This Lecture:

- Feasibility Results of MPC
- Yao's Garbled Circuits and Optimizations
Secure Two-Party Computation (2PC)

Alice

Bob

$A \xrightarrow{} L \xrightarrow{} A$

$X \xleftarrow{} C \xrightarrow{} z = f(x, y)$

$B \xrightarrow{} o \xleftarrow{} y$
Secure Multi-Party Computation (MPC)

\[ z = f(x_1, \ldots, x_n) \]
Setting

- n parties P₁, P₂, ..., Pₙ
  
  with private inputs X₁, X₂, ..., Xₙ

- Jointly compute f(X₁, X₂, ..., Xₙ)

Communication:

Authenticated secure point-to-point channels between each pair (Pᵢ, Pⱼ)

(sometimes also assume broadcast channel)

- The adversary can “corrupt” a subset of the parties
  
  (e.g. at most t parties)
General Security Properties

- **Correctness**: The function is computed correctly.

- **Privacy**: Only the output is revealed.

- **Independence of Inputs**: Parties cannot choose inputs depending on others' inputs.

- **Security with Abort**: Adversary may “abort” the protocol.
  (preventing honest parties from receiving the output)

- **Fairness**: If one party receives output, then all receive output.

- **Guaranteed Output Delivery (GOD)**: Honest parties always receive output.
Adversary's Power

Allowed adversarial behavior:

- Semi-honest/passive/honest-but-curious:
  Follow the protocol description honestly, but try to extract more information by inspecting transcript.

- Malicious/active:
  Can deviate arbitrarily from the protocol description.

Adversary's Computing Power:

- Unbounded computing power \Rightarrow Information-Theoretic (IT) Security
- PPT bounded \Rightarrow Computational Security
Feasibility Results

**Computational Security:**

- Semi-honest Oblivious Transfer (OT)

\[ \downarrow \]

- Semi-honest MPC for any function with \( t < n \)

\[ \downarrow \]

- Malicious MPC for any function with \( t < n \)

**Information-Theoretic (IT) Security:**

- Semi-honest/malicious MPC for any function with \( t < n/2 \)

(honest majority)
Oblivious Transfer (OT)

Input: $m_0, m_2 \in \mathbb{F}_2^n$

Input: $b \in \mathbb{F}_2$

Output: $L$

Output: $m_b$
Semi-honest OT \xrightarrow{\text{Yao's Garbled Circuit}} \text{semi-honest 2PC for any function}

\text{semi-honest MPC for any function}
Example: Private Dating

Alice

Bob

\[ f(x, y) = x \land y \]

Truth Table:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Example: Private Dating

Alice

Bob

$X \in \{0, 1\}$

$y \in \{0, 1\}$

Garbled Truth Table:

| $\text{Enc}_{\alpha_0}(\text{Enc}_{\beta_0}(0))$ | $\alpha_0 \beta_0 \Rightarrow 0$ |
| $\text{Enc}_{\alpha_0}(\text{Enc}_{\beta_1}(0))$ | $\alpha_0 \beta_1 \Rightarrow 0$ |
| $\text{Enc}_{\alpha_1}(\text{Enc}_{\beta_0}(0))$ | $\alpha_1 \beta_0 \Rightarrow 0$ |
| $\text{Enc}_{\alpha_1}(\text{Enc}_{\beta_1}(1))$ | $\alpha_1 \beta_1 \Rightarrow 1$ |
Example: Private Dating

Alice

\[ X \in \{0, 1\} \]

\[ \text{AND} \]

Bob

\[ Y \in \{0, 1\} \]

\[ \text{Input label for } X: \alpha_X \]

Input label for \( y \): 

\[ \alpha_0, \alpha_1, p_0, p_1 \xleftarrow{\$ \{0, 1\}^*} \text{ (labels)} \]
Arbitrary Function: Represent it as a Boolean circuit

Alice

$X \in \{0,1\}^2$

Bob

$Y \in \{0,1\}^2$

\[
\begin{align*}
X & \quad \text{AND} \quad \text{AND} \\
\quad & \quad \text{XOR} \\
\quad & \\
\quad & \\
\end{align*}
\]
Yao's Garbled Circuit

Each label $\xleftarrow{\$} \{0,1\}^*$
Secure 2PC

Alice (Garbler)
\[ X \in \{0,1\}^2 \]

Bob (Evaluator)
\[ y \in \{0,1\}^2 \]

Garbled Circuit

Garbled Gates

Input labels for x

Input labels for y?

· Output labels?

· How to decide which ciphertext to decrypt?
Optimization 1: Point-and-Permute

\[ Y_0 \parallel b_1 \]
\[ Y_1 \parallel \overline{b}_1 \]
\[ \alpha_0 \parallel \overline{b}_1 \]
\[ \alpha_1 \parallel b_1 \]
\[ \beta_0 \parallel b_1 \]
\[ \beta_1 \parallel \overline{b}_1 \]

\[ \alpha, \beta, b \in \{0, 1\} \] (signal bits)

\[ b_\alpha \quad b_\beta : \quad Enc_{\alpha_0} (Enc_{\beta_0} (Y_0 \parallel b_1)) \]
\[ \overline{b}_\alpha \quad b_\beta : \quad Enc_{\alpha_1} (Enc_{\beta_1} (Y_0 \parallel b_1)) \]
\[ \alpha \quad b_\beta : \quad Enc_{\alpha_2} (Enc_{\beta_1} (Y_1 \parallel \overline{b}_1)) \]
Optimization 2: Row Reduction

\[ H(\alpha \| \beta_0) \oplus Y_0 \]
\[ H(\alpha \| \beta_1) \oplus Y_0 \]
\[ H(\alpha_1 \| \beta_0) \oplus Y_0 \]
\[ H(\alpha_1 \| \beta_1) \oplus Y_1 \]
Optimization 3: Free XOR

Sample a global $\Delta \leftarrow \{0, 1\}^n$

$\alpha_2 := \alpha_0 \oplus \Delta$

$\beta_2 := \beta_0 \oplus \Delta$

$Y_2 := Y_0 \oplus \Delta$

$Y_0 := \alpha_0 \oplus \beta_0$

Other Optimizations:

- Half-gates: $2\lambda$ bits per AND gate + free XOR
- Slicing-and-dicing: $\approx 1.5\lambda$ bits per AND gate + free XOR