Problem 1. Application of the DLP

1. (a) Because p is a prime and $\varphi(p) = (p-1)$. According to Bob's strategy, Alice will have following knowledge
$$\gamma \equiv \alpha^r \bmod p$$
$$y \equiv r \bmod (p-1) \text{ or } y \equiv x + r \bmod (p-1)$$
Knowing y she requested, she has either of two relations below
$$\alpha^y \equiv \alpha^r \equiv \gamma \bmod p$$
$$\alpha^y \equiv \alpha^{x+r} \equiv \gamma\alpha^x \bmod p$$
Since Bob knows that $x = \log_\alpha \beta$, Alice just need to compare if $\alpha^y \bmod p = \gamma\beta$ or γ.

   (b) Assume Bob doesn't have knowledge about $x = \log_\alpha \beta$, and he fakes
$$x' = \log_\alpha \beta$$
Then when Alice asks for $x + r \bmod (p-1)$, she will get
$$y' \equiv x' + r \bmod (p-1)$$
$$\alpha^{y'} \equiv \alpha^{x'+r} \equiv \gamma\alpha^{x'} \bmod p$$
Now that Alice can easily compute $\alpha^{x'}$ and compare with β, only if two results match then Bob could prove his identity.

2. (a) Since for every time Alice asks for
$$y \equiv r \bmod (p-1) \text{ or } y \equiv x + r \bmod (p-1)$$
Bob has 50% probability to fake having the knowledge. To ensure that the attacker has to at least apply 2^{128} operations, Alice needs to repeat the procedure for 128 times.

   (b) Apply the same strategy as 2(a), we need 192 times to ensure 256 bits security level.

3. Digital Signature Protocol.

Problem 2. Pohlig-Hellman

Algorithm.
Let G be a cyclic group of order n with generator g, let $h \in G$ and $x = \log_g h$, we have a prime factorization $n = \prod_{i=1}^r p_i^{e_i}$. For each $i \in \{1, \ldots, r\}$,

1. Compute $g_i = g^{n/p_i^{e_i}}$, which has order $p_i^{e_i}$
2. Compute $h_i = h^{n/p_i^{e_i}}$.

3. Compute $x_i \in \{0, \ldots, p_i^{e_i} - 1\}$ such that $g_i^{x_i} = h_i$. The procedure can be listed as follows
   (a) Initialize $x_0 = 0$.
   (b) Compute $\gamma = g^{p^{e-1}}$.
   (c) For each $k \in \{0, \ldots, e-1\}$, 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 $d_k \in \{0, ..., p-1\}$ such that $\gamma^{d_k} = h_k$ and set $x_{k+1} = x_k + p^k d_k$.
   (d) Return x_e
4. Solve the simultaneous congruence $x \equiv x_i \bmod p_i^{e_i}, i \in \{1, \ldots, r\}$
5. Return x.

Example. Calculate $\log_3 3344$ in $G = U(Z/24389Z)$
Given $G = U(Z/24389Z)$, and $24389 = 29^3$, the order of G is $n = 28 \cdot 29^2 = 2^2 \cdot 7 \cdot 29^2$. Knowing that 3 is a generator of G,

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

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

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

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

$$x_a = 2 \bmod 4$$

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

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

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

$$x_b = 2 \bmod 7$$

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

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

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

$$x_c = 260 \bmod 841$$

According to Chinese remainder theorem, we can simply get

$$\begin{aligned} x &\equiv 2 \bmod 28 \\ x &\equiv 260 \bmod 841 \\ 841 \cdot 1 &\equiv 1 \bmod 28 \\ 28 \cdot 811 &\equiv 1 \bmod 841 \end{aligned}$$

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

Problem 3. Elgamal

1. For $X^3 + 2X^2 + 1$ in $\mathbb{F}_3[x]$, the possible factors are $X + A$ and $X^2 + BX + C$.

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

Since $AC = 1$, we have

(a) $A = C = 1$, $A + B = 2 \to B = 1$, but $C + AB = 2$, fail.

(b) $A = C = 2$, $A + B = 2 \to B = 0$, but $C + AB = 2$, fail.

Therefore, $X^3 + 2X^2 + 1$ is not reducible over $\mathbb{F}_3[x]$.

Let $\mathbb{F}_{3^3}$ be the set of all the polynomial of degree less than 3 in $\mathbb{F}_3[X]$, which obviously has $3^3 = 27$ elements. Since $X^3 + 2X^2 + 1$ is not reducible over $\mathbb{F}_3[x]$ and has order 3, $X^3 + 2X^2 + 1$ defines $\mathbb{F}_{3^3}$.

2. Assume each letter in the alphabet represent a number from 1 to 26, we could define

$$\alpha \to f(\alpha) : X^\alpha \bmod P(X)$$

as the map, where α denotes any letter and $P(x) = X^3 + 2X^2 + 1$.

3. We could show by calculating all the elements generated by X.

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

Therefore, we can see X is the generator of $\mathbb{F}_{3^3}$, and the group order is 26.

4. Since we've known that 11 is the private key, we have public key calculated as

$$\beta = X^{11} \equiv X - 1 \text{ mod } P(X)$$

$$\beta = X - 1 = X + 2$$

5. First, we derive message m by mapping "goodmorning" to

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

Next, we pick random integer $k = 1$.

$$r = X^1 \equiv X \text{ mod } P(X)$$

$$t \equiv \beta^k m = (X+2)m \text{ mod } P(X)$$

which can be calculated respectively as

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

which can be mapped back to alphabet letters

$$rzzoxzcytyr$$

To decrypt, we just use

$$tr^{-11} \equiv tX^{-11} \text{ mod } P(X)$$

And the result gives

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

which can be mapped back to "goodmorning".

Problem 4. Simple Questions

1. Given $h(x) \equiv x^2 \text{ mod } n$. Find x.
   (a) It's pre-image resistant. Because the hardness of this problem equals to factorization of $n = pq$, where p, q are large prime. Therefore we cannot find x.
   (b) It's not second pre-image resistant, because $h(x) = h(-x) \equiv x^2 \text{ mod } n$.
   (c) It's not collision resistant, because for any $x, h(x) = h(-x) \equiv x^2 \text{ mod } n$.
2. (a) Efficiently comuputed for any input: Verified.
   (b) Pre-image resistant: Not verified. Given $h(m)$, for m has 160 bits, we have $h(m) = m$.
   (c) Second pre-image resistant: Not verified. Given m with 160 bits, we can find $m' = m\overbrace{0\cdots0}^{160\text{bits}}$, so that $h(m) = h(m') = m$.

(d) Collision resistant: Not verified. For any m with 160 bits and $m' = m\overbrace{0\cdots0}^{\text{160bits}}$, we can find $h(m) = h(m') = m$.

Problem 5. Merkle-Damgård construction

1. (a) If $x \to y$ is not injective, then suppose we have $x \neq x', y = y'$. Given map $y = y' = 11||f(x_1)||\cdots||f(x_x)$ and $f(0) = 0, f(1) = 01$. For all bits of $y_i = 0$, there must be $x_i = 0$, and for all $y_i = 01, x_i = 1$. Therefore, $x = x'$, and $x \to y$ is injective.

   (b) From(a), we see that there is no string $x \neq x'$, such that $s(x) = s(x')$.
   If $s(x) = z||s(x')$, since we can only find substring $11 = s(x)_0||s(x)_1$, when z is not empty, we are not able to find the substring $s(x)_i||s(x)_{i+1} = s(x')_0||s(x')_1$, therefore, we have a contradiction.

2. From 1, we have shown that the collision cannot be found either by exchange bits or add paddings, which means this construction is a collision resistant hash function.

3. Given compression function g defined from $\{0,1\}^{m+t} \to \{0,1\}^m$ and $t \geq 2$, we can prove that hash function h is collision resistant.

   Proof.

   Assume we have a collision on h, i.e. $x \neq x'$, and $h(x) = h(x')$. If $|x| \neq |x'|$, then 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'.

Case 1: consider $x \neq x'$ with $|x| \neq |x'| \bmod (t-1)$. Then $d \neq d'$ and $y_{k+1} \neq y'_{k'+1}$.
We then have

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

which is a conlision on g since $y_k \neq y_{k'}$

Case 2A: consider $|x| \equiv |x'| \bmod (t-1)$ with $k = k'$.
This implies $y_{k+1} = y'_{k'+1}$, 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 $z_{k-1} \neq z'_{k-1}$, then a collision is found. Otherwise we repeat the process and get

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

Then either we have found a collision or we continue backward until one is obtained. If none is found then we get

$$z_1 = z'_1, \ldots, z_k = z'_k$$

Case 2B: consider $|x| \equiv |x'| \bmod (t-1)$ with $k \neq k'$. Without loss of generality assume $k' > k$ and proceed as in the previous case. If no collision is found before $k = 1$ then we have

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

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