Homework 2.b Solutions

- **Group Members: Yu Wang(wangy52 661834351)**
- **Collaborators: no**

Problem   1

(a)

$$Y_A = \alpha^{X_A} \mod q = 7^5 \mod 71 = 51 \tag{1-1}$$

(b)

$$Y_B = \alpha^{X_B} \mod q = 7^{12} \mod 71 = 4 \tag{1-2}$$

(c)

The shared secret key K_{AB} is

$$K_{AB} = Y_B^{X_A} \mod q = Y_A^{X_B} \mod q = 4^5 \mod 71 = 51^12 \mod 71 = 30 \tag{1-3}$$

(d)

Sending $y = x^\alpha \mod q$ is a bad idea because the adversary who knows y, α and q can compute the private key x using the following method:

$$
\begin{aligned}
x =& y^{-\alpha} \mod q \\
=& 1 \cdot y^{-\alpha} \mod q \\
=& y^{\phi(q)} \cdot y^{-\alpha} \mod q \\
=& y^{q-1-\alpha} \mod q
\end{aligned}
\tag{1-4}
$$

For example, choose $x = 5$, $\alpha = 7$ and $q = 71$,

$$
\begin{aligned}
y = x^\alpha \mod q = 5^7 \mod 71 = 25 \\
y^{q-1-\alpha} \mod q = 25^{71-1-7} \mod 71 = 25^{63} \mod 71 = 5 = x
\end{aligned}
\tag{1-5}
$$

In conclusion, the user's private key is not protected.

Problem 2

(a)

Mallory wants to fraud Bob to add a valid signature on a fraudulent message m'. The original message is m. Mallory first generates a large set of messages $\{M\}$ from m such that each message in $\{M\}$ looks the same as m. Then, he generates another big set of messages $\{M'\}$ such that the message in $\{M'\}$ are distorted. Applying the hashing function H to all the elements in both $\{M\}$ and $\{M'\}$ and find a pair of two messages m and m' such that:

$$m \in M, m' \in M', H(m) = H(m'). \tag{2-1}$$

Mallory sends m to Bob and gets a valid signature, then he attachs the acquired signature on the fraudlent message m'. Now a valid signature with fraudulent message is generated successfully.

(b)

Assume the hash value is uniformly distributed in $\{0, 2^{64} - 1\}$. Denote the number of messages in $\{M\}$ and $\{M'\}$ are N_M and $N_{M'}$ repectively. Use indicator function $\mathbb{1}[H(m_i) = H(m'_j)]$ to show whether the hash value of $m_i \in \{M\}$ equals the hash value of $m'_j \in \{M'\}$. The probability that $H(m_i) = H(m'_j)$ is

$$Pr[H(m_i) = H(m'_j)] = 1/2^{64}. \tag{2-2}$$

On average, the number of pairs of messages (m_i, m'_j) which has hash collision, i.e. $H(m_i) = H(m_j)$, is

$$\mathbb{E}[\text{Number of collisions}] = \mathbb{E}\{\sum_{i=1}^{N_M} \sum_{j=1}^{N_{M'}} \mathbb{1}[H(m_i) = H(m_j)]\} \tag{2-3}$$

$$= \sum_{i=1}^{N_M} \sum_{j=1}^{N_{M'}} \mathbb{E}\{\mathbb{1}[H(m_i) = H(m_j)]\} \tag{2-4}$$

$$= N_M \cdot N_{M'} \cdot 1/2^{64}. \tag{2-5}$$

If we want the number of collisions on average to be greater than 1, we should have $N_M \cdot N_{M'} \geq 2^{64}$. If we also require $N_M + N_{M'}$ to be as small as possible, then the optimal solution is

$$N_M = N_{M'} = \sqrt{2^{64}} = 2^{32}. \tag{2-6}$$

So the number of bits, including $M - bit$ messages and their hashes, are

$$N_M \cdot (M + 64) + N_{M'} \cdot (M + 64) = 2^{33}(M + 64). \tag{2-7}$$

(c)

The number of seconds is

$$\frac{N_M + N_{M'}}{2^{20} \text{ hashs/second}} = \frac{2^{32} + 2^{32}}{2^{20}} = 2^{13} \text{ seconds} \approx 2.27 \text{ hours.} \tag{2-8}$$

(d)

When 128-bit hash is used, the probability of having a collision is

$$Pr[H(m_i) = H(m'_j)] = 1/2^{128}. \tag{2-9}$$

Then the number of collisions between $\{M\}$ and $\{M'\}$ is

$$\mathbb{E}[\text{Number of collisions}] = N_M \cdot N_{M'} \cdot 1/2^{128}. \tag{2-10}$$

The optimal solutions to N_M and $N_{M'}$ are

$$N_M = N_{M'} = \sqrt{2^{128}} = 2^{64}. \tag{2-11}$$

Then the required size of memory is

$$N_M \cdot (M + 128) + N_{M'} \cdot (M + 128) = 2^{65}(M + 128). \tag{2-12}$$

The amount of time required is

$$\frac{N_M + N_{M'}}{2^{20} \text{ hashs/second}} = \frac{2^{64} + 2^{64}}{2^{20}} = 2^{45} \text{ seconds} \approx 1.12 \times 10^7 \text{ years}. \tag{2-13}$$

Problem 3

- Encrypt P

    1. Compute set T
       For $t_i \in T$ and $s_i \in S$ where $i = 1, 2, \cdots, 8$,

       $$t_i = a \cdot s_i \mod p. \tag{3-1}$$

       We have

       $$
       \begin{aligned}
       t_1 &= a \cdot s_1 \mod p = 1019 \cdot 5 \mod 1999 &=1097 \\
       t_2 &= a \cdot s_2 \mod p = 1019 \cdot 9 \mod 1999 &=1175 \\
       t_3 &= a \cdot s_3 \mod p = 1019 \cdot 21 \mod 1999 &=1409 \\
       t_4 &= a \cdot s_4 \mod p = 1019 \cdot 45 \mod 1999 &=1877 \\
       t_5 &= a \cdot s_5 \mod p = 1019 \cdot 103 \mod 1999 &=1009 \\
       t_6 &= a \cdot s_6 \mod p = 1019 \cdot 215 \mod 1999 &=1194 \\
       t_7 &= a \cdot s_7 \mod p = 1019 \cdot 450 \mod 1999 &=779 \\
       t_8 &= a \cdot s_8 \mod p = 1019 \cdot 946 \mod 1999 &=456
       \end{aligned} \tag{3-2}
       $$

    2. Compute Ciphertext Y
       Ciphertext Y is

       $$
       \begin{aligned}
       Y &= \sum_{i=1}^{8} p_i \cdot t_i \\
       &=0 \cdot 1097 + 1 \cdot 1175 + 0 \cdot 1409 + 1 \cdot 1877 + 0 \cdot 1009 + 1 \cdot 1194 + 1 \cdot 779 + 1 \cdot 456 \\
       &=5481
       \end{aligned} \tag{3-3}
       $$

- Decrypt Y

    1. Compute Z

       $$
       \begin{aligned}
       Z &=a^{-1}Y \mod p \\
       &=1589 \cdot 5481 \mod 1999 \\
       &=1665
       \end{aligned} \tag{3-4}
       $$

3

2. Use Greedy Algorithm to Solve P based on (S, Z) Start from the largest element to the smallest.

$$
\begin{aligned}
s_8 &= 946 < Z = 1665 &\Rightarrow p_8 &= 1. Z = Z - s_8 = 719. \\
s_7 &= 450 < Z = 719, &\Rightarrow p_7 &= 1. Z = Z - s_7 = 269. \\
s_6 &= 215 < Z = 269, &\Rightarrow p_6 &= 1. Z = Z - s_6 = 54. \\
s_5 &= 103 > Z = 54, &\Rightarrow p_5 &= 0. \\
s_4 &= 45 < Z = 54, &\Rightarrow p_4 &= 1. Z = Z - s_4 = 9. \\
s_3 &= 21 > Z = 9, &\Rightarrow p_3 &= 0. \\
s_2 &= 9 = Z = 9, &\Rightarrow p_2 &= 1. Z = Z - s_2 = 0. \\
s_1 &= 5 > Z = 0, &\Rightarrow p_1 &= 0.
\end{aligned}
\tag{3-5}
$$

So the decrypted plaintext $\hat{P} = [p_1, p_2, \cdots, p_8] = [0, 1, 0, 1, 0, 1, 1, 1]$ which equals P.