University of Illinois, Urbana Champaign
CS/ECE 498AC3/4 Applied Cryptography

Instructor: Dakshita Khurana
Scribe: Sourav Das
Date: November 12, 2020

LECTURE
23

On the Security of Garbled Circuits

In the last lecture, we have studied how to compute any general function on private inputs of two mutually untrusted parties using *garbled circuits* (GC) and oblivious transfer (OT). Here on, we will refer to the protocol based on GC and OT, we learned in the last lecture with Π. Recall from the last lecture that one major advantage of Π is that the round complexity of Π is identical to the round-complexity of underlying OT protocol. Hence, using a two-round OT, Π gives us a secure two-party computation protocol for general circuits with only two-rounds of communications. Additionally, Π can be used to build a 2PC protocol where the function itself is part of one of the participants' private input.

In this lecture, we will formally define the notion of security of Π along with circuit privacy and prove it under standard cryptographic assumptions. We will also argue that in Π, the security against malicious garbler/sender will follow directly from the security of the underlying OT. Thus we will spend most of the lecture proving the security of Π protocol against malicious evaluator/receiver. Intuitively, in Π, only information the garbler receives is the OT protocol's output. On the contrary, the evaluator receives the entire garbled circuit along with keys corresponding to the senders' input.

The outline of this lecture is as follows; we will first quickly review the inner working of Π and set up some notations. Then we will model the security of the protocol and formally prove its security under the standard cryptographic assumptions. Briefly, we define simulation-based security for Π and illustrate that if secure OT and multi-message Chosen Plaintext Attack (CPA)-secure Encryption exists then the Π is secure.

23.1 Background on Garbled Circuits

Garbled circuits and Oblivious Transfer enables us to run any secure two-party computation (2PC) between two mutually untrusted parties, namely a *garbler* and a *evaluator*. Sometimes, the garbler and the evaluator are also referred to as the sender and the receiver, respectively, for reasons which will be apparent later.

Let f be the two-party boolean functionality the sender and the receiver want to evaluate securely. Let C be the boolean circuit consisting of AND and XOR gates that implements f. By implementing f we mean that for all sender's input $\mathbf{x}$ and for all receiver's input $\mathbf{y}$,

$f(\mathbf{x}, \mathbf{y}) = C(\mathbf{x}, \mathbf{y})$. Also, let's assume that only the sender knows the description of f and hence C, i.e., we will consider the circuit C as a private input of the sender.

Without loss of generality, let's assume that C is a layered circuit. Then, in $\Pi(C, x, y)$, the garbler first garbles the circuit C by garbling each of its gates layer by layer. The garbler then sends the garbled table of each of the gate to the evaluator. Additionally, the garbler sends the keys corresponding to inputs of the garbler for the input layer. The evaluator and garbler then engage in multiple oblivious transfers (one for each input bit of the evaluator). The evaluator then evaluates the garbled circuit using the keys received directly from the garbler and the keys received using the oblivious transfers. The garbled circuit's output is a uniform random bit string associated with either 0 or 1.

23.2 Garbled Circuit Security

Informally, the security of the Π against a malicious evaluator is modelled as the existence of a probabilistic polynomial time (PPT) simulator, $\mathcal{S}$, which on input $\mathbf{y}$ and $f(\mathbf{x}, \mathbf{y}) = C(\mathbf{x}, \mathbf{y})$ simulates the view of the adversarial evaluator $\mathcal{A}$, where for all PPT distinguishers $\mathcal{D}$, the simulated view of $\mathcal{A}$ is computationally indistinguishable from the view of $\mathcal{A}$ in the actual protocol.

Formally, let $\mathbf{x}, \mathbf{y} \in \{0,1\}^m$ be the input of the garbler and the evaluator, respectively. Additionally, the garbler has the additional private input C, which is the circuit implementing a boolean function $f : \{0,1\}^m \times \{0,1\}^m \to \{0,1\}$, i.e., $\forall \mathbf{x}, \mathbf{y} \in \{0,1\}^m, C(x, y) = f(x, y)$. Then,

Definition 23.1. For any given function $f : \{0,1\}^m \times \{0,1\}^m \to \{0,1\}$ implemented by a boolean circuit C, the protocol Π is secure if $\forall \mathbf{x}, \mathbf{y} \in \{0,1\}^m, \forall$ PPT $\mathcal{A}$, $\exists$ a PPT simulator, $\mathcal{S}$, such that $\forall$ polynomial time distinguisher, $\mathcal{D}$, that the following inequality holds:

$$|\Pr[\mathcal{D}(1^n, \Pi(C, \mathbf{x}, \mathbf{y}), \mathbf{y}, \mathcal{A}) = 1] - \Pr[\mathcal{D}(1^n, \mathcal{S}(C(\mathbf{x}, \mathbf{y}), \mathbf{y}), \mathcal{A}) = 1]| < \epsilon(\lambda) \tag{23.1}$$

where λ is the security parameter and $\epsilon(\lambda)$ is a negligible function in λ.

Recall from the last lecture that, to simulate the view of an $\mathcal{A}$, the simulator $\mathcal{S}$ needs to generate the garbled tables for each gate and send these garbeld gates to the evaluator. Additionally, the simulator also needs to simulate the view of the underlying OTs and finally compute a mapping between the output of the garbled circuit and the output of f. Intuitively, the main idea behind the proof is as follows. First, the $\mathcal{S}$ internally invokes the OT simulator $\mathcal{S}_{OT}$ to extract the inputs of the adversarial evaluator. Recall, by the security of OT, such a simulator exists and runs in probabilistic polynomial time. Next, we use the fact that in the real execution of Π, for each garbled gate, the evaluator receives keys to successfully decrypt only one of the rows of the garbled circuit. Hence, intuitively, the content of the other rows for each gate remain hidden from the evaluator.

We next formalize on the above intuition and describe the simulation strategy. For simplicity, we will first describe the simulation strategy for single garbled gate, i.e., a garbled circuit with a single boolean gate with two inputs. Then using a sequence of hybrids, we will illustrate that for all efficient distinguishers, the view of the adversary in the simulated world is computationally indistinguishable from the view of the adversary in the real protocol execution. Lastly, recall from last class, one could build a protocol Π with an arbitrary polynomial number of gates from a protocol for a single garbled gate. We argue that using

the same approach, the security of Π for a single gate can be extended to the security of Π with an arbitrary polynomial number of gates.

Let $G : \{0,1\} \times \{0,1\} \to \{0,1\}$ be the gate the two-party Alice and Bob wants to evaluate in a secure fashion. Without loss of generality, let G be a AND gate and let $x, y \in \{0,1\}$ be the inputs to G, where x and y are the inputs of Alice and Bob, respectively. The same arguments will also hold when G is an XOR gate. Then, in $\Pi(G, x, y)$, the garbled table created by Alice the garbler has the following structure where $k_{a,x}$ and $k_{b,y}$ corresponds to a uniformly randomly chosen key for the input x, y respectively. Also, Enc is a multi-message CPA-secure encryption scheme.

$\mathsf{Enc}_{k_{a,0}}(\mathsf{Enc}_{k_{b,0}}(k_{c,0}))$
$\mathsf{Enc}_{k_{a,0}}(\mathsf{Enc}_{k_{b,1}}(k_{c,0}))$
$\mathsf{Enc}_{k_{a,1}}(\mathsf{Enc}_{k_{b,0}}(k_{c,0}))$
$\mathsf{Enc}_{k_{a,1}}(\mathsf{Enc}_{k_{b,1}}(k_{c,1}))$

TABLE 23.1: Garbled table for gate G in $\Pi(x, y)$

In addition to the garbled table shown in Table 23.1, in Π, the garbler sends $k_{a,x}$, the key corresponding to the input of the garbler and a output map $\phi = \{k_{c,0} : 0; k_{c,1} : 1\}$ which enable the evaluator to map the output of the garbled circuit to output of the original function f.

According to definition 23.1, $\Pi(G, x, y)$ securely implements the functionality f, if there exists a PPT simulator $\mathcal{S}$, which on the input $y, z = C(x, y)$ produces a garbled table, the keys, and an output map such that the view of the $\mathcal{A}$ in the simulated world is indistinguishable from the view of the $\mathcal{A}$ in the real execution of Π.

As we described earlier, the intuition behind the simulation strategy is that the garbled gate contains exactly four rows, and the evaluator could decrypt only one of the rows. Thus, from the evaluator's perspective, the garbled table looks similar even if the other rows are encryptions of all zero string. The simulator exploits this in the following manner. The simulator will first choose a row at random and key k_z uniformly at random, double encrypt k_z in the chosen row, and double encrypt all zero strings in every row. The simulator then send one of the keys corresponding to row encrypting k_z directly and the other using the simulator of the oblivious transfer. Lastly, the simulator computes the output map as $\phi' = \{k_z : z, k_r : 1 - z\}$, where k_r is a key chosen uniformly at random from the key space. More formally, the $\mathcal{S}$ works as follows:

$\mathcal{S}_\Pi$**:**

1. Sample $\{k_1, k_2, k_3, k_4, k_5\}$ using the key generation algorithm of Enc
2. Without loss of generality, set k_1 as the key corresponding to the input of the garbler, k_2 as the input to the $\mathcal{S}_{OT}$, and $k_z = k_5$.
3. Invoke $\mathcal{S}_{OT}$ using k_2 to extract the input of the evaluator. Let y be the input of the evaluator.
4. Forward y to the ideal functionality to get $z = C(x, y)$ for some circuit C and some input x.
5. Compute the garbled gate as below:

$\mathsf{Enc}_{k_1}(\mathsf{Enc}_{k_3}(k_z))$
$\mathsf{Enc}_{k_1}(\mathsf{Enc}_{k_4}(00...0))$
$\mathsf{Enc}_{k_2}(\mathsf{Enc}_{k_3}(00...0))$
$\mathsf{Enc}_{k_2}(\mathsf{Enc}_{k_4}(00...0))$

TABLE 23.2: Simulated garbled table

6. Randomly permute the garbled table.
7. Set $\phi' = \{k_z : z; k_r : 1 - r\}$, where k_r is key sampled according to key generation algorithm of Enc
8. Send Table 23.2, k_1, and ϕ' to $\mathcal{A}$.

THEOREM 23.2. *Let G be a binary gate, then $\forall x, y \in \{0,1\}$, $\forall$ PPT adversary $\mathcal{A}$, $\forall$ PPT distinguishers $\mathcal{D}$, the distribution generated by $\mathcal{S}_\Pi$ is computationally indistinguishable the the view of $\mathcal{A}$ in the real protocol $\Pi(G, x, y)$, i.e.,*

$$|\Pr[\mathcal{D}(1^n, \Pi(G, x, y), y, \mathcal{A}) = 1] - \Pr[\mathcal{D}(1^n, \mathcal{S}_\Pi(G(x, y), y), \mathcal{A}) = 1]| < \epsilon(\lambda). \tag{23.2}$$

This implies that $\Pi(G, x, y)$ securely implements $G(x, y)$.

We will prove Theorem 23.2 using a sequence of Hybrid games $\mathbf{Game}_i$, for $i = 1, 2, 3, 4$.

$\mathbf{Game}_1$: The $\mathbf{Game}_1$ is identical to the execution of $\Pi(G, x, y)$, except OT step for the keys corresponding to gate G is replaced by the $\mathcal{S}_{OT}$ with y as the input of the OT receiver and the $k_{b,y}$ as the output of the OT ideal functionality.

Informally, we next argue that assuming the underlying OT protocol is secure against all PPT adversaries, then the view of $\mathcal{A}$ in $\mathbf{Game}_1$ is computationally indistinguishable from the view of $\mathcal{A}$ in $\Pi(G, x, y)$.

CLAIM 23.3. *If the underlying OT protocol in Π is secure, i.e., $\mathcal{S}_{OT}$ exists, then $\forall x, y \in \{0,1\}, \forall$ PPT adversary $\mathcal{A}$, $\forall$ PPT distinguisher $\mathcal{D}$, the following holds, where $\epsilon_1(\lambda)$ is a negligible function of in the security parameter λ, i.e.,*

$$|\Pr[\mathcal{D}(1^n, \Pi(G, x, y), \mathcal{A}) = 1] - \Pr[\mathcal{D}(1^n, \mathbf{Game}_1(G, x, y), \mathcal{A}) = 1]| < \epsilon_1(\lambda) \tag{23.3}$$

The proof of claim 23.3 follows directly from the security of $\mathcal{S}_{OT}$.[1]

$\mathbf{Game}_2$: The $\mathbf{Game}_2$ identical to $\mathbf{Game}_1$ except that the garbled table has the structure shown in Table 23.3.

We will next argue that if the underlying encryption scheme Enc is multi-message CPA secure, then for all PPT adversaries, the view of the adversary in $\mathbf{Game}_1$ and $\mathbf{Game}_2$ are computationally indistinguishable. We will prove this by showing a reduction that if the views in $\mathbf{Game}_1$ and $\mathbf{Game}_2$ are not computationally indistinguishable, then using the corresponding distinguisher we can build an adversary $\mathcal{A}_{\mathsf{Enc}}$ that breaks the CPA security of Enc.

[1] The formal proof for Claim 23.3 will involve a building a distinguisher $\mathcal{D}_{OT}$, that distinguish the distribution generated by $\mathcal{S}_{OT}$ from the actual OT protocol using the distinguisher of the view of $\Pi(x, y)$ and $\mathbf{Game}_1$.

$\mathsf{Enc}_{k_{a,0}}(\mathsf{Enc}_{k_{b,0}}(k_{c,0}))$
$\mathsf{Enc}_{k_{a,0}}(\mathsf{Enc}_{k_{b,1}}(00...0))$
$\mathsf{Enc}_{k_{a,1}}(\mathsf{Enc}_{k_{b,0}}(k_{c,0}))$
$\mathsf{Enc}_{k_{a,1}}(\mathsf{Enc}_{k_{b,1}}(k_{c,1}))$

TABLE 23.3: Garbled table for gate G in $\mathbf{Game}_2$

CLAIM 23.4. *If* Enc *is a multi-message CPA-secure encryption scheme, then* $\forall$ *PPT adversary* $\mathcal{A}$, $\forall$ *PPT distinguisher* $\mathcal{D}$, *the following inequality holds where* $\epsilon_2(\cdot)$ *is a negligible function,*

$$|\Pr[\mathcal{D}(1^n, \mathbf{Game}_1(G, x, y), \mathcal{A}) = 1] - \Pr[\mathcal{D}(1^n, \mathbf{Game}_2(G, x, y), \mathcal{A}) = 1]| < \epsilon_2(\lambda) \quad (23.4)$$

Proof. For the sake of contradiction, assume the existence of a PPT distinguisher $\mathcal{D}^*$ that distinguishes between the view in $\mathbf{Game}_1$ and $\mathbf{Game}_2$. Next, we will use $\mathcal{D}^*$ to build an adversary $\mathcal{A}_{\mathsf{Enc}}$ that breaks the CPA security of Enc, under the which is given as below.

$\mathcal{A}_{\mathsf{Enc}}$:

1. Sample $k_{a,0}, k_{a,1}, k_{b,0}, k_{c,0}, k_{c,1}$ by running the key generation algorithm of Enc.
2. Query the Enc challenger for encryption of $k_{c,1}$. Let $\mathsf{ct}_{c,1} = \mathsf{Enc}_{k^*}(k_{c,1})$ where k^* is the underlying key used by the $\mathsf{Chal}_{\mathsf{Enc}}$, the challenger of Enc.
3. Set the challenge message $\mathsf{m}_0 = k_{c,1}$ and $\mathsf{m}_1 = 00...0$. Let $\mathsf{ct}_b = \mathsf{Enc}_{k^*}(\mathsf{m}_b)$
4. Create the garbled table as:

$\mathsf{Enc}_{k_{a,0}}(\mathsf{Enc}_{k_{b,0}}(k_{c,0}))$
$\mathsf{Enc}_{k_{a,0}}(\mathsf{ct}_b)$
$\mathsf{Enc}_{k_{a,1}}(\mathsf{Enc}_{k_{b,0}}(k_{c,0}))$
$\mathsf{Enc}_{k_{a,1}}(\mathsf{ct}_{c,1})$

TABLE 23.4: Garbled table of $\mathcal{A}_{\mathsf{Enc}}$

5. Invoke $\mathcal{D}^*$ for the garbled table 23.4, $k_{a,0}$, OT input $k_{b,0}$ and the output map $\phi' = \{k_{c,0} : 0, k_{c,1} : 1\}$.
6. Output whatever $\mathcal{D}^*$ outputs.

Observe that, depending upon the choice bit b of the $\mathsf{Chal}_{\mathsf{Enc}}$, the distribution of inputs on which $\mathcal{A}_{\mathsf{Enc}}$ invokes $\mathcal{D}^*$ is either identical to the distribution of inputs in $\mathbf{Game}_{1+b}$. Hence, the probability of $\mathcal{A}_{\mathsf{Enc}}$ breaking the CPA security of Enc is same as the success probability of $\mathcal{D}^*$ distinguishing view of $\mathbf{Game}_1$ and $\mathbf{Game}_2$. Hence, if success probability $\mathcal{D}^*$ is non-negligible so does the success probability of $\mathcal{A}_{\mathsf{Enc}}$, thus leading to a contradiction. □

Next, we will replace the next row of the garbled table of $\mathbf{Game}_2$ with encryption of zeros and argue that the new view is indistinguishable from the view of the adversary in $\mathbf{Game}_2$. Formally,

$\mathsf{Enc}_{k_{a,0}}(\mathsf{Enc}_{k_{b,0}}(k_{c,0}))$
$\mathsf{Enc}_{k_{a,0}}(\mathsf{Enc}_{k_{b,1}}(00...0))$
$\mathsf{Enc}_{k_{a,1}}(\mathsf{Enc}_{k_{b,0}}(00...0))$
$\mathsf{Enc}_{k_{a,1}}(\mathsf{Enc}_{k_{b,1}}(k_{c,1}))$

TABLE 23.5: Garbled table for gate G in $\mathbf{Game}_3$

$\mathbf{Game}_3$: In $\mathbf{Game}_3$, the third row of the garbled table in $\mathbf{Game}_2$ is replaced by the double encryption of all zero strings, i.e.,

CLAIM 23.5. *If* Enc *is a multi-message CPA-secure encryption scheme, then* $\forall$ *PPT adversary* $\mathcal{A}$, $\forall$ *PPT distinguisher* $\mathcal{D}$, *the following inequality holds where* $\epsilon_3(\cdot)$ *is a negligible function,*

$$|\Pr[\mathcal{D}(1^n, \mathbf{Game}_2(G, x, y), \mathcal{A}) = 1] - \Pr[\mathcal{D}(1^n, \mathbf{Game}_3(G, x, y), \mathcal{A}) = 1]| < \epsilon_3(\lambda) \quad (23.5)$$

The proof of claim 23.5 follows a similar structure as in the proof of claim 23.4 where using a distinguisher between $\mathbf{Game}_2$ and $\mathbf{Game}_3$, we can build an adversary that breaks the CPA security of the Enc under the key $k_{a,1}$. Observe that $k_{a,1}$ chosen according to the distribution of the key generation algorithm of Enc. Similarly, we define $\mathbf{Game}_4$ where the last row is replaced by double encryption of all zero string.

$\mathbf{Game}_4$. In $\mathbf{Game}_4$, the fourth and the final row of the garbled table in $\mathbf{Game}_4$ is replaced by the double encryption of all zero strings, i.e.,

$\mathsf{Enc}_{k_{a,0}}(\mathsf{Enc}_{k_{b,0}}(k_{c,0}))$
$\mathsf{Enc}_{k_{a,0}}(\mathsf{Enc}_{k_{b,1}}(00...0))$
$\mathsf{Enc}_{k_{a,1}}(\mathsf{Enc}_{k_{b,0}}(00...0))$
$\mathsf{Enc}_{k_{a,1}}(\mathsf{Enc}_{k_{b,1}}(00...0))$

TABLE 23.6: Garbled table for gate G in $\mathbf{Game}_4$

Similar to the $\mathbf{Game}_2$ and $\mathbf{Game}_3$, we can argue that view of the adversary in $\mathbf{Game}_4$ is computationally indistinguishable from $\mathbf{Game}_3$ under the multi-message CPA security of Enc scheme. We will omit the proof of the following claim as it is very similar to the proof of claim 23.4.

CLAIM 23.6. *If* Enc *is a multi-message CPA-secure encryption scheme, then* $\forall$ *PPT adversary* $\mathcal{A}$, $\forall$ *PPT distinguisher* $\mathcal{D}$, *the following inequality holds where* $\epsilon_4(\cdot)$ *is a negligible function,*

$$|\Pr[\mathcal{D}(1^n, \mathbf{Game}_4(G, x, y), \mathcal{A}) = 1] - \Pr[\mathcal{D}(1^n, \mathbf{Game}_4(G, x, y), \mathcal{A}) = 1]| < \epsilon_4(n) \quad (23.6)$$

Now, observe that the garbled table in $\mathbf{Game}_4$ is identically distributed to the garbled table constructed by the simulator $\mathcal{S}_\Pi$ up to a renaming of the keys. This this implies that view of the $\mathcal{A}$ in $\mathbf{Game}_4$ is identically distributed with the view of $\mathcal{A}$ in the interaction

with simulator $\mathcal{S}$. Hence, using claims 23.4,23.5,23.6, the view of the $\mathcal{A}$ in the interaction with $\mathcal{S}$ is computationally indistinguishable from the view of the adversary $\mathcal{A}$ from the real protocol execution $\Pi(G, x, y)$.

So far, we have only looked at the security of Π only for the case where the circuit G has only one gate. To extend the security of Π to arbitrary polynomial number of gates, we can construct a simulator that simulates the garbled table for each gate in a layer by layer manner, starting with the input layer.

Acknowledgement

These scribe notes were prepared by editing a light modification of the template designed by Alexander Sherstov. retain this acknowledgement in all scribe notes.

References