**Secure 2-Party Computation**

\[ t(x, y) = \begin{cases} 1 & \text{if } x \geq y \\ 0 & \text{if not} \end{cases} \]

**Yao's Millionaire Problem**

\[ t(x, y) = \begin{cases} 1 & \text{if } x \geq y \\ 0 & \text{if not} \end{cases} 
\]

**Functionality Protocol**

In which given inputs \( x \) for Alice and \( y \) for Bob, Bob's output equals \( t(x, y) \).

**Security.** \( (s, e) - \text{Simulatability} \) = simulators \( S_A, S_B \) s.t. \( \forall x, y \) \( S_A(x) \) ind. from A's view and \( S_B(x, t(x, y)) \) ind. from Bob's view.

\[
\text{Ex. } t(x, y) = x \text{ XOR } y. 
\]

\[
t(x, y) = y
\]

\[
\text{Sim}_B(y, z) = y \text{ XOR } z
\]

\[
\text{NO INTERACTION}
\]
\[ f(x, y) = x \text{ AND } y = \begin{cases} x & \text{when } y = 1 \\ 0 & \text{when } y = 0 \end{cases} \]

**Protocol:**

\[
\begin{array}{c}
\text{Sim}_B \\
\xrightarrow{y=0} \xrightarrow{x} \xrightarrow{y=1}
\end{array}
\]

Bob can simulate, but Alice can't.

**Idea.**

Alice \( Enc(Pk, x) \) Bob \( \begin{cases} y=0 : \text{doesn't know } Sk \\ y=1 : \text{knows } Sk \end{cases} \)

Bob

\( \begin{cases} y=1 : \text{samples } (Sk, Pk) \text{ for El Gamal ENC} \\ y=0 : \text{sample } Pk \text{ without knowing } Sk \end{cases} \)

\( Pk = g^{Sk} \) (Pick random QR)

\[
\begin{array}{c}
Pk \\
\xrightarrow{c = Enc(x, Pl)} \\
\xleftarrow{Alice}
\end{array}
\]

Bob:

- If \( y=1 \) output \( \text{Dec}(C) \).
- If \( y=0 \) output 0.

\[
\text{Sim}_B \left( y, y \right) \begin{cases} 0 & \text{when } y = 0 \\ 2 \end{cases}
\]

Use \( \text{Sim}_B \) for El Gamal

Choose random \( Pk \)

Output \( (Pl, Enc(2Pk)) \).
OBLIVIOUS TRANSFER

\[
\text{OT}(x_0, x_1, b) = (x_0 \land \overline{b}) \cup (x_1 \land b)
\]

OT PROTOCOL

\[
\text{Sim}_{ot}(x_0, x_1) = (\text{Sim}_{\land}(x_0), \text{Sim}_{\land}(x_1))
\]

SECURITY

\[
\text{Sim}_{ot}(b, x_b) = \begin{cases} 
(\text{Sim}_{\land}(1, x_b), \text{Sim}_{\land}(0, 0)) & \text{IF } b = 0 \\
(\text{Sim}_{\land}(0, 0), \text{Sim}_{\land}(1, x_b)) & \text{IF } b = 1
\end{cases}
\]
A GENERAL $f$ IS SOME CIRCUIT

A DIFFERENT PROTOCOL FOR AND

**Goal:** Alice to compute $z = x \land y$ without learning anything except $x_A, y_B$ (and $z_{xy}$)

1. Alice learns $x_A, y_B, z_{xy}$ and nothing else.
2. Output $x$ and $y$. 
3. Choose $x_0, y_1, y_0, y_1, z_0, z_1$ at random.
4. Bob by $\checkmark$
Bob

\[
\begin{align*}
X_0 \\
X_1 \\
Y_0 \\
Y_1 \\
\end{align*}
\]

\[\{ \quad 8 \text{ bits} \]

\[Z_0, Z_1 \in 2^{48} \quad k \text{ bits} \]

\text{IDEA: ENCRYPT} \quad \begin{cases} 
Z_0 \text{ UNDER KEY } X_0, Y_0 \\
Z_0 \text{ " } X_1, Y_0 \\
Z_1 \text{ " } X_1, Y_1 \\
\end{cases} \]

\text{Randomly permuted}

Alice can only decrypt \(Z_x \text{ and } y\)

Eg. \(x = 0, j = 1\)

Alice should not know what she decrypted.
USE OTP TO ENCRYPT $0^k \cdot z_c$ AND $b$.

8k BITs

E6. $x=0$, $y=1$
CAN SIMULATE (IF IN RANDOM ORDER) FROM $x_x, y_d, z_x$ AND $y$

$(0 \text{ of } 4 \text{ succeeds}) = 1 - 3 \cdot 2^{-k}$

$(z_0, z_1)$ k BITS LONG

IF WE TRY TO COMPOSE GARBLED DATA WILL GROW EXPONENTIALLY

GARBLED AND TRANSFER $x_0, x_1, y_0, y_1 \in \{0, 1\}^k$

USE $G(x_0), G(x_1), G(y_0), G(y_1)$, FOR PBF $\alpha : \{0, 1\}^k \rightarrow \{0, 1\}^8$

$(z_0, z_1)$ k BITS LONG

$(x_0, y_1) \in \{0, 1\}^k$
Garbled Circuit Protocol

0. Bob samples $w_0^w, w_1^w$ for each wire $w$.

1. Bob’s input wires: send $w_b^{val(b)}$ to Alice.
   Alice’s input wires: run OT with Alice choosing $w_{val(a)}^a$ from $(w_0^a, w_1^a)$.

2. Run Garbled gate protocol in order of gates so Alice learns $w_{val(w)}^w$ in sequence.

3. Alice sends output $Z = w_{val(out)}^w$ to Bob. Bob outputs $z \in \{0, 1\}$ s.t. $Z = w_2^w$. 
**Functionality:** Unless any garbled gate protocol fails, \( z \) must equal \( f(x, y) \), so it works with prob. \( 1 - o(\text{size}(c) \cdot 2^{-k}) \).

**Security:** Alice’s view = random values \( W^{w}_{\text{alice}} \) + views in OT/garbled gate protocols. Can compose simulators for them.

Bob’s OT/garbled gate protocol views can be simulated from his randomness \( W^{w}_{0}, W^{w}_{1} \). Alice’s last message \( W_{\text{out}}^{w} \) \( \text{val}(\text{out}) \) can be simulated as \( W^{z}_{2} \).

Bob can execute garbled gate protocols in parallel \( \rightarrow \) 3 message protocol independent of \( \text{size}(c) \)!

**Next 2 Lectures:** Malicious Alice/Bob that may deviate from protocol instructions.

**Strategy:** Commit to follow instructions and prove that you did without revealing your inputs! Zero-knowledge.
Security assumes Alice & Bob follow instructions.

Ensure security against adversaries that might deviate from instructions.

Strategy compile protocol secure against honest-but-curious to security against malicious zero-knowledge proofs.