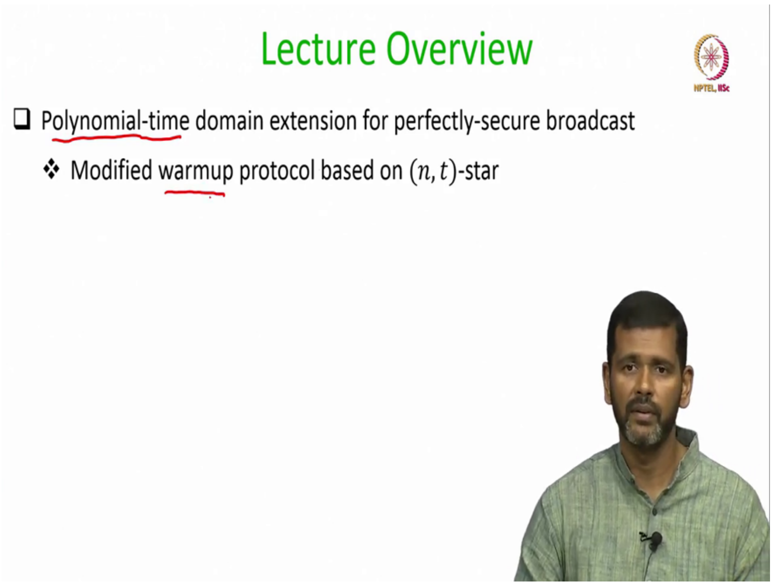
Secure Computation: Part II
Prof. Ashish Choudhury
Department of Computer Science and Engineering
Indian Institute of Science, Bengaluru

Lecture - 30
Domain Extension for Perfectly-Secure Broadcast Based on RS Error-Correcting Codes: IV

Hello everyone. Welcome to this lecture. So, this will be now the last part of our Domain Extension Protocol for Perfectly-Secure Broadcast Based on Reed-Solomon Error-Correcting Codes.

(Refer Slide Time: 00:33)



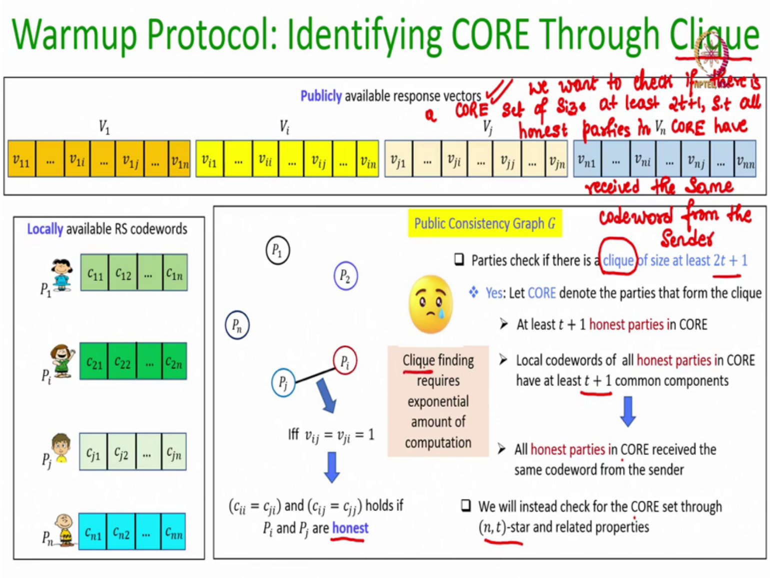
Lecture Overview

❑ Polynomial-time domain extension for perfectly-secure broadcast

❖ Modified warmup protocol based on (n, t)-star

Now, we will see the actual polynomial time protocol. And to get the polynomial time protocol what we just need to do is, we just need to modify the warmup protocol. Rest of the things remains the same as it was earlier for the exponential time protocol, ok.

(Refer Slide Time: 00:51)



Warmup Protocol: Identifying CORE Through Clique

Publicly available response vectors

we want to check if there is a CORE set of size at least 2t+1, s.t all honest parties in CORE have received the same codeword from the Sender

Locally available RS codewords

Public Consistency Graph G

□ Parties check if there is a clique of size at least $2t+1$

- Yes: Let CORE denote the parties that form the clique
  - At least $t+1$ honest parties in CORE
  - Local codewords of all honest parties in CORE have at least $t+1$ common components
  - All honest parties in CORE received the same codeword from the sender

Clique finding requires exponential amount of computation

Iff $v_{ij} = v_{ji} = 1$

$(c_{ii} = c_{ji})$ and $(c_{ij} = c_{jj})$ holds if P_i and P_j are honest

□ We will instead check for the CORE set through (n, t)-star and related properties

V_1
V_i
V_j
V_n
v_{11} … v_{1i} … v_{1j} … v_{1n}
v_{i1} … v_{ii} … v_{ij} … v_{in}
v_{j1} … v_{ji} … v_{jj} … v_{jn}
v_{n1} … v_{ni} … v_{nj} … v_{nn}
P_1 c_{11} c_{12} … c_{1n}
P_i c_{21} c_{22} … c_{2n}
P_j c_{j1} c_{j2} … c_{jn}
P_n c_{n1} c_{n2} … c_{nn}
P_1
P_2
P_n
P_i
P_j

So, let us quickly recall the warmup protocol and where exactly it requires the parties to perform exponential amount of computation. So, in that warmup protocol, the part, each party has prepared its response vector and each party also would have received a local code word from the sender.

They do not know whether the sender has distributed the same code word to all the honest parties or not. To identify the same, each pair of parties would have exchanged a constant number of supposedly common points on their respective local code words and based on that exchange they have prepared these response vectors and broadcasted them using instances of the bit broadcast protocol.

Now, based on the publicly available response vectors, the parties would have prepared a consistency graph, where an edge between the nodes representing the parties Pi and Pj implies that the ith component in the ith parties code word is the same as the ith component in the jth parties code word. And the jth component of the jth parties code word is the same as the jth component in the ith parties code word, provided both P i and P j are honest. Because the edge would have been added only if Pi is not in conflict with P j in the response vector, and Pj is not in conflict with Pi in its own response vector, ok.

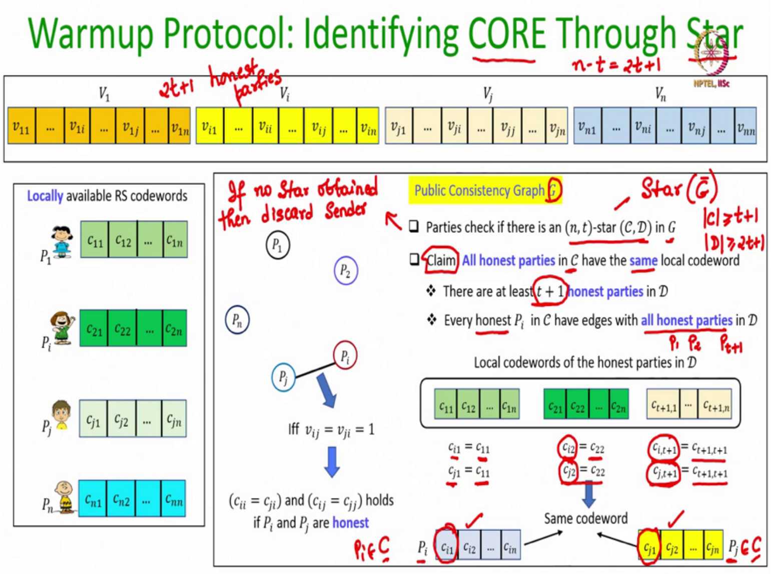
And then in the earlier protocol, earlier warmup protocol, the parties would have checked if there exists a clique of size at least 2t + 1 which should be there in the consistency graph if the sender has behaved honestly. If it is not present, if the clique of size 2t + 1 is not present that definitely implies that the sender is corrupt. So, simply stop the protocol there and take some default message as the output on the behalf of the sender.

But if a CORE is obtained, then at least t + 1 honest parties are present in the CORE. Then, we have proved that the local code words of all honest parties in the CORE are the same because since the honest parties in the CORE constitute a clique. It means that pair wise every pair of honest parties have the same common components in their respective local code words and based on that we can say that all honest parties in CORE have at least t plus 1 common components.

And now using the properties of Reed-Solomon code words, we know that all honest parties in CORE have the same code word received from the sender. And then there were some more properties which were there which are not required at the moment. So, at this particular step in the warmup protocol requires the parties to perform exponential amount of computation to check whether in their respective consistency graph a clique of size at least 2t + 1 is present or not.

Now, instead of checking whether a clique of size at least 2t + 1 is present or not, we will check for a CORE set of parties by checking for an n, t star and using the related properties.

(Refer Slide Time: 04:38)



Warmup Protocol: Identifying CORE Through Star

Public Consistency Graph G

❑ Parties check if there is an (n, t)-star $(\mathcal{C}, \mathcal{D})$ in G

❑ Claim: All honest parties in $\mathcal{C}$ have the same local codeword

- There are at least $t + 1$ honest parties in $\mathcal{D}$
- Every honest P_i in $\mathcal{C}$ have edges with all honest parties in $\mathcal{D}$

Local codewords of the honest parties in $\mathcal{D}$

Locally available RS codewords

Iff $v_{ij} = v_{ji} = 1$

$(c_{ii} = c_{ji})$ and $(c_{ij} = c_{jj})$ holds if P_i and P_j are honest

Same codeword

NPTEL, IISc
$n-t = 2t+1$
$2t+1$ honest parties
V_1
V_i
V_j
V_n
v_{11} ... v_{1i} ... v_{1j} ... v_{1n}
v_{i1} ... v_{ii} ... v_{ij} ... v_{in}
v_{j1} ... v_{ji} ... v_{jj} ... v_{jn}
v_{n1} ... v_{ni} ... v_{nj} ... v_{nn}
If no Star obtained then discard Sender
Star $(\bar{G})$
$|\mathcal{C}| \geq t+1$
$|\mathcal{D}| \geq 2t+1$
P_1 P_2 P_{t+1}
P_1
c_{11} c_{12} ... c_{1n}
P_i
c_{21} c_{22} ... c_{2n}
P_j
c_{j1} c_{j2} ... c_{jn}
P_n
c_{n1} c_{n2} ... c_{nn}
P_1
P_2
P_n
P_i
P_j
c_{11} c_{12} ... c_{1n}
c_{21} c_{22} ... c_{2n}
$c_{t+1,1}$ ... $c_{t+1,n}$
$c_{i1} = c_{11}$
$c_{j1} = c_{11}$
$c_{i2} = c_{22}$
$c_{j2} = c_{22}$
$c_{i,t+1} = c_{t+1,t+1}$
$c_{j,t+1} = c_{t+1,t+1}$
$P_i \in \mathcal{C}$
P_i c_{i1} c_{i2} ... c_{in}
c_{j1} c_{j2} ... c_{jn} $P_j \in \mathcal{C}$
✓
✓

So, this is now the modified protocol which is a polynomial time protocol where we are now trying to identify the CORE set of parties through an n, t star and related properties. Just to recall what we want here is to check is if there is a CORE set of parties of size at least 2t + 1, such that all honest parties in CORE have received the same code word from the sender. That is our goal. And we want to check this in polynomial time.

If I do not put this constraint that you should be able to check CORE in polynomial time then we already have a method, check for a clique which will work, but that requires exponential amount of computation. So, this is now the modified method.

The parties will make public the response vectors, based on that the consistency graph will be constructed, that part remains the same. But, now instead of checking whether a clique of size 2t + 1 is present in the consistency graph, parties check if an n, t star is present in the graph.

For this the parties use the star finding algorithm, where the input will be now G complement. The output of the star finding algorithm will be either a star in the graph G or the message star is not present. Now, if the message star not present, so if no star obtained then discard the sender. That means, sender is definitely corrupt and terminate the protocol with some default output on the behalf of the sender.

The claim is that an honest sender will never be discarded because if the sender has behaved honestly. That means, it has given the same code word to all the honest parties in the system and there are at least 2t + 1 honest parties in the system, and those 2t + 1 honest parties will constitute a clique in the graph G.

And remember, if in the graph G a clique of size at least n - t is there, and n - t will be 2t + 1, because we are working in the setting where n is 3t + 1. So, if n - t is 2t + 1, and if a clique of size 2t + 1 is there in the consistency graph which is guaranteed to be there for an honest sender, then the output of the star finding algorithm will always be a star. It will never be the message star not present.

So, an honest sender is guaranteed not to be discarded. If at all the sender is discarded, definitely he is corrupt because he has not distributed consistent or common code word to sufficiently many number of honest parties. So, we will not consider that case. We will consider the case when a star is obtained, and from that point onward how to take the protocol forward.

So, if a star is obtained then the first claim is that all honest parties in C, have the same local code word. This is because there are at least t + 1 honest parties in D. Why? Because if a star is obtained then the cardinality of C will be at least t + 1 because it will be of size n - 2t and n - 2t will be t + 1. And the cardinality of D will be at least 2t + 1, because the cardinality of D will be n - t.

Now, if there are 2t + 1 total number of parties in D, then among them t could be corrupt whose edges with the honest parties in C may not be trusted. Because even though they are in conflict with the honest parties in C, they may simply say I am fine with honest parties in C. But there are at least t + 1 honest parties in D whose edges with the honest parties in C are genuine. That means, those edges represent that among those pair of parties, there is no conflict.

So, there are at least t + 1 honest parties in D and due to the property of the n, t star, every honest party in C is guaranteed to have an edge with all the honest parties in D. Now, let us see what is the consequence of this. For simplicity, for the purpose of demonstration assume that the honest parties in D are P1, P2, Pt+1. Since, D is guaranteed to have at least t + 1 honest parties, for the purpose of demonstration I am taking P1, P2, Pt+1 to be those honest parties, but that need not be always the case. It could be any t + 1 honest parties.

Now, those t + 1 honest parties have received their code words from the sender. And, now consider an honest party Pi who is present in C. So, this Pi is present in C and it is honest. And this Pi has an edge with P1 in the star. Now, why it has an edge with P1? It has an edge with P1 because they have exchanged their supposedly common components and they found that ci1 is the same as c11. And that is why P i would have said in its response vector that I am fine with P1, and P 1 would have said in its response vector that I am fine with Pi, after checking this condition is satisfied.

In the same way, Pi has an edge with P2. That means, during the pair wise consistency test Pi would have found that ci2 is the same as c22 and it would have said in its response vector that I am fine with P2. And in the same way, P2 would have found that c22 is the same as ci2, and P2 would have said in its response vector that P2 is fine with Pi. And like that, since, Pi has an edge with Pt+1; that means, the component $c_{i,t+1}$ is same as the component $c_{t+1,t+1}$.

Now, let us consider another honest party Pj who is also present in C. Now, the party Pj also has an edge with all these parties P1, P2, Pt+1 in the star graph, in the star structure. Now, Pj has an edge with P1, which implies that c_{j1} is the same as c11.

In the same way, Pj has an edge with P2 that implies component wise the second component in jth parties code word is the same as the second component of second parties code word. And like that since Pj has an edge with the party Pt+1, that means, the t +1th component in the jth party's code word is the same as the t+1th component in the t+1th party's code word.

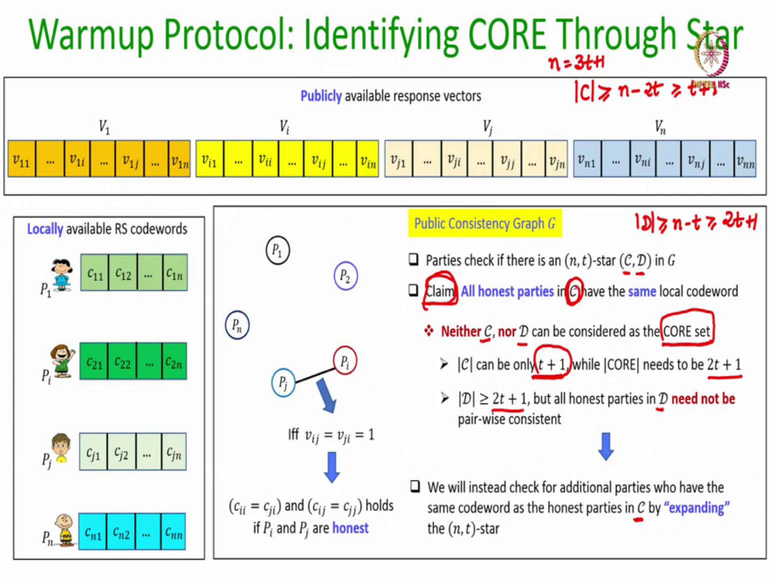
Now, based on these things we can simply say that c11 is the same as ci1 and c 11 is the same as c j1; that means, the first component of the ith party's code word and the jth party's code word are the same. The second component in the ith party's code word and the second component in the jth party's code word, they are also the same. And like that the t + 1th component in the ith party's code word and the t + 1th component in the jth party's code word, they are the same.

And recall, we have proved that if we have a Reed-Solomon code word corresponding to a message encoding polynomial of t degree where the two code words are having same t+1 or more number of components, then the two Reed-Solomon code words correspond to the same message.

So, now P i has received a code word from the sender where the message encoding polynomial has degree t and P j also has received a code word from the sender, a local code word, corresponding to a message encoding polynomial of t degree.

Both of them are members of the set C and we have proved that they have t + 1 or more number of the same components, that automatically implies that P i's and P j's code words are the same. And that proves this claim. All honest parties in C have the same local code word. So, that is an important claim.

(Refer Slide Time: 15:48)



Warmup Protocol: Identifying CORE Through Star

$n = 3t+1$

$|C| \geq n - 2t \geq t+1$

Publicly available response vectors

V_1							V_i							V_j							V_n						
v_{11}	...	v_{1i}	...	v_{1j}	...	v_{1n}	v_{i1}	...	v_{ii}	...	v_{ij}	...	v_{in}	v_{j1}	...	v_{ji}	...	v_{jj}	...	v_{jn}	v_{n1}	...	v_{ni}	...	v_{nj}	...	v_{nn}

Locally available RS codewords

P_1	c_{11}	c_{12}	...	c_{1n}
P_i	c_{21}	c_{22}	...	c_{2n}
P_j	c_{j1}	c_{j2}	...	c_{jn}
P_n	c_{n1}	c_{n2}	...	c_{nn}

Public Consistency Graph G

$|D| \geq n - t \geq 2t+1$

Iff $v_{ij} = v_{ji} = 1$

$(c_{ii} = c_{ji})$ and $(c_{ij} = c_{jj})$ holds if P_i and P_j are honest

❑ Parties check if there is an (n, t)-star $(\mathcal{C}, \mathcal{D})$ in G

❑ Claim: All honest parties in $\mathcal{C}$ have the same local codeword

❖ Neither $\mathcal{C}$, nor $\mathcal{D}$ can be considered as the CORE set

➢ $|\mathcal{C}|$ can be only $t + 1$, while |CORE| needs to be $2t + 1$

➢ $|\mathcal{D}| \geq 2t + 1$, but all honest parties in $\mathcal{D}$ need not be pair-wise consistent

❑ We will instead check for additional parties who have the same codeword as the honest parties in $\mathcal{C}$ by "expanding" the (n, t)-star

However, neither the subset C nor the subset D can be considered as the CORE set. Recall, we are trying to identify a CORE set of parties of size at least 2t + 1, where it is guaranteed that all the honest parties in that CORE set have the same local code word. What we have argued till now is that all the honest parties in the C component of a star, if at all a star is obtained, have the same local code word. But, now the fact is that neither the subset C nor the subset D can be considered as the potential CORE set.

Why C cannot be considered as the potential CORE set? Because the cardinality of C can be only t + 1 in the worst case. Recall that the cardinality of C is at least n - 2t and if n = 3t + 1 then n - 2t is guaranteed to be at least t +1.

Well, at least t plus 1 does not mean that there could be more than, there is necessarily more than t plus 1 parties in C. It could be possible that we have a consistency graph where a clique is present in such a way that the C component of a star has exactly t plus 1 parties. That could be possible.

If that happens, then the C component of the star cannot be considered as the CORE set because we require a CORE set of parties to be of size at least 2t + 1, which will be used later in the remaining stages of the warmup protocol.

Now, you might be wondering that, ok it is fine, C component cannot be considered as the potential CORE. Why cannot we work with the D component, because the D component of the star is guaranteed to have a cardinality of n - t which is at least 2t + 1. Well, size wise the D component satisfies the requirement of the CORE set. But all honest parties in D need not be pairwise consistent.

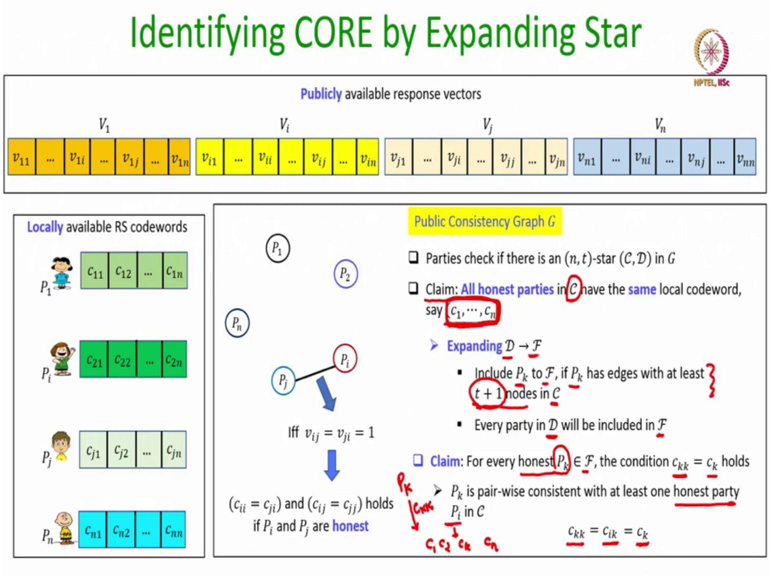
Because, recall that in the definition of n, t star, there is no compulsion that all the parties in D should constitute a clique. There is a guarantee that every node in C has an edge with every node in D, but it is not guaranteed that every node in D has an edge with every node in D.

That means, even though the D component of the star will have 2t + 1 parties, it is not necessary that all honest parties in D have the same common code word received from the sender because they need not be pair wise consistent among each other, right.

So, now what we will do is, it we will instead check for additional parties who have the same local code word as the honest parties in C, right. So, remember the claim. The claim that we have proved is all honest parties in C have the same local code word. But the problem right now is that we do not have sufficient number of honest parties in C. We want a CORE set where CORE should have at least t plus 1 honest parties, but we might be stuck with a C whose cardinality is exactly t plus 1.

So, what we would like to now do is we will check instead for additional parties who are right now not members of C, by trying to expand the C set to check whether we can include additional parties in this expanded C set who are also guaranteed to have the same code word as the honest parties in C.

(Refer Slide Time: 20:09)



Identifying CORE by Expanding Star

Publicly available response vectors

V_1 V_i V_j V_n

v_{11} … v_{1i} … v_{1j} … v_{1n} | v_{i1} … v_{ii} … v_{ij} … v_{in} | v_{j1} … v_{ji} … v_{jj} … v_{jn} | v_{n1} … v_{ni} … v_{nj} … v_{nn}

Locally available RS codewords

P_1 c_{11} c_{12} … c_{1n}

P_i c_{21} c_{22} … c_{2n}

P_j c_{j1} c_{j2} … c_{jn}

P_n c_{n1} c_{n2} … c_{nn}

Iff $v_{ij} = v_{ji} = 1$

$(c_{ii} = c_{ji})$ and $(c_{ij} = c_{jj})$ holds if P_i and P_j are honest

Public Consistency Graph G

❑ Parties check if there is an (n, t)-star $(\mathcal{C}, \mathcal{D})$ in G

❑ Claim: All honest parties in $\mathcal{C}$ have the same local codeword, say $(c_1, \cdots, c_n)$

- Expanding $\mathcal{D} \rightarrow \mathcal{F}$
  - Include P_k to $\mathcal{F}$, if P_k has edges with at least $t+1$ nodes in $\mathcal{C}$
  - Every party in $\mathcal{D}$ will be included in $\mathcal{F}$

❑ Claim: For every honest $P_k \in \mathcal{F}$, the condition $c_{kk} = c_k$ holds

- P_k is pair-wise consistent with at least one honest party P_i in $\mathcal{C}$

$c_{kk} = c_{ik} = c_k$

P_k c_{kk} $c_1\, c_2\, c_k\, c_n$

So, now let us see how this expansion works. So, first of all as we have proved in the claim all the honest parties in the C component of the star, if at all a star is obtained will have the same local code word. That means, they have received the same code word from the sender.

Let us denote that common code word which is available only to the honest parties in the C set as $(c_1, \cdots, c_n)$. We now want to find additional honest parties who also would have received the same code word c 1 to c n from the sender. For that, we first try to expand the D component of the star. The expanded D set will be denoted by F.

Now, what will be the criteria to include a party in the F set? We will include a party P sub k to the F set, if that party has edges with at least t + 1 nodes in the existing C set in the consistency graph.

Now, because of this criteria all the parties who are already part of the D component of the star will be included in F, because every party in the D component of the star has edges with t + 1 nodes in the C component of the star because that is the definition of the n, t star.

So, all the parties in D will be by default included in F. But, now there might be some additional parties P k who were not present in D, but now they are included in F because those parties have edges with at least t + 1 nodes in the C component of the star.

Now, the claim here is that if we take all the honest parties in the F component, the kth component of such parties; that means, if I take any honest party P k in this set F and if I focus on the kth component of such honest parties P k that kth component is nothing but the kth component of this common Reed-Solomon code word held by all the honest parties in CORE. Why so?

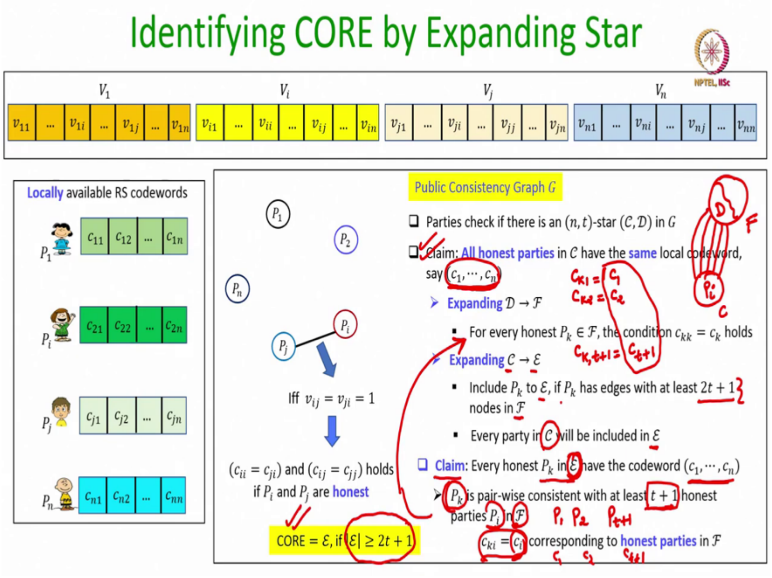
So, what is the criteria for including P k to the F set? We have included P k to the F set because it has edges with t + 1 nodes in C. Now, among those t + 1 nodes in C, at least one of the nodes is honest. Call that honest node P i. That means, in the consistency graph there is an edge between P k and P i.

Now, why there is an edge between P k and P i? Because during the pair wise consistency test, P k and P i would have found that the kth component in the kth parties code word is same as the kth component in the ith parties code word. And what is the kth component in the ith parties code word? Well, the kth component at the ith parties code word is c k because P i is an honest party in C, and as per our assumption every honest party P i in C have this code word.

So, P i would have received the code word c 1, c 2, c k, c n from the sender. And during the pair wise consistency test when P k would have sent the supposedly common values on its code word, so, one of those components would have been c kk, P i would have checked that c kk is same as c k. That guarantees that the kth component in kth parties code word is same as the kth component of the common code word held by the honest parties in C.

So, that is the property of the expanded D set. But we are not yet done. We have to expand the C set as well now.

(Refer Slide Time: 24:29)



Identifying CORE by Expanding Star

Locally available RS codewords

Public Consistency Graph G

❑ Parties check if there is an (n,t)-star $(\mathcal{C},\mathcal{D})$ in G

❑ Claim: All honest parties in $\mathcal{C}$ have the same local codeword, say $(c_1,\cdots,c_n)$

- ➢ Expanding $\mathcal{D} \rightarrow \mathcal{F}$
  - For every honest $P_k \in \mathcal{F}$, the condition $c_{kk} = c_k$ holds
- ➢ Expanding $\mathcal{C} \rightarrow \mathcal{E}$
  - Include P_k to $\mathcal{E}$, if P_k has edges with at least $2t+1$ nodes in $\mathcal{F}$
  - Every party in $\mathcal{C}$ will be included in $\mathcal{E}$

❑ Claim: Every honest P_k in $\mathcal{E}$ have the codeword $(c_1,\cdots,c_n)$

- ➢ P_k is pair-wise consistent with at least $t+1$ honest parties P_i in $\mathcal{F}$
  - $c_{ki} = c_i$ corresponding to honest parties in $\mathcal{F}$

Iff $v_{ij} = v_{ji} = 1$

$(c_{ii} = c_{ji})$ and $(c_{ij} = c_{jj})$ holds if P_i and P_j are honest

CORE = $\mathcal{E}$, if $|\mathcal{E}| \geq 2t+1$

NPTEL, IISc
V_1
V_i
V_j
V_n
v_{11} ... v_{1i} ... v_{1j} ... v_{1n}
v_{i1} ... v_{ii} ... v_{ij} ... v_{in}
v_{j1} ... v_{ji} ... v_{jj} ... v_{jn}
v_{n1} ... v_{ni} ... v_{nj} ... v_{nn}
P_1 c_{11} c_{12} ... c_{1n}
P_i c_{21} c_{22} ... c_{2n}
P_j c_{j1} c_{j2} ... c_{jn}
P_n c_{n1} c_{n2} ... c_{nn}
P_1
P_2
P_n
P_j
P_i
D
F
P_i
C
$c_{k1} = c_1$
$c_{k2} = c_2$
$c_{k,t+1} = c_{t+1}$
P_1 P_2 P_{t+1}
c_1 c_2 c_{t+1}

So, we will expand the C set and call the expanded C set as the E set, where the criteria to include a party P k to this expanded set E is the following. We should check whether P k has edges with at least 2t + 1 nodes in F, ok. I stress 2t + 1 nodes in F not in D.

Now, due to this condition for including a party in the E set, every party in C will be by default included in E because in F, the members of D will be anyhow present, that we have already discussed.

And, now if there is a party in C it will be anyhow having edges with all the nodes in D and D's cardinality would have been 2t + 1. That means, P i has edges with at least 2t+1 nodes in F because D is a part of F. So, that means, P i will be included in E as well. So, every party P i who was part of C will be included in E as well.

Now, comes the crux of the whole thing. We can now claim that every honest party P k in this E set, expanded C set, which is E set, have this common code word $(c_1, \cdots, c_n)$, ok. Why this is so? Well, E set will already have the honest parties in C and they already have this common code word c 1 to c n. That comes as part of the claim itself.

Now, what if we have added a new honest party P k in this expanded E set? That was not earlier part of the C set, but that is now a part of the expanded C set namely E set, our claim is that even those parties have the same common code word $(c_1, \cdots, c_n)$, as possessed by the honest parties in the C set. Why so?

This is because such parties P k who are now newly included to the expanded E set, they have edges with at least 2t + 1 nodes in F set. Among those 2t + 1 nodes at least t + 1 are honest. Say those t + 1 nodes are P i1, P i2, P it + 1 and corresponding to each such honest party P i from the F set with which P k has an edge or with whom P k's codeword is pairwise consistent, this condition will be satisfied.

This comes from the property that we had just proved in the earlier slide regarding the honest parties in the F set. Namely, every honest party P i in the F set their ith component; so, if I consider a party P i in the F set the ith component of their code word, the ith component of its code word is same as the ith component of this common code word. And P k is pairwise consistent with P i, that means, the ith component of the kth parties code word is also same as c i.

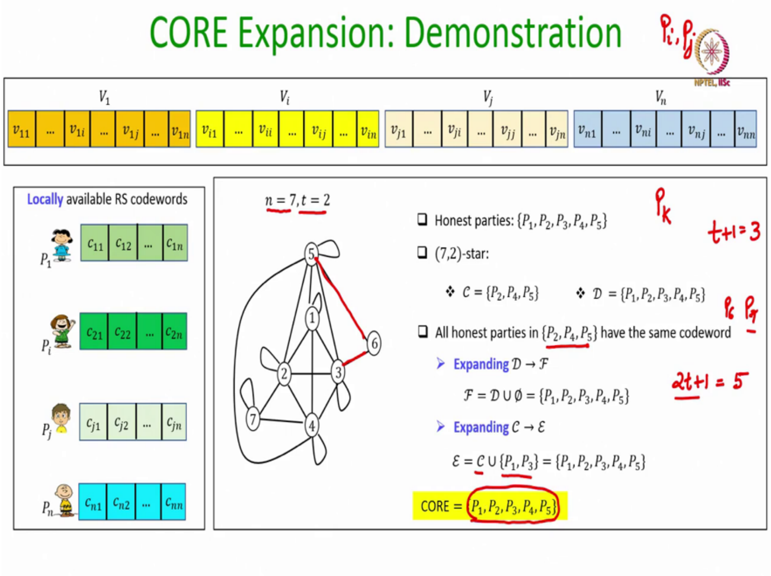
So, that means, if I take the t + 1 honest parties P i who are there in F, say P 1, P 2, P sub t plus 1 and P1 has c1, P2 has c2, Pt + 1 has ct +1 as part of their local code words.

Then, since Pk has an edge with P1, that means, c k1 is the same as c1 because of the pair wise consistency test. Since, P k has an edge with P2; that means, ck2 is the same as c2 during the pair wise consistency test. And like that since Pk has an edge with the t + 1th party, that means, the t + 1th component in the kth party's code word is the same as the t+1 th component in the t + 1th party's code word which is ct+1.

Now, if this is the case; that means, the code word held by the party P k is nothing, but the code word c 1, c 2, c n because the code word held by the party P k has t + 1 or more number of same components as the common code word c 1 to c n held by the honest parties in the C set. So, that ensures that all the honest parties in E, even the newly included parties, have this common code word c 1 to c n.

Now, we will check whether this expanded E set is of size at least 2t plus 1 or not. If this expanded E set is of size at least 2t plus 1, we will assign it as a CORE. Otherwise, we will simply discard the sender because sender is corrupt and it has not distributed sufficient number of common code words during the protocol, ok.

(Refer Slide Time: 30:34)



CORE Expansion: Demonstration

P_i, P_j

V_1							V_i							V_j							V_n						
v_{11}	...	v_{1i}	...	v_{1j}	...	v_{1n}	v_{i1}	...	v_{ii}	...	v_{ij}	...	v_{in}	v_{j1}	...	v_{ji}	...	v_{jj}	...	v_{jn}	v_{n1}	...	v_{ni}	...	v_{nj}	...	v_{nn}

Locally available RS codewords

P_1	c_{11}	c_{12}	...	c_{1n}
P_i	c_{21}	c_{22}	...	c_{2n}
P_j	c_{j1}	c_{j2}	...	c_{jn}
P_n	c_{n1}	c_{n2}	...	c_{nn}

$n = 7, t = 2$

P_k

$t+1=3$

❑ Honest parties: $\{P_1, P_2, P_3, P_4, P_5\}$

❑ (7,2)-star:

❖ $\mathcal{C} = \{P_2, P_4, P_5\}$ ❖ $\mathcal{D} = \{P_1, P_2, P_3, P_4, P_5\}$

P_6 P_7

❑ All honest parties in $\{P_2, P_4, P_5\}$ have the same codeword

➢ Expanding $\mathcal{D} \rightarrow \mathcal{F}$

$\mathcal{F} = \mathcal{D} \cup \emptyset = \{P_1, P_2, P_3, P_4, P_5\}$

$2t+1 = 5$

➢ Expanding $\mathcal{C} \rightarrow \mathcal{E}$

$\mathcal{E} = \mathcal{C} \cup \{P_1, P_3\} = \{P_1, P_2, P_3, P_4, P_5\}$

CORE = $\{P_1, P_2, P_3, P_4, P_5\}$

So, now, let us see the demonstration of this expansion of core, ok. So, imagine the honest parties in the system are 1, 2, 3, 4 and 5. We are taking the case when n is equal to 7 and t is equal to 2. So, two parties could be corrupt. For simplicity we are taking the first 5 parties to be honest and the last 2 parties to be corrupt. And say the sender has distributed the code words, sender is honest suppose for simplicity.

So, in the consistency graph, you can see that there is an edge between every pair of nodes in this subset 1, 2, 3, 4, 5. Namely, the edge 1 to 1 is also present. Because when I say a pairwise consistency test happens between every pair of nodes P i and P j that also includes the case when i is equal to j.

So, P 1 would have checked its own code word with its own code word. And by default it is considered to be pair wise consistent. So, 1 will have an edge with 1, 2 has an edge with 2, 3 has an edge with 3, 4 has an edge with 4, and 5 has an edge with 5. We do not care what the corrupt party says, regarding their response vector, ok.

So, now you can see that in this consistency graph there is an edge between every pair of honest nodes, ok and the self-loops are also there. So, since the consistency graph is guaranteed here to have a clique of size 5 namely, a clique involving 5 nodes is present, the star finding algorithm will output a star. And the star will be having a C component 2, 4 and 5 and the D component having the nodes 1, 2, 3 and 4 and 5. That means, all the honest parties in this C component have the same common code word received from the sender.

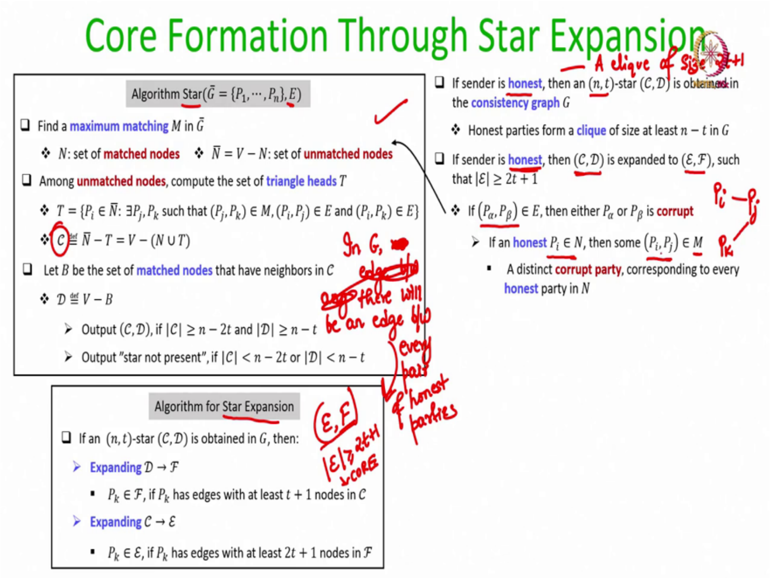
Now, let us see what happens during the expansion, whether D will be expanded to F and new parties are included in F or not. Well, there will be no new party included in D. All the old parties in D will be a part of F set, but no new party will be included in F set because the criteria for including a party in the F set is that the party P k which you want to include in the F set should have edges with t plus 1 nodes in C. So, who are the parties outside D? P 6 and P 7 and t plus 1 is 3.

So, P 6 does not have edges with 3 nodes, at least 3 nodes in this collection 1, 2, 3, 4, 5. So, it has an edge with 3 fine. It has an edge with 5, but it does not have an edge with 1. So, 6 cannot be included in F. And in the same way 7 also cannot be included because 7 has an edge with 2 and 7 has an edge with 4, but 7 does not have an edge with either 1 or 3 or 5, ok. So, F set remains the same as the D set.

But when we now try to expand the C set you can see that all the parties in C will be present in the expanded C namely E and there will be now two parties who will be included due to the expansion criteria. Namely, P 1 which was not earlier part of the C component of the star will be included in the C component because it now has edges with 2t + 1 nodes in the F components, and 2t + 1 is 5.

So, P1 has an edge with at least 5 parties in the F set. Namely, 1 has an edge with 1, 2, 3, 4 and 5. So, that is why it is included in C. And in the same way 3 which was earlier not included in the C component will be included because 3 has an edge with 1, 2, 3, 4 and 5. And since the size of E is 2t + 1, namely 5, the CORE will be set to 1, 2, 3, 4 and 5. And all the honest parties in this CORE are guaranteed to have received the same code word from the sender.

(Refer Slide Time: 35:11)



Core Formation Through Star Expansion

Algorithm Star($\bar{G} = \{P_1, \cdots, P_n\}, E$)

- ❑ Find a maximum matching M in $\bar{G}$
  - ❖ N: set of matched nodes ❖ $\bar{N} = V - N$: set of unmatched nodes
- ❑ Among unmatched nodes, compute the set of triangle heads T
  - ❖ $T = \{P_i \in \bar{N}: \exists P_j, P_k \text{ such that } (P_j, P_k) \in M, (P_i, P_j) \in E \text{ and } (P_i, P_k) \in E\}$
  - ❖ $\mathcal{C} \overset{\text{def}}{=} \bar{N} - T = V - (N \cup T)$
- ❑ Let B be the set of matched nodes that have neighbors in $\mathcal{C}$
  - ❖ $\mathcal{D} \overset{\text{def}}{=} V - B$
    - ➢ Output $(\mathcal{C}, \mathcal{D})$, if $|\mathcal{C}| \geq n - 2t$ and $|\mathcal{D}| \geq n - t$
    - ➢ Output "star not present", if $|\mathcal{C}| < n - 2t$ or $|\mathcal{D}| < n - t$

Algorithm for Star Expansion

- ❑ If an (n, t)-star $(\mathcal{C}, \mathcal{D})$ is obtained in G, then:
  - ➢ Expanding $\mathcal{D} \rightarrow \mathcal{F}$
    - ▪ $P_k \in \mathcal{F}$, if P_k has edges with at least $t + 1$ nodes in $\mathcal{C}$
  - ➢ Expanding $\mathcal{C} \rightarrow \mathcal{E}$
    - ▪ $P_k \in \mathcal{E}$, if P_k has edges with at least $2t + 1$ nodes in $\mathcal{F}$

- ❑ If sender is honest, then an (n, t)-star $(\mathcal{C}, \mathcal{D})$ is obtained in the consistency graph G
  - ❖ Honest parties form a clique of size at least $n - t$ in G
- ❑ If sender is honest, then $(\mathcal{C}, \mathcal{D})$ is expanded to $(\mathcal{E}, \mathcal{F})$, such that $|\mathcal{E}| \geq 2t + 1$
  - ❖ If $(P_\alpha, P_\beta) \in E$, then either P_α or P_β is corrupt
    - ➢ If an honest $P_i \in N$, then some $(P_i, P_j) \in M$
      - ▪ A distinct corrupt party, corresponding to every honest party in N

— A clique of size 2t+1

In G, ~~edge b/w~~ there will be an edge b/w every pair of honest parties

$(\mathcal{E}, \mathcal{F})$ $|\mathcal{E}| \geq 2t+1$ → CORE

P_i — P_j P_k

So, this is the actual process of finding the CORE. We will first check whether star is present or not through the star finding algorithm. And then we will run the expansion process. And if the expansion process gives us an E, F pair, where the E component is guaranteed to be of at least size 2t + 1, then we will set it to a CORE, otherwise we will discard the sender.

Now, we want to prove here that if the sender is honest, this expansion process will indeed output an E, F pair where the E component is of size 2t plus 1. That means, an honest sender is not going to be discarded. Let us see why. If the sender is honest then first of all this n, t star will produce a star in the graph because if the sender is honest a clique of size at least 2t + 1 is guaranteed in the graph. So, the star finding algorithm will output an n, t star.

Now, the second claim is if the sender is honest then eventually, not eventually, then through this expansion process the star will be expanded to a new pair E, F where the cardinality of E will be 2t + 1. Let us see why.

So, when the star algorithm is run on the G complement graph, so in the G complement graph the edge set is E, and if the sender is honest, and an edge is present in the G complement graph then at least one of the end points of that edge corresponds to a corrupt party.

Because in the G complement graph, an edge will be present either between a pair of corrupt nodes or between an honest node and a corrupt node. Because in G there will be no edges between any pair of honest nodes or honest parties, if the sender is honest.

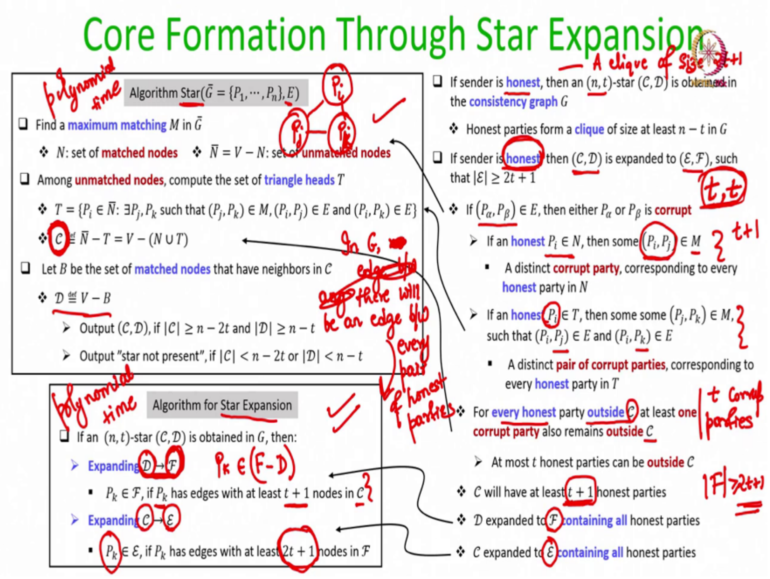
That means, in G complement there will be no edge involving a pair of honest parties. That means, if at all there is an edge in the G complement graph one of the end points of that edge corresponds to a corrupt party.

Now, when we run the star algorithm on such a graph and if an honest party becomes a part of a matching; that means, it becomes a matched node; that means, it is part of the maximum matching; that means, corresponding to that party P i there is some corrupt party P j who is present in the maximum matching.

That means, for every honest party P i who is not part of the C component here in the star finding algorithm, who is not part of the C component there is a corresponding distinct distinct corrupt party who is of course also outside the C component, ok. Why distinct? Because we are considering matching here.

So, it cannot be possible that there are two honest parties say P i who is pulled out of the C component because of this corrupt P j in the graph G prime. And there is another honest party P k who is also pulled out because the same P j has an edge with P k and P i, P j, P j, P k are all part of the matching. That is not going to be the case because in that case P i, P j and P j, P k does not constitute a valid matching, ok.

(Refer Slide Time: 39:47)



Core Formation Through Star Expansion

Algorithm Star($\bar{G} = \{P_1, \cdots, P_n\}, E$)

- ❑ Find a maximum matching M in $\bar{G}$
  - ❖ N: set of matched nodes ❖ $\bar{N} = V - N$: set of unmatched nodes
- ❑ Among unmatched nodes, compute the set of triangle heads T
  - ❖ $T = \{P_i \in \bar{N}: \exists P_j, P_k \text{ such that } (P_j, P_k) \in M, (P_i, P_j) \in E \text{ and } (P_i, P_k) \in E\}$
  - ❖ $\mathcal{C} \overset{\text{def}}{=} \bar{N} - T = V - (N \cup T)$
- ❑ Let B be the set of matched nodes that have neighbors in $\mathcal{C}$
  - ❖ $\mathcal{D} \overset{\text{def}}{=} V - B$
    - ➢ Output $(\mathcal{C}, \mathcal{D})$, if $|\mathcal{C}| \geq n - 2t$ and $|\mathcal{D}| \geq n - t$
    - ➢ Output "star not present", if $|\mathcal{C}| < n - 2t$ or $|\mathcal{D}| < n - t$

Algorithm for Star Expansion

- ❑ If an (n, t)-star $(\mathcal{C}, \mathcal{D})$ is obtained in G, then:
  - ➢ Expanding $\mathcal{D} \rightarrow \mathcal{F}$
    - ▪ $P_k \in \mathcal{F}$, if P_k has edges with at least $t + 1$ nodes in $\mathcal{C}$
  - ➢ Expanding $\mathcal{C} \rightarrow \mathcal{E}$
    - ▪ $P_k \in \mathcal{E}$, if P_k has edges with at least $2t + 1$ nodes in $\mathcal{F}$

- ❑ If sender is honest, then an (n, t)-star $(\mathcal{C}, \mathcal{D})$ is obtained in the consistency graph G
  - ❖ Honest parties form a clique of size at least $n - t$ in G
- ❑ If sender is honest then $(\mathcal{C}, \mathcal{D})$ is expanded to $(\mathcal{E}, \mathcal{F})$, such that $|\mathcal{E}| \geq 2t + 1$
  - ❖ If $(P_\alpha, P_\beta) \in E$, then either P_α or P_β is corrupt
    - ➢ If an honest $P_i \in N$, then some $(P_i, P_j) \in M$
      - ▪ A distinct corrupt party, corresponding to every honest party in N
    - ➢ If an honest $P_i \in T$, then some some $(P_j, P_k) \in M$, such that $(P_i, P_j) \in E$ and $(P_i, P_k) \in E$
      - ▪ A distinct pair of corrupt parties, corresponding to every honest party in T
  - ❖ For every honest party outside $\mathcal{C}$ at least one corrupt party also remains outside $\mathcal{C}$
    - ➢ At most t honest parties can be outside $\mathcal{C}$
  - ❖ $\mathcal{C}$ will have at least $t + 1$ honest parties
  - ❖ $\mathcal{D}$ expanded to $\mathcal{F}$ containing all honest parties
  - ❖ $\mathcal{C}$ expanded to $\mathcal{E}$ containing all honest parties

Now, let us see another category of honest parties who may be outside the C component, right. So, in the star finding algorithm which parties are outside the C component? All the matched nodes and all the triangle heads. So, we have argued that if an honest party is outside the C component because it was a part of the matched nodes; that means, along with that honest party, a corrupt party also is outside.

Now, suppose there is an honest party P i who is not a member of the C component of the star, because it is a part of the triangle head, in that case what is the scenario in our graph, we have in the graph G prime a structure like this is present.

We have the node representing P i and an edge P j, P k part of the maximum matching such that the edge between P i and P j is there and the edge between P i and P k is there, ok. That means, corresponding to this honest party P i who is a part of the triangle head and who is not included in the C component, a pair of corrupt parties is also outside the C component.

So, now the summary here is that for every honest node who is not included in the C component of the star at least one corrupt party also remains outside the C component because that honest party P i could be outside C either because of this condition or because of this condition. If it is outside due to this condition, that means, one corrupt party P j has pulled P i outside. Whereas, if the honest P i is outside because of this triangle head condition, that means, P j and P k have together pulled out P i out of the C component.

So, irrespective of the case we can say that for every honest party outside the C component who does not make it to the C component, there is at least one corrupt party who also does not make it to the C component. But there are at most t corrupt parties in the system. If there are at most t corrupt parties in the system they can pull out at most t honest parties from being part of the C component. So, t corrupt parties not making it to C component and along with that t honest parties are also not making it to the C component.

If we ignore this worst case scenario, we will be still left with at least t plus 1 parties who are guaranteed to be honest and who will be now present in the C component of the star if the sender is honest. All this I am arguing, if the sender is honest. Because we want to argue that sender will not be discarded during this expansion process. So, that means, if the sender is honest it is guaranteed that the star finding algorithm will output a star where the C component will have at least t plus 1 honest parties.

Now, if the C component has at least t plus 1 honest parties and if in the consistency graph, there is an edge between every pair of honest parties, then during this expansion process all the honest parties who do not make it to D will end up making it to F, right.

So, there might be some honest parties who were not present in the D component because of this condition, but will be now included in the F component if we run this expansion process. Because every honest P k who is present in F, but not in D they will have edges with at least t plus 1 nodes in C.

Because C is guaranteed to have at least t + 1 honest parties and every honest P k outside D will have an edge with every honest party in C. So, D will be expanded to F. And as a result, F will now have all the honest parties, if sender is honest. That means, the cardinality of F will be at least 2t + 1.

And, now let us see the expansion for the C to E part. If we see the expansion for the C to E part it turns out that C will be expanded to E and it will also include all honest parties. That means it could be possible that the C component does not have all the honest parties because t corrupt parties could have pulled out t honest parties from being part of the C component during the star finding algorithm.

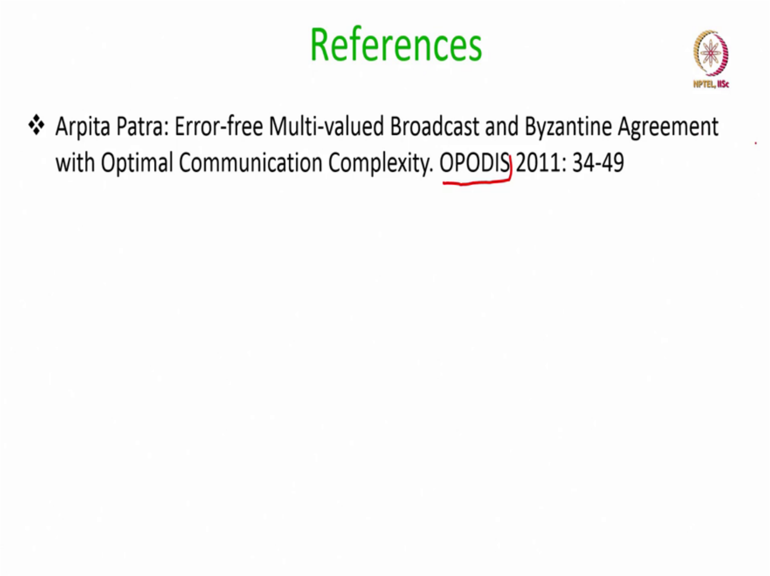
But when we run this expansion process those parties P k who missed out to be a part of the C component will end up to be a part of E component. Because all those missing parties will now have edges with 2t + 1 nodes in F because F is guaranteed to have all the honest parties, and there are 2t + 1 honest parties guaranteed in the system.

That means, if the sender is honest this expansion process will end up including all honest parties who missed out to be up to being part of D and C and they will be now part of F and E components. And E component will now have all the honest parties at least 2t +1 in number and an honest sender will not be discarded.

However, if the E set does not turn out to be of size 2t + 1, definitely sender is corrupt. So, it is safe to discard the sender and take some default input, default message as the output on the behalf of the sender. Now you can see that the star finding algorithm is polynomial time and this expansion process is also polynomial time because we are not finding any clique in this graph. At no step we are checking for the clique in the graph.

So, now if we incorporate this modification in the warmup protocol, we get a warmup protocol with polynomial computation complexity. And then, when we run that warmup protocol in the exponential time domain extension, domain extension protocol where we have every party broadcasting a combined response vector, we get a polynomial computation complexity, domain extension protocol.

(Refer Slide Time: 46:52)



References

❖ Arpita Patra: Error-free Multi-valued Broadcast and Byzantine Agreement with Optimal Communication Complexity. OPODIS 2011: 34-49

So, this is the full domain extension protocol present in this very beautiful work. So, this is the reference for today’s lecture is this paper. And I would like to stress that this whole area of broadcast and byzantine agreement domain extension is a very active area of research, very fundamental work is going on. So, if you are interested to know more about this line of work you can refer to this paper.

Thank you.