CSA E0 312: Secure Computation 11 Aug 2015

Lecture 1

Instructor: Arpita Patra *Submitted by: N S Bharath Kumar*

1 Motivation

The Earth is orbited by nearly 7000 man-made satellites and more than 21000 orbital debris larger than 10 centimeters. The growing number of satellites and space debris orbiting the planet is increasing the danger of collisions. This is not a hypothetical scenario. There are many such reported collisions. In 2009, two communication satellites belonging to the US and Russia collided in orbit completely destroying both the satellites. Most recently in 2013 debris from the destroyed Chinese satellite Fengyun 1C collided with a small Russian laser-ranging retro-reflector satellite called BLITS. Hence there is a necessity to calculate that whether any two satellites are on collision path to avoid such losses. But the challenge is that the orbital parameters of satellite is a national secret. Any country wouldn't like to share such information . Hence there is a need of trusted third party who on giving these orbital data will come out with the information that whether the satellites will collide or not, but will never reveal out the orbital data to anybody . But unfortunately no such trusted third party exist in this world .

One more classic example of protecting the privacy of individual and also to achieve fairness comes in case of auction . Ex:- auctioning of 3G spectrum . Here nothing other than the winner and winning bid should be revealed.

A lot of datamining is being done today to understand the nature or trend thats going on in various arenas . One example is in the field of medical science where census of medical data over years needs to be analysed to understand the trend of disease or mutation of microbes . But no hospital would like to reveal out sensitive information that they are holding which is needed for this analysis .

Hence the above examples show the need for some SECURE COMPUTATION where the input data of parties are never revealed but the necessary output is computed correctly.

2 Introduction

In the present world we find information every where . There is lot of data from various entities which needs to be processed jointly by different parties. But these data may be public or private based on the level of secrecy which needs to be maintained.The sources of data and nature of secrecy of such data is tabulated below From above listed table we come to know that a lot of data is confidential in nature .But a large amount of added value can be obtained by combining confidential information from several sources and from this computing some result can be obtained that holds an interest for all parties .

Source of data	Public data	Private Data
Individual	Identity details (passport no., PAN card, Voter ID, AADHAR id), Bank Details (netbanking login), Income Tax Details, Your vehicle details (cycle, two wheeler, car)	Age, Salary Bank word),Medical dat ric traits (face, fi genome signature, ing porn/taking dr tails
Educational Organization (IISc/IITs/IIITs/IISERs/NISERs/NITs)	List of Employees students and , awards, recognitions, scientific publications, products	Employee details,s details, drug addic rassments
Profitable Organization (MS/IBM/TCS/Infosys):		List of employees a loss, turnover, sala
Hospitals	List of patients doctors, nurses	Patient's medical h tails , Details of D
Security Agencies (RAW/ IB/ CBI/NIA)	list of criminals and details, list of incidents and details	List of employees a cepted messages a
Military Organizations (Army/Air Force/Navy)	List of soldiers, colonels and details, list of operations	Operation details , sages etc and their
Country	List of citizens and details, prime minister, presidents, MLA, MPs, celebrities, underprivileged	Satellites / Nuclear information

Table 1: Types of data and their resources

3 Secure multi party computation : Definition

Now that we know the problem of security in multi party computation let us try to define secure multi party computation. There are n parties $p_1, , , p_n$ .Each p_i holds a secret input x_i . They would like to compute y= f($x_1, , , , , , X_n$) while making sure that following conditions are met

- Correctness:- The correct value of y is computed
- Privacy :Nothing about the inputs is revealed. In other words nothing more than the function output should be revealed.

4 Trusted Third party

The above definition can be met if we have a trusted third party who can take x_i input from p_i party , compute y and give it back to all p_is . But this trusted third party doesn't exist in this real world . If at all it exists , it may become a single point of failure in case it turns corrupt. Hence such things doesn't exist in real world. Even if it exists nobody will trust it due to the existence of distrust.

5 A simple Secure Computation in addition and voting function

With the above preamble in mind we will now try to explore whether we can achieve secure computation on simple functions like addition or vote counting in a scenario where we don't have trusted third party (TTP).An function like addition requires n inputs from "n" parties which sums up these input and provides the sum as output . A voting function counts the vote of each party and declares the result . In both the cases the input must stay as a secret. Before we proceed with this , we need to define the concept of secret sharing . **Definition**

1 Secret Sharing: It provides a way for a party, say P_1 to spread information about a secret x across all the parties so that together they hold full information about x and yet an individual (or subset of parties) has no information about x ♢

6 Secret Sharing Instantiation

The secret s will belong to Z_p = {0,1,,,,,p-1} where p is prime.

Theorem 1 *F_p = (Z_p, +modp(+), .modP(.)) is a field . It satisfies closure and associative property . It has identity element 0 (addition modulo p operation) and 1 (multiplication modulo p operation). For every a ∈ Z_p ,∃ , -a, a^{-1} such that a ⊕ (-a) = 0 and a ⊗ (a^{-1})=1. Also .mod p is distributive over +mod p.*

It is necessary to highlight here that the secret s ∈ F_p. To instantiate the secret s, we will choose from random shares s_1 ,,,, s_n such that each $s_i \in F_p$ and $s_1+, s_2+, , s_n$ = s. We will denote these set of s_i as S ={ $s_1, s_2, , s_n$}. Now we will be sharing this secret with n parties such that each party p_i has S/s_i. That is party p_i has all the elements of S with him other than s_i.Hence we ensure following things.

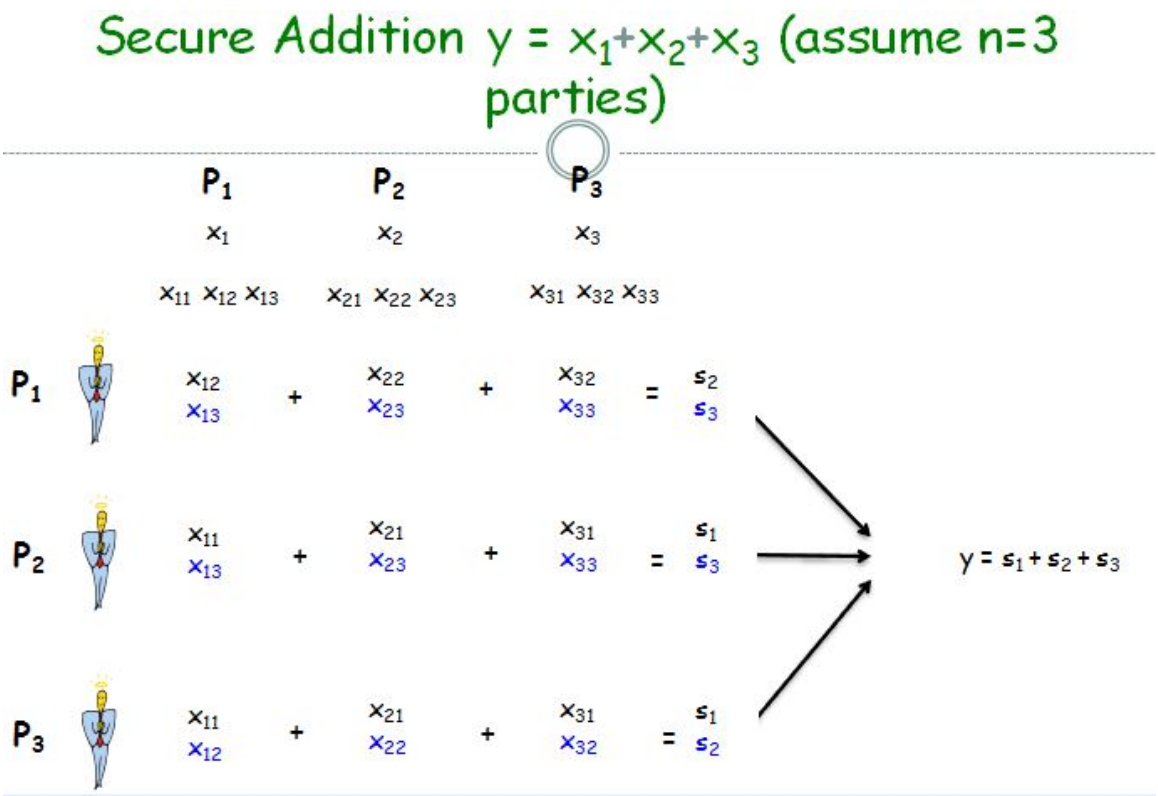
- Together all the parties know s (in fact any two parties know s)
- Individual party has no information about s
- Pr{Guessing s before secret sharing }= Pr{Guessing s after secret sharing}

7 Secure addition using Secret Sharing

Let us consider three party addition , where p_1, p_2 and p_3 have their secret inputs for addition as x_1, x_2 and x_3.
p_1 splits his input x_1 into three parts x_{11}, x_{12} and x_{13}. He keeps input { x_{12} x_{13}} with himself and gives { x_{11} x_{13}},{ x_{11} x_{12}} to p_2 and p_3 respectively.
p_2 splits his input x_2 into three parts x_{21}, x_{22} and x_{23}. He keeps input { x_{21} x_{23}} with himself and gives { x_{22} x_{23}},{ x_{21} x_{22}} to p_1 and p_3 respectively.
p_3 splits his input x_3 into three parts x_{31}, x_{32} and x_{33}. He keeps input { x_{31} x_{32}} with himself and gives { x_{32} x_{33}},{ x_{31} x_{32}} to p_1 and p_3 respectively.
Now each party p_i sums up all the shares that he has with him . that is

- sum of p_1 will be $s_2 = x_{12} + x_{22} + x_{32}$ and $s_3 = x_{13} + x_{23} + x_{33}$
- sum of p_2 will be $s_1 = x_{11} + x_{21} + x_{31}$ and $s_3 = x_{13} + x_{23} + x_{33}$
- sum of p_3 will be $s_1 = x_{11} + x_{21} + x_{31}$ and $s_2 = x_{12} + x_{22} + x_{32}$

Finally we get sum of three inputs as y = $s_1 + s_2 + s_3$. The same calculation is depicted pictorially below



8 Impossibility of bit multiplication using Secret Sharing

It is not possible to perform bit multiplication using secret sharing technique as used in addition . It is Because of as explained below:-

- Assume n =2 parties
- Let p_1 split his input x_1 as x_{11} and x_{12}. He keeps x_{12} with himself and gives x_{11} to p_2
- similarly p_2 split his input x_2 as x_{21} and x_{22}. He keeps x_{21} with himself and gives x_{22} to p_1
- The product of x_1 x_2 will be ($x_{11} + x_{12}$) ($x_{21} + x_{22}$) =x_{11} .$x_{21} + x_{11}$. $x_{22} + x_{12}$. $x_{21} + x_{12}.x_{22}$

- p_1 can only calculate x_{12} . x_{22} and p_2 can only calculate x_{11} .x_{21} . Hence products x_{11} . x_{22} and x_{12} . x_{21} are left out .

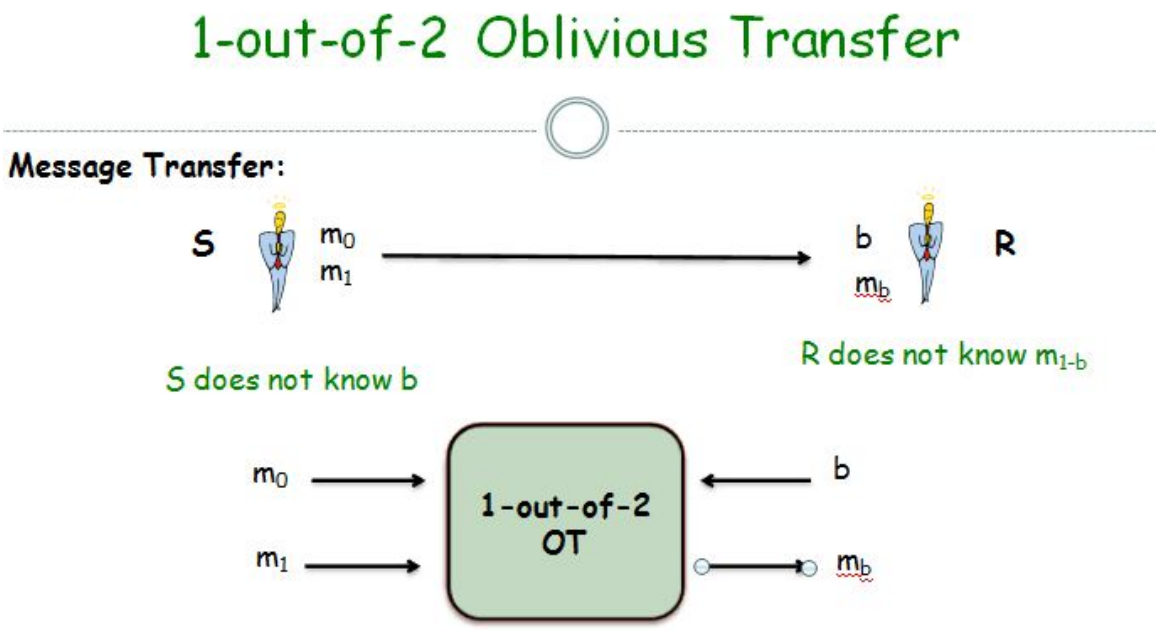
Thus the concept of secret sharing doesn't work here .

9 1-out-of-2 Oblivious Transfer

The above problem can be solved by using 1-out-of-2 Oblivious Transfer (OT) mechanism. Let us consider the following case.

- Let Sender S have two messages m_0 and m_1.
- Let receiver R have an input b $\in$ {0,1}.
- Message oblivious transfer is sending of message m_b from R to S on receipt of b from R as input .
- On conclusion of OT the sender will lot be aware of input value b and receiver will not be aware of m_{1-b}.

Below is the pictorial depiction of message transfer using OT.



Bit multiplication using OT

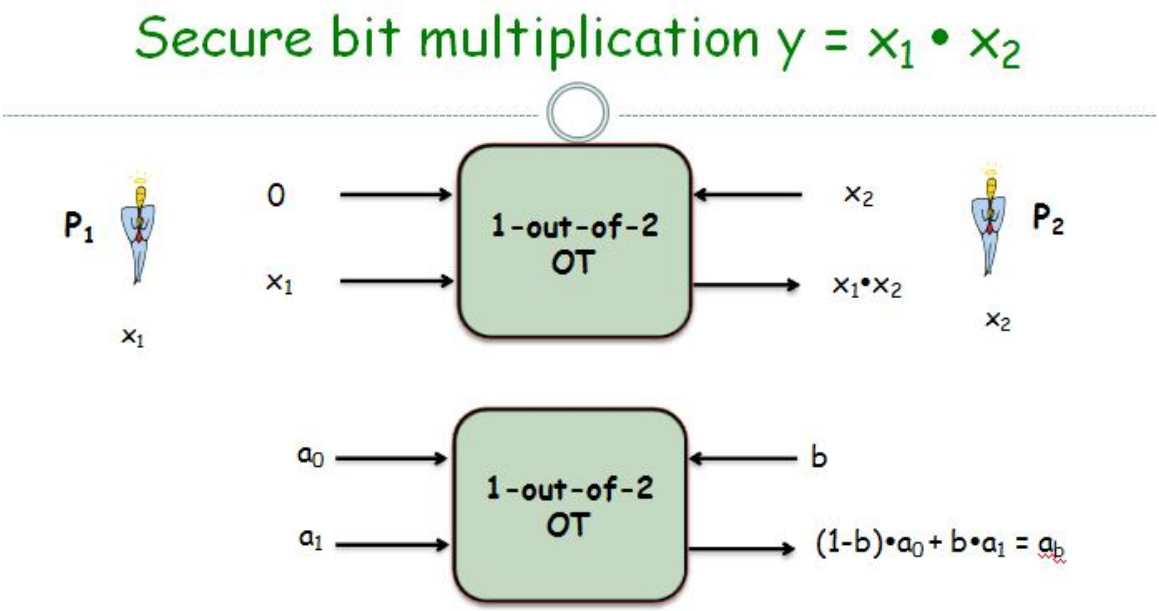
- P_1 plays the senders part and P_2 is the receiver.
- P_1 sets the senders inputs to be x_0=0, x_1=a.

- They run an OT protocol, and P_2 sends the final answer to P_1.
- The output is (1-j) x_0+j·x_1 = (1-b)j·0+b·a = a·b .

Privacy in Bit multiplication using OT

- If b=0 then the result that P_2 obtains in the OT protocol is always 0 and does not reveal anything about a.
- If b=1 then the result obtained in the OT protocol is equal to a, but it is also equal to a·b which is the legitimate output of P_2.

Below is the pictorial depiction of bit multiplication using OT.



10 Conclusion

In this lecture we have given introduction to Secure computation and its utilities . We have also come across two primitives namely **Secret Sharing** and **Oblivious Transfer** which are going to form the building blocks of secure multi party computation (MPC)