دانشکده‌ی علوم ریاضی

مقدمه‌ای بر رمزنگاری

پاسخنامه تمرین شماره ۸

نگارنده: الهه کوشافر

For any questions contact Elahe Kooshafar via `kooshafar.elahe@gmail.com`.

Problem 1

For this purpose, we introduce a simple protocol in 3 steps for everyone in the group to follow.

1. Each party shares his input value among other parties and himself using Benaloh's secret sharing scheme.
2. After the last step, each person has received n shares from himself and others in the group. Each party calculates the sum of his received shares and announces the result to the group.
3. Each of the parties calculate the sum of the declared values.Sum of the declared values is equal to the desired sum. Therefore,

As the sum of the declared values by all parties in step 2 is equal to the desired sum, each party in step 3 will acheive the sum of the group's initial values.

Proof) Suppose n people in the group are $p_1, ..., p_n$ and their inital values are $v_1, \dots, v_n$, respectively. In step 1, each party shares their value among everyone in the group using Benaloh's scheme. Let's suppose that in this step, p_i will create shares $s_{i,1}, \dots, s_{i,n}$ and gives the share $s_{i,j}$ to p_j for $1 \leq j \leq n$. As a result of the secret sharing scheme we used, we have $v_i = s_{i,1} + \cdots + s_{i,n}$. In step 2, everyone has received their shares and announce the sum of their shares, let's call the declared values $x_1, \dots x_n$. Let's see what will the parties reach by calculating the sum of these declared values, in step 3.

$$\sum_{1 \leq i \leq n} x_i = \sum_{1 \leq i,j \leq n} s_{i,j} = \sum_{1 \leq i \leq n} v_i$$

Therefore, at the end of the proposed protocol, each party will have the sum of the group's values without gaining any further information about anyone else's value.

As for the security of the protocol, it is easy to see that any information about anyone's value would have to come from the shares the owner of that value had distributed among the group. We have discussed the security of Benaloh's secret sharing scheme in the lecture notes. As a result of Benaloh's security, our protocol is proved to be secure.

Problem 2

Perfectly hiding)

To show this, we need to prove that given commitment c, every value x is equally likely to be the value commited in c. We will do so by proving that given x, r and any $x' \in Z_q$, exists $r' \in Z_q$ such that $g^x h^r = g^{x'} h^{r'}$.

$$g^x h^r = g^{x'} h^{r'} \to g^x g^{ar} = g^{x'} g^{ar'} \to x + ar \equiv x' + ar' \to r' \equiv (x - x')a^{-1} + r \pmod q$$

Therefore, it was shown that for any $x' \in Z_q$, exists a unique $r' \in Z_q$ that would result in the commitment c (Note that a must be known to compute r')

Computationally binding)

The discrete logarithm problem is defined as: given a group G, a generator g of the group and an element h of G, to find the discrete logarithm to the base g of h in the group G.

We know that for the group G in the problem, solving the discrete logarithm problem is difficult, meaning no polynomial-time algorithm exists that can solve it.
We will show that if the sender can find different x and x' that both of which open commitment $c = g^x h^r$, then he can solve the discrete log problem. Suppose the sender knows x, r, x', r' s.t. $g^x h^r = g^{x'} h^{r'}$, because $h = g^a$, as explained in the previous part, this means that $x + ar \equiv x' + ar' \pmod q$. Therefore, the sender can compute a as $(x' - x)(r - r')^{-1}$. But this means that the sender could compute the discrete logarithm of h in polynomial time! which contradicts with the fact that the discrete logarithm problem is difficult for G, as stated above. Therefore, the sender can not find such x', r' in polynomial time, and as a result, the commitment is computationally binding.

Problem 3

Part a)

We have,

$$B_2 = B_1^x \Leftrightarrow (\frac{A_2}{g^b})^r A_0^s = (A_1^r g^s)^x \Leftrightarrow (g^{xy+a-b})^r g^{xs} = g^{yrx} g^{sx} \Leftrightarrow xy+a-b \equiv xy \Leftrightarrow a \equiv b \pmod{p}.$$

Therefore, Alice could calculate B_1^x and compare it with B_2, if they were equal she would know that they live in the same province, and otherwise, she would know that they live in different provinces.

We claim that in case they are not in the same province, Alice will not gain any further information about Bob's province. We will do so by proving that given r, s, b and any $b' \in \{1, \ldots, 50\}$ that $b' \neq a$, there exist unique $r', s' \in Z_p$ such that for $\langle B_1', B_2' \rangle$ computed from r', s' and b', the equality $\langle B_1, B_2 \rangle = \langle B_1', B_2' \rangle$ would hold. This equality means $\frac{B_2}{B_1^x} = \frac{B_2'}{(B_1')^x}$, which results in $g^{(a-b)r} = g^{(a-b')r'}$. Because $a \neq b'$, the element $a - b'$ has a multiplicative inverse element in G.

$$g^{(a-b)r} = g^{(a-b')r'} \Leftrightarrow (a-b)r \equiv (a-b')r' \Leftrightarrow r' \equiv (a-b')^{-1}(a-b)r \pmod{p}$$

From $B_1 = B_1'$ it can be concluded that

$$B_1 = B_1' \Leftrightarrow g^{yr+s} = g^{yr'+s'} \Leftrightarrow yr + s \equiv yr' + s' \Leftrightarrow s' = yr + s - yr'.$$

Therefore, unique r', s' exist that the equality $\langle B_1, B_2 \rangle = \langle B_1', B_2' \rangle$ would hold. As a result, Alice can gain no further information about Bob's province if $a \neq b$.

Part b)

Consider a cyclic group G of order q, and with generator g. The DDH assumption states that no efficient algorithm can distinguish between the two distributions $\langle g^a, g^b, g^{ab} \rangle$ and $\langle g^a, g^b, g^c \rangle$ where a,b,c are chosen at random in Z_q.

Lemma1. Suppose X_0 and X_1 are two distributions that $X_0 \simeq X_1$, if M is an efficient algorithm we have that $M(X_0) \simeq M(X_1)$.

Lemma2. Suppose X_0, X_1 and X_2 are three distributions that $X_0 \simeq X_1$ and $X_1 \simeq X_2$. We have that $X_0 \simeq X_2$.

From the DDH assumption we have that $\langle g^x, g^y, g^{xy} \rangle \simeq \langle g^x, g^y, g^z \rangle$ where x, y, z are chosen at random in Z_q. Using lemma1, we can get $\langle g^x, g^y, g^{xy+a} \rangle \simeq \langle g^x, g^y, g^{z+a} \rangle$ (1). As a and z are both random from Z_q, we have that $\langle g^x, g^y, g^{z+a} \rangle \simeq \langle g^x, g^y, g^z \rangle$ (2).

Applying Lemma2 on (1) and (2) results in $\langle g^x, g^y, g^{xy+a}\rangle \simeq \langle g^x, g^y, g^z\rangle$. Therefore, what Bob sees, can not be distinguished from the random distribution of $\langle g^x, g^y, g^z\rangle$ by any efficient algorithm. As a result, Bob can gain no information about Alice's province in polynomial time.