1. 108062313 黃紀易

(a.)

To encrypt a message $m=25$, choose a random integer k, and $1 \le k \le p-2$, let $k=2$.

$C=(C_1, C_2)=(g^k, my^k)=(16^2, 25 \cdot 59^2)=(7, 41)$

In order to decrypt a ciphertext $C=(C_1, C_2)$, we need to compute $C_1^{-x} \cdot C_2 = y^{-k} \cdot y^k \cdot m = m$, Hence $m = 58^{-29} \cdot 13 \bmod 83 = 16$ #

(b.)

Since $\gcd(k, p-1) = \gcd(23, 82) = 1$, we compute $k^{-1} \bmod p-1$

$r = g^k \bmod p$
$= 16^{23} \bmod 83$
$= 28 \bmod 83$

$s = k^{-1}(m - rx) \bmod p-1$
$= 23^{-1}(25 - 28 \cdot 29) \bmod 82$
$= 25 \cdot 33 \bmod 82$
$= 5$

#

2.

(a.) Let $k=2$, $H(9876543210) = 9876543210^{21} \bmod 37 \equiv 1$

$r = (g^k \bmod p) \bmod q$
$= (41^2 \bmod 149) \bmod 37$
$= 42 \bmod 37$
$= 5$

$s = k^{-1}(h(m) + rx) \bmod q$
$= 2^{-1}(1 + 5 \cdot 26) \bmod 37$
$= 19 \cdot 20 \bmod 37$
$= 10$

#

2.

(b.)

Condition $1 \le 12, 25 \le q-1$ holds.

$t = s^{-1} \bmod q$

$= 25^{-1} \bmod 317$

$= 3$

$v = ((g^{h(m)} y^r)^t \bmod p) \bmod q$

$= ((141^{31} \cdot 144^{12})^3 \bmod p) \bmod q$

$= (65 \bmod p) \bmod q$

$= 28$

Since $v \neq r$, $(12, 25)$ is not a valid signature for $m = 324$

3.

Let S be a probabilistic polynomial-time simulator for DL-based interactive proof system, where S is defined as:

(1) Guess $c' \in Z_2$ and choose $B \in_R Z_n^*$

(2.) Compute $A' = B^2 / y^{c'} \bmod n$.

(3.) Invoke $V(A') = c$

(4.) If $c' = c$, return (A', c', B) as an accepting transcript. otherwise, repeat the step (1).

The expected execution time of simulator S is $t_S = 2(t_B + t_V)$, where t_B is the execution time of step (1), (2) and (4)., and t_V is the execution time of verifer V.

The sequential DL-based interactive proof system is zero-knowledge because we can construct the probabilistic polynomial-time simulation S' to simulate the accepting transcript

$tr_{P,V}(x)$ by $tr_{S',V}(x)$ as follows:

```
for (i=0; i<k; ++i)
    cout << S(v);
```

The expected execution time of simulator S' is $t_{S'} = k \cdot t_S$.

The parallel DL-based interactive proof system is not zero-knowledge because we can only construct the probabilistic exponential-time simulator S' to simulate the accepting transcript $tr_{P,V}(x)$ by $tr_{S',V}(x)$ as follows:

(1) Guess $(c_1', c_2', \ldots c_k') \in_R (Z_2)^k$ and choose $B \in_R Z_n^*$

(2) Compute $A' = B^2 / \prod_{i=1}^{k} y_i^{c_i'} \bmod n$

(3) Invoke $V(A') = (c_1, c_2 \ldots c_k)$

(4) If $(c_1', c_2' \ldots c_k') = (c_1, c_2 \ldots c_k)$, return $(A', c_1', c_2' \ldots c_k', B)$ as an accepting transcript. Otherwise, repeat the step (a).

The expected time of simulator S' is $t_{S'} = 2^k (t_b + t_v)$, where t_b is the execution time of the step (1). (2). (4), and t_v is the execution time of verifier V.

4. Let $p = kq+1$ be a large prime, q is a large prime. $G = \langle g \rangle$ be a cyclic multiplicative group of order q. Suppose that $(A_1, A_2, \ldots A_n)$ are the n authorities and any t-out-of-n authorities can tally votes.

Each A_i select x_i and a $(t-1)$ degree polynomial $f(x) = \sum_{k=0}^{t-1} b_{i,k} x^k$ with $f_i(0) = x_i$, and publishes $h_i = g^{x_i}$ and $B_{i,k} = g^{b_{i,k}}$ for $0 \le k \le t-1$. Then the public key is $h = \prod_{i=1}^{t} h_i$ and the secret key is $x = \sum_{i=1}^{t} x_i = \sum_{i=1}^{t} f_i(0) = f(0)$, where $f(x) = \sum_{i=1}^{t} f_i(x)$.

Each A_i sends $S_{i,j} = f_i(j)$ to A_j via a secure channel. Each A_j checks whether $g^{S_{i,j}} \equiv \prod_{k=0}^{t-1} (B_{i,k})^{j^k} \bmod p$ and computes share $S_j = \sum_{i=1}^{t} S_{i,j} = \sum_{i=1}^{t} f_i(j) = f(j)$ ∎