq \bot \land c \notin \{c_1, \ldots, c_q\} \). Otherwise the Challenger outputs \( b = 0 \).
8. \( b \) is the outcome of the experiment.

5.2 Ciphertext Integrity Advantage

Let \( \mathcal{I} = (\text{KeyGen}, E, D) \) be a cipher, and let \( A \) be an adversary. We define \( A \)'s \textbf{ciphertext integrity advantage} as:

\[ \text{Adv}_{CI}[A, I] = \Pr[\text{Challenger outputs 1}] \]
5.3 Ciphertext Integrity

Let $I = (\text{KeyGen}, E, D)$ be a cipher. We say that $I$ has ciphertext integrity iff for all efficient adversaries $A$:

$$\text{Adv}_{CI}[A, I] < \epsilon$$

5.4 Authenticated Encryption

**Definition 15.** Let $I = (\text{KeyGen}, E, D)$ be a cipher where:

1. $E : \mathcal{K} \times \mathcal{M} \times \mathcal{N} \rightarrow \mathcal{C}$ (same as before)
2. $D : \mathcal{K} \times \mathcal{C} \times \mathcal{N} \rightarrow \mathcal{M} \cup \{\bot\}$

The decryption algorithm $D$ would return $\bot$ if the ciphertext is determined to be invalid. $I$ is said to provide authenticated encryption (AE) if it is:

1. IND-CPA secure
2. provides ciphertext integrity

6 IND-CCA Security

6.1 IND-CCA Adversarial Game

**Definition 16.** Let $E = (\text{KeyGen}, E, D)$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. The **IND-CCA game** is defined as follows:

1. The experiment takes as input bit $b \in \{0, 1\}$.
2. The Challenger runs $k \leftarrow \text{KeyGen}(\lambda)$.
3. The Adversary runs some logic and selects a $(m_{i,0}, m_{i,1})$ from $\mathcal{M} \times \mathcal{M}$.
4. The Challenger replies with $c_i = E(k, m_{i,b})$.
5. The Adversary sends $c$ to the Challenger, where $c \notin \{c_1, ..., c_i\}$
6. The Challenger replies with $m \leftarrow D(k, c)$
7. Repeat steps 3 through 6 some polynomial $q$ number of times.
8. The Adversary runs some logic and outputs $b \in \{0, 1\}$, which is the output of the experiment.

6.2 IND-CCA Advantage

**Definition 17.** Let $E = (\text{KeyGen}, E, D)$, and let $A$ be an adversary. We define $A$’s **IND-CCA advantage** as:

$$\text{Adv}_{CCA}[A, E] := \text{Pr}[\text{Exp}(1) = 1] - \text{Pr}[\text{Exp}(0) = 1]$$
6.3 IND-CCA Secure

Definition 18. Let $\mathcal{E} = (\text{KeyGen}, E, D)$. We say that $\mathcal{E}$ is IND-CCA secure if for all efficient adversaries $A$:

$$\text{Adv}_{\text{CCA}}[A, \mathcal{E}] < \epsilon$$

Claim: Authenticated encryption implies IND-CCA secure.