Today: Yao’s garbled circuits

Last lecture:

- Defined secure 2-party computation in the Honest-But-Curious (HBC) model.

- Defined a building block: Oblivious Transfer (OT), which is a secure 2-party protocol for the function

  \[ f_{OT}(m_0, m_1, b) = (1, m_b). \]

- Constructed OT in the HBC model, based on any public-key encryption scheme that has an oblivious \( \mathrm{OGen} \) alg \( \{ \) that samples public keys without knowing a secret key \( st: \)

\[
(\Gamma, pk = \mathrm{OGen}(\mathbb{1}^\nu; \Gamma), \, \mathrm{Enc}_{pk}(0)) \equiv (\Gamma, pk = \mathrm{OGen}(\mathbb{1}^\nu; \Gamma), \, \mathrm{Enc}_{pk}(i))
\]

\[
\begin{align*}
S & \quad m_0, m_1, \quad R \\
\leftrightharpoons & \quad pk_0, pk_1, \quad (pk_b, sk_b) \leftarrow \mathrm{Gen}(\mathbb{1}^\nu) \\
\leftrightharpoons & \quad pk_{1-b} \leftarrow \mathrm{OGen}(\mathbb{1}^\nu)
\end{align*}
\]

\[ C_0 = \mathrm{Enc}_{pk_0}(m_0) \]

\[ C_1 = \mathrm{Enc}_{pk_1}(m) \]

\[ \rightarrow C_0, C_1 \]
We saw that this protocol is secure in the HBC model.

In the HW we give an OT protocol which offers better security.

**Thm:** If $\exists$ secure OT in HBC model, then $\exists$ secure 2-party comp. in HBC model & func. $f$.

**Main tool:** Yao Garbled Circuits

**Def:** Yao's Garbled Circuit is defined by two PPT alg's (Garble, Eval).

**Syntax** Garble takes as input a pair $(1^k, C)$ where $k$ is the security parameter, &
$C : \{0,1\}^n \to \{0,1\}$ is a Boolean circuit. It outputs

$$\left( \tilde{C}, \{ \ell_i, b_i \}_{i \in [n]} \right).$$

"garbled" \(C\)

labels corresponding to input wires.
Eval takes as input a garbled circuit $\tilde{C}$ and $n$ labels $\{l_i, x_i \}_{i \in [n]}$ corresponding to some input $x = (x_1, x_n) \in \{0, 1\}^n$ and outputs a bit $b$.

**Correctness:** $\forall C : \{0, 1\}^n \rightarrow \{0, 1\} \forall x \in \{0, 1\}^n \forall \{l_i, x_i \}_{i \in [n]} \Pr[\text{Eval}(\tilde{C}, \{l_i, x_i \}_{i \in [n]}) = C(x)] = 1 - \text{negl}(x)$

where $(\tilde{C}, \{l_i, x_i \}_{i \in [n]}) \leftarrow \text{Garble}(1^k, C)$

**Security:** Intuitively, $(\tilde{C}, \{l_i, x_i \}_{i \in [n]})$ reveal only $(C, C(x))$ but do not reveal anything else about $x$.

Formally: $\exists$ PPT alg $S$ s.t. $\forall C : \{0, 1\}^n \rightarrow \{0, 1\}$ $\forall x \in \{0, 1\}^n$

$(\tilde{C}, \{l_i, x_i \}_{i \in [n]}) \equiv \text{Sim}(C, C(x))$

where $(\tilde{C}, \{l_i, b \}_{i \in [n]}) \leftarrow \text{Garble}(1^k, C)$.

**Note:** We assume w.l.o.g. $C$ is public. If $C = C_{sk}$ is
associated with some secret $sk$, then simply consider the "universal" circuit $U$ that takes as input a pair $(sk, x)$ and outputs $C_{sk}(x)$, and garble $U$.

**Thm:** HBC OT & Yao Garbled circuit

$\Rightarrow$ Secure 2-party comp. in HBC model.

**Pf:** Let $f$ be a function $f : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$

Let $C$ be a circuit computing $f$.

\[ A \quad \quad \quad C \quad \quad \quad B \]
\[ x \in \{0, 1\}^n \]
\[ y \in \{0, 1\}^n \]

\[ (\widetilde{C}, \{i, x \mid i \in [n]\}) \leftarrow \text{Garble}(A^n, C) \]

\[ \widetilde{C}, \{i, x \mid i \in [n]\} \rightarrow \]

\[ \text{OT} \]

\[ \forall i \in [n]: \quad y_i \in \{0, 1\} \]

\[ l_{n+i}, y_i \]
$B$ computes $\text{Eval}(\tilde{C}, \tilde{l}_1, x, \tilde{l}_n, x, \tilde{l}_{n+1}, y, \ldots, \tilde{l}_{2n}, y)$

$$= C(x, y).$$

**Claim:** The view of $A$ can be simulated by an OT simulator that simulates the sender view as follows:

**PF:**

$S_A(x)$: Compute $(\tilde{C}, \{l_i, b_i : i \in [2n]\}) \leftarrow \text{Garble}(1^k, C)$

Compute $S^\text{OT}_{\text{sender}}(1^k, l_{n+i}, 0, l_{n+i}, 1)$.

Output: $(\tilde{C}, \{l_i, x, b_i \mid i \in [n]\}, \{(S^\text{OT}_{\text{sender}}(1^k, l_{n+i}, 0, l_{n+i}), i \in [n]\})$ together with randomness of Garble.

The view of $B$ can be simulated by OT Simulator $S^\text{OT}_{\text{receiver}}$ & Yao Simulator $S^\text{Yao}$, as follows:

$S_B(y, c(x, y))$:

Compute $(\tilde{C}, \{l_i b_i \mid i \in [2n]\}) \leftarrow S^\text{Yao}(C, c(x, y))$

$\forall i \in [n]$ Compute $S^\text{OT}_{\text{receiver}}(1^k, y_i, l_{n+i})$
Output \((\overline{C}, (i, \overline{b_i})_{i \in [n]}, (S_{\text{receiver}}(1^n, y_i, l_{n+i}))_{i \in [n]} )\)

**Construction of Yao's Garbled Circuits**

Let \(C : \{0,1\}^n \rightarrow \{0,1\}\). Assume w.l.o.g. that \(C\) consists only of NAND gates of fanin 2:

\[
\begin{array}{ccc}
0, 1 & \rightarrow & 1 \\
1, 0 & \rightarrow & 1 \\
0, 0 & \rightarrow & 0 \\
1, 1 & \rightarrow & 0
\end{array}
\]

NAND is known to be a complete operation.

Let \((\text{Gen}, \text{Enc}, \text{Dec})\) be a secret-key semantically secure encryption scheme w. an extra property:

\[
\Pr \left[ \text{Dec}_{sk'}(C) = 1 : sk \leftarrow \text{Gen}(1^K), sk' \leftarrow \text{Gen}(1^K), c \leftarrow \text{Enc}_{sk}(m) \right] \geq 1 - \epsilon(K)
\]

I.e.: When a ciphertext is decrypted w. incorrect key, then the output is almost always 1.

We call this a **special enc. scheme**.
Construction of special enc. scheme:

Let \( \{ f_s : \{0,1\}^{1*} \rightarrow \{0,1\}^{a*} \} \) a PEF family.

\( \text{Gen}(1^k) : \) Output \( s \in \{0,1\}^k \)

\( \text{Enc}(m) : \) Sample \( r \in \{0,1\}^k \) and output \( \{0,1\}^k (r, (0^k, m) \oplus f_s(r)) \)

\( \text{Dec}_s(c_1, c_2) : \) Compute \( (m_0, m_1) = c_2 \oplus f_s(c_1) \)

If \( m_0 = 0^k \) output \( m_1 \),

\( \) otherwise output \( 1 \).

Garble \( (1^k, C) : \) let \( (\text{Gen}, \text{Enc}, \text{Dec}) \) be a special enc. scheme.

- For wire \( i \) in \( C \) sample 2 keys \( k_{i0}, k_{i1} \in \text{Gen}(1^k) \)

  For the output wire out let \( k_{out, 0} = 0 \) and \( k_{out, 1} = 1 \)

- For gate \( g \), with input wires \( i, j \) and output wire \( l \), sample a "garbled" gate as follows:

\[
\text{NAND} (k_{i0}, k_{i1}, k_{j0}, k_{j1}, k_{l0}, k_{l1})
\]
Garble \( (g) \): \[
\begin{align*}
\text{Output} & \quad \left\{ \text{Garble} \ (g) \right\} \forall g \in \text{GATES}, \quad \left\{ k_{io}, k_i, i \in [n] \right\} \\
\text{Eval} \ (\overline{c}, \left\{ k_{i, x} \right\}, [n]) & \quad \text{Compute for every wire } w \\
\end{align*}
\]

- **Compute for every wire** \( w \) the key \( k_{w_0, w_0} \), where \( w_0 \) corresponds to the value of the wire on input \( x \).

This is done by induction starting with the input wires where \( k_{i, x} \) are given.

- Given keys for layer \( i \), compute keys for the next layer by simply stripping off the double encryption from the garbled gate.
Each garbled gate consists of 4 ciphertexts. With overwhelming prob. 3 of the ciphertexts will decrypt to 1, and only one of them will decrypt to a valid value $K_e(i,0)$.  

Finally, compute $K_{out,b}$. If $K_{out,b} = 0$ then output 0, and if $K_{out,b} = 1$ then output 1. 

Correctness: Follows from the special property of the encryption scheme.

Security: $S(C,C(x))$ does the following: 

Emulate $(\tilde{C}, \{K_{i,b}^{j} \in \{0,1\} \}) \leftarrow \text{Garble}(1^{k},C)$, but rather than setting $K_{out,0} = 0$ & $K_{out,1} = 1$, set $K_{out,0} = K_{out,1} = C(x)$.  

Output $(\tilde{C}, \{K_{i,0}^{j} \in \{0,1\} \})$ 

Security follows from semantic security, and from the fact that each garbled gate is randomly ordered.