CS 65500
Advanced Cryptography

Lecture 8: Garbled Circuits - I

Instructor: Aarushi Goel

Spring 2025

Agenda

- Secret-Key Encryption
- Garbled Circuits
- Secure Computation from Garbled Circuits.

Secure Two-Party Computation of *General* Functions

Alice

Bob

Input: Alice: $x_1, \ldots, x_m \in \{0,1\}^m$ Bob: $y_1, \ldots, y_m \in \{0,1\}^m$

Function: $f : \{0,1\}^{2m} \longrightarrow \{0,1\}^{\ell}$

Output: $f(x_1, \ldots, x_m, y_1, \ldots, y_m) = z_1, \ldots, z_\ell$

How? Using the GMW Protocol.

Drawback of the GMW Protocol.

How many rounds of interaction are needed between Alice and Bob in the GMW Protocol?

- Polynomial in the depth of the circuit representing the function.
- This could potentially take a long time to run if the network latency is high.

Can we design a protocol where the number of rounds of interaction are independent of the circuit size?

Can it potentially be constant?

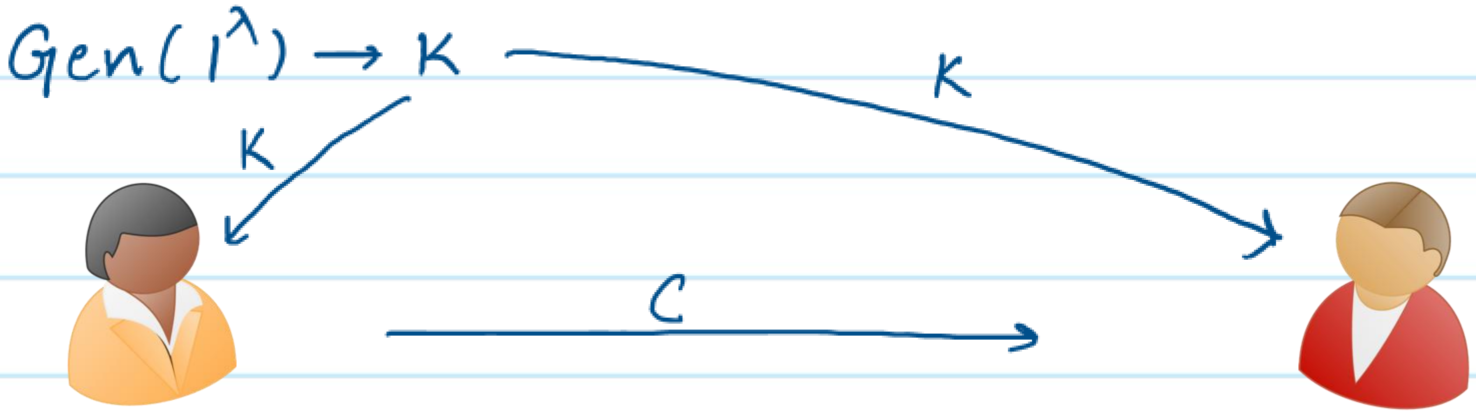
Yao's Garbled circuit [Yao'86]

Andrew Yao.

- A technique for constant-round secure 2-party computation of boolean circuits.

- Building Blocks:
  1. Oblivious Transfer
  2. Secret-Key Encryption.

Secret-Key Encryption



$C = Enc(K, m)$ $\qquad\qquad$ $m = Dec(K, c)$

Correctness: Alice can compute an encryption c of the message m using K, Bob can correctly decrypt c using K to learn m.

Security: No eavesdropper can distinguish between encryptions of m & m'

Secret-Key Encryption

Definition: Gen, Enc, Dec are PPT algorithms, s.t.,

Correctness: $K \xleftarrow{\$} \mathrm{Gen}(1^\lambda)$, $\forall m$

$$\Pr[\mathrm{Dec}(K, \mathrm{Enc}(K, m)) = m] \geq 1 - \mathrm{negl}.$$

Multi-message Security: $\forall$ n.u. PPT adversaries A, $\forall$ polynomials $q(\cdot)$,

$$\Pr\left[A\left(\{\mathrm{Enc}(m_b^i)\}_{i=1}^{q(n)}\right) = b \;\middle|\; \begin{array}{c} \text{Let} \\ K \xleftarrow{\$} \mathrm{Gen}(1^\lambda), \\ \{(m_0^i, m_1^i)\}_{i=1}^{q(\lambda)} \leftarrow A(1^\lambda), \\ b \xleftarrow{\$} \{0,1\} \end{array} \right] \leq \frac{1}{2} + \mathrm{negl}.$$

Special Encryption Scheme

We need a secret-key encryption scheme with an extra property: $\forall \lambda$, $\forall m \in \{0,1\}^\lambda$,

$$\Pr[k \xleftarrow{\$} Gen(1^\lambda),\ k' \xleftarrow{\$} Gen(1^\lambda),\ Dec(k', Enc(k,m)) = \perp] \geq 1 - negl$$

That is, if a ciphertext is decrypted using the "wrong" key, then the answer is always $\perp$

Exercise: Think about how you can design a secret key encryption with such a property

Garbled Circuits

A garbling scheme consists of two procedures:

- Garble(C): Takes as input a circuit C and outputs a collection of garbled gates $\hat{G}$ & garbled input wires $\hat{In}$:

$$\hat{G} = \{ \hat{g}_1, \ldots, \hat{g}_{|C|} \}, \quad \hat{In} = \{ \hat{in}_1, \ldots, \hat{in}_n \}$$

- Eval($\hat{G}$, $\hat{in}_x$): Takes as input a garbled circuit $\hat{G}$ and garbled input wires $\hat{In}_x$ corresponding to an input x, and outputs $z = C(x)$.

Garbled Circuits : High-Level Idea

- Each wire in the circuit is associated with two keys (K_0^i, K_1^i) of a secret-key encryption scheme. K_0^i corresponds to the wire value being 0 & K_1^i corresponds to the wire value being 1.

- For an input x, the evaluator is given the input wire keys $(K^1_{x_1}, \ldots K^n_{x_n})$. Also, for every gate g in the circuit C, it is given an encrypted truth table of g.

- We want the evaluator to use the input wire keys & encrypted truth tables to uncover a single key K_v^i for every internal wire i. corresponding to value v of that wire. We want K^i_{1-v} to remain hidden from the evaluator.

Construction of Garbled Circuits

Assign an index i to each wire in C, s.t., the input wires have indices $1, \ldots, n$.

Garble (C)

garble wires | garble gates.

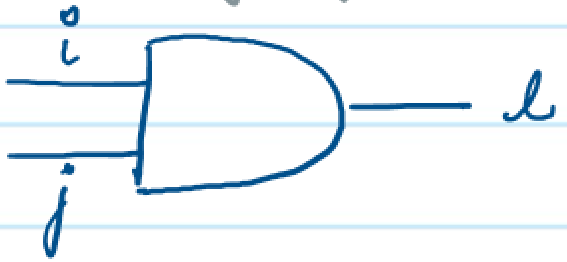
Garbling Wires

1. $\forall$ non-output wires i : $K_0^i \xleftarrow{\$} Gen(1^\lambda)$, $K_1^i \xleftarrow{\$} Gen(1^\lambda)$
2. $\forall$ output wires i : $K_0^i = 0$, $K_1^i = 1$
3. $\forall$ $i \in [n]$, Set $\hat{in}_i = (K_0^i, K_1^i)$, Set $\hat{In} = (\hat{in}_1, \ldots, \hat{in}_n)$

Construction of Garbled Circuits

Garbling gates

$\forall$ gates $g \in C$



Truth table

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

Encrypted truth table:

K_0^i	K_0^j	$Z_1 = Enc(K_0^i, Enc(K_0^j, K_1^l))$
K_0^i	K_1^j	$Z_2 = Enc(K_0^i, Enc(K_1^j, K_1^l))$
K_1^i	K_0^j	$Z_3 = Enc(K_1^i, Enc(K_0^j, K_1^l))$
K_1^i	K_1^j	$Z_4 = Enc(K_1^i, Enc(K_1^j, K_0^l))$

- Set $\hat{g} = RandShuffle(Z_1, Z_2, Z_3, Z_4)$ $\leftarrow$ why is this necessary?
- Output $\hat{G} = (\hat{g}_1, \ldots, \hat{g}_{|C|}), \hat{In}$.

Construction of Garbled Circuits

Eval $(\hat{G}, \hat{In}_x)$:

1. Parse $\hat{G} = (\hat{g}_1, \ldots, \hat{g}_{|C|})$, $\hat{In}_x = (K^1, \ldots, K^n)$
2. Parse $\hat{g}_i = (\hat{g}_i^1, \hat{g}_i^2, \hat{g}_i^3, \hat{g}_i^4)$

3. For each gate g:
   - Let K^i & K^j be input wire keys for gate g.
   - Repeat the following $\forall p \in [4]$:

     $$\alpha_p = \mathrm{Dec}(K^i, (\mathrm{Dec}(K^j, \hat{g}^p))$$

     if $\exists \alpha_p \neq \perp$, set $K^l = \alpha_p$.

4. Let out_i be the value obtained for output wire i.
   Output $out = (out_1, \ldots, out_n)$.

Secure Computation from Garbled Circuits

Want to compute

input : $x = x_1, \ldots x_n$ $\quad z = C(x, y)$ $\quad$ input : $y = y_1, \ldots, y_n$

- $\hat{G}, \hat{In} \leftarrow \text{Garble}(C)$

$$\xrightarrow{\hat{G},\ \hat{In}_x}$$

How does Bob learn $\hat{In}_y$?

Can Alice send both keys for the input wires corresponding to Bob's input?

Secure Computation from Garbled Circuits

Want to compute

input : $x = x_1, \dots, x_n$ $\quad z = C(x, y) \quad$ input : $y = y_1, \dots, y_n$

- $\hat{G}, \hat{In} \leftarrow \text{Garble}(C)$

$$\xrightarrow{\quad \hat{G},\ \hat{In}_x \quad}$$

$\forall i \in [n]$, $\quad K_0^{i+n} \rightarrow$, $K_1^{i+n} \rightarrow \boxed{OT} \leftarrow y_i$ $\quad \rightarrow K_{y_i}^{i+n}$

$\hat{In}_y = (K_{y_1}^{n+1}, \dots, K_{y_n}^{2n})$

$z = \text{Eval}(\hat{G}, \hat{In}_x, \hat{In}_y)$

$$\xleftarrow{\quad z \quad}$$

Output z $\qquad$ Output z

Security

- What does Alice learn about Bob's input wires?
- Does Alice learn anything about the internal wire values?

- What does Bob learn about Alice's input wires
- What does he learn about the internal wire values?
- Do the keys corresponding to internal wires leak any information?