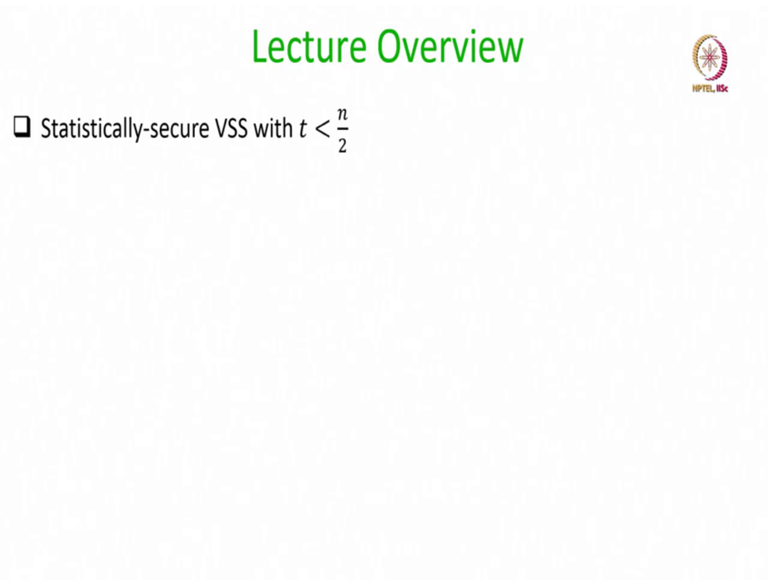
Secure Computation: Part II
Prof. Ashish Choudhury
Department of Computer Science and Engineering
Indian Institute of Science, Bengaluru

Lecture - 58
Statistically-Secure VSS

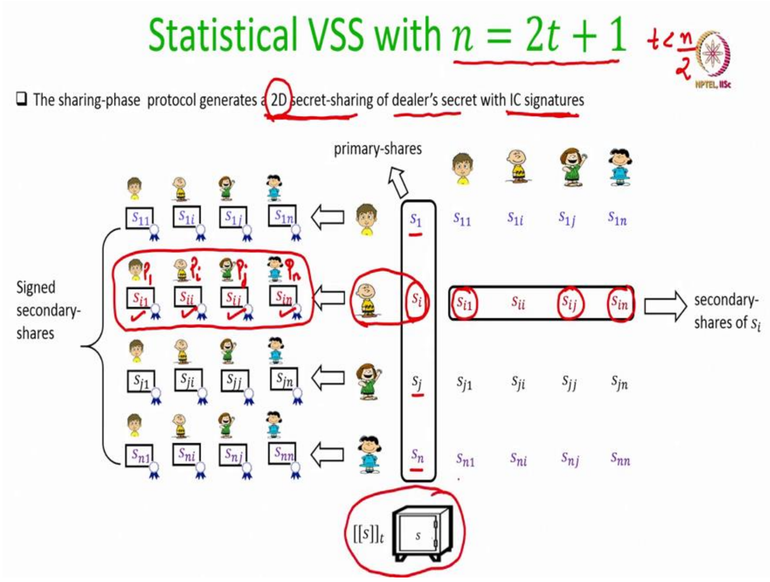
Hello everyone, welcome to this lecture.

(Refer Slide Time: 00:25)



So, in this lecture we will see a statistically secure verifiable secret sharing scheme with the condition t < n/2.

(Refer Slide Time: 00:33)



So, for simplicity we will assume that the number of parties is 2t+1, this is the smallest value of n satisfying the condition t < n/2. And remember that t < n/2 is the optimal resilience bound for statistically secure multi party computation.

The sharing phase protocol of this VSS scheme generates a 2D secret sharing of the dealers secret with IC signatures in a verifiable fashion. So, if the dealer is honest then the privacy of the dealer's secret will be maintained. And the verifiability ensures that even if the dealer is corrupt, at the end of the sharing phase protocol, there is some value, which the dealer has secret shared and which has been secret shared in a 2D secret shared fashion with IC signatures.

So, just to recap what exactly is a 2D secret sharing of a value with IC signatures. So, a value s is said to be 2D secret shared with IC signatures, if you have a set of primary shares lying on a t degree polynomial with the ith party holding the ith share. And each primary share is further secret shared through a t degree polynomial or Shamir secret shared and the shares for the primary share si are called the secondary shares.

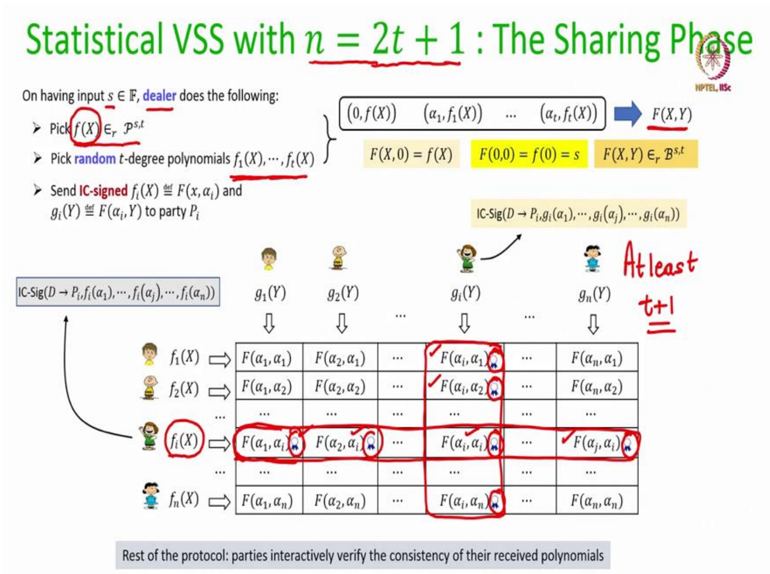
So, pi will have the primary share si. And P1 will have the secondary share si1 for the primary share si, P2 will have the secondary share si2 for the primary share si. The jth party will have a secondary share sij for the primary share si and the nth party will have a secondary share sin for the primary share si. So, that is why it is called 2D secret sharing because you have the secret being shared in one dimension through Shamir secret sharing and in the other dimension, each Shamir share is further Shamir shared.

We also have IC signatures on the secondary shares available. So, if I consider the ith party, it will have the full vector of secondary shares of si. So, it has not only the primary share si, the party Pi will also have the secondary shares si1, si2, sii, sij, sin and they will be individually IC signed by the respective secondary shareholder.

So, the secondary share S i1 will be IC signed with will be IC signed by party p 1. The ith secondary share will be IC signed with will be IC signed by the ith party P i. The jth secondary share will be IC signed by the party P j and the nth secondary share will be IC signed by the party P n.

So, that is the entire structure, entire data structure which will be generated by the statistical verifiable secret sharing scheme. I would also like to stress that in the in one of our earlier lectures we had seen that if you have a value s which is 2D secret shared with IC signatures, then we can robustly reconstruct it. And also recall that we will be using these notations for representing a 2D secret sharing of any value with IC signatures.

(Refer Slide Time: 04:23)



So, here is the sharing phase protocol. So, the dealer will have some value s from a finite field F which it would like to secret share. And the idea here will be similar to our perfectly secure verifiable secret sharing scheme, where dealer will first pick a Shamir secret Shamir sharing polynomial which is a random t degree polynomial whose constant term is the secret s.

And to prove that it is secret sharing its secret in a consistent way, what dealer is going to do is, it is going to embed this Shamir secret sharing polynomial f(X) in a random t degree bivariate polynomial F(X, Y). To pick this random t degree bivariate polynomial, the dealer will additionally pick t random univariate polynomials, each of degree t, and then using the Shamir sharing polynomial and the additional t univariate polynomials, it interpolates this bivariate polynomial F(X, Y).

The constant term of this bivariate polynomial will be the dealer's secret and rest of the coefficients of this bivariate polynomial will be random. Now as we have done in the perfectly secure VSS, the ith party will be provided the ith row and ith column polynomial on this bivariate polynomial. But now since we are in the statistical world and since we are now working with the condition n being 2t + 1, we are no longer in the setting where n is at least 3t + 1.

So, that is why to ensure robustness in the protocol what the dealer is going to do is, it is going to IC sign all the individual points on the ith row and ith column polynomial before giving those row and column polynomial to the ith party. So, if you imagine this n cross n matrix of values, matrix of points on the bivariate polynomial, then the ith party will be getting the ith row polynomial in the form of these points.

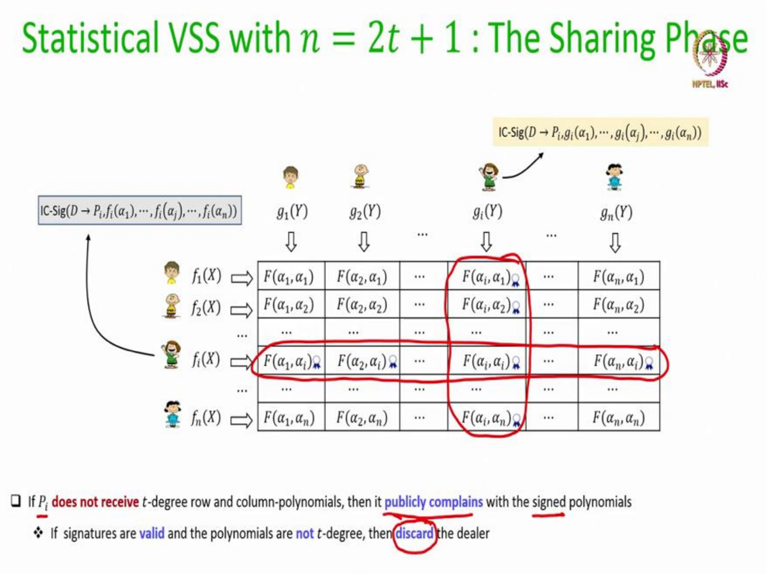
Namely the ith row which are the points on the ith row and all these points are distinct points on the bivariate polynomial F(X, Y) and at the same time the values along the ith column will be given to the ith party.

And dealer will also individually IC sign these points ok. So, each point is IC signed by the dealer. This is to ensure that later if there are any disputes and if Pi wants to claim anything regarding the values which it has got from the dealer, it can publicly show the IC signature of the dealer on those points and everyone can then verify the claim of the ith party if there are any disputes in the future rounds.

In the same way the distinct points on the ith column are also IC signed by the dealer. So, this notation here denotes that these points are IC signed and how they would have been IC signed by running an instance of the ICP which we had discussed in our earlier lecture.

Now once dealer has distributed the ith signed row and column polynomial to the ith party and it will be doing for each individual party, the rest of the protocol involves interaction among the parties to verify whether the dealer has distributed consistent polynomials to the honest parties. Because if it is guaranteed that dealer has distributed consistent polynomials to all the honest parties, since, we have at least t plus 1 honest parties in the system, it will be guaranteed that the row and column polynomials which those honest parties have received actually constitute distinct row and column polynomials on a unique t degree bivariate polynomial. And this comes because of the pair wise consistency lemma ok.

(Refer Slide Time: 09:00)



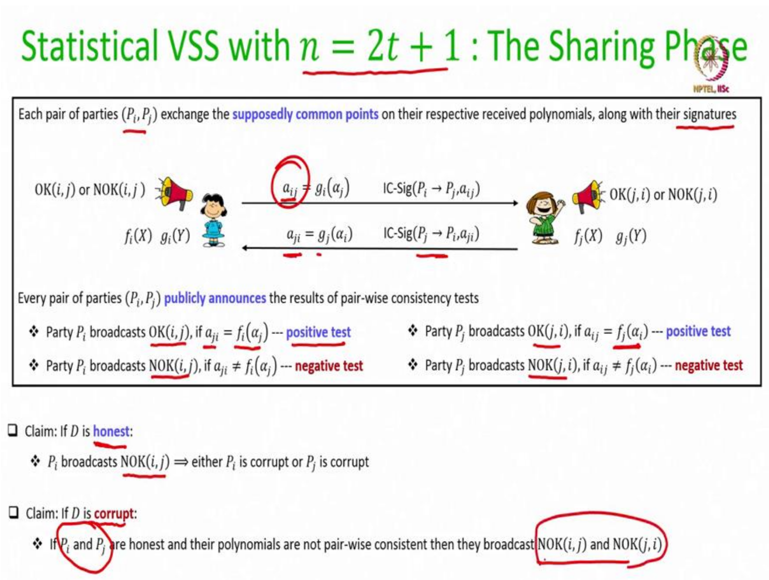
So, now let us see the remaining rounds of the protocol how exactly the pair wise consistency check happens and how the disputes are resolved and so on. So, the first thing which every party P i does is the following.

So, Pi would be receiving the points on the supposedly ith row polynomial and the points on the supposedly ith column polynomial of the dealer's bivariate polynomial. So, those points should lie individually on t degree polynomials, but if Pi finds that the row and column polynomial that it has received are not t degree polynomials then it publicly complains, namely by broadcasting the points that it has received on its row and column polynomials along with the signature of the dealer.

And now everyone can verify whether the signatures are correct. And if the signatures are correct and indeed if the points revealed by the ith party does not lie on t degree row or t degree column polynomial then everyone will conclude that dealer is corrupt and they will discard the dealer and stop the protocol there itself. And they will take some default value s on the behalf of the dealer.

So, it is easy to see that if the dealer is honest it will not be discarded with very high probability, because an honest Pi will never complain against an honest dealer. So, if the dealer is honest, a potentially corrupt Pi might try to unnecessarily complain against the dealer and it may try to reveal incorrect points and there is a non zero error probability that it is successfully able to forge dealers IC signatures on the points which dealer has not given to the ith party, in which case the parties may end up discarding an honest dealer. But the probability that a corrupt Pi can forge an honest dealer's signature on points which have not been provided by the dealer to the ith party is very negligible. So, if at all a dealer is discarded because of this check then it is guaranteed with a very high probability the dealer is corrupt.

(Refer Slide Time: 11:24)



Now, if the dealer is not discarded and if every party has locally checked that the row and column polynomials which they have received are t degree polynomials, then they go ahead with the pair-wise consistency check where every pair of parties exchange the supposedly common points on their respective polynomials.

But, now they also put their IC signatures on the common points. This is because looking ahead we want to generate a 2D secret sharing of dealers secret with IC signatures right.

So, the pair wise consistency test is very similar to what we have done for perfectly secure protocols except that in the perfect secure protocol the pair wise consistency check does not involve any kind of signatures. But since we are in the statistical world and we are working with n equal to 2t + 1 and we want to generate 2D secret sharing with IC signatures as part of the pair wise consistency test, the parties also put their individual IC signature. They give IC signatures on the supposedly common points. So, for instance party Pi will give the value of its column polynomial at alpha j, let us denote that value by aij, so that value it gives to Pj along with the IC signature. And whenever I say it gives the IC signature, remember that we have an instance of ICP running in the background where there will be a generation phase and a verification phase ok.

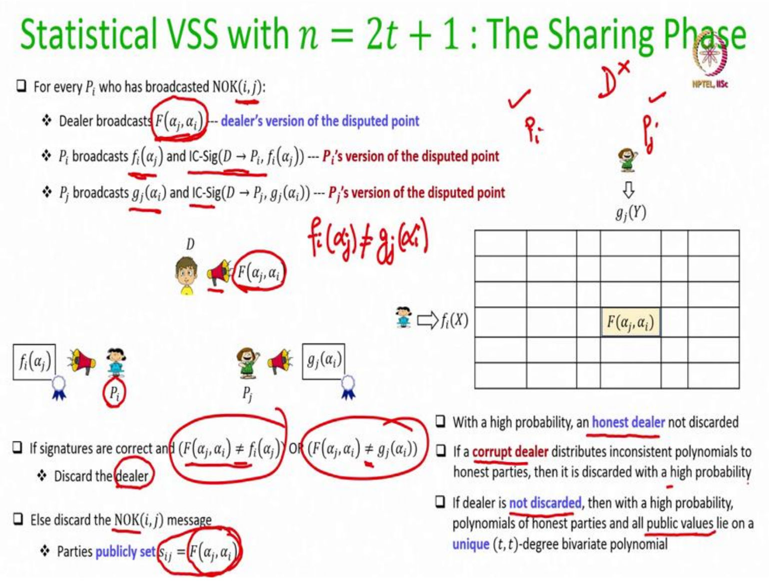
And later on if the signature needs to be revealed, the corresponding revelation phase of the ICP gets invoked. In the same way party Pj evaluates its column polynomial at alpha i and gives the resultant value aji to Pi, but now it also puts its IC signature, it gives its IC signature on that value. Now once the common points are exchanged, every pair of parties publicly announces the results of pair wise consistency test.

So, let us see what party Pi does. If it sees that the common point which it receives from P j also lies on its row polynomial then it is considered as a positive test in which case it broadcasts an OK message for the jth party, otherwise it broadcasts an NOK message for Pj. And similarly Pj checks whether the supposedly common point which it is receiving from Pi lies on Pj’s row polynomial. If it is the case then it is a positive test in which case an OK message is broadcasted otherwise an NOK message is broadcasted.

So, now let us make few claims regarding what happens after the pair wise consistency tests. So, if the dealer is honest and if some party Pi has broadcasted an NOK message against party Pj, then it implies that either the party Pi is corrupt or Pj is corrupt. Because if the dealer is honest and if both Pi and Pj are honest then the then the polynomials of the ith party and jth party will be pair wise consistent and they will only broadcast an OK message.

So, if at all an NOK message is broadcasted by Pi and if the dealer is honest, then at least one of the two parties Pi or Pj is corrupt. The second claim is that if the dealer is corrupt and if we consider a pair of honest parties Pi, P j and if their polynomials are not pair wise consistent then both of them will broadcast an NOK message against each other and this is very trivial to verify.

(Refer Slide Time: 15:32)



So, now once the OK and NOK messages are made public let us see what actions are taken by the parties. So, every party who has broadcast an NOK message, a complain message against Pj, for every such party Pi, the following is done. Dealer goes and makes public the dealer's version of the disputed point.

So, if at all there is a complaint by Pi against Pj, that means, Pi has checked and verified and is complaining that the point on the Pj’s column polynomial which is supposed to also lie on Pi’s row polynomial are not same. So, what dealer is going to now is, dealer is going to make the corresponding point public, that is the dealer’s version of the disputed point.

And when I say public; that means, by broadcasting. And in parallel the complainee and the complainant, the complainee here is party Pi, it makes public its version of the point which it has received in the first round from the dealer. And not only its version of the point, but also the dealers IC signature on that, because otherwise how anyone can verify indeed whether P i has received that point during the first round from the dealer.

So, to prove that indeed it has received the same point fi of alpha j from the dealer during the first round, it makes public the IC signature. And when I say makes public the IC signature; that means, the revelation phase of the corresponding ICP gets invoked.

And in parallel Pj makes public its version of the supposedly common point which is going to be a point on its column polynomial. And again to convince everyone that whatever point Pj is making public is indeed the one which it has received from the dealer during the first round, it makes public the corresponding IC signature on that point ok. So, that means, whenever now there is a dispute between Pi and Pj, the dealer's version of the point and Pi's version of the point and Pj's version of the point are now available in public.

And now we will check the following, if the signatures revealed by the complainee and complainant are correct; that means, they are verified and if it turns out that either the Pi's version or Pj's version does not match the dealer's version then discard the dealer, because dealer is corrupt and take some default value as the dealer's secret ok. So, with a very high probability, an honest dealer will not be discarded. Why so, because if the dealer is honest and if at all there is a complaint by Pi against Pj then one of the two parties Pi or Pj is corrupt. Then the only way an honest dealer will be discarded is when the corrupt party among Pi or Pj is able to forge dealer's signature on an incorrect version of the disputed point, which dealer has never given to that corrupt party. But the probability that a corrupt party can forge an honest dealer's signature is very negligible.

So, that means, if at all a dealer is discarded, it is corrupt. And because of this pair wise consistency check and the way we are resolving the disputes, it will be guaranteed that if a corrupt dealer has distributed inconsistent polynomials then it will be discarded with a very high probability.

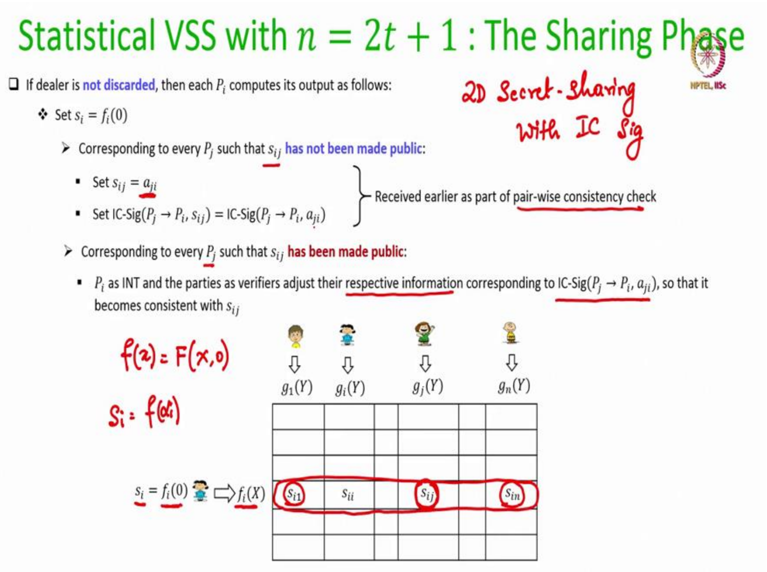
Because if the dealer is corrupt and say there is a pair of parties Pi, P j who are honest and they have received inconsistent polynomials, then Pi would have broadcasted an NOK message. And then when Pi makes public the assigned value and Pj makes public the assigned values, the signatures will be accepted with very high probability because of the non repudiation property of IC sig. And once the signatures are verified everyone will find out that either the dealer's version mismatches Pi's version or Pj's version.

Because it cannot be the case that the dealer's version matches both Pi's version as well as Pj's version, because we are considering the case where Pi and Pj are honest and fi of alpha j is not equal to gj of alpha i. If this is the case, then the dealer's version of the point can either be the same as Pi's version that is fi of alpha j or it can be the same as Pj's version. It cannot be same to both Pi's version as well as Pj's version and dealer will be discarded ok.

However, it could be possible that even though Pi has broadcasted an NOK message against Pj, when the dealer and the complainee and the complainant make their disputed points public, then all three of them turn out to be the same. If that is the case, then we simply discard the complaint and what we do in this case is, we set the jth secondary share of the ith primary share to be the value which dealer has made public ok.

So, to summarize if the dealer is not discarded then with a very high probability it will be guaranteed that the polynomials of all the honest parties plus all the public values which dealer has made public lie on a unique bivariate polynomial ok.

(Refer Slide Time: 21:52)



So, now let us see how the output is computed by individual parties. So, if the dealer is not discarded then every party Pi computes its output as follows. So, it takes its primary share to be the constant term of the row polynomial that it has received, and the secondary shares are computed as follows.

So, there might be some parties Pj corresponding to which dealer might have made the dealer's version of the disputed point public. And there might be some Pjs corresponding to which dealer has not made any point public. So, if we take such parties P j corresponding to which the secondary share sij has not been set yet, then what P i does is the following.

The common point on the Pjs column polynomial which P i has received as part of the pair wise consistency check earlier, that is said to be the jth secondary share of the ith primary share. And whatever signature Pj has provided to Pi on that common point that is taken as Pj's IC signature on the secondary share sij ok.

So, for instance if P1 has given the value aj1 to the party Pj then that value a j1 is taken as sj1. In the same way if say the nth party has given the common point ajn to the party Pj then that is set as sjn and so on. And the signatures would have been provided by P1,…, Pn respectively on those points. But there might be some parties P j corresponding to which the secondary shares has been already set because they have been made public ok.

So, if there is any such party P j then what P i now just have to do is it just has to adjust the information which it has received as an intermediary during the corresponding instance of ICP.

So, remember when Pj would have been doing the pair wise consistency check with Pi, P j would have given a common point on Pj's column polynomial to Pi. And Pi would have found a dispute with Pj. And then dealer would have made public sij and everyone would have set sij to be the jth secondary share for the ith primary share, that is fine.

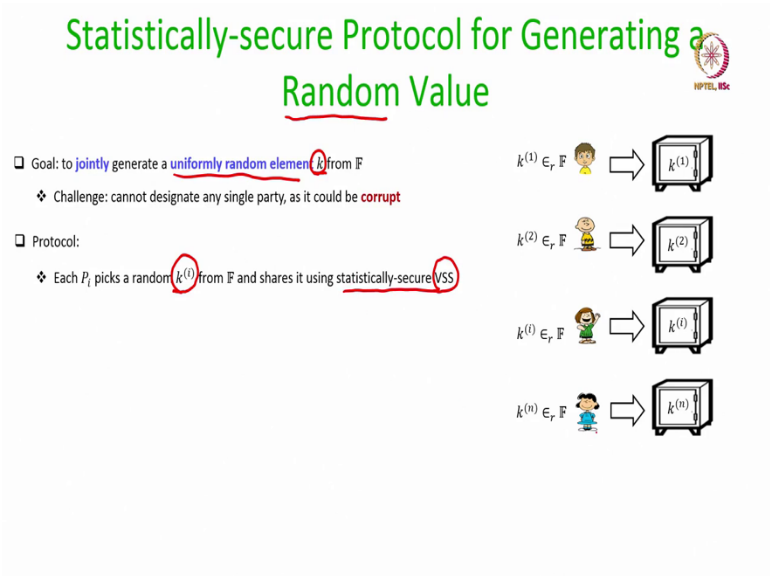
But, we also need P js signature on this new value sij ok. So, we cannot afford now to run a fresh instance of ICP and ask Pj to give a signature on sij to Pi. Rather what we do is whatever IC signature Pj has given on the old version of the common point, namely aji, as part of that instance of ICP, INTs role would have been played by Pi ok. So, INT would have received some authentication information in the form of mac tag and etcetera and other verifiers would have received verification information in the form of mac keys.

So, what the parties now do is, they simply adjust their respective information. When I say respective information, I mean INT adjust its mac tag and verifiers adjust their respective mac keys. So, that whatever signatures have been given in the earlier instance of ICP, it constitutes now a signature with respect to this new point, new value sij, ok. And with this the 2D secret sharing with IC signature is done, why so?

Because, it is indeed the case that the primary shares, which are s1, s2, si, sn, which are basically the constant term of individual row polynomials, they lie on the Shamir secret sharing polynomial f(x) which is the same as the bivariate polynomial of the dealer evaluated at y equal to 0. So, si is nothing, but f of alpha i. So, we have the primary shares lying on a t degree polynomial. And anyhow the secondary shares si1, si2, sij, sin, they lie on the t degree polynomial fi of x ok.

And indeed the jth secondary share is held by Pj. Of course, Pi holds all the secondary shares because it has the ith row polynomial. And if we take the individual secondary shares, they are IC signed and given to Pi, ok. Because those IC signatures would have been given to Pi as part of the consistency check, right.

(Refer Slide Time: 27:34)



So, that completes the description of the statistical verifiable secret sharing. Of course, I have not focused here on optimizing the number of rounds, communication complexity and so on. And there are various ways to further optimize this protocol in terms of the number of communication rounds and the amount of communication involved.

Now, let us discuss a statistically secure protocol for generating a random value, you might be wondering why suddenly we require this protocol. So, remember as part of the statistical polynomial verification protocol there was a step in that protocol where the parties need to publicly generate a random value which should not be known to the adversary beforehand. So, a very simple way to do that is the following.

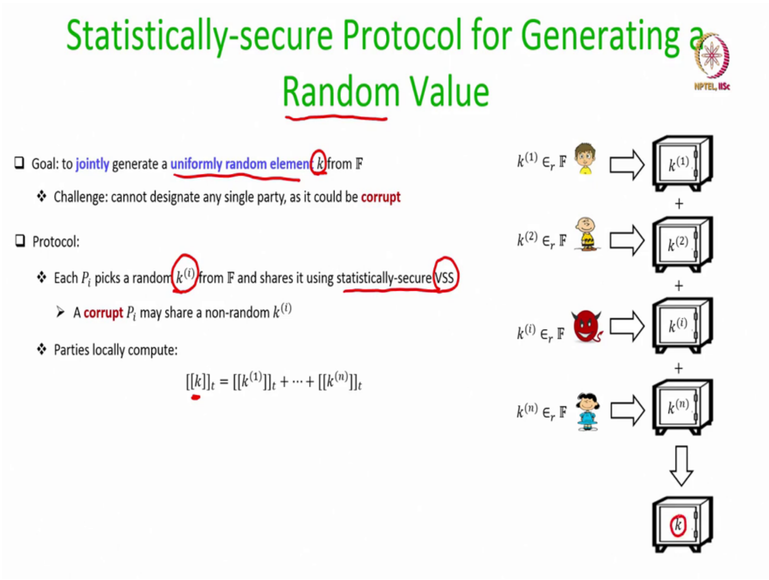
So, the goal here is to jointly generate a uniformly random element, say k, from the field. The challenge here is that we cannot designate this task to any single party in the system. We cannot say that let Pi be the designated party who should pick a value k and broadcast it. Because if Pi is under the control of the adversary, then adversary will be knowing this value k beforehand which we do not want ok.

And the protocol here is the following: instead of asking any single party to pick a random value and make it public, we ask each party to pick a random value and instead of making it public, rather secret share it, using an instance of the statistical verifiable secret sharing which we had discussed just now. You might be wondering why cannot we ask every party Pi to make its contribution ki public. Because if we do that then again there is a very nice attack here based on the rushing nature of the adversary.

So, what the adversary can do is it can first wait to listen to the contributions of the honest participants and once it listens to the contribution of the honest participants, it can pick its own contribution, so that its own contribution along with the other parties' contribution gives a value of adversaries choice, which the adversary can easily fix. Again in that case the resultant value will not be a random value.

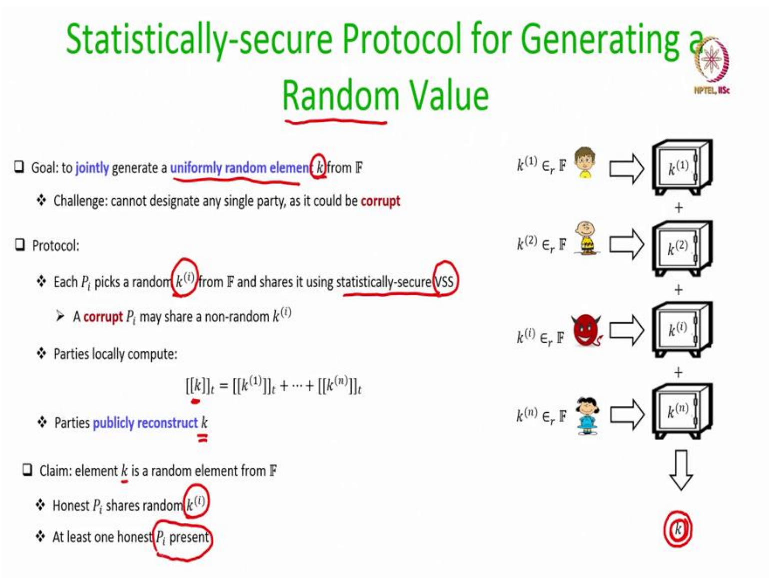
So, that is why we are asking now each individual party to pick a random value and instead of making it public, rather secret share them. The since the honest parties will be secret sharing their contributions, their input, using an instance of VSS, adversary will have absolutely no idea what exactly are the random values picked by the honest parties. But at the same time adversary will be forced here to pick some value and secret share.

(Refer Slide Time: 30:37)



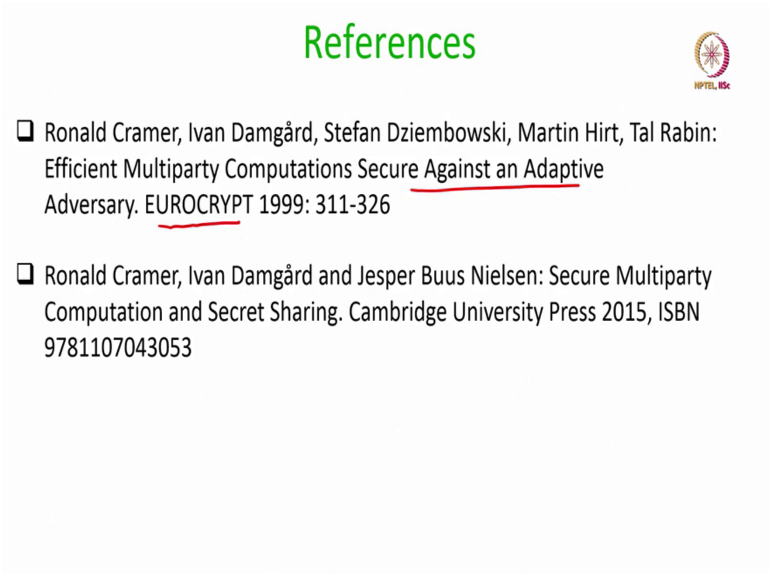
Now, once all the values have been secret shared, what we do here is the following: We set the resultant value, which is going to be the output of the protocol, to be the sum of all the contributions, namely the sum of the individual values which have been secret shared by the respective parties. And because of the linearity property of 2D secret sharing with IC signatures, this value k can be computed in a non interactive way by just performing the linear operation on the shares of k1, k2, kn. So, now the value k is secret shared, but we would like the value k to be available in a public fashion.

(Refer Slide Time: 31:21)



So, what we do is, we run the reconstruction protocol to publicly reconstruct the value k. And now the claim is that this element k is a random element from the field and this simply comes from the fact that the honest parties in the system, they pick uniformly random values and the values which have been shared by the honest parties they will be not known when they are secret shared, to the adversary. Because that comes due to the from the privacy property of VSS. And once all the values are fixed and added then only the resultant value k is reconstructed. Now since the honest parties in the system secret shares random values, even if the corrupt parties secret non random values, we have now a bunch of random values, added with a non random value. So, the resultant sum, its probability distribution will be a uniform distribution. And it is guaranteed that we have at least one honest party in the system who is going to secret share a random value ok.

(Refer Slide Time: 32:26)



So, with that I end this lecture. So, the verifiable secret sharing protocol which I had discussed is taken from this paper. Of course, there are several optimizations possible for this protocol in terms of number of rounds and communication complexity.

Thank you.