


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

Lecture - 41
Perfectly-Secure VSS with n greater than $3t$: A Round-Reducing Technique

(Refer Slide Time: 00:25)

Lecture Overview



- Perfectly-secure VSS with $n > 3t$ based on bivariate polynomials
 - ❖ A round-reducing technique

Hello everyone, welcome to this lecture. So, in this lecture, we will consider the Perfectly-Secure VSS Scheme which we had discussed in the last lecture and we will introduce a round reducing technique, which is very often used in the domain of verifiable secret sharing.

(Refer Slide Time: 00:40)

VSS Scheme with $n > 3t$: Analysis

R1: On having input $s \in \mathbb{F}$, dealer does the following:

- Pick $f(X) \in \mathcal{P}^{k,t}$
- Pick $F(X, Y) \in \mathcal{B}^{k,t}$, such that $F(X, 0) = f(X)$
- Send $f_i(X) \equiv F(X, \alpha_i)$ and $g_i(Y) \equiv F(\alpha_i, Y)$ to party P_i

R2: Every party P_i sends the following to every party P_j : $a_{ij} = f_i(\alpha_j)$ and $b_{ij} = g_i(\alpha_j)$

R3: Every party P_i upon receiving (a_{ij}, b_{ij}) broadcasts the following corresponding to every party P_j

- OK(i, j), if $a_{ji} = b_{ij}$ and $b_{ji} = a_{ij}$
- NOK(i, j, a_{ij}, b_{ij}), if $a_{ji} \neq b_{ij}$ or $b_{ji} \neq a_{ij}$

R4: For every P_i who has broadcast NOK(i, j, a_{ij}, b_{ij}) during R3, the dealer and party P_j does the following:

- Dealer broadcasts $F(\alpha_j, \alpha_i)$ and $F(\alpha_i, \alpha_j)$
- P_j broadcasts $g_j(\alpha_i)$ and $f_j(\alpha_i)$
- ❖ $P_i \in \mathcal{UH}$, if $a_{ij} \neq F(\alpha_j, \alpha_i)$ or $b_{ij} \neq F(\alpha_i, \alpha_j)$
- ❖ $P_j \in \mathcal{UH}$, if $g_j(\alpha_i) \neq F(\alpha_j, \alpha_i)$ or $f_j(\alpha_i) \neq F(\alpha_i, \alpha_j)$

If $|\mathcal{UH}| > t$, then **discard the dealer** and take a default Shamir-sharing of 0 on the behalf of the dealer

R5: For every **unhappy party** $P_i \in \mathcal{UH}$, the dealer and every **happy party** $P_j \notin \mathcal{UH}$ does the following:

- Dealer broadcasts $F(X, \alpha_i)$ and $F(\alpha_i, Y)$
- P_j broadcasts $g_j(\alpha_i)$ and $f_j(\alpha_i)$
- ❖ For every **unhappy party** $P_i \in \mathcal{UH}$ for which the dealer has broadcast $F(X, \alpha_i)$ and $F(\alpha_i, Y)$, check if there are at least $(2t + 1)$ parties $P_j \notin \mathcal{UH}$ who broadcasted $(g_j(\alpha_i), f_j(\alpha_i))$, such that:
- $F(\alpha_j, \alpha_i) = g_j(\alpha_i)$
- $F(\alpha_i, \alpha_j) = f_j(\alpha_i)$

discard the dealer, if the condition is not satisfied for ANY $P_i \in \mathcal{UH}$

Output decision: If the dealer is **not discarded**, then every P_i outputs its **shares** as follows

- ❖ If $P_i \in \mathcal{UH}$, then $s_i = F(\alpha_i, 0)$, where $F(\alpha_i, Y)$ is the polynomial broadcasted by the dealer during R5
- ❖ If $P_i \notin \mathcal{UH}$, then $s_i = g_i(0)$, where $g_i(Y)$ is the polynomial received from the dealer during R1

Round complexity:

- ❖ 5 rounds
- Can be **reduced to 4 rounds**

Idea:

- ❖ **Prepone** the pair-wise consistency test, so that the results are **publicly available** at the end of **R2** itself
- ❖ Every pair of parties exchange **random OTPs in advance** during **R1**, which are **later** used to broadcast **OTP encryptions** of supposedly common points

So, this is the five round VSS scheme; namely, the sharing phase protocol and we want to reduce the number of rounds to 4 and the idea here will be the following. Somehow, we want to postpone the pair wise consistency check so that the results of the pair wise consistency check are available at the end of second round itself.

So, currently, the results of the pair wise consistency check are available at the end of third round, due to which during the fourth round only we are able to identify the happy and unhappy parties and then, during the fifth-round dealer is resolving the polynomials of the unhappy parties.

Somehow, we want to ensure that the results of the pair wise consistency check are made available at the end of the second round itself. But by looking at the protocol that seems to be an impossible task because only when the dealer makes dealer gives the polynomials and then, the parties exchange their common points, they can come to know whether their polynomials are pair wise consistent not and that is what currently is happening.

During the round one right, dealer is giving the polynomials to every pair of parties P_i, P_j and then, during the second round P_i and P_j talk with each other, talk to each other, and find out what exactly is the status of the common points. And then, they publicly go and announce in the third round whether they are OK or NOK right. What we are proposing here seems to be an impossible task.

We want a mechanism so that at the end of the second round itself, it should be publicly known whether P_i and P_j are in dispute with each other or not. Well, we can do something interesting here. The idea will be to utilize the first round itself and ask every pair of parties P_i and P_j to exchange some random information among themselves in advance, which can be then later utilized during the round 2 to do the pair wise consistency check of the polynomials and that too publicly.

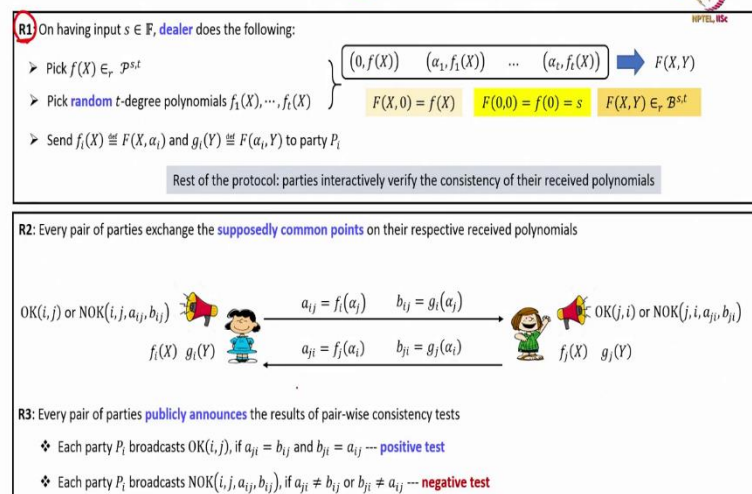
So, right now, during the round 2, the pair wise consistency check is happening privately over the pair wise communication channels and the results are made public during the round 3. What we are trying to do here is incorporate a mechanism where the pair wise consistency check happens publicly, but in the form of encryptions.

Namely, one time pad encryptions of the common values and those one-time pads, we are asking the parties to pre-exchange during the round 1 itself because during the round 1, there is no communication happening between any pair of parties currently. It is only the dealer distributing the polynomials.

So, the parties can utilize the first round and pre-exchange some common random one-time pads which they can later utilize to make public, the common points on their respective polynomials in a massed fashion. So, let us go into the details.

(Refer Slide Time: 04:08)

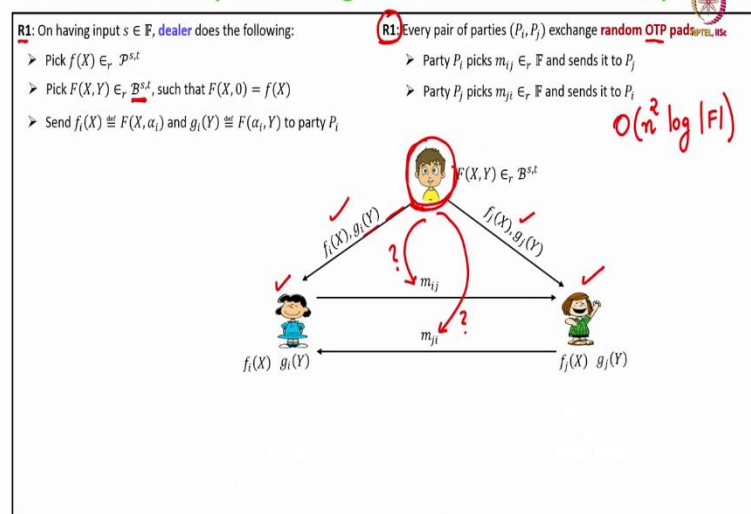
VSS Scheme: The Existing Pair-wise Consistency Test



So, this is the existing pair wise consistency check, the way it is currently happening. During round 1, dealer is distributing the polynomials and then, during round 2, every pair of parties P_i and P_j exchange the supposedly common points and then, during round 3, P_i and P_j make public the OK or the NOK messages. The point to note here is that during round 1, there is no communication happening between every pair of parties P_i, P_j . It is only dealer distributing the polynomials.

(Refer Slide Time: 04:45)

The New Way of Doing Pair-wise Consistency Test



So, the new way of doing the pair wise consistency check which will finally end in a four round sharing phase protocol will be the following. In round 1, dealer will be distributing the row and column polynomials on it by on its bivariate polynomial, as it was doing earlier. So, it will pick a random t degree bivariate polynomial. To the i th party, it will give the i th row and column polynomial; to the j th party, it will give the j th row and column polynomial.

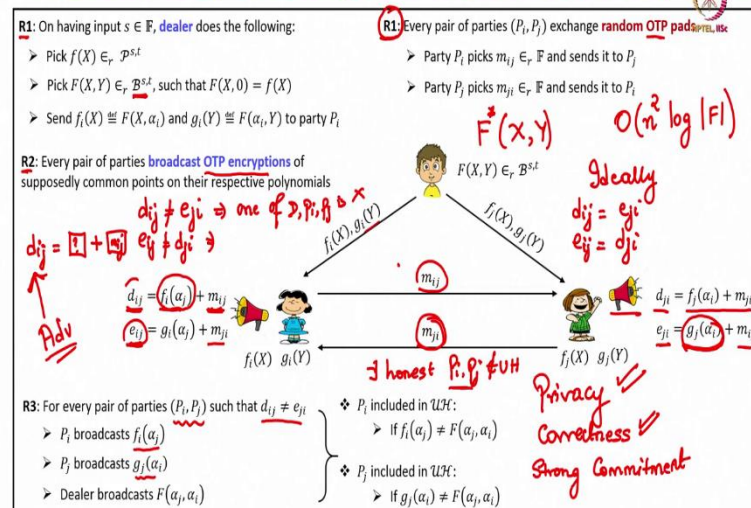
But now, in parallel in the same round, every pair of parties P_i, P_j exchange random one-time pad encryption pads, OTP pads. What are those OTP pads? They are basically random field elements. So, P_i will send a random field element m_{ij} to P_j and P_j in parallel will pick a random field element and send to P_i .

Notice that irrespective of whether dealer is honest or corrupt, it has absolutely no information regarding the pads which are exchanged between honest pair of parties P_i, P_j ;

Because these pads are exchanged over the pair wise communication channel available between P_i and P_j . This round, this additional communication, was not present in the earlier protocol; but now to reduce the number of rounds in the sharing phase, we are incorporating this additional communication. How much this additional communication will cost? Well, this will cost a communication of $n^2 \log |\mathbb{F}|$ number of bits.

So, we are not overshooting the existing communication complexity, the overall communication complexity of the protocol remains the same. So, at the end of the first round, now what is the addition what is the information available with the parties? Well, every pair of parties will have their respective row and column polynomials; but now, they also have the private pads private masks which they have exchanged among themselves.

The New Way of Doing Pair-wise Consistency Test



Now, during the round 2, what every pair of parties can do is they can broadcast OTP encryptions; one time pad encryptions of the supposedly common points on their respective polynomials. So, earlier in the round 2, P_i would have sent a common point on its polynomials to P_j and P_j would have sent the common points on its polynomial to P_i and then, in the third round, P_i would have sent OK or NOK messages.

We are not doing that now. In the round 2, what P_i is doing is it is simply taking the j th point on its row polynomial and masking it with the one-time pad which it has selected and making that one-time pad encryption public through the broadcast channel. It is also taking the j th point on its column polynomial and it is not giving it now to P_j ; rather it is masking it with the pad which P_j has selected for P_i and the OTP encryption, it is making public.

In parallel, in the same round what is P_j doing? Earlier, it would have sent the i th point on its row and column polynomial to the party P_i ; but now, it is not doing that, but rather it is now computing one time pad encryptions of those common points using the pad which it has received from P_i and using the pad which it has sent to P_j and corresponding OTP encryptions, it is making public through the broadcast channel.

Now, at the end of round 2, everyone will know d_{ij} , e_{ij} ; d_{ji} , e_{ji} . At the end of round 2, everyone will now learn whether there is a dispute between party P_i or P_j or not; between P_i P_j or not. Namely, if d_{ij} is not equal to e_{ji} , then it automatically implies that one of P_i , P_j is corrupt. In the same way, if e_{ij} is not equal to d_{ji} , the same conclusion can be drawn.

That means, there is a dispute between P_i and P_j . So, now, you can see just based on these OTP encryptions at the end of round 2 itself parties come to know whether there is a dispute between the party P_i and P_j or not. Ideally if the dealer is honest, if P_i is honest and if P_j is honest, then ideally if all the three parties, all the three entities dealer P_i and P_j are honest, then we expect d_{ij} to be same as e_{ji} . Because the $f_i(\alpha_j)$ will be same as $g_j(\alpha_i)$ and whatever pad P_i has communicated to P_j would have used the same pads.

So, ideally d_{ij} should be same as e_{ji} and using the same argument if dealer P_i and P_j are honest, then e_{ij} is supposed to be same as d_{ji} . That means, if any of these two conditions does not hold, then we can definitely say that there is a dispute between P_i and P_j which further implies that at least one of these three entities is corrupt.

So, the way the disputes are now resolved is as follows. So, for every pair of parties P_i and P_j , for every pair of conflicting parties to be more specific, such that d_{ij} is not equal to e_{ji} , the dealer party P_i and party P_j makes public their respective version of the disputed point.

So, if d_{ij} is not equal to e_{ji} , then automatically implies that P_i feels that $f_i(\alpha_j)$ is different from $g_j(\alpha_i)$.

So, P_i goes and makes public its version of the disputed point; namely, the j th point on its row polynomial and P_j also should now respond because it is a complaint against P_j . So, P_j goes public and P_j makes public the i th point on its column polynomial and dealer should also now come into picture and make public the dealer's version of the disputed point.

And now, we can easily identify the set of unhappy parties. If the dealer's version does not match with P_i 's version of the disputed point; that means, dealer is not taking the side of P_i . So, P_i will be considered as unhappy; whereas, if the dealer's version of the disputed point does not match the P_j 's version that means dealer is not taking the side of P_j , then P_j will be considered as an unhappy.

So, now, you can see that at the end of round 3 itself we have identified the set of happy and unhappy parties and now, whatever action was performed during round 5, namely dealer was making public the polynomials of the unhappy parties in the round 5 and every parties, who are outside unhappy to ensure that dealer does not cheat.

They were making public their version of the supposedly common points on those polynomials that step will now be executed during the round 4 and accordingly, we can identify; we can decide whether to discard the dealer or not and so on. So, this will be the four round sharing phase protocol. It is easy to see that all the properties that we had for the previous protocol are maintained.

So, the privacy is maintained. Why is privacy maintained? Because if the dealer is honest and if P_i , P_j are honest, then even though the OTP encryptions of the common points between P_i and P_j are public; that does not reveal it does not reveal anything about the supposedly common points because the corresponding pads are exchanged privately between P_i and P_j .

So, even though d_{ij} is public, it does not leak anything about what is $f_i(\alpha_j)$ because it could be any $f_i(\alpha_j)$ padded with any m_{ij} . So, if the dealer is honest if P_i is honest and if P_j is honest and if there is an adversary, who now has access to d_{ij} , it cannot tell what

exactly the value $f_i(\alpha_j)$ was and same holds for e_{ij} as well. Because the corresponding pad m_{ji} is not known to the adversary.

So, even though now the pair wise consistency check is happening over the public channel, the common points between the common points on the polynomials of honest parties between every pair of honest parties is not revealed. So, privacy is achieved, correctness is anyhow achieved because no honest party will make it to the unhappy set and remaining arguments remain the same and strong commitment is also achieved.

Because again, we can argue that if the number of unhappy parties is at most t ; that means, there are at least $t + 1$ or more number of honest happy parties, their polynomials are pairwise consistent. Because if there exist some honest P_i, P_j not belonging to the unhappy set, whose polynomials are not pairwise consistent; then it will be publicly identified because either d_{ij} will not be equal to e_{ji} or e_{ij} will not be equal to d_{ji} and then, dealer has to come into picture and resolve the conflict and either it can make P_i happy or P_j happy; it cannot make both of them happy.

So, one of them will end up being unhappy, but that is against this assumption that both P_i and P_j are outside the unhappy set. So, if the dealer is not discarded, then there will be at least $t + 1$ happy and honest parties, whose polynomials will be pairwise consistent and they will lie on a single t degree bivariate polynomial, call it as $f^*(X, Y)$.

And then, rest of the arguments remain the same as we had for the strong commitment property for the earlier protocol. I am not going through that argument for the to avoid the sake for the sake of avoiding repetition.

(Refer Slide Time: 16:33)

References



- ❑ Rosario Gennaro, Yuval Ishai, Eyal Kushilevitz, Tal Rabin: The round complexity of verifiable secret sharing and secure multicast. STOC 2001: 580-589
- ❑ Anirudh Chandramouli, Ashish Choudhury, Arpita Patra: A Survey on Perfectly-Secure Verifiable Secret-Sharing. ACM Computing Surveys, 2022

So, that is a cute round reducing technique and as I said earlier it is a very common technique utilized to reduce the number of rounds in the sharing phase protocol of the VSS scheme based on bivariate polynomials, where the parties have to do the pair wise consistency check of their row and column polynomials. Namely, during the first round itself pre-exchange random one-time pads and use it to publicly perform the pair wise consistency check during the from the second round onwards.

With that, I end this lecture. These are the references used. The round reducing technique was introduced in this paper and you can find a survey on perfectly secure verifiable secret sharing schemes in this ACM computing surveys is given.

Thank you.