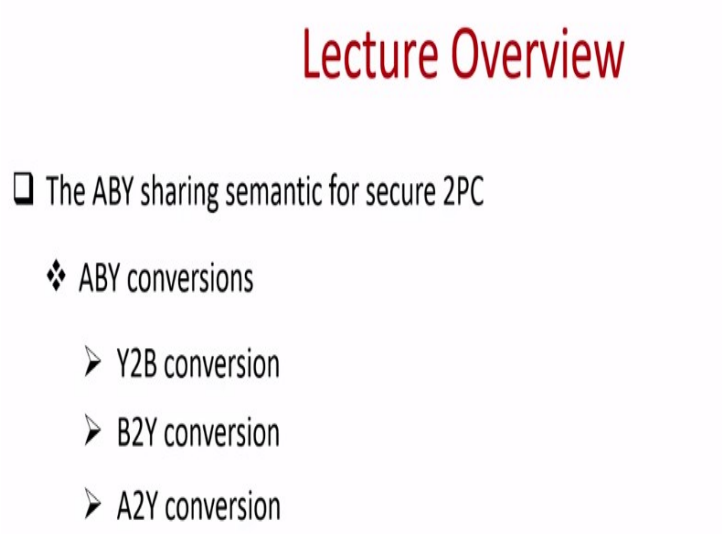
Secure Computation – Part 1
Prof. Ashish Choudhury
Indian Institute of Technology – Bangalore

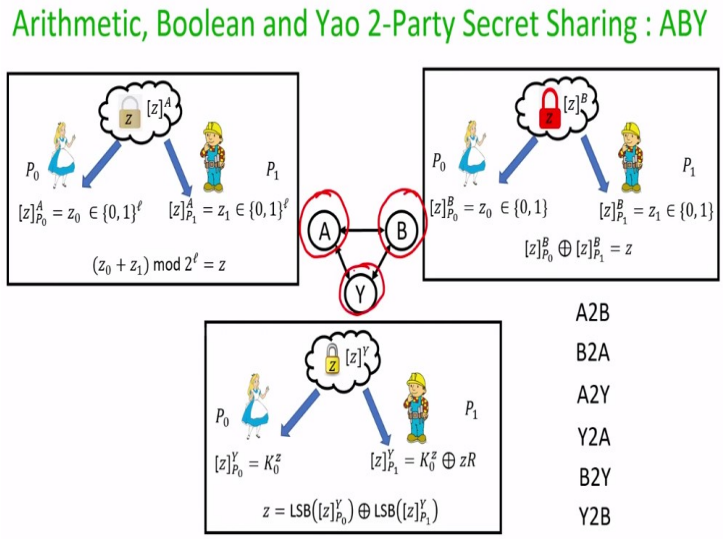
Lecture – 55
The ABY Conversions

(Refer Slide Time: 00:31)



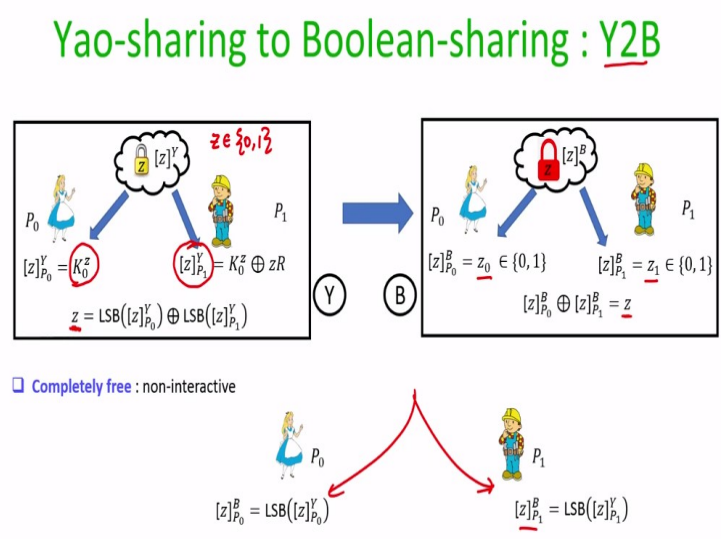
Hello everyone, welcome to this lecture. So, we will now start discussing the conversions among the various sharing semantics. And in today's lecture, we will see three of the conversions; Yao's to Boolean conversion, Boolean's to Yao conversion and arithmetic to Yao conversion.

(Refer Slide Time: 00:48)



So, just to quickly recap these are our 3 differentiating semantics; we have the arithmetic secret sharing semantics, we have the Boolean secret sharing semantics and we have the Yao secret sharing semantics.

(Refer Slide Time: 01:04)



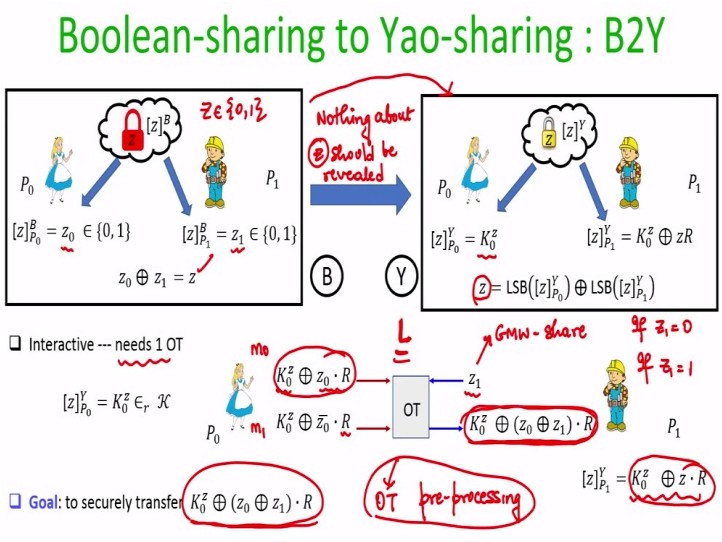
So now let us start with a very simple conversion, namely Yao sharing two Boolean sharing. So the input for this conversion is as follows. We have a bit Z which is secret shared as per the Yao's representation. Namely, the share for Alice is the 0 key and the share for Bob is the key corresponding to the actual value of Z. And the permutation bits of the Alice's share and Bob's share if we XOR them that gives us the value of Z that is a relationship.

We now want to do the following. Without disclosing the value of the bit Z, we want to compute for Alice her GMW share and we want to compute for Bob his GMW share, which are random GMW shares such that if we XOR them that gives us the same value Z which is Yao's secret share that is the output we require here. And it turns out that this conversion is completely free. It is noninteractive. It does not require any interaction between Alice and Bob.

What Alice and Bob have to do? Alice can set her GMW share to be the permutation bit of the key which is her Yao's share. So, she is processing her Yao share which is a 0 key, she just takes the permutation bit of that key and that she sets as her GMW share. And in the same way Bob, he takes his Yao's share, focus on the permutation bit, and that permutation bit he is setting as his GMW share.

And it is easy to see that since the XOR of the permutation bits of Alice Yao's share and Bob's Yao's share is actually the bit Z. The same property holds even for the GMW share which Alice and Bob have computed and that is what we want to ensure. And that is why going from the Yao world to the Boolean world is completely free, it does not require any interaction whatsoever among the parties. So that is very simple.

(Refer Slide Time: 03:40)



Now, let us discuss how to go from Boolean world to the Yao world. Namely, we have a big Z which is secret shared as per the GMW Boolean representation. Namely, Alice's share is Z 0, Bob's share is Z 1 such that the XOR of Z 0 and Z 1 is the bit Z that is an input for this conversion. The output of the conversion should be as follows. We require that Alice should possess a 0 key associated with Z.

And Bob should possess the key corresponding to the actual value of Z such that if you take the XOR of the permutation bits of Alice's Yao's share and Bob's Yao's share that should be the value of Z and in the process nothing about Z should be revealed that is important. Because if this constraint is not there, we do not require the value of Z to be kept private, then the conversion is very simple, we can ask Alice and Bob to run the GMW reconstruction protocol, reconstruct Z.

And now if Z is publicly known, Alice and Bob can come up with the default Yao sharing for the value of Z. But that is not what we desire here, we want that the value of Z should not be disclosed and even without disclosing the value of Z, we want to go from this input setting to

this output setting. And this conversion is no longer noninteractive, this requires interaction between Alice and Bob and this interaction is through oblivious transfer.

So, to get the Yao shares Alice and Bob can do the following. So since we want the output to be available in the Yao's representation, Alice has to do the garbling of the bit Z. So, she will pick the 0 key uniformly randomly and she will also have the key corresponding to Z = 1 which she can obtain by XORing this 0 key with a global offset. And our goal is to do the following, we want to somehow transfer this key to Bob because that will be Bob's Yao share for the value Z.

But in the process, nothing about the bit Z should be revealed. And this can be done through an instance of oblivious transfer where Bob participates with his GMW share as the selection bit for the OT. So, he is participating as a receiver, his selection bit will be the GMW share and Alice participates as the sender where her two messages are these factors okay and she has everything to prepare the two messages.

So, Z 0 is her GMW share, 0 key associated with 0 she has computed randomly and global offset is anyhow known to Alice. Now, as per the correctness property of the oblivious transfer, Bob as a receiver will obtain this value and that is what we want him to obtain, why so? If Z 1 = 0, then he will obtain m 0 and if I substitute Z 1 = 0 indeed the output of Bob in the OT is the message m 0.

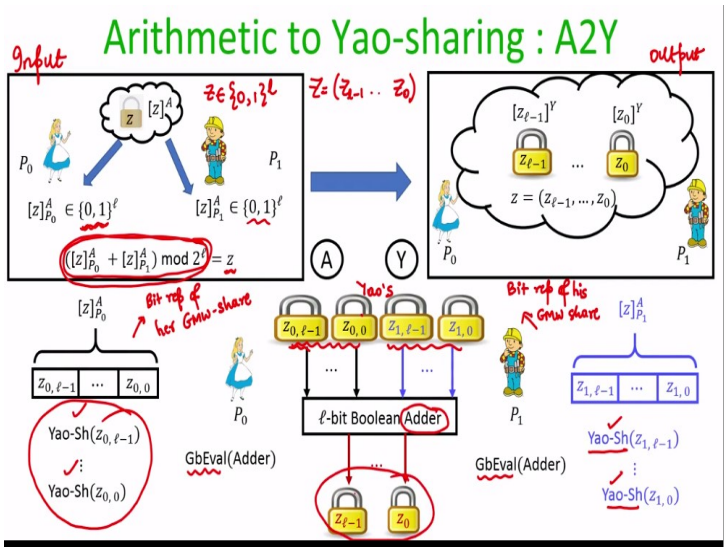
Whereas if Z 1 = 1, then he should get the message m 1 and indeed if Z 1 = 1 the receiver's output or Bob's output is the message m 1 and if I see this value Bob's output in the OT that is nothing but what he is supposed to receive as per the Yao's two-party sharing semantic right. As per the Yao's two-party secret sharing semantic Bob should have this key and indeed he has got this key by acting as a receiver in the OT instance.

So, now we can imagine that in our mixer world computation, parties can identify beforehand what are the Boolean to Yao conversion which are going to be involved in the protocol. They can identify that because they will be knowing the description of the circuit, which part of the circuit requires what kind of computation and they will be now knowing what exactly are the OT instances.

But what are the inputs for those OT instances they will be determined at the run time depending upon the values which are obtained during the circuit evaluation. But what Alice and Bob can do is they can always do the OT pre-processing, namely the OT extension and if they know beforehand that they have to participate in l number of Boolean to Yao conversions, then they can prepare those many OTs beforehand by doing the pre-processing of the OTs.

And now once the GMW shares of Z are ready, as and when the conversions are required they can run the desired OTs without actually running the OT instances because the OT instances would have been already executed with dummy inputs in the pre-processing phase during the OT extension protocol. So that is how the Boolean to Yao conversion can be done.

(Refer Slide Time: 10:17)



Now, let us see the last conversion for today's lecture, namely the arithmetic to Yao sharing. And input for this conversion is the following. We have an l-bit number now which is secret shared as per the GMW representation. Namely Alice share is an l-bit number, Bob's share is an l-bit number such that if we sum them modular 2 power l, then that gives us the l-bit number Z that is the input.

The output that we desire here is the following, we want each and every bit of Z. So, imagine that the Boolean representation of Z is this the bits of Z are Z 0 to Z l–1 from LSB to MSB. Neither Alice nor Bob knows the full binary representation of Z. They only have the arithmetic shares of Z, namely they have two l-bit numbers which when added modular 2 power l gives Z.

What we now want is we want a switching mechanism where without even disclosing the values of the bits of Z, the individual bits of Z should be made available to Alice and Bob in Yao's secret sharing representation that is what we desire here. So, this can be done in several ways. One of the ways using which we can do this conversion is as follows. So, what Alice can do is the following. She can take the bit representation of her GMW share.

She has her GMW shares, so I am focusing on the bit representation of her GMW share. And in the same way Bob has his GMW share, so he focuses on the bit representation of his GMW share. And now what we know is the following. We know that there is a publicly known l-bit Boolean adder which when given the GMW shares of Alice and Bob in bit representation can add them and produce the bit representation of the input Z, we know this.

There is always a l-bit Boolean adder. Why so because we know that the l-bit number Z is the summation of these two l-bit numbers where one of the l-bit numbers is held by Alice as her GMW share and the other l-bit number is held by Bob as his GMW share. So, what I am simply saying is that there is always this l-bit Boolean adder which can add the GMW shares of Alice and Bob and produce the l-bit number Z.

So, the idea behind this A to Y conversion is to let Alice and Bob securely evaluate this l-bit Boolean adder. What does that mean? That means that even without disclosing the bit representation of her Boolean share and without letting Bob disclosing the Boolean representation of his GMW share, we would like Alice and Bob to evaluate this Boolean adder and that too in the Yao's representation.

So, for doing that what Alice can do is the following. She takes her bits of GMW share and she acts as a dealer and invoke instances of Yao's secret sharing protocol and as the dealer she secret shares these bits. So now you can imagine that these bits, namely the bits of a GMW share are secret shared between Alice and Bob as per the Yao's secret sharing representation.

And this is done because Alice has acted as a dealer and secret shared those bits and independently Bob can do is the following. He acts as a dealer and secret share the bits of his GMW share by acting as a dealer and invoking the corresponding instances of Yao secret

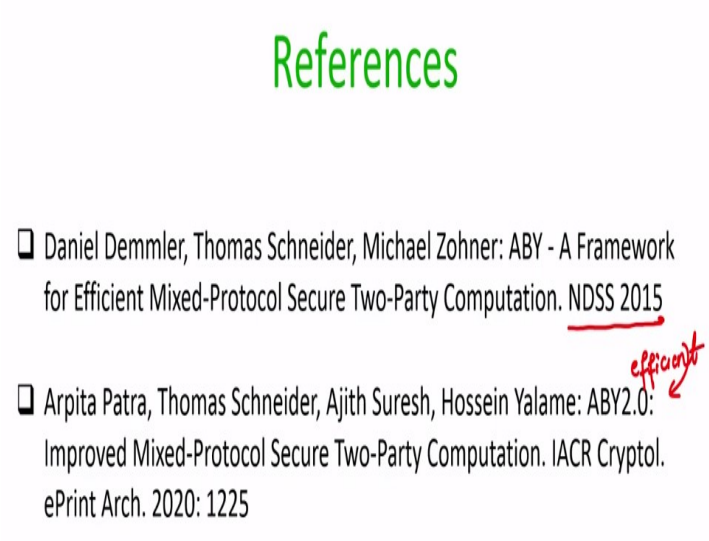
sharing protocol. Now, in this process Alice does not learn the exact bits of Bob's GMW share because that is coming from the privacy of the Yao secret sharing instances.

And in the same way Bob does not learn anything about the bits of Alice's GMW share because those are secret shared as per the instances of Yao secret sharing. Now, both Alice and Bob have the inputs of this Boolean adder circuit available in the secret sharing representation of Yao's sharing, what they can do is they can evaluate this circuit, but in the Yao's representation, not in the GMW representation because the inputs of this circuit are secret shared as per the Yao secret sharing.

So, they can now evaluate this circuit and now we know how to evaluate a Boolean circuit if the inputs of that circuit are available in the Yao secret sharing representation. And that will ensure that the output of Alice and Bob will be Yao's share for the individual bits of Z and that is what precisely we want here. So, the conversion protocol here is Alice secret share the bits of her GMW share, Bob secret share his bits of GMW share.

And then together they evaluate this l-bit Boolean adder in the Yao domain and obtain the output of the adder circuit in the Yao domain in the secret sharing representation. So that is how we can do the arithmetic to Yao conversion.

(Refer Slide Time: 16:58)



So, these are the references used. The conversions that are discussed in today's lecture they are taken from the original ABY paper. Of course, as I said, now we have more efficient

conversions available. So, if you are interested to know more about the efficient conversion, you can refer to this paper. Thank you.