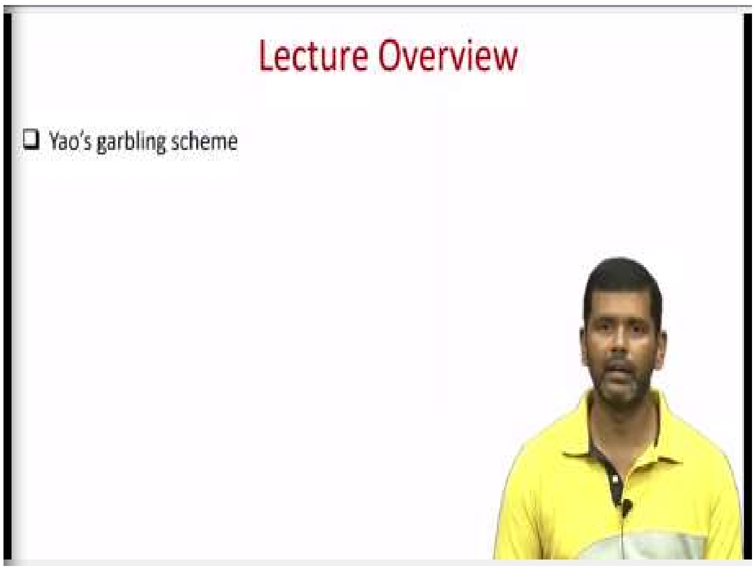
Secure Computation: Part I

Prof. Ashish Choudhury
Department of Computer Science Engineering
Indian Institute of Technology-Bengaluru

Lecture-49
Yao's Garbling Scheme

(Refer Slide Time: 00:30)

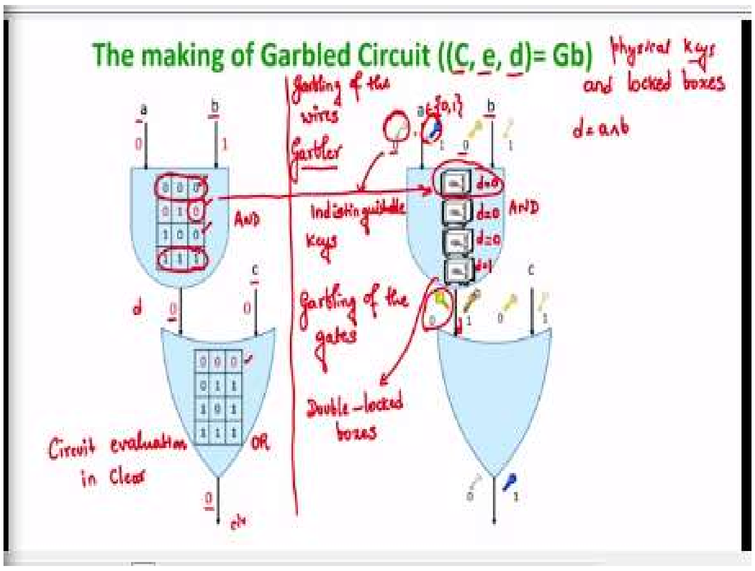


Lecture Overview

❑ Yao's garbling scheme

Hello everyone, welcome to this lecture. So, in today's lecture we will discuss the Yao's garbling scheme. In the last lecture we had seen the syntax and semantics of the abstract garbling scheme. But today's lecture we will see how to instantiate a garbling scheme by looking into the Yao's garbling scheme.

(Refer Slide Time: 00:54)



So, we will have to first see how exactly the gates are garbled, the inputs are garbled and how exactly a garbled circuit is evaluated on garbled inputs. So, we have to see into the construction of this garbled circuit and the details of this encoding information, e enter in details of the decoding information. And for the purpose of illustration I will take this simple Boolean circuit which has an AND gate here and OR gate.

And again for simplicity assumes that the exact values of the inputs a, b and c are 0, 1, 0 respectively. So, if the values of a, b and c are available and clear, then it is very easy to evaluate the circuit. So, what we have to do is the following. So, if you see the AND gate, we have a corresponding truth table and depending upon the values of a and b, we will know that ok we have to go and get the value from the 2nd row of the truth table.

And if a and b are 0 and 1 respectively then the output of the AND gate will be 0. And hence over the output wire of this AND gate the value 0 will be there. And now we go to the next gate which is the OR gate and we have the truth table of the OR gate, since in both the inputs of the OR gate are 0. We have to consider the 1st row of the truth table of the OR gate and then we will obtain finally the function output 0.

So, this will be the circuit evaluation, this is your circuit evaluation in clear and clearly this is not what we will do in our MPC protocol because this circuit evaluation in clear requires the exact values of the inputs of this function to be available and that completely violates privacy. Rather we have to evaluate the circuit in a different way, so let me 1st explain the idea behind the Yao's garbling scheme in terms of physical keys and locked boxes.

And later once the idea is clear and we will go and see how exactly the keys and the locked boxes are instantiated using mathematical primitives. So, the 1st step in the Yao's garbling scheme is the garbling of the wires. What do we mean by wires here? So, we are given a Boolean circuit here, we can imagine that the values a and b are passing through the input wires and then there is a wire coming out as an output of the AND gate.

And then we have a OR gate, we were these are the wires, input wires and this is output wires. So, the 1st step in the Yao's garbling scheme is the garbling of the wires and to do the garbling of the wires one of the parties. So remember, Yao's protocol is a 2-party protocol, so one of the parties is assigned the role of garbler. And garbler will be doing the garbling of the wire, the garbling of the circuit and he has to prepare the encoding information and he has to prepare the decoding information.

So, whatever I am discussing now is in the context of the garbler. So, what a gobbler does is the following? We know that this value a can take only one of the 2 possible values. The input a can take one of the 2 possible values either it can take the value 0 or it can take the value 1. So, corresponding to the value of a = 0, the gobbler assigns a key and corresponding to the value of a = 1, the garbler assigns another key.

And these 2 keys are indistinguishable keys, indistinguishable in the sense that only garbler knows that ok this key is corresponding to a = 0 and the other case corresponding to a = 1. That means only garbler knows the labels of this case. Later on during the circuit evaluation, the other party who is going to evaluate the circuit will be given one of these 2 keys but not the corresponding label.

So, if later the circuit evaluator obtains one of these 2 keys and he is not told about the label, then by indistinguishability of the keys, I mean that it cannot figure out whether the key that it has obtained corresponds to with a = 0 or it corresponds to bit a = 1. But a garbler if he is seeing a key it can find out whether the key corresponds to a = 0 or a = 1 because he will be knowing the label of those keys.

In the same way garbler does the following. Corresponding to the wire b, it assigns 2 indistinguishable keys, they are random keys 2 indistinguishable keys. And in only the garbler knows the labels of those keys. In the same way the output of this OR gate, it is assigned 2 possible keys one corresponding to value 0 another corresponding to value 1. In the same way the wire that label c is assigned 2 indistinguishable keys one corresponding to 0 another corresponding to 1, all of them are random keys.

And in the same way, finally the output of this OR gate is assigned 2 indistinguishable keys one with labels 0 another with label 1, that is constitutes the garbling of the wires. Now the next step is the garbling of the gates. And again this step is performed by the garbler itself. So, we can imagine that we have 2 parties, 1 party will be assigned a role of garbler and he will be making an entire scan of the circuit starting from input wires all the way to the output wire.

And it will 1st garble each and every wire and then it will garble each and every gate and then it will give the garbled circuit to the evaluator and the garbled inputs to the evaluator. And that will complete the Garbler's route. And then the route of the evaluator start, where evaluator now have to make an entire scan of the circuit and it has to evaluate each gate but not on clear inputs but rather on garbled inputs.

So, what I am right now discussing is the other steps of the garbler. So, once the wires are garbled, the next step is to garbling of the gates. And by garbling of the gates I mean the following. So, this is an AND gate both the parties will know that, ok, this is an AND gate. The structure of the circuit is publicly known to both the parties and they will know that, ok, what is the 1st gate and then what is the next gate and what is the third gate and so on.

They will know the full topological ordering. So, for garbling this AND gate, the garbler prepares 4 double locked boxes, so they are double locked boxes. By double locked boxes I mean that each of the box will be locked with 2 keys, we will see what are those 2 keys. And inside the box another key will be kept. So, we have to now see what are the keys with which each of the box is locked and what exactly is kept inside the box, that constitutes the garbling of this gate.

And similar logic will be performed for the OR gate and then the next gate and the next gate and so on. So, for the 1st double locked box, that garbler is going to put the key corresponding to bit 0 over the output wire of the AND gate. So, remember the output wire of the AND gate, let us call it as d, so d = a and b, let us give the wire, label d. So, corresponding to d, the garbler has assigned 2 keys 1 corresponding to d = 0, 1 corresponding to d = 1.

What I am saying here is that in the 1st 2 double locked box, the garbler is going to keep a copy of the key the d = 0. In the 2nd box, also he is going to keep a copy of the key d = 0. In the 3rd box also he is going to keep a copy of the key d = 0. And in the 4th box, he keep a copy of the key d = 1. By the way I am assigning it is possible to create multiple copies of a physical key, garbler can do that.

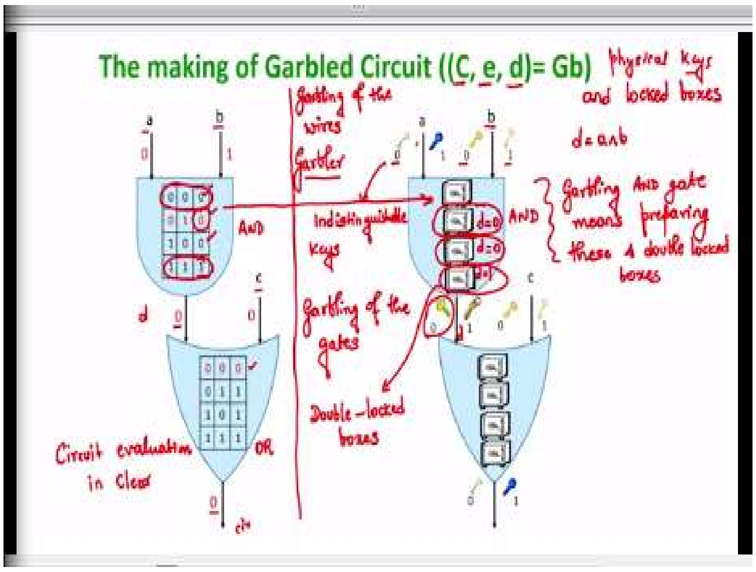
Now why he has kept the keys corresponding to d = 0 in 3 boxes and key corresponding to d = 1 in one of the boxes. This is because what the garble is doing is, it is trying to imitate the physical truth table here. If you see the physical truth table or the actual truth table of the AND gate, we know that the output of the AND gate d can take the value 1 only in one case. Namely, when the inputs a and b are both 1, whereas in all the remaining 3 cases it is going to take the output, it is going to take the value 0.

And that is what precisely the garble has done here. In one of the boxes, he has kept the key corresponding to d = 1 because that imitates the 4th row of the truth table. Whereas the remaining 3 boxes they are the key corresponding to d = 0 is kept because those 3 rows imitates the remaining 3 entries of the truth table. And now he is going to lock each of these 4 boxes using 2 keys.

One of the keys will be corresponding to the bit a another key will be corresponding to bit b. Now for a there are 2 possible keys one for a = 0, one for a = 1 and in the same way for b there are 2 keys one for b = 0 and one for b = 1. We have to decide which of the 4 boxes is going to be locked using which a, b key combination. So, let us take the first box here. The first box is going to be locked by the garbler using the keys corresponding to a = 0 and b = 0 and why so?

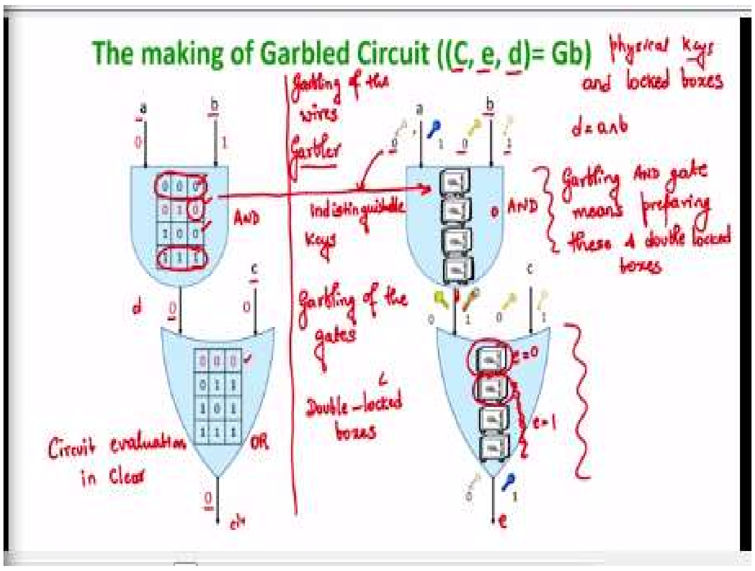
Because if you see here he is trying to imitate the 1st row of this truth table, in the 1st row of the AND truth table. We are considering the case where a is 0, b is 0 and output is 0, hence this 1st double locked box is locked using the keys corresponding to a = 0, b = 0. And what is kept inside that box? The key corresponding to d = 0.

(Refer Slide Time: 14:03)


The making of Garbled Circuit ((C, e, d)= Gb)
Physical keys and locked boxes
Garbling of the wires
Garbler
Indistinguishable keys
Garbling of the gates
Double-locked boxes
d = a∧b
Garbling AND gate means preparing these 4 double locked boxes
AND
OR
Circuit evaluation in Clear
d=0
d=0
d=1

If we take the 2nd locked box, then that is locked using the key combination a = 0, b = 1 because that imitates your 2nd row of the truth table. If we take the 3rd double locked box, it is locked using the keys a = 1, b = 0. And if you take the 4th double locked box it is locked with the keys corresponding to a = 1 and b = 1 and that completes the garbling of this AND gate. So, garbling the AND gate means preparing these 4 double locked boxes. Now the garbler has to go to the next gate which is the OR gate.

(Refer Slide Time: 15:18)

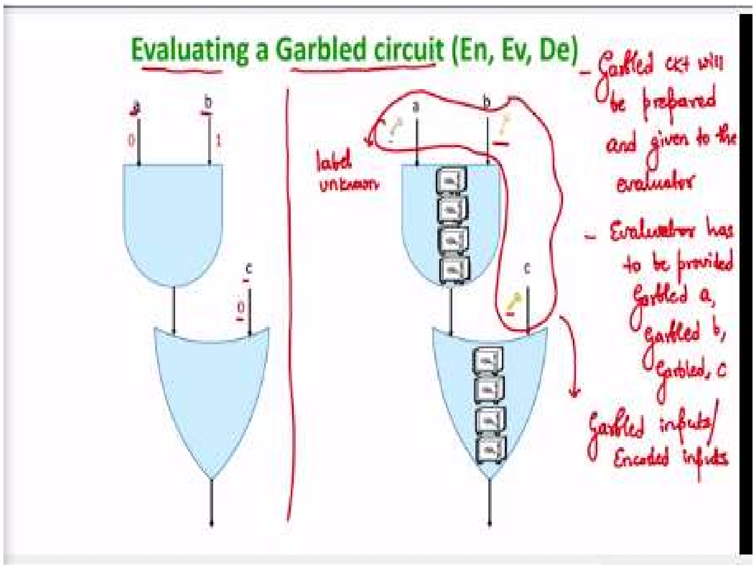

The making of Garbled Circuit ((C, e, d)= Gb)
Physical keys and locked boxes
Garbling of the wires
Garbler
d = a∧b
AND
Indistinguishable keys
Garbling AND gate means preparing these 4 double locked boxes
Garbling of the gates
Double-locked boxes
Circuit evaluation in Clear
OR
e=0
e=1
a
b
c
d
e

And again for the OR gate he will be doing the same thing here. So, now let us call this wire as d, so the input of this OR gate have the labels c and d respectively. So, he prepares 4 double locked boxes now. And in 3 of the boxes he is going to keep the key corresponding to value 0 here. So, let us give a label to the output wire of this gate as well, so let us call the label of this wire as e.

So, in this last 3 double locked boxes the key corresponding to e = 1 will be kept and in the 1st locked box the key corresponding to e = 0 will be kept. The 1st locked box will be locked using the key combination d = 0, c = 0, 2nd locked box will be locked using the key combination d = 0, c = 1 and so on, that will complete the garbling of this 2nd OR gate. Remember, all the steps are done by the garbler; the role of the evaluator has not yet started.

Now the garbler if there is a follow up gate, he will do the same thing for the 3rd gate and so on. So, that constitutes the garbling of the gates. So, garbling of the wires have been done, garbling of the wires means assigning 2 indistinguishable keys to each wire and noting down the labels and garbling of the gate means preparing 4 double locked boxes for each of the gates, imitating the truth table.

(Refer Slide Time: 16:59)


Evaluating a Garbled circuit (En, Ev, De)
a
b
0
1
c
0
label unknown
a
b
c
Garbled ckt will be prepared and given to the evaluator
Evaluator has to be provided garbled a, garbled b, garbled c
Garbled inputs/ Encoded inputs

Now, let us see how exactly a garbled circuit will be evaluated. So, imagine that the role of the garbler has been done; he has done the wire garbling and the gate garbling. Now depending upon the inputs of the function, namely the exact values of a, b and c, we have to give some information to the evaluator, so that it can evaluate the circuit. So, the way evaluation will happen is as follows.

So, this circuit, this garbled circuit will be prepared and given to the evaluator, who is the other party. And now depending upon the concrete values of a, b and c, evaluator has to be provided garbled a, garbled b and garbled c or encoded a, encoded b and encoded c. So, let us see the encoding information here. So, imagine a = 0, who owns the value of a? Forget about that, those things will be coming later when we will see that full details of the Yao's protocol.

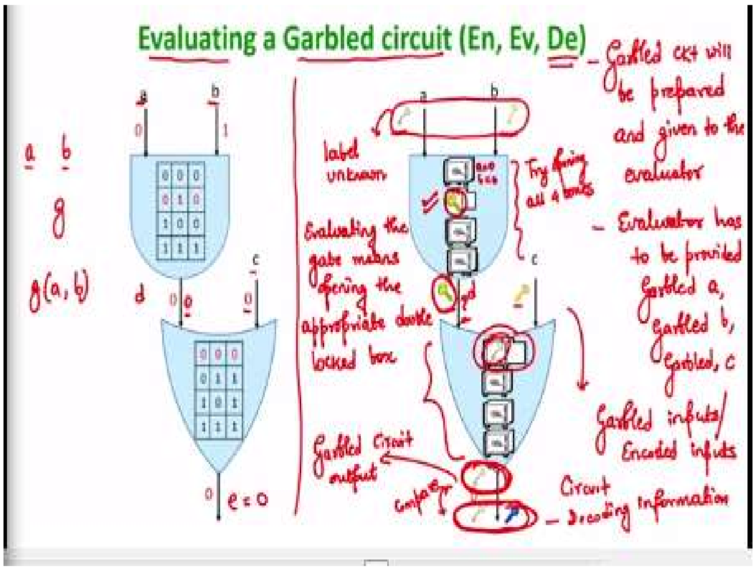
Imagine for the moment the values of a, b and c are completely owned by the garbler itself. But we can later consider the cases when some of the inputs a, b, c are held by one of the parties and remaining inputs are held by the other party. So if a = 0, the evaluator will be provided with the key corresponding to a = 0, but it will not be told, it will not be the evaluator will not be knowing what exactly is the label of the key.

So, label will be unknown and it will be given this key. Now since we assumed that the keys are indistinguishable, that means the garbler has prepared 2 keys one corresponding to a = 0 and a = 1 and only the garbler knows the labels. But since only the unlabeled key is given to the evaluator, it cannot figure out whether it is seeing the key corresponding to a = 0 or a = 1 and that maintains the privacy of the value of a, it just got 1 key corresponding to this wire a.

And imagine the value of b = 1, the encoded b will be the key assigned to b = 1 without the label, so again the label will not be known. So, you can imagine that getting the encoded input is equivalent to getting the keys corresponding to those input values but without knowing the label. And since the keys are built in such a way that they are indistinguishable, just by seeing the keys evaluator cannot figure out whether it has seen the key corresponding to a = 0 or a = 1 or whether it has seen the key corresponding to b = 0 and b = 1.

And in the same way, imagine say c = 0. So, the evaluator will be seeing the key corresponding to c = 1. So, this is what constitutes the garbled inputs or encoded inputs. So, you can imagine that the encoding function here is very simple corresponding to the value of the inputs, just output the keys corresponding to those values of the inputs that is all, without the labels. Now, the evaluator has to start evaluating the garbled circuit. So, it has to evaluate the circuit gate by gate, so it will start with the 1st gate here.

(Refer Slide Time: 21:50)



Evaluating a Garbled circuit (En, Ev, De)
a b
g
g(a, b)
a
b
0
1
000
010
100
111
d
0
c
0
000
011
101
111
0
e = 0
label unknown
Evaluating the gate means opening the appropriate double locked box
Try opening all 4 boxes
Garbled circuit output
output
- Garbled ckt will be prepared and given to the evaluator
- Evaluator has to be provided garbled a, garbled b, garbled, c
Garbled inputs/ Encoded inputs
Circuit - Decoding information

And evaluating the gate means opening the appropriate double locked box, what does it mean? So, if I consider this AND gate, the evaluator is saying 4 locked boxes here. And now he has got one key corresponding to some value of a, another key corresponding to some value of b, right now, it does not know what are the values of a and b because the labels of these keys are not known. And for evaluating this garbled AND gate, what the evaluator has to do is?

It has to find out what should be the unlabeled key over this wire during the evaluation? We require the following invariant. If the values of a and b or 0 and 1 respectively for this AND gate, we know that during the circuit evaluation in clear over this output wire of the AND gate, the value 0 will be coming. We require that during the garbled circuit evaluation, if the evaluator has the keys corresponding to the bits a and b by evaluating the garbled AND gate somehow he should obtain the appropriate key corresponding to the output wire of the AND gate corresponding to output bin 0.

So, if the wire label here is d, we know that if a and b are 0 and 1 respectively, then the value of d should be 0. The invariant that should be maintained in the garbled circuit evaluation is that if the evaluator has the keys corresponding to a being 0 and b being 1. Then after evaluating the garbled AND gate, he should obtain the key corresponding to d = 0. But in the process, it should not learn whether the key that it got over the d wire is actually for d = 0 or d = 1.

Now, let us see whether this happens or not. So, in this case, the evaluator has to open the appropriate double-locked box and it does not know which of the locked box it has to go and open. So, the only thing that it has to do is, try opening all 4 boxes and it is only one of the 4 boxes which will be opened for him, why so? Because we are assuming here that if you do not have the appropriate key, then you cannot open the corresponding locked boxes.

So, if I consider saying, for instance, the 1st locked box it is locked using the key combination a = 0 and b = 0. But the keys which you evaluated has right now are corresponding to a = 0 and b = 1. And using that key combination the evaluator can open only this 2nd locked box. And if it opens the 2nd locked box, it will see a key kept inside and that is the key which will be now assigned to the output of this AND gate.

However, the label of this key which it now assigned to the output of this AND gate is not known to the evaluator. Because when the gates were prepared or the 4 double-locked boxes for prepared, the garbler has not kept the label of the keys which are kept inside the boxes. Only garbler knows what key is keeping inside the boxes, whether it is the 0 key or whether it is the 1 key, but the labels are not kept inside the box.

So, from the viewpoint of the evaluator, he just obtained a key corresponding to some value of d whether it is for d = 0 or whether it is d = 1, it cannot figure out. And now it has to go and repeat the same process for the next gate. For the next gate also, he has 4 locked boxes here. Now he has 1 key over the d wire, 1 key for over the c wire, he should try to open the appropriate one of the 4 locked boxes and find out what the key it obtains over the output wire here.

So, again if we compare the circuit evaluation in clear, we know that if the value of d would have been 0 and the value of c would have been 0, e would have taken the value 0. Now let us see whether the same invariant is maintained during the garbled circuit evaluation. Over the d wire, the evaluator has the key corresponding to b = 0, over the c wire evaluator has the key corresponding to c = 0.

And using those 2 keys, it is only the first double locked box which will get opened. But again, after opening the first double locked box, the evaluator will not be able to find out what exactly that key corresponds to, whether it corresponds to e = 0 or e = 1. And after this, there is no other gate, so the garbled circuit evaluation is over. So, in some sense you can imagine that garbled circuit evaluation is nothing but evaluating the circuit over unlabeled keys.

For each of the wires, the evaluator will be processing only one of the 2 keys, even though 2 keys are assigned for each of the wires by the garbler. But at the time of evaluation for each wire the evaluator will be seeing only one of the 2 are labeled keys. And the invariant that will be maintained is the following. If the gate is a g with input a and b and the output is g of a, b. And depending upon the exact values of a and b the evaluator will be having the corresponding unlabeled at a key and b key.

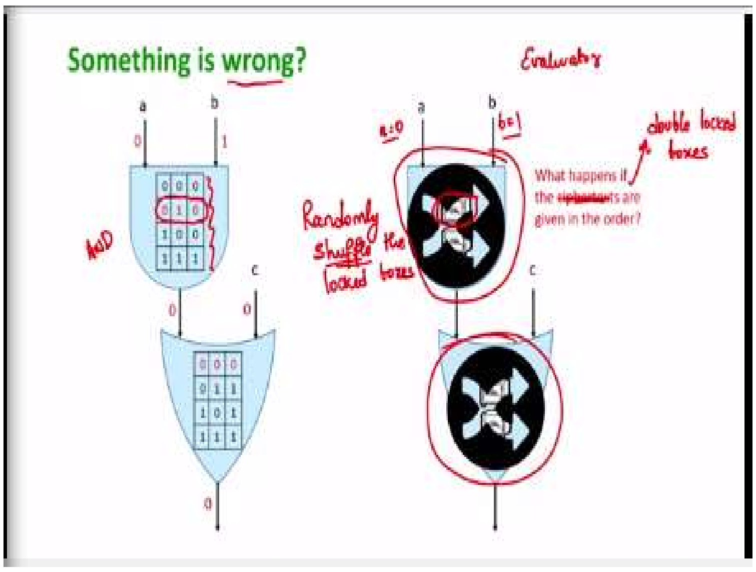
And by opening the appropriate double locked box, the evaluator will obtain the key corresponding to g of a, b over the output of that corresponding gate. That same variant which will be maintained during the evaluation of the garbled circuit and evaluation should happen in such a way that just by seeing the keys it labels cannot be identified. So, now once the circuit evaluation is over. So, we have seen that how the garbling happens? How the input is encoded?

We have to now see the decoding information because that is also a part of the garbling function. Because right now the evaluator has obtained the garbled output, this is the garbled circuit output. But this garbled circuit output has no meaning for the evaluator because it is just an unlabeled key, whether that key is for equal to 0 or whether that key is for equal to 1, it cannot figure out, until and unless the decoding information is provided to the evaluator.

So, the decoding information is nothing but the label of both the keys assigned to e, so this is the decoding information. Namely, both labeled keys corresponding to the output wire e along with the labels. And this the garbler can provide to the evaluator, when the garbled circuit was transferred to the evaluator the decoding information can also be transferred. And that is not a breach of privacy, learning both the keys corresponding to e because after that, there is no follow up gate to be evaluated.

Now, how exactly this decoding information will help the evaluator to decode the garbled circuit output? It has to just to do a simple comparison. So, after evaluating the whole circuit, the evaluator will obtain this unlabeled key. And now it has both the labels corresponding to the keys assigned to the output wire e, by doing the simple comparison it can find out at ok, it has obtained a garbled output which is the key corresponding to e = 0. And hence the function output in this case is e = 0, that is the way evaluation will happen.

(Refer Slide Time: 31:09)


Something is wrong?
Evaluator
a
b
c
AND
Randomly shuffle the locked boxes
a=0
b=1
double locked boxes
What happens if the ciphertexts are given in the order?

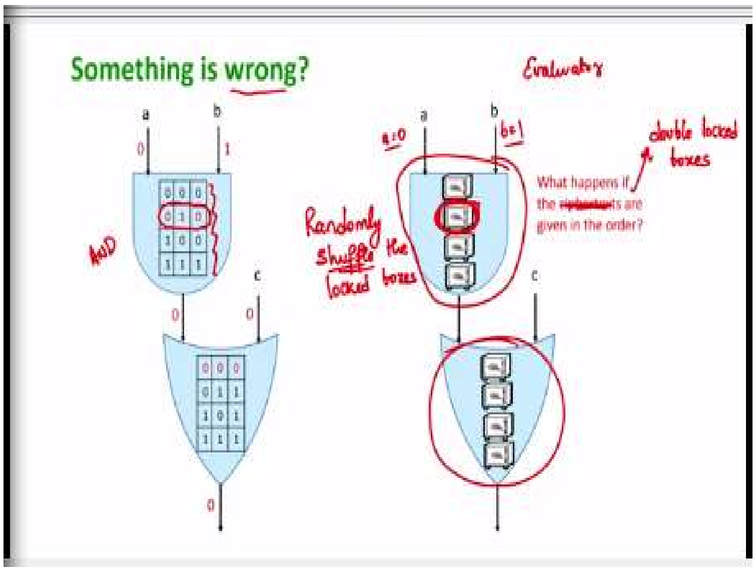
So, we had seen roughly the process of garbling the circuit, garbling the input and decoding the output. And how exactly the circuit is evaluated over the garbled inputs? But it looks like that something is still wrong here, will the evaluator learn the values of a, b and c even if it is given only one of the keys corresponding to the a wire, b wire and c wire and it evaluates the circuit? So, the problem here is that the double locked boxes are given in the same order as corresponding to the appropriate truth tables.

If I consider the AND gate here, we know that the 1st row is for 0 0, 2nd row is for 0 1, 3rd row is for 1 0 and last row is 1 1. And the 4 double locked boxes for this garbled AND gates are also kept in the same order. That means, if say for instance, the evaluator obtains a key, say the key corresponding to a = 0, a key corresponding to b = 1. And when it tries to open one of these 4 boxes and sees that it is the 2nd box which is opening.

Then even though he does not know the labels of the a key and b key, the very fact that it has opened the 2nd box out of these 4 boxes, means that it has opened a double locked box corresponding to the 2nd row of the actual truth table of the AND function. And that itself is sufficient to reveal the values of a and b to the evaluator. However, there is a very simple way to fix this problem.

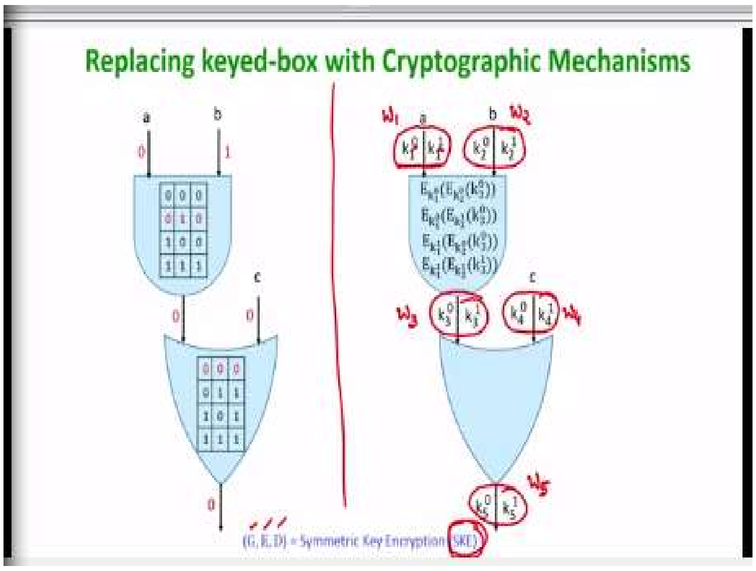
The way we can fix this problem is that once the garbler has prepared the double locked boxes for each of the gate, it has to just randomly shuffle the locked boxes. That means if I consider say for instance, this AND gate, it is no longer the case that the first double locked box corresponds to the 1st row of the truth table or the last double locked box corresponds to the last row of the truth table. Each of the locked box could corresponds to any of the 4 entries of the truth table. The shuffling is decided randomly by the garbler and it is kept private. And for each of the gate a random shuffling is picked.

(Refer Slide Time: 34:19)


Something is wrong?
Evaluator
a
b
0
1
AND
c
0
0
a=0
b=1
double locked boxes
What happens if the ~~[illegible]~~ts are given in the order?
Randomly shuffle the locked boxes

That means now when the evaluator opens, say for instance the 2nd locked box here using the a key and b key, the labels of the a key and the b key are not known to the evaluator. And suppose using those 2 keys, it is able to open the 2nd locked box. Now it cannot tell with guarantee that definitely it has opened the box corresponding to the 2nd row of the truth table of the AND function. Because it could be the case that actually the box that it has opened corresponds to say the last entry of the truth table of the AND function, so that takes care of this problem.

(Refer Slide Time: 34:55)


Replacing keyed-box with Cryptographic Mechanisms
a
b
0
1
c
0
0
0
w_1
w_2
w_3
w_4
w_5
k_1^0 k_1^1
k_2^0 k_2^1
$E_{k_1^0}(E_{k_2^0}(k_3^0))$
$E_{k_1^0}(E_{k_2^1}(k_3^0))$
$E_{k_1^1}(E_{k_2^0}(k_3^0))$
$E_{k_1^1}(E_{k_2^1}(k_3^1))$
k_3^0 k_3^1
k_4^0 k_4^1
k_5^0 k_5^1
(G, E, D) = Symmetric Key Encryption (SKE)

Now we have to think about how exactly we instantiate the physical keys and the physical locked boxes with some cryptographic mechanisms. And a simple way to instantiate the locked boxes is using a symmetric key encryption. Why symmetric key encryption? Because if you see closely the way I have explained the Yao's garbling mechanism, the same keys which garbler's have used for preparing the locked boxes, a subset of those keys are used by the evaluator to open the locked boxes.

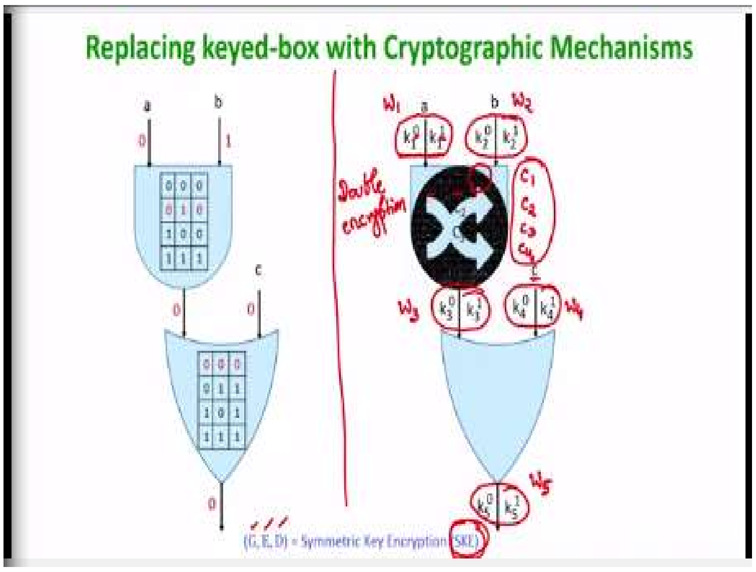
So, that is why the same keys have to be used for the opening as well as closing the boxes. So, that is why to instantiate such locked boxes, we have to use symmetric encryption where the same key is used both for encryption and decryption. So, imagine we are given a secure symmetric encryption scheme which has it is own key generation algorithm, encryption algorithm and decryption algorithm.

Then garbling of the wires means assigning 2 random keys encryption keys 1 corresponding to a = 0 and 1 corresponding to a = 1. So, let us try to understand the semantics of this notation here, it looks quite ugly because you have a subscript, you have a superscript and so on. So, what we have done here is the following. We can imagine that we gave a label to each of the wires of the circuit, so this is wire number 1, this is wire number 2, this is wire number 3, this is wire number 4, this is wire number 5.

So, the subscripts here under with the keys correspond to the wire label. So, the 1st key we have 2 keys assigned $k^1{}_0$, $k^1{}_1$. And in the superscript we denote whether the key is for the wire value bin 0 or the wire value bin 1, so the wire value is a, a can take the value 0 or a can take the value 1. So, depending upon whether it corresponds to 0 or 1 we use the corresponding superscript here.

So, these are the 2 keys assigned to the a wire or the wire number w 1, they are encryption keys for this symmetric encryption scheme. In the same way 2 random encryption keys are assigned to the 2nd wire, 2 random encryption keys are assigned with the 3rd wire, 2 random encryption keys are assigned with the 4th wire, 2 random encryption keys are assigned with a 5th wire.

(Refer Slide Time: 37:47)

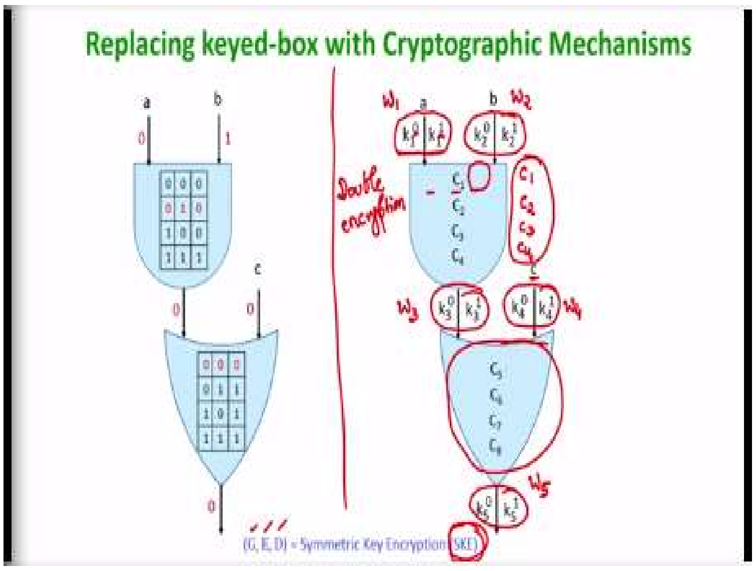


Replacing keyed-box with Cryptographic Mechanisms
a
b
c
Double encryption
W_1
W_2
W_3
W_4
W_5
c_1
c_2
c_3
c_4
(G, E, D) = Symmetric Key Encryption (SKE)

And the garbling of the AND gate means double encryption. So, the double locked box is nothing but encrypting something twice using 2 independent keys. So, what we are encrypting here? Again, if you see here the 1st double locked box encrypts the key corresponding to wire 3 assigned the value 0 and the wire 3 being assigned a value 0, for that this is the key. So, we are going to encrypt this value using the key combinations, wire 1 being assigned a value 0 and wire 2 being assigned a value 0.

And like that you can interpret the remaining entries. And now these 4 double locked boxes or double encryptions are computed they have to be shuffled. So, remember the locked boxes cannot be sent in the same order 0, 0, 0, 1, 1, 0, 1, 1, so they are randomly shuffled. And now after shuffling we will be having 4 ciphertext c 1, c 2, c 3, c 4 that constitutes the garbling of this AND gate. So, that collection of the ciphertext I call it at c.

(Refer Slide Time: 39:09)



Replacing keyed-box with Cryptographic Mechanisms

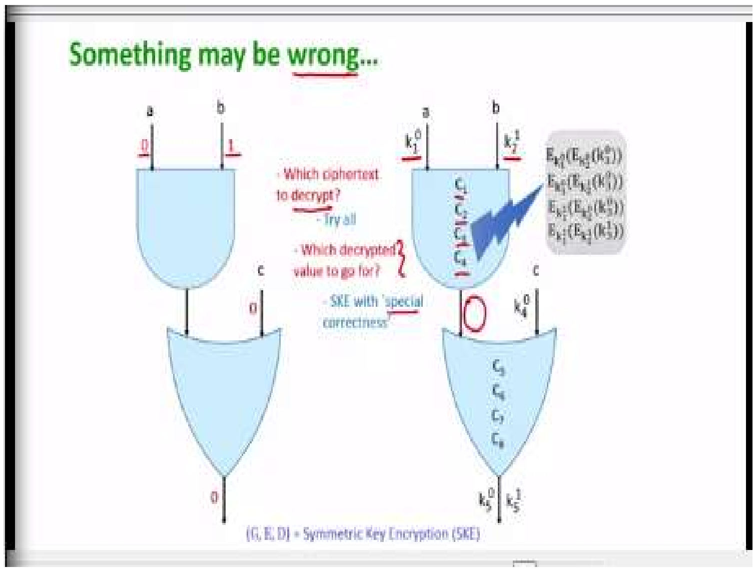
0	0	0
0	1	0
1	0	0
1	1	1

0	0	0
0	1	1
1	0	1
1	1	1

(G, E, D) = Symmetric Key Encryption (SKE)

In the same way, for garbling this OR gate 4 double encryptions have to be computed either encrypting the key k sub 5 0, k sub 5 1 with the key combinations k $_3^0$, k $_3^1$, k $_4^0$, k4^1 and then shuffling the ciphertext and preparing the garbled.

(Refer Slide Time: 39:30)



Something may be wrong...

a
b
0
1
c
0
0
k_1^0
k_2^1
c_1 c_2 c_3 c_4
$E_{k_1^0}(E_{k_2^0}(k_3^0))$
$E_{k_1^0}(E_{k_2^1}(k_3^0))$
$E_{k_1^1}(E_{k_2^0}(k_3^0))$
$E_{k_1^1}(E_{k_2^1}(k_3^1))$
k_4^0
c_5 c_6 c_7 c_8
k_5^0 k_5^1

- Which ciphertext to decrypt?
  - Try all
- Which decrypted value to go for?
  - SKE with 'special correctness'

(G, E, D) = Symmetric Key Encryption (SKE)

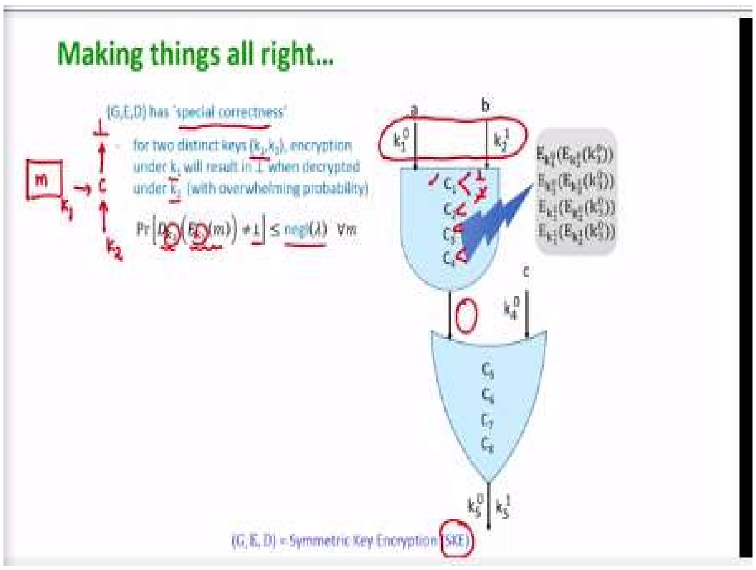
It might look like that we have almost instantiated the Yao’s garbling scheme. That means the double locked box instantiations have now been, we have now instantiated the double locked boxes using symmetric key encryption. But there is still some problem here, the problem is that once the evaluator sees the garbled gate. Say for instance evaluator at the time of evaluation, he will get one of the keys for the 1st wire, one of the keys for the 2nd wire depending upon what exactly are the values of a and b.

So, if a and b are 0 and 1, the evaluator will be seeing the unlabeled key corresponding to a bin 0 unlabeled key corresponding to b = 1. And they are just random bits strings encryption keys for the evaluator. So, it cannot figure out whether it is seeing the key for a = 0, b = 1 or a = 0, b = 0 and so on. And now it has to open one of these 4 locked boxes and these 4 locked boxes basically corresponds to the encryption of the 3rd wire key according to various combinations.

Now, the evaluator has to find out which ciphertext to decrypt because opening the double locked box means decrypting the ciphertext and then trying to find out what should be the key assigned over this wire during the evaluation. So, in the context of physical locked boxes, we assumed there that until and unless you do not have the right key combination, you cannot open the appropriate box.

But now we have replaced the locked boxes using encryption mechanisms and the opening has to be instantiated using decryption mechanism. And the ciphertexts are bit strings, so if we apply the decryption function d on these 4 ciphertexts using the keys k_1^0, k_2^1, we will obtain some value. How we will figure out which value to take and proceed for the next gate from the viewpoint of the evaluator? That is now the question here. And to solve this problem, we will assume that our symmetric key encryption has some special correctness property.

(Refer Slide Time: 41:56)



Making things all right...

(G,E,D) has 'special correctness'

- for two distinct keys (k_1, k_2), encryption under k_1 will result in $\perp$ when decrypted under k_2 (with overwhelming probability)

$\Pr[D_{k_2}(E_{k_1}(m)) \neq \perp] \leq \mathsf{negl}(\lambda) \;\; \forall m$

(G, E, D) = Symmetric Key Encryption (SKE)

m
k_1
c
$\perp$
k_2
a
b
k_1^0
k_2^1
c_1
c_2
c_3
c_4
$E_{k_1^0}(E_{k_2^0}(k_3^0))$
$E_{k_1^0}(E_{k_2^1}(k_3^0))$
$E_{k_1^1}(E_{k_2^0}(k_3^0))$
$E_{k_1^1}(E_{k_2^1}(k_3^0))$
c
0
k_4^0
c_5
c_6
c_7
c_8
k_5^0
k_5^1

What exactly that special correctness property mean? The special correctness means that if you take 2 distinct encryption keys k $_1$ and k $_2$ and a ciphertext which is encrypted using k $_1$. Now if you try to decrypt a ciphertext which is encrypted using k $_1$ using a different key k $_2$ then with very high probability you should obtain a garbage output. And that garbage output should be an indication for you that ok this is not the right ciphertext which you are suppose to decrypt.

That means if you have prepared if you have encrypted any message m using say k $_1$ and say this keeps you the ciphertext c. But you try to decrypt it using k $_2$, you will not get any valid message from the message space, you will get output part with very high probability, that is what is the meaning of this special correctness. Formally this means the following, for any pair of key is k $_1$, k $_2$ which are generated by the key generation algorithm.

And encryption of any message under the key k $_1$ when being decrypted using the key k $_2$ should produce an output part with very high probability. That means it is only a negligible it is possible with a negligible probability that even though you have encrypted something using k $_1$ and later try to decrypt it using k $_2$. You obtain a valid message from the message space with very high probability, you will obtain a bought output.

Or in terms of physical locked example, if you have locked the box using k $_1$ but you do not have k $_1$ but rather you have k $_2$. And you are now trying to opening the lock using k $_2$ you should fill, that is what is the interpretation of this special correctness property. And it turns out that we can very easily prepare we can construct symmetric key encryption scheme using with special correctness property using pseudorandom functions and so on.

So, achieving special correctness from symmetric key encryption is not a very challenging task. So, assuming we have this special correctness property, the evaluator has to do the following, it will take the pair of keys try to decrypt c_1. And c is whether it gives a bought output or some meaningful thing. Do the same thing with c_2, do the same thing with c_3 and do the same thing for c_4.

If the special correctness property is there, it is only one of the 4 ciphertext which will give a non bought output and that non bought output has to be assigned as the key over the output of this AND gate and then it has to proceed. That is how we can now instantiate the Yao's Garbling scheme which we have explained in terms of the physical locked boxes and physical case, thank you.