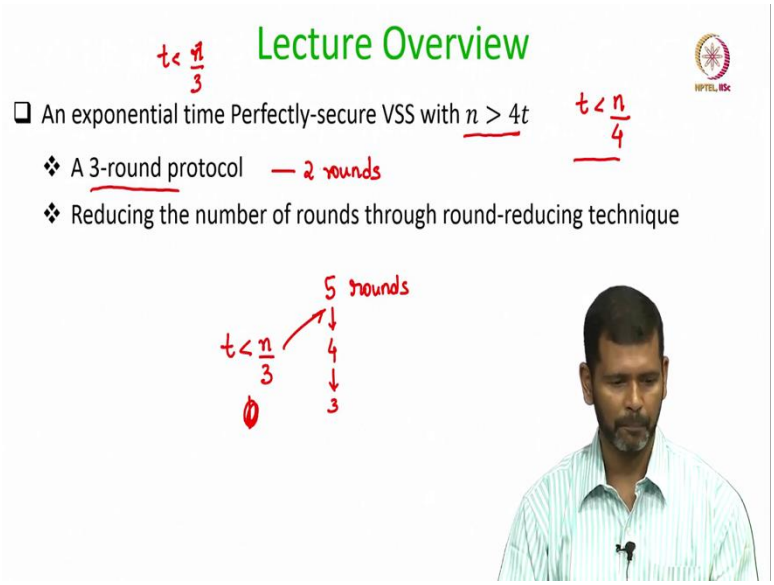
Secure Computation: Part II
Prof. Ashish Choudury
Department of Computer Science and Engineering
Indian Institute of Science, Bengaluru

Lecture - 42
Perfectly-Secure VSS with n greater than 4t: Part I

Hello everyone, welcome to this lecture.

(Refer Slide Time: 00:25)



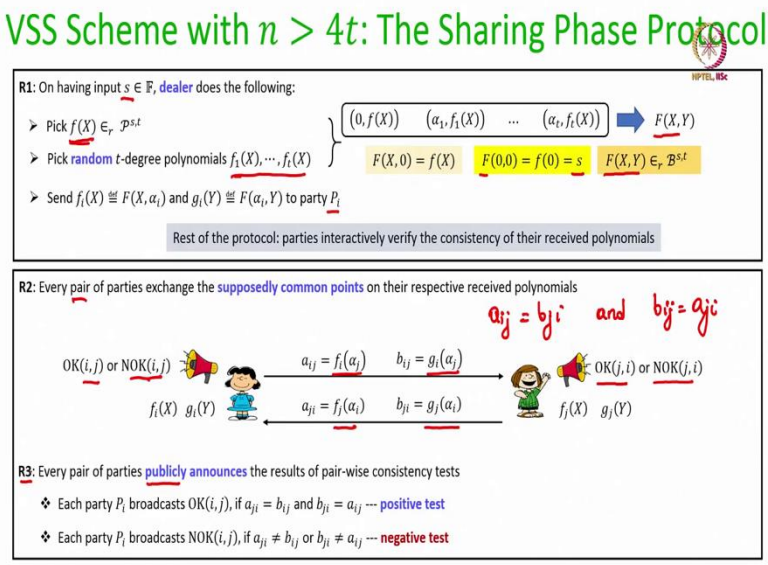
So, in this lecture we will discuss Perfectly Secure Verifiable Secret Sharing Scheme, where we have less number of corrupt parties compared to the previous schemes. Namely, the number of curved parties is less than n over 4 and since the number of corrupt parties are less than the previous protocol. So, in the previous protocol the number of corrupt parties were up to n over 3 and that is why the verification process, namely to verify whether the dealer has shared consistent polynomials or not was quite technical right.

It involves a lot of steps, but now what we will see in today's lecture is that if you have less number of corrupt parties then the verification process becomes relatively simpler. So, for today's lecture we will consider first a 3 round protocol ok and then we will see that by using the round reducing technique which we had discussed in the previous lecture, we can bring down the number of rounds to 2 rounds ok. And, this is clearly an improvement over the previous protocol the previous verifiable secret sharing scheme, because in the previous VSS scheme the number of rounds were 5.

So, you can see that you have a trade off. If you want to tolerate more number of corruptions namely t less than n over 3, then the protocol required 5 rounds. Of course, we can we could reduce the number of rounds to 4 by using the round reducing technique. And we have not discussed, but we know that a minimum number of rounds required by any VSS scheme, perfectly secure VSS scheme with t less than n over 3 is 3 nodes.

Whereas what we are going to discuss today is a 2 round protocol with t less than n over 4. So, if you have a trade-off between the number of faults which you can tolerate in the protocol and the number of rounds in the sharing phase.

(Refer Slide Time: 02:36)



So, let us start with the sharing phase protocol. So, the most of the steps of the sharing phase protocol remains the same as it was for the previous VSS scheme, where dealer embeds its secret in our random bivariate polynomial. And, then it distributes the row and column polynomials to individual parties and then the parties interact and verify whether the row and column polynomials of all honest parties lie on a single t degree bivariate polynomial.

But, now the steps become a simpler compared to the previous VSS scheme. So, in the first round dealer on having a secret s embeds it in a random t degree bivariate polynomial. For that it first picks a random t degree Shamir's sharing polynomial and additionally t random univariate polynomials in x, each in variable x and each as degree t. And, using

this t plus 1 polynomials namely the Shamir's sharing polynomial and the t random univariate polynomials, it interpolates and gets a random t degree bivariate polynomial.

And because of the way the secret is embedded in the bivariate polynomial, it holds that the constant term of the bivariate polynomial is the dealers secret which also happens to be the constant term of the Shamir's sharing polynomial f of X.

And, since this polynomials f 1 X f 2 f t X are picked randomly and so, is the Shamir's sharing polynomial. It guarantees that the overall bivariate polynomial is a random t degree bivariate polynomial. It then hands over the ith row and ith column polynomial of this bivariate polynomial to the ith party.

And, then the rest of the protocol the involves interaction among the parties to verify whether the dealer has distributed consistent row and column polynomials to all the honest parties, because if the dealer is corrupt and it may not follow the protocol instructions. So, right now I am explaining the protocol without using the round reducing technique. At the end of the lecture, we will see that how we can reduce the number of rounds during the pair wise consistency check using the round reducing technique which we had discussed in the earlier lecture.

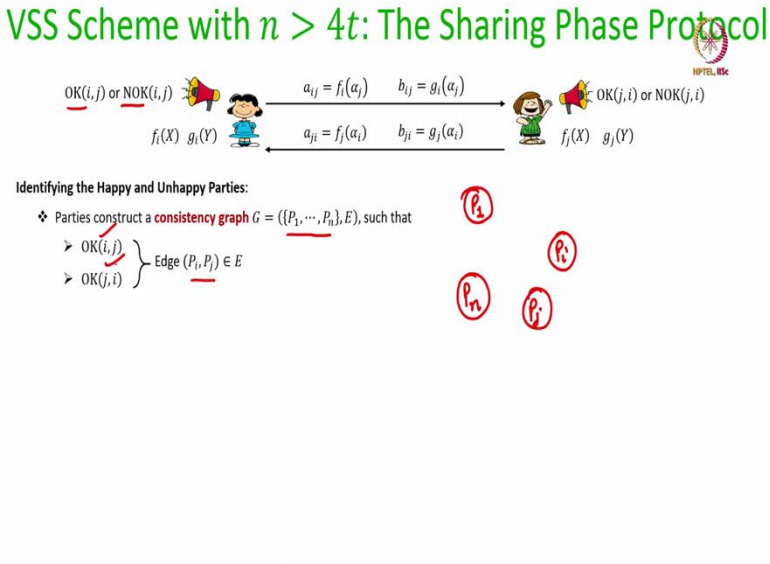
So, once the first round is over, then during the second round every pair of parties exchanged the supposedly common points on their respective polynomials. So, you have party P i and party P j with their respective row and column polynomials.

And, they exchange the supposedly common points on their respective row and column polynomials with each other over the private channel during the second round. So, for instance P i sends the value of its row polynomial evaluated at alpha j and the value of its column polynomial evaluated at alpha j to P j.

And, in the same way P j sends the value of its row polynomial at alpha i and the value of its column polynomial at alpha i to P i. And, then ideally we expect that if dealer P i and P j are honest then we expect that a ij should be equal to b ji. And, we also expect that b ij should be equal to a ji; ideally this should be the case if dealer P i and P j are honest. So, during the third round, the parties after performing the pairwise consistency check announces the results of the pairwise consistency checks publicly.

So, that involves the usage of broadcast channel. So, P i says that it is fine or OK with j, if the pairwise consistency passes, yes if it is positive. Otherwise, it sends an NOK message against P j; that means, it is in dispute with P j and similarly P j also after performing the pairwise consistency check either broadcasts an OK message or NOK message indicating that it is in a dispute with P i.

(Refer Slide Time: 06:52)



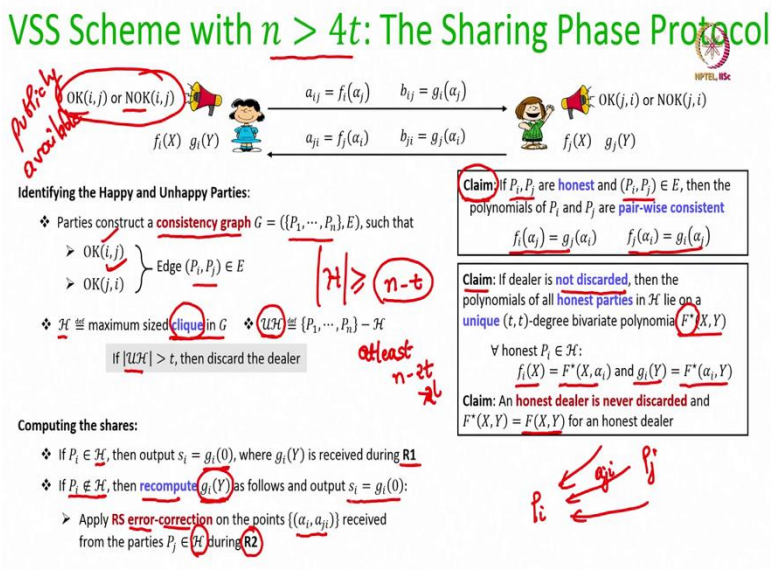
Now, we have already utilized 3 rounds. Now, based on the results of the pairwise consistency checks which are publicly available, why they are publicly available? Because, all these OK and NOK messages are made public using the broadcast channel and anything which has been made public using the broadcast channel will be received identically by all the honest parties. So, at the end of the third round, all the parties will know whether party P i is in dispute with P j or not and vice versa for every pair of parties P i P j.

Now, based on this results the parties identified the set of happy and unhappy parties as follows. So, the parties construct a consistency graph where the vertices are the parties themselves. So, there will be a node representing party P 1, there will be a node representing party P 2, there will be a node representing party P i, there will be a node representing party P j and so on ok.

So, there will be n nodes in the vertex, in the graph and the edge set in this consistency graph will be as follows. So, there will be an edge between the nodes representing the

parties P i and P j, if and only if both of them are ok with each other. That means, neither P i is in dispute with P j nor P j is in dispute with P i; if such that is the case then there will be an edge between the nodes representing P i and P j in the graph. Otherwise, there will not be any edge between the nodes representing P i and P j.

(Refer Slide Time: 08:26)



Now, notice that this graph, this consistency graph will be common; that means, all the parties will be constructing the same consistency graph. That means, it will not be the case that if say for instance P 1 has added an edge between P i and P j in its copy of the consistency graph, then P 2 is not adding that edge because this edges are added based on the OK and NOK messages which are publicly available ok.

Now, let us make some claims regarding the properties of this consistency graph. So, the first property is that if we consider a pair of parties P i and P j who are honest and if there is an edge involving the nodes P i and P j in the consistency graph, then their polynomials are pairwise consistent ok. This is because the edge between P i and P j is added only when P i has broadcasted an OK message for P j and when P j has broadcasted an OK message for P i.

And, why these parties would have broadcasted OK messages for each other? Only after verifying the only after verifying that the pairwise consistency test is positive for P i and P j individually, which automatically implies that they are polynomials the row and column polynomials are pairwise consistent ok. Now, based on this claim, the parties identify the

set of happy and unhappy parties as follows. So, the set of happy parties is defined to be the parties in the maximum sized click in the graph G ok.

So, what is a click? It is a complete graph. So, the parties once they construct the consistency graph, they try to find out the maximum sized complete graph in that consistency graph. And, the parties representing the corresponding nodes constitute the happy set of parties H and the remaining parties who are not part of that click, maximum sized click is said to be the set of unhappy parties ok.

Now, the first sanity check which we make here in the protocol is the following. If we see that the number of unhappy parties is more than t, then it automatically implies that the dealer is corrupt because of this claim and so, it is safe to discard the dealer. And, discard the dealer means terminator sharing phase protocol here itself and assume that dealer wanted to share the value 0 and everyone takes a default share 0 on the behalf of the dealer.

Now, why this sanity check makes sense here, because if the dealer is honest then all the honest parties in the system will be pairwise consistent because of this claim because, dealer will be giving consistent row and column polynomials to all the honest parties and, as a result of that the pairwise consistency check between every pair of honest parties will be positive, they will be broadcasting OK messages for each other. And, how many honest parties are guaranteed to be in the system?

At least n minus t; that means there could be up to t corrupt parties who may unnecessarily broadcast an NOK message against an honest parties ok so, how many; that means, what could be the maximum number of unhappy parties, if the dealer is honest? Well, no honest party will be unhappy if the dealer is honest, only the potentially corrupt parties can be unhappy. And, how many corrupt parties are there in the system? There are up to t corrupt parties.

So, that is why this cardinality of the set of unhappy parties for an honest dealer will be at most t, it never crosses t. And, that is why an honest dealer will never get discarded because of this sanity check that shows that if at all the number of unhappy parties is more than t then definitely the dealer is corrupt and so, it is safe to discard the dealer right. So, now, based on the sanity check, let us make the second claim here. We claim here that if the dealer is not discarded and we have already argued that an honest dealer is definitely not getting discarded.

Then we claim here that the polynomials of all the happy and honest parties lie on a single t degree bivariate polynomial, call that bivariate polynomial as F star X, Y. And, what do I mean by saying that the polynomials of all the honest parties lie on this bivariate polynomial? That means, if I take an honest happy party P i, then its X univariate polynomials lies on this F star bivariate polynomial; that means, it is this f i of X polynomial is the value of the bivariate polynomial F star at Y equal to alpha i.

And, similarly the Y univariate polynomial which P i has received from the dealer also lies on the bivariate polynomial F star. Moreover, this polynomial F star will be same as F of X Y, if the dealer is honest; because an honest dealer distributes unit row and column polynomials lying on F of X Y. And, we have already argued that an honest dealer is never discarded in the protocol and for the honest dealer the polynomials the F, the row and column polynomials of all the happy and honest parties will lie on F of X, Y.

Namely, the polynomial selected by the dealer itself. So, now, let us prove this claim, why this claim is correct. Well, this claim is correct because if the dealer is not discarded then we know that there are at most t unhappy parties; that means, there are at least n minus t parties in the happy set and up to t parties in the happy set might be corrupt fine. So that means, if I focus on the happy parties who are also honest then they are at least n minus 2 t in number and n minus 2 t is clearly greater than 2 t.

Because, we are working with the condition n greater than 4 t; that means, we have at least 2 t plus 1 happy parties in the on we have at least 2 t plus 1 parties who are happy and who are honest and whose polynomials are pairwise consistent because of this claim. So, now, we can trigger the pairwise consistency lemma and conclude that the row and column polynomials of all the happy and honest parties together lie on a single bivariate polynomial F star of X Y.

Now, the protocol is over, in the sense that we have no more rounds involved required in the protocol. If the dealer is not discarded; that means, the verification is done and what is left now is to compute the shares for the respective parties. So, how the shears are computed? So, depending upon whether a party is a part of the happy set or not, the shares are computed differently. If a party P i is part of the happy set, then it simply takes the constant term of the column polynomial, which it has received from the dealer during the first round as its share.

Whereas, if the party P i is not part of the happy set; that means, it is not part of the click, maximum sized click in the consistency graph then it has to do something else to get its share. What it does is the following. It simply ignores the column polynomial which it has received during the first round from the dealer and it now recomputes its column polynomial.

So, the recomputed column polynomial is also denoted by g i of Y. So, the previous g i of Y which it has received from the dealer during round 1 is ignored and it computes a new g i of Y as follows. And, once it computes the new g i Y polynomial, it takes the constant term of that polynomial as its shear. So, now, let us see how this new g i Y polynomial or the new column polynomial is computed.

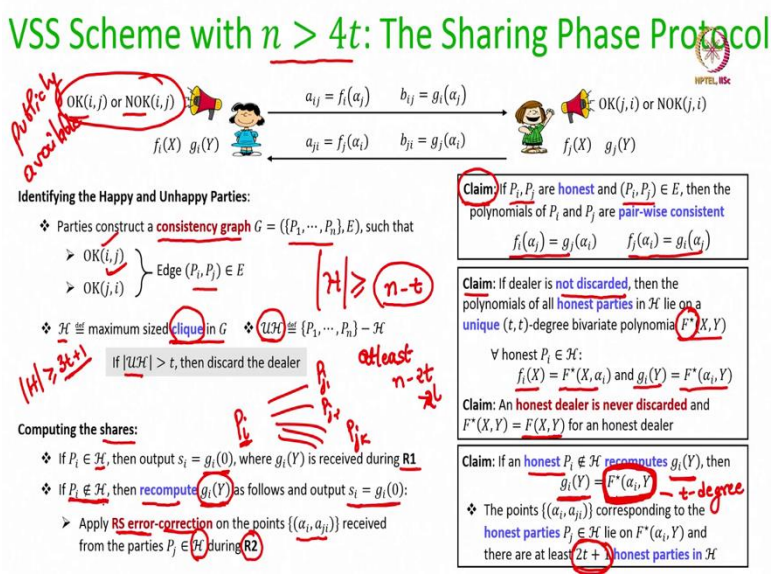
So, during the second round as part of the pairwise consistency check, this party P i who is not part of the happy set would have received the common values, the supposedly common values from different parties P j ok during the second round. So, it would have received the value a ji from different parties P j during the second round as part of the pairwise consistency check, it focuses only on the a ji values received from the happy parties ok.

So, during the second round, it would have received the supposedly common value a ji from all the n parties. What we are saying here is that among those a ji values, take only the a ji values the common values which have been received from the happy parties. And, the happy parties are now identified right; it has been identified at the end of third round.

So, it may not be the case that all the parties are happy, some parties might be unhappy. So, for the unhappy parties whatever common points you have received during the second round ignore them, focus just on the a ji values, the common values which P i would have received during second round from the happy parties.

And, then it applies the Reed-Solomon error correction and try to get a t degree polynomial. And, we will show it will get a t degree polynomial, that t degree polynomial which it obtains by applying the Reed-Solomon error correction on this common a ji values received during round 2 is taken as the recomputed column polynomial by P i. And, as I said earlier the constant term of this recomputed polynomial is taken as the shear by party P i.

(Refer Slide Time: 19:39)



Now, what we are going to claim here is that if there is some unhappy honest party P i, then as per this process of computing the shares, it will be recomputing its column polynomial g i of Y. We are claiming here that this recomputed column polynomial g i of Y is nothing but a column polynomial on this bivariate polynomial F star defined by the polynomials of the happy honest parties.

That means, through this recomputation process whatever P i is computing, whatever column polynomial is computing; it is not an arbitrary column polynomial. It is actually a column polynomial lying on this bivariate polynomial F star which is defined or which is guaranteed to exist, because of the pairwise consistency check of the between the column and the row polynomials of the happy honest parties. So, let us prove this claim. So, how this recomputed polynomial is calculate computed here?

So, this recomputed polynomial is obtained by applying the Reed-Solomon error correction process. So, imagine that there are different parties P j so, say P j 1 P j 2 P j k, k number of P j parties are there in the happy set. And, they would have sent the common points on their column polynomials, sorry the common points on their respective row polynomials to party P i. Now, among all these parties in the happy set who would have sent the common points on their respective row polynomials to P i during round 2?

At least 2 t plus 1 are guaranteed to be honest. Why so? Because, the cardinality of the happy set is at least n minus t; that means, the cardinality of the happy set is at least 3 t

plus 1. And, the common points on the row polynomials of all the honest parties in this happy set, it uniquely determines this column polynomial F star alpha i Y. And, what is the degree of this column polynomial F star alpha i Y? It is a t degree polynomial.

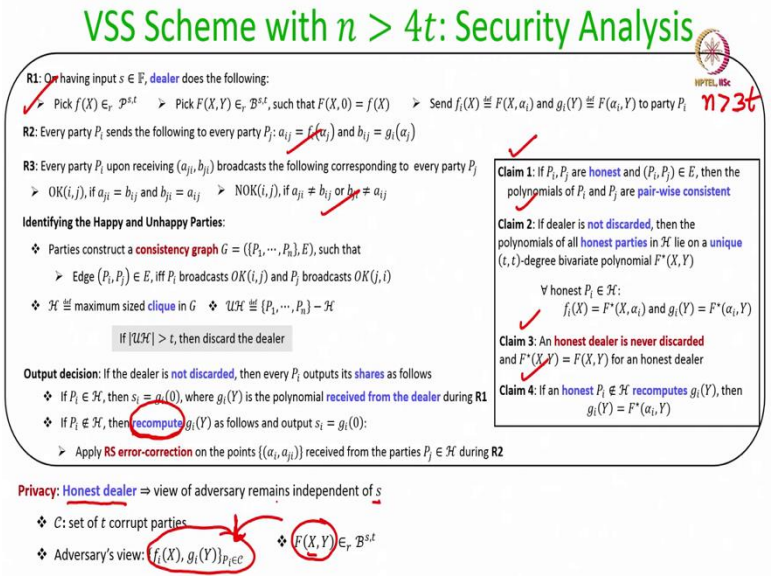
Now, there could be up to t corrupt parties in the happy set, who during the second round might have given incorrect common points to P i. And, it is precisely those points which are error corrected through this Reed-Solomon error correction process.

So, it is not guaranteed that every party in the happy set would have given the correct point on its row polynomial to P i during the second round. The honest parties would have given the right points on their respective row polynomials to P i, but the corrupt parties might have given incorrect points to P i.

There could be up to t such incorrect points, but there are at least 2 t plus 1 correct points and overall the degree of this F star alpha i Y polynomial is t. So, now, we know that if the degree of the polynomial is t and there are at least 3 t plus 1 values of which up to t could be corrupt, the Reed-Solomon error correction process will work correctly.

And, whatever polynomial is obtained by P i, namely this recomputed g i Y polynomial; it will be nothing but the supposedly column ith column polynomial on this F star bivariate polynomial. That also shows that whatever shear is computed by whatever shear is computed by P i because of the second case is nothing but a valid shear. We will prove that very soon ok.

(Refer Slide Time: 24:09)



So, these are the 4 claims with respect to this sharing phase protocol and, now you can see that the sharing phase protocol is relatively simpler here compared to the previous VSS scheme. Dealer distribute the polynomials, parties perform the pairwise consistency check, make the result public and that is all. We are not asking the dealer to resolve the polynomials of the unhappy parties which was happening in the previous protocol.

The unhappy parties, they are automatically going to recompute their column polynomials by applying a Reed-Solomon error correction process. Now, you might be wondering that why this recomputation may not work for the previous VSS scheme with n greater than 3 t. Well, I leave it as an exercise for you, you can go through the analysis that we are doing in this lecture.

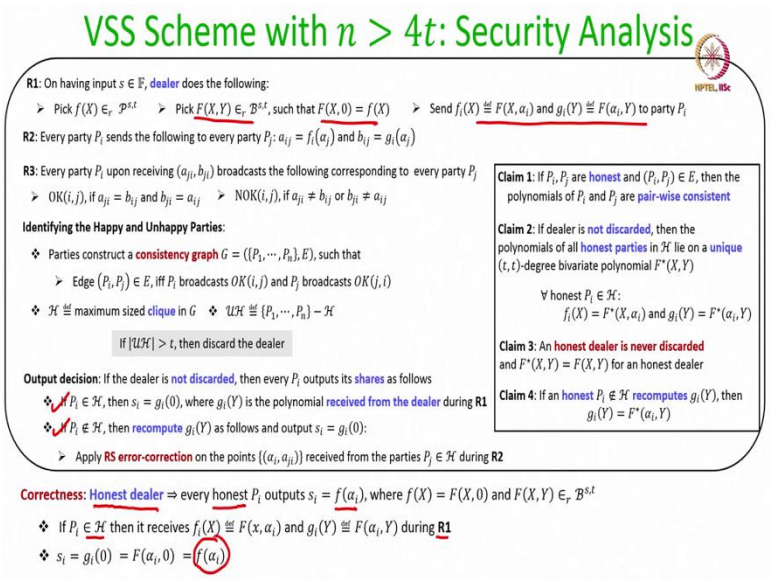
And, then you can see that where that analysis fails for the previous protocol; that means, why this protocol will fail with n greater than 3 t you can verify that, why it works with n greater than 4 t. So, now, let us prove the properties of the VSS for this VSS scheme; namely the privacy, correctness and the strong commitment.

For privacy, we have to consider an honest dealer and we want to show that if the dealer is honest during the sharing phase and the view of the adversary remains independent of the dealers secret and the proof for this more or less remains the same as it was for the previous VSS scheme.

Namely, what exactly the t corrupt parties receive in this protocol, what exactly is their view? Their view is basically limited to up to t number of row and column polynomials, nothing more than that and this t row and column polynomials lie on a random t degree bivariate polynomial. So, we can now apply the privacy lemma.

We can trigger the privacy lemma here which says that if there is a random t degree bivariate polynomial, then the probability distribution of any number of any t number of row and column polynomials is independent of the constant term of the bivariate polynomial and that guarantees the privacy here.

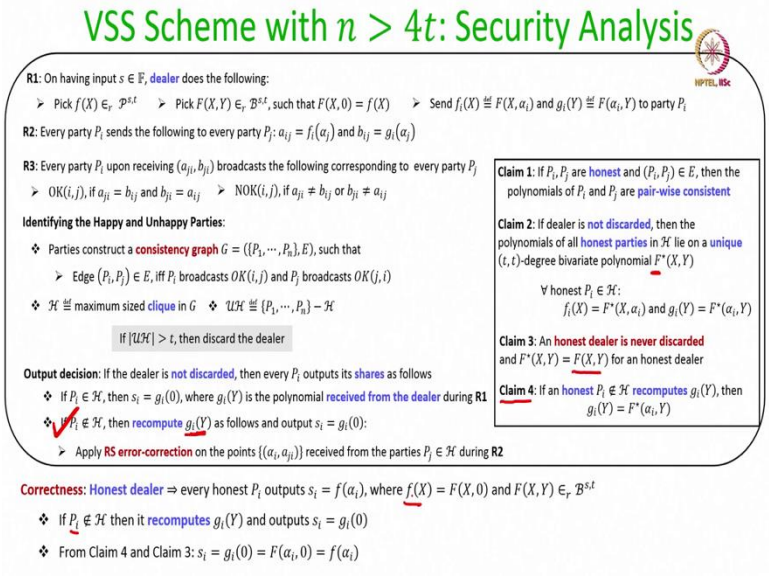
(Refer Slide Time: 26:24)



Let us prove next the correctness property for which again we have to consider an honest dealer. And, we want to show here that if the dealer is honest then there exists t degree Shamir's sharing polynomial such that every honest party outputs a share which is the value of that Shamir's sharing polynomial at alpha i. Now, you can see there are two different ways by which a party would have computed its share.

So, if the party P i is a part of the happy set, then what exactly is the share of that party P i? Well, it is the constant term of the column polynomial which that party would have received from the dealer during round 1. And, now because of the way dealer would have embedded the Shamir's sharing polynomial in the bivariate polynomial, it follows that the constant term of the column polynomial g i Y is nothing but the value of the Shamir's sharing polynomial at X equal to alpha i.

(Refer Slide Time: 27:32)



Whereas, it could be possible that there is an honest party P i who is not a part of the maximum sized click; that means, it is not a part of the happy set in which case the party would have recomputed its column polynomial and output, the constant term of that recomputed polynomial as its share. But, we have already proved in claim 4 that even the recomputed polynomial is going to lie on the bivariate polynomial F star as defined by the polynomials of the happy honest parties.

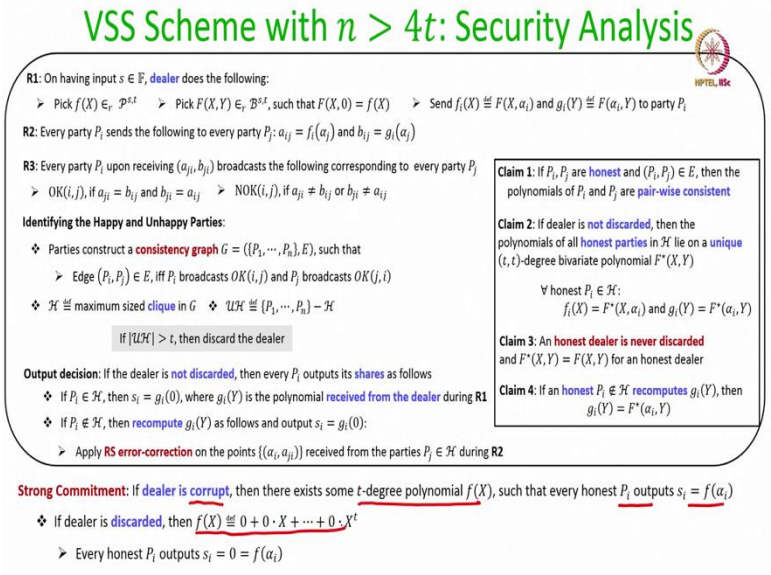
And, for the case of the honest dealer, the polynomial F star defined by the polynomials of the happy honest parties is same as the polynomial F of X, Y which automatically shows that the recomputed polynomial g i of Y is going to be the same g i of Y polynomial, which P i would have received from the honest dealer during the first round.

Now, you might be wondering if the dealer is honest, why P i is again recomputing its polynomial, if it is not a part of the happy set? Well, P i would not know actually whether the dealer is honest or corrupt.

It has to follow the protocol steps and what we are guaranteeing here is that even if the P i is recomputing its column polynomial for the case when dealer is honest, its recomputed polynomial will end up to be the same polynomial, same column polynomial, which it would have received from the dealer during the first round. So, that automatically shows that even for the parties who output or who set their shares based on the recomputed

column polynomials, their shares are basically lying on the summit sharing polynomial f of X.

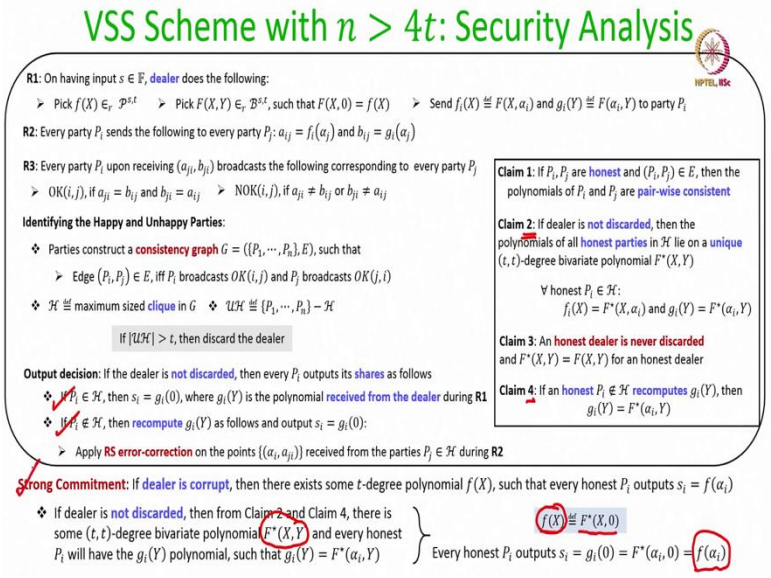
(Refer Slide Time: 29:18)



So, that completes the correctness property. Now, let us see the strong commitment property for which we have to consider a potentially corrupt dealer. And, for a potentially corrupt dealer, we want to show that there exists some t degree polynomial such that the shear of every honest party is going to lie on that Shamir's sharing polynomial.

And, again there are two possible cases depending upon whether the dealer is discarded of this because of the sanity check or not. If the dealer is discarded and well straight strong commitment is guaranteed, because in that case we can take the Shamir's sharing polynomial to be the constant 0 polynomial, which guarantees that the share of every honest party is 0.

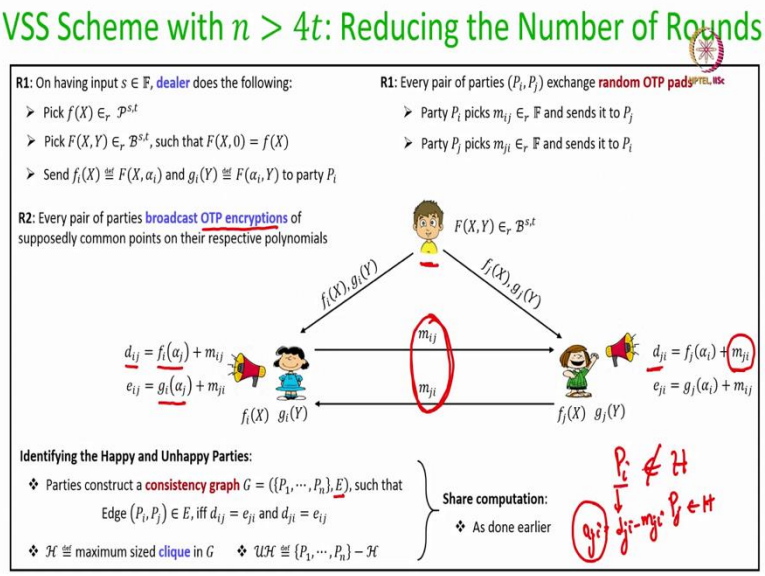
(Refer Slide Time: 30:01)



Whereas, if the dealer is not discarded then because of this claim 2 and claim 4, we know that there exists some t degree bivariate polynomial F star such that the column polynomials of all honest parties at the end of third round is going to are going to lie on that F star X Y by bivariate polynomial right. And, that is because of the claim 2 and claim 4.

Now, let us take the Shamir's sharing polynomial f of X to be the value of this defined bivariate polynomial at y equal to 0. So, this f of X is actually a t degree polynomial because F star is t degree bivariate polynomial. And, now it is easy to see that irrespective of whether the party P i, an honest party P i is happy or unhappy; the shear computed by it is nothing but the value of this defined Shamir's sharing polynomial at X equal to alpha i which implies that the strong commitment property is satisfied ok.

(Refer Slide Time: 31:08)



So, we have we are now done with the 3 round VSS scheme. But, I promised you that we will see that this protocol actually requires 2 rounds. So, we what we can do is we can reduce the number of rounds to 2 by applying the round reducing technique. So, what we are going to do is the protocol remains more or less same, but in the first round parties will pre exchange some random pads for OTP, which simplifies the pairwise consistency check during the second round.

Namely, during the first round between every pair of parties P i and P j, they will exchange among themselves random OTP pads. And, then during the second round, this random OTP pads are used to make public OTP encryptions of the supposedly common points. Now, since the pads are exchanged privately between P i and P j, even if this OTP encryptions are made public, nothing is going to be learned about the individual values of the row and column polynomials of an honest P i and honest P j, if dealer is honest.

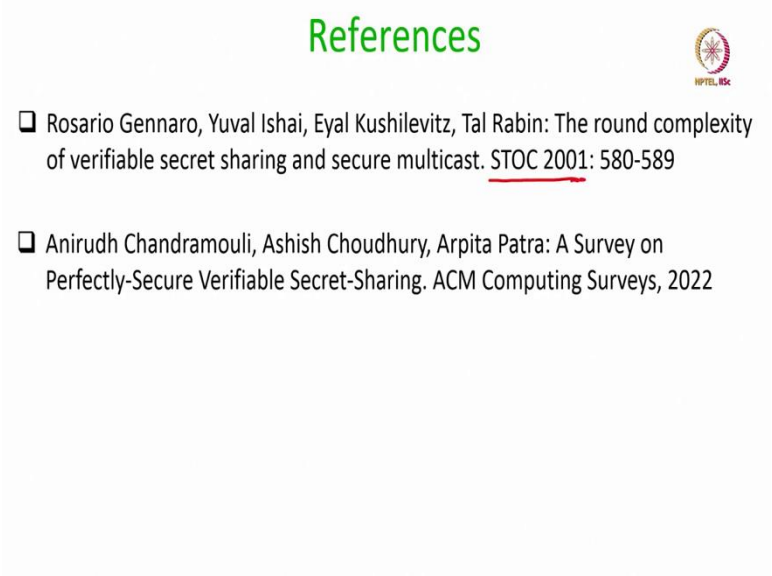
Whereas, if the dealer is corrupt, it will not be knowing beforehand what are the pads which have been exchanged between P i and P j. And, that implies that if dealer has given inconsistent polynomials P i inconsistent polynomials to P i and P j, then the OTP encryptions will mismatch and there will be a conflict between P i and P j. Now, how do we identify the happy and unhappy parties here? Well, the construction of the consistency graph remains the same. Namely, we will have n nodes representing the n parties.

And, now there will be an edge between P i and P j, if they are OTP encryptions are consistent; otherwise there will not be any edge. And, then we take the happy parties to be the parties in the maximum size click and so on. And, shear computation happens as it was happening earlier ok, similar in the similar in the same way.

Namely, if there would have been a party P i who is not part of the happy set, then it needs to find out the common values on its column volume polynomial which it would have received from the parties in the happy set.

So, what it can do is that each party in the happy set, if there is party P j in the happy set it would have made public the value d ji and m ji would have been received by P i. So, what P i can do is, it can set a ji to be d ji minus m ji. And, then on this a ji values, it can apply the Reed-Solomon error correction as it was doing earlier and try to recompute its column polynomial g i of Y. So, rest of the details remains the same.

(Refer Slide Time: 34:12)



So, the 2 round protocol which I had discussed today is taken from this paper. And, if you want to know more about perfectly secure verifiable secret chaining schemes, you can refer to this survey paper.

Thank you.