CS 203.4860 **Secure Multi-Party Computation** Fall 2021

Homework 3: Report

Lecturer: Dr. Adi Akavia *Student(s): (Aya)*

Work space:

We used Python to implement BeDOZa Protocol

Classes,Functions and implementation:

BeDOZa Dealer:

We deal with this class as the dealer in this class through the function init we define random a,b (0 or 1 values) and define c that is equal to a*b - Beaver triples (same as u,v,w in lecture 3), we defined random a_0 b_0 c_0 (0 or 1 values) (same as u_a v_a w_a in lecture 3) and we defined $a_1 = a_0 + amod2$, $b_1 = b_0 + bmod2$, $c_1 = c_0 + cmod2$ (secret share) and add these variables to arrA and arrB wich are arrays of random values for Alice and Bob, we repeat this t(size according to our code) times. We defined also RandA and RandB functions which return the arrays of random values to Alice and Bob.

BeDOZa Agent:

Instead of defining classes as Alice and Bob we defined BeDOZa Agent which has same functionality as Alice and Bob, We added more than init function to this class , first one is getting the circle and the iteration number and the second init gets the input and the Beaver Tripes that were defined by the dealer. Same Send and Recieve functions as we defined in lecture 3.

Sum Circuit:

We defined the boolean circle as we defined it in assignment 1 Problem 6, we defined the circuit this way: e will first create a circuit c_1, evaluating function $sum^4 : \{0,1\}^3 \times \{0,1\}^3 \to \{0,1\}^3$ (considering the inputs as two numbers in the range $(0, 7)$) with the following properties

$$x + y \geq 4 \Rightarrow sum^4(x, y) \geq 4$$

$$x + y < 4 \Rightarrow sum^4(x, y) = x + y$$

We will right formulas for the output of the circuit

$$O_0 = x_0 + y_0$$

$$O_1 = (x_1 + y_1) + (x_0 \cdot y_0)$$

$$O_2 = 1 + ((1 + x_1 \cdot y_1)((1 + x_2)(1 + x_1 \cdot (x_0 \cdot y_0)))((1 + y_2)(1 + y_1 \cdot (x_0 \cdot y_0))))$$

It easy to see that the composition of sum^4 is ≥ 4 iff the sum is ≥ 4, therefore we can use it to sum numbers. Additionally multiplication of $x = (x_0, x_1), y = (y_0, y_1)$ can be represented as $(x_0 \cdot y_0, x_0 \cdot y_1, 0) + (0, x_1 \cdot y_0, x_1 \cdot y_1)$, when for summation we can use sum^4

now we can directly compute $x_0 \cdot \alpha_0 + x_1 \cdot \alpha_1$ via sum^4 as showed above, and then output the msb of $x_0 \cdot \alpha_0 + x_1 \cdot \alpha_1$ indicating if it is grater or equal to 4. for implementing the circuit this way we also defined functions such as: changeVars that utilities to create a circuit,addToVars, appendHorizontaly, copyOf, mulSizeOfCirc.

We defined the main circuit in Equation3 array.

Function runner:

In function runner we get the inputs of Alice and Bob (x,y or x,a) and return the output/circuit(x,y) we do that using the implementation of classes BeDOZaDealer BeDOZaAgent and SumCircuit that we wrote before.

Test and complexity:

We tried all the possible inputs of a_0, a_1, x_0, x_1 with 100 repetitions of the tests (To eliminate the effect of randomness) and checked if all the circuit outputs are right through these inputs, and measured the communication complexity and got always 73.

Execution Time:

We also calculated the executing time by using package time and subtract the start time from the time when we end running the tests and we got different times of execution such as 9.218376636505127 seconds, 10.232978582382202 seconds, 8.361205816268921 seconds (for $100 \cdot 4^4$ repetitions, resulting in 0.4ms per run of the circuit).