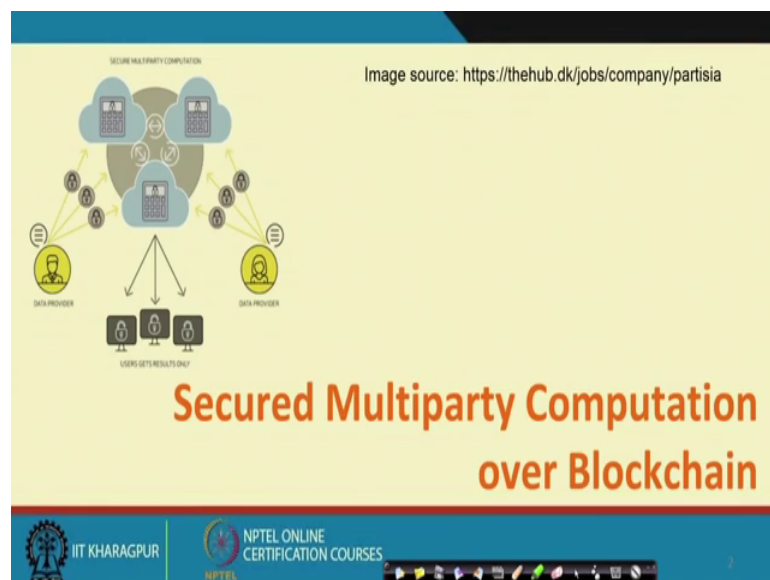


**Blockchains Architecture, Design and Use Cases**  
**Prof. Sandip Chakraborty**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology Kharagpur**

**Lecture - 47**  
**Secured Multiparty Computation over Blockchain**

Welcome back to the course on Blockchain. So today we will discuss the one interesting topic of utilizing one security protocol or to better explain that implement one impossibility result of a security protocol with the help of this Blockchain technology.

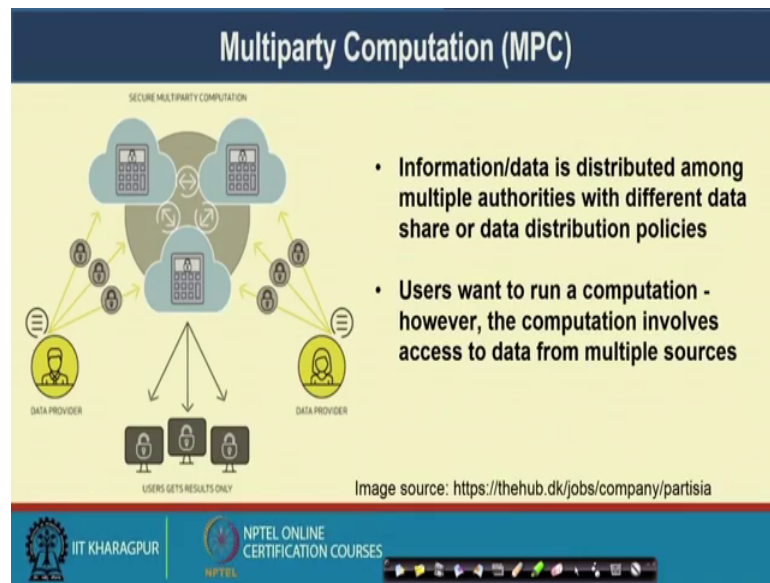
(Refer Slide Time: 00:38)



So, we will try to implement a fair secured multiparty computation platform with the help of a blockchain technology. So, this is a vast topic there are multiple intrinsic details in it, so I will not touch upon the theoretical proof or the very intrinsic details of this particular topic. I will just give you an brief overview of this entire technology. And we will give you an idea that what is the power of blockchain where you can utilize this blockchain technology even to run or to design an protocol for security which is otherwise impossible to execute.

So, let us look into this particular example of multiparty computation. So, what is a multiparty computation?

(Refer Slide Time: 01:28)



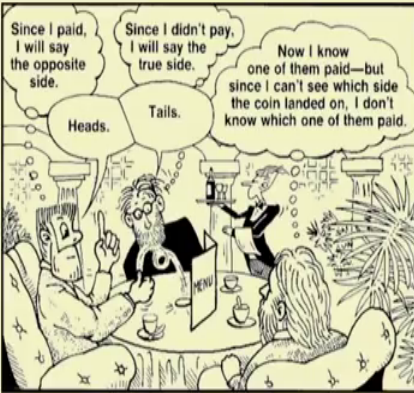
So, in case of a multiparty computation your information or data it is distributed among multiple authorities with different data centres or data distribution policies. They may have different stakeholders and users want to run a computation on top of that platform. However, the computation involves access to data from multiple sources.



Now the idea here is that you do not want to reveal your private result, private information you have certain data which you believe that it is your private data. So, you do not want to reveal that data, but at the same time you want to get the result of certain computation.

(Refer Slide Time: 02:17)

### Dining Cryptographer Problem

- Three cryptographers are sitting down to dinner at their favorite restaurant
- Any of the cryptographer can pay the bill, or the bill can be directly paid by the National Security Council (NSC)



 IIT KHARAGPUR |  NPTEL ONLINE CERTIFICATION COURSES

So, let us look into one example which is popularly known as Dining Cryptographer Problem. So, assume that there are 3 cryptographers who are sitting down for a dinner at their favourite restaurant. And in that environment any of the cryptographers can pay the bill after the dinner is over or the bill can be paid directly by the National Security Council, so where they are actually employed. So either it is paid by their employer or any of them can pay the bill.

But interestingly the 3 cryptographer they respect each other right to make an anonymous payment. So, they do not want to reveal who has actually paid. So assume that there are 3 parties I have paid it. So, I know it but others 2 should not know that I have paid it. So, it is just like that if I have paid the bill; I know obviously that I have paid, but if I have not paid the bill I have no way to understand whether any one of the other 2 has paid or whether my employer the NSA has paid the bill or not. So, this payment protocol this kind of payment protocol can be designed using a secured multiparty computation platform.

So, here is the idea that we believe that this payment is my private data, I do not want to share it, but at the end of the day one of us has been paid and a bill got paid finally and I got the information that the bill got paid; so you do not need to paid it any further.

(Refer Slide Time: 03:56)

### Formal Definition

- There are  $n$  players  $p_1, p_2, \dots, p_n$
- They wish to evaluate a function  $f(x_1, x_2, \dots, x_n)$
- $x_i$  is a secret value provided by  $p_i$
- **Goal:**
  - Preserve the privacy of the player's input
  - Guarantee the correctness of the computation

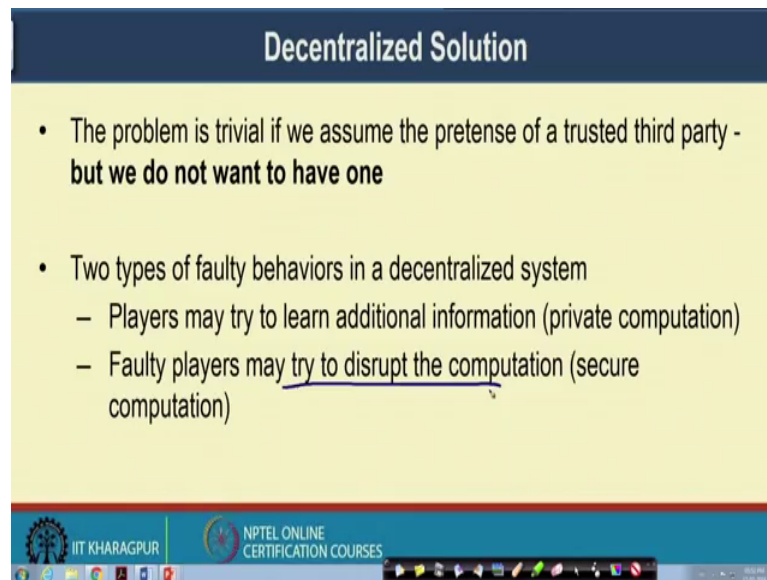
The diagram illustrates the formal definition of a multiparty computation. It shows two systems: an 'untrusted system' and a 'trusted subsystem'. In the untrusted system, three parties (A, B, and C) are connected in a triangle. In the trusted subsystem, three secret modules (sec. mod. A, sec. mod. B, and sec. mod. C) are connected in a triangle. The untrusted system is connected to the trusted subsystem via lines representing communication channels.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, this is the typical dining cryptographer problems which explain the platform of multiparty computation. So, here is the formal definition of a multiparty computation that there are  $n$  number of players  $p_1$  to  $p_n$  they wish to evaluate a function  $f$  of  $x_1$  to  $x_n$  where  $x_i$  is a secret value provided by  $p_i$ . So, a free user has a secret value and they do not want to reveal that secret value, but at the same time they want to make a computation with the help of those secret values. So, the goal is to preserve the privacy of the players input and guarantee the correctness of the computation.

So, we want to preserve the privacy of the user input and at the same time we want to guarantee the correctness of the computation platform ok.

(Refer Slide Time: 04:48)



The slide is titled "Decentralized Solution" in a dark blue header. The main content is on a light yellow background and consists of three bullet points. The first bullet point states that the problem is trivial if we assume the presence of a trusted third party, but we do not want to have one. The second bullet point lists two types of faulty behaviors in a decentralized system: players may try to learn additional information (private computation), and faulty players may try to disrupt the computation (secure computation). The slide footer includes the IIT Kharagpur logo and the text "NPTEL ONLINE CERTIFICATION COURSES".

- The problem is trivial if we assume the presence of a trusted third party - **but we do not want to have one**
- Two types of faulty behaviors in a decentralized system
  - Players may try to learn additional information (private computation)
  - Faulty players may try to disrupt the computation (secure computation)


So, the problem is trivial if we assume the presence of a trusted third party. In that case you upload your private data to the trusted third party the trusted third party will do the computation and return back the result to you. But secured multiparty computation assumes that there is no such trusted third party or we do not trust any of the third parties in the world. So, that way whenever the computation will be done the computation will be done in a total distributed environment. So, it is just like that I will throw certain garbled information, which actually embedded the actual information which is there, inside this apparently random or garbled information and the computation will return me the correct result which I am intended to.

So, in this kind of decentralized architecture we can have 2 types of faulty behaviours the first one is the player may try to learn additional information, so we call them as the semi honest player. So, although they are participating in the process, but they try to learn the additional information, so the private computation requirement is getting violated; and the second this more problematic case that the adversaries or the faulty players may try to disturb the computation, so you want to have a secure computation. So, the faulty players because you are not revealing your actual information, it may happen that the faulty players will disturb the computation in a architecture or in a MPC based platform. So, if majority of the users are dishonest you will never be able to get the correct result of the computation, so that is the impossibility theorem that we are going to state.

(Refer Slide Time: 06:38)

### Yao's Millionaire Problem

- Two millionaires wish to find out who is wealthier
- They do not want to reveal any other information




IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, let us look into one particular use case and a solution of multiparty computation problem which is called Yao's millionaire problem. There are multiple protocols for multiparty computation I am just presenting one protocol for your understanding about how multiparty computation is being done. So, this Yao's millionaire problem, so it is like that 2 millionaires they wished to find out who is wealthier, but they do not want to reveal any other information. So, they do not want to reveal what is their share of the money, but they want to find out that who is the richest among these 2 millionaires.

(Refer Slide Time: 07:17)

### Preconditions

- We know the range of the inputs:  $(0, N)$
- A: Public key  $e$ , Private key  $d$
- B: Can access  $e$ , not  $d$
- $D_d(E_e(X)) = X$
- $D_d(E_e(X) + Y) = \text{some random looking thing if you do not know } d$

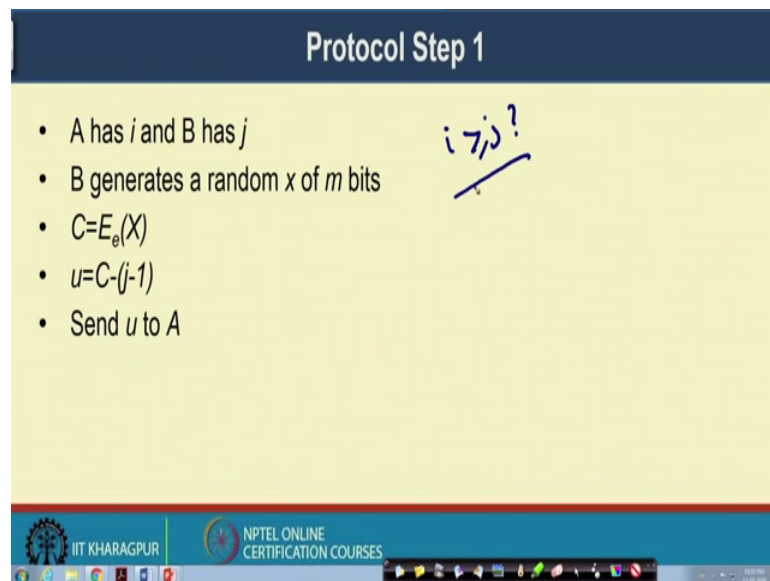


IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So the protocol works in this way, so we assume that the range of the input is known so their money is something in between 0 to N and the 2 millionaires are A and B. So, A has a public key  $e$  which is used to encrypt and a private key  $d$  which is used to decrypt and we can access  $e$  but not  $d$  similar to the public key cryptography concept that we have discussed earlier.

So, now if you have a share secret  $X$  if you encrypt it with  $E$  and then decrypt it with  $D$  you will get  $X$  that is well known from this public key cryptography concept. But then if you have encrypted  $X$  with the encryption key  $E$  and add it up some value  $Y$  which is known to you and then decrypt this entire thing with the decryption key  $D$  then you will get some random looking thing if you do not know  $D$ . So, if you know  $d$  in that case if you know the private key  $D$ . Then you will be able to correctly find out what is happening there, but if you do not know that then it will look like a random thing to you ok.

(Refer Slide Time: 08:37)



The slide is titled "Protocol Step 1" and contains the following text:

- A has  $i$  and B has  $j$
- B generates a random  $x$  of  $m$  bits
- $C = E_e(X)$
- $u = C - (j - 1)$
- Send  $u$  to A

Handwritten in blue ink next to the first bullet point is the expression  $i > j ?$ .

The slide footer includes the IIT KHARAGPUR logo and the text "NPTEL ONLINE CERTIFICATION COURSES".

Now the first step of the protocol is something like this so say A has  $i$  and B has  $j$ ; so B generates a random  $x$  of  $m$  bits so and our objective here is to find out whether  $i$  is greater than equal to  $j$  or not or the vice versa. Whether  $j$  is greater than equal to  $i$  or not that you want to find out now you get  $C$  by encrypting this value of  $X$  from this  $C$  you subtract  $j$  minus 1 so that is done by B.

(Refer Slide Time: 09:10)

### Protocol Step 1

- A has  $i$  and B has  $j$
- B generates a random  $x$  of  $m$  bits
- $C = E_e(X)$
- $u = C - (j-1)$
- Send  $u$  to A

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

And B finally gets value  $u$  and B sends this value  $u$  to A ok.

(Refer Slide Time: 09:22)

### Protocol Step 2

- A Computes: for  $(t = 1 \text{ to } N)$   $y_m = D_d(u+t)$
- A takes a prime  $p$  of size  $\sqrt{m}$  and computes
  - $z_i = y_i \bmod p$  for  $i = 1 \text{ to } N$
- $p$  is chosen such that  $|z_m - z_n| \geq 2$  for any  $m, n$  in  $[1 \text{ to } N]$

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Then a computes the following for  $t$  equal to  $0$  to  $N$ , A had got this value  $u$ . So, he adds up this  $t$  with  $u$  decrypt it and find out the value  $Y_m$ .




(Refer Slide Time: 09:42)


### Protocol Step 1

- A has  $i$  and B has  $j$
- B generates a random  $x$  of  $m$  bits
- $C = E_a(X)$
- $u = C - (j-1)$
- Send  $u$  to A


$C = u + (j-1)$



IIT KHARAGPUR



NPTEL ONLINE CERTIFICATION COURSES




So, here if you just look into that in this case whenever you are making this computation it is like this that  $C$  is equal to  $u$  plus some value  $j$  minus 1 well. So,  $b$  has this value  $j$  the  $j$  minus 1 it is added with the value  $u$  that you got and  $C$  is generated from this encrypted data of  $X$  which is working like a random string. So,  $a$  takes a prime  $p$  of size square root of  $m$  and computes this  $z_i$ ,  $z_i$  equal to  $y_i \bmod p$  for  $i$  equal to 1 to  $n$  and we choose  $p$  such that  $z_m$  minus  $z_n$  is greater than equal to 2 for any input  $m$   $n$  in 1 to  $n$ .

(Refer Slide Time: 10:40)


### Protocol Step 3

- A sends B the following list
  - $p, z_1, z_2, \dots, z_i, (z_{i+1}+1), (z_{i+2}+1), \dots, (z_N+1)$
- B compares the  $j^{\text{th}}$  entry of this list excluding prime  $p$  with  $(x \bmod p)$
- If  $(x \bmod p) = j^{\text{th}}$  entry of the list, then  $i \geq j$


$z_j$



IIT KHARAGPUR



NPTEL ONLINE CERTIFICATION COURSES

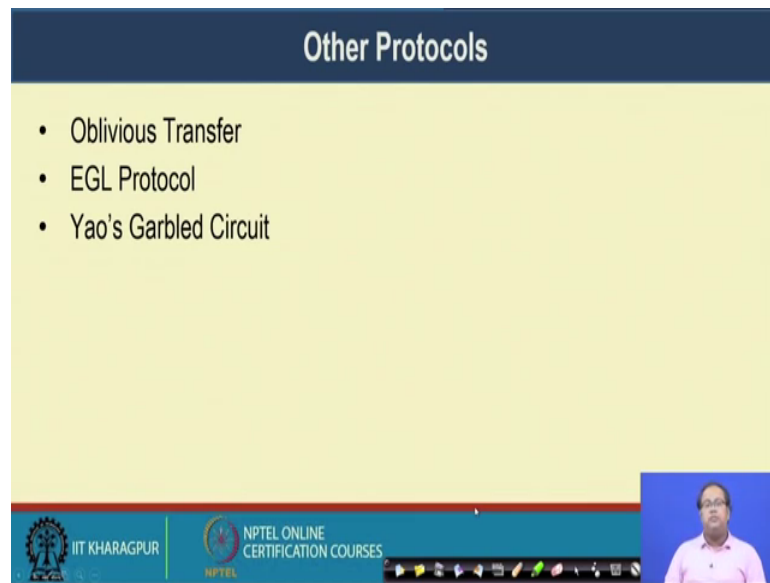


So, from this computation a gets a sequence of numbers so A sends to B this value  $p$  the value  $z_1$  to  $z_n$  plus, so on the values that it has computed. Now, B compares the  $j$ -th entry of this list including prime  $p$ . So, it finds out this value  $z_j$  and compare it with  $x \bmod p$ , so if this  $x \bmod p$  is equal to the  $j$ -th entry of the list then we can say  $i$  equal to  $i$  greater than equal to  $j$ . So, this is the protocol so I have to explain this protocol not to give you the correctness about how this works, but what I wanted to convey the message that here apparently we are sending some kind of random data we are extracting or we are sharing some kind of random data between the users. So, in the first step so in the first step a was generating some randomly looking data  $u$  which is sent to A.

Now, A interestingly a do not have this decryption  $d$  because A do not have that decryption value  $d$  a will not be able to find out what is the corresponding value which has been generated by A and what is this value of  $j$ . So, that is impossible for A to find it out, now whenever a is doing the computation a is actually utilizing this value  $i$  while constructing the series this entire series and in that entire series it is embedding it is own number with the help of this  $y_i \bmod p$  with this module operator with the value of the prime  $p$ .

Now the computation is done between these 2 random numbers and mathematically we can show that if this  $X \bmod p$  is equal to the  $j$  the entry of the list, in that case  $i$  will be always greater than equal to  $j$ . So, that way you can get your target objective, but at the same time you are not revealing what is your actual wealth what is the information that you have with you. So the private information remains private in this entire protocol.

(Refer Slide Time: 13:10)



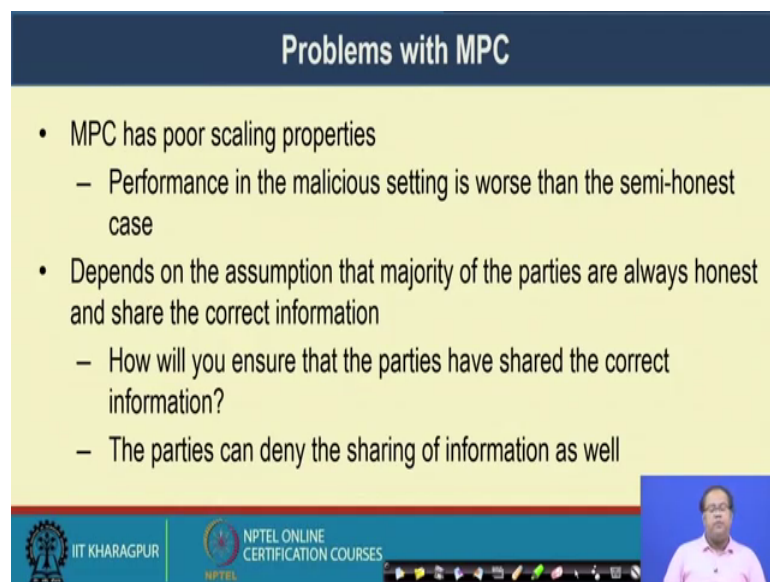
The slide is titled "Other Protocols" and lists three items:

- Oblivious Transfer
- EGL Protocol
- Yao's Garbled Circuit

The slide footer includes the IIT KHARAGPUR logo, the NPTEL ONLINE CERTIFICATION COURSES logo, and a small video inset of the presenter.

So, there are other protocols like Yao's protocol. So there is this oblivious transfer based protocol, EGL protocol, or Yao's garbled circuit based protocol which is used for these secured multiparty computation.

(Refer Slide Time: 13:26)



The slide is titled "Problems with MPC" and lists several issues:

- MPC has poor scaling properties
  - Performance in the malicious setting is worse than the semi-honest case
- Depends on the assumption that majority of the parties are always honest and share the correct information
  - How will you ensure that the parties have shared the correct information?
  - The parties can deny the sharing of information as well

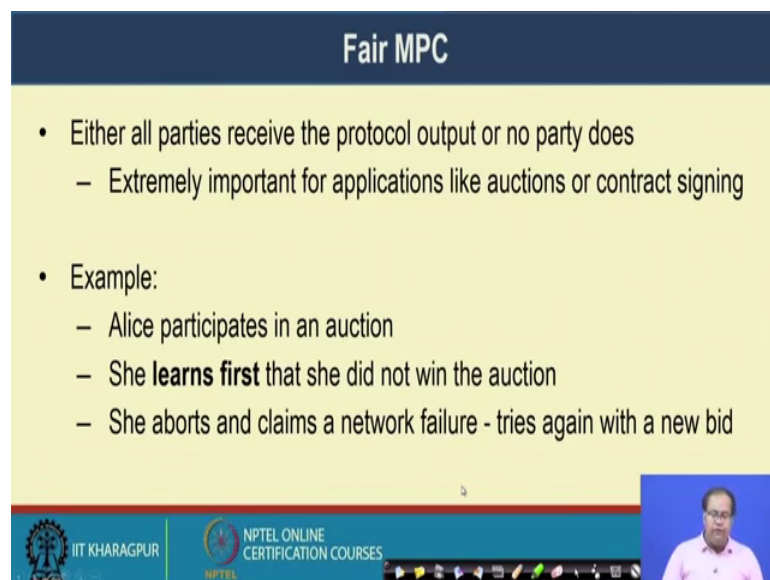
The slide footer includes the IIT KHARAGPUR logo, the NPTEL ONLINE CERTIFICATION COURSES logo, and a small video inset of the presenter.

Now there are multiple problems which are associated with MPC, the first problem is that MPC has a poor scaling property. So, the performance in the malicious setting is worse than the semi-honest case, semi-honest case means when the parties individually participate in the computation. But they are trying to reveal

the additional information out of it and malicious setting means, so whenever the parties they are trying to twig the result so they are trying to give some false input, so that the correct output of the computation cannot be revealed.

Now it depends on the assumption that majority of the parties are always honest and share the correct information. Now, the question comes that how will you ensure that the parties have share the correct information. Now, the thing is that the parties can deny the sharing of information as well you have to prevent that in case of this multiparty computation platform.

(Refer Slide Time: 14:28)



The slide is titled "Fair MPC" and contains the following content:

- Either all parties receive the protocol output or no party does
  - Extremely important for applications like auctions or contract signing
- Example:
  - Alice participates in an auction
  - She **learns first** that she did not win the auction
  - She aborts and claims a network failure - tries again with a new bid

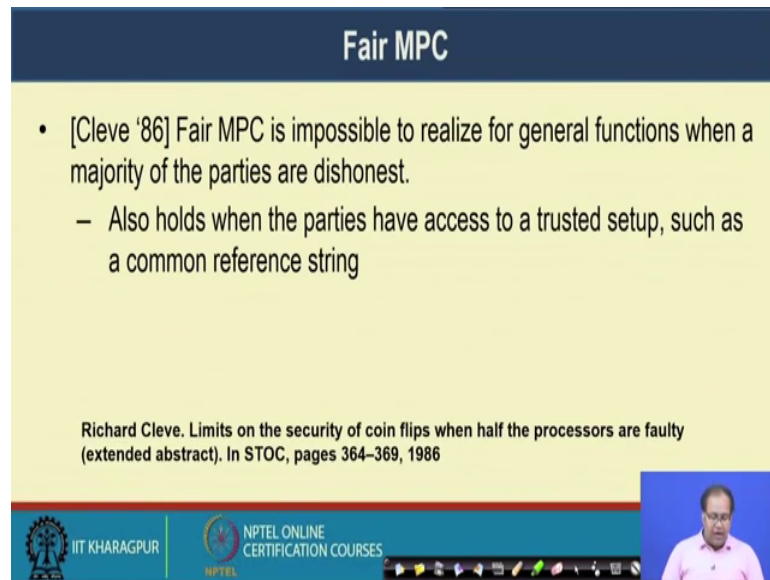
At the bottom of the slide, there are logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES, along with a small video inset of a speaker.

So, we have this kind of fair MPC protocol, so the fair MPC protocol says that either all parties receive the protocol output or no party does. So, either all will receive the output from the protocol or no one will receive any output, this is extremely important for the application like auction or contracts sign. Say assume an example that Alice participate in an auction and this auction is done with the help of an multiparty computation architecture.

Now, this MPC is not fair this MPC is not fair that means that any party it can get the result first. If it happens that Alice learned a result first that she had not win the auction what she can do that she aborts and she can claim a network failure, if she claim a network failure then this entire auction will get invalidated and the entire protocol will run again, when the entire protocol will run then Alice can try again with a new bid. So,

with this kind of particular application it is very important that all the parties receive the protocol output simultaneously either that happens or no party receives anything from this computation platform.

(Refer Slide Time: 15:49)



The slide is titled "Fair MPC" in a dark blue header. The main content is on a light yellow background and includes a bulleted list and a citation. At the bottom, there are logos for IIT Kharagpur and NPTEL, along with a small video inset of a speaker.

- [Cleve '86] Fair MPC is impossible to realize for general functions when a majority of the parties are dishonest.
  - Also holds when the parties have access to a trusted setup, such as a common reference string

Richard Cleve. Limits on the security of coin flips when half the processors are faulty (extended abstract). In STOC, pages 364–369, 1986


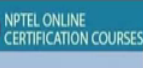

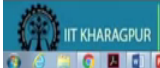
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Now, there is an impossibility result which was shown by Cleve Richard Cleve in 1986, it says that the fair multiparty computation is impossible to realize for general functions when a majority of the parties are dishonest. Now if majority of the parties are dishonest then you will never be able to have a fair MPC multiparty computation platform, this particular impossibility result also holds when the parties have access to a trusted setup. Such as a common reference string during that time also you will not be able to find out a solution of a particular function in the fair MPC protocol.

(Refer Slide Time: 16:35)

### Solve Fair MPC - Use a Public Bulletin Board

- Parties have access to a public ledger
  - Allows anyone to publish arbitrary strings - used for MPC protocol
  - The strings contain proof about who has published the string - anyone can verify
- Run an unfair MPC protocol to compute an encryption of the function output - design a fair decryption protocol using the public ledger - either everyone can decrypt or no one can



Now to solve the fair MPC we can utilize a Blockchain based architecture. So these particular blockchain based architecture you can have a access to a public ledger, so I will just give you a brief overview about how this system works. So, the parties have access to a public ledger, this public ledger allows anyone to publish arbitrary string; this arbitrary string are used in the MPC protocol. So, this arbitrary strings although it seems arbitrary, but as I have shown in to the Yao's millionaire problem they are generated from the result of certain computation; so they are the output from certain computation from the normal one it will look something random or arbitrary. But the person who has generated that if he or she has the decryption key he or she will be able to correctly decrypt that particular information.

Now, these string contain the proof about who has published the string, so you know that who has published the string. So, anyone can verify that with that particular proof and they can verify with the proof that the string which has actually been posted in the public bulletin board or the public ledger which can be implemented with the help of a blockchain, that is indeed valid string that has been posted by an individual party. Now that has multiple advantage that has the advantage, like in the future if you find out that. Well, the string is not been posted following the normal MPC protocol you can certainly identify that well the person who has posted the string he or she is working like a adversary.

So, identifying the adversary is easier if you have access to this kind of logs within the system. So, we are using the blockchain property here that people will be able to insert

information to that log, but they will not be able to deny later that they have inserted they have not inserted the information to the log or they will not be able to tamper with the information; once that has been put into the log which has been maintained by a blockchain platform. So, the idea here is that you would run an unfair MPC protocol to compute an encryption of the function output; so rather than directly producing the function output you make an encryption of the function output as public.

Now, you design a fair decryption protocol so this fair decryption protocol will be designed using the public ledger using the help of the public ledger. So, in the public ledger whenever you are posting something and whenever all the parties will post that arbitrary string that arbitrary string will be used to find out or extract the information which will be helpful to generate the decryption key. So, once you have the decryption key every party has an result in the encrypted format under the description key they will be able to decrypt it and find out the result of the MPC protocol. So, that is the entire idea that either everyone can decrypt or no one can.

So, if everyone provides the correct string in the public bulletin board, you can utilize that correct string to generate the decryption key once you have that decryption key in hand everyone can have that decryption key, because it is written in the blockchain you will not be able to erase it. So, everyone will get the decryption key at the same once you are getting the decryption key you can use the description key to decrypt the result which has been encrypted and you can get the final output ok.

(Refer Slide Time: 20:26)

The slide is titled "Witness Encryption" and contains the following text:

- The parties first run a standard MPC protocol to compute a randomized function that takes the private inputs  $(y_1, y_2)$  of the parties and returns a witness encryption cyphertext
- To access the cyphertext, the parties need to post a "release token"  $\alpha$  on the public ledger
  - Obtain the corresponding proof of positing  $\alpha$  - the witness
- The witness is used to decrypt the cyphertext and obtain the result of the MPC

Handwritten annotations on the slide include:  $E_c(R)$ ,  $d$ , and "Enigma".

The slide footer includes the IIT KHARAGPUR logo and the text "NPTEL ONLINE CERTIFICATION COURSES". A small video inset shows a man speaking.

So, for that this particular work which called Enigma, so Enigma is a architecture that uses. So, the title of the paper is Enigma fairness in an unfair work for secure multiparty computation that was been published in CCSA CMCCS conference last year. So, it uses this concept of witness encryption that I have just mentioned, so this witness encryption says that the parties they first run a standard MPC protocol to compute a randomized function that takes the private inputs say  $y_1$  and  $y_2$  of the parties and returns a witness encryption cyphertext. So, this witness encryption cyphertext means you have the result of the computation that result is encrypted.

Now, to get a result you require decryption key  $d$ . Now to generate the description key to access the cyphertext the parties need to post a release token  $\alpha$  on the public ledger. So, once you post this release token  $\alpha$  on the public ledger, so this release token  $\alpha$  is generated following certain cryptographic algorithm. So, you can look into the Enigma paper I am not going to that particular details; so the parties need to post that release token  $\alpha$  on the public ledger and you can obtain the corresponding proof of posting  $\alpha$  which we call as the witness. So, once you have that proof the witness can be used to decrypt the cyphertext and obtain the result of the MPC.

So, that way by making by applying this encryption decryption protocol and making the description fair. With the help of this witness concept you can make the MPC architecture fair, so that we can implement a fair MPC protocol with the help of a public



bulletin board which can be implemented with the help of a blockchain. So, if you are interested further I will suggest all of you to look into the enigma paper that have been published in the last year CCS in details. So, that gives all the cryptographic details, so I have not touched the internals of that cryptographic protocol because, it will require a lots of basic understanding of cryptographic different cryptographic algorithm like this concept of yours garbled circuit and so on.

So, that that is why I have avoided the theoretical details of that, just give a new certain pointer about the idea if you are interested in this particular topic you can explore it further by looking into the corresponding enigma paper.

(Refer Slide Time: 23:20)



So, that is all about this lecture, so if you are interested just look into this paper the paper was published by A.R Choudhuri M Green A Jain G Kaptchuk I Miers in 2017 SIGSAC the last year, SIGSAC the title of the paper is fairness in an unfair world fair multiparty computation from public bulletin board. So, it utilizes this blockchain architecture with this basic of witness encryption concept that I have just mention with the help of a public ledger or a public bulletin board over they mention to implement this entire protocol. So, that gives you a brief idea about the entire architecture of developing a fair multiparty computation protocol with the help if a help of a blockchain architecture, which is otherwise impossible based on the impossibility that I have mentioned.

So, thank you all for attending this class. Hope you will be able to explore further following this topic.

Thank you.