Secure multi-party computation

Homework 2

Najah Kamal 325829133

Salam Qais 327876116

Sami Serhan 327876298

Semester 1

This is the truth table for : $f_{a,4}(x) = \begin{Bmatrix} 1 & if\ ax \geq 4 \\ 0 & otherwise \end{Bmatrix}$

$input$	$x = 0$	$x = 1$	$x = 2$	$x = 3$
$a = 0$	0	0	0	0
$a = 1$	0	0	0	0
$a = 2$	0	0	1	1
$a = 3$	0	0	1	1

So our matrix going to be of size 4*4

First to solve that we will do it according to the algorithm:

So first of all we created the dealer class and with out it nothing going to work
In the dealer class we does all of the computation of offline phase(pre-processing)

1. Get two random number $r, c < -_r\{1, \dots, 4\}$ and shift the truth table based on this value
2. Get a random 2*2 matrix (M_b)
3. Compute $M_a = M_b[i,j] \oplus T_{r,c}[i,j]$ // 1 operation

All of that computation going to be In the __init__ and it's stored in private variable of the dealer class assuming trusted dealer

4 . $(r, M_a) to\ alice\ when\ she\ aske\ (c, M_b)\ to\ bob$

In the dealer class we will have two functions rand_a which sends massage to Alice and Rand_b send to bob

In the other two class we do the things needed the in the online phase:

Alice calculate it u and send it to bob by send() and receive() function for receiving back the calculation from bob (v,zb)

And then just after that Alice able to show the output.

As we proven in the lecture the correctness of OTTT:

$$
\begin{aligned}
z \quad &= M_A[u, v] \oplus z_B && //\text{def of } z \\
&= M_A[u, v] \oplus M_B[u, v] && //\text{def of } z_B \\
&= T[u - r, v - c] && //\text{def of } M_A \\
&= T[x, y] && //\text{def of } u, v \\
&= f(x, y) && //\text{def of } T
\end{aligned}
$$

And the privacy of it:

The real view is:

$$view_A = (A's\ input, A\ randomness, msgs\ received)$$

$$= \left(x, \perp, (\mathrm{r}, \mathrm{M_A}, v, z_B)\right)$$

$$view_B = (B's\ input, B\ randomness, msgs\ received)$$

$$= \left(a, \perp, (\mathrm{c}, \mathrm{M_B}, u)\right)$$

simulator for Alice

$\boldsymbol{S_A}$: $input\ x \in \{0,1,2,3\}\ \ z \in \{0,1\}$

$sample\ \ z_B \leftarrow \{0,1\}$, $v, r \leftarrow_r \{0,1\}^n$

Construct M_A follows:

M_A [x+r , v] = z $\oplus z_B$

M_A [i , j] $\leftarrow_R$ {0,1} $\forall$(i , j) ≠ (x+r , v)

Output : "simulated-view" := (x, $\perp$, (r , M_a , v, z_B)

In both views

x is identical (the input)

r, v and M_A [i , j] for all (i , j) ≠ (u , v) are independent. uniformly random

(M_A [u , v], z_B) is uniformly random subject to M_A [u, v] $\oplus Z_B$= z .

So (real) $view_A \equiv$ simulated- $view_A$

simulator for Bob

$\boldsymbol{S_B}$: $input\ a \in \{0,1,2,3\}$

$sample \quad c, u \leftarrow_r \{0,1\}^n$

Construct M_B as follows:

M_B [i , j] $\leftarrow_R$ {0,1} $\quad\quad\quad\quad$ $\forall$(i , j)

Output: " simulated-view " := (a, $\perp$, (c , M_B , u)

In both views

a is identical (the input)

c, u and M_B [i , j] for all (i , j) are independent. uniformly random

So (real) $view_B \equiv$ simulated- $view_B$

Time complexity:

Offline phase:

The dealer have truth table from length 4*4

the dealer make XOR between Ma Xor Mb->we have 16 XOR

in this example the Time is O(1) ,Because the inputs between [0,3],but if we assume that the input between [0,n] we will have complexity =O(2^n) for the offline phase

Online phase:

Alice computes u -> make one +

Bob computes v -> make one +

Alice make Xor between the Zb with Ma-> One Xor

So we make 3 opreations->T=3

Test.py:

this file make check for all the possible inputs of a and x (a,x->{0..3})

so we have 16 inputs ,and the test file check if every input get the answer that we need to get

We have 16 inputs=16 checks -> we have 16T(T is the number of operations that Alice and Bob make)

T=3

T is CONST in this case -> 16T=48=CONST -> complixety time =O(1)