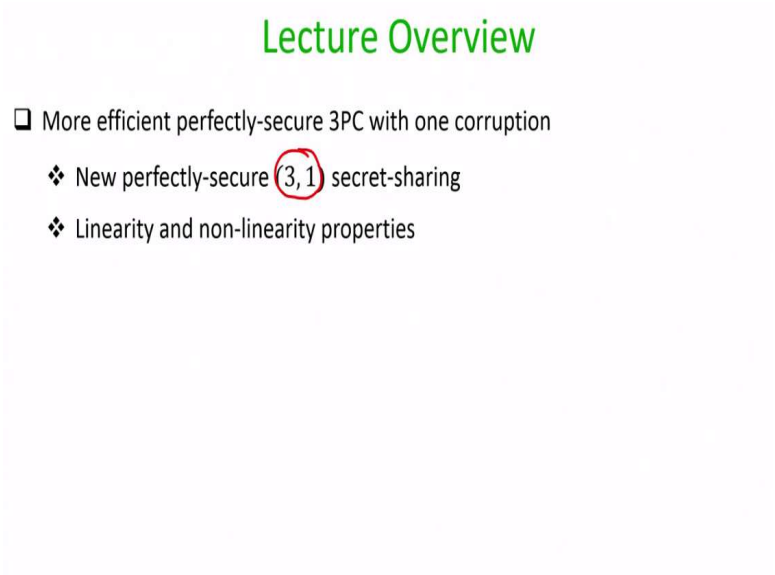
Secure Computation: Part 1
Prof. Ashish Choudhury
Department of Computer Science
Indian Institute of Science – Bengaluru

Lecture – 33
More Efficient Perfectly-Secure 3PC

Hello everyone. Welcome to this lecture.

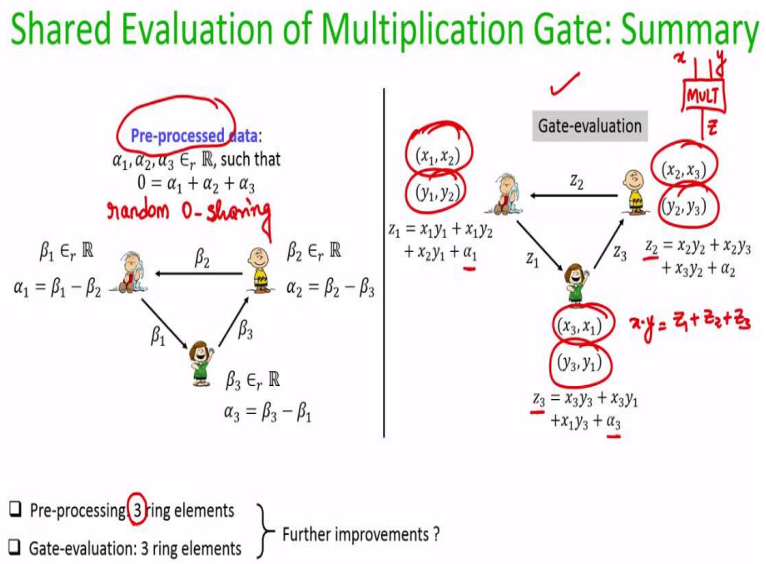
(Refer Slide Time: 00:31)



So, the plan for this lecture is as follows. In this lecture, we will continue our discussion on perfectly secure 3 party computation with one semi honest corruption and our goal will be to get more efficient protocol compared to the one which we have designed earlier based on replicated secret sharing. With that goal in mind, we will design first a new perfectly secure secret sharing scheme among for 3 parties tolerating one corruptions.

I am tolerating one corruption and then we will discuss the linearity and non-linearity properties of the secret sharing scheme.

(Refer Slide Time: 01:14)



So, here is the motivation why we need a more efficient perfectly secure 3 party protocol. So, this was the protocol based on the replicated secret sharing wherein the pre processing phase we generate a random zero sharing relatively shared form. Namely P_1 will hold α_1, P_2 will hold α_2, P_3 will hold α_3 such that together they sum up to 0 and assuming that such a setup has been done in the pre processing phase.

Secure data has been generated in the pre processing phase the multiplication gate whose inputs are x and y and which are available in replicated secret shared form can be evaluated in a secret shared form replicated secret shared form as follows. So, each of the parties have two of the shares for x each of the parties have two of the shares of y and now if we expand x dot y it is a summation of 9 summands.

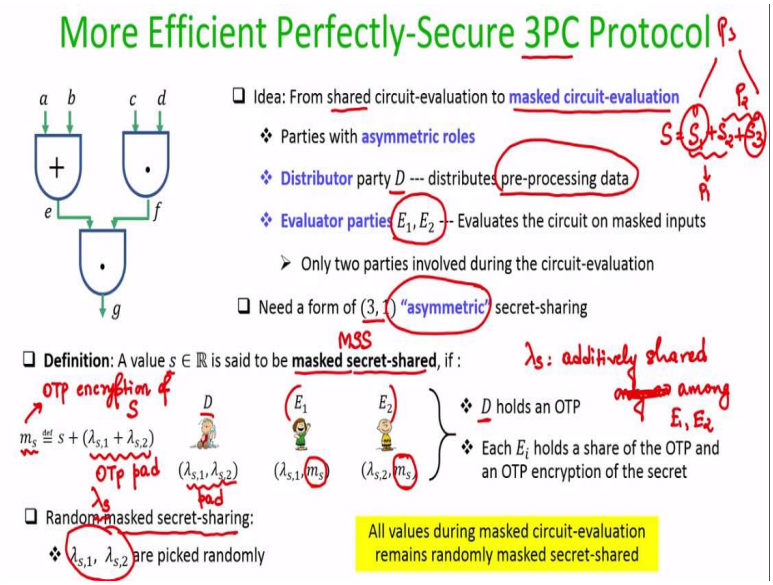
So, P_1 is assigned to compute sum of three of those summands and randomize it by adding the α_1 component of the zero sharing and send it to P_3. P_3 is supposed to sum up three of the summands and randomize it by adding α_3 and similarly P_2 computes the sum of three of the summands and randomize it by adding α_2 and sent this to P_1 and now it is easy to see that the summation of Z_1, Z_2, Z_3 is your x dot y.

And each of the parties hold a two out of these three pieces from Z_1, Z_2, Z_3 and hence the output x dot y is available in replicated secret shared form. So, the pre processing phase here to generate this random zero sharing it requires one round of communication and communication of 3 ring elements. Namely each party has to send one ring elements to one of its neighbors.

And assuming that such a set up has been done the actual gate evaluation for evaluating the multiplication gate also requires one round and communication of 3 ring elements. So, the question that we are interested to answer here is that can we make further improvements in this process in terms of communication complexity because right now we are talking about concrete efficiency of 3 party secure computation.

So, any concrete improvement will have significant impact when you go and implement the protocol.

(Refer Slide Time: 04:24)



So, the idea for designing more efficient perfectly secure 3PC protocol is that we will go from the paradigm of shared circuit evaluation to masked circuit evaluation that means it will still be the case that parties will be evaluating the circuits in collaboration, but the semantic of the values over which the computation is performed will be different now. Namely we will envision a scenario where parties will have asymmetric roles.

Namely one of the parties will perform the task of distributor. So, you have 3 parties remember and it is up to us means as part of the protocol description we can assign the role of distributor to any one of those 3 parties and it is only the distributor who will do the entire pre processing. What kind of pre processing that we will seeing later, but it is only the single distributor party who will be doing the entire pre processing.

And this distributor will not be later involved when the actual circuit evaluation happens. So, at the time of circuit evaluation the remaining 2 parties are assigned the role of evaluator which

we denote as E_1, E_2 and they evaluate the circuit on masked inputs, masked values. So, with this idea in mind what we are trying to do is basically we are trying to kind of create asymmetric roles for the parties.

And this is unlike your previous MPC protocols where all the parties perform symmetric role when it comes to shared circuit evaluation, but now we are actually assigning different roles, different tasks for different parties. One of the parties will be playing the role of distributor and two of the parties will perform the role of evaluation or evaluators and to implement this idea we require a new form of 3, 1 asymmetric secret sharing.

Asymmetric in the sense that the sharing semantic will be kind of asymmetric, but different parties will hold different types of shares which will have different meaning unlike your previous secret sharing schemes where the secret sharing was symmetric in nature in the sense all the parties get same type of shares. So, for instance, if we consider Shamir secret sharing the share of each party was the evaluation of dealer sharing polynomial on a publically known evaluation point and so on.

So, in that sense it was symmetric, but now we will see a form of secret sharing for 3 parties and threshold t being 1 and this secret sharing will be asymmetric in nature. So, the asymmetric secret sharing that we are going to discuss I call it as masked secret sharing MSS you can give it any name and imagine you have a value s from the ring. So, again like the replicated secret sharing based MPC protocol.

The 3 party protocol the new 3PC protocol that we are going to design there also all the computations can be performed over a ring. You do not require that the function should be expressed as a circuit over a finite field. So, imagine there is a value s from the ring and we will say that the value s is secret shared as per masked secret sharing if the following sharing semantic hold.

The dealer should have two of the pieces λs_1 λs_2 E_1 should have a piece λs_1 and a value MS and E_2 should have the piece λs_2 and the piece MS. So, now you can see why it is asymmetric because different parties have different information here related to the secret shares of s, D has only the lambda values E_1 has one of these lambda values and MS.

And E_2 has another lambda value and the same MS which is held by E_1. Now the way to understand this secret sharing semantic is the following. What is MS here? MS you can imagine as OTP encryption one time pad encryption of the secret s and this entire thing the summation of the two λ components here can be considered as the OTP pad. So, I can call the summation of these two lambda pieces as lambda s.

So, you can imagine that there is this one OTP pad λs and MS is the encryption OTP encryption of the value s with respect to the pad being λs . The entire pad is available with the distributor not dealer D is for the distributor. So, the entire pad is held by the distributor here, but if the entire pad is held by the distributor I cannot afford to give the OTP encryption also to the distributor because if the distributor has both the pad as well as the OTP encryption then it knows the value s as well.

But we want to ensure that our secret sharing should have threshold t = 1 that means we want to ensure that among these 3 parties if any one of the parties try to learn about the secret it should fail. So, that is why we cannot afford to give the distributor the OTP encryption as well that is why the value MS is not available with the distributor it has the (()) (10:13) only to the pad.

And what exactly is the information available with the evaluators. So, the evaluators both the evaluators have the OTP encryption and only one of the secret shares of the pad. So, you can imagine that the pad here λs is additively shared the pad λs additively shared only among E_1 E_2 the two evaluators. So, evaluator 1 will have the OTP encryption plus one of the shares of the pad and that does not help the evaluator 1 to learn the secret because to learn the secret s it also needs the other part of the OTP pad which is not available with E_1.

And in the same way if E_2 only tries to learn the secret it will fail because even though it has the OTP encryption and one of the OTP pads it does not have the full OTP pad because it has only one of the shares for the OTP pad the other share of the OTP pad namely λs_1 is available with the other evaluator. So, that is why this secret sharing is kind of asymmetric when it comes to which entity posses what kind of information.

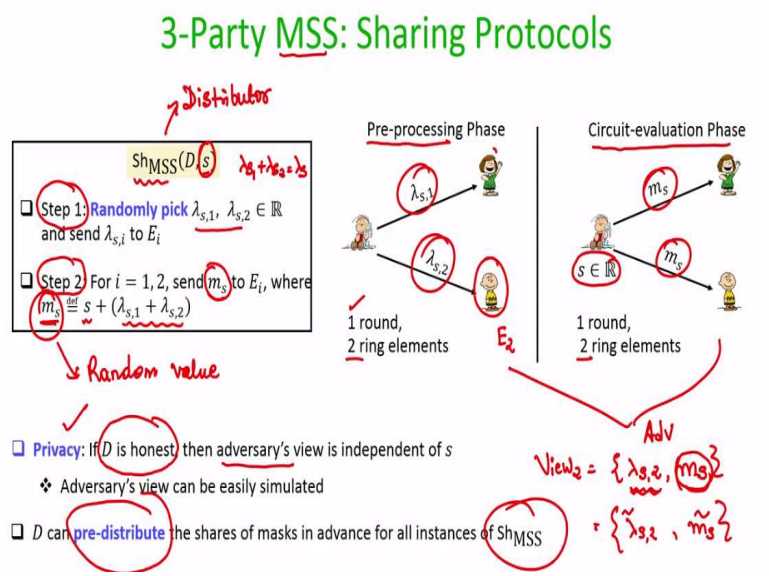
And here also the size of share of each party consist is 2 ring elements like your replicated secret sharing, but in the replicated secret sharing the sharing semantic was that the secret

should be splitted into three pieces such that their sum is s and each party should hold two of those pieces. So, there also each party was holding two of the pieces, but the type of information held by each party is kind of symmetric.

But here when it comes to the masked secret sharing the type of information held by the different parties is different. The distributor hold the entire OTP pad and both its secret shares the evaluators hold only half of the pad and the OTP encryption. Now, we will say that a masked secret sharing is random if the OTP pads are picked uniformly at random and the idea behind the new 3PC protocol is that the evaluators E_1, E_2 will be performing the circuit evaluation over the OTP encryptions of the values in the circuit.

And in the process since none of those two parties will be knowing the full OTP pads they would not be knowing that what exactly are the underlying values that are obtained during the computation and we would not be involving D during the circuit evaluation. The role of the D will be over once it distributes the OTP pads for all the values which are going to be obtained during the computation. So, that is the rough idea for this new efficient secure 3PC protocol.

(Refer Slide Time: 13:31)



So, since we have seen the sharing semantic for masked secret sharing let us see the sharing protocol how exactly a dealer can secret share a value as per the 3 party MSS secret sharing and how exactly we can reconstruct a value which is available in a secret sharing format as per the MSS secret sharing scheme. So, imagine distributor D wants to secret share a value. So, again the secret sharing protocol here will be different upon who is playing the role of the dealer.

Till now whatever secret sharing scheme we had seen it does not matter who is the dealer the role was symmetric. If we consider Shamir secret sharing and if there is a dealer with a value which he wants through secret share it has to pick a polynomial of degree t randomly and evaluate the polynomial at evaluation points and distribute the shares. It does not matter which of the n parties is going to play the role of the dealer.

The sharing protocol was symmetric, but now since our masked secret sharing has asymmetric role for distributor and evaluators depending upon who is going to play the role of the dealer whether it is the distributor or whether it is the evaluator the steps of the secret sharing protocol will be different. So, let us first see the secret sharing protocol assuming that the distributor itself is the dealer.

It has a value which it wants to secret share. So, now if the distributor wants to secret share a value s what does is the following. It first picks the uniformly random OTP pad and once it has picked the pad it can actually split it into two random pieces. So, I am denoting the random pieces of the pads associated with the value s λs_1 λs_2 such that the summation of λs_1 and λs_2 is actually the pad lambda s which is going to be held by the dealer.

And now the ith share of this λs is given to the ith evaluator. So, the first evaluator gets λs_1 the second evaluator gets λs_2 and step 2 of the protocol is the following. Once the value s is available with the distributor who is the dealer also in this case what it does is the following. It computes the OTP encryption of the secret s assuming that δs is the pad and it sends the OTP encryption to both the evaluators that is the second step.

So, now you can imagine here that this sharing protocol consists of two steps and the goal of the two steps are different. The goal of the first step is to associate the pad along with the value s which needs to be secret shared and the goal of the step 2 is to compute the OTP encryption of that value and send it to the evaluators. Now during the circuit evaluation of our new 3PC protocol the value s will be available only when the function starts getting computed because looking ahead the values which needs to be secret shared will be the inputs of the parties and so on.

So, this step 1 can be executed by this distributor during the pre processing phase itself. Namely what I am saying here is that distributor will be knowing that what are the values it would like to secret share when it comes to the actual circuit evaluation. For all those values for which distributor is the dealer and he has to secret share it as per the MSS sharing semantic it can pick the OTP pads and pre distribute split them and give it to the evaluators during the pre processing phase itself.

So, step 1 for all the sharing instances where the distributor is the dealer can be done in advance and in parallel during the pre processing phase that will require one round of communication and 2 ring elements will be communicated. The step 2 will be executed during the circuit evaluation when the value s is now available for the dealer to be secret shared.

Once it has the value s it will know that for this s it has associated the pads λs_1, λs_2. So, that mapping will be known to everyone that this pad is for this value, the next pad is for the next value and so on. What exactly are the contents of the pad that is known only to the distributor and half of those pads are with one of the evaluator and half of the pad is with the other evaluators.

So, now during the circuit evaluation phase what this distributor is going to do is it is also the dealer it will create the OTP encryption for this value and sent it to the respective evaluators and this can be done in parallel and this will require one round of communication and for each instance of secret sharing the step 2 will require a communication of 2 ring elements. So, now let us see whether this protocol satisfies the privacy property.

And for privacy property we have to argue that if the dealer who in this case is the distributor is honest and it is not under the control of adversary that means adversary can control only the evaluator 1 or the evaluator 2 then irrespective of which of the evaluators is under the adversary's control. The view of the adversary is completely independent of the underlying value s which is secret shared.

And this hold even if the adversary is computationally unbounded. So, let us assume that say evaluator number 2 is corrupt it is under the control of the adversary. So, what will be the view of adversary here with respect to this secret sharing? So, it will see the value λs_2. So, it will have a probability distribution over λs_2 and why probability distribution? Because each time

this distributor wants to secret share a value the pads λs_1 and λs_2 they are picked uniformly at random.

So, it is not that this distributor is going to pick the same λs_1 and same λs_2 for all the instances of secret sharing no they are picked randomly. So, that is why from the view point of a corrupt evaluator E_2 there is a probability distribution generated over the value λs_2 and what else is there in the view of corrupt E_2 the OTP encryption MS. Now my claim is that this view is independent of the actual value s which is secret shared.

Indeed this piece λs_2 is independently picked irrespective of what is the value of s because that is picked in step 1 and in step 1 whatever values are picked by the distributor it has got nothing to do with the actual s which has to be secret shared. So, definitely this λs_2 is independent of the actual secret hence it can be regenerated, reproduced or simulated by evaluator E_2 himself.

Namely, it can just write down a random value as the potential λs_2 which is going to receive from the dealer or the distributor and what is the second component? Well, second component is the OTP encryption and that OTP encryption is the summation of s along with a random λs_1 which is not known to the evaluator E_2. So, even though it knows half of the OTP pad namely λs_1 part it does not know what is the λs_2 part.

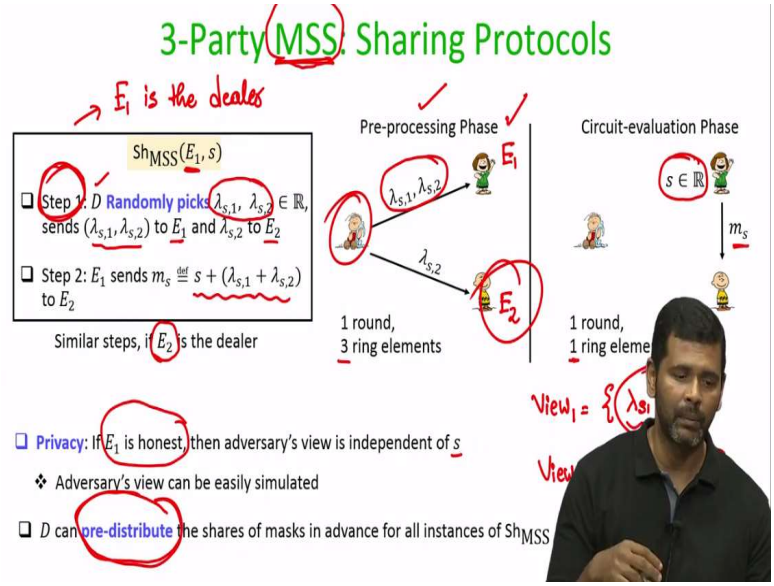
And hence this MS value is a random value from the ring because it is an OTP encryption. So, that is why from the view point of this potentially corrupt evaluator E_2 this value MS could be any value from the ring. It is a random element from the ring because this MS could be the sum of any candidate s from the ring and corresponding to that any corresponding λs_1 which along with the λs_2 which E_2 has produces the value MS.

So, that is why this value MS can also be recreated in the probability distribution can be recreated by the evaluator E_2 and that means whatever is the view evaluator E_2 gets interacting with the dealer during this secret sharing protocol that view can be reproduced by the evaluator E_2 if E_2 is corrupt without even talking with the dealer and hence whatever it learns by talking with the dealer is of no use and that is why it gives you perfect privacy.

And this holds even if E_2 is computationally unbounded and as I said the distributor D can pre distribute the shares of the pads or the mask in advance for all the instances of sharing protocol

for which the distributor itself is the dealer. So, this will be sharing protocol if the distributor is the dealer.

(Refer Slide Time: 23:18)



Now, let us take the case where E_1 is the dealer and it wants to secret share a value as per the sharing semantic of 3 party MSS. So, here also the step 1 will be identical to the previous case. Remember we want to ensure that the values are secret shared as per the 3 party MSS sharing scheme or sharing semantic and in that sharing semantic the pads are always picked by the distributor and completely held by the distributor.

So, it does not matter that evaluator E_1 is going to share the value the pads have to be picked by the distributor. So, that is why in this case also the distributor as part of step 1 randomly picks the shares of the pad λs and one of the pads it gives to the second evaluator and the full pad is given to the evaluator E_1 because evaluator E_1 is the owner of the value which needs to be secret shared because until and unless we do not provide the full pad to E_1 how can it compute the OTP encryption and make it available with the evaluator E_2.

And as part of the sharing semantic of 3 party MSS secret sharing we need to ensure that both the evaluators should have the OTP encryption, but OTP encryption can be computed by the dealer who in this case is evaluator E_1 only when it has both the components of the pads. So, that is why the full pad is provided to the dealer, but whatever thing the second evaluator is suppose to receive it gets only that much for the pad component.

And now in the step 2 when the actual value s which needs to be secret shared by the evaluator 1 who is the dealer is available and this will be happening during the circuit evaluation phase what this evaluator has to do? It has to just compute the OTP encryption namely it has to compute the summation of s and the two components of the pad and it has to just send it to the second evaluator.

And now it will be ensured that your value s is a secret shared as per MSS sharing semantic. So, what will the cost of pre processing phase? The step 1 can be executed in the pre processing phase it will have the same cost not same cost as earlier it will now have more cost in the pre processing phase for the case when distributor itself is the dealer requires a communication of 2 ring elements.

But for the case where one of the evaluators is the dealer it requires a communication of 3 ring elements, but you save during the circuit evaluation phase because now in the circuit evaluation phase the evaluator has to just send one ring element to the other evaluator and this is unlike your previous secret sharing protocol where distributor would have been the dealer.

There the circuit evaluation phase the distributor has to send the OTP encryption to both the evaluators. So, that is why you can see now the sharing protocol itself is kind of asymmetric depending upon who is playing the role of the dealer whether it is the distributor or whether it is the evaluator and similar steps could be executed if instead of evaluator E_1 it is the evaluator E_2 who is the dealer.

So, E_2 will have the full OTP pad and E_1 will have only the first component of the OTP pad. During the circuit evaluation phase when the value is ready for secret sharing E 2 has to compute the OTP encryption and just send it to party evaluator number E_1 and again I can now argue here that if evaluator E_1 who is the dealer in this case is honest and not under adversary's control.

That means adversary could control either the distributor or the second evaluator then the view of the adversary is independent of the underlying secret s even if the adversary is computationally unbounded and hence the adversary's view can be easily reproduced simulated, recreated and so on. So, for instance, if the distributor is corrupt but evaluator E_1 is honest what will be the view of the adversary?

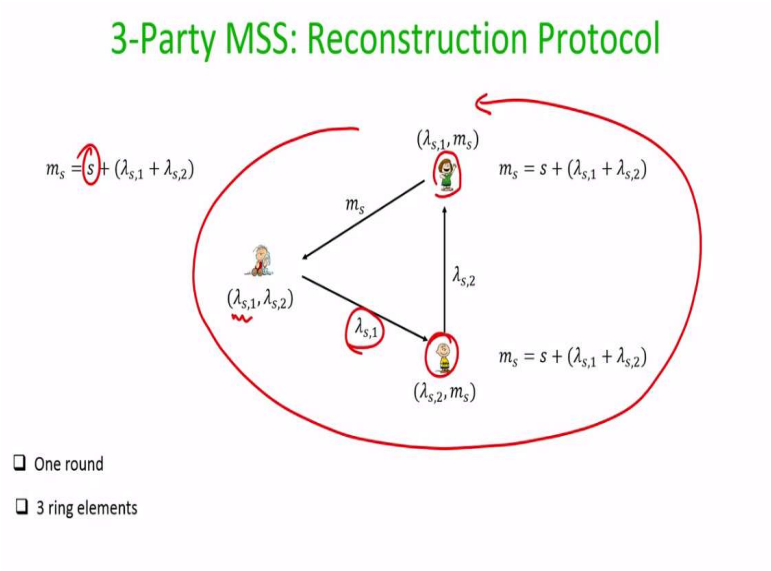
Well, the view 1 will have a probability distribution of λs_1, λs_2 and that is all because the distributor is not saying the OTP encryption, but this pads are completely independent of your secret s which has been shared by the evaluator E_1 because the distributor is not seeing the OTP encryption. Hence, the probability distribution of this pad is just a uniform distribution over the ring that is all which can be easily simulated.

Whereas if I consider a potentially corrupt E_2, what will be the view of E_2? The view of E_2 will be the second component of the pad namely λs_2 which is a uniformly random element from the ring which can be easily recreated or simulated it has got nothing to do with the secret and the OTP encryption of MS which is again a random element from the view point of a corrupt E_2 because it does not know the first component of the first part or the first share of the pad and hence this can be easily recreated.

And again like the previous secret sharing protocol for all the inputs or for all the sharing instances where E_1 is the dealer the distributor can pick the mask or the OTP pads and redistribute it during the pre processing phase well in advance. Even though the values which are going to be secret shared by E_1 they will be available during the circuit evaluation phase step 1 for all such future instances can be executed by the distributor during the pre processing phase.

So, that means what I am saying here is that you have now secret sharing protocol which has two steps associating the OTP pads and step 2 computing the OTP encryptions. The OTP encryptions computation can happen during the circuit evaluation phase by distributing the pads can happen during the pre processing phase. So, what I am saying is that the distribution of the pads can happen for all the instances of secret sharing which are going to happen during the circuit evaluation phase in advance during the pre processing phase itself.

(Refer Slide Time: 30:22)

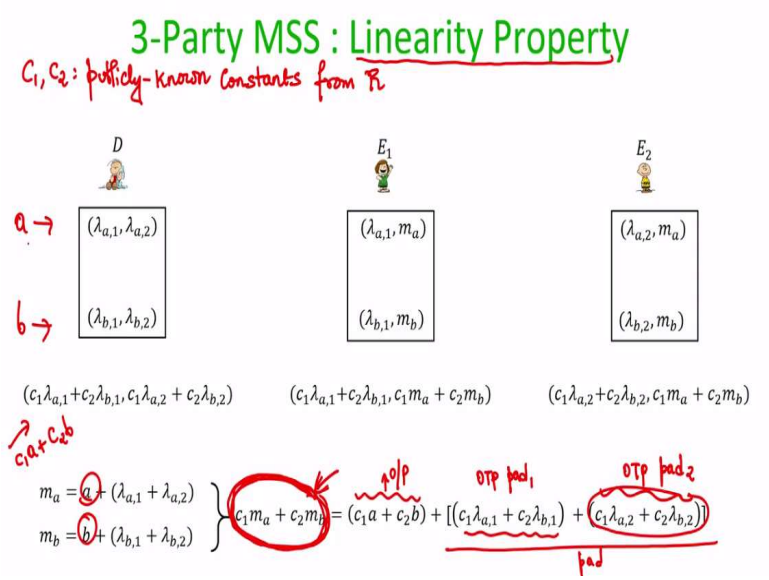


Now how do we reconstruct a value which is secret shared as per our 3 party MSS sharing semantic. So, imagine there is a value s which has been shared. Now we want to reconstruct this value. Suppose, this is the requirement. Now let us take the case of D the distributor. What is missing for the distributor to reconstruct a value the OTP encryption. So, we can ask in the sense we can designate the first evaluator to transfer the OTP encryption to the distributor.

And now he will have all the three things to get back the secret. What is missing for the second evaluator? He is missing the first share of the OTP pad that can be designated to the distributor, distributor can be designed to provide that information to the second evaluator and hence he will now have all the three pieces and what is missing for the first evaluator? He is missing the second component of the OTP pad that can be provided by the evaluator 2.

So, now you can see that we have followed a circular ordering in the communication where each party talks only to one of its neighbors. One of its neighbors means one of the parties and that is why it requires one round of communication and 3 ring elements.

(Refer Slide Time: 31:52)



Now like our all previous secret sharing schemes since we want to use this secret sharing scheme for shared circuit evaluation we want to explore whether it allows to compute linear functions of secret shared inputs namely whether this secret sharing schemes satisfies the linearity property and the answer is yes. So, imagine you have two values a and b two secrets which have been secret shared as per this three party MSS sharing semantic.

And imagine that you have two public constants c_1 and c_2 they are publically known constants from the ring. So, now we want to do the following without even disclosing anything about a and b is it possible for the parties to obtain the output of this linear function. Our linear function is $c_1 \cdot m_a + c_2 \cdot m_b$. So, we want to see here whether it is possible for the parties to compute this function output in the 3 party MSS sharing semantic where each of the 3 parties the distributor, the first evaluator, the second evaluator somehow obtain their respective shares as per the sharing semantic.

And the answer is yes because if we expand this value the output. This is supposed to be this long expression so let us pass this expression so this is the output and this is the OTP pad component 1 and this is the OTP pad component 2 sorry this is not the function output. What I want to say here is that if the parties perform this operation namely if the evaluators E_1 and E_2 compute $c_1 \cdot (OTP\ encryption\ of\ a) + c_2 \cdot (OTP\ encryption\ of\ b)$.

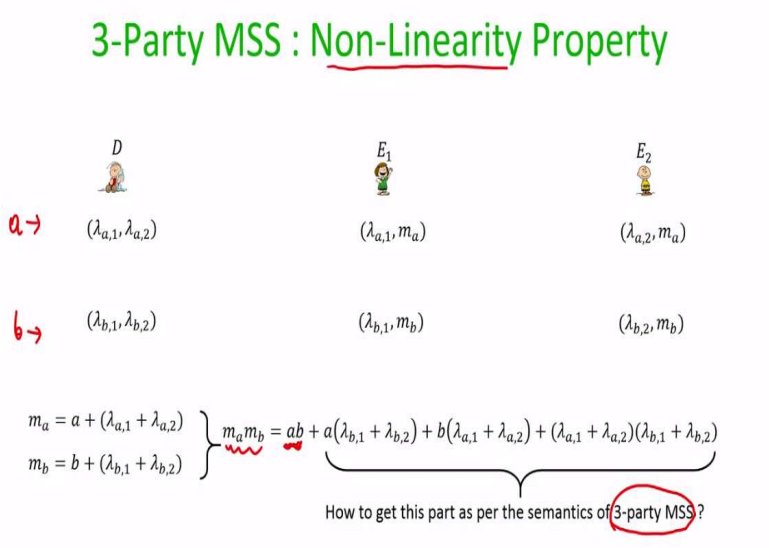
That will give you this whole expanded value and the expanded value in the RHS can be rewritten as or interpreted as the function output namely $c_1 \cdot a + c_2 \cdot b$ which is the output of the linear function being encrypted using this whole thing as a pad and this pad can be further

splitted into two components where the component 1 is $c_1 \cdot (first\ component\ a\ pad) + c_2 \cdot (first\ component\ of\ b\ pad)$ which the evaluator 1 can locally compute.

And the second component here is $c\,1 \cdot (2^{nd}\ component\ of\ a\ pad) + c_2 \cdot (2^{nd}\ component\ of\ b\ pad)$ which the second evaluator can locally compute. So, that means each of the 3 parties here the distributor, the evaluator 1, evaluator 2 based on whatever information they have corresponding to the secret sharing of a and b they can locally compute their respective information for the secret sharing of the output of the linear functions $c_1 \cdot a + c_2 \cdot b$.

And this shows that any linear function over the ring of secret shared inputs can be locally or non-interactively computed.

(Refer Slide Time: 35:36)

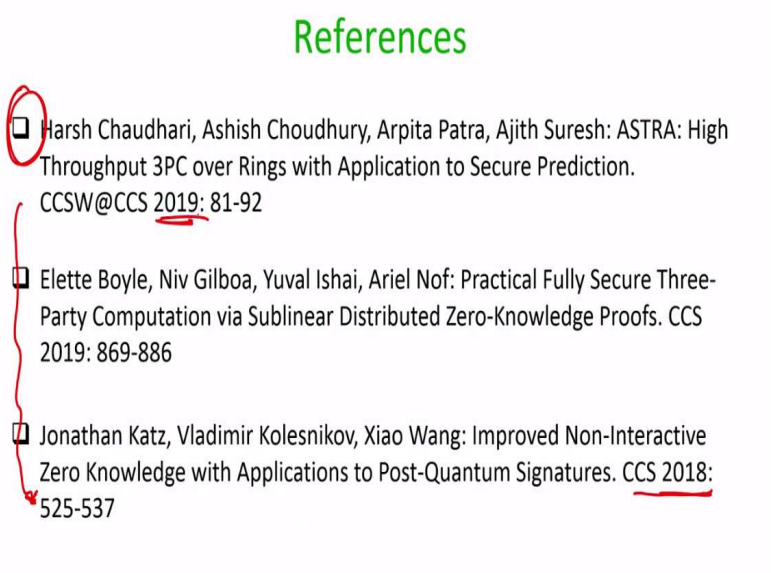


Now, what about the non interactivity property or whether the parties can compute something similar with respect to multiplication of inputs and the answer is no because imagine that value a and value b they have been secret shared as per 3 party MSS sharing semantic and if we just ask the evaluator E_1 and E_2 to multiply the OTP encryptions of a and b then the expansion will be this.

So, we want to check whether this is the output we want so a secret shared, b secret shared we want $a \cdot b$ to be secret shared $a \cdot b$ secret shared means $a \cdot b$ should be encrypted with some OTP pad and that OTP pad should be splitted into two pieces where one of the pieces is with one evaluator other piece with other evaluator and the whole pad is with the distributor.

But it turns out that we cannot take the rest of the things in the expansion that we have in the RHS and ensure that it somehow falls as per the semantics of 3 party MSS secret sharing semantic. So, that means this 3 party MSS secret sharing which we have discussed has the non linearity property. It does not allow you to locally compute non-linear functions of secret shared inputs.

(Refer Slide Time: 37:11)



So, in the next lecture we will see that how do we evaluate multiplication gates or compute non-linear functions of secret shared inputs and the 3 party MSS sharing semantic. So, the protocol or the sharing semantic that I discussed today has been taken from this paper. It also appeared in various other forms. So, it actually can be taken as a special case of a secret sharing protocol proposed in this 2018 paper.

But when we wrote this paper in 2019 we were not aware of that fact and later on this paper further improved the 3 party protocol which we are going to see in the next lecture and proposed in this 2019 paper. Thank you.