Garbled Circuits: Intro to MPC

Multiparty Computation

Cryptography

Cryptography

Secure Communication

- AES, RSA, etc.
- WhatsApp, HTTPS

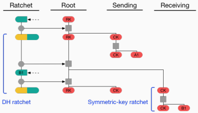


Figure 1: Double Ratchet: Signal

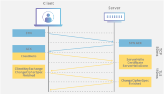


Figure 2: TLS: Cloudflare

Cryptography

Secure Communication

- AES, RSA, etc.
- WhatsApp, HTTPS

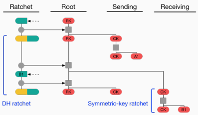


Figure 1: Double Ratchet: Signal

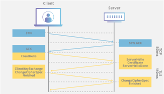
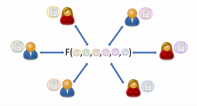


Figure 2: TLS: Cloudflare

Secure **Computation**

- Modern day constructions
- MPC, FE, FHE, ZK, and more
- Secure Voting and Auctions, Cryptocurrency, Secure ML



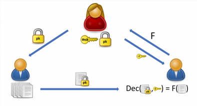


Figure 3: MPC: COSIC

Secure Multiparty Computation

- Can we compute a function of data from multiple parties **securely**?

Secure Multiparty Computation

- Can we compute a function of data from multiple parties **securely**?

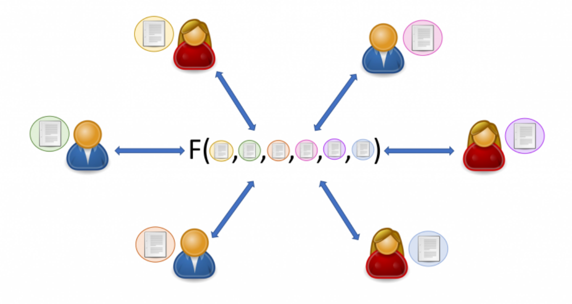


Figure 4: Secure MPC: Cosic

Secure Multiparty Computation

- We have two employees, who would like to compute the average salary without revealing individual salaries.

Secure Multiparty Computation

- We have two employees, who would like to compute the average salary without revealing individual salaries.
- Train a machine learning model without revealing training datasets

Secure Multiparty Computation

- We have two employees, who would like to compute the average salary without revealing individual salaries.
- Train a machine learning model without revealing training datasets
- Evaluate some classifier on data without revealing the data or the trained model

Secure Multiparty Computation

- We have two employees, who would like to compute the average salary without revealing individual salaries.
- Train a machine learning model without revealing training datasets
- Evaluate some classifier on data without revealing the data or the trained model
- Hiding auction bids on a smart contract

Secure Multiparty Computation

- First two party protocols introduced by Yao - Garbled Circuits
- Lets build it from the ground up!

Secure Multiparty Computation

- First two party protocols introduced by Yao - Garbled Circuits
- Lets build it from the ground up!
- Compute function between two parties P_1 and P_2
- Both P_1 and P_2 are honest

Garbled Circuits

Some Notation

- Let $\mathcal{F}(x, y)$ be a function of x and y.

Some Notation

- Let $\mathcal{F}(x, y)$ be a function of x and y.
- Let $X = \{x_0, x_1, \ldots\}$ be the set of possible x
  Let $Y = \{y_0, y_1, \ldots\}$ be the set of possible y.

Some Notation

- Let $\mathcal{F}(x, y)$ be a function of x and y.
- Let $X = \{x_0, x_1, \ldots\}$ be the set of possible x
  Let $Y = \{y_0, y_1, \ldots\}$ be the set of possible y.
- P_1 is the party choosing x, P_2 is the party choosing y
- P_1, P_2 want to compute $\mathcal{F}(x_a, y_b)$ without revealing a, b.

Garbled Circuits

- Given a function $\mathcal{F}(x, y)$, how do we evaluate $\mathcal{F}(x_a, y_b)$?

Garbled Circuits

- Given a function $\mathcal{F}(x, y)$, how do we evaluate $\mathcal{F}(x_a, y_b)$?
- The function $\mathcal{F}$ is just a table!

Table 1: A function with $X = \{x_0, x_1\}$, $Y = \{y_0, y_1\}$

	y_0	y_1
x_0	$\mathcal{F}(x_0, y_0)$	$\mathcal{F}(x_0, y_1)$
x_1	$\mathcal{F}(x_1, y_0)$	$\mathcal{F}(x_1, y_1)$

Garbled Circuits

- P_1 assigns key k_i^x to x_i, and assigns key k_i^y to y_i.

Garbled Circuits

- P_1 assigns key k_i^x to x_i, and assigns key k_i^y to y_i.

Table 2: A Garbled Function $\tilde{\mathcal{F}}$

	y_0	y_1
x_0	$\mathrm{Enc}(k_0^x \oplus k_0^y, \mathcal{F}(x_0, y_0))$	$\mathrm{Enc}(k_0^x \oplus k_1^y, \mathcal{F}(x_0, y_1))$
x_1	$\mathrm{Enc}(k_1^x \oplus k_0^y, \mathcal{F}(x_1, y_0))$	$\mathrm{Enc}(k_1^x \oplus k_1^y, \mathcal{F}(x_1, y_1))$

Garbled Circuits

- P_1 assigns key k_i^x to x_i, and assigns key k_i^y to y_i.

Table 2: A Garbled Function $\tilde{\mathcal{F}}$

	y_0	y_1
x_0	$\mathrm{Enc}(k_0^x \oplus k_0^y, \mathcal{F}(x_0, y_0))$	$\mathrm{Enc}(k_0^x \oplus k_1^y, \mathcal{F}(x_0, y_1))$
x_1	$\mathrm{Enc}(k_1^x \oplus k_0^y, \mathcal{F}(x_1, y_0))$	$\mathrm{Enc}(k_1^x \oplus k_1^y, \mathcal{F}(x_1, y_1))$

- We can send the encrypted values to P_2

Garbled Circuits

- P_1 assigns key k_i^x to x_i, and assigns key k_i^y to y_i.

Table 2: A Garbled Function $\tilde{\mathcal{F}}$

	y_0	y_1
x_0	$\mathrm{Enc}(k_0^x \oplus k_0^y, \mathcal{F}(x_0, y_0))$	$\mathrm{Enc}(k_0^x \oplus k_1^y, \mathcal{F}(x_0, y_1))$
x_1	$\mathrm{Enc}(k_1^x \oplus k_0^y, \mathcal{F}(x_1, y_0))$	$\mathrm{Enc}(k_1^x \oplus k_1^y, \mathcal{F}(x_1, y_1))$

- We can send the encrypted values to P_2
- If P_2 has the two keys associated to x_a, y_b, they can get the output!

Garbled Circuits

- Problem: P_2 knows which row/column is based on which value

Garbled Circuits

- Problem: P_2 knows which row/column is based on which value
- **Shuffle** the table

Table 3: A Garbled Function $\tilde{\mathcal{F}}$

	y_0	y_1
x_1	$\mathrm{Enc}(k_1^x \oplus k_0^y, \mathcal{F}(x_1, y_0))$	$\mathrm{Enc}(k_1^x \oplus k_1^y, \mathcal{F}(x_1, y_1))$
x_0	$\mathrm{Enc}(k_0^x \oplus k_0^y, \mathcal{F}(x_0, y_0))$	$\mathrm{Enc}(k_0^x \oplus k_1^y, \mathcal{F}(x_0, y_1))$

Garbled Circuits

- Problem: P_2 knows which row/column is based on which value
- **Shuffle** the table

Table 3: A Garbled Function $\tilde{\mathcal{F}}$

	y_0	y_1
x_1	$\text{Enc}(k_1^x \oplus k_0^y, \mathcal{F}(x_1, y_0))$	$\text{Enc}(k_1^x \oplus k_1^y, \mathcal{F}(x_1, y_1))$
x_0	$\text{Enc}(k_0^x \oplus k_0^y, \mathcal{F}(x_0, y_0))$	$\text{Enc}(k_0^x \oplus k_1^y, \mathcal{F}(x_0, y_1))$

- Pointer value tells P_2 which row/column is correct
- p_0^x, $p_1^x = 1 - p_0^x$

Garbled Circuits

- P_1 sends P_2 the garbled $\tilde{\mathcal{F}}$
- P_1 sends P_2 the key k_a^x and p_a^x
- How does P_2 get k_b^y without P_1 learning b?

Oblivious Transfer

- Introducing a new protocol: Oblivious Transfer!

Oblivious Transfer

- Introducing a new protocol: Oblivious Transfer!
- Alice has k_0^y and k_1^y
- Bob has a bit b

Oblivious Transfer

- Introducing a new protocol: Oblivious Transfer!
- Alice has k_0^y and k_1^y
- Bob has a bit b
- Alice learns nothing, and Bob learns k_b^y

Oblivious Transfer

- Oblivious transfer exists
- Probably ask me later about how to construct OT
- Relies on some more cryptography concepts not taught

Garbled Circuits

- Back to the garbled table
- P_1 sends $\tilde{\mathcal{F}}$, k_a^x, p_a^x
- P_1 and P_2 do Oblivious Transfer to send k_b^y, p_b^y to P_2

Garbled Circuits

- Back to the garbled table
- P_1 sends $\tilde{\mathcal{F}}$, k_a^x, p_a^x
- P_1 and P_2 do Oblivious Transfer to send k_b^y, p_b^y to P_2
- With both pointers, P_2 can find the location in the table

Garbled Circuits

- Back to the garbled table
- P_1 sends $\tilde{\mathcal{F}}$, k_a^x, p_a^x
- P_1 and P_2 do Oblivious Transfer to send k_b^y, p_b^y to P_2
- With both pointers, P_2 can find the location in the table
- With both keys, P_2 can decrypt that row, to get the output

Garbled Circuits

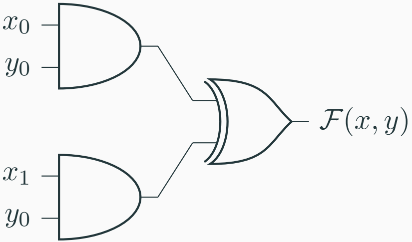


Figure 5: Example Circuit

Garbled Circuits

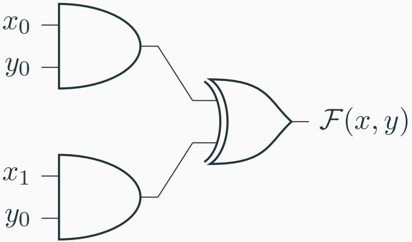


Figure 5: Example Circuit

- Lookup table?? Exponential

Garbled Circuits

Table 4: A Garbled Gate $\tilde{\mathcal{F}}$

	y_0	y_1
x_1	$\mathrm{Enc}(k_1^x \oplus k_0^y, k^{\mathsf{out}}_{\mathcal{F}(x_1,y_0)})$	$\mathrm{Enc}(k_1^x \oplus k_1^y, k^{\mathsf{out}}_{\mathcal{F}(x_1,y_1)})$
x_0	$\mathrm{Enc}(k_0^x \oplus k_0^y, k^{\mathsf{out}}_{\mathcal{F}(x_0,y_0)})$	$\mathrm{Enc}(k_0^x \oplus k_1^y, k^{\mathsf{out}}_{\mathcal{F}(x_0,y_1)})$

Garbled Circuits

Table 4: A Garbled Gate $\tilde{\mathcal{F}}$

	y_0	y_1
x_1	$\mathrm{Enc}(k_1^x \oplus k_0^y, k^{\mathsf{out}}_{\mathcal{F}(x_1,y_0)})$	$\mathrm{Enc}(k_1^x \oplus k_1^y, k^{\mathsf{out}}_{\mathcal{F}(x_1,y_1)})$
x_0	$\mathrm{Enc}(k_0^x \oplus k_0^y, k^{\mathsf{out}}_{\mathcal{F}(x_0,y_0)})$	$\mathrm{Enc}(k_0^x \oplus k_1^y, k^{\mathsf{out}}_{\mathcal{F}(x_0,y_1)})$

Garbled Circuits

- Assign pair of keys (and pointers) **per wire**
- Can evaluate circuits gate by gate!

More Fun Things

- This is just the surface: many more questions
- How can we do this with multiple parties?
- Can we improve communication efficiency?

More Fun Things

- Very useful in the world of secure computation
- For example, say the circuit is a circuit which takes in an ML classifier and an image, and outputs the classification
- Allows doctors to provide medical images / data to ML models without violating HIPAA

More Fun Things

Bonus: Funny card trick (related to MPC)!!

More Fun Things

$$(C, C)C(C, C)$$

More Fun Things

Both say yes:

CCCCC

More Fun Things

One or both says no:

CCCCC