University of New Brunswick
Faculty of Computer Science
CS4413/6413: Foundations of Privacy
Theory Homework Assignment 2, ***Due Time, Date*** *5:00 PM, March 28, 2019*

Student Name: ____________________________ Matriculation Number: ________________

Instructor: Rongxing Lu
The marking scheme is shown in the left margin and [100] constitutes full marks.

[**20**] 1. Please prove the following results.

[10] (a) Let p be a prime number, and $a^p \equiv b^p \bmod p$, prove $a^p \equiv b^p \bmod p^2$.

[10] (b) Let $\gcd(m,n) = 1$, prove $m^{\phi(n)} + n^{\phi(m)} \equiv 1 \bmod mn$.

Answer:

(a) $a^p - b^p = (a-b)(a^{p-1} + a^{p-2}b + \cdots + b^{p-1}) = (a-b)\Sigma_{i=0}^{p-1} a^{p-1-i}b^i$.

Since a and b are prime numbers, according to the Femat's Little Theorem,

$a^{p-1} \equiv 1 \mod p$ and $b^{p-1} \equiv 1 \mod p$.

So $a^p \equiv b^p \mod p \Rightarrow a^{p-1} \cdot a \equiv b^{p-1} \cdot b \mod p \Rightarrow a \equiv b \mod p$.

$a - b \equiv 0 \mod p \Rightarrow p|(a-b)$.

In addition, for any $i \geq 0$, $a^i \equiv b^i \mod p$, so

$\Sigma_{i=0}^{p-1} a^{p-1-i}b^i \equiv \Sigma_{i=0}^{p-1} a^{p-1-i}a^i \equiv \Sigma_{i=0}^{p-1} a^{p-1} \equiv \Sigma_{i=0}^{p-1} 1 \equiv p \equiv 0 \mod p \Rightarrow p|\Sigma_{i=0}^{p-1} a^{p-1-i}b^i$.

Then, $p|(a-b)\Sigma_{i=0}^{p-1} a^{p-1-i}b^i \Rightarrow p|(a^p - b^p) \Rightarrow a^p \equiv b^p \mod p$.

(b) According to the Euler's theorem,

$$gcd(m,n) = 1 \Rightarrow \begin{cases} m^{\phi(n)} + n^{\phi(m)} & \equiv & 1 \bmod n \\ m^{\phi(n)} + n^{\phi(m)} & \equiv & 1 \bmod m \end{cases}$$

According to the Chinese Remainder Theorem,

$a_1 = 1, a_2 = 1, m_1 = n, m_2 = m \Rightarrow M = mn, M_1 = m, M_2 = n$

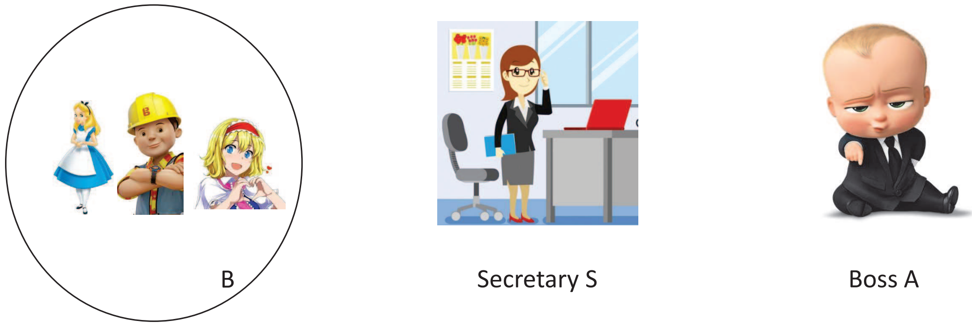
$gcd(m,n) = 1 \Rightarrow \exists a, b$ such that $am + bn = 1$

$M_1^{-1} \bmod m_1 \Rightarrow m^{-1} \bmod n \equiv a$ and $M_2^{-1} \bmod m_2 \Rightarrow n^{-1} \bmod m \equiv b$.

$m^{\phi(n)} + n^{\phi(m)} \equiv (a_1 \cdot (M_1^{-1} \bmod m_1) \cdot M_1 + a_2 \cdot (M_2^{-1} \bmod m_2) \cdot M_2) \bmod M$

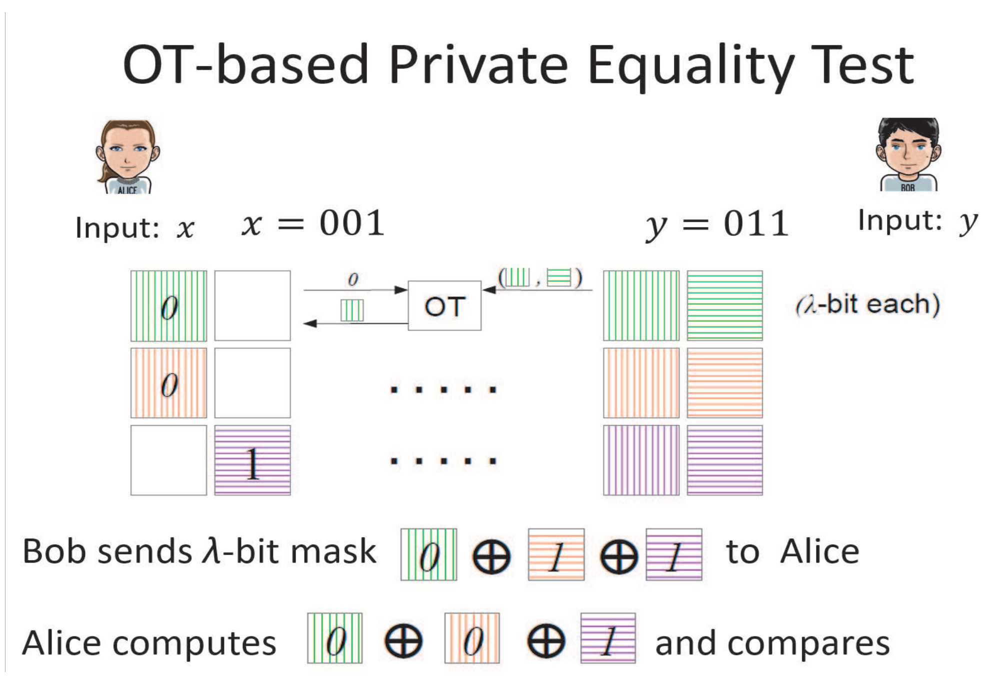
Thus, $m^{\phi(n)} + n^{\phi(m)} \equiv am + bn \equiv 1 \mod mn$.

[**30**] 2. Boss A has a list of keywords $K = \{k_1, k_2, \cdots, k_n\}$ and a set of friends $B = \{B_1, B_2, \cdots, B_n\}$. Boss A asks his secretary S to only forward messages (that include at least one keyword in K and the sender belongs to B) to him. The conditions are i) the secretary S cannot know K; ii) the secretary cannot know B and the message content. (Hint: you can apply the symmetric key encryption and the bloom filter techniques.)



Answer: The set of friends generates and shares a security key SK, and sends SK to Boss A. After receiving SK, Boss A encrypts $\{k_1, k_2, \cdots, k_l\}$ as $\{H(k_1||SK), \cdots, H(k_n||SK)\}$. Then, he/she generates a Bloom Filter array and stores encrypted keywords $\{H(k_1||SK), \cdots, H(k_n||SK)\}$ to the Bloom Filer array. Boss A also sends the Bloom filter array to his secretary S. When B_j sends a message to Boss A, he/she will encrypt each keyword k_j in the message as $H(k_j||SK)$, and then send these encrypted keywords to the secretary S. After receiving the message, S uses the Bloom Filter array to check whether there is a keywords belonging to the keywords set K. If yes, forward to the Boss A. Otherwise, drop out the message.

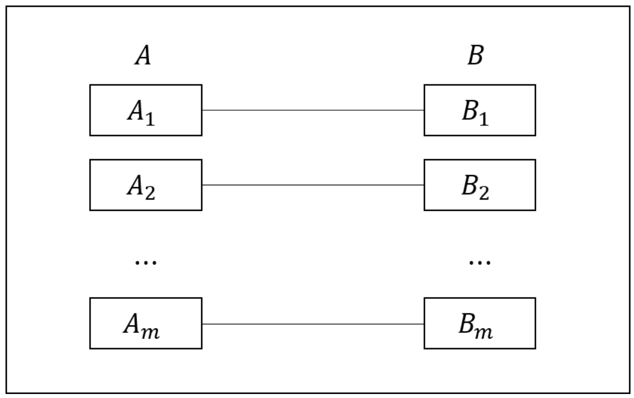
[**25**] 3. Assume Alice, Bob and Carlo respectively have the data set $A = \{\cdots\}$, $B = \{\cdots\}$, $C = \{\cdots\}$. How to use the OT-protocol to design a Private Set Intersection protocol among Alice, Bob, and Carlo, so that each one can obtain $A \cap B \cap C$. You can design your solution based on the OT-based PET (Private Equality Test) protocol in the figure.



Answer: The intersection of $A \cap B$ can be computed as follows.

- As shown in the following figure, put the elements of A and B into m hash tables $\{A_1, A_2, \cdots, A_m\}$ and $\{B_1, B_2, \cdots, B_m\}$, respectively.
- Compute $A_i \cap B_i$ by using the OT-based private equality test protocol to compare each element of A_i with each element of B_i for $i = 1, 2, \cdots, m$.
- Then, $A \cap B = (A_1 \cap B_1) \cup (A_2 \cap B_2) \cup (A_m \cap B_m)$.

Finally, $A \cap B \cap C$ is the intersection of $A \cap B$ and C, and can be computed with the same method.

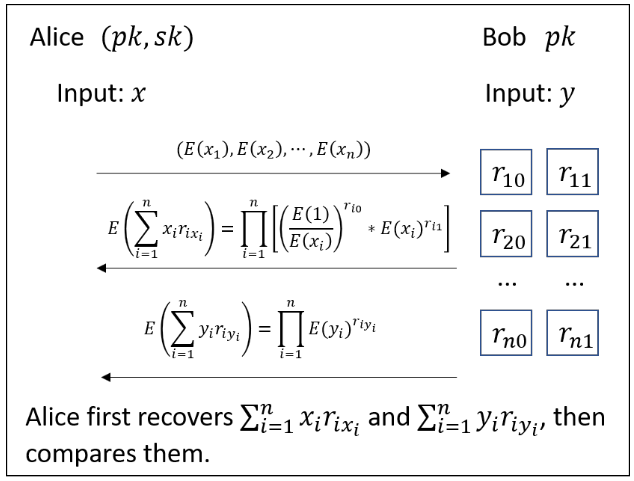


[**25**] 4. How to use the OT-protocol to design a privacy-preserving integer comparison protocol between two parties, e.g., two integers x, y, both of them are n bits, where $x > y$, $x < y$, or $x = y$. You can design your solution based on the above OT-based PET protocol. (Hint: you may disclose two bits information in your solution!)

Answer: Suppose that Alice has x and Bob has y, and x and y can be denoted as $x = x_1 x_2 \cdots x_n$ and $y = y_1 y_2 \cdots y_n$, respectively. As shown in the following figure, Alice and Bob can use the Paillier homomorphic encryption technique and OT protocol to compare x and y. In specific, Alice has public key pk and private key sk, while Bob only has public key pk. They can compare x and y as follows.

- Alice encrypts x as $E(x) = (E(x_1), E(x_2), \cdots, E(x_n))$. Then, he/she sends $E(x)$ to Bob.
- On receiving $E(x)$, Bob selects $2n$ random positive numbers $\{r_{i0}, r_{i1} | i = 1, 2, \cdots, n\}$ such that $r_{i1} > r_{i0} + \sum_{j=i+1}^{n} r_{j1}$. Then, Bob computes $\prod_{i=1}^{n}[(\frac{E(1)}{E(x_i)})^{r_{i0}} * E(x_i)^{r_{i1}}]$, i.e., $E(\sum_{i=1}^{n} x_i r_{ix_i})$, and returns it to Alice. At the same time, Bob computes $\prod_{i=1}^{n} E(y_i)^{r_{iy_i}}$, i.e., $E(\sum_{i=1}^{n} y_i r_{iy_i})$, and sends it to Alice.
- Alice recovers $\sum_{i=1}^{n} x_i r_{ix_i}$ and $\sum_{i=1}^{n} y_i r_{iy_i}$ from $E(\sum_{i=1}^{n} x_i r_{ix_i})$ and $E(\sum_{i=1}^{n} y_i r_{iy_i})$, respectively. Then, he/she compares them to obtain the comparison result of x and y.

Since the positive numbers satisfy that $r_{i1} > r_{i0} + \sum_{j=i+1}^{n} r_{j1}$ for $i = 1, 2, \cdots, n$, the comparison result of $\sum_{i=1}^{n} x_i r_{ix_i}$ and $\sum_{i=1}^{n} y_i r_{iy_i}$ is equal to that of x and y.



Alice (pk, sk)
Bob pk
Input: x
Input: y
$(E(x_1), E(x_2), \cdots, E(x_n))$
r_{10} r_{11}
r_{20} r_{21}
r_{n0} r_{n1}
$E\left(\sum_{i=1}^{n} x_i r_{ix_i}\right) = \prod_{i=1}^{n}\left[\left(\frac{E(1)}{E(x_i)}\right)^{r_{i0}} * E(x_i)^{r_{i1}}\right]$
$E\left(\sum_{i=1}^{n} y_i r_{iy_i}\right) = \prod_{i=1}^{n} E(y_i)^{r_{iy_i}}$
Alice first recovers $\sum_{i=1}^{n} x_i r_{ix_i}$ and $\sum_{i=1}^{n} y_i r_{iy_i}$, then compares them.