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

$$Adv_{Pre}[A, h, q] := Pr[h(x) = h(x')]$$

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

$$Adv_{Pre}[A, h, q] < \epsilon$$

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:*

$$Adv_{2Pre}[A, h, q] := Pr[h(x) = h(x') \wedge x \neq x]$$

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

$$Adv_{2Pre}[A, h, q] < \epsilon$$

1.3 Collision Resistance

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

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

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):*

Algorithm 1: Merkle-Damgard construction H (for fixed IV, blockSize, and h)

```
state ← IV
m ← input to the algorithm
m' ← pad(m), where |m'| = blockSize * numBlocks
for i ∈ [0, numBlocks) do
    state ← h(state, m'[i])
end
return state
```

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 $salt||h(password||salt)$
Verification: extract $salt$ and $h(password||salt)$ from stored file, check $h(input||salt) == h(password||salt)$

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) $MAC = (S, V)$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{T})$ is a pair of algorithms:*

1. *Sign: $S(k, m)$ outputs $t \in \mathcal{T}$*
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 $(\mathcal{K}, \mathcal{M}, \mathcal{T})$. The* ***Secure MAC game*** *is defined as follows:*

1. *The Challenger generates a key $k = KeyGen(l)$*
2. *The Adversary selects $m_1, .., m_q \in \mathcal{M}$ 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 $(\mathcal{K}, \mathcal{M}, \mathcal{T})$, and let A be an adversary. We define A's MAC advantage as:

$$Adv_{MAC}[A, I] = Pr[\text{Challenger outputs 'yes'}]$$

4.3 Secure MAC

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

$$Adv_{MAC}[A, I] < \epsilon$$

4.4 HMACs

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

Algorithm 2: HMAC H for a fixed IV, $ipad$, and $opad$

$state \leftarrow IV$
$m \leftarrow$ input to the algorithm
$m' \leftarrow (k \oplus ipad)||pad(m)$, where $|m'| = blockSize * numBlocks$
for $i \in [0, numBlocks)$ **do**
    $state \leftarrow h(state, m'[i])$
end
$pad_{final} \leftarrow h(IV, k \oplus opad)$
return $h(pad_{final}, state)$

$$S(k, m) = H((k \oplus opad)||H((k \oplus ipad)||m)$$

5 Authenticated Encryption

5.1 Adversarial Game for Ciphertext Integrity

Definition 14. *Let $\mathcal{I} = (KeyGen, E, D)$ be a cipher. The* ***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 $k \leftarrow KeyGen(\lambda)$ for security parameter λ.*
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 $D(k, c) \neq \perp \wedge c \notin \{c_1, .., c_q\}$. Otherwise the Challenger outputs $b = 0$.*
8. *b is the outcome of the experiment.*

5.2 Ciphertext Integrity Advantage

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

$$Adv_{CI}[A, I] = Pr[\text{Challenger outputs 1}]$$

5.3 Ciphertext Integrity

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

$$Adv_{CI}[A, I] < \epsilon$$

5.4 Authenticated Encryption

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

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

*The decryption algorithm D would return $\perp$ 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 $\mathcal{E} = (KeyGen, E, D)$ 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 $k \leftarrow KeyGen(\lambda)$.*
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 $\mathcal{E} = (KeyGen, E, D)$, and let A be an adversary. We define A's **IND-CCA advantage** as:*

$$Adv_{CCA}[A, \mathcal{E}] := Pr[Exp(1) = 1] - Pr[Exp(0) = 1]$$

6.3 IND-CCA Secure

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

$$Adv_{CCA}[A, \mathcal{E}] < \epsilon$$

Claim: Authenticated encryption implies IND-CCA secure.