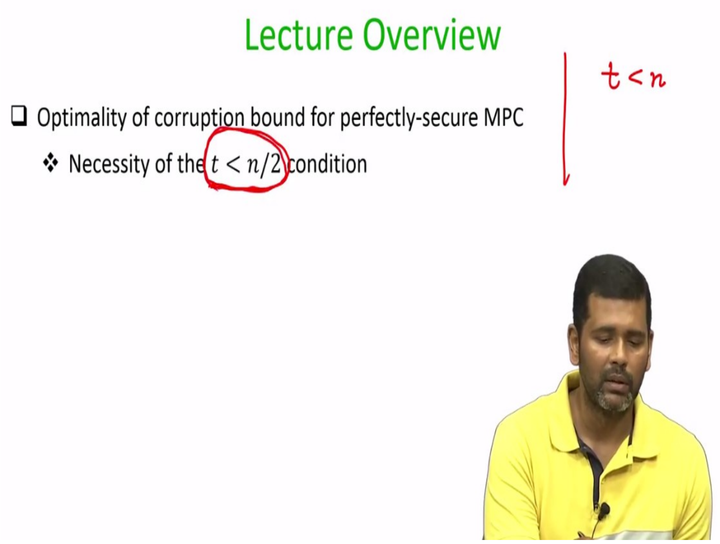
Secure Computation: Part 1
Prof. Ashish Choudhury
Department of Computer Science
Indian Institute of Science – Bengaluru

Lecture – 28
Optimality of Corruption Bound for Perfectly-Secure MPC

Hello everyone. Welcome to this lecture.

(Refer Slide Time: 00:31)



Lecture Overview

❑ Optimality of corruption bound for perfectly-secure MPC

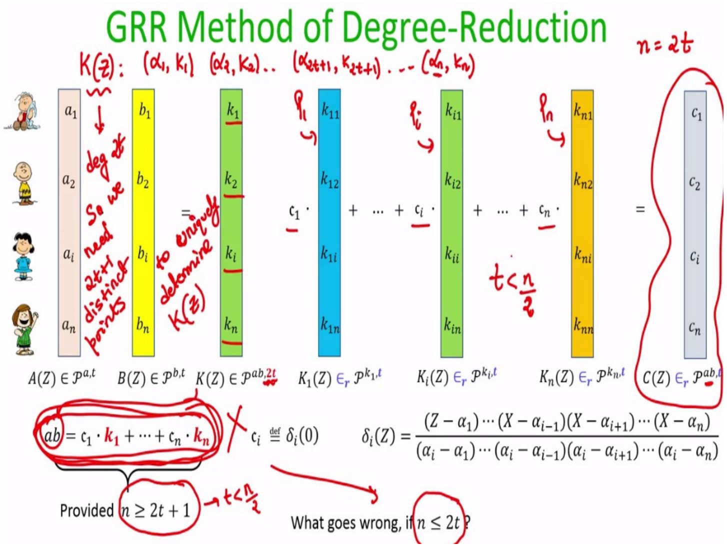
- ❖ Necessity of the $t < n/2$ condition

$t < n$

So, in this lecture we will see that the condition t less than n / 2 is a necessary condition for designing any generic MPC protocol namely we will show that there exist some functions which we cannot securely compute if the number of corruptions in the system is not upper bounded by n / 2. Remember, we have shown that if we have a linear function then the linear function can be securely computed even if up to n – 1 corruptions are there in the system.

What we are going to show here is that there exist some functions there are some functions for which we require this much condition, this condition to be necessary.

(Refer Slide Time: 01:16)



GRR Method of Degree-Reduction

$$= c_1 \cdot [\ldots] + \cdots + c_i \cdot [\ldots] + \cdots + c_n \cdot [\ldots] =$$

$A(Z) \in \mathcal{P}^{a,t}$ $B(Z) \in \mathcal{P}^{b,t}$ $K(Z) \in \mathcal{P}^{ab,2t}$ $K_1(Z) \in_r \mathcal{P}^{k_1,t}$ $K_i(Z) \in_r \mathcal{P}^{k_i,t}$ $K_n(Z) \in_r \mathcal{P}^{k_n,t}$ $C(Z) \in_r \mathcal{P}^{ab,t}$

$$\underbrace{ab = c_1 \cdot k_1 + \cdots + c_n \cdot k_n}_{\text{Provided } n \geq 2t+1} \quad c_i \overset{\text{def}}{=} \delta_i(0) \qquad \delta_i(Z) = \frac{(Z-\alpha_1)\cdots(X-\alpha_{i-1})(X-\alpha_{i+1})\cdots(X-\alpha_n)}{(\alpha_i-\alpha_1)\cdots(\alpha_i-\alpha_{i-1})(\alpha_i-\alpha_{i+1})\cdots(\alpha_i-\alpha_n)}$$

Provided $n \geq 2t + 1$ → $t < \frac{n}{2}$

What goes wrong, if $n \leq 2t$?

So, before going into the proof let us first recall the GRR degree reduction method and try to understand that where exactly it will fail if we do not make the assumption that t is less than n over 2. So, the inputs for the GRR degree reduction method are two values a and b which are randomly t shared. The parties compute locally the product of a shares and b shares that will collectively generate a vector of shares lying on a 2t degree polynomial.

And then we argue that the constant term of this K polynomial which is ab can be expressed as a linear combination of this k 1 value, k 2 value, k i value, k n value provided n is greater than equal to 2t + 1 because the degree of this K polynomial is 2t and is K polynomial is uniquely determined only if n is greater than equal to 2t + 1 because remember this K polynomial is a polynomial passing through alpha 1, k 1, alpha 2 k 2.

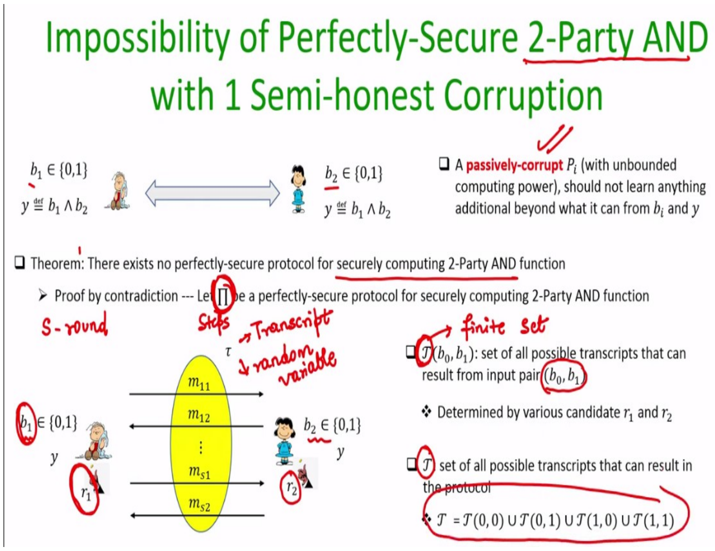
And alpha 2t + 1 k 2t + 1 and like that alpha n k n. So, only when n is greater than or equal to 2t + 1 whatever we are writing here make sense and then assuming that n is greater than equal to 2t + 1 or t less than n over 2 so this condition also means t less than n over 2. Assuming that condition is ensured then the parties can secret shares their respective k values so we can ask P 1 to do this.

We can ask P i to do this, we can ask P n to do this. Do this means secret share individually their k shares and then we can apply the public Lagrange linear combiners and obtain the random vector of c shares lying on a t degree polynomial. Now what goes wrong if we do not ensure that n is greater than equal to 2t + 1. If n is not greater than equal to 2t + 1 then whatever we have written down here that is not correct.

Because this K polynomial it has degree 2t. So, we need 2t + 1 distinct points to uniquely determine this K polynomial and hence we require 2t + 1 distinct points to uniquely express ab a times b as a linear combination of those 2t + 1 distinct values. If n would have been say just equal to 2t there are only 2t number of parties in the system then this linear combination it would not work because I cannot write ab as a unique linear combination of those 2t number of k values.

And hence we cannot say that these vector of c shares lie on a random t degree polynomial whose constant term is ab whose constant term need not be ab. So, that means definitely for the GRR degree reduction method we require t to be strictly less than n over 2, but that does not mean that every other degree reduction method also requires t less than n over 2. There might be a clever way of doing a degree reduction or first of all there might be a completely different approach of securely evaluating multiplication gates which need not be based on shared circuit evaluation. Some other approach which may not require this t less than n over 2 bound.

(Refer Slide Time: 05:38)



Impossibility of Perfectly-Secure 2-Party AND with 1 Semi-honest Corruption

$b_1 \in \{0,1\}$
$y \overset{\text{def}}{=} b_1 \wedge b_2$
$b_2 \in \{0,1\}$
$y \overset{\text{def}}{=} b_1 \wedge b_2$

□ A **passively-corrupt** P_i (with unbounded computing power), should not learn anything additional beyond what it can from b_i and y

□ Theorem: There exists no perfectly-secure protocol for securely computing 2-Party AND function

- Proof by contradiction --- Let $\prod$ be a perfectly-secure protocol for securely computing 2-Party AND function

□ $\mathcal{T}(b_0, b_1)$: set of all possible transcripts that can result from input pair (b_0, b_1)

❖ Determined by various candidate r_1 and r_2

□ $\mathcal{T}$ set of all possible transcripts that can result in the protocol

❖ $\mathcal{T} = \mathcal{T}(0,0) \cup \mathcal{T}(0,1) \cup \mathcal{T}(1,0) \cup \mathcal{T}(1,1)$

What we are now going to see is that is not the case. There exist some functions which we cannot securely compute if there are more than half of the participants who are corrupt. So, what we will do here is the following. We will show that there is no perfectly secured 2 party protocol for computing and of the bits of the two parties. If one of them is passively corrupt.

Namely the setting is the following. We have two parties here. Party 1 has the input bit b 1 second party has input bit b 2 we require a method which allows the parties to finally learn and of their respective bits, but in this process and one of these two parties could be corrupt by a computationally unbounded adversary and we want that the protocol should ensure that the view of the corrupt parties should be independent of the other parties input.

Namely through the function output and the input of the corrupt party nothing additional should be revealed about other parties input. So, what I mean to say here is that if b 1 would have been 1 if b 1 = 1 and y = 0 then of course other parties input is learnt that is not the privacy breach definitely that leaks b 2 = 0 that is fine that acceptable that is anyhow infer from the if P 1 would have been corrupt then anyhow just based on its input and function output it could easily conclude what is the other parties input that is not privacy breach.

What we want here is if b 1 = 0 then definitely y = 0 then we want that if P 1 is corrupt then whatever is view 1 that is generated in the protocol that view 1 should be independent of whether b 2 = 0 or whether b 2 = 1 that is what the protocol should ensure. So, we are arguing here we are going to show here is that there is no such protocol possible which gives you these guarantees.

That means the theorem statement that we are going to prove here is that there exist no perfectly secured protocol for securely computing the two party and function and the proof will be by contradiction. We will assume on contrary there is some abstract protocol pi according to which the parties can exchange messages and finally they obtain the AND of their respective bits and it satisfies these condition.

That means you can imagine that abstract protocol pi by the way pi need not be just secret sharing based protocol it could be any abstract protocol. We are trying to rule out the possibility of any kind of perfectly secured protocol. So, it could be based on secret sharing, it could be based on any cryptographic primitive we are going to abstract out that protocol. The way we are going to abstract out the protocol is as follows.

So, we will assume that the protocol pi will have the following form and this form is a very generic form it captures any kind of two parties secure protocol that you can think of. The protocol will take the respective inputs of the parties and of course the protocol has to be randomized namely the messages which parties are going to exchange they have to be random messages.

They cannot be fixed they cannot be just based on the inputs of the parties because otherwise that clearly is going to breach the privacy of the inputs of the honest party. So, there will be some internal randomness, random strengths which are produced by the parties generated by the parties during the execution of the protocol and based on the inputs and respective randomness the party start computing messages as determined by the protocol pi.

So, they are not going to send arbitrary messages they are the messages which they are now going to communicate it will depend on the value of the input, the internal randomness and the steps of the protocol pi that means if the value of b 1 is fixed, but the randomness changes then based on the steps the value of the message m 11 will be different and the value of m 12 which P 2 will send back will be different and so on.

So, assume that the protocol is an S round protocol of course the number of steps or the number of times the parties have to interact that will be publically known that is not an unknown quantity that is publically known because you are assuming that the steps of the pi are publically known and then finally based on whatever messages the parties have exchanged, the respective parties obtain the function output which is the AND of their individual bits.

Now this highlighted thing namely the messages which parties have exchanged I call it as the protocol transcript namely whatever values, messages P 1, P 2 have communicated that full thing that is now publically known because whatever P 1 is sending to P 2 that is known to P 1 as well as to P 2 and whatever P 2 is communicating back to P 1 that known both to P 1 and P 2.

So, this transcript is publically known and this transcript is a random variable remember. Why it is a random variable because even if the same two parties execute the protocol pi with the same value of b 1 and b 2 depending upon what exactly are the random coins which parties are using during the run time of the protocol that will determine the messages m 11, m 12, m s 1 and m s 2 and that is why this messages are not going to take fixed values.

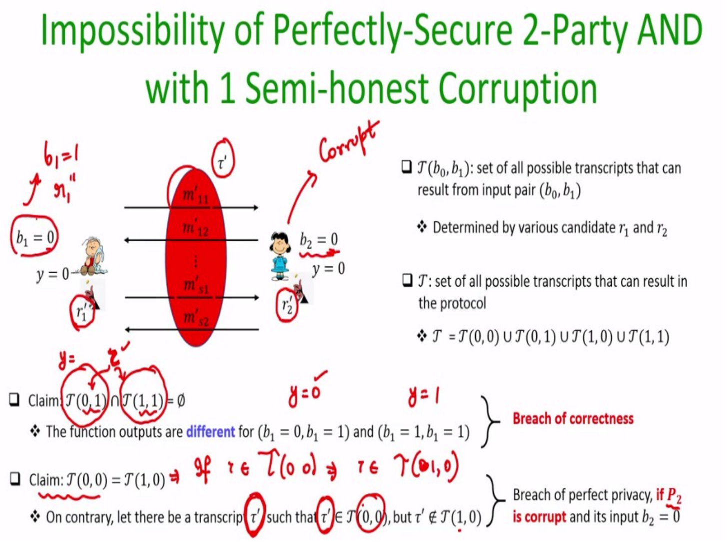
They are going to take different values with different probability which depends upon the value of the internal randomness. So, that is why this transcript tau is a random variable. So, now let us introduce some notations here. Let this fancy (()) (12:15) and input b 0, b 1 denotes the set of all possible transcripts which the protocol pi can generate if the two parties would have participated in the protocol with inputs b 0 b 1 and this is a finite set remember.

Why finite set? Because as per the steps of the protocol there are only finitely many random coins which P 1 and P 2 could generate and if we now execute all possible executions of the protocol pi by fixing b 1 and b 2 and with different candidate r 1, r 2 we can generate this entire set of transcript. This might be enormously large it could be exponentially large that is fine exponentially large means it might take exponential amount of computation to generate that is fine, but that is finite that is not an infinite set.

And an adversary who is computationally unbounded it can always generate this entire set of transcripts. Why what it has to do? It just has to run the steps of pi in its mind assuming that the inputs of the parties are b 0 b 1 and just by changing randomness component it can reproduce all the transcripts and then list it out and that will be this set fancy t (()) (13:44) and now let fancy t with a set of all possible transcripts that can be generated for all possible input combinations of the parties.

So, of course it is a bigger set it is a universal set of transcripts which the protocol pi could generate.

(Refer Slide Time: 14:06)



Impossibility of Perfectly-Secure 2-Party AND with 1 Semi-honest Corruption

❑ $\mathcal{T}(b_0, b_1)$: set of all possible transcripts that can result from input pair (b_0, b_1)

❖ Determined by various candidate r_1 and r_2

❑ $\mathcal{T}$: set of all possible transcripts that can result in the protocol

❖ $\mathcal{T} = \mathcal{T}(0,0) \cup \mathcal{T}(0,1) \cup \mathcal{T}(1,0) \cup \mathcal{T}(1,1)$

❑ Claim: $\mathcal{T}(0,1) \cap \mathcal{T}(1,1) = \emptyset$ $y = 0$ $y = 1$

❖ The function outputs are different for $(b_1 = 0, b_1 = 1)$ and $(b_1 = 1, b_1 = 1)$

Breach of correctness

❑ Claim: $\mathcal{T}(0,0) = \mathcal{T}(1,0)$ $\Rightarrow$ If $\tau \in \mathcal{T}(0\ 0) \Rightarrow \tau \in \mathcal{T}(1,0)$

❖ On contrary, let there be a transcript τ' such that $\tau' \in \mathcal{T}(0,0)$, but $\tau' \notin \mathcal{T}(1,0)$

Breach of perfect privacy, if P_2 is corrupt and its input $b_2 = 0$

So, now we are going to argue some properties regarding the distribution, the nature of transcript which the protocol pi can produce and then based on that we will finally rule out that definitely the protocol pi breaches the privacy property and hence we will show we will conclude that whatever we assume regarding the existence of the protocol pi that was incorrect.

So, the first claim is that there is no transcript possible which can be generated if the inputs of the parties would have been 0, 1 and if the inputs of the parties would have been 1, 1 that means the set of transcripts which can be produced by this input combination that is completely disjoint from the set of transcript which would have been generated in this input combination.

And this could be argued very easily. If you see closely here the function output for the case where the inputs of the parties would have been 0, 1 is 0 and the function output which should have been generated if the parties participate with these input combination should be 1 that means the function outputs are different and your transcript determines the function output as well because the final output is decided based on whatever messages the parties have exchanged till now.

And the steps of your protocol pi that means there will be a final output determination step of the protocol pi which should say that okay these are the things which you have seen till now then output this. So, how can it be possible that the same output decision step for this case as well as this case outputs are same values that cannot be possible because we are trying to argue here that there cannot be a transcript tau which is a member of both this input configuration as well as this input configuration.

On contrary assume that there is such a transcript tau which is member of both the set of transcript for the 0, 1 case as well as for the set of transcript as well as it is a member for the set of transcript for the 1, 1 case then the same transcript tau should tell that okay generate the output 0 as well as generate the output 1 which is a breach of the correctness property because we are assuming that the protocol pi gives you absolutely correctness and absolutely privacy that is what we are trying to argue here.

We argued that there exist the perfectly secured protocol and as part of perfect security we also assume perfect correctness, there is no error in the protocol output that means there cannot be a member a transcript which can be generated both for this input combination as well as for this input combination they are disjoint members and obvious that. The other claim regarding the set of transcript is very certain.

My claim is that the set of transcript which could be generated for the input combination 0, 0 is exactly the same as the set of transcripts which would have been generated for the input combination 1, 0 that means this means that if there is some transcript tau belonging to the set of transcript for 0, 0 then it also belongs to a set of all transcript in 1, 0 and vice versa of course for different randomness.

How we prove this? On contrary suppose this claim is not correct what does it mean? That means there is some transcript some bad transcript let me call it bad transcript tau prime which can be generated for input configuration 0, 0, but it can never result if the parties have their respective function input as 1, 0 and if this is the case then we will show we will argue that the protocol breaches the perfect privacy.

But since the protocol is perfectly secure it has to satisfy the perfect privacy and hence this claim is indeed correct. So, let us see how exactly we prove this claim assuming that this claim is not true how exactly we arrive at a contradiction. So, imagine there is a bad transcript which can never result if the parties participate with inputs 1, 0. If that is the case and suppose that transcript prime results when the randomness of the parties are r 1 and r 2 or r 1 and r 2 prime.

So, that means assume a scenario where P 2 is corrupt and it participates with input 0 P 1 has participated with input 0 of course the set of messages which are communicated is tau prime the bad transcript and of course the function output is 0 because both the parties have participated with input 0. Now, if tau prime constitutes a bad transcript then a corrupt P 2 if it sees that okay the messages which are exchanged as per the protocol during the execution r is tau prime.

Then it can easily infer that the input of the other party is 0 it is not 1 because if it is computationally unbounded and if it knows that there is such a bad transcript tau prime which can be present in the set of transcript with 0, 0 but it can never be present in the set of transcript with 1, 0 and if indeed during the execution of the protocol the same transcript tau prime ends up.

And P 2 knows that okay I have actually participated in an execution where the input of other party is 0 it cannot be 1 which is a breach of perfect privacy because if indeed protocol pi satisfied the privacy property then this transcript tau prime could also result for b = 1 and some candidate randomness r 1 double prime, but that is not the case. We are assuming here that there is no possibility that this transcript tau prime could also result if tau input of P 1 would have been 1.

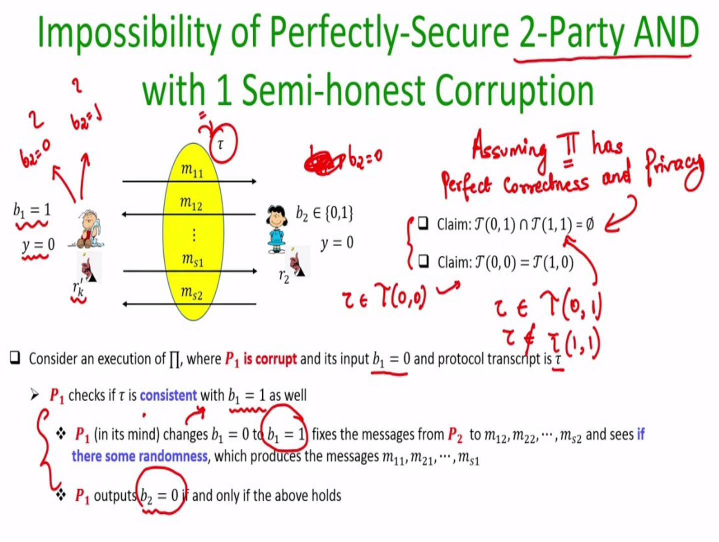
And we participated with some other randomness r 1 double prime that is what we are assuming. When we say that this transcript tau prime is not present in this set that means if b 2 = 0 r 2 prime is fixed it cannot be possible that b 1 = 1 and randomness is r 1 double prime where the set of messages which P 1 would have sent is actually the same messages which P 1 has sent as per the transcript tau prime that is what we are assuming here.

And if this is the case then indeed if P 2 participates in the protocol with b 2 = 0 and if indeed the transcript tau prime is the resultant transcript. Again the probability of happening this is might be less it depends this transcript tau prime can indeed occur only if P 1 is also participating with randomness r 1 prime and P 2 is also participating with randomness r 2 prime with their respective inputs being 0, 0.

Only in that case this transcript tau prime could be generated. The probability that the respective randomness of P 1 and P 2 or r 1 prime and r 2 prime might be very, very small. It might be negligible, but that is a non zero probability. So, what we are arguing here is that if this claim is not true then there is always a non negligible chance that P 2 is a lucky guy who ends up seeing the protocol transcript tau prime.

And if indeed she is lucky and she is the protocol tau prime and her input was b 2 = 0 and randomness was r 2 prime she can easily infer that she has seen or she has participated in a protocol execution where b 1 was 0 and not b 1 was 1 and that is a breach of privacy that is why this claim is also true.

(Refer Slide Time: 23:15)



Impossibility of Perfectly-Secure 2-Party AND with 1 Semi-honest Corruption

$b_1 = 1$
$y = 0$
r'_k
m_{11}
m_{12}
$\vdots$
m_{s1}
m_{s2}
τ
$b_2 \in \{0,1\}$
$y = 0$
r_2
?
$b_2=0$
?
$b_2=1$
$b_2=0$
Assuming Π has Perfect Correctness and Privacy
$\tau \in \mathcal{T}(0,0)$
$\tau \in \mathcal{T}(0,1)$
$\tau \notin \mathcal{T}(1,1)$

□ Claim: $\mathcal{T}(0,1) \cap \mathcal{T}(1,1) = \emptyset$

□ Claim: $\mathcal{T}(0,0) = \mathcal{T}(1,0)$

□ Consider an execution of $\prod$, where P_1 is corrupt and its input $b_1 = 0$ and protocol transcript is τ

- P_1 checks if τ is consistent with $b_1 = 1$ as well
  - P_1 (in its mind) changes $b_1 = 0$ to $b_1 = 1$ fixes the messages from P_2 to $m_{12}, m_{22}, \cdots, m_{s2}$ and sees if there some randomness, which produces the messages $m_{11}, m_{21}, \cdots, m_{s1}$
  - P_1 outputs $b_2 = 0$ if and only if the above holds

So, I have written down these two claims so assuming what we have done till now assuming pi has perfect correctness and privacy we have shown that these two properties must hold. Now based on these two facts we will see an adversary strategy which can allow a corrupt P 1 who participates in the protocol execution pi with b 1 = 0 to analyze the protocol transcript and come to the (()) (24:06) and find out what exactly was the input of the other party with 100% probability.

So, consider an execution where now P 1 is corrupt it participates with input b 1 = 0 and of course its randomness is r 1 it does not know whether input b 2 = 0 or input b 2 = 1. We are now going to see a strategy which will allow P 1 to analyze the transcript tau based on these two facts and come to the conclusion that whether it has participated with b 2 being 0 or b 2 being 1 which will be contradiction to the fact that protocol pi has perfect privacy.

So, here is the strategy of P 1. P 1 once the protocol is over anyhow it will learn that y = 0. It now starts building a table various tables in his mind. He checks in his mind that is it possible that the messages which have been exchanged in the protocol namely this concrete transcript tau is consistent even with b 1 = 1 as well. What does that mean? That means he has now build several incomplete transcript where the messages from P 2 has been fixed as per tau because that is what he has seen.

And this is a similar exercise which we have done for our secret sharing based protocols where a corrupt party it fixes whatever he has seen and put question mark with respect to what could be the candidate input of the other honest parties and try to fill those candidate inputs and see whether that matches with whatever the adversary has seen that is what we are doing here.

He is trying to check whether the same transcript is consistent with b 1 = 1 as well that means P 1 in this mind he has participated in the protocol with b 1 = 0 and that has produced this transcript tau. Now he is doing an exercise in his mind he is trying to change his input to b 1 = 1 and trying different candidate randomness and for this changed b 1 and candidate randomness he is seeing that whether the protocol pi would have produced the messages m 11, m 21, m s 1 from P 1 being sent to P 2.

He is not changing the messages from P 2 that is fixed remember that is fixed to m 12, m 22, m s 2 namely all the round messages which P 2 would have communicated as per tau that is fixed P 1 is basically just changing his input and his randomness and communicating messages as per protocol pi and seeing whether the messages which he would have sent as per protocol pi with the changed input and changed randomness are consistent or the same which tau has.

So, if it is not satisfied with first candidate randomness then it goes to the second incomplete table where again the messages from the second parties have been fixed and he is changed his input from b 1 = 0 to b 1 = 1 and now tries another candidate randomness and tries to fill those missing messages and check whether that is equal to tau or not. If it does not then it tries the next candidate randomness and so on.

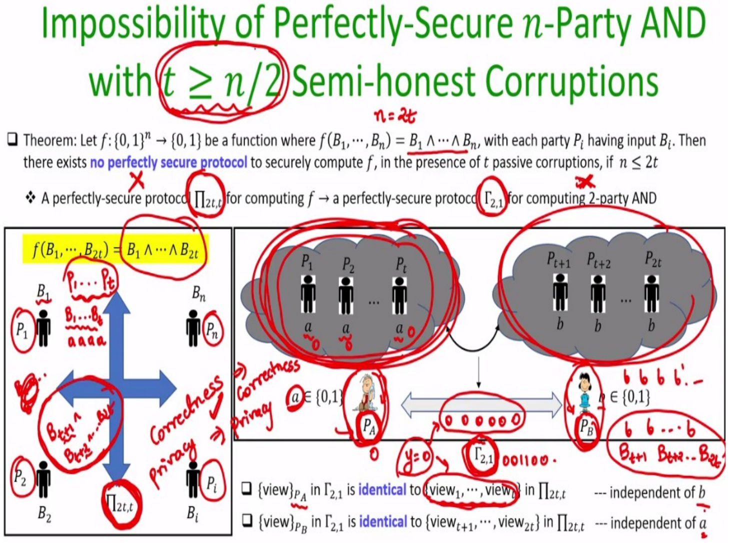
And there are only finitely many candidate randomness of course it could be exponentially large, but remember we are considering a strategy for a potentially computationally unbounded P 1. So, it can run through all candidate randomness and fill all the incomplete tables. Finally, after filling all the tables P 1 comes to the conclusion that other parties input was 0.

If by doing the above exercise he can find a candidate randomness such that candidate randomness along with b 1 = 1 would have produced the same transcript tau. Otherwise he says or he infers that I have participated in the protocol where b 2 was 1 and why this strategy will work. This strategy will work because of these two claims. If indeed b 2 was 0 then we know that and that means this tau belongs to the real transcript tau that has been produced in the real execution actually belongs to the set of all transcripts 0, 0.

Then as per this claim it should also be produced if P 1 have participated with input 1 and with some other candidate randomness whereas if this tau would have been a member of set of transcript where the input of party P 2 would have been 1 then as per this first claim the same transcript tau can never be a member of the set of transcripts where the input of the first party would have been 1 another party input would have been 1.

And that is what precisely P 1 is doing here. So, based on whether this exercise and P 1 mind is successful or not it can easily find out by analyzing the protocol tau whether the other party input was 0 or 1 which is a contradiction to the perfect privacy of the protocol pi and that shows that we cannot design a secure 2 party and protocol.

(Refer Slide Time: 30:19)



Impossibility of Perfectly-Secure n-Party AND with $t \geq n/2$ Semi-honest Corruptions

❑ Theorem: Let $f: \{0,1\}^n \rightarrow \{0,1\}$ be a function where $f(B_1, \cdots, B_n) = B_1 \wedge \cdots \wedge B_n$, with each party P_i having input B_i. Then there exists no perfectly secure protocol to securely compute f, in the presence of t passive corruptions, if $n \leq 2t$

❖ A perfectly-secure protocol $\prod_{2t,t}$ for computing $f \rightarrow$ a perfectly-secure protocol $\Gamma_{2,1}$ for computing 2-party AND

❑ $\{\text{view}\}_{P_A}$ in $\Gamma_{2,1}$ is identical to $\{\text{view}_1, \cdots, \text{view}_t\}$ in $\prod_{2t,t}$ --- independent of b

❑ $\{\text{view}\}_{P_B}$ in $\Gamma_{2,1}$ is identical to $\{\text{view}_{t+1}, \cdots, \text{view}_{2t}\}$ in $\prod_{2t,t}$ --- independent of a

Now we want to generalize this argument to argue that if there are n parties with their respective private bits B 1 to B n and if we want to securely compute the AND of those n bits and up to t of the parties could be corrupted by the computationally unbounded adversary and if we are in the setting where t is greater than equal to n / 2 then there exist no secure protocol to securely compute the AND of the n of the bits of the n parties.

And how exactly we are going to do this? We are going to show it we are going to prove the theorem as follows. So, we will show that if there exist a protocol for securely computing the AND of n parties where n = 2t I am taking the case where n = 2t of course t greater than equal to n / 2 means n can be less than 2t as well, but I take the largest possible value of n satisfying t greater than equal to n / 2 name n = strictly 2t.

So, I will show that if there is a protocol for securely computing the AND of 2t number of parties where even up to half of them can be corrupted by computationally unbounded adversary then using that protocol we can design another protocol which can allow two parties to securely compute the AND of their respective bits, but we already know that this no such 2 party protocol is possible.

That means no such 2t party protocol is also possible that is what we are going to do. So, assume there is some hypothetical protocol for 2t number of parties assuming n = 2t which is secure even if up to half of them can be corrupted that means even if any set of t parties get corrupt the view that this protocol generates for those t bad parties that view will be independent of the inputs of the remaining t parties that is what is the privacy property of this protocol pi 2t, t.

Now, using this protocol we will design our 2 party and protocol. In the 2 party and protocol I am taking the parties to be P A and P B with their respective input bits being 0 A and B respectively. Now the protocol code of pi is publically known. So, what P A can do is the following. P A can think in its mind that it is going to play the role of first t parties P 1 to P t as per the protocol pi assuming that all their input bits are a.

It can do that and party b can do the following. It can play the role of the remaining t parties as per the protocol pi 2t. So, we are assuming that there are 2t number of hypothetical parties. A is playing the role of half of them, B is playing the role of half of them that is a protocol gamma here the two party and protocol. A is playing the role of t parties as per the protocol pi assuming that all their inputs are this a the bit a for the 2 party protocol.

And b is playing the role of another set disjoint t parties as per the protocol pi assuming that all their inputs are b that is a protocol pi that means if protocol pi has instructions for party number P 1, party number P 2, party number P i, party number P n assuming that their bits are B 1, B 2, B i, B n that is what will be the protocol pi it will have instructions for P 1, it will have instructions for P 2, it will have instructions for P i, it will have instructions for P n.

What I am proposing here is the following. In the protocol gamma which is the 2 party protocol Party A takes the instructions of first t parties that are instructed as per the protocol pi and he is going to run those instructions assuming that t parties are participating with input a as per the protocol pi and party P B is going to run the instructions of t + 1th party, t + 2th party, and the 2th party as per the protocol pi assuming that their inputs are the bit P.

Now, if in the protocol pi some interaction is involved between any party in this collection and any party in this collection then whatever messages they are supposed to communicate P A and P B will communicate in this 2 party protocol whereas if in the protocol pi if there is any communication from any party in this group to any party within the same group that will be treated as if P A is communicating that message to itself.

So, the idea here is basically whatever communication 2t parties would have done as per the protocol pi the same communication P A and P B are emulating by dividing the role of the 2t parties among themselves. Whenever there is a requirement of communication among this group and this group P A and P B will communicate otherwise they would not communicate that is the protocol gamma.

So, it is easy to see that the final outcome if the protocol pi satisfies the correctness property then gamma also satisfies the correctness property because the output of the protocol pi would be the AND of the bits of 2t parties and the AND of the emulated 2t parties are here A and B. So, the correctness properties guaranteed that means correctness of this protocol pi implies the correctness of this protocol gamma as well.

Now, let us try to argue the (()) (37:06). Can I say that the privacy of this protocol pi also implies the privacy of protocol gamma and the answer is yes why so? In the protocol gamma it could be either the party P A or it could be the party P B who could be corrupt because in the protocol gamma it is a secure 2 party protocol there are no 2t parties. The role of t parties or the protocol code of t parties are run by P A and the protocol code of t parties are run by P B.

So, there are only 2 parties. What will be view of P A in the 2 party protocol? Well, it will be identical to the view of t parties the first t parties that they would have generated in the protocol pi if all of them would have participated with input bit A and what will be the view of this second party P B in the 2 party protocol? Well, it will be identical to the view of the last t parties if they would have participated in the protocol pi with input being B.

That has come because of the way P A is playing or running the code of first t parties that means whatever random values the P 1, P 2, P t would have been asked to picked those random values P A would have been picked. If P 1 would have been asked to secret share some value P A would have done that. If P 1 would have asked to communicate some message to P 2 okay P A will say that this is what will be view of view 2 and so on.

So, this combined view of this party t parties in the protocol pi will be treated as the view of P A and same for P B. Now what we know is the following. The combined view of first t parties as per the protocol pi should have been independent of the view of the remaining t parties if in protocol pi the first t parties would have been corrupt namely what I am saying is that if P 1 to P t gets corrupt in the protocol pi.

So, they will be having access to B 1 up to B t which in this case in the context of gamma protocol is all set to a, a, a, a and a and they will be anyhow finally learning the output of all the remaining bits and of all remaining bits. So, the final outcome is also consisting of B t + 1 and B t + 2 and B 2t that is also learned as part of the final outcome. Now can I say that if P 1 to P 2 gets corrupt they conclude they learn any additional information about the exact bits of the remaining t parties in the protocol pi.

No, the remaining t input bits of the remaining t parties in the protocol pi could be any combination. It could be the case that all of them are the same bit or they are different bits depending upon just that it has to be ensured that finally the AND of those remaining t bits along with the AND of the t bits of these t parties should produce the final outcome which the pi protocol should have produced.

So, translating it to the context of the gamma protocol I can say that the view P A which the protocol gamma would have generated in case if protocol P A is corrupt will be independent of B t + 1, B t + 2 up to B 2t that means even a possibility of all of them being b is possible or some of them being b and some of them being b prime that is also possible all of them are equiprobable.

That means if P A inputs would have been 0 say that means it would have run all the first t parties with input 0 and it would have learned the final output to be y = 0. Now based on this it cannot rule out whether the inputs of all the remaining t parties are 0 or it cannot rule out that the inputs of some of them are 0 or some of them are 1 and so on because that is coming from the privacy property of pi.

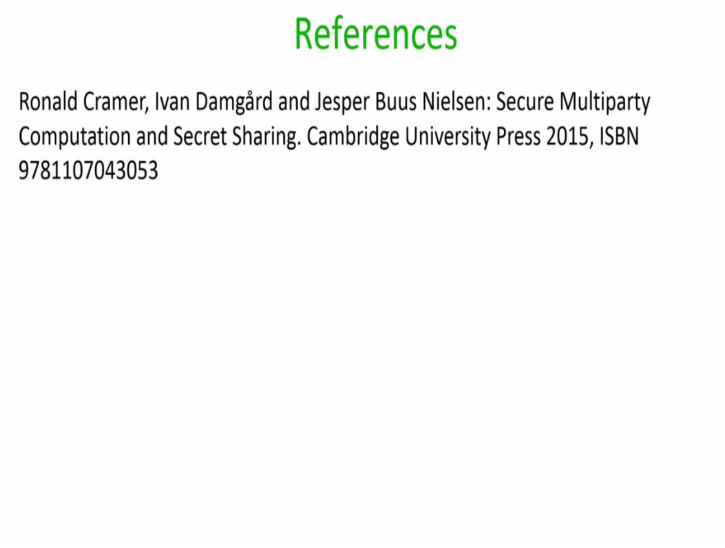
That means y being 0 and the input of first t parties being all 0s as per protocol pi that view could also be produced. The view that is produced there can also result equiprobably from the case when the inputs of all the remaining t parties would have been 0 or half of the remaining t parties have participated with input 0 or with input 1 and so on that means in the context of this translated protocol gamma I can say that the view of the parties P A is completely independent of the actual bit b.

Whereas if P B would have been corrupt in the protocol gamma 2 1 then since its view is identical to the view of last t parties and if the last t parties would have been corrupt in the protocol pi 2t their view would have been independent of the view of the first t parties and the view of the first t parties in this gamma protocol is basically depending on your bit a. So, that means I can say that the view of the party P B it would have been corrupt is independent of the bit a.

So, that is why this implication namely the privacy of the protocol pi translates to the privacy of the protocol gamma is correct that means if at all I have this protocol pi 2t, t I can design a protocol pi gamma 2, 1 by doing this exercise A copying or running the code of first t parties. B running the code of last t parties and whatever messages are supposed to be exchange as per the protocol pi they exchange messages among themselves depending upon the messages are supposed to be exchanged within the group or between the group.

But we know that no such protocol gamma is possible hence no such protocol pi is possible as well and that shows the optimality of this bound for certain functions.

(Refer Slide Time: 43:55)



References

Ronald Cramer, Ivan Damgård and Jesper Buus Nielsen: Secure Multiparty Computation and Secret Sharing. Cambridge University Press 2015, ISBN 9781107043053

So, the proof of the impossibility of secure 2 party and this reduction from the 2 party generalization of the 2 party impossibility to the n party impossibility has been taken from this book. Thank you.