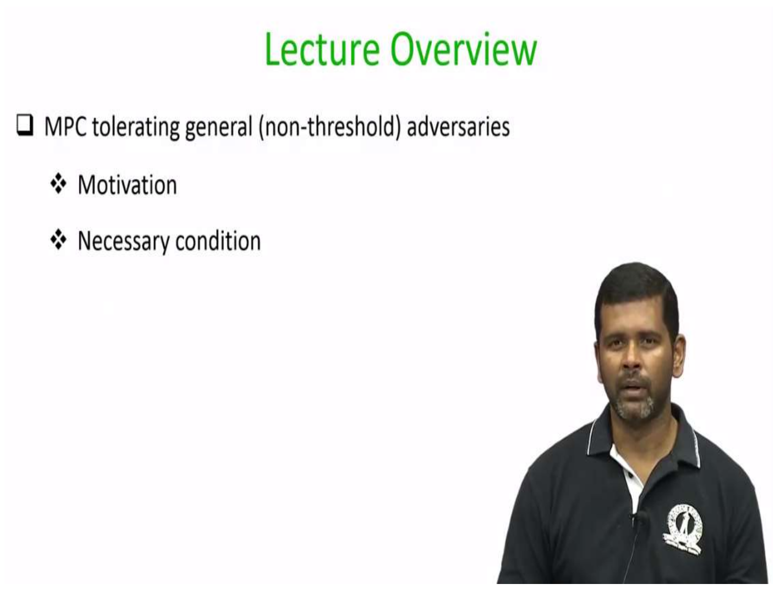
Secure Computation: Part 1

Prof. Ashish Choudhury
Department of Computer Science
Indian Institute of Science – Bengaluru

Lecture – 29
Perfectly-Secure MPC Tolerating General (Non-Threshold) Adversaries

Hello everyone. Welcome to this lecture.

(Refer Slide Time: 00:31)



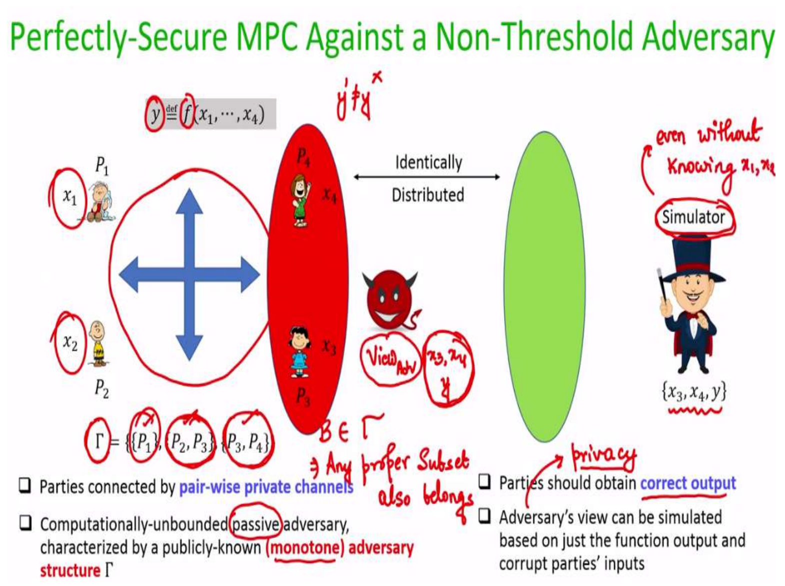
Lecture Overview

- MPC tolerating general (non-threshold) adversaries
  - Motivation
  - Necessary condition

So, till now our focus was on Perfectly Secure Multiparty Computation tolerating a threshold adversary where we upper bounded the corruption capability of the adversary by a publically known threshold. Now, we will be focusing on perfectly secure multiparty computation against the general adversary also known as the non-threshold adversary. So, we will first see the motivation for doing that.

And then we will derive the necessary condition which is required for designing perfectly secure MPC protocol against a non-threshold adversary.

(Refer Slide Time: 01:16)


$y \overset{\text{def}}{=} f(x_1, \cdots, x_4)$
$y' \neq y$ ×
P_1
x_1
P_2
x_2
P_4
x_4
x_3
P_3
Identically
Distributed
even without Knowing x_1, x_4
Simulator
$View_{Adv}$
x_3, x_4
y
$\{x_3, x_4, y\}$
$\Gamma = \{\{P_1\}, \{P_2, P_3\}, \{P_3, P_4\}\}$
$B \in \Gamma$
⇒ Any proper Subset also belongs
privacy

Perfectly-Secure MPC Against a Non-Threshold Adversary

- Parties connected by **pair-wise private channels**
- Computationally-unbounded passive adversary, characterized by a publicly-known **(monotone) adversary structure** Γ
- Parties should obtain **correct output**
- Adversary's view can be simulated based on just the function output and corrupt parties' inputs

So, let us first see the problem definition for perfectly secure MPC against the non-threshold adversary. So, we assume here that we have set of mutually (01:29) parties who are connected by pair wise private channels. So, for simplicity I am taking here the case where we have 4 parties; each party has some private input and the distressed in the system is modeled by this publically known adversary structure which is a collection of potential subsets of parties which can be under the control of a computationally unbounded adversary.

However, we assume here that the adversary is passive that means the corrupt parties will be following the protocol instructions properly they would not deviate from the protocol instructions. So, for instance, an example of an adversary structure could be this where we assume that it could be the case that either P $_1$ gets corrupt during the protocol execution or party P $_2$ or P $_3$ can collectively may get corrupt during the protocol execution or the parties P $_3$ or P $_4$ may collectively get corrupt during the protocol execution which of these three subsets is going to be corrupted during the protocol execution that would not be known beforehand.

The adversary has the flexibility to either corrupt P $_1$ or collectively collect P $_2$, P $_3$ or collectively collect P $_3$, P $_4$. However, it is not allowed to corrupt one party from one subset another party from another subset and so on. So, it has to decide which subset it has to go and target and the adversary structure here is monotone. So, monotone in the sense that if I have some subset P, a potentially corrupt subset of parties which is present in the adversary structure.

Then any proper subset of B also belongs to the adversary structure. That means if P $_2$ and P $_3$ can collectively become corrupt by the adversary then it is also possible that P $_2$ alone can get corrupt by the adversary or P_3 alone can get corrupt by the adversary. Similarly, if there is a possibility that both P_3 and P_4 may get corrupt by the adversary then by monotone property here I mean that it is also possible that only P_3 get corrupt or only P_4 gets corrupt and so on.

So, there will be a publically known function f which could be over some algebraic structure it could be either a ring or a field and for simplicity we assume that each party has a single or input for the function again this is without loss of generality and the function output is y. Again we assume without loss of generality that there is a single function output which is supposed to be publically learn by everyone.

So, this function f takes the inputs of the parties and produces the output y and the goal of the parties is to securely compute the function f without disclosing any additional information about their respective inputs. So, we require a mechanism which allows the parties to interact with each other that will be the MPC protocol and that protocol should have the property that if any subset from this adversary structure gets corrupt then it should not learn anything additional about the inputs of the honest parties apart from what the adversary could infer from the inputs of the corrupt parties and the function output.

So, for instance, suppose if adversary corrupts the subset P_3, P_4 during the protocol then whatever is the view generated let us call it view adversary and by borrowing the terminology that we have used for the threshold setting view consists of the inputs of the corrupt parties, their function output, the local randomness which are used by the corrupt parties during the protocol execution and whatever message they have communicated.

And whatever messages that have received that will be entire view of the adversary. So, intuitively we want to design a protocol which should satisfy two properties. The first property is the correctness that means finally the protocol should guarantee that the honest parties obtain the correct output that means it should not be the case that they obtain output y prime which is different from y that should not be the case.

So, that is what we call as perfect correctness and the privacy property which require from the MPC protocol is that adversary should not learn anything additional beyond the function output and the corrupt parties input. So, the term here additional is very loose. What do we mean by saying that adversary does not learn anything additional? Well, we know now how to formalize this.

We have seen that already in the context of threshold model, threshold adversary setting. So, there by saying that adversary does not learn anything additional we mean that adversaries view can be simulated based on just the function output and the corrupt parties input that means anyhow the view of the adversary will have the inputs of the bad guys. So, in this case they are x_3, x_4 and the function output y.

We want that the protocol should not reveal anything additional beyond what could be inferred by the adversary from x_3, x_4 and y, but in the protocol adversary might have received messages from the honest parties. So, we want to formerly capture that those messages from the honest parties are of no use for the adversary to infer anything additional about the inputs of the honest parties.

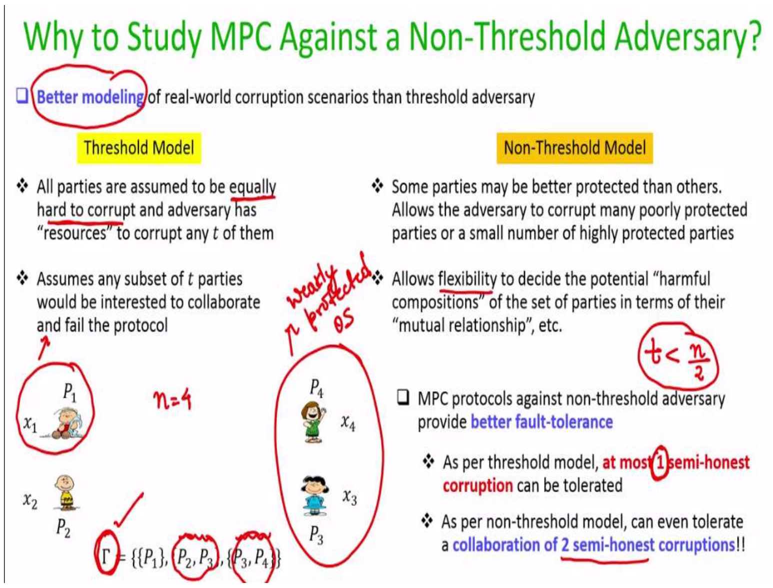
And that can be formally stated by requiring that adversary's view that means whatever messages it has received from the honest parties specifically could be recreated or simulated based on just the knowledge of the corrupt parties input and the function output. If that is the case if the view of the adversary could be simulated just based on the corrupt parties inputs and the function output that means whatever interaction has happened with the honest party and the adversary between the honest party and the adversary does not leak anything about the inputs of the honest parties.

So, more formally we require the following. We will say that the MPC protocol satisfies the privacy property if there exist the simulator algorithm which will be a randomized algorithm and which when given the inputs of the corrupt parties and the function output could produce a view by whatever mechanism and that view should be identically distributed as the view of the adversary which could have been generated during the execution of the MPC protocol by interacting with the honest parties.

And we had seen that how we design the simulator in the case of threshold setting, similar thing we are going to do even in the non-threshold setting so that will be our privacy definition. If we can design such a simulator who can reproduce, regenerate, simulate the view of the corrupt parties in the MPC protocol then that basically concludes that whatever information adversary has received or whatever messages adversary has received from the honest parties they are completely meaningless they are of no use.

They are completely independent of the inputs of the honest parties because this simulator could reproduce the same view even without knowing x 1, x 2.

(Refer Slide Time: 10:51)



Why to Study MPC Against a Non-Threshold Adversary?

- **Better modeling** of real-world corruption scenarios than threshold adversary

Threshold Model

- All parties are assumed to be equally hard to corrupt and adversary has "resources" to corrupt any t of them
- Assumes any subset of t parties would be interested to collaborate and fail the protocol

Non-Threshold Model

- Some parties may be better protected than others. Allows the adversary to corrupt many poorly protected parties or a small number of highly protected parties
- Allows flexibility to decide the potential "harmful compositions" of the set of parties in terms of their "mutual relationship", etc.

$\Gamma = \{\{P_1\}, \{P_2, P_3\}, \{P_3, P_4\}\}$

- MPC protocols against non-threshold adversary provide **better fault-tolerance**
  - As per threshold model, **at most 1 semi-honest corruption** can be tolerated
  - As per non-threshold model, can even tolerate a **collaboration of 2 semi-honest corruptions**!!

So, now let us see the motivation to study MPC against a non-threshold adversary. So, the main reason that we want to study MPC against a non-threshold adversary is that it allows for a better modeling of real world corruption scenario compared to the threshold model. So, let us try to understand that what do we mean by better modeling? So, again for the purpose of demonstration I will take the case where we have 4 parties and then we will see that what are the limitations which are imposed when we model the corruption by a threshold.

So, if we assume a threshold adversary basically what we are saying here is the following. We are saying that all the parties in the system all the n parties they are assumed to be equally hard to corrupt and when we say that adversary can corrupt any t of them by that I mean that adversary has resources to corrupt any t of them adversary do not have enough resources to corrupt more than t to them.

But this is under the assumption that all the n parties are equally hard to corrupt. So, for instance, if there are four parties and if I say t = 2 then basically we are trying to model here that all the 4 parties are equally well protected and adversary has the resources. Resources means computing resources, it could be any kind of resources it has resources to corrupt only t of them or two of them.

It does not have sufficient resources to corrupt more than t of them. So, that kind of scenario is modeled by the threshold adversary where we are considering a setting where all the n parties are equivalent in terms of their computing resources, equivalent in term of their security guarantees or how well they are protected and so on whereas in a non-threshold model we are trying to basically capture the following.

We give the adversary the flexibility. It could be the case that in the real world deployment of the MPC protocol where we have n parties not all the n parties are equally protected. What does that mean? Say for instance it could be the case that P_1 is running weak operating system; weak in the sense less secure operating system sorry it should be the other way round it is running highly protected OS.

Whereas say P_3 and P_4 they are running weakly protected OS. Suppose that is a deployment scenario and that is quite possible that is quite realistic. So, for instance, if someone is running windows operating system and someone is running MAC operating system then the security guarantees which are provided by this respective operating systems are different. If someone is running Linux operating system then the level of security guaranteed by Linux operating system they are different compared to whatever security guarantees they are provided by the windows or the Mac operating system.

So, if we are in a deployment scenario where different parties are not equivalent in terms of how well they are protected than a non threshold model allows the adversary the flexibility to either corrupt many number of poorly protected parties or computer or small number of highly protected parties that means we cannot force a restriction that okay adversary you can just control any t of them.

It depends upon whether he is interested to corrupt many number of small poorly protected operating system or poorly protected parties or small number of highly protected parties. So, that flexibility is provided or allowed in the non-threshold model. If we consider threshold model then there basically we assume that any subset of t parties would be equally interested to collaborate and cause harm to the protocol or breach the privacy property, breach the correctness property and so on.

Whereas again non-threshold model allows you the flexibility, it gives you the flexibility to decide the potential harmful subsets of parties depending upon their mutual relationship. So, again for instance in this example it could be the case that party P_3 and P_4 they are friends of each other and they might be definitely interested to go and attack the protocol whereas if I say $t = 2$ then it models the case that it could also be the case that P_1 and P_3 are interested to collaborate and cause harm to the protocol.

But it might be possible that P_1 and P_4 are actually not in talking term we do not want to collaborate together and cause harm to the system. So, that kind of flexibility to kind of model prior relationship among parties, what are their preferences with whom they would like to collaborate and cause harm to the protocol etcetera that flexibility is provided in the non-threshold model that flexibility is not there in the threshold model.

Another reason motivation for studying the non threshold model it is a generalized form of adversary. So, if we can design an MPC protocol against the non-threshold adversary of course we can derive a protocol against the threshold adversary from the non-threshold model MPC protocol because remember threshold model is always a special case of the non-threshold model where the size of every subset and the adversary structure is t.

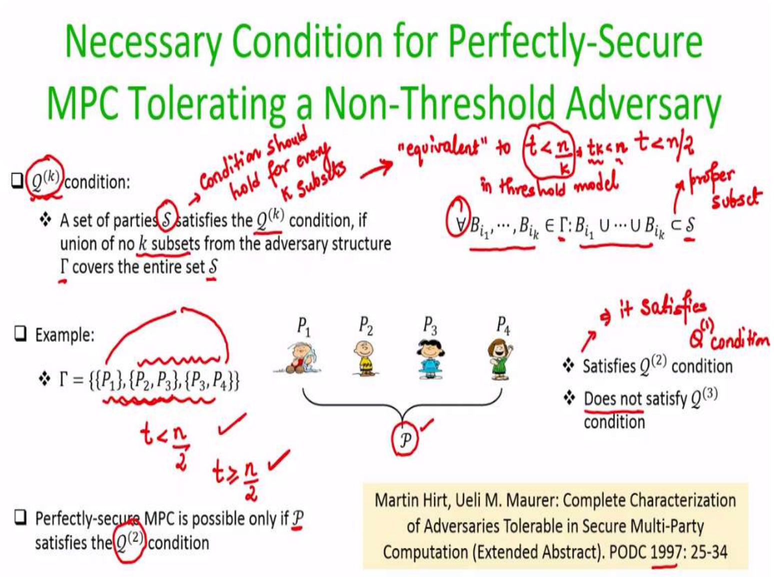
So, again in terms of generality studying the non-threshold model is really motivated and finally MPC protocols against non threshold adversary provides better fault tolerance that means if you model the adversary as a non-threshold model then there is a possibility that the resultant protocol that you design tolerates more number of corruptions than what you could tolerate if you model the adversary by a threshold.

So, again if we take this deployment scenario where $n = 4$ and if this adversary structure is there if I try to run any threshold MPC protocol among these 4 parties you can tolerate at most one semi honest corruption because remember we have derived the lower bound that you need t less than $n / 2$ condition for designing perfectly secure MPC protocol if you have non-linear gates in the circuit.

So, if we have n = 4 definitely you cannot tolerate two corruptions, you cannot run the BGW protocol with t = 2, you cannot run the GRR protocol with t = 2 and in fact you cannot run any threshold protocol with t = 2 because of this impossibility result, but as we will see later it is indeed possible to design an MPC protocol tolerating this given adversary structure and if you see closely here in this given adversary structure you have potential bad compositions consisting of up to 2 parties.

That means the protocol that we will design assuming that this is the adversary structure, we want to tolerate this adversary structure will also give you security guarantees even if P_2, P_3 come together and collaborate to breach the privacy property or even if P_3 and P_4 come together and collaboratively try to breach the privacy property that means you can tolerate two semi honest corruptions compared to one semi honest corruption which you could have tolerated if you would modeled the corruption by a threshold t. So, that justifies that why we should study MPC against the non-threshold adversary.

(Refer Slide Time: 19:56)



Necessary Condition for Perfectly-Secure MPC Tolerating a Non-Threshold Adversary

- ❑ $\boxed{\mathcal{Q}^{(k)}}$ condition:
  - ❖ A set of parties $\mathcal{S}$ satisfies the $\mathcal{Q}^{(k)}$ condition, if union of no k subsets from the adversary structure Γ covers the entire set $\mathcal{S}$ → condition should hold for every k subsets → "equivalent" to $\boxed{t < \frac{n}{k}}$, $tk < n$ $t < n/2$ in threshold model

$$\forall B_{i_1}, \cdots, B_{i_k} \in \Gamma : B_{i_1} \cup \cdots \cup B_{i_k} \subset \mathcal{S}$$

↗ proper subset

- ❑ Example:
  - ❖ $\Gamma = \{\{P_1\}, \{P_2, P_3\}, \{P_3, P_4\}\}$

$t < \frac{n}{2}$ ✓ $t \geq \frac{n}{2}$ ✓

$\underbrace{P_1 \quad P_2 \quad P_3 \quad P_4}_{\mathcal{P}}$ ✓

- ❖ Satisfies $\mathcal{Q}^{(2)}$ condition → it satisfies $\mathcal{Q}^{(1)}$ condition
- ❖ Does not satisfy $\mathcal{Q}^{(3)}$ condition

- ❑ Perfectly-secure MPC is possible only if $\mathcal{P}$ satisfies the $\mathcal{Q}^{(2)}$ condition

Martin Hirt, Ueli M. Maurer: Complete Characterization of Adversaries Tolerable in Secure Multi-Party Computation (Extended Abstract). PODC 1997: 25-34

So, now let us try to derive the necessary condition for perfectly secure MPC tolerating a non threshold adversary assuming a semi honest corruption. Again remember in this course our focus is only on semi honest corruptions. So, in the next course we will see the conditions which are required for tolerating a non-threshold adversary which is (20:19). So, similar to the case of threshold setting where we had the requirement or the necessary condition that t should be strictly less than $\frac{n}{2}$ for any generic MPC perfectly secure MPC protocol we have an equivalent necessary condition in the non-threshold setting as well.

So, the condition is derived based on what we call as Q_k condition. So, let us first try to understand what is Q_k condition where Q is within parenthesis. So, imagine you are given a set of parties S it could be the entire set of n parties or it could be a subset of those parties. So, imagine you are given a set of parties S. We will say that this subset S satisfies the Q_k condition if the following hold.

Given the adversary structure gamma you take any k subsets from that adversary structure. The union of those k subsets which you have decided should not cover the entire set of parties S that means there should be at least one party which is still present in S, but not present in the union of those k subsets. If that is the case then we will say that the set of parties S satisfies the Q_k condition.

And this should hold for any k subsets from the adversary structure even if there exist one collection of k subsets from the adversary structure which covers the entire set of parties S we can no longer say that S satisfies the Q_k condition. So, this should hold for every k subsets the condition should hold for every k subsets that is important. So, formally what I am saying is the following you are given the adversary structure gamma for every subsets B_{i1}, B_{i2}, B_{ik} which you take from your adversary structure.

The union of those k subset should be a proper subset of your set of parties and this should hold for every k subsets. If that is the case then we will say that the set of parties S satisfies the Q_k condition. So, let us see some example here. Imagine this is the adversary structure given and I take the set of parties P to be the set S itself. Now, let us see whether the set of parties P which is consisting of 4 parties here satisfies the Q_2 condition with respect to the adversary structure.

And indeed it satisfies the Q_2 condition because if you take the union of these two subsets then you have party P_4 which is missing. If you take the union of the last two subsets you have party 1 which is missing and if you take the union of the first and the last subset you have party P_2 who is missing. So, that is why we will say that this set satisfies the Q_2 condition.

And if it is easy to see that if it satisfies Q_2 condition this also implies it satisfies Q_1 condition because if the union of any two subsets cannot cover the entire set of parties that also means the union of any one subset does not cover the entire set of parties so that we can always conclude. So, if some set satisfies Q_k condition it also satisfies Q_{k-1} condition, Q_{k-2} condition and so on.

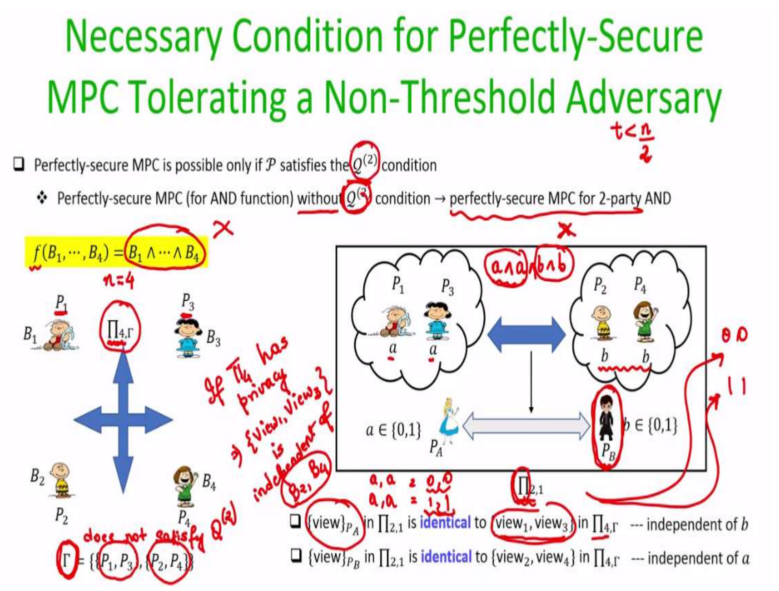
However, this same set P with respect to the same adversary structure does not satisfy Q_3 condition because if you take the union of these three subsets of gamma you basically have the entire set of parties P covered. So, that is why it does not satisfies the Q_3 condition. So, now intuitively if you observe here closely this Q_k condition is equivalent to saying that your t should be less than n over k in threshold model because if I say t less than n over k.

And if I am considering threshold model then basically this means that t times k is strictly less than n that means you take k subsets of size t overall their union is strictly less than n. So, Q_k condition is basically generalization of this condition t less than n over k in the non-threshold model that is intuitively the way to understand this Q_k condition. Now, what is the necessary condition for designing any generic MPC protocol in the non-threshold model?

So, this beautiful work published in 1997 shows that the necessary condition is that the set of parties P with respect to your adversary structure should satisfy the Q_2 condition only then you can design a perfectly secure MPC protocol that means you have the set of n parties and now you give me an adversary structure with respect to that adversary structure Q_2 condition is satisfied then I can design a protocol.

But if you give me an adversary structure with respect to which the Q_2 condition is not satisfied I cannot design an MPC protocol. It is equivalent to saying that if you ensure t less than n / 2 in the threshold model then protocol is possible, but if t is greater is greater than equal to n / 2 then depending upon whether the non-linear gates are present in the circuit or not I may or may not be able to design the MPC protocol.

(Refer Slide Time: 27:24)


Necessary Condition for Perfectly-Secure MPC Tolerating a Non-Threshold Adversary

- Perfectly-secure MPC is possible only if $\mathcal{P}$ satisfies the $Q^{(2)}$ condition
  - Perfectly-secure MPC (for AND function) without $Q^{(2)}$ condition → perfectly-secure MPC for 2-party AND

$f(B_1, \cdots, B_4) = B_1 \wedge \cdots \wedge B_4$

$\Gamma = \{\{P_1, P_3\}, \{P_2, P_4\}\}$

- $\{\text{view}\}_{P_A}$ in $\prod_{2,1}$ is identical to $\{\text{view}_1, \text{view}_3\}$ in $\prod_{4,\Gamma}$ --- independent of b
- $\{\text{view}\}_{P_B}$ in $\prod_{2,1}$ is identical to $\{\text{view}_2, \text{view}_4\}$ in $\prod_{4,\Gamma}$ --- independent of a

So, now let us prove the necessity of this condition. So, we will show that there exist certain functions which cannot be computed in a perfectly secure way if the set of parties does not satisfy the Q 2 condition and how do we prove that? We will prove that by proving the following implications. We will show that on contrary if you can design a perfectly secure MPC protocol for AND function for n party AND function tolerating an adversary structure which does not satisfy the Q 2 condition.

Then we will show that using that protocol we can design another protocol which can allow 2 parties to secure compute the AND of their respective bids. Even in the presence of an adversary who can corrupt any one of them, but we know that this later thing is not possible, but if this implication is overall true then that means that whatever we assume regarding the existence of this n party protocol for computing the AND function tolerating an adversary structure not satisfying the Q 2 condition is also incorrect. So, that is a proof idea.

So, we will assume that you have a protocol even if the Q 2 condition is not satisfied and then we have to use it and show that how you can convert it into a secure 2 party protocol and whatever the correctness and privacy guarantees are there for the n party protocol tolerating a non Q 2 adversary will be translated to the privacy and correctness of the resultant 2 party and protocol.

So, I will demonstrate the proof I will show how to convert the n party protocol into the 2 party protocol. For the n party protocol I am taking n = 4 and imagine you have a protocol π_4 you can imagine you have a protocol π_n for the n party and function the parties want to securely compute here is the AND of the inputs of the n parties each party has a private bit here.

And as per our assumption this protocol can tolerate this adversary structure it can tolerate a computationally unbounded adversary modeled by this adversary structure and you can see that this adversary structure does not satisfy Q_2 condition. Why it does not satisfy Q_2 condition? Because there are only first of all two sets given in the adversary structure and if you take the union of those two subsets you get the entire set of parties P_1, P_2, P_3, P_4.

We do not know what are the steps of the protocol π_4 the protocol is assume to exist and if it is there then π_4 will have certain steps for P_1, it will have certain steps for P_2, it will have certain steps for P_3, it will have certain steps for P_4. So, it may ask P_1 to pick random coins of certain form, P_2 to pick random coins of certain form, P_3 to pick random coins of certain form, P_4 to pick random coins of certain form.

And then depending upon what are the random coins of respective parties and what are the respective inputs of the parties π_4 will have instructions of it and this message if you receive this message than do this etcetera, etcetera. We do not care what are the internal details of the protocol π_4 it is an abstract protocol which we are assuming that it exist. It has the correctness property.

It has the privacy property even though your adversary structure violates the Q 2 condition that is what we are assuming here. Now using this protocol assuming such a protocol is there we will design a protocol π_2 which will allow 2 parties say Alice and Bob with private inputs A and B respectively which are bits and their goal is to basically compute the AND of A and B.

We know no such protocol is there, but we will see that assuming protocol π_4 is there how we can design a protocol which allows Alice and Bob to compute securely the AND of their respective bits and again the proof will be very similar to what we have seen when we derived the necessity condition of t less than $\frac{n}{2}$ there we first argued the impossibility of existence of any perfectly secure 2 party and protocol.

And then we generalize that argument to show that there exists no protocol for securely compute n parties and if t is greater than equal to n / 2 we are going to use similar arguments here. So, what P A is going to do? It will take the protocol code of P $_1$ and protocol code of P $_3$ as per the protocol π_4 and run their actions assuming that they are participating in the protocol π_4 with input bits a and a respectively.

That means whatever is the input of Alice she is basically playing the role of P $_1$ and P $_3$ as per the protocol π_4 assuming that P $_1$ and P $_3$ would have participated in the protocol π_4 with their respective inputs being A and the same way as part of the protocol π_2 Bob is going to do the following. His private bit is B. He is going to play the role of party P_2 and P_4 as per the protocol π_4 assuming that P $_2$ and P $_4$ have their respective inputs B.

And if in the protocol π_4 there is any step where say P $_1$ needs to communicate to any of the parties P $_2$, P $_4$ or P $_3$ needs to communicate to any of the parties P $_2$, P $_4$ and vice-versa. Say P $_2$ needs to communicate to P $_1$ or P $_3$ or P $_4$ needs to communicate to P $_1$, P $_3$ then in that case Alice and respectively Bob exchange those messages with each other in the protocol π_2 so that is what the whole protocol 1.

The protocol code is Alice runs the steps of π_4 assuming P 1 and P 3 have their inputs A, Bob runs P 2, P 4 assuming that they are participating in protocol π_4 with inputs B whenever P 1 or P 3 any of the parties in P 1, P 3 is suppose to communicate to the parties P 2 or P 4 Alice sends those messages to Bob whenever P 2 or P 4 is suppose to communicate any message to P 1 or P 3 in the protocol π_4.

Bob sends those messages to Alice in the protocol π_2 that is what is the whole protocol and it is easy to see that basically what Alice and Bob are trying to do is they are trying to basically compute, they are basically running an instance of the protocol π_4 to compute the AND of a and a and b and b. Assuming that Alice has inputs of two of the parties namely P 1, P 3 and Bob has inputs of two of the parties P 2, P 4.

That is what Alice and Bob basically are doing in the protocol π_2. Now what we can say correctness of the protocol π_2 is obviously there. If indeed π_4 satisfies the correctness property and it outputs the AND of bits of the n parties n is 4 in this case then so is the protocol $\pi_2\pi_2$ will allow Alice and Bob to compute the AND of their respective bits.

So, the correctness property of π_4 translates the correctness property of π_2. Now let us argue the privacy. In the protocol π_2 either Alice could get corrupt or Bob could get corrupt if Alice gets corrupt by the adversary then what will be the view of Alice or the adversary in the protocol π_2. Well, it will be identical to the view of party 1 and party 3 if they would have participated in the protocol π_4 with inputs a and a each.

And if adversary would have corrupted parties P $_1$, P $_3$ that is possible because in the protocol π_4 it is either P $_1$ P $_3$ which can get corrupt or P $_2$, P $_4$ which can get corrupt which translates to the fact that either Alice could get corrupt in protocol π_2 or Bob could get corrupt in the protocol π_2. So, if Alice gets corrupt her view will be identical to the view of P $_1$,P $_3$ as per the protocol π_4.

But we know that if π_4 has privacy property and we are assuming it has privacy property then it implies that the collective view 1, view 3 when the parties P $_1$, P 3 gets corrupt is independent of the inputs B $_2$ B $_4$ that means it could be any B $_2$, B $_4$ which could have produced the view 1, view 3 and B $_2$, B $_4$ in this case are the inputs of Bob and Alice knows that Bob has actually played the role of P $_2$, P $_4$ with the same inputs.

So, it could be either 0, 0 or 1, 1, but the privacy properties of π_4 guarantees that the view of the Alice which is actually the view of P $_1$, P $_3$ is independent of whether the inputs of 2, 4 are 0, 0 or 1, 1 or in a sense whether the input of Bob is 0 or 1 and running the same argument if adversary would have corrupted P $_2$, P $_4$ in the protocol π_4 that is equivalent to saying that Bob gets corrupt in the protocol π_2.

And if Bob gets corrupt in the protocol π_2 then its view is identical to the view of corrupt P $_2$ and corrupt P $_4$ as per protocol π_4, but that view is independent of whether a, a is 0, 0 or whether a, a is 1, 1. So, if Bob gets corrupt then whatever view is generated in the protocol π_2 for Bob that will be independent of the inputs being 0, 0 or inputs being 1, 1 or equivalent to saying that whether Alice input is 0 or whether Alice input is 1.

So, that means if at all the protocol π_4 is there tolerating this adversary characterized by this non-threshold adversary structure then so is protocol π_2, but we have already proved that there is no secured 2 party protocol allowing them to securely compute the AND of their respective bits that means whatever we assume regarding the existence of π_4 that is false and that shows that this Q 2 condition is indeed a necessary condition. With that, I end this lecture. Thank you.