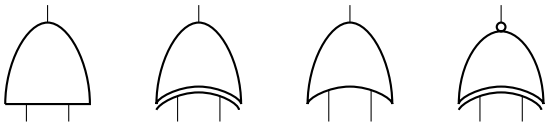
CS 600.442 – Modern Cryptography October 15, 2015

Lecture 14: Secure Computation - II Garbled Circuits

Instructor: Abhishek Jain *Scribe: Yuanqi Zhu*

1 Garble circuit

One of the two constructions of secure two-party protocol is Garbled Circuit. Here are some simple circuits:



And Gate, XOR Gate, OR Gate, XNOR Gate

While the whole Garble circuit should be a set of polynomial time circuits. For each single gate, there also should be some table to identify its values of input and output. Take AND Gate as example:

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

Our method to construct a secure protocol here should be divided to two steps:

1. Define and construct Garble Circuits (polynomial time circuits)
2. use it to produce secure two party computation from Garbled Circuits

1.1 Goal

Goal here is to construct a "Garble Circuit".

Take Circuit 'C' and any input 'x', then garble these two value and transfer these two to garble version (C_{garble}, x_{garble}) such that, we can do computation and finally get C(x), which should only reveal C(x).

1.2 Syntax

Here look at two algorithm:

(1). Garbling Algorithm:

Garble() take the value of Circuits, and output their garble version. $Garble(C)$ outputs (C_{garble}, x_{garble})

(2). Evaluation Algorithm:

Evaluation take these value from G(), and output.
$Eval(C_{garble}, xgarble)$ outputs a value z

1.3 correctness

Here we define the correctness:
The output of Evaluation should be the actual $C(x)$, and (C_{garble}, x_{garble}) should be $G(C, x)$, what we want to be.

For every (C, x),
$Pr[C(x) = Eval(C_{garble}, x_{garble})|(C_{garble}, x_{garble}) = Garble(C, x) = 1 - negligible(n)]$

1.4 Security

We require that the party receiving the labels and outputs of Garble version should no distinguish the result from simulator who have no access to x, and the evaluation of real Garble circuits.

There exists some PPT simulator S, such that
For every (C, x), $(C_{garble}, x_{garble}) \sim S(1^n, C, C(x))$ where $(C_{garble}, x_{garble}) = Garble(C, x)$

This means if there is simulator with input C and $C(x)$,$S(C, C(x))$ output $(C'_{garble}, x'_{garble})$
And Garble(C,x) output (C_{garble}, x_{garble}),
then $(C'_{garble}, x'_{garble})$ and (C_{garble}, x_{garble}) should be indistinguishable. Here the simulator didn't see x, if this hold, then the input could be hidden.

Hiding C: To make sure Circuits cannot be modified, here we can also use universal circuits and pass in C as input to hide C.

1.5 Construction

1. $Garble(C, x)$

Here we start to construct the Garble circuit.

pick random labels for every wire in the circuit Think random labels as SK to Secret key encryption SK encryption choosing two key for every wire in the circuit for the first wire we choose a_0,a_1

Here we define a more general steps for Garble(C, x). Let w denote the set of Wires,G to be set of gates

step 1. $\forall$ wire $w \in W$, choose labels,S_w^0, S_w^1, (each wire carry one bit value, whether 0 or 1. However, S could be a n bit string)

now we need to implement the encryption, if given two labels, we should compute it's encryption output:

$\forall$ gate g, Given $(S_{w_1}^{b_1} in, S_{w_2}^{b_2} in)$, should be able to compute, $S_w^{g(b_1,b_2)} out$

step 2. $\forall$ gate g, $E_{S_{w_1}^{b_1} in, S_{w_2}^{b_2} in}(S_w^{g(b_1,b_2)} out)$, $b_1, b_2 \in 0, 1$

Thus we implement encryption to all gates. The implementing way could be double encryption as follows:

$E_{s_1,s_2}(s) = Enc_{s_1}(Enc_{s_2}(s))$

Example:

If input x=1001, then we choose A_0 for 1, $B_1$for 0, and C_0,D_1. However, others have no idea what these labels' content are. They are just random. Then, according to the encrypt table, we can get E_0 For this particular circuit we get two wire keys. 4 Ciphertext. find and decrypt each of them and only one of decryption will succeed.

2. $Evaluation(C_{garble}, x_{garble})$

for evaluation, we should satisfy two property:

(1). Only one ciphertext per gate is decryptable

(2). Result of decryption should be equal to value on outgoing wire

then we are left with two problems:

(1). how to interpret output?

The answer is providing the mapping of value to evaluator. Create table such as:

$$\begin{bmatrix} I_0 & 0 \\ I_1 & 1 \end{bmatrix}$$

In this way, the matrix map I_0 to value o, I_1 to value 1, and provide this to garble circuit. At this end of calculation, we can check the output with the matrix.

(2). How to determine which of the 4 entries to decrypt in each encrypted gate table?

For the garble circuit, we don't want others know which one should be decrypted. Otherwise they can know about the labels. Thus, we let them try each of encryptions one by one with the same key and only one of them can succeed, which is the right decryption. To indicate the correct decryption, we add some sign to the encryption, which is a string of 0:

$E_{s_1,s_2}(s) = Enc_{s_1}(Enc_{s_2}(s||0^n))$

In this way, only decryption follows a string of 0 is the correct decryption.

Privacy(intuition)

(1). for each wire (including input wires), adversary only sees one label W_b

(2). The 4 entries in each encrypted table are in random order

(3). adversary tries to decrypt each entry. Only one decryption succeeds

(4). Adversary as no idea whether b=0 or b=1 for any label W_b

Interpreting the output

For every output wire, reveal the mappings(b, W_b)

High level intuition for the proof(security):

The Garble circuit maintain this property in very end that to all the computation, every computation adversary only sees one of the labels, whether wire corresponding to 0 or 1. They only know that one of the decryption is right. The only wire he know is at the output.

Formal proof is from Lindell-Pinkas 05/06

2 Secure computation from Garbled Circuits

Here are Alice and Bob who would like to communicate with each other. We take Alice as the side of x,and Bob as the side of y. Alice is garble circuit generator, as she know they want to compute the $f(x)$, she will first write down the $f()$ as boolean circuits, and then apply garbling circuits on them to produce $f_{garble}()$. And she also knows her own input x, and she can produce x_{garble} independently. According to the image, the input is divided into two parts. Input wires x corresponding to Alice and y to Bob.

Now Alice know about part of the input, however, she still need y to compute. However, if Bob give y directly to Alice, then Alice may know about y. As Alice generated the garble circuit, she compute the output independently. On the other hand, Alice cannot reveal both x_{garble}, and f_{garble} to Bob, otherwise Bob can evaluate garble circuit multiple times, and get the $f(x_{garble}, y_{garble})$ and even $f(x_{garble}, y'_{garble})$.

Here is the problem: How to transmit y_{garble}?

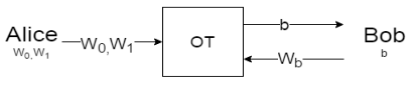
Imaging Alice and Bob have magic box. It takes all labels of second input from x and y from Bob, and finally output y_{garble}. It will choose appropriate wires to compute.

What we want here is that:

(1) Alice learns nothing about y

(2) Bob does not learn the other labels

This is Olivious Transfer. Here is a one wire sample:



we want that: (1) Alice does not learn b (2) Bob does not learn W_{1-b}

Alice is obliviously transfer W_0,W_1 without knowledge of b. She is oblivious of b. This definition is called One-out-of-two Oblivious Transfer by Even-Goldreich-Lempel.