VE475 Homework 7

Liu Yihao 515370910207

Ex. 1 — Cramer-Shoup cryptosystem

1. Cramer–Shoup cryptosystem consists of three algorithms: the key generator, the encryption algorithm, and the decryption algorithm.

   a) The key generator
   First, Alice generates a cyclic group G of order q and finds two generators g_1 and g_2 for it. Then she randomly chooses x_1, x_2, y_1, y_2, z from $\{0, \ldots, q-1\}$ and computes $c = g_1^{x_1} g_2^{x_2}$, $d = g_1^{y_1} g_2^{y_2}$ and $h = g_1^z$. At last, she publishes $(c, d, h, G, q, g_1, g_2)$ as the public key and keeps (x_1, x_2, y_1, y_2, z) as the private key.

   b) The encryption algorithm
   First, Bob converts m into an element of G and choose a random k from $\{0, \ldots, q-1\}$. Then he computes $u_1 = g_1^k$, $u_2 = g_2^k$, $e = h^k m$, $\alpha = H(u_1, u_2, e)$ where $H(x)$ is a collision-resistant cryptographic hash function, and $v = c^k d^{k\alpha}$ At last, he sends the ciphertext (u_1, u_2, e, v) to Alice.

   c) The decryption algorithm
   First, 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 the verification fails, the decryption algorithm ends with failure output. Otherwise, she computes the plaintext $m = e/h^k$. The decryption stage correctly decrypts any properly-formed ciphertext, since $u_1^z = g_1^{kz} = h^k$.

2. Adaptive chosen ciphertext attacks can be applied if a ciphertext can be modified in specific ways that will have a predictable effect on the decryption of that message. However, The decryption algorithm of Cramer-Shoup cryptosystem rejects all invalid ciphertexts constructed by an attacker through verifying the result generated by a collision-resistant cryptographic hash function. It limits ciphertext malleability so that it can be considered secure under this kind of attack.

3. a) Similarities: Both are public key cryptosystems computed in a cyclic group G, the private keys are both based on the difficulty of solving Discrete Logarithm Problem.

   b) Differences: Cramer–Shoup cryptosystem consists a collision-resistant cryptographic hash function which is used to verify the ciphertext while Elgamal cryptosystem doesn't.

Ex. 2 — Simple questions

1. Since p is a prime and $p \nmid \alpha$, we can find $\gcd(p, \alpha) = 1$, so $\alpha^{p-1} \equiv 1 \bmod p$. First, $h(x)$ isn't second pre-image resistant. Given x, we can simply find $x' = x + p - 1$ so that $h(x) = h(x')$. Second, $h(x)$ isn't collision resistant. For any x, we can simply find $x' = x + p - 1$ so that $h(x) = h(x')$. So it is not a good cryptographic hash function.

2.

$$\lfloor 2^{30}\sqrt{2}\rfloor = \lfloor 40000000 \cdot \sqrt{2}\rfloor = 5A827999$$
$$\lfloor 2^{30}\sqrt{3}\rfloor = \lfloor 40000000 \cdot \sqrt{3}\rfloor = 6ED9EBA1$$
$$\lfloor 2^{30}\sqrt{5}\rfloor = \lfloor 40000000 \cdot \sqrt{5}\rfloor = 8F1BBCDC$$
$$\lfloor 2^{30}\sqrt{10}\rfloor = \lfloor 40000000 \cdot \sqrt{10}\rfloor = CA62C1D6$$

I found the results identical to $K_0||\cdots||K_{19}$, $K_{20}||\cdots||K_{39}$, $K_{40}||\cdots||K_{59}$ and $K_{60}||\cdots||K_{79}$.

Ex. 3 — Birthday paradox

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

$$g'(x) = -\frac{1}{1-x} + 1 + 2x$$

When $g'(x) = 0$,

$$1 + x - 1 + 2x(x-1) = 0$$
$$x_1 = 0, x_2 = \frac{1}{2}$$
$$g''(x) = -\frac{1}{(x-1)^2} + 2$$
$$g(0) = 1,\ \text{it is a local minimum point}$$
$$g\left(\frac{1}{2}\right) = -2,\ \text{it is a local maximum point}$$

So we can conclude that when $x \in \left[0, \frac{1}{2}\right]$, $g(x) \in \left[g(0), g\left(\frac{1}{2}\right)\right] \geqslant 0$

Similarly, let $h(x) = \ln(1-x) + x$, we know

$$h'(x) = -\frac{1}{1-x} + 1$$

When $h'(x) = 0$,

$$1 + x - 1 = 0$$
$$x = 0$$
$$h''(x) = -\frac{1}{(x-1)^2}$$
$$h(0) = -1,\ \text{it is a local maximum point}$$

So we can conclude that when $x \in \left[0, \frac{1}{2}\right]$, $h(x) \in \left[h\left(\frac{1}{2}\right), h(0)\right] \leqslant 0$

According to the above,

$$-x - x^2 \leqslant \ln(1-x) \leqslant -x$$

2. Since $j \in [1, r-1]$ and $r \leqslant \frac{n}{2}$, we can find that $\frac{j}{n} \in \left[0, \frac{1}{2}\right]$, so

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

$$\sum_{j=1}^{r-1}\left[-\frac{j}{n} - \left(\frac{j}{n}\right)^2\right] \leqslant \sum_{j=1}^{r-1}\ln\left(1 - \frac{j}{n}\right) \leqslant \sum_{j=1}^{r-1} -\frac{j}{n}$$

$$-\frac{(r-1)r}{2n} - \frac{(r-1)r(2r-1)}{6n^2} \leqslant \sum_{j=1}^{r-1}\ln\left(1 - \frac{j}{n}\right) \leqslant -\frac{(r-1)r}{2n}$$

When $r > 1$,

$$\frac{(r-1)r(2r-1)}{6n^2} = \frac{r^3 - \frac{3}{2}r^2 + r}{3n^2} < \frac{r^3}{3n^2}$$

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

3. Exponentiate the inequation above, we can get

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

Let $\lambda = \frac{r^2}{2n}$, $c_1 = \sqrt{\frac{\lambda}{2}} - \frac{(2\lambda)^{3/2}}{3}$ and $c_2 = \sqrt{\frac{\lambda}{2}}$.

$$-\lambda + \frac{c_1}{\sqrt{n}} = -\frac{r^2}{2n} + \frac{r}{2n} - \frac{r^3}{n^2} = -\frac{(r-1)r}{2n} - \frac{r^3}{3n^2}$$

$$-\lambda + \frac{c_2}{\sqrt{n}} = -\frac{r^2}{2n} + \frac{r}{2n} = -\frac{(r-1)r}{2n}$$

So

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

4. If n is large and $\lambda < \frac{n}{8}$

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

Since λ is a constant, c_1 and c_2 are also constants.

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

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

Then we can conclude that

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

Ex. 4 — Birthday attack

1. $$P = 1 - \prod_{j=1}^{39} \left(1 - \frac{j}{1000}\right) \approx 0.5464$$

2. $$P = 39 \left(\frac{1}{1000}\right) \left(\frac{999}{1000}\right)^{38} \approx 0.0375$$

3.

Ex. 5 — Faster multiple modular exponentiation

1. The complexity of computing $\alpha^a \bmod n$ is $O(\log a)$, the complexity of computing $\beta^b \bmod n$ is $O(\log b)$, so the total time complexity is $O(\log ab)$.

2.

3.

4.