[CS304] Introduction to Cryptography and Network Security

Course Instructor: Dr. Dibyendu Roy — Winter 2023-2024
Scribed by: Aniruddh Modi (202151022) — Lecture 12 13 (Week 7 - 8)

1 Lecture 12

1.1 Collision finding algorithm

We define a hash function $h : X \to Y$. Let $|Y| = M$. Suppose we have $x \in X and h(x)$ and we want to find $x' \neq x \in X$ such that $h(x) = h(x')$.

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

$$X_0 = \{x_1, x_2, ..., x_Q\}$$

for each $x', x \in X_0$ **do**
  $y_x \leftarrow h(x)$
  $y_{x'} \leftarrow h(x')$
  if $y_x = y_{x'}$ **then**
    return (x, x')
  end if
end for
return failure

Let E_i be a event that $h(x_i) \notin \{h(x_1), h(x_2), h(x_3), ..., h(x_{i-1})\}$. Hence, we can say $Pr[E_1] = 1$ (probability). Let $E_2 : h(x_2) \notin \{h(x_1)\}$.

$$\implies Pr[E_2|E_1] = \frac{M-1}{M}$$

$$\implies \frac{Pr[E_2 \cap E_1]}{Pr[E_1]} = \frac{M-1}{M}$$

$$\implies Pr[E_2 \cap E_1] = \frac{M-1}{M}$$

Now, continuing this for E_3.

$$Pr[E_2|E_1 \cap E_2] = \frac{M-2}{M}$$

$$Pr[E_2 \cap E_1 \cap E_2] = \frac{M-2}{M} \times \frac{M-1}{M}$$

Continuing this,

$$Pr[E_1 \cap E_2 \cap ... \cap E_Q] = \frac{M-1}{M} \times \frac{M-2}{M} \times ... \times \frac{M-Q+1}{M}$$

Finally, probability that collision finding algorithm will not fail is given as :

$$Pr[Collision] = 1 - Pr[E_1 \cap E_2 \cap ... \cap E_Q]$$

$$Pr[Collision] = 1 - \frac{M-1}{M} \times \frac{M-2}{M} \times ... \times \frac{M-Q+1}{M}$$

We know that e^{-x} can be written as

$$e^{-x} = 1 - x + \frac{x^2}{2!} + \frac{x^3}{3!} + ...$$

If x is very small, we can approximate it to $1 - x$.

$$\implies \frac{M-i}{M} = 1 - \frac{i}{M} \textbf{ where } M \gg i$$

$$\implies \frac{M-i}{M} = e^{\frac{-i}{M}}$$

$$\implies Pr[Collision] = 1 - \prod_{i=1}^{Q-1} e^{\frac{-i}{M}}$$

$$\implies Pr[Collision] = 1 - e^{\sum_{i=1}^{Q-1} \frac{-i}{M}}$$

$$\implies Pr[Collision] = 1 - e^{\frac{-Q\times(Q-1)}{2M}}$$

$$\implies e^{\frac{-Q\times(Q-1)}{2M}} = 1 - Pr[Collision]$$

$$\implies \frac{-Q\times(Q-1)}{2M} = \ln 1 - Pr[Collision]$$

$$Q^2 + Q = -2M \ln 1 - Pr[Collision]$$

Since Q is very large, $Q^2 \gg Q$. Therefore,

$$Q^2 = -2M \ln 1 - Pr[Collision]$$

$$Q^2 = 2M \ln \frac{1}{1 - Pr[Collision]}$$

$$Q = \sqrt{2M \ln \frac{1}{1 - Pr[Collision]}}$$

Now if suppose $Pr[Collision]$ is very large, say 0.99. Then $Q = 3.03\sqrt{M}$. Hence, complexity of finding collision is $O(M)$

1.2 Secure Hash Function

A secure hash function is the one which satisfies :

- Complexity of finding second preimage $= O(2^M)$
- Complexity of finding collision $= O(2^{\frac{M}{2}})$

1.3 Compression Function

Compression function $h : \{0,1\}^{m+t} \rightarrow \{0,1\}^m$ where $t \geq 1$

Our target is to construct $H : \{0,1\}^* \rightarrow \{0,1\}^*$. Now, security of H is completely dependent on that of h.

Given $x \in \{0,1\}^*$ such that $|x| \geq m+t+1$. From x construct y using a public function such that $|y| \equiv O(\mod t)$.

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

We will select a public parameter $IV \in \{0,1\}^m$.

$$y = y_1||y_2||y_3||...||y_r$$

where $|y_i| = t, 1 \leq i \leq r$. Now, we define $Z_r = H(x)$.

$$Z_0 = IV$$

$$Z_1 = h(Z_0||y_1)$$

$$Z_2 = h(Z_1||y_2)$$

$$.$$

$$.$$

$$.$$

$$Z_r = h(Z_{r-1}||y_r)$$

We call such a hash function an iterative hash function.

1.4 Merkle-Damgard

We define $h : \{0,1\} \rightarrow \{0,1\}$, $compress : \{0,1\}^{m=t} \rightarrow \{0,1\}^m$ where $t \geq 2$.

Let, $n = |x|$, $K = \lceil \frac{n}{t-1} \rceil$

$$d = K \times (t-1) - n$$

$$x = x_1||x_2||...||x_K$$

for $i = 1$ **to** $K-1$ **do**
  $y_i \leftarrow x_i$
end for

$$y_k = x_k||0^d$$

$$y_{k+1} = binary(d)$$

$$Z_1 = 0^{m+1}||y_1$$

$$g_1 = compress(Z_1)$$

for $i = 1$ **to** K **do**
  $Z_{i+1} \leftarrow g_i||1||y_{i+1}$

$g_{i+1} \leftarrow compress(Z_{i+1})$
end for
$h(x) = g_{k+1}$
return $h(x)$

2 Lecture 13

2.1 Secure Hash Algorithm

We have three SHAs, namely, **SHA-160**, **SHA-256** and **SHA-512**.

We define $SHA : \{0,1\}^* \rightarrow \{0,1\}^n$. Let's start with $SHA-1-PAD(x)$

$$|x| \leq 2^{64} - 1$$

$$d = (477 - |x|) \mod 512$$

$$l = binary(|x|)$$

$$y = x||1||0^d||l$$

$$|y| = |x| + 1 + d + l$$

$$|y| \equiv 0 \mod 512$$

2.1.1 Standard Operations

Standard Operations which are used :

1. $X \wedge Y$: bitwise AND operation
2. $X \vee Y$: bitwise OR operation
3. $X \oplus Y$: bitwise XOR operation
4. $\rceil X$: bitwise compliment
5. $X + Y$: Addition modulo 2^{32}

2.1.2 Standard Functions

Functions that are involved :

1. $ROTL^s(x)$: Circular left shift on x by S position.
2. $$f_t(B,C,D) = \begin{cases} (B \wedge C) \vee ((\neg B) \wedge D), & \text{if } 0 \leq t \leq 19 \\ B \oplus C \oplus D, & \text{if } 20 \leq t \leq 39 \\ (B \wedge C) \vee (B \wedge D) \vee (C \wedge D), & \text{if } 40 \leq t \leq 59 \\ B \oplus C \oplus D, & \text{if } 60 \leq t \leq 79 \end{cases}$$

Now, let's look at $SHA-1(x)$. Let $y = SHA-1-PAD(x)$.

$$y = M_1||M_2||...||M_N \textbf{ where } |M_i| = 512$$

$$H_0 = 67452301$$

$$H_1 = EFCDBA89$$

$$H_2 = 98BADCFE$$

$$H_3 = 10325476$$

$$H_4 = C3D2E1F0$$

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

for $i = 1$ **to** n **do**
  $M_i \leftarrow W_0||W_1||W_2||...||W_{15}$ ▷ $|W_i| = 32$
  for $t = 16$ **to** 79 **do**
    $W_t \leftarrow ROTL^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}$
  end for
  $A \leftarrow H_0$
  $B \leftarrow H_1$
  $C \leftarrow H_2$
  $D \leftarrow H_3$
  $E \leftarrow H_4$
  for $t = 0$ **to** 79 **do**
    $temp \leftarrow ROTL^5(A) + f_t(B, C, D) + E + W_t + K_t$
    $E \leftarrow D$
    $D \leftarrow C$
    $C \leftarrow ROTL^{30}(B)$
    $B \leftarrow A$
    $A \leftarrow temp$
  end for
  $H_0 \leftarrow H_0 + A$
  $H_1 \leftarrow H_1 + B$
  $H_2 \leftarrow H_2 + C$
  $H_3 \leftarrow H_3 + D$
  $H_4 \leftarrow H_4 + E$
end for
return $H_0||H_1||H_2||H_3||H_4$

2.2 Message Authentication Code

Suppose Alice and Bob are exchanging some messages between them. Both have the same secret key K. Let the ciphertext generated by Alice for message M be C. Also, a hash of M with K is also generated by Alice. Let that hash be MAC. Now suppose after transmission, the data received by Bob is $\tilde{C}$ and $M\tilde{A}C$. Now, if Bob decrypts $\tilde{C}$, he will get a plaintext $\tilde{M}$. To decide whether to accept or reject this message, what Bob can do is generate a hash with the same hash function Alice used. $MAC = Hash(\tilde{M}, K)$. If MAC and $M\tilde{A}C$ are same, then the cipher text recieved by Bob was not altered. So, he can safely accept it.

2.2.1 Hash Based Message Authentication Code

In HMAC, we define $ipad = 363636...36 \rightarrow 512 \text{ bits}$ and $opad = 5c5c5c..5c \rightarrow 512 \text{ bits}$. Let, K be the secret key.

$$HMAC_K(x) = H((K \oplus opad)||(H(K \oplus ipad))||x)$$

2.2.2 CBC-MAC(x, K)

$$x = x_1||x_2 \dots x_n$$

$$IV = 00 \dots 0$$

$$y_0 = IV$$

for $i = 1$ **to** n **do**
  $y_i = Enc((y_{i-1} \oplus x_i), K)$
end for
return y_n

2.2.3 Introduction to SHA-256

$$\text{Message Size} \leq 2^{64} - 1$$

$$\text{Block Size} = 512$$

$$\text{Word Size} = 32$$

Functions used in SHA-256 are :

1. Rotate
2. Shift
3. Choose
4. $Majority(X, Y, Z) = (X \wedge Y) \oplus (Y \wedge Z) \oplus (Z \wedge X)$
5. 2 $\sigma functions$