Garbled circuits
Recap

• Federated learning
  • Private summation of private model updates
  • Arithmetic secret sharing, Shamir secret sharing
• Password breach alert
  • Specialized private set intersection (PSI)
  • Oblivious PRF, DDH assumption
Garbled circuits

- Generic computation using a circuit-based computation model
  - Each party inputs a set of bits
  - Circuit made up of XOR and AND gates
  - Each gate has two input wire, and one output wire
Garbled circuits definitions

- Garble($1^n, F$) → ($GC_F, e, d$): $n$ is the security parameter, $F$ is an input function to be garbled (represented as a boolean circuit), $GC_F$ represents the garbled circuit for function $F$, $e$ is the encoding information, and $d$ is the decoding information

- Encode($e, x$) → $E_x$: $k$ is a key, and $x$ is corresponding input, $E_x$ is the corresponding garbled input

- Eval($GC_F, E_x$) → $E_y$: on input a garbled circuit $GC_F$ and an encrypted input $E_x$, produce a garbled output $E_y$

- Decode($d, E_y$) → $F(x)$: using decoding information $d$ and garbled output $E_y$, output $y = F(x)$
Garbled circuits

- Alice & Bob want to figure out whether they should collaborate on a project, but doesn’t want to reveal their own input

- Alice: $x$, Bob: $y$; want to compute $x \land y$ (circuit with a single gate)

- Two parties: Garbler (Alice) & Evaluator (Bob)
  - Garbler generates the circuit
  - Evaluator evaluates the circuit

- Basic idea: encode the truth table of a gate using encryption
Garbled circuits

• Alice & Bob want to figure out whether they should collaborate on a project, but doesn’t want to reveal their own input

• Alice: $x$, Bob: $y$; want to compute $x \land y$ (circuit with a single gate)

• Two parties: Garbler (Alice) & Evaluator (Bob)
  • Garbler generates the circuit
  • Evaluator evaluates the circuit

• Basic idea: encode the truth table of a gate using encryption

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$x \land y$</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>
Garbled circuits: Garbler

- Let $H$ be a key derivation function
- Pick four random labels: $W^0_x, W^1_x, W^0_y, W^1_y$, which correspond to the four possible values for $x$ and $y$
Garbled circuits: Garbler

- Let $H$ be a key derivation function.
- Pick four random labels: $W_x^0$, $W_x^1$, $W_y^0$, $W_y^1$, which correspond to the four possible values for $x$ and $y$.

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$x \land y$</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>0</td>
<td>0</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>

![Diagram of AND gate with random labels](image-url)
Garbled circuits: Garbler

- Let $H$ be a key derivation function
- Pick four random labels: $W^0_x, W^1_x, W^0_y, W^1_y$, which correspond to the four possible values for $x$ and $y$
- For each row
  - Use $H$ to derive a key using the corresponding labels
  - Encrypt the content

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$x \land y$</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>
Garbled circuits: Garbler

- Let $H$ be a key derivation function.
- Pick four random labels: $W^0_x, W^1_x, W^0_y, W^1_y$, which correspond to the four possible values for $x$ and $y$.
- For each row:
  - Use $H$ to derive a key using the corresponding labels.
  - Encrypt the result.

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$x \land y$</th>
<th>Enc($H(W^0_x, W^0_y), 0$)</th>
<th>Enc($H(W^0_x, W^1_y), 0$)</th>
<th>Enc($H(W^1_x, W^0_y), 0$)</th>
<th>Enc($H(W^1_x, W^1_y), 1$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Garbled circuits: Garbler

- Let $H$ be a key derivation function.
- Pick four random labels: $W^0_x, W^1_x, W^0_y, W^1_y$, which correspond to the four possible values for $x$ and $y$.
- For each row:
  - Use $H$ to derive a key using the corresponding labels.
  - Encrypt the result.
  - Randomly permute the rows.
Garbled circuits: Garbler

- Let $H$ be a key derivation function.
- Pick four random labels: $W_x^0, W_x^1, W_y^0, W_y^1$, which correspond to the four possible values for $x$ and $y$.
- For each row:
  - Use $H$ to derive a key using the corresponding labels.
  - Encrypt the result.
  - Randomly permute the rows.
**Garbled circuits: Evaluator**

- In order to evaluate the gate, needs to know the correct label for each input wire.
- Alice the Garbler can send her input wire label over directly — nothing is revealed since it’s random.
- What about Bob’s value?
  - Alice should give the right label, without learning Bob’s input.
  - Bob should only learn one label, not two.
- Use **Oblivious Transfer**!

![Diagram of Garbled circuits: Evaluator](image)
Garbled circuits: Evaluator

- In order to evaluate the gate, needs to know the correct label for each input wire.
- Alice the Garbler can send her input wire label over directly — nothing is revealed since it’s random.
- What about Bob’s value?
  - Alice should give the right label, without learning Bob’s input.
  - Bob should only learn one label, not two.
  - Use **Oblivious Transfer**!

\[
\begin{align*}
W^0_y, W^1_y & \rightarrow \text{OT} \quad y = 1 \\
W^1_y & \leftarrow \text{OT}
\end{align*}
\]

Garbler doesn’t learn \(y\)
Evaluator doesn’t learn \(W^0_y\)
Garbled circuits: Evaluator

- In order to evaluate the gate, needs to know the correct label for each input wire.
- Alice the Garbler can send her input wire label over directly — nothing is revealed since it’s random.
- What about Bob’s value?
  - Alice should give the right label, without learning Bob’s input.
  - Bob should only learn one label, not two.
- Use Oblivious Transfer!
- Use $H$ to generate key, decrypt all four entries using key; if succeeds, output the result.
Garbled circuits

- Extending to a circuit requires encrypting labels
- **Security**
  - Semihonest construction (otherwise garbler could choose an incorrect circuit, which requires other techniques)
  - Garbler is corrupted: security of OT
  - Evaluator is corrupted:
    - Labels are random
    - Permutation ensures no information is leaked from the organization of the circuit
    - Only one label is learned per wire
Today’s reading: SecureML
Next class

• 2-party, convolutional neural network inference

• Setup:
  • Server provides model
  • Client provides input
  • Client wants to run inference on the model without revealing the input; server does not want to reveal the model

• Techniques: HE (linear) & GC (non-linear)