


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

**Lecture - 61**  
**Cryptographically-secure VSS and MPC**

Everyone welcome to this lecture.

(Refer Slide Time: 00:23)

## Lecture Overview



- ❑ Cryptographically-secure VSS scheme
  - ❖ Linearity property
- ❑ Cryptographically-secure MPC

So, in this lecture we will briefly discuss the Cryptographically secure VSS scheme based on Pedersen's commitment and how we can use it to get a cryptographically secure multi party computation.

(Refer Slide Time: 00:35)

## Pedersen VSS: The Sharing Phase

**Public setup:**

- Finite field  $(\mathbb{F}, +, \cdot)$ , where  $|\mathbb{F}| > n$
- Generator  $g$  for the cyclic group  $\mathbb{F}^* \cong \mathbb{F} - \{0\}$
- A random  $h \in \mathbb{F}^*$

To share a secret  $s \in \mathbb{F}$ , dealer does the following:

- Select random  $t$ -degree polynomials:
  - $F(Z) = s + a_1Z + \dots + a_tZ^t$  - Sharing poly
  - $R(Z) = r + b_1Z + \dots + b_tZ^t$  - Randomness
- Send the following to each  $P_i$ :
  - $s_i \triangleq F(a_i)$
  - $r_i \triangleq R(a_i)$

So, the Pedersen VSS scheme is based on Shamir secret sharing coupled with Pedersen's commitment scheme which is used for verifiability purpose. The public setup will be that of a finite field of size at least  $n + 1$  and it is a well known fact that if we focus on the non zero elements of this field then with respect to the multiplication operation it constitutes a cyclic group.

So, let me denote it by  $F^*$ , it will have a generator  $g$ . So, the generator  $g$  also is publicly available. Apart from that, a random element  $h$  from this group  $F^*$ . So, this  $F^*$  will be a cyclic group. To share a secret  $s$  from the field, the dealer will do the following it will share the secret as per the Shamir's secret sharing scheme.

So, it will pick a random  $t$  degree polynomial whose constant term is the secret  $s$ . But to prove that it has distributed points on a  $t$  degree polynomial to all the honest parties for the verification purpose it additionally picks now a random  $t$  degree univariate polynomial as well. And now to each party it gives the evaluation of the sharing polynomial and the randomness polynomial. So, I am calling the  $F$  polynomial as the sharing polynomial and  $R$  polynomial as the randomness polynomial.

(Refer Slide Time: 02:30)

## Pedersen VSS: The Sharing Phase

**Public setup:**

- Finite field  $(\mathbb{F}, +, \cdot)$ , where  $|\mathbb{F}| > n$
- Generator  $g$  for the cyclic group  $\mathbb{F}^* \cong \mathbb{F} - \{0\}$
- A random  $h \in \mathbb{F}^*$

To share a secret  $s \in \mathbb{F}$ , dealer does the following:

- Select random  $t$ -degree polynomials:
  - $F(Z) = s + a_1Z + \dots + a_tZ^t$
  - $R(Z) = r + b_1Z + \dots + b_tZ^t$  (Random polynomial)
- Send the following to each  $P_i$ :
  - $s_i \leftarrow F(a_i)$
  - $r_i \leftarrow R(a_i)$
- Publish the commitments of the coefficients publicly:
  - $C_{s,r} \leftarrow \text{PedCom}(s, r) \cong g^s h^r$
  - $C_{a_1, b_1} \leftarrow \text{PedCom}(a_1, b_1) \cong g^{a_1} h^{b_1}$
  - $\vdots$
  - $C_{a_t, b_t} \leftarrow \text{PedCom}(a_t, b_t) \cong g^{a_t} h^{b_t}$

Claim: If  $D$  is honest, then the view of the adversary is independent of  $s$ .

Rest of the protocol: to verify if  $D$  has shared consistent shares

Additionally it also makes public the commitment of each of the coefficients of the sharing polynomial as per the Pedersen's commitment scheme. So, what are the coefficients of the sharing polynomial?  $s, a_1, a_2, \dots, a_t$ , so they are individually committed as per the Pedersen's commitment scheme.

Now recall in the Pedersen's commitment scheme we also need a randomness to commit a value. So, if you see here closely what are the randomness used here in the individual commitments, they are the coefficients of the randomness polynomial. So, that is why we have a sharing polynomial and we have a randomness polynomial and these commitments are available publicly.

So, that will be broadcasted by the dealer using any reliable broadcast protocol ok and the  $i$ th point on the sharing polynomial and the randomness polynomial will be given to the  $i$ th party. Now this protocol is unlike your bivariate polynomial based VSS schemes which we had seen in the perfect security world and the statistical security world. Their to prove the the dealer is following the protocol properly, it has distributed consistent polynomials, we used the pairwise consistency properties of bivariate polynomial. But here we do not require any bivariate polynomials. So, we have an instance of Shamir secret sharing. In fact, we have two instances of Shamir secret sharing running, one with respect to the polynomial  $F$ ; one respect to the polynomial  $R$  and the additional thing here is that the commitments of the coefficients for the polynomial  $F$  are made public which is used to

verify whether dealer has distributed points on a single  $t$  degree polynomial to the individual parties.

Now, let us make few claims here first before going further. If the dealer is honest then the view of the adversary will be independent of dealer's secret. So, there could be up to  $t$  corrupt parties. From the privacy property of Shamir's secret sharing, the  $t$  shares reveal nothing about dealer's secret, but here the commitments of the coefficients of  $F$  polynomial are also public.

But those coefficients are committed using the coefficients of another random  $t$  degree polynomial. And the hiding property of the Pedersen's commitment scheme guarantees that these commitments does not reveal anything to the corrupt parties about the secret  $s$ . Now the rest of the protocol will be to verify if the dealer has distributed consistent shares, namely whether it has used a  $t$  degree polynomial  $F$  and a  $t$  degree polynomial  $R$  and distributed the points on those polynomials to all the honest parties, ok.

(Refer Slide Time: 05:54)

### Pedersen VSS: The Sharing Phase

□ To share a secret  $s \in \mathbb{F}$ , dealer does the following:

- ❖ Select **random**  $t$ -degree polynomials:
  - $F(Z) = s + a_1Z + \dots + a_tZ^t$
  - $R(Z) = r + b_1Z + \dots + b_tZ^t$
- ❖ Send the following to each  $P_i$ :
  - $s_i \cong F(\alpha_i) \quad r_i \cong R(\alpha_i)$
- ❖ Publish the **commitments** of the coefficients **publicly**:
  - $C_{s,r} \cong g^s h^r \quad C_{a_1,b_1} \cong g^{a_1} h^{b_1} \quad C_{a_t,b_t} \cong g^{a_t} h^{b_t}$

□ If  $D$  has distributed **consistent** shares, then:

$$\begin{aligned} s_i &= s + a_1\alpha_i + \dots + a_t\alpha_i^t \\ r_i &= r + b_1\alpha_i + \dots + b_t\alpha_i^t \end{aligned} \Rightarrow g^{s_i} h^{r_i} = (g^s h^r) \cdot (g^{a_1} h^{b_1})^{\alpha_i} \cdot \dots \cdot (g^{a_t} h^{b_t})^{\alpha_i^t}$$

$$= (C_{s,r}) \cdot (C_{a_1,b_1})^{\alpha_i} \cdot \dots \cdot (C_{a_t,b_t})^{\alpha_i^t}$$

□ If  $P_i$  broadcasts (NOK,  $i$ ), then either  $D$  is corrupt or  $P_i$

(NOK,  $i$ ), if

$$g^{s_i} h^{r_i} \neq (C_{s,r}) \cdot (C_{a_1,b_1})^{\alpha_i} \cdot \dots \cdot (C_{a_t,b_t})^{\alpha_i^t}$$

Claim: If  $D$  has distributed inconsistent shares to an honest  $P_i$ , then with a high probability  $P_i$  broadcasts (NOK,  $i$ )

So, the idea behind the consistency check is based on the following observation. If the dealer has distributed consistent shares to the honest parties, then for every honest party  $P_i$ , the following should hold: what would be the share of  $s_i$ , if indeed  $s_i$  is a point on the  $F$  polynomial and  $r_i$  is the point on the  $R$  polynomial. Then  $s_i$  will be this value and  $r_i$  will be this value. Now notice that this  $i$ th party, it will have only  $\alpha_i$ , it will have only  $s_i$ , it will have only  $r_i$  and of course, it will have this full commitment vector.

It does not know the full  $F$  polynomial and it does not know the full  $R$  polynomial. How it can verify whether indeed  $s_i$  is equal to  $F$  of  $\alpha_i$  and  $r_i$  is equal to  $R$  of  $\alpha_i$ , where the  $R$  polynomial and  $F$  polynomial coefficients are committed. So, notice that  $s_i$  and  $r_i$  satisfy these conditions.

That means if we compute  $g$  to the power  $s_i$  times  $h$  to the power  $r_i$ , then that will be same as this. Now if I closely observe these products then I can consider them to be the product of the commitment of  $s$  multiplied by the commitment of  $a_1$  raised to the power  $\alpha_1$ , sorry raise to the power  $\alpha_i$  times commitment of  $a_2$  raised to the power  $\alpha_i^2$  and like that commitment of  $a_t$  raised to the power  $\alpha_i^t$ .

So, now what are the things available with  $P_i$ ? So,  $P_i$  has  $s_i$  and  $r_i$ . So, it can compute this and it also has these commitments available, the individual commitments of the coefficients of the  $F$  polynomial and anyhow it knows  $\alpha_i$ . So, what  $P_i$  does is the following, it checks whether this condition holds or not. If it holds then its fine otherwise it publicly complains against the dealer.

And the claim here is that if the dealer is corrupt and it has distributed inconsistent shares to an honest  $P_i$ . That means,  $s_i$  and  $r_i$  which it has given to  $P_i$  does not lie on the polynomials which it has publicly committed, then with a very high probability  $P_i$  will broadcast an nok message. You might be wondering why with a high probability, why not always? Because it might be possible for a corrupt dealer to break the binding property of the commitment scheme.

And if it is possible for the dealer to break the binding property of the commitment scheme then even though it might have given inconsistent shares to  $P_i$ , this check may end up getting passed. But what is the probability that a corrupt dealer can break the binding property of the commitment scheme, it is very very small, assuming that it cannot solve a random instance of discrete log ok.

So; that means, if any party complains against the dealer, we can conclude that either the dealer is corrupt or it could be possible that dealer is honest, but this party  $P_i$  is corrupt and unnecessarily accusing the dealer right. So, it could be possible the dealer is honest, it has distributed consistent shares to everyone, but still this party  $P_i$  simply complains no, the check is not passing.

(Refer Slide Time: 10:07)

### Pedersen VSS: The Sharing Phase

□ To share a secret  $s \in \mathbb{F}$ , dealer does the following:

- ❖ Select **random**  $t$ -degree polynomials:
  - $F(Z) = s + a_1Z + \dots + a_tZ^t$
  - $R(Z) = r + b_1Z + \dots + b_tZ^t$
- ❖ Send the following to each  $P_i$ :
 
$$s_i \triangleq F(\alpha_i) \quad r_i \triangleq R(\alpha_i)$$
- ❖ Publish the **commitments** of the coefficients **publicly**:
 
$$C_{s,r} \triangleq g^s h^r \quad C_{a_1,b_1} \triangleq g^{a_1} h^{b_1} \quad C_{a_t,b_t} \triangleq g^{a_t} h^{b_t}$$

□ If more than  $t$  parties broadcast **(NOK,  $i$ )**, then **discard  $D$**

□ Corresponding to every **(NOK,  $i$ )**, dealer makes **public**  $(s_i, r_i)$

- ❖ Parties publicly check and **discard  $D$**  if:
 
$$g^{s_i} h^{r_i} \neq (C_{s,r}) \cdot (C_{a_1,b_1})^{\alpha_i} \dots (C_{a_t,b_t})^{\alpha_i^t}$$
- ❖ An **honest**  $D$  never gets discarded

$(\text{NOK}, i)$ , if  
 $g^{s_i} h^{r_i} \neq (C_{s,r}) \cdot (C_{a_1,b_1})^{\alpha_i} \dots (C_{a_t,b_t})^{\alpha_i^t}$

Claim: If a **corrupt**  $D$  is not discarded, then with a high probability, it has distributed consistent shares to honest parties

Claim: If  $D$  is **honest**, then the view of the adversary remains independent of  $s$

Whenever there is a complaint we do the following. So, first of all we check whether there are more than  $t$  complaints against the dealer. If that is the case then we can simply and safely discard the dealer and terminate the protocol there itself. Because if the dealer is honest it will never get discarded because in that case no honest party will complain against the dealer. Only potentially corrupt parties, corrupt shareholders can complain against the dealer and there could be at most  $t$  corrupt shareholders.

Now if there are at most  $t$  complaints, how the complaints are resolved publicly? Corresponding to every complaint dealer makes public the shares  $s_i$  and  $r_i$  by using a reliable broadcast protocol. But wait we cannot blindly believe the dealer right, we cannot simply take oh dealer you have made the shares of the  $i$ th party public, we agree fine that is correct.

We have to check whether indeed the publicly made shares  $s_i$  and  $r_i$  are consistent with respect to the polynomial coefficients which have been committed by the dealer. So, now, if the complaint is publicly trying to get resolved, before accepting the shares  $s_i$  and  $r_i$  we have to publicly check whether the test which  $P_i$  had checked privately holds with respect to the publicly made  $s_i$  and  $r_i$ .

If the check fails; that means, again dealer is corrupt and it is safe to discard the dealer. Because if the dealer is honest then the complain would have been made by a corrupt  $P_i$  only and an honest dealer will make public the right  $s_i$  and the right  $r_i$  values. So, this test

will always pass for an honest dealer when it is done publicly and as a result an honest dealer will never get discarded. So, now the sharing phase protocol is over.

If the dealer is not discarded and if the dealer is corrupt then assuming that the binding property of the commitment scheme holds it is guaranteed that it has distributed consistent shares to the honest parties and the protocol is over. So, now, you can see the protocol is so simple we do not have any pairwise consistency checks here and several stages of complaint and resolution and so on.

And we do not form any sophisticated data structures in the consistency graphs and so on. It is also easy to see that if the dealer is honest then throughout the protocol adversary does not learn anything about the dealer's secret. So, there could be up to  $t$  corrupt shareholders. So, throughout the protocol they only see the  $t$  shares lying on the Shamir sharing polynomial  $F$  and on the randomness polynomial  $R$ . Because whatever shares are made public for the case of honest dealer they correspond to only corrupt shareholders which adversary anyhow will be knowing beforehand. And as argued earlier these commitments does not reveal anything additional about the Shamir sharing polynomials and the randomness polynomial ok.

(Refer Slide Time: 13:45)

**$(n, t)$  Shamir-Sharing with Committed Shares**

□ An element  $s \in \mathbb{F}$  is said to be  $(n, t)$  Shamir-shared with committed shares, if:

- There exist some  $t$ -degree polynomials:
  - $F(Z) = s + a_1 Z + \dots + a_t Z^t$
  - $R(Z) = r + b_1 Z + \dots + b_t Z^t$
- Each (honest)  $P_i$  holds:
  - $s_i \cong F(\alpha_i) \quad r_i \cong R(\alpha_i)$
- Coefficients of the polynomials are **publicly committed**:
  - $C_{s,r} \cong g^s h^r \quad C_{a_1,b_1} \cong g^{a_1} h^{b_1} \quad C_{a_t,b_t} \cong g^{a_t} h^{b_t}$

$(s)_t$   $\cong$  vector of information held by the honest parties

So, that is the secret sharing phase for the Pedersen's VSS scheme. Now, based on that we now define a new form of secret sharing we call it as  $n, t$  Shamir secret sharing with committed shares. So, imagine you have a value  $s$  from the field. We say that it is Shamir

secret shared with committed shares, if the following holds: there should exist some  $t$  degree Shamir sharing polynomial and a  $t$  degree randomness polynomial with each honest party  $P_i$  holding the shares on the polynomials  $F$  and  $R$ , plus the coefficients of the Shamir sharing polynomials should be publicly committed with respect to the coefficients of the randomness polynomial.

So, if this is ensured then we say that a value  $s$  is Shamir secret shared with committed shares. And for pictorial illustration we will use this box representation to denote that there is a value  $s$  which has been secret shared with committed shares. Sometimes we will also use this bracket notation to denote the vector of information held by the honest parties.

So, now you can see we have various kinds of representations which we have used across various MPC protocols for perfectly secure VSS we had used  $n, t$  Shamir secret sharing then for statistical protocols, we used  $n, t$  2D secret sharing with IC signatures where each primary share is further secret shared. And now for cryptographically secure MPC, we are augmenting Shamir secret sharing with committed shares by making public the commitment of the coefficients of the sharing polynomial ok.

As usual for this new form of secret shared data the linearity property holds. Because anyhow the linearity property holds for Shamir secret sharing. Since we are using the Pedersen's commitment scheme which has the homomorphic property, we can perform linear operations on committed data. So, that is why we can exploit the linearity property.

So, what does it mean? So, imagine you have value  $u$  and value  $v$  which are already secret shared with committed shares. And suppose  $w$  is a value which is  $c_1$  times  $u$  plus  $c_2$  times  $v$ , where  $c_1$  and  $c_2$  are some publicly known constants from the field. And you would like now to compute a secret sharing for  $w$  with committed shares, that will not require to run a fresh instance of verifiable secret sharing to secret share  $w$ .

Based on whatever information the parties have for the secret sharing of  $u$  and the secret chaining of  $v$ , they can perform some local operations, local computations and end up getting their respective data for a Shamir secret sharing of  $w$  with committed shares.



(Refer Slide Time: 17:31)

### (n, t) Shamir-Sharing with Committed Shares: Reconstruction Protocol

Let  $s$  be  $(n, t)$  Shamir-shared with committed shares:

- $t$ -degree polynomials:  
 $F(Z) = s + a_1Z + \dots + a_tZ^t$      $R(Z) = r + b_1Z + \dots + b_tZ^t$
- Each (honest)  $P_i$  holds:  
 $s_i \equiv F(a_i)$      $r_i \equiv R(a_i)$
- Coefficients of the polynomials are **publicly committed**:  
 $c_{s,r} \equiv g^s h^r$      $c_{a_1,b_1} \equiv g^{a_1} h^{b_1}$      $c_{a_t,b_t} \equiv g^{a_t} h^{b_t}$

Goal: to **publicly reconstruct**  $s$

- Each  $P_i$  makes **public**  $(s_i, r_i)$  **If Binding holds  $\Rightarrow$  output is correct**
- $(s_i, r_i)$  is publicly **discarded**, if:  
 $g^{s_i} h^{r_i} \neq (c_{s,r}) \cdot (c_{a_1,b_1})^{a_i} \dots (c_{a_t,b_t})^{a_i^t}$
- Use any  $t+1$  **non-discarded**  $s_i$  to interpolate  $F(Z)$  and output  $s = F(0)$

Claim: if  $t < n/2$ , then with a high probability, the honest parties output  $s$

Now what will be the reconstruction protocol to reconstruct a Shamir shared data with committed shares? So, imagine there is a value  $s$  which is already Shamir shared with committed shares. You want to reconstruct the value  $s$  and say we are in the setting where  $t < n/2$ .

So, in the perfect world, for perfectly secure case we could have asked each party to make it share public and in the perfect world  $t$  would have been less than  $n/3$ . So, we can apply the Reed-Solomon error correction and recover back the Shamir sharing polynomial correctly. But we are not in the setting of  $t < n/3$ , we will be in the setting of  $t < n/2$ .

So, how do we reconstruct the secret  $s$ ? Well the idea will still remain the same our goal will be to now try to recover back the polynomial  $F$  by asking every party to make its share public. But to verify whether the party  $P_i$  has made the share correct on whether it has made public the correct share we can use the publicly available commitments ok.


So, this is where this data structure, this  $(n, t)$  secret sharing with committed shares differ from  $(n, t)$  2D secret sharing with IC signatures. So, if you recall the reconstruction protocol in the statistical domain, there to reconstruct the value, each primary shareholder would have made public the secondary shares along with the signatures and then we would have tried to interpolate the primary shares.

And once  $t + 1$  primary shares are available we use them to interpolate back the underlying Shamir sharing polynomial. But here the idea will be different. Each party makes public its share and then those shares are checked with respect to the publicly available commitment by performing this check. If this check fails then that means, that the share produced by  $P_i$  is corrupt. So, simply discard it.


And once we have  $t + 1$  non discarded shares, we use those shares to get back the Shamir sharing polynomial and output the constant term of that polynomial. Now since we are in the setting of  $t < n/2$ , there will be at least  $t + 1$  honest parties, who will be never discarded. So, this protocol will give us some output and assuming that the binding property holds, if binding holds, then the output is correct. Because if a corrupt  $P_i$  is able to break the binding property, then even though it makes public incorrect shares say it has made public  $s_i$  prime and  $r_i$  prime, if it is able to break the binding property then even though it has made public incorrect shares, it may get accepted in which case we will end up getting an incorrect polynomial as output. But the probability that the polynomial time adversary is able to break the binding property is negligible.

(Refer Slide Time: 21:07)

## Cryptographically-Secure Robust MPC



- Idea: shared circuit-evaluation of the circuit
  - ❖ Each value remains  $(n, t)$ -Shamir-shared with committed shares
- Protocol will follow Offline-Online paradigm
  - ❖ **Offline phase:** parties jointly generate  $(n, t)$ -Shamir-shared multiplication triples with committed shares
    - Similar to the statistically-secure protocol except that VSS is now cryptographically-secure
  - ❖ Online phase: Joint evaluation of the circuit
    - **Input stage:** each  $P_i$  acts as a dealer and  $(n, t)$ -Shamir-shares its input with committed shares
    - **Computation stage:**
      - Linear gates: non-interactive
      - Multiplication gates: Beaver's trick
  - ❖ **Output stage:** public reconstruction of secret-shared function output



So, now based on this cryptographically secure VSS we can design a cryptographically secure robust MPC as follows. So, the idea will remain the same namely shared circuit evaluation which has been used for previous MPC protocols. The difference is now that

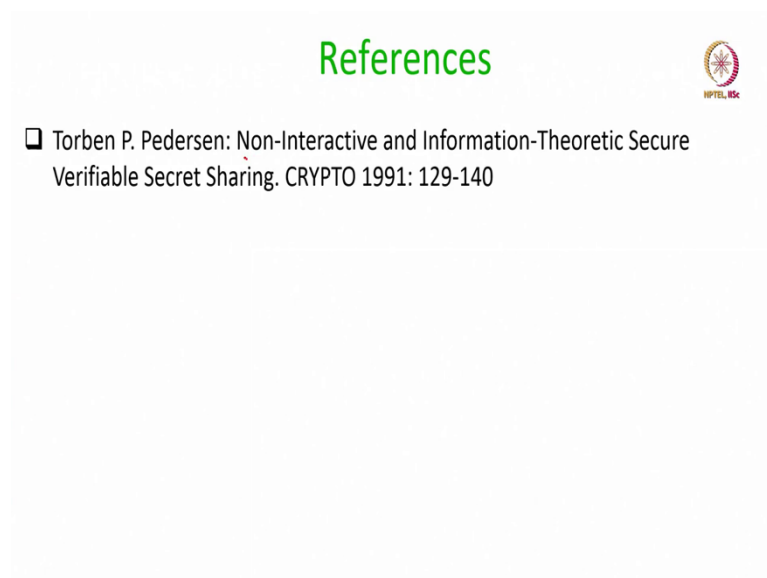
each value during the circuit evaluation will be Shamir secret shared with committed shares and for efficiency purpose we can use the offline online paradigm.

In the offline phase the parties can jointly generate secret shared multiplication triples with committed shares. And this protocol will be similar to the statistically secure protocol for the offline phase except that the VSS is now cryptographically secure, the rest of the components like triple transformation, polynomial verification, triple sharing all of them remains the same.

And in the online phase the parties will be jointly evaluating the circuit in the secret shared fashion starting with the input stage where each party will first act as a dealer and secret share its input for the function to be computed using an instance of the Pedersen's VSS. And then each gate will be evaluated in a secret shared fashion: the linear gates will not require any interaction thanks to the linearity property of the underlying secret sharing scheme.

For the multiplication gates we can use the Beavers method by deploying the triples which the parties would have generated in the offline phase. And now once the circuit output is available in a secret shared fashion we can publicly reconstruct it ok.

(Refer Slide Time: 23:07)



So, I am not going through the details of the full protocol you can find those details in this paper.

Thank you.