VE475 Introduction to Cryptography
Homework 7

Jiang, Sifan
jasperrice@sjtu.edu.cn
515370910040

July 12, 2019

Homework 6

Ex. 5 - Merkle-Damgård construction

1. a) Since $f(0) = 0$ and $f(1) = 01$, $f(x_i)$ is always start with 0. So y can be separated into several segments start from 0, except for the first two digits. Those segments are injective with x_i, so the map s from x to y is injective.
   b) If z is empty, from what previous proved, there's no such x'. If z is not empty, since we have 11 at the beginning of y_{i+1}, so there's no x' and z such that $s(x) = z \| s(x')$.

2. Because the previous conditions guarantee the mapping is collision resistant.

3. Assuming we have a collision on h, i.e. $x \neq x'$ and $h(x) = h(x')$, we will prove that a collision on the compression function g can be efficiently found.

   First note that if $x \neq x'$, since the map s from x to y is injective, it would always lead to $y \neq y'$. Let k and k' denote the number blocks for y and y'.

   Case 1: consider $k = k'$. This implies $y_k = y_{k'}$, and we have

$$\begin{aligned} g(z_{k-1} \| y_k) &= z_k = h(x) \\ &= h(x') = z_{k'} \\ &= g(z_{k'-1} \| y_{k'}) \end{aligned}$$

   If $z_k \neq z_{k'}$, then a collision is found. Otherwise we repeat the process and get

$$\begin{aligned} g(z_{k-2} \| y_{k-1}) &= z_{k-1} = h(x) \\ &= h(x') = z_{k'-1} \\ &= g(z_{k'-2} \| y_{k'-1}) \end{aligned}$$

   Then either we have found a collision or we continue backward until one is obtained. If none is found then we get $z_i = z_{i'}$, where $i \in [1, k]$.

   Case 2: consider $k \neq k'$. Without loss of generality assume k' and proceed as in case 1. If no collision is found before $k = 1$, then we have

$$\begin{aligned} g(0^m \| y_1) &= z_1 \\ &= z_{k'-k+1} \\ &= g(z_{k'-k} \| y_{k'-k+1}) \end{aligned}$$

   Since there is not strings $x \neq x'$ and z such that $s(x) = z \| s(x')$, the collision is found.

Homework 7

Ex. 1 - Cramer-Shoup cryptosystem

1. **Key generation:**

   - Alice generates a cyclic group G of order q with two distinct generators g_1, g_2. G could be $\mathrm{U}(\mathbb{Z}/p\mathbb{Z})$.
   - Alice chooses five random values (x_1, x_2, y_1, y_2, z) from $\{0, 1, \cdots, q-1\}$.
   - Alice computes $c = g_1^{x_1} g_2^{x_2}$, $d = g_1^{y_1} g_2^{y_2}$, and $h = g_1^z$.
   - Alice publishes (c, d, h) and G, q, g_1, g_2 as her public key. She retains (x_1, x_2, y_1, y_2, z) as her private key.

   Encryption:

   - Bob converts plaintext into an element m in group G.
   - Bob chooses a random k from $\{0, 1, \cdots, q-1\}$, then calculates:
     - $u_1 = g_1^k$, $u_2 = g_2^k$.
     - $e = h^k m$.
     - $\alpha = H(u_1, u_2, e)$, where H is a collision-resistant cryptographic hash function.
     - $v = c^k d^{k\alpha}$.
   - Bob sends the ciphertext (u_1, u_2, e, v) to Alice.

   Decryption:

   - Alice computes $\alpha = H(u_1, u_2, e)$ and verifies that $u_1^{x_1} u_2^{x_2} (u_1^{y_1} u_2^{y_2})^\alpha = v$. If it fails, further decryption is aborted.
   - Since $u_1^z = g_1^{kz} = h^k$, and $m = \frac{e}{h^k}$, Alice computes the plaintext as $m = \frac{e}{u_1^z}$.

2. Adaptive chosen ciphertext attacks is an iterative chosen ciphertext attack scenario in which the attacker gradually reveal information about an ciphertext c or private key by iteratively sending new ciphertexts c', c'', $\cdots$ that are related to the original ciphertext c to the receiver and analysis the response. However, in the decryption stage of Cramer-Shoup cryptosystem, there's a verification stage where invalid ciphertexts would be rejected. Also, since H is a collision-resistant cryptographic hash function, it's practically infeasible to find enough chosen ciphertext to attack.

3. - **Similarities:** Both cryptosystems are based on the DLP in a cyclic group.
   - **Differences:** Cramer-Shoup cryptosystem uses a collision-resistant cryptographic hash function for verification to avoid adaptive chosen ciphertext attacks, while the Elgamal cryptosystem doesn't.

Ex. 2 - Simple questions

1. According to Fermat's little theorem, if p is prime and $p \nmid \alpha$, then $a^{p-1} \equiv 1 \mod p$. $h(x)$ is not second pre-image resistant because given x, it is easy to find $x' = x + p - 1$ such that $h(x) = h(x')$. So, it is not a good cryptographic hash function.

2. We would have

$$\begin{aligned}
\left\lfloor 2^{30}\sqrt{2} \right\rfloor &= \left\lfloor 40000000\sqrt{2} \right\rfloor = 5A827999 && = K_i, \quad \text{where } 0 \le i \le 19 \\
\left\lfloor 2^{30}\sqrt{3} \right\rfloor &= \left\lfloor 40000000\sqrt{3} \right\rfloor = 6ED9EBA1 && = K_i, \quad \text{where } 20 \le i \le 39 \\
\left\lfloor 2^{30}\sqrt{4} \right\rfloor &= \left\lfloor 40000000\sqrt{4} \right\rfloor = 8F1BBCDC && = K_i, \quad \text{where } 40 \le i \le 59 \\
\left\lfloor 2^{30}\sqrt{5} \right\rfloor &= \left\lfloor 40000000\sqrt{5} \right\rfloor = CA62C1D6 && = K_i, \quad \text{where } 60 \le i \le 79
\end{aligned}$$

Ex. 3 - Birthday paradox

1. Since $g(x) = \ln(1-x) + x + x^2$, we have

$$\frac{dg(x)}{dx} = -\frac{1}{1-x} + 1 + 2x$$

When $\frac{dg(x)}{dx} = 0$, we have $x_1 = 0$ and $x_2 = \frac{1}{2}$. Also, since

$$\begin{aligned}
\frac{d^2g(x)}{dx^2} &= -\frac{1}{(1-x)^2} + 2 \\
\left.\frac{d^2g(x)}{dx^2}\right|_{x=0} &= 1 > 0 \\
\left.\frac{d^2g(x)}{dx^2}\right|_{x=\frac{1}{2}} &= -2 < 0
\end{aligned}$$

we could conclude that for $x \in \left[0, \frac{1}{2}\right]$, $g(x) \in \left[g(0), g\left(\frac{1}{2}\right)\right]$. So $g(x) \ge g(0) = 0$, which gives $-x - x^2 \le \ln(1-x)$.

Then having $h(x) = \ln(1-x) + x$, we can apply similar method and finally get $\ln(1-x) \le -x$.

In all, when $x \in \left[0, \frac{1}{2}\right]$, $-x - x^2 \le \ln(1-x) \le -x$.

2. Since $j \in [1, r-1]$ and $r \le \frac{n}{2}$, we would have $\frac{j}{n} \in \left[0, \frac{1}{2}\right]$, thus, according to the result from previous problem

$$-\frac{j}{n} - \left(\frac{j}{n}\right)^2 \le \ln(1 - \frac{j}{n}) \le -\frac{j}{n}$$

Then, since $r > 1$, apply the sum to each parts,

$$\begin{aligned}
\sum_{j=1}^{r-1}\left[-\frac{j}{n} - \left(\frac{j}{n}\right)^2\right] &= -\frac{(r-1)r}{2n} - \frac{r(r-1)(2r-1)}{6n^2} > -\frac{(r-1)r}{2n} - \frac{r^3}{3n^2} \\
\sum_{j=1}^{r-1}\left[-\frac{j}{n}\right] &= -\frac{(r-1)r}{2n}
\end{aligned}$$

So, in all, we would have

$$-\frac{(r-1)r}{2n} - \frac{r^3}{3n^2} \le \sum_{j=1}^{r-1} \ln\left(1 - \frac{j}{n}\right) \le -\frac{(r-1)r}{2n}$$

3. Apply exponentiate to the previous inequality equation, we would have

$$\exp\left(-\frac{(r-1)r}{2n}-\frac{r^3}{3n^2}\right) \le \prod_{j=1}^{r-1}\left(1-\frac{j}{n}\right) \le \exp\left(-\frac{(r-1)r}{2n}\right)$$

Let $\lambda = \frac{r^2}{2n}$, $c_1 = \sqrt{\frac{\lambda}{2}}$, and $c_2 = \sqrt{\frac{\lambda}{2}}$, we would have

$$e^{-\lambda}e^{c_1/\sqrt{n}} \le \prod_{j=1}^{r-1}\left(1-\frac{j}{n}\right) \le e^{-\lambda}e^{c_2/\sqrt{n}}$$

4. When $\lambda < \frac{n}{8}$, we would have

$$\lambda = \frac{r^2}{2n} < \frac{n}{8}$$

which gives $r < \frac{n}{2}$. Also, when n is large,

$$\lim_{n\to\infty} e^{c_1/\sqrt{n}} = \lim_{n\to\infty} e^0 = 1$$
$$\lim_{n\to\infty} e^{c_2/\sqrt{n}} = \lim_{n\to\infty} e^0 = 1$$

So, we would have

$$\prod_{j=1}^{r-1}\left(1-\frac{j}{n}\right) \approx e^{-\lambda}$$

Ex. 4 - Birthday attack

1. The probability of seeing two plates ending with the same three digits is calculated as

$$P = 1 - \prod_{j=1}^{39}\left(1-\frac{j}{1000}\right) \approx 0.54637$$

2. The probability that one of the 40 license plates observed has the same 3 last digits is calculated as

$$P = 40\left(\frac{1}{1000}\right)\left(\frac{999}{1000}\right)^{39} \approx 0.038469$$

3. From question 1, we can tell that it's not collision resistant, but from question 2, we can tell that it's second pre-image resistant. So Alice can prevent the collision by changing the message from Eve a little bit.

Ex. 5 - Faster multiple modular exponentiation

1. The time complexity to compute $\alpha^a \mod n$ is $\mathcal{O}\left((\log n)^2 \log a\right)$. The time complexity to compute $\beta^b \mod n$ is $\mathcal{O}\left((\log n)^2 \log b\right)$. So the time complexity of this method is $\mathcal{O}\left((\log n)^2(\log a + \log b)\right)$.

2. The algorithm is shown in alg 1.

3. If a and b are l bits long, l squaring and multiplications are necessary to compute $\alpha^a\beta^b \mod n$.

Algorithm 1 Faster Multiple Modular Exponentiation

Input: α, β, $a = (a_{k_a-1} \cdots a_0)_2$, $b = (b_{k_b-1} \cdots b_0)_2$, and n five integers

Output: $\alpha^a \beta^b \mod n$

$k \leftarrow \max(k_a, k_b)$
$power \leftarrow 1$
for $i \leftarrow k - 1$ **to** 0 **do**
    $power \leftarrow (power \cdot power) \mod n$
    if $a_i = 1$ **then**
        $power \leftarrow (\alpha \cdot power) \mod n$
    end if
    if $b_i = 1$ **then**
        $power \leftarrow (\beta \cdot power) \mod n$
    end if
end for
return $power$