

Secure Computation: Part II
Prof. Ashish Choudhury
Department of Computer Science and Engineering
Indian Institute of Science, Bengaluru


Lecture - 17
Lower Bound for Number of Parties for Byzantine Agreement: Part III

Hello everyone, welcome to this lecture.

(Refer Slide Time: 00:25)

Lecture Outline

- Characterization for perfectly-secure Byzantine agreement in incomplete graphs

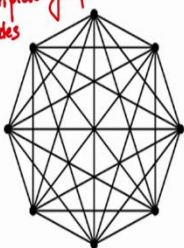


So, in this lecture we will discuss about the characterization for perfectly secure Byzantine agreement in incomplete graphs.

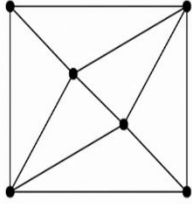
(Refer Slide Time: 00:32)

Characterization of Perfectly-Secure BA in Incomplete Graphs

K_n denotes a Complete graph with n nodes



Complete graph K_n



□ Perfectly-secure BA possible if and only if $n > 3t$

□ When is perfectly-secure BA possible in an incomplete graph with t Byzantine faults?

So, we had already seen that if we are interested to design a perfectly secure Byzantine agreement protocol in a network which is modeled by a complete network complete graph, then the condition $n > 3t$ is necessary.

So, K_n here the notation K_n it denotes a complete graph with n nodes. We now want to find out the necessary condition required to design a perfectly secure Byzantine agreement protocol in incomplete graphs, where there could be t Byzantine faults.

(Refer Slide Time: 01:33)

Characterization of Perfectly-Secure BA in Incomplete Graphs

□ Perfectly-secure BA possible in an incomplete graph with n nodes and t Byzantine faults is possible if and only if:

- ❖ $n > 3t$
- AND**
- ❖ The vertex-connectivity of the graph is at least $2t + 1$

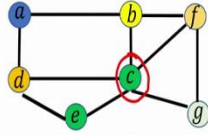
□ Vertex connectivity of a graph $\kappa(G)$

- ❖ Minimum number of vertices to be deleted to either disconnect the graph or produce a graph with a single node

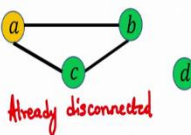
If the graph G is already disconnected

if $G = K_n$

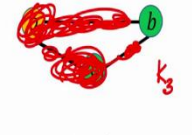
$0 \leq \kappa(G) \leq n - 1$



$\kappa = 2$



Already disconnected
 $\kappa = 0$



$\kappa = 2$

So, the characterization or the necessary condition is the following. It can be shown that perfectly secure Byzantine agreement in an incomplete graph, where we have n nodes and up to t of them could be Byzantine corrupted is possible if both the two conditions are satisfied. The first condition is that n has to be compulsorily greater than $3t$ and the second condition is that the vertex connectivity of the underlying graph underlying network has to be at least $2t + 1$.

So, let us first try to understand what we mean by the vertex connectivity of a graph. I am sure people who have studied graph theory they will know what vertex connectivity is. So, this vertex connectivity of a graph G is denoted by the notation $\kappa(G)$ and it denotes the minimum number of vertices which I have to delete from the graph so that the graph either becomes a disconnected graph or, after deleting those nodes, I am left with a graph which has only a single node left.

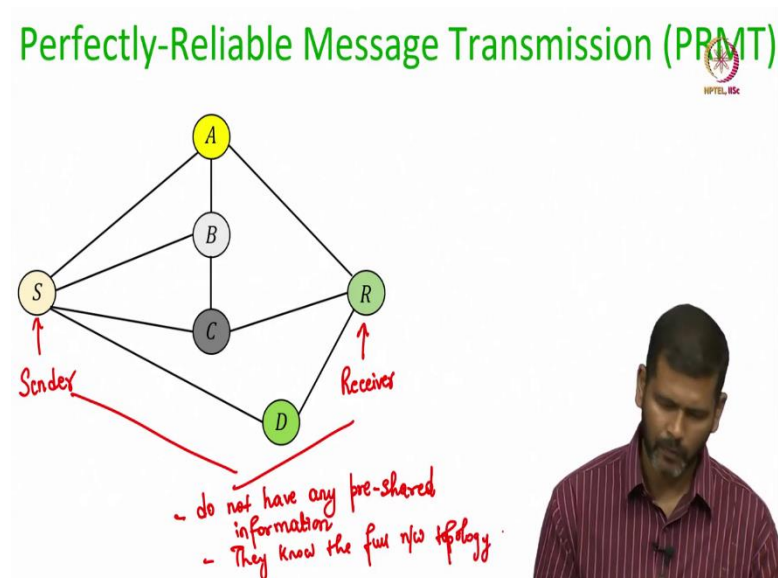
The minimum number of such nodes whose deletion either leaves a disconnected graph or a graph with a single node is called as the vertex connectivity of that graph and that is denoted by the notation $\kappa(G)$. It is easy to see that $\kappa(G)$ will be a quantity in the range 0 to $n - 1$. 0 if the graph G is already disconnected. That means, I do not have to delete any other node to disconnect the graph G . Or the vertex connectivity could be $n - 1$ if the graph G is a complete graph with n nodes. Because in a complete graph of n nodes where there is an edge between every pair of vertices even if I delete up to $n - 1$ nodes, I cannot disconnect the graph.

I can only ensure there that I am left with only a graph consisting of a single node if I delete $n - 1$ nodes. So, pictorially let me demonstrate the vertex connectivity few with few examples. So, in this graph the vertex connectivity is 2. This is because, for instance, suppose I delete the nodes c and f . Then this node g will be left alone because as soon as I delete the node c this edge between c and g vanishes.

And the edge between g and f vanishes if I delete the node f . Whereas, if I just delete any one node in the graph, say for instance if I delete only the node c , then the graph remains connected. Because from g I can still reach f . And through g I can reach to every other node in the graph.

The vertex connectivity of this graph is 0, because it is already disconnected. And this is the complete graph K_3 . So, its vertex connectivity is 2. If I just delete c my graph remains connected and if I now remove the node a and the edges incident with the node a , I will be left with a single node graph. So, that is why the vertex connectivity will be 2.

(Refer Slide Time: 06:03)

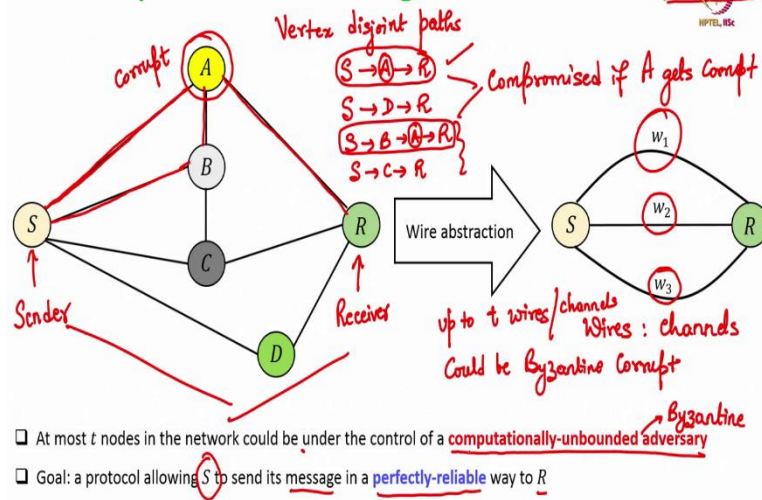


Now, let me introduce a problem called perfectly reliable message transmission or the PRMT problem and this will be useful later while understanding the characterization for perfectly secure BA in an incomplete graph.

So, we are given here a network communication network and we have a sender S and we have a receiver R and they do not have any pre shared information. They do not have any pre shared information, but they know the full network topology.

(Refer Slide Time: 07:18)

Perfectly-Reliable Message Transmission (PRMT)



Also it is guaranteed that apart from the sender and the receiver up to t nodes in this network could be under the control of a computationally unbounded Byzantine adversary. Our goal is to do the following: We will first abstract this underlying network in the form of a collection of a wires or channels between the sender and the receiver.

So, wires you can imagine to be some kind of communication channels and these wires are disjoint. So, w_1, w_2, w_3 are disjoint wires. So, how do we get 3 wires between S and R here? So, how many vertex disjoint paths do we have between S and R ? So, we have the path S to A and A to R . We have the path S to D and then from D to R . Now you might be saying that there are two more paths $S \rightarrow B \rightarrow A \rightarrow R$ and $S \rightarrow C \rightarrow R$, but the point is that the node A could get compromised.

Then any communication which happens between S and R and which goes through this intermediate node A will be completely under the control of the adversary. So, that means, whatever sender communicates through the path S to A to R and whatever sender communicates through the path S to B and then B to A and then A to R . Both these paths are compromised if A gets corrupt.

So, that is why the best we can do here is to assume that we have three vertex disjoint paths. Indeed we have 3 vertex disjoint paths. So, the paths S to A to R and the path S to B to A to R . They are not vertex disjoint they share a common node namely A . So, that is

why both these paths will be abstracted by a single wire w_1 , then the path S to D to R will be abstracted as wire number 2 and the path S to C to R will be abstracted as a wire w_3 ok.

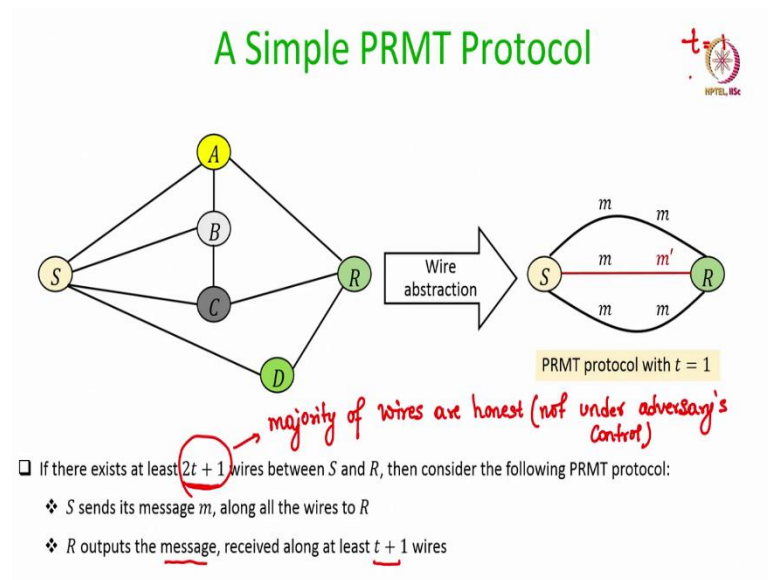
So, since up to t intermediate nodes could be corrupt among these n wires. Not n wires whatever is the number of wires between S to R up to t wires or channels whatever you consider it could be Byzantine corrupt. That means, the communication over those t channels is completely under adversary's control. Adversary can do whatever it wants over those t channels. So, it can simply block the communication over those t channels or it can change the contents over those t channels and so on.

The exact identity of the t channels which are corrupted by the adversary will not be known to the sender and the receiver because neither sender nor receiver will be knowing the exact identity of the corrupt nodes in the network. So, that is why they will not be knowing beforehand which wires are going to be under the control of the adversary. Now what is the goal here? What is the goal of the PRMT problem?

So, we are given this setting a network which is modeled by a collection of vertex disjoint wires between a sender party and a receiver party and sender will have some message from some message space. We want a mechanism a protocol which would allow the sender to send its message in a perfectly reliable way to the receiver. That means, we need a protocol according to which sender should send some information over these wires.

So, that even if the communication over those t wires is changed or blocked in whatever way, receiver should be able to recover back the message and this should hold even if the adversary is computationally unbounded.

(Refer Slide Time: 13:02)



So, let us see a very simple PRMT protocol. Imagine there exist $2t + 1$ wires between the sender party and the receiver party.

Say for instance let us take $t = 1$. So, so we have 3 wires between the sender and the receiver and suppose sender has the message m . Then what the sender can do is the following. It simply sends the message m along the first path or the first wire, sends the same message along the second wire, and sends the same message along the third wire.

Since one of these 3 wires could be Byzantine corrupt. Say for instance the second wire is Byzantine corrupt, the adversary can change the copy of m to m' when it is going over the second wire. The receiver will not be knowing which wire among these 3 wires have delivered the incorrect copy of m , but what it knows is that since, among the $2t + 1$ wires, the majority of wires are honest.

And when I say honest, I mean to say not under adversary's control. It knows that it will receive at least $t + 1$ copies of sender's message and there could be at most t messages which are different from what sender has communicated. So, it can simply output the message which has been received $t + 1$ times along $t + 1$ different wires, that is what is going to be the senders message. So, that is a very simple PRMT protocol.

(Refer Slide Time: 15:11)

Characterization of Perfectly-Secure BA in Incomplete Graphs

Then we can emulate a complete graph K_n over an incomplete graph

Perfectly-secure BA possible in an incomplete graph with n nodes and t Byzantine faults is possible if and only if:

- $n > 3t$ AND The vertex-connectivity of the graph is at least $2t + 1$

we every pair of parties (P_i, P_j) there are atleast $2t+1$ wires

Menger's Theorem (Graph Theory): A graph is k -connected if and only if there exists at least k wires between every pair of nodes in the graph

Sufficiency proof

□ (\Rightarrow part): If $n > 3t$ and The vertex-connectivity of the graph is at least $2t + 1$ then perfectly-secure BA is possible in an incomplete graph with n nodes and t Byzantine faults

- ❖ Π_{BA} : perfectly-secure BA over a complete graph with $n > 3t$ (EIG, phase-king)
- ❖ Run Π_{BA} over the given incomplete graph with $n > 3t$
 - Any message from P_i to P_j as part of Π_{BA} is sent through a PRMT protocol if no direct edge between P_i and P_j

- If in the incom graph, the edge (P_i, P_j) is not
- If the edge (P_i, P_j) is missing

Now, coming back to the characterization of perfectly secure BA. So, our characterization was that in an incomplete graph if at all you want to design a perfectly secure BA protocol then that requires $n > 3t$ as well as the vertex connectivity of the graph to be $2t + 1$. So, I will be just showing the sufficiency proof here. So, this characterization is both necessary and sufficient because this is an if and only if condition.

So, I will show that if the condition $n > 3t$ holds and if the vertex connectivity of the graph is at least $2t + 1$; that means, both these conditions are there, then there exists a perfectly secure BA protocol in an incomplete graph ok. Even if up to t nodes or t parties out of the n parties are Byzantine corrupt and the proof is very simple. You take any perfectly secure BA protocol over a complete graph with the condition $n > 3t$.

Say the EIG protocol or the phase king protocol whichever protocol you want where we have a complete graph, but now we want to design a protocol over an incomplete graph. So, we cannot run the EIG protocol or the phase king two protocol over an incomplete graph because in the EIG protocol there are instructions that every party sends a message to everyone else. But, in this incomplete graph we do not have a direct channel between every pair of parties.

So, what we do here is the following: We run the existing protocol over the complete graph with $n > 3t$ and whenever as part of that protocol a message is supposed to be communicated from the party P_i to party P_j we check the following. If in the incomplete

graph the edge $P_i - P_j$ is present, then its fine send the message directly. That means P_i sends the message directly to P_j over that channel.

But if the channel from P_i to P_j , the edge $P_i - P_j$ is not there, then what we can do is the following: Since we know that the vertex connectivity of the graph is $2t + 1$, then by invoking the Menger's theorem from the graph theory which states that if your graph is k connected then there exists k wires between every pair of nodes in the graph.

So, that means, if the vertex connectivity of the graph is guaranteed to be $2t + 1$. This means that between every pair of parties (P_i, P_j) there are at least $2t + 1$ wires. So, if P_i is supposed to send any message to P_j as per the protocol π_{BA} and if the direct channel from P_i to P_j is not there, then what P_i can do is it can invoke a PRMT protocol which we have discussed just now and send that message to P_j .

Say for instance if we take this incomplete graph here and say we take $t = 1$ what I am saying here is that suppose in the BA protocol u_1 is supposed to send its message to u_3 , but there is no direct channel from u_1 to u_3 . But how many wires are there between u_1 and u_3 ? I have the wire 1 namely the path from u_1 to u_2 and u_2 to u_3 that is wire number 1. I have another wire from u_1 to u_3 namely the one going through the intermediate node u_4 that is wire number 2 and I have another wire between u_1 and u_3 going through the intermediate node v_2 .

Let me call it w_3 , there are 3 wires. Now what u_1 can do is it is supposed to send the message m in the protocol π_{BA} . What it can do is it can trigger the previous PRMT protocol. So, it will send the message m along this wire, this wire and this wire. Say for instance the node u_2 or the party who is controlling the node u_2 is Byzantine corrupt. That means, the wire w_1 is corrupt then it can forward m' instead of m .

But, the wires w_3 and w_2 will forward m to u_3 and u_3 the party who is controlling u_3 who is sitting over the node u_3 will be able to recover m and then it will proceed in whatever way it is supposed to after receiving the message m from the node u_1 according to the protocol π_{BA} .

If the vertex connectivity of the graph is at least $2t + 1$, then we can emulate a complete graph. We can emulate a complete graph K_n over an incomplete graph; that means, even

though physically we do not have a complete graph, we can imagine that we have a virtual complete graph where every communication between P_i and P_j can be emulated through a PRMT protocol because there will be at least $2t + 1$ wires or vertex disjoint paths guaranteed between P_i and P_j .

And now if we have a complete graph either physical for complete graph or a virtual complete graph, we know that we have plenty of BA protocols perfect which are perfectly secure if the condition $n > 3t$ holds which will be guaranteed because of the first part of the necessity condition. So, that shows that you can design the perfectly secure BA protocol you can run a perfectly secure BA protocol even over an incomplete graph if you have sufficient connectivity in the underlying network.

(Refer Slide Time: 23:46)

Characterization of Perfectly-Secure BA in Incomplete Graphs

□ (\Leftarrow part): If perfectly-secure BA is possible in an incomplete graph with n nodes and t Byzantine faults then:

- ✱ $n > 3t$
- AND
- ✓ The vertex-connectivity of the graph is at least $2t + 1$

If any of these two properties are violated

□ Necessity of $n > 3t$:

- ✱ Let there exist a perfectly-secure BA protocol Π_{BA} in an incomplete graph with $n \leq 3t$
- ✱ Π_{BA} will also be a perfectly-secure BA in a complete graph with $n \leq 3t$
- But there exists no perfectly-secure BA in a complete graph with $n \leq 3t$

□ Necessity of vertex-connectivity of the graph to be at least $2t + 1$

- ✱ If the vertex-connectivity is not $2t + 1$, then a sequence of "inconsistent" executions can be shown

Now, that shows the sufficiency proof. So, whatever we have shown here is the sufficiency proof here. Because we showed that if these two conditions are guaranteed then they are sufficient to design a perfectly secure BA protocol. Now we will argue about the necessity of the condition 1 and condition 2. That means, if we have an incomplete graph and if any of these two properties are violated.

If any of these two properties are violated, then we want to argue that we cannot design the perfectly secure BA protocol in the underlying graph. So, imagine that the first condition is violated; that means, instead of $n > 3t$ we have the condition $n \leq 3t$ and say my graph is having the network connectivity which is at least $2t + 1$. That means, the

second condition is not violated it is only the first condition which is violated, then the contradiction we get here is that if at all we have a perfectly secure BA protocol in an incomplete graph with the condition $n \leq 3t$.

Then the same protocol will also be a perfectly secure BA protocol in a complete graph with the condition $n \leq 3t$. But we know that in a complete graph we cannot have any perfectly secure BA protocol with the condition $n \leq 3t$; that means, the existence of π_{BA} which we assume to exist is wrong.

Now, what about the necessity of the second condition? Well, the proof is slightly involved here. So, I will not go into the exact proof, but the idea there is that if the vertex connectivity is not $2t + 1$, then we can show a sequence of inconsistent executions of the assumed Byzantine agreement protocol and we can compose them and then we can arrive at a contradiction that at least one of the properties of the assumed BA protocol is violated.

(Refer Slide Time: 26:46)



So, I will not be going through the complete proof, but if you are interested you can refer to the textbook by Nancy A. Lynch. With that I end this lecture.

Thank you.