VE475 Homework 6

Liu Yihao 515370910207

Ex. 1 — Application of the DLP

1. (a) For Alice, she knows that

$$\gamma \equiv \alpha^r \bmod p$$

   If Bob replies

$$b \equiv r \bmod p-1 \text{ or } b \equiv x+r \bmod p-1$$

   She can get

$$\alpha^{p-1} \equiv 1 \bmod p$$

$$\alpha^b \equiv \alpha^r \equiv \gamma \bmod p \text{ or } \alpha^b \equiv \alpha^{x+r} \equiv \gamma\beta \bmod p$$

   So after calculating $\alpha^b \bmod p$ and compare it with γ or $\gamma\beta$, she can prove Bob's identity if he can calculate $x = \log_\alpha \beta$.

   (b) For Bob, if he doesn't know x, then he can't compute $b \equiv x+r \bmod p-1$.If he want to know x, it becomes a DLP problem which is very difficult to solve, so he can prove his identity.

2. (a) 128 times.

   (b) 192 times.

3. It is Digital Signature Protocol.

Ex. 2 — Pohlig-Hellman

First, let g be a generator of the group, let $x = \log_g h$, let n be the order of the group, obtain a prime factorization so that

$$n = \prod_{i=1}^{r} p_i^{e_i}$$

Then, for each $i \in \{1, \dots, r\}$, compute $g_i = g^{n/p_i^{e_i}}$, which has order $p_i^{e_i}$, and compute $h_i = h^{n/p_i^{e_i}}$. Then we can use the Pohlig-Hellman algorithm for prime-power order to compute $x_i \in \{0, \dots, p_i^{e_1} - 1\}$, which is described as follow:

1. Let $x = \log_g h$ ($x = x_i$, $g = g_i$, $h = h_i$ from previous part), where $g = p^e$, and first initialize $x_0 = 0$.

2. Set $\gamma = g^{p^{e-1}}$.

3. For each $k \in \{0, \dots, e-1\}$, compute $h_k = (g^{-x_k} h)^{p^{e-1-k}}$, By construction, the order of this element must divide p, hence $h_k \in \langle\gamma\rangle$. Then compute d_k such that $\gamma^{d_k} = h_k$ and set $x_{k+1} = x_k + p^k d_k$.

4. Obtain $x = x_e$.

After get all x_i, solve the simultaneous congruence

$$x \equiv x_i \bmod p_i^{e_i}, i \in \{1, \dots, r\}$$

according to Chinese reminder theorem to get $x = \log_g h$.

As an example, we try to find $\log_3 3344$ in $G = U(Z/24389Z)$. Note that $24389 = 29^3$, so the order $n = 28 \cdot 29^2 = 2^2 \cdot 7 \cdot 29^2$.

And 3 is a generator of G, so we can get

$$g_1 \equiv 3^{7 \cdot 29^2} \equiv 10133 \bmod 24389$$
$$h_1 \equiv 3344^{7 \cdot 29^2} \equiv 24388 \bmod 24389$$
$$g_2 \equiv 3^{2^2 \cdot 29^2} \equiv 7302 \bmod 24389$$
$$h_2 \equiv 3344^{2^2 \cdot 29^2} \equiv 4850 \bmod 24389$$
$$g_3 \equiv 3^{2^2 \cdot 7} \equiv 11369 \bmod 24389$$
$$h_3 \equiv 3344^{2^2 \cdot 7} \equiv 23114 \bmod 24389$$

First, for $p = 2$, $e = 2$, $g = 10133$ and $h = 24388$, we should determine $x_a = \log_g h$. We can get

$$\gamma \equiv 10133^2 \equiv 24388 \equiv -1 \bmod 24389$$

$$h_0 \equiv (10133^0 \cdot -1)^2 \equiv 1 \bmod 24389, \quad d_0 = 0, \quad x_1 \equiv 0 \bmod 4$$
$$h_1 \equiv (10133^0 \cdot -1)^1 \equiv -1 \bmod 24389, \quad d_1 = 1, \quad x_2 \equiv 2 \bmod 4$$

$$x_a = 2 \bmod 4$$

Second, for $p = 7$, $e = 1$, $g = 7302$ and $h = 4850$, we should determine $x_b = \log_g h$. We can get

$$\gamma \equiv 7302^1 \equiv 7302 \bmod 24389$$

$$h_0 \equiv (7302^0 \cdot 4850)^1 \equiv 4850 \bmod 24389, \quad d_0 = 2, \quad x_1 \equiv 2 \bmod 7$$

$$x_b = 2 \bmod 7$$

Third, for $p = 29$, $e = 2$, $g = 11369$ and $h = 23114$, we should determine $x_c = \log_g h$. We can get

$$\gamma \equiv 11369^{29} \equiv 12616 \bmod 24389$$

$$h_0 \equiv (11369^0 \cdot 23114)^{29} \equiv 11775 \bmod 24389, \quad d_0 = 28, \quad x_1 \equiv 28 \bmod 841$$
$$h_1 \equiv (11369^{-28} \cdot 23114)^1 \equiv 3365 \bmod 24389, \quad d_1 = 8, \quad x_2 \equiv 260 \bmod 841$$

$$x_c = 260 \bmod 841$$

According to Chinese remainder theorem, we can simply get

$$x \equiv 2 \bmod 28$$
$$x \equiv 260 \bmod 841$$
$$841 \cdot 1 \equiv 1 \bmod 28$$
$$28 \cdot 811 \equiv 1 \bmod 841$$

$$x \equiv 841 \cdot 1 \cdot 2 + 28 \cdot 811 \cdot 260 \equiv 18762 \bmod 23548$$

Ex. 3 — Elgamal

1. If the polynomial X^3+2X^2+1 is reducible in $F_3[x]$, it can be factored as

$$X^3+2X^2+1=(X+A)(X^2+BX+C)=X^3+A(B+1)X^2+(B+C)X+AC$$

   There are two possible pairs of (A,C), which are $(1,1)$ and $(2,2)$ so that $AC=1$.

   First, if $A=C=1$, then $B=2$, but $A(B+1)=0\neq 2$, so it is wrong.

   Second, if $A=C=2$, then $B=1$, but $A(B+1)=1\neq 2$, so it is also wrong.

   Then we can conclude that X^3+2X^2+1 is irreducible in $F_3[x]$.

   According to Theorem 2.38, X^3+2X^2+1 is an irreducible polynomial of degree 3 in $F_3[x]$, let F_{3^3} be the set of all the polynomial of degree less than 3 in $F_3[x]$, then F_{3^3} is a finite field with $3^3=27$ elements.

2. We can use 26 lower-case letters and define a map $\xi \leftrightarrow f(\xi)$, where ξ is one of 26 letters. That is, $a\leftrightarrow 1$, $b\leftrightarrow 2$, ..., $z\leftrightarrow 26$.

   Let $P(x)=X^3+2X^2+1$,

$$\begin{array}{ccc}
X^1\equiv X \bmod P(X) & X^2\equiv X^2 \bmod P(X) & X^3\equiv X^2-1 \bmod P(X)\\
X^4\equiv X^2-X-1 \bmod P(X) & X^5\equiv -X-1 \bmod P(X) & X^6\equiv -X^2-X \bmod P(X)\\
X^7\equiv X^2+1 \bmod P(X) & X^8\equiv X^2+X-1 \bmod P(X) & X^9\equiv -X^2-X-1 \bmod P(X)\\
X^{10}\equiv X^2-X+1 \bmod P(X) & X^{11}\equiv X-1 \bmod P(X) & X^{12}\equiv X^2-X \bmod P(X)\\
X^{13}\equiv -1 \bmod P(X) & X^{14}\equiv -X \bmod P(X) & X^{15}\equiv -X^2 \bmod P(X)\\
X^{16}\equiv -X^2+1 \bmod P(X) & X^{17}\equiv -X^2+X+1 \bmod P(X) & X^{18}\equiv X+1 \bmod P(X)\\
X^{19}\equiv X^2+X \bmod P(X) & X^{20}\equiv -X^2-1 \bmod P(X) & X^{21}\equiv -X^2-X+1 \bmod P(X)\\
X^{22}\equiv X^2+X+1 \bmod P(X) & X^{23}\equiv -X^2+X-1 \bmod P(X) & X^{24}\equiv -X+1 \bmod P(X)\\
X^{25}\equiv -X^2+X \bmod P(X) & X^{26}\equiv 1 \bmod P(X) &
\end{array}$$

   So X is a generator of F_{3^3}, and we can define the map as

$$\xi \to g(\xi): g(\xi)=X^{f(\xi)} \bmod P(X)$$

3. According to Part 2, the order of the subgroup generated by X is 26,

4. Use X as the generator and 11 as the secret key,

$$X^{11}\equiv X-1\equiv X+2 \bmod P(X)$$

   So $X+2$ is the public key.

5. Choose $k=18$, we can get

$$r\equiv X^{18}\equiv X+1 \bmod P(X)$$
$$\beta^k\equiv (X+2)^{18}\equiv \bmod P(X)$$

   Then we can map the message "goodmorning" into F_{3^3} as

$$X^2+1, -X^2, -X^2, X^2-X-1, -1, -X^2, X+1, -X, -X^2-X-1, -X, X^2+1$$

which can be encrypted by the equation

$$c \equiv \beta^k m \equiv (X+2)^{18} m \bmod P(X)$$

The result r is

$$X^2+X, X, X, -X^2+1, -X^2+X, X, X^2-X-1, 1, -X^2-X+1, 1, X^2+X$$

Mapping them back to letters, we get the ciphertext "saapyadzuzs".

Then we can use

$$m \equiv tr^{-x} \equiv t(X+1)^{-11} \bmod P(X)$$

The result m is

$$X^2+1, -X^2, -X^2, X^2-X-1, -1, -X^2, X+1, -X, -X^2-X-1, -X, X^2+1$$

So the plaintext is successfully decrypted.

Ex. 4 — Simple Questions

1. (i) Yes. We know $h(x) \equiv x^2 \bmod pq$, and we can find x by computing $\sqrt{h(x)} \bmod p$ and $\sqrt{h(x)} \bmod q$ and then use Chinese remainder theorem. However, p, q are large primes, the factorization of n is very difficult, so we can't efficiently find x.
   (ii) No. Given x, we can find $x' = -x$ so that $h(x) = h(x')$.
   (iii) No. For any x and x' so that $x' = -x$, we can find $h(x) = h(x')$.
2. (i) Efficiently computed for any input can be verified. Any length of message m can be computed into an 160 bits length result efficiently through xor.
   (ii) Pre-image resistant is not verified. Given y, let $m = y$, we can get $h(m) = y$.
   (iii) Second pre-image resistant is not verified. Given m, we can add 160 bits 0 after m to form m', so that $h(m) = h(m')$.
   (iv) Collision resistant is not verified. For any m and m' so that 160 bits 0 after m are added to to form m', we can find $h(m) = h(m')$.

Ex. 5 — Merkle-Damgård construction

1. a) Suppose the map s is not injective, that is, $\exists x \neq x'$ so that $y = y'$. Than we can apply the following strategy to examine. Let $y_0 = y$, if $y_{0,|y_0|-1}||y_{0,|y_0|} = 01$, we can find $x_{|x|} = x'_{|x'|} = 1$, and let $y_0 = y_{0,1}||\cdots||y_{0,|y_0|-2}$. Otherwise, if $y_{0,|y_0|} = 0$, we can find $x_{|x|} = x'_{|x'|} = 0$, and let $y_0 = y_{0,1}||\cdots||y_{0,|y_0|-1}$. Repeating the strategy until $|y_0| = 11$, we can find all bits of x and x' are the same, so $x = x'$, which makes a contradiction. So map s is injective.
   b) If z is empty, according to a), we know there is no strings $x \neq x'$ and z such that $s(x) = z||s(x')$.
   If z is not empty, let $a = z||s(x')$, we can find a substring 11 in $a_1||a_2||\cdots||a_{|a|}$. However, we can only find 11 in $s(x)_0||s(x)_1$, which makes a contradiction.
   So we can conclude that there is no strings $x \neq x'$ and z such that $s(x) = z||s(x')$.

2. From the two previous conditions, we know collisions can't be found through changing bits of input or adding paddings, which means the map s 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.

   Since $x \neq x'$, they are padded with two different values d and d', respectively. Similarly $k+1$ and $k'+1$ denote the number of blocks for x and x' .

   Since $t-1 = 0$, we don't need to consider $x \not\equiv x' \bmod (t-1)$ any more, then we can only consider $k = k'$ and $k \neq k'$.

   First, consider $k = k'$, this implies $y_{k+1} = y_{k'+1}$, and we have

   $$g(z_{k-1}||y_k) = z_k = h(x) = h(x') = z_k = g(z'_{k-1}||y'_k)$$

   If $z_{k-1} \neq z'_{k-1}$, a collision is found. Otherwise we repeat the process and get

   $$g(z_{k-2}||y_{k-1}) = z_{k-1} = h(x) = h(x') = z_{k-1} = g(z'_{k-2}||y'_{k-1})$$

   Then either we have found a collision or we continue backward until one is obtained. If none is found then we get $z_1 = z'_1, \ldots, z_k = z'_k$, which makes a contradiction.

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

   $$g(0^m||y_1) = z_1 = z'_{k'-k+1} = g(z'_{k'-k}||1||y'_{k'-k+1})$$

   By construction the m bit on the left is 0 while on the right it is 1. Hence we have found a collision.

   All the cases being covered this completes the proof.

Ex. 6 — Programming

In the ex6 folder, with a README file inside it.