[CS304] Introduction to Cryptography and Network Security

Course Instructor: Dr. Dibyendu Roy — Winter 2023-2024

Scribed by: Priyansh Vaishnav (202151120) — Lecture 12&13 (Week 7&Week 8)

LECTURE 12

Ideal Hash Function

Let $h : P \to S$ be a hash function. h will be called an Ideal Hash Function if, given $x \in P$, to find $h(x)$, either you have to apply h on x or look into a table corresponding to h (hash table).

$$\text{Pr[Pre-image finding]} \simeq \frac{Q}{M}$$
$$\text{Complexity of finding pre-image} = O(M)$$

Collison Finding Algorithm: -

$$h : X \to Y,\ |Y| = M$$
Find $x, x' \in X$ such that $x \neq x'$ and $h(x) = h(x')$.

Let $X_0 \subseteq X$, $|X_0| = Q$.

For each $x \in X_0$:

- Compute $y_x = h(x)$.
- If $y_x = y_{x'}$ for some $x \neq x'$, return (x, x').

Define events E_i: $h(x_i) \notin \{h(x_1), \dots, h(x_{i-1})\}$.

$P[E_1] = 1$.

$P[E_2 \mid E_1] = \frac{M-1}{M}$.

Continuing this process:

$$P[E_1 \cap E_2 \cap \ldots \cap E_Q] = \prod_{i=1}^{Q-1} \frac{M-i}{M}$$

Probability of success in collision finding:

$$P[\text{Success}] = 1 - \prod_{i=1}^{Q-1} \frac{M-i}{M} \simeq 1 - e^{-\frac{i}{M}\prod_{i=1}^{Q-1} i}$$

If Q is very large, then:

$$Q^2 \simeq 2M \cdot m \left(\frac{1}{1-\epsilon}\right)$$

Therefore:

$$Q = \sqrt{2m \cdot \frac{1}{1-\epsilon}} \cdot \sqrt{M}$$

The complexity is $O(\sqrt{M})$.

Secure Hash Function: -

A secure hash function is one that satisfies the following conditions:

- Complexity of finding the second preimage $= O(2^M)$
- Complexity of finding a collision $= O(2^{M/2})$

Compression Function; -

Let $h : \{0,1\}^{m+t} \to \{0,1\}^m$ be a compression function where $t \geq 1$.

Our objective is to construct $H : \{0,1\}^* \to \{0,1\}^*$. The security of H heavily relies on the security of h.

Given $x \in \{0,1\}^*$ with $|x| \geq m+t+1$, we derive y using a public function such that $|y| \equiv 0 \mod t$.

$$y = \begin{cases} (x, |x| \equiv 0 \mod t) \\ (x||0^d, |x| + d \equiv 0 \mod t) \end{cases}$$

Here, $IV \in \{0,1\}^m$ is a publicly chosen parameter.

We split y into blocks: $y = y_1||y_2||y_3||\ldots||y_r$, where $|y_i| = t$ for $1 \leq i \leq r$. Then, we define $Z_r = H(x)$.

$$\begin{aligned} Z_0 &= IV \\ Z_1 &= h(Z_0||y_1) \\ Z_2 &= h(Z_1||y_2) \\ &\vdots \\ Z_r &= h(Z_{r-1}||y_r) \end{aligned}$$

This type of hash function is known as an iterative hash function.

Merkle-Damgard: -

Let $h : \{0,1\}^* \longrightarrow \{0,1\}$ be a hash function.

Define a compression function $compress : \{0,1\}^{m+t} \longrightarrow \{0,1\}^m$, where $t \geq 2$.

Given an input x with length $n = |x|$, let $K = \lceil \frac{n}{t-1} \rceil$ and $d = K(t-1) - n$. Split x into blocks: $x = x_1||x_2 \ldots x_k$.

For $i = 1$ to $K - 1$:

- Set $y_i = x_i$.

Set $y_k = x_k||0^d$ and $y_{k+1} = binary(d)$.

Initialize $Z_1 = 0^{m+1}||y_1$ and compute $g_1 = compress(Z_1)$.

For $i = 1$ to K:

- Compute $Z_{i+1} = g_i||1||y_{i+1}$.
- Update $g_{i+1} = compress(Z_{i+1})$.

Finally, define $h(x) = g_{k+1}$ and return $h(x)$.

. . . .

. . . .

Secure Hash Algorithm : -

We have three Secure Hash Algorithms (SHAs), namely, SHA-160, SHA-256, and SHA-512.

Let SHA : $\{0,1\}^* \to \{0,1\}^n$. Let's start with SHA-1:

Given an input x, where $|x| \leq 2^{64} - 1$, calculate:

$$\begin{aligned} d &= (477 - |x|) \mod 512 \\ l &= \text{binary}(|x|) \\ y &= x||1||0^d||l \end{aligned}$$

where $|y| = |x| + 1 + d + |l|$ and $|y| \equiv 0 \mod 512$.

SHA-1 :-

Given $x < 2^{64} - 1$:

$$\begin{aligned} d &= (447 - |x|) \mod 512 \\ l &= \text{binary}(|x|) \\ y &= x||1||0^d||l \end{aligned}$$

where $|x| + d \equiv 447 \mod 512$.

Standard operations:

- $X \wedge Y$: bitwise AND operation
- $X \vee Y$: bitwise OR operation
- $X \oplus Y$: bitwise XOR operation
- $\rceil X$: bitwise complement
- $X + Y$: addition modulo 2^{32}

Functions:

- $ROTL^s(x)$: Circular left shift of x by s positions.
- $f_t(B, C, D)$: Hash function defined as:

$$f_t(B,C,D) = \begin{cases} (B \wedge C) \vee ((\rceil B) \wedge D), & \text{if } 0 \leq t \leq 19 \\ B \oplus C \oplus D, & \text{if } 20 \leq t \leq 39 \\ (B \wedge C) \vee (B \wedge D) \vee (C \wedge D), & \text{if } 40 \leq t \leq 59 \\ B \oplus C \oplus D, & \text{if } 60 \leq t \leq 79 \end{cases}$$

Let $y = \text{SHA-1-PAD}(x)$:

- $y = M_1||M_2||\ldots||M_n$, where $|M_i| = 512$.

- Initial values: $H_0 = 67452301$, $H_1 = EFCDAB89$, $H_2 = 98BADCFE$, $H_3 = C3D2E1F0$.
- Constants:

$$K_t = \begin{cases} 5A827999, & \text{if } 0 \leq t \leq 19 \\ 6ED9EBA1, & \text{if } 20 \leq t \leq 39 \\ 8F1BBCDC, & \text{if } 40 \leq t \leq 59 \\ CA62C1D6, & \text{if } 60 \leq t \leq 79 \end{cases}$$

Message Authentication Code (MAC) : -

Alice (K) $\rightarrow$ Bob (K):

- $C = \text{Enc}(M, K) \rightarrow \tilde{C}$
- MAC $= \text{Hash}(M, K) \rightarrow \tilde{\text{MAC}}$
- $\text{Dec}(\tilde{C}, K) = \tilde{M}$
- $\text{Hash}(\tilde{M}, K) = \text{MAC}_1$
- If $\text{MAC}_1 = \{\text{MAC}\}$, then accept $\{M\}$, else reject

HMAC: -

- $ipad = 3636 \ldots 36$ (512 bits)
- $opad = 5656 \ldots 56$ (512 bits)
- K: Secret Key
- $\text{HMAC}_K(x) = \text{H}((K \oplus \text{opad})||\text{H}((K \oplus \text{ipad})||x))$

CBC-MAC(x,K) : -

- $x = x_1||x_2|| \ldots ||x_n$
- $IV = 00 \ldots 0$
- $y_0 = IV$
- For $i = 1$ to n: $y_i = \text{Enc}((y_{i-1} \oplus x_i), K)$
- Return $y(n)$