

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

**Lecture - 08**  
**Efficient Protocols for Perfectly - Secure Byzantine Agreement: Part II**

Hello everyone, welcome to this lecture. So, we will continue our discussion regarding Efficient Protocols for Perfectly Secure Byzantine Agreement.

(Refer Slide Time: 00:33)

**Lecture Outline**

- Efficient protocol for perfectly-secure BA
- ❖ The phase-king based protocol with  $n > 3t$

Handwritten notes in red:

- $(n > 4t)$
- $t < \frac{n}{4}$
- $t \leq 24$
- $n = 100$
- $t \leq 24$
- $t < \frac{n}{3}$
- $t \leq 33$

Video inset of Prof. Ashish Choudhury in a yellow shirt.


In the last lecture, we had seen a protocol with  $n > 4t$ , we will follow the same phase king based protocol; but now we will see that how that protocol can be modified to even tolerate up to  $n/3$  number of corruptions. So, the previous protocol that we have discussed in the last lecture could tolerate only up to  $t$  less than  $n/4$  corruptions.

So, it is like saying that if you have say 100 parties, then the previous protocol will work as long as the maximum number of corruptions is up to 24, ok. Whereas, the protocol that we are going to discuss in today's lecture will work even if there are at most 33 corruptions; that means you can tolerate more number of corrupt participants in the protocol.

Of course, for that you have to do some more communication in the protocol, some more number of rounds of communication will be involved, ok.

(Refer Slide Time: 01:44)

## An Efficient BA Protocol with $n > 3t$

  
IIT DELHI

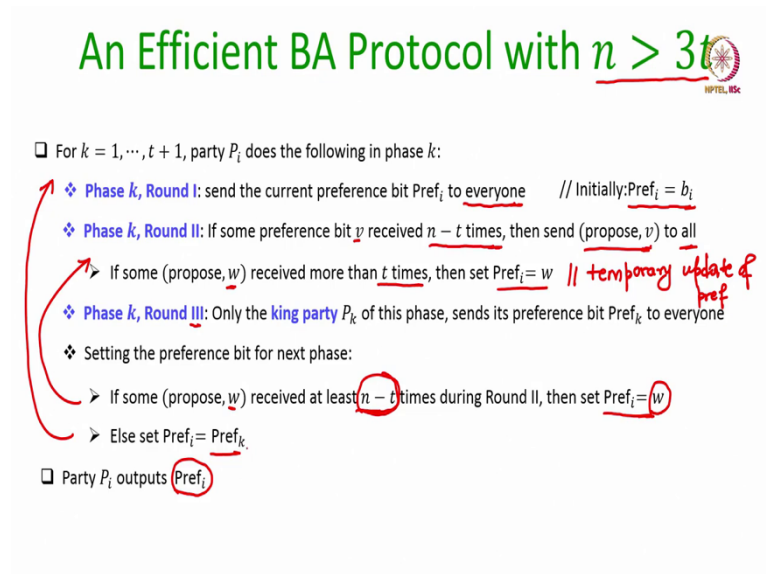
- $t + 1$  phases
  - ❖ Each phase consisting of three rounds (Earlier each phase had 2 rounds)
  - ❖ A designated "king" party in each phase (for simplicity:  $P_k$  will be the king for phase  $k$ )
- General idea:
  - ❖ In each phase  $k$ , the parties first try to find out whether all honest parties have the same bit
    - Yes: the parties stick to that bit in all subsequent phases, irrespective of king ... (1)
    - No: Take the help of king so that if king is honest then at the end of phase  $k$ , all honest parties have the same bit ... (2)
- Validity:
  - ❖ Guaranteed by (1)
- Consistency:
  - ❖ Guaranteed by (2) and the fact that there is at least one phase with an honest king.

So, the idea remains the same, as in the previous protocol, we will have  $t + 1$  phases; each phase will now have three rounds, earlier each phase had 2 rounds, but now in this modified protocol each phase will have three rounds and there will be a designated king in each phase.

For simplicity, we can imagine that party  $P_k$  will be the king for phase  $k$  and the general idea behind the protocol remains the same, we try to achieve the following two properties. We first try to find out in each phase whether all the honest parties have the same bit or not; if yes, then we stick to that bit in all the subsequent phases irrespective of whether the king for that phase is honest or corrupt.

Whereas, if in the phase  $k$  all the honest parties do not have the same bit, then we try to take the help of the king and see whether the king helps the parties to reach agreement, provided king is honest; of course if king of that phase is corrupt, then the help will be useless, ok. And as we have argued in the previous lecture, the validity property will be guaranteed; if the property number 1 that we are aiming for is satisfied, whereas consistency property will be guaranteed, because of the property number 2 which we are ensuring in the protocol and the fact that there will be at least one phase among the  $t+1$  phases, where the designated king will be honest, right.

(Refer Slide Time: 04:08)



So, let us see the protocol code now. Now, in each phase remember there are three rounds. So, again to begin with every party will first try to check whether they have the same preference bit for the current iteration or the current phase; where to begin with their preference bits are initialized to their respective inputs for the byzantine agreement protocol.

Now, recall in the previous protocol immediately at the end of first round every party sets its majority bit depending upon whether a value  $v$  has been received majority of times as the preference bit from the other parties; but now we are not going to do that, because we now want to design a protocol with  $n > 3t$ . So, the change that we are making here is the following.

Every party would have received preference bits from the other parties. So,  $P_i$  would have received  $n$  preference bits from  $n$  different parties including its own preference bit; remember when I say sent to everyone, it means sending to itself as well. So, among the  $n$  preference bits which  $P_i$  has received, it checks whether there is any preference bit  $v$  which has been received from  $n - t$  parties.

Again it is not necessary that this will be the case; but if there is some preference bit which has been received from  $n - t$  parties at the end of first round of the phase  $k$ , then during the second round party  $P_i$  sends a message that I am proposing a value  $v$  and it sends to everyone. Of course, if  $P_i$  is a corrupt party, it can send different versions of the proposed

message to different parties or it could be possible that party  $P_i$  has not received any bit  $n - t$  number of times, but still it is claiming that I am proposing this value  $v$ , ok.

So, corrupt parties can do anything, but if  $P_i$  is honest and if  $P_i$  has indeed received the preference bit  $v$  from at least  $n - t$  parties; then it sends a proposed message for that bit  $v$  to everyone else during the second round. Now, the preference bit is set for the time being as follows; party  $P_i$  will check that if some proposed message for the bit  $w$  is received more than  $t$  times, then set that bit as the preference bit, right.

So, this step is executed by everyone including the king as well, right. So, king also would have updated its preference bit if this condition would have been satisfied, otherwise it would have stick to its earlier preference bit. And now during the third round of the phase  $k$ , only the king party does the following; it sends its preference bit to everyone, again if the king is corrupt, it might send different versions of its preference bit to everyone else and so on.

And now the final assignment of the preference bit for the next phase is done as follows. So, you can imagine that this is a temporary update of preference bit; the final update of the preference bit at the end of the phase  $k$  happens as follows. So, when the temporary assignment of the preference bit was done,  $P_i$  checks whether that preference bit  $w$  or the message proposed  $w$  is received at least  $n - t$  times ok; that means if during this round II of the phase  $k$ , it is receiving proposed messages from many parties, right.

So, the temporary update was if it would have received the proposed message  $w$  at least  $t+1$  times from  $t + 1$  different parties, then it sends it sets the preference bit to  $w$ . But now the permanent update here is happening as follows, it checks whether the proposed message  $w$  for some bit  $w$  is received at least  $n - t$  number of times during the second round.

If that is the case, then set the preference bit finally for the next iteration to  $w$ ; otherwise replace the preference bit with whatever has been received from the king and then go to the next iteration. And as usual after  $t + 1$  phases are over, party  $P_i$  outputs whatever is its current preference bit. So, you see structure wise this protocol is very different from the previous protocol, which had only two rounds in each phase; first round everyone sends their current preference bit to everyone, assigned the majority depending upon whether a majority is there or not among the received preference bits.

And during the second round, king sends its preference bit to everyone else. And then the update of the preference bit happens depending upon whether party  $P_i$ 's own majority bit has been received more than  $n/2 + t$  number of times or not. But here we are doing something else; here we are using these proposed messages and depending upon the number of proposed messages, if a proposal for  $w$  is received at least  $n - t$  number of times then stick to that as the preference bit, otherwise change the preference bit to the preference bit which king is proposing.

(Refer Slide Time: 10:29)

### BA Protocol with $n > 3t$ : Analysis

□ For  $k = 1, \dots, t + 1$ , party  $P_i$  does the following in phase  $k$ :

- ❖ Phase  $k$ , Round I: send the current preference bit  $\text{Pref}_i$  to everyone // Initially:  $\text{Pref}_i = b_i$
- ❖ Phase  $k$ , Round II: If some preference bit  $v$  received  $n - t$  times, then send (propose,  $v$ ) to all
  - If some (propose,  $w$ ) received more than  $t$  times, then set  $\text{Pref}_i = w$
- ❖ Phase  $k$ , Round III: Only the king party  $P_k$  of this phase, sends its preference bit  $\text{Pref}_k$  to everyone
- ❖ Setting the preference bit for next phase:
  - If some (propose,  $w$ ) received at least  $n - t$  times during Round II, then set  $\text{Pref}_i = w$
  - Else set  $\text{Pref}_i = \text{Pref}_k$

□ Party  $P_i$  outputs  $\text{Pref}_i$

*Handwritten notes on slide:*

- Liveness: trivial**
- $3t + 3$  rounds** (with note: *Each round:  $\Delta$  clock*)
- $(3t + 3) \cdot \Delta$**
- $n - t > t$**

□ **Lemma:** If all honest parties have same preference bit  $b$  at the beginning of phase  $k$ , then they retain  $b$  as preference bit at the end of phase  $k$

- ❖ Each honest  $P_i$  receives at least  $n - t$  copies of  $b$  during the first round of phase  $k$
- ❖ Each honest  $P_i$  proposes  $b$  during the second round of phase  $k$
- ❖ Each honest  $P_i$  receives (propose,  $b$ ) at least  $n - t$  times during the second round of phase  $k$
- ❖ The value from king **not considered** for setting the preference bit

*Handwritten notes on Lemma:*

- At least  $n - t$  honest parties**
- Implies validity**

So, now again we will do the analysis here, the liveness analysis is trivial here again. Why it is trivial? Because there are  $3t + 3$  number of rounds as there are three rounds involved in each phase; each round will be over within  $\Delta$  clock cycles, where  $\Delta$  is publicly known.

So, after time  $(3t + 3)\Delta$ , the protocol will produce an output for every honest party. So, it will not be the case that the honest parties keep on running the protocol forever. Now, we will prove the validity property first and again for that we prove this lemma that if all the honest parties have the same preference bit at the beginning of any phase  $k$ ; then they will stick to that bit as their preference bit at the end of the phase  $k$  as well, ok.

So, if all the honest parties have the same preference bit at the beginning of the phase  $k$ ; say it is the value  $b$ , then every honest party would have sent the value  $b$  as its preference bit to everyone else and there are at least  $n - t$  honest parties. That means, at the end of round I each  $P_i$  each honest  $P_i$  will receive  $n - t$  copies of the bit  $b$ . Of course, corrupt parties may say

that ok, my preference bit is  $\bar{b}$ . But clearly  $n - t$  is greater than  $t$  right, because we are having the condition  $n > 3t$ . As a result of this during the second round of phase  $k$ , each honest  $P_i$  will say that ok I am proposing the bit  $b$ ; because it has received  $n - t$  copies of the preference bit  $b$  during the first round of the phase  $k$ , right.

Again, a corrupt  $P_i$  might propose  $\bar{b}$ , but we have  $n - t > t$ . And we do not care whether the king is honest or corrupt, king might do anything; when it comes to finally, update the preference bit for the  $k$ th phase, each honest  $P_i$  will see that it has received the proposal for this bit value  $b$  at least  $n - t$  number of times from  $n - t$  honest parties during the second round, right.

So, it will stick to that bit; that means the value from the king will not be considered at all to do the final update of the preference bit. So, that that proves the lemma and this lemma immediately implies the validity property; because if at the beginning of the protocol itself all the honest parties have the same input bit  $b$ , that means their individual preference bit is assigned the value  $b$ .

Then this lemma states that at the end of the first phase, they will stick to the value  $b$  as the output for the preference bit; then in the second phase they will start with their preference bit being  $b$  and at the end of the second phase, they will stick to their preference bit being  $b$  only.

And like that in all the  $t + 1$  phases, they start the phase with preference bit  $b$  and end the phase with preference bit  $b$  and output  $b$  and the value from the king will not be considered in any of the phases and that proves the validity property, right.

(Refer Slide Time: 14:38)

## BA Protocol with $n > 3t$ : Analysis

□ For  $k = 1, \dots, t + 1$ , party  $P_i$  does the following in phase  $k$ :

- ❖ Phase  $k$ , Round I: send the current preference bit  $\text{Pref}_i$  to everyone // Initially:  $\text{Pref}_i = b_i$
- ❖ Phase  $k$ , Round II: If some preference bit  $v$  received  $n - t$  times, then send (propose,  $v$ ) to all
  - If some (propose,  $w$ ) received more than  $t$  times, then set  $\text{Pref}_i = w$
- ❖ Phase  $k$ , Round III: Only the king party  $P_k$  of this phase, sends its preference bit  $\text{Pref}_k$  to everyone
- ❖ Setting the preference bit for next phase:
  - If some (propose,  $w$ ) received at least  $n - t$  times during Round II, then set  $\text{Pref}_i = w$
  - Else set  $\text{Pref}_i = \text{Pref}_k$

□ Party  $P_i$  outputs  $\text{Pref}_i$

$n=4 \quad t=1$

Every phase

□ Lemma: If any honest party  $P_i$  proposes  $v$ , then no other honest party  $P_j$  proposes  $w$ , where  $w \neq v$

- ❖  $P_i$  receives  $v$  from  $n - t$  parties  $\mathcal{A}$
- ❖ At least  $n - 2t$  of them are honest
- ❖ At most  $t$  of them are corrupt

Phase  $k$

- ❖  $P_j$  may receive  $w \neq v$  from at most  $2t$  parties
  - $t$  corrupt parties in  $\mathcal{A}$  and  $t$  honest parties outside  $\mathcal{A}$
- ❖  $2t < n - t$  holds

Now, we prove the consistency property and it is slightly tricky compared to the previous protocol. So, we first take the help of this helping lemma, which claims that if in any phase  $P_i$  proposes some value  $v$ ; then no other honest party will propose a value  $w$  bit  $w$ , where  $w$  is different from  $v$  and it holds in any phase provided  $n > 3t$ . Let us see why. So, imagine some phase  $k$  and consider two different honest parties; an honest party  $P_i$  and an honest party  $P_j$ . Suppose  $P_i$  has proposed the value  $v$ . Now, why it has proposed the value  $v$ ?

So, let us see the condition under which the party  $P_i$  would have proposed the value  $v$ ; it would have proposed the value  $v$  only if it has received  $n - t$  copies of the preference bit  $v$  from  $n - t$  parties.

So, let  $A$  be the set of parties  $n - t$  parties from whom party  $P_i$  has received  $v$  as the preference bit; then it is not necessary that all the parties in  $A$  are honest, some of them could be corrupt as well, who might have unnecessarily reported to  $P_i$  that their preference bit is  $v$ . But what we know is that, there are at least  $n - 2t$  parties in the set  $A$ , ok.

Now, let us see that how many copies of the bit  $w$ , where  $w$  is different from  $v$  will be received as the preference bit from the other honest party  $P_j$ , right. So,  $P_j$  is another honest party different from  $P_i$  and it would be also receiving  $n$  preference bits during the same round, during round number I from different parties, right. Let us see among those  $n$  copies; how many preference bits will be for  $w$ , where  $w$  is different from  $v$ .

The claim is that, there could be at most  $2t$  parties who might send their preference bit as  $w$ . Who could be those  $2t$  parties? They could be the  $t$  corrupt parties in  $A$  right. So, in this diagram I am assuming  $n$  is equal to 4 and  $t$  is equal to 1. So, there could be up to  $t$  corrupt parties. So, this is one corrupt party, which might send  $v$  as its preference bit to  $P_i$ , but  $w$  as its preference bit to  $P_j$ .

And there could be  $t$  honest parties outside the set  $A$  who would have sent  $w$  as the preference bit to  $P_i$  and since they are honest, they will stick to their preference bit as  $w$  and send  $w$  as their preference bit even to the party  $P_j$ , ok. So, in any for the case for  $n$  is equal to 4 and  $t$  is equal to 1 what I am stating here is that; in order for  $P_i$  to send a proposal for  $v$ , it should have received three copies of  $v$  as preference bit. Among those three copies, one copy might be coming from a corrupt party, who might send  $w$  now as the preference bit to  $P_j$ .

And how many parties we have outside, overall one more party. So, overall this one corrupt party in the set  $A$  and the outside party might send  $w$  as their preference bits. So, these are total 2, but 2 is not the number using which this party  $P_j$  would have proposed the value  $w$ , it needs  $n - t$  copies of the value  $w$ .

Now, the thing is that  $n - t$  is always strictly greater than the value  $2t$ ; that means even though the proposal for  $w$  is coming to this party  $P_j$ , it will be receiving only  $2t$  number of proposal,  $2t$  number of copies of  $w$  as the preference bit and  $2t$  is strictly less than  $n - t$ .

You might be wondering that why can not the honest parties in the set  $A$  send a proposal for  $w$  to  $P_j$ ; well the honest parties will send only one version of their preference bit, they cannot behave like corrupt parties and send  $v$  as their preference bit to  $P_i$  and  $w$  as their preference bit to  $P_j$ , honest parties will not do that, they will stick to their preference bit to be  $v$  towards everyone. It is only the corrupt parties in the set  $A$  who may change; to one subset of honest parties they may say that ok, their preference bit is  $v$  and to another set of honest parties it may say that, their preference bit is  $w$ .

But even in that case the corrupt parties copies of the preference bit plus the honest parties outside  $A$  their copies of the preference bit does not sum up to the required quantity  $n - t$  in order that the party  $P_j$  sends a proposal for  $v$  ok; that means if at all  $P_j$  sends a proposal, it could be only for  $v$ , it cannot be for any other value different from  $v$ .



(Refer Slide Time: 20:57)

## BA Protocol with $n > 3t$ : Analysis

□ For  $k = 1, \dots, t+1$ , party  $P_i$  does the following in phase  $k$ :

- ❖ Phase  $k$ , Round I: send the current preference bit  $\text{Pref}_i$  to everyone // Initially:  $\text{Pref}_i = b_i$
- ❖ Phase  $k$ , Round II: If some preference bit  $v$  received  $n - t$  times, then send (propose,  $v$ ) to all
  - If some (propose,  $w$ ) received more than  $t$  times, then set  $\text{Pref}_i = w$
- ❖ Phase  $k$ , Round III: Only the king party  $P_k$  of this phase, sends its preference bit  $\text{Pref}_k$  to everyone
- ❖ Setting the preference bit for next phase:
  - If some (propose,  $w$ ) received at least  $n - t$  times during Round II, then set  $\text{Pref}_i = w$  Step 1 ✓
  - Else set  $\text{Pref}_i = \text{Pref}_k$  Step 2 ✓
- Party  $P_i$  outputs  $\text{Pref}_i$  consistency

□ Lemma: If the king  $P_k$  of phase  $k$  is honest, then all honest parties have the same preference bit at the end of phase  $k$

- ❖ Case I: If each honest  $P_i$  set its preference bit to king's preference bit  $\text{Pref}_k$  — lemma holds trivially as  $P_k$  has  $\text{Pref}_k$  identically
- ❖ Case II: If some honest  $P_i$  does not set its preference bit to king's preference bit  $\text{Pref}_k$ 
  - $P_i$  receives some (propose,  $w$ ) at least  $n - t$  times during Round II  $n - t - t$ 
    - Every other honest party including king  $P_k$  receives (propose,  $w$ ) at least  $n - 2t > t$  times during Round II
    - From previous lemma, no honest party proposes any  $v \neq w$

Every other honest party including king  $P_k$ , sets its preference bit to  $w$  during Round II

That is a very important lemma. So, final lemma which will help us to prove the consistency property here is that, if the king of a phase is honest; then at the end of that phase, all the honest parties will reach agreement on their preference bit, they will have the same preference bit. So, again let us look into the code and there are two steps in each phase ok, where every party finally updates its preference bit.

So, consider the case one when no honest party executes the step number 1 to set their preference bit; that means during the protocol execution, the initial configuration of the inputs of the parties and the messages exchanged by the parties are such that, every party  $P_i$  ends up executing the else statement to update their respective preference bits. That means, they are setting the final preference bit to whatever preference bit, temporary preference bit king has communicated to everyone.

And since we are assuming that the king for that phase is honest, the king will be honestly communicating an identical copy of its preference bit to everyone and that will be taken as the output preference bit for that phase by every honest party and that shows the lemma holds. But it could be possible that there is one subset of parties who executes step number 1 to update their preference bit finally; while there is another subset of honest parties who might be executing step number 2 to finally update their preference bit.

So, this could happen right this this can happen depending upon the initial configuration of the values of the parties and what messages they would have received; it is not necessary that

if the king is honest, everyone will be only using this else statement or everyone is using only the if statement to finally update their preference bit. It could happen that one subset of parties using the step number 1, one subset of parties execute this step number 2 depending upon which condition is satisfied in their respective case, ok.

We will prove that even if that happens at the end of the phase  $k$  all the honest parties will have the same preference bit. So, let us see why. So, imagine there is a party  $P_i$  who does not consider king's preference bit to update its preference bit; that means it executes the if statement to update its preference bit, that means it has received a proposal for the value  $w$  at least  $n - t$  times during round II, right.

That is why it has executed step number 1 to update its preference bit finally. Now, among these  $n - t$  copies of the proposal for the value  $w$ ; how many copies will be received by  $P_k$ ? The claim is at least  $n - 2t$  copies of this proposed  $w$  message would have been received by the honest king as well; this is because among these  $n - t$  copies of the proposal message for  $w$ , there could be  $t$  copies coming from the corrupt parties, those corrupt parties might send a proposal for  $\bar{w}$  to the king.

But there are at least  $n - 2t$  number of honest parties among the set of parties who have sent the proposal message to  $P_i$  for  $w$ , who would be sending the proposal message for  $w$  even to the king  $P_k$ . And since we are operating under the condition  $n > 3t$ ,  $n - 2t$  will be strictly greater than  $t$ ; that means at the end of round II during phase  $k$ ,  $P_k$  would have received the proposal for  $w$  from at least  $t + 1$  parties, ok.

Moreover from the previous lemma we also know that the king  $P_k$  would never receive a proposal for any other message different from  $w$ , if it is receiving a proposal for  $w$ , right. So, all in all what is happening is that, since  $P_k$  has received a proposal for  $w$  at least  $t + 1$  number of times during the round II; it would have temporarily set its preference bit at the end of the second round to the same value  $w$ .

Now, that means that if we consider the other subset of parties who are executing the step number 2 or the else statement to finally update their preference bit; that means they are changing their preference bit to the preference bit which king is communicating, then they are also setting their final preference bit for this phase to  $w$  only, which is the same as the preference bit which has been set by the party  $P_i$ .

So, that shows that it does not matter whether the parties execute step number 1 or they execute step number 2 to update their final preference bit for the  $k$ th phase, they will end up updating it to the same value. And that shows that the consistency property is achieved in the protocol; because it is guaranteed that there will be at least one phase among the  $t + 1$  phases, where the designated king is guaranteed to be honest, ok.

(Refer Slide Time: 27:31)

### BA Protocol with $n > 3t$ : Analysis

□ For  $k = 1, \dots, t + 1$ , party  $P_i$  does the following in phase  $k$ :

- ❖ Phase  $k$ , Round I: send the current preference bit  $\text{Pref}_i$  to everyone // Initially:  $\text{Pref}_i = b_i$
- ❖ Phase  $k$ , Round II: If some preference bit  $v$  received  $n - t$  times, then send (propose,  $v$ ) to all
  - If some (propose,  $w$ ) received more than  $t$  times, then set  $\text{Pref}_i = w$
- ❖ Phase  $k$ , Round III: Only the king party  $P_k$  of this phase, sends its preference bit  $\text{Pref}_k$  to everyone
- ❖ Setting the preference bit for next phase:
  - If some (propose,  $w$ ) received at least  $n - t$  times during Round II, then set  $\text{Pref}_i = w$
  - Else set  $\text{Pref}_i = \text{Pref}_k$

□ Party  $P_i$  outputs  $\text{Pref}_i$

In each phase:  $O(n^2)$  bits

EIG

Phase-King I

$t < n/3$

Fault tolerance (resilience)

$t < n/3$

Rounds

$t + 1$

Comm/Com

Exponential

more than EIG/previous phase-king protocol

□ Round Complexity:

- ❖  $3t + 3$  rounds

□ Communication Complexity:

- ❖  $O(n^2)$  bits

Let us do the complexity analysis; the total number of communication rounds will be now  $3t + 3$ , which is more than EIG and the previous phase king protocol. And the communication remains the same, because we have  $t + 1$  phases and in each phase,  $O(n^2)$  bits are communicated, so total  $n^3$  bits. So, you can see that the number of rounds here is more; but the complexity is polynomial and we can design a protocol with the same resilience or the same fault tolerance as the EIG.

So, now we have three parameters here. So, you have fault tolerance or the resilience, namely the number of faults which can be tolerated by the protocol and rounds, number of rounds and communication and computation complexity, these are the three measures based on which we can compare the three BA protocols which we have considered till now. In EIG protocol, the resilience was  $t < n/3$ , number of rounds was  $t + 1$  very good, but the computation and communication was exponential.

In the phase king protocol number 1 which we had discussed in the previous lecture, the resilience was bad; we can tolerate less number of faults compared to EIG protocol, but the number of rounds was less.

(Refer Slide Time: 29:44)

### BA Protocol with $n > 3t$ : Analysis

□ For  $k = 1, \dots, t + 1$ , party  $P_i$  does the following in phase  $k$ :

- ❖ Phase  $k$ , Round I: send the current preference bit  $\text{Pref}_i$  to everyone // Initially:  $\text{Pref}_i = b_i$
- ❖ Phase  $k$ , Round II: If some preference bit  $v$  received  $n - t$  times, then send (propose,  $v$ ) to all
  - If some (propose,  $w$ ) received more than  $t$  times, then set  $\text{Pref}_i = w$
- ❖ Phase  $k$ , Round III: Only the king party  $P_k$  of this phase, sends its preference bit  $\text{Pref}_k$  to everyone
- ❖ Setting the preference bit for next phase:
  - If some (propose,  $w$ ) received at least  $n - t$  times during Round II, then set  $\text{Pref}_i = w$
  - Else set  $\text{Pref}_i = \text{Pref}_k$

□ Party  $P_i$  outputs  $\text{Pref}_i$

In each phase:  $O(n^3)$  bits

	Fault tolerance (resilience)	Rounds	Comm/Com con
EIG	$t < n/3$	$t+1$	Exponential
Phase-King I	$t < n/4$	$2t+2$	polynomial
Phase-King II	$t < \frac{n}{3}$	$3t+3$	polynomial

□ Round Complexity:  $3t + 3$  rounds more

□ Communication Complexity:  $O(n^3)$  bits

But the good part was that the computation and communication was polynomial, so it was an efficient protocol.

And the current protocol that we have discussed is phase king II; it has now the same resilience as the EIG protocol, but it requires more communication, a much more interaction and the computation and communication is efficient.

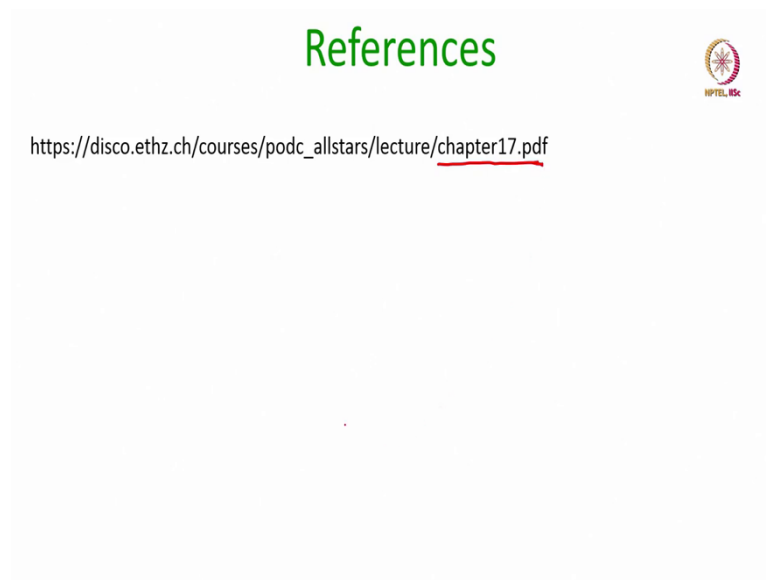
So, you can see now you have a tradeoff; if you do not want a protocol where the parties are support supposed to interact for many number of rounds, because every round of communication means parties have to send messages to the other parties through some channel, over the channel right; that means they have to invoke some SSL protocol, TLS protocol to send those messages securely to the other parties, ok.

So, a protocol with  $t + 1$  number of rounds will require the parties to open the SSL connection less number of times compared to a protocol which has  $2t + 2$  number of rounds compared to a protocol which has  $3t + 3$  number of rounds. So, if it is the number of rounds which is the critical resource for you which you do not want to spend, where you do not want to spend more; then EIG protocol is preferred, but for that we have to spend a huge price.

Even though the number of times the parties have to communicate is less, the amount of communication they have to do during those rounds is enormously large compared to the other two protocols. Similarly, if your criteria is the number of bad parties or the corrupt parties which you want to tolerate during the protocol execution; then again EIG and phase king II protocols are preferred compared to phase king I; because phase king I protocol can tolerate only up to  $n/4$  corrupt components in the system.

Whereas, EIG protocol and phase king II protocol can tolerate more number of corrupt components in the system.

(Refer Slide Time: 32:05)



So, the protocol that I discussed in today's lecture it is not available in any standard textbook; I have taken the protocol code and its analysis from this reference.

Thank you.