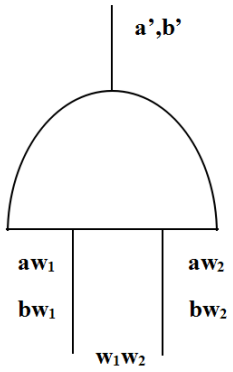
CS 600.442 – Modern Cryptography November 12, 2015

Lecture 18: Secure Computation - III (Oblivious Transfer and Malicious Security

Instructor: Abhishek Jain *Scribe: Sen Li*

1 Review(GMW Protocol)

First, we want to finish the GMW Protocol and Oblivious Transfer, and here we have a AND gate. We want to find that s.t. a'⊕b'=$(a_{w1} \oplus b_{w1}) \wedge (a_{w2} \oplus b_{w2})$.

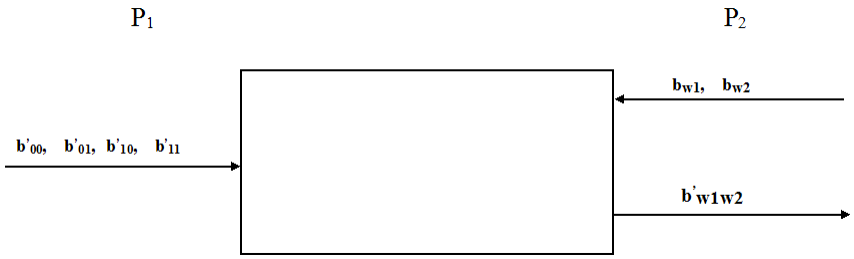


The two(2) parties P_1 and P_2, P_1 had the shares and P_2 had the matrix. The question is how can they computing the function together under this property.

Here we also give the table of the four(4) possible values of b_{w1} and b_{w2}, then give a random string α to a' ,then we will get four(4) possible values of b'

b_{w1}	b_{w1}	a'	b'
0	0	α	$\alpha \oplus (a_{w1} \oplus 0) \wedge (a_{w2} \oplus 0)$
0	1	α	$\alpha \oplus (a_{w1} \oplus 0) \wedge (a_{w2} \oplus 1)$
1	0	α	$\alpha \oplus (a_{w1} \oplus 1) \wedge (a_{w2} \oplus 0)$
1	1	α	$\alpha \oplus (a_{w1} \oplus 1) \wedge (a_{w2} \oplus 1)$

In this situation, we know the correct value of b' is one of these, but we do not know which one, and we want that P_1 should somehow transmit one of these values to P_2. P_2 should not allow any other possible values.P_1 had b_{w1} and b_{w2} and P_2 had the four(4) possible values, they will run the protocol in some kind of box. Then we write down there things.

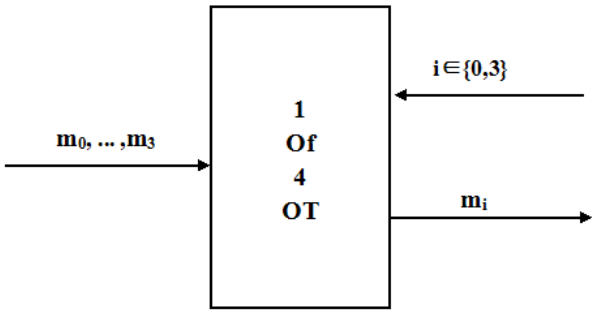


What we want about this transfer is that:

1. P_1 should not learn $(b_{w1}\ ,\ b_{w2}) \in \{0.3\}$
2. P_2 should not learn b'_{xy} where $(x,y) \neq (b_{w1}\ ,\ b_{w2}) \in \{0,3\}$

2 Question(Oblivious Transfer):

Here we have a question that how to build 1 out of 4 OT from 1 out of 2 OT. And the 1 out of 4 OT should be like this:



Well, the solution should be like this:

1.S chooses $S_i \leftarrow 0,1$ $i \in \{0,...,5\}$

2.S computes:

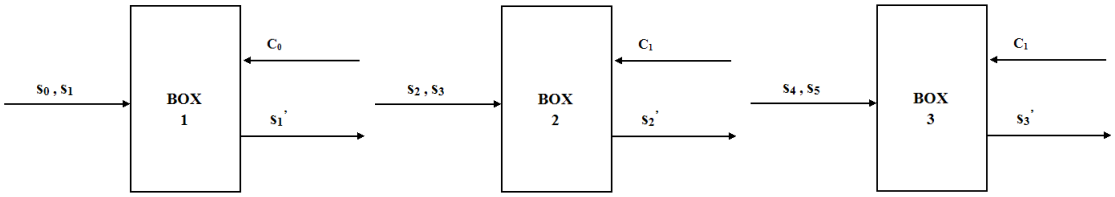
$\alpha_0 = s_0 \oplus s_2 \oplus m_0$

$\alpha_1 = s_0 \oplus s_3 \oplus m_1$

$\alpha_2 = s_1 \oplus s_4 \oplus m_2$

$\alpha_3 = s_1 \oplus s_5 \oplus m_3$

The idea is that we use two(2) masks, because one(1) mask is not enough for the security, the mask is just like the lock. Here we got three(3) boxes like these:



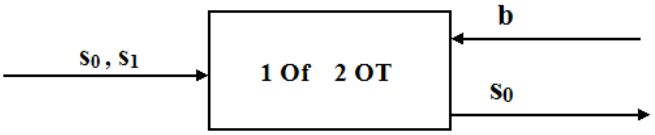
We suppose that
i=c_0c_1, c_0=0, c_1=1

Then we want the Recv algorithm should be:
1.Use c_0=0 as input in box 1 and out=s_0
2.Use c_1=1 as input in box 2 and box 3 and box 2 out=s_3, box 3 out =s_5

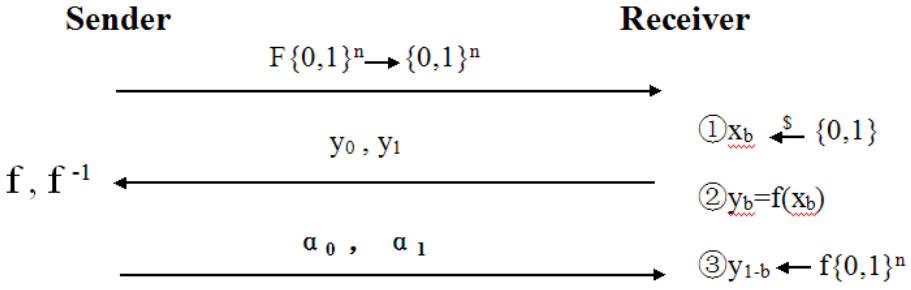
Then we will got that Out=$\alpha_1 \oplus s_0 \oplus s_3$=$m_1$

3 Security against malicious Adv:

Firstly, given a simple protocol like this:



We want to decrypt these values, but it only decrypt one of them and we do not know which one to decrypt. So here is the process.



x_b is chosen from domain of f.

Then we compute that:

1.$x_0^{'}=f^{-1}(y_0)$, $x_1^{'}=f^{-1}(y_1)$

2.α_0=h($x_0^{'}$)$\oplus s_0$, similarly, α_1=h($x_1^{'}$)$\oplus s_1$, notice that the h(•) is hardcore prediction proof.

Then we predict that α_b=h($x_b^{'}$)$\oplus s_b$

We assume that $x_b^{'}=x_b$, therefore:

1.Compute h(x_b)

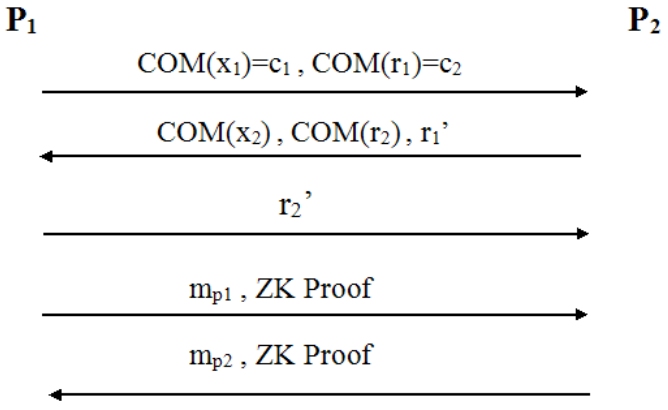
2.s_b=h(x_b)$\oplus \alpha_b$

We could also know when $y_{1-b}^{'}$, the α_{1-b} will be h($x_{1-b}^{'}$)$\oplus s_{1-b}$

Then, the end of the protocol.

4 Semihonest to Malicious Adv:

Suppose the given protocol π^{SH}=(P_1^{SH}, P_2^{SH}), the basic idea that we want is that along with every message under this protocol, we want the parties to prove that this is a honest protocol. Here, following the instruments:



P_1^{SH}=(x_1,r_1,m_{p1})=m_{p1},st, then ∃ s, s' , x_1, r_1,st, such that

1.c_1=COM(x_1,s)

2.c_2=COM(r_1,s)

And m_{p1}=P_1^{SH}(x_1, r_1,m_{p1}), and r_1^*=$r_1 \oplus r_1^{'}$

And we also need ZK proof, if ZK proof is invalid at any step, then abort.