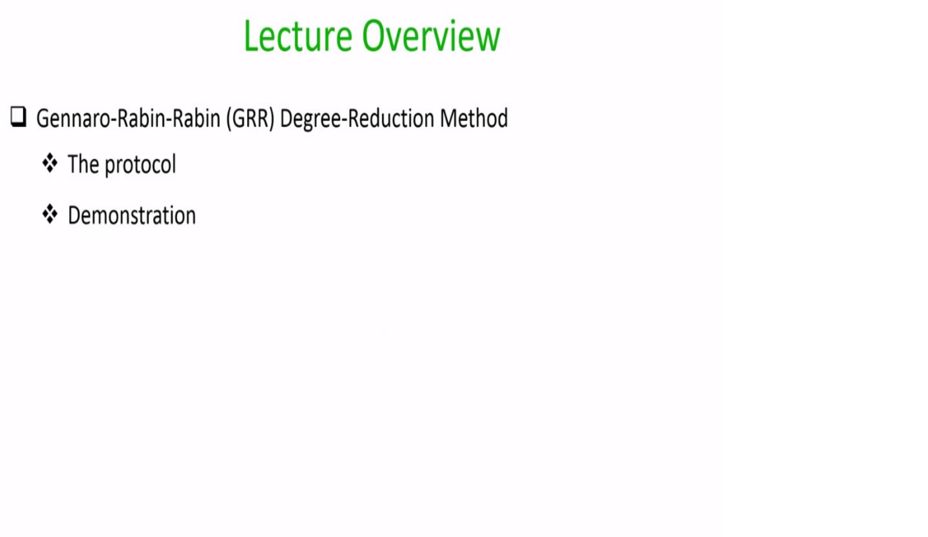
Secure Computation: Part 1

Prof. Ashish Choudhury
Department of Computer Science
Indian Institute of Science – Bengaluru

Lecture – 24
The Gennaro-Rabin-Rabin (GRR) Degree-Reduction Method

Hello everyone. Welcome to this lecture. So, in this lecture we will see the more efficient solution for the degree reduction problem due to Gennaro Rabin-Rabin this is called as the GRR degree reduction method.

(Refer Slide Time: 00:45)



Lecture Overview

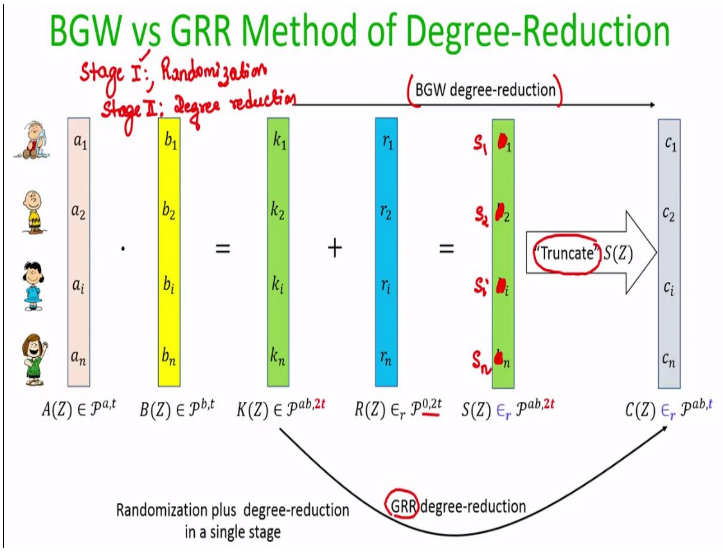
❑ Gennaro-Rabin-Rabin (GRR) Degree-Reduction Method

❖ The protocol

❖ Demonstration

And we will also see an example to get a better understanding of the degree reduction method.

(Refer Slide Time: 00:52)



BGW vs GRR Method of Degree-Reduction

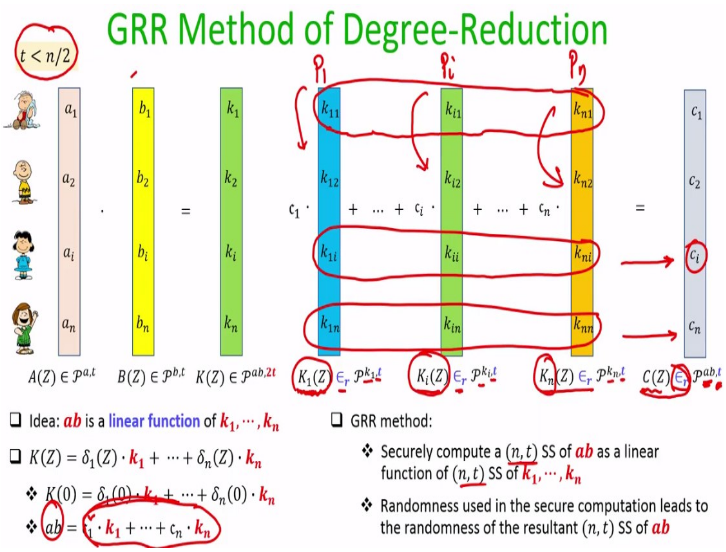
Stage I: Randomization
Stage II: degree reduction
BGW degree-reduction
a_1 a_2 a_i a_n
b_1 b_2 b_i b_n
k_1 k_2 k_i k_n
r_1 r_2 r_i r_n
S_1 S_2 S_i S_n
c_1 c_2 c_i c_n
"Truncate" $S(Z)$
$A(Z) \in \mathcal{P}^{a,t}$
$B(Z) \in \mathcal{P}^{b,t}$
$K(Z) \in \mathcal{P}^{ab,2t}$
$R(Z) \in_r \mathcal{P}^{0,2t}$
$S(Z) \in_r \mathcal{P}^{ab,2t}$
$C(Z) \in_r \mathcal{P}^{ab,t}$
Randomization plus degree-reduction in a single stage
GRR degree-reduction

So, just to; quickly recap this was the BGW reduction method that we had seen in the previous lecture which is done in two stages. So, let me write down the two stages here. So, there is a typo here I should have used the different variable S 1, S 2, S i and S n here. So, in stage one we do the randomization and in stage 2 we do the degree reduction. The randomization is done by adding shares of 0 random shares of 0 lying on a 2t degree polynomial.

And then the degree reduction of that 2t degree random sharing happens by a truncate operation. Now what the GRR degree reduction method does is the following. It cleverly combines both these two stages in a single stage namely in the same stage it will be doing the randomization plus the degree reduction and that is why it is more efficient because if you follow this BGW reduction.

Then we have to do the interaction both in stage 1 as well as in stage 2, but if we can combine both those two stages in a single stage then the parties need to interact only once. They do not need to interact twice and that will be a saving both in the number of rounds as well as the amount of communication which is needed.

(Refer Slide Time: 03:01)



GRR Method of Degree-Reduction

$t < n/2$
P_1
P_i
P_n
a_1 a_2 a_i a_n
b_1 b_2 b_i b_n
k_1 k_2 k_i k_n
k_{11} k_{12} k_{1i} k_{1n}
k_{i1} k_{i2} k_{ii} k_{in}
k_{n1} k_{n2} k_{ni} k_{nn}
c_1 c_2 c_i c_n
$\cdot$ $=$ $c_1 \cdot$ $+ \cdots + c_i \cdot$ $+ \cdots + c_n \cdot$ $=$
$A(Z) \in \mathcal{P}^{a,t}$
$B(Z) \in \mathcal{P}^{b,t}$
$K(Z) \in \mathcal{P}^{ab,2t}$
$K_1(Z) \in_r \mathcal{P}^{k_1,t}$
$K_i(Z) \in_r \mathcal{P}^{k_i,t}$
$K_n(Z) \in_r \mathcal{P}^{k_n,t}$
$C(Z) \in_r \mathcal{P}^{ab,t}$

- ❑ Idea: ab is a linear function of $k_1, \cdots, k_n$
- ❑ $K(Z) = \delta_1(Z) \cdot k_1 + \cdots + \delta_n(Z) \cdot k_n$
  - ❖ $K(0) = \delta_1(0) \cdot k_1 + \cdots + \delta_n(0) \cdot k_n$
  - ❖ $ab = c_1 \cdot k_1 + \cdots + c_n \cdot k_n$
- ❑ GRR method:
  - ❖ Securely compute a (n, t) SS of ab as a linear function of (n, t) SS of $k_1, \cdots, k_n$
  - ❖ Randomness used in the secure computation leads to the randomness of the resultant (n, t) SS of ab

So, let me explain the GRR degree reduction method again this will require t to be less than n over 2 and as I said earlier this is a necessary condition not only with respect to the GRR degree reduction method later on we will prove formally that any generic MPC protocol through that if you want to evaluate securely a multiplication gate we need this necessary condition of t less than n over 2.

So, to begin with the parties will first compute a non random vector of shares of ab lying on a 2t degree polynomial how they can do that just locally multiplying their respective shares of a and b no communication has happened at this point. Now the idea is that since t is less than n over 2 that means I can say that n is greater than equal to 2 t + 1 and the degree of K polynomial is 2 t.

And we have more than 2 t + 1 number of shareholders or 2 t + 1 number of shares of this K polynomial available. By the way I will interchangeably use the term shares of the polynomial or the shares of the value both mean the same thing shares of the value means distinct points on the sharing polynomial. When I say shares of the polynomial by that I also again mean the distinct evaluations of that sharing polynomial.

So, the first thing to observe here is that the constant term of this K polynomial which is the unknown value ab is a linear function of this k values k 1, k 2, k i, k n. This is because if you recall the Lagrange interpolation we know that alpha 1 k 1, alpha 2 k 2, alpha i k i and alpha n k n they together interpolate uniquely this K Z polynomial why uniquely because n is greater than equal to 2t + 1 and the degree of K polynomial is 2t.

And we know the form of K polynomial can be expressed in terms of the distinct points on the K polynomials in terms of the delta polynomials and if this is the case then by substituting Z = 0 I obtain the following relationship I can say that K of 0 is the delta 1 0 times k 1, delta 2 0 times k 2, delta n 0 times k n. So, now let me call this delta 1 0 to be the constant c 1, delta 2 0 to be the constant c 2 and delta n 0 to be the constant c n.

They are basically your Lagrange interpolation coefficients which we had discussed when we discussed Lagrange polynomial interpolation over the finite fields and K of 0 is this ab. So, the summary here is that unknown ab why unknown ab because it is secret shared right now through the K polynomial no one knows the value of a times b because a was secret shared, b was secret shared and parties have just locally multiplied their respective shares of the a and b.

So, what we have summarized here is that based on Lagrange interpolation we know that this unknown ab a times b is a publically known linear combination of unknown values k 1, k 2, k n why publically known? Because the Lagrange's coefficient fancy c 1, fancy c 2 and this fancy c n they are publically known. They have got absolutely nothing to do with your k 1, k 2, k n because this c 1 is delta 1 0 and delta 1 polynomial is basically publically known delta 2 polynomial is publically known.

Hence delta 2 0 is publically known, delta n polynomial is publically known hence delta n of 0 is publically known what is not known are the actual k 1, k 2, k n not known in the sense it is not publically known k 1 is available only with P 1, k 2 is available only with P 2, k i is available only with P i and k n is available only with P n and we know that how to securely compute linear functions of private inputs.

What does it mean here and why is it applicable here? So, we are considering right now a scenario where P 1 has the value k 1, P 2 has the value k 2, P i has the value k i and P n has the value K n. In that sense they are private inputs of the respective parties and we know that there is this unknown ab which can be expressed as this public linear function of this private inputs.

So, if we ask the parties P 1, P 2, P i, P n to respectively secret shares k 1, k 2, k i, k n then we can apply the same linear function the Lagrange's interpolation linear function on the shares of k 1, k 2, k i and k n and end up getting shares of ab and our goal is finally to obtain shares of ab. We do not want to make a pubic we do not want to make b public in the degree reduction problem.

And we do not want to make ab also public somehow we want to obtain shares of ab lying on a random t degree polynomial and computing shares of ab is equivalent to computing this linear function in a secret shared fashion and remember we had already seen that how to securely compute linear functions in a secret shared fashion that was our starting point to discuss the BGW protocol.

The simpler case for the BGW protocol was when the inputs of the parties are available with the respective parties and we want to compute securely just a linear function of that it is very easy to compute that and that is what is the scenario here based on these observation and that is precisely what the GRR method does. We want to compute securely a secret sharing of ab as a publically known linear function of secret function of secret sharing of k 1 to k n.

Right now the secret sharing of k 1 to k n is not available because k 1 is right now not available in a secret shared fashion it is available only with P 1 k 2 is not available in a secret shared fashion it is only available with P 2 and so on, but if we want to compute securely this function that is always possible by asking the respective parties to secret share their private inputs.

And the private inputs of the respective parties are now k 1, k 2, k i, k n we do not want to make the values k 1, k 2, k i, k n public, but still we want to compute the output a times b which is the linear combination of this k 1, k 2, k i, k n in a secret shared fashion and this can be done very easily and why it will be ensured that the final shares of ab which are computed as linear function of secret sharing of k 1, k 2, k i, k n will be a random vector of shares because that will be triggered by the randomness which is used to securely compute this linear function.

So, now let us see what exactly happens in the GRR degree reduction method based on whatever we have discussed till now. So, we want to securely compute a linear function of publically known linear function of k 1, k 2, k i and k n and that is a linear since it is a linear function we know how to securely compute it basically each party has to secret share their private inputs which are k 1, k 2, k i, k n respectively.

And since the linear function is publically known may be linear combiners are publically known the parties can apply that linear combination on the shares of k 1, k 2, k i and k n and then obtain the respective shares of ab. So, we ask P 1 to do this namely it picks a random polynomial k 1 Z of degree t and secret shares k 1 namely it runs an instance of Shamir secret sharing to secret share the input k 1.

Similarly, P 2 will be asked to execute an instance of Shamir secret sharing and secret share k 2 P i will be asked to act as a dealer and run an instance of Shamir secret sharing to share k i so for that it will be picking a random polynomial of degree t whose constant term is k i and like that we will ask the nth party to pick a random polynomial of degree t with constant term being k n and distributor shares.

And after that we compute this linear function why we can compute this linear function now why it is possible because the input for this linear function namely k 1, k 2, k n is now available in a secret shared fashion and since c 1 is publically known each party say the first party it computes c 1 times the first share of k 1 + c 2 times the first share of k 2 + c 1 times the first share of k i + c 1 times the first share of k n.

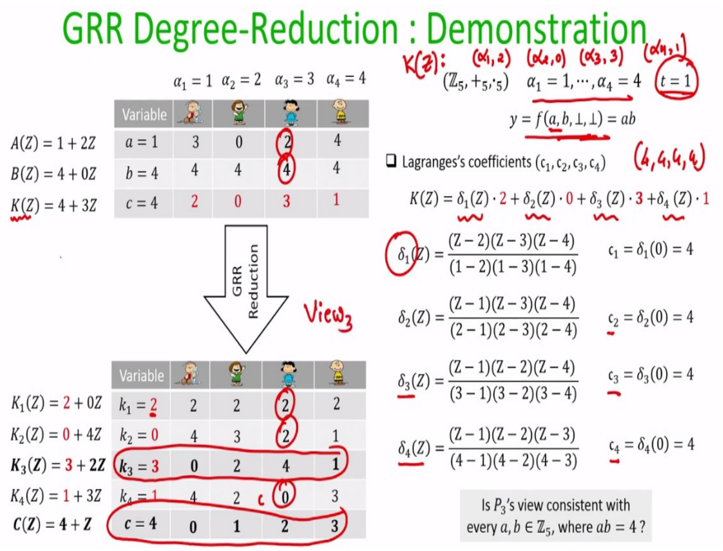
Similarly, P i it will take all the Ith shares for k 1, k 2, k i, k n multiply them respectively with this public constant c 1, c 2, c i, c n and it will obtain its new share called c i and it is similarly the nth party it will take all the nth shares that it has got in the respective secret sharing instances multiply them with c 1, c 2, c i, c n respectively and obtain this share and the claim is now that this vector of c values which now the parties have collectively with ith party having the ith component lie on a polynomial C Z whose constant term is ab.

And its degree is t and it is a random polynomial that means all its coefficients except the constant coefficient are random elements. Why so because each of this polynomials k 1, k 2, k i, k n which are picked in the individual instances of Shamir secret sharing they are random polynomials of course except the constant coefficient. So, k 1 has all its coefficients randomly chosen.

Except the constant coefficient k i has all its coefficient randomly chosen k n has all its coefficients randomly chosen and if you take a linear combination of this random polynomials with random coefficient you obtain a random polynomial whose coefficients will be random of course except the constant coefficient and why the constant coefficient ab because the constant coefficient will be c 1 times the constant coefficient of K 1 polynomial + c 2 times the constant coefficient of the K 2 polynomial + c i times the constant coefficient of K i polynomial + c n times the constant coefficient of the K n polynomial which we know is ab.

So, now you can see we do not have to do a two stage approach. In the single stage just because we want to now compute this linear function and to compute that linear function anyhow parties have to run instances of random secret sharing we are now able to do the ridge reduction as well as bring in the randomness or the randomization just in a single stage.

(Refer Slide Time: 17:56)



GRR Degree-Reduction : Demonstration

	$\alpha_1 = 1$	$\alpha_2 = 2$	$\alpha_3 = 3$	$\alpha_4 = 4$
Variable				
$A(Z) = 1 + 2Z$	$a = 1$	3	0	2	4
$B(Z) = 4 + 0Z$	$b = 4$	4	4	4	4
$K(Z) = 4 + 3Z$	$c = 4$	2	0	3	1

GRR Reduction

View$_3$

	Variable				
$K_1(Z) = 2 + 0Z$	$k_1 = 2$	2	2	2	2
$K_2(Z) = 0 + 4Z$	$k_2 = 0$	4	3	2	1
$K_3(Z) = 3 + 2Z$	$k_3 = 3$	0	2	4	1
$K_4(Z) = 1 + 3Z$	$k_4 = 1$	4	2	c 0	3
$C(Z) = 4 + Z$	$c = 4$	0	1	2	3

$K(Z)$: $(\alpha_1, 2)$ $(\alpha_2, 0)$ $(\alpha_3, 3)$ $(\alpha_4, 1)$

$(\mathbb{Z}_5, +_5, \cdot_5)$ $\alpha_1 = 1, \cdots, \alpha_4 = 4$ $t = 1$

$y = f(a, b, \perp, \perp) = ab$

❑ Lagranges's coefficients (c_1, c_2, c_3, c_4) $(4,4,4,4)$

$$K(Z) = \delta_1(Z) \cdot 2 + \delta_2(Z) \cdot 0 + \delta_3(Z) \cdot 3 + \delta_4(Z) \cdot 1$$

$$\delta_1(Z) = \frac{(Z-2)(Z-3)(Z-4)}{(1-2)(1-3)(1-4)} \qquad c_1 = \delta_1(0) = 4$$

$$\delta_2(Z) = \frac{(Z-1)(Z-3)(Z-4)}{(2-1)(2-3)(2-4)} \qquad c_2 = \delta_2(0) = 4$$

$$\delta_3(Z) = \frac{(Z-1)(Z-2)(Z-4)}{(3-1)(3-2)(3-4)} \qquad c_3 = \delta_3(0) = 4$$

$$\delta_4(Z) = \frac{(Z-1)(Z-2)(Z-3)}{(4-1)(4-2)(4-3)} \qquad c_4 = \delta_4(0) = 4$$

Is P_3's view consistent with every $a, b \in \mathbb{Z}_5$, where $ab = 4$?

So, let us see a demonstration here and imagine that our circuit is a simple circuit consisting of just one multiplication gate a times b that is what we want to compute our t = 1 so imagine a was secret shared through this A polynomial, b was secret shared through this B polynomial the parties first locally multiply their respective shares of a and b, but right now they are not exchanging them publically and reconstructing the function output because we have to first do the degree reduction.

And now since after the multiplication there is no more gate we have to publically reconstruct the function output that means now our circuit is of this form a is coming from one party, b is coming from one party there is only one multiplication gate and then there is a function output y. So, the parties first secret share the inputs then they do the degree reduction here and then they reconstruct because there are no more gates after this multiplication.

If there would have been follow of gates after this multiplication then they would have proceeded with whatever they have obtained after the degree reduction and remember we had seen in the earlier lecture that if we do not do this degree reduction and directly reconstruct y then that leads to the breach of privacy namely the corrupt party could narrow down that what could be the possible input combinations and what could not be.

And that is a privacy breach, but now we can show that is not going to be the case if we do the degree reduction. So, right now this shares 0, 3, 1 they are not publically reconstructed they are your intermediate k 1, k 2, k 3, k 4 lying on a 2 degree polynomial why 2 degree because t = 1 so 2 times t will be 2. So, the first the parties will do the degree reduction and as part of the degree reduction we need to know the Lagrange's coefficients here.

So, I have worked out the Lagrange coefficient for you here so this is the K polynomial as per our notation and this K polynomial is actually passing through your alpha 1, alpha 1, 2, alpha 2, 0, alpha 3, 3 and alpha 4, 1 so K polynomial passes through these four points. So that means we now need the delta 1, delta 2, delta 3, delta 4 polynomials here. The delta 1 polynomial will be a 3 degree polynomial wherein the numerator you have all the alphas except alpha 1 as the route.

So, alpha 1, alpha 2, alpha 3, alpha 4 in this setting we have set to 1, 2, 3, 4. So, you can see in alpha 1 polynomial in the numerator all the alphas except alpha 1 are the routes and then to ensure that this alpha 1 polynomial evaluated at alpha 1 should survive or give the value 1 we have set the denominator like this. So, you can now see that the delta polynomial at alpha 2 and the delta polynomial 1 at alpha 3 and this delta polynomial 1 at alpha 4 will vanish, but it will survive at alpha 1.

As a result the first Lagrange coefficient will be the constant term of this delta polynomial will be 4. So, remember all the operations are performed over Z 5 where the plus operations are addition modulo 5 and multiplication operations are multiplication module 5. The delta 2 polynomial in the numerator will have all the routes except alpha 2 and to ensure that it survives at alpha 2 we have the corresponding denominator by substituting Z = 0 we get the second Lagrange coefficient.

Similarly, we obtain the third delta polynomial and correspondingly the third Lagrange coefficient and the fourth delta polynomial and the corresponding fourth Lagrange coefficient so this c 1, c 2, c 3, c 4 they are publically known, they have got nothing to do with the shares of ab and the shares of 4 which right now are available with the parties. So, now we know that unknowns constant c not constant the unknown value c is a linear combination namely it is c 1 times the share 2 which is available with the party 1.

Second Lagrange coefficient time the value k 2 which is 0 third Lagrange coefficient time k 3 which is 3 + c 4 times k 4 which is 1, but the value 2, 0 and 3 and 1 they are right now the private inputs of the first party, second party, third party and fourth party respectively. So, we have to compute this linear function in a secret shared fashion so that is what will be done now.

So party 1 will act as a dealer and he will secret share its private input k 1 through a one degree polynomial, party 2 will act as a dealer and secret share its private input which is k 2 through a one degree polynomial, party 3 will act as a dealer and secret share its private input namely k 3 through a one degree polynomial and similarly party 4 will act as a dealer and secret shares its private input k 4 through a one degree polynomial.

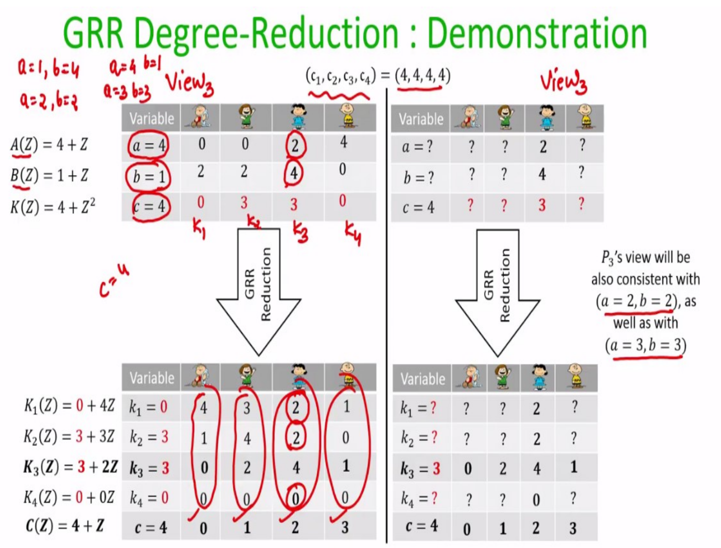
And now we have to compute these linear functions so k 1, k 2, k 3, k 4 they are now already available in secret shared fashion, apply the same liner combination so on the respective shares of k 1, k 2, k 3, k 4. So, the parties will individually apply the Lagrange coefficients, apply the linear combination and then they will obtain the respective shares of c. So, this will be now your vector of shares of c which will be considered as the output of the degree reduction.

So, now in your circuit there are no more gates to evaluate it further. We are considering a function which has just one multiplication gate. So, since there are no more gates to evaluate further the parties will stop at this point and it is now time to reconstruct the function output. So, they will make the final shares of c public and the parties will now interpolate alpha 1 0, alpha 2 1, alpha 3 2 and alpha 4 3 and now they will obtain this c polynomial.

So, we will again take the case where P 3 is corrupt similar to what we had done when we saw this same example to understand the challenges associated with the process of evaluating the multiplication gates. So, I have now highlighted in bold the view of corrupt P 3. So, corrupt P 3 will have the shares of a, shares of b and whatever shares it obtain during the degree reduction problem.

And it itself will be acting as a dealer and secret sharing k 3 and then finally it will be seeing all the shares of c being made public. So, everything which has been highlighted in bold constitutes view 3.

(Refer Slide Time: 26:41)



GRR Degree-Reduction : Demonstration

$a=1, b=4$ $a=4$ $b=1$

$a=2, b=2$ $a=3$ $b=3$

$View_3$

$(c_1, c_2, c_3, c_4) = (4, 4, 4, 4)$

	Variable				
$A(Z) = 4 + Z$	$a = 4$	0	0	2	4
$B(Z) = 1 + Z$	$b = 1$	2	2	4	0
$K(Z) = 4 + Z^2$	$c = 4$	0	3	3	0
		k_1	k_2	k_3	k_4

$c=4$

GRR Reduction

	Variable				
$K_1(Z) = 0 + 4Z$	$k_1 = 0$	4	3	2	1
$K_2(Z) = 3 + 3Z$	$k_2 = 3$	1	4	2	0
$K_3(Z) = 3 + 2Z$	$k_3 = 3$	0	2	4	1
$K_4(Z) = 0 + 0Z$	$k_4 = 0$	0	0	0	0
$C(Z) = 4 + Z$	$c = 4$	0	1	2	3

$View_3$

Variable				
$a = ?$	?	?	2	?
$b = ?$	?	?	4	?
$c = 4$	?	?	3	?

GRR Reduction

P_3's view will be also consistent with $(a = 2, b = 2)$, as well as with $(a = 3, b = 3)$

Variable				
$k_1 = ?$	?	?	2	?
$k_2 = ?$	?	?	2	?
$k_3 = 3$	0	2	4	1
$k_4 = ?$	?	?	0	?
$c = 4$	0	1	2	3

And it also knows the Lagrange coefficient they are publically known so this is the view 3 the question mark denotes they are the unknown values. So, now a malicious P 3 with unbounded computing power would like to infer that okay the final result is 4. Is it because of the case that a = 1, b = 4 or is it because a = 2, b == 4 or is it because a = 4, b = 1 or is it because a = 3 and b = 3.

Again let us recall that if we do not do the degree reduction and directly evaluate it as per the BGW invariant then a corrupt P 3 based on view 3 could narrow down the two combinations are possible two input combinations are not possible, but now we have to see that after doing the degree reduction can P 3 throw away certain ab combinations from a view or not. So, let us first try the combination a = 4 and b = 1.

And again based on the previous exercises if P 3 fixes a to be 4 then given that she has seen the share 2 for that unknown a that freezes the A polynomial in her view and if she freezes b = 1 then given that she has seen the share 4 for that unknown b that freezes the B polynomial to 1 + Z in her view and that freezes also the shares that other parties would have obtained for that unknown a and b.

Then once the shares of a and b are fixed with respect to whatever hypothesis P 3 is making in her mind that also fixes k 1, k 2, k 3, k 4. Now if k 1 is fixed to 0 given that P 3 has seen the share 2 for that unknown k 1 together that freezes the K 1 polynomial that P 1 would have used and if k 2 is fixed to 3 then given that P 3 has seen the share 2 for that unknown k 2 together with k 2 equal to 3 and her share being 2 that fixes the polynomial that P 2 would have used to share that k 2 and that would have fixed the other shares of k 2 as well.

K 3 anyhow was shared by P 3 so she is not going to make any hypothesis and k 4 if it is fixed to 0 that along with the fact that unknown k 4 P 3 has received the share 0 that fixes the K 4 polynomial from the view point of the corrupt P 3 and the other shares of k 4 that the other parties would have received and now the corrupt P 3 will check whether the linear combination the Lagrange linear combination of this shares of k 1, k 2, k 3, k 4 gives you this values and answer is yes that is quite possible.

Whereas without degree reduction based on the view 3, corrupt P 3 was able to throw away certain values of ab. I am leaving it as an exercise, but we can show that the same view 3 which adversary P 3 has seen by participating in the protocol with be consistent with the hypothesis a = 2, b = 2 as well as with the hypothesis a = 3, b = 3 that means if we do the degree reduction and then finally reconstruct the circuit output.

Then there would not be any breach of privacy adversary will only learn the final outcome c equal to 4, but it cannot narrow down whether it was actually because of a = 1, b = 4 a being 2, b being 2 a being 4, b being 1 or a being 3 or b being 3 all four possible input scenarios are equiprobable from the view point of corrupt P 3 and hence the privacy properties satisfied. So, with that I end this lecture. Thank you.