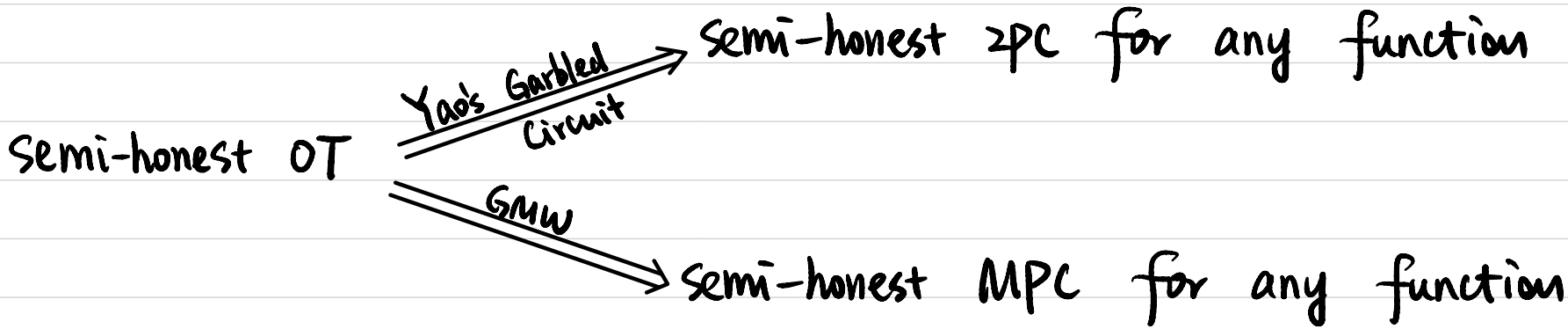
# CSCI 1515 Applied Cryptography

## This Lecture:

- Oblivious Transfer (OT)
- Putting it Together: Semi-Honest 2PC for Any Funtion
- GMW: Semi-Honest MPC for Any Funtion


semi-honest OT
Yao's Garbled Circuit
semi-honest 2PC for any function
GMW
semi-honest MPC for any function

# Oblivious Transfer (OT)

**Sender**

Input: $m_0, m_1 \in \{0,1\}^\ell$

**Receiver**

Input: $c \in \{0,1\}$

Output: $\perp$

Output: $m_c$

# Oblivious Transfer (OT)

CDH: $g^\alpha, g^\beta \not\Rightarrow g^{\alpha\beta}$ ← Then can also solve

$g^\alpha \not\Rightarrow g^{\alpha^2}$ ← If a PPT $A$ can solve

## Sender

Input: $m_0, m_1 \in \{0,1\}^\ell$

$a \xleftarrow{\$} \mathbb{Z}_q$

Given $g^\alpha, g^\beta$:

$g^\alpha \Rightarrow g^{\alpha^2}$

$g^\beta \Rightarrow g^{\beta^2}$

$g^{\alpha+\beta} \Rightarrow g^{(\alpha+\beta)^2}$

$\Downarrow$

$\left(\frac{g^{\alpha^2+\beta^2+2\alpha\beta}}{g^{\alpha^2} \cdot g^{\beta^2}}\right)^{z^{-1}}$

$A = g^a$ →

## Receiver

Input: $c \in \{0,1\}$

$b \xleftarrow{\$} \mathbb{Z}_q$

← $B = g^b \cdot A^c$

If $c = 0 \Rightarrow B = g^b$

If $c = 1 \Rightarrow B = g^b \cdot A = g^{a+b}$

$k_0 := H(B^a)$

$k_1 := H\left(\left(\frac{B}{A}\right)^a\right)$

$k_0 = H(g^{ab})$

$k_1 = H(g^{ab-a^2})$ → $g^{ab}/g^{a^2}$

$k_0 = H(g^{ab+a^2})$ → $g^{ab} \cdot g^{a^2}$

$k_1 = H(g^{ab})$

$Ct_0 \leftarrow Enc_{k_0}(m_0)$

$Ct_1 \leftarrow Enc_{k_1}(m_1)$ →

Output:

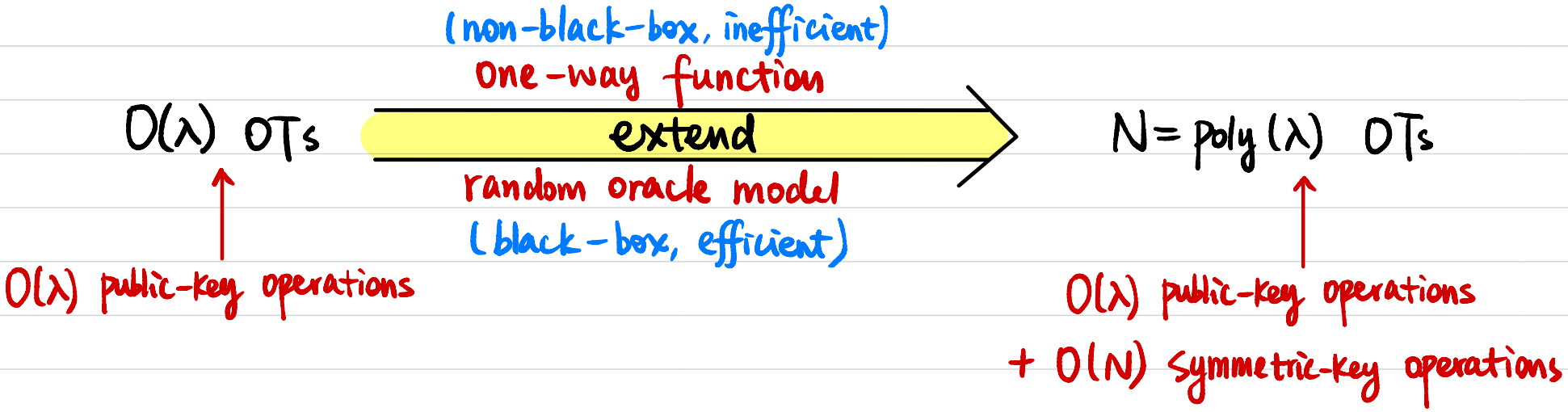
$k_c := H(A^b) = H(g^{ab})$

$m_c := Dec_{k_c}(Ct_c)$

# OT Extension

Can we construct OT from symmetric-key primitives only?

Unlikely! (theoretical impossibility)



# Putting it Together: Semi-Honest 2PC

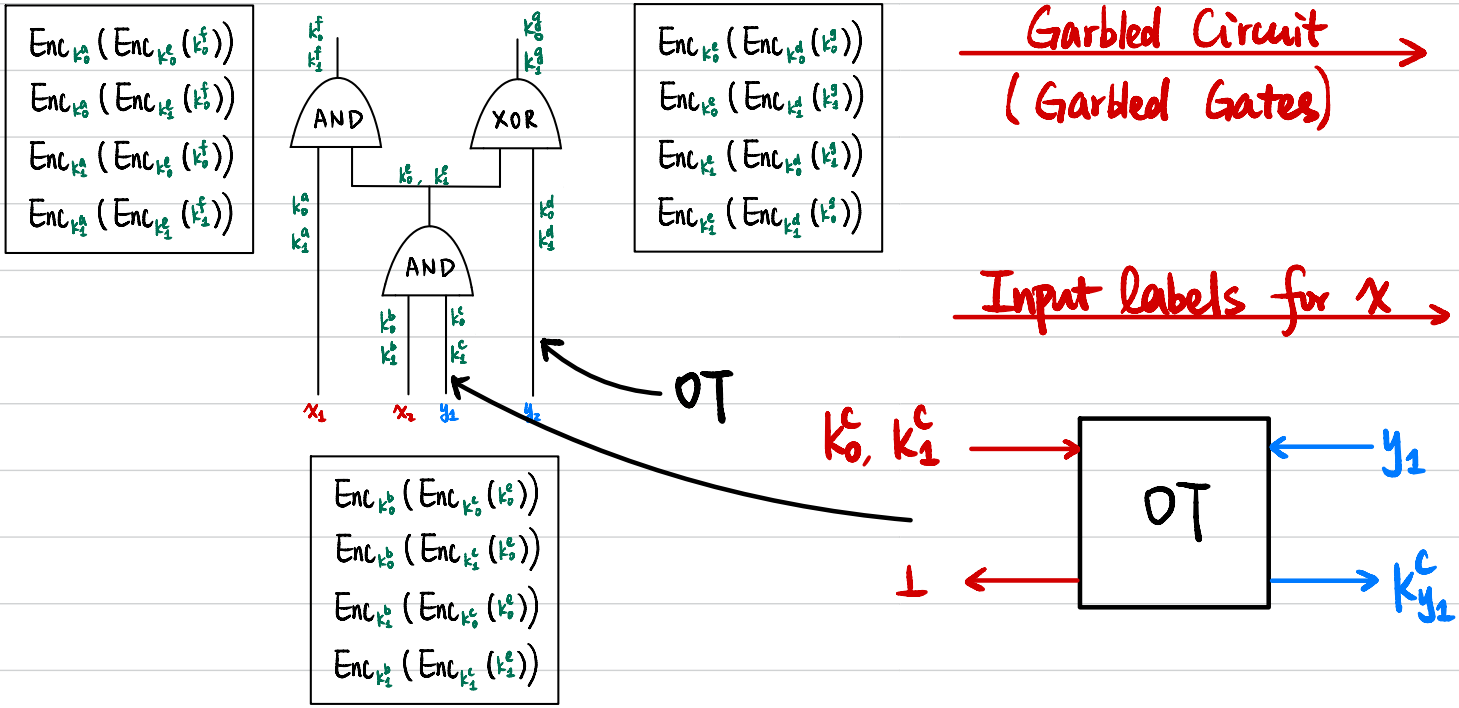
What could go wrong against malicious adversaries?

## 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?

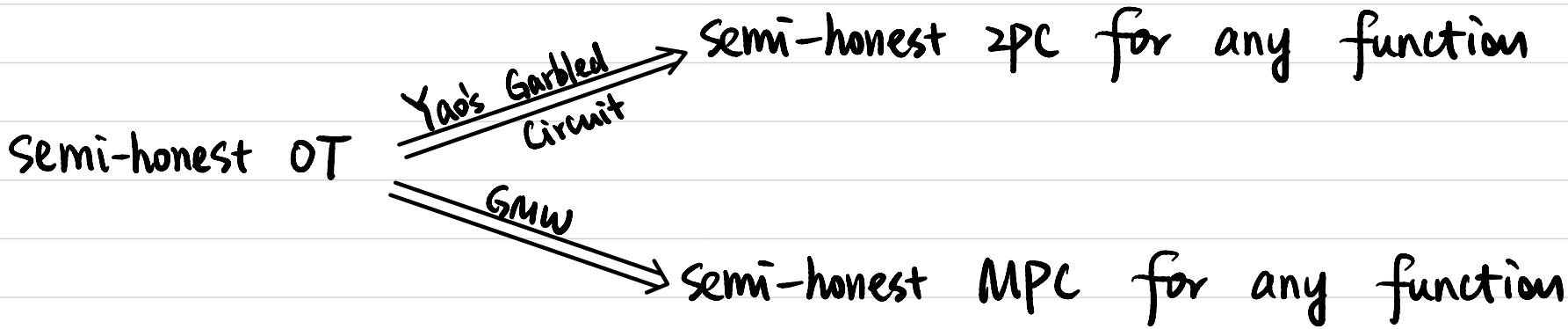
  Evaluator sends $k^f, k^g$ back to garbler

- How to decide which ciphertext to decrypt?

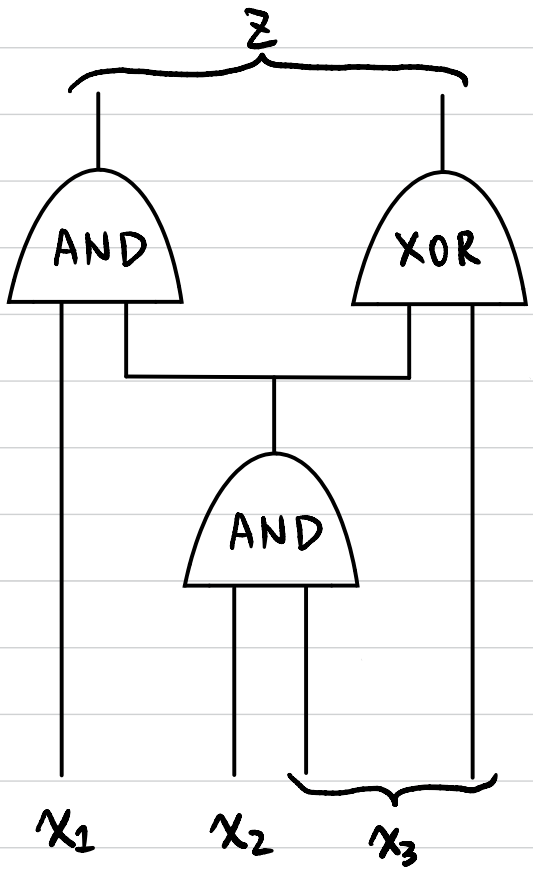
Shuffle

- $\mathrm{Enc}_{k_0^b}(\mathrm{Enc}_{k_0^c}(k_0^e \| \overbrace{0 \cdots 0}^{\lambda}))$ → garbage
- $\mathrm{Enc}_{k_0^b}(\mathrm{Enc}_{k_1^c}(k_0^e \| 0 \cdots 0))$ → $k_0^e \| 0 \cdots 0$
- $\mathrm{Enc}_{k_1^b}(\mathrm{Enc}_{k_0^c}(k_0^e \| 0 \cdots 0))$ → garbage
- $\mathrm{Enc}_{k_1^b}(\mathrm{Enc}_{k_1^c}(k_1^e \| 0 \cdots 0))$ → garbage

$2\lambda$


semi-honest OT
Yao's Garbled Circuit
semi-honest 2PC for any function
GMW
semi-honest MPC for any function

# MPC for any function with $t \leq n-1$ (GMW)




Throughout the protocol, we keep the invariant:

For each wire $w$:

If the value of the wire is $v^w \in \{0,1\}$,

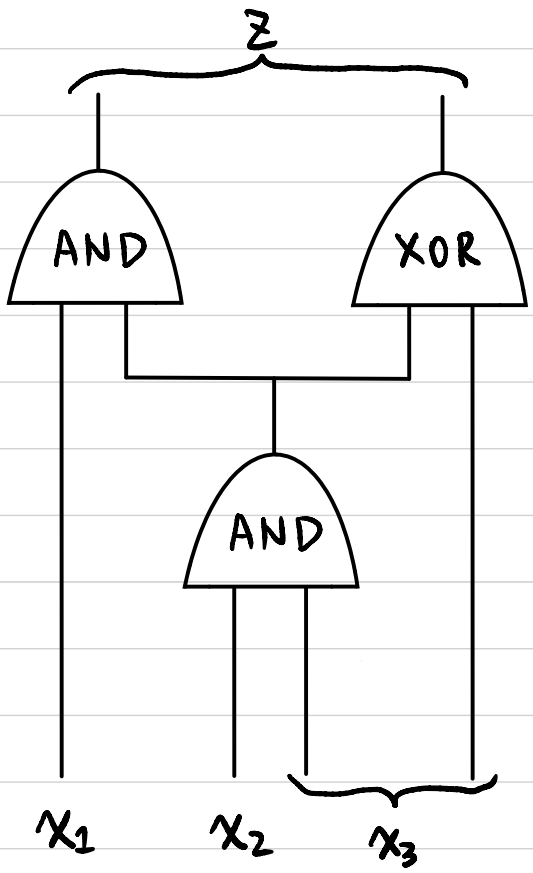
then the $n$ parties hold an additive secret share of $v^w$

Each party $P_i$ holds a random share $v_i^w \in \{0,1\}$ s.t.

$$\bigoplus_{i=1}^{n} v_i^w = v^w$$

Any $(n-1)$ shares information theoretically hide $v^w$.

# MPC for any function with $t \leq n-1$ (GMW)




Each party $P_i$ holds a random share $v_i^w \in \{0,1\}$ s.t. $\bigoplus_{i=1}^{n} v_i^w = v^w$

**Inputs:**

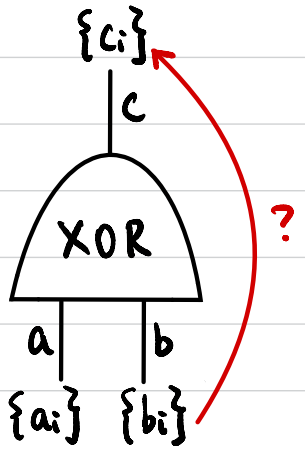
For each input wire $w$:

If it's from party $P_k$ with input value $v^w \in \{0,1\}$.

$P_k$ randomly samples $v_i^w \xleftarrow{\$} \{0,1\}$ s.t. $\bigoplus_{i=1}^{n} v_i^w = v^w$

↳ Sends $v_i^w$ to party $P_i$.

**XOR gates:**

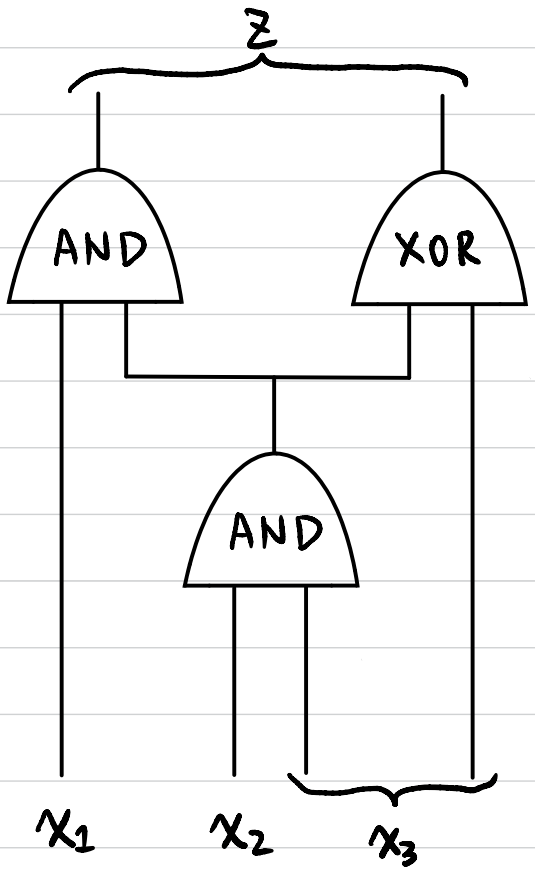



**GIVEN:** $\bigoplus_{i=1}^{n} a_i = a$ $\bigoplus_{i=1}^{n} b_i = b$

**WANT:** $\{c_i\}$ s.t. $\bigoplus_{i=1}^{n} c_i = c = a \oplus b$

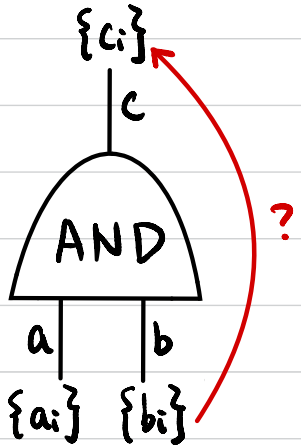
$c_i = a_i \oplus b_i$

# MPC for any function with $t \leq n-1$ (GMW)




Each party $P_i$ holds a random share $v_i^w \in \{0,1\}$ s.t. $\bigoplus_{i=1}^{n} v_i^w = v^w$

**AND gates:**




**GIVEN:** $\bigoplus_{i=1}^{n} a_i = a$ $\bigoplus_{i=1}^{n} b_i = b$

**WANT:** $\{c_i\}$ s.t. $\bigoplus_{i=1}^{n} c_i = c = a \cdot b$

$c_i = ?$

**Outputs:**

For each output wire w:

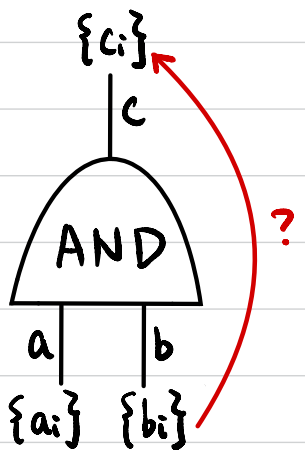
Each party $P_i$ holds a random share $v_i^w \in \{0,1\}$

→ Sends $v_i^w$ to all parties

Each party computes the value $v^w = \bigoplus_{i=1}^{n} v_i^w$

# MPC for any function with $t \leq n-1$ (GMW)

## AND gates:




**GIVEN:** $\bigoplus_{i=1}^{n} a_i = a$ $\quad \bigoplus_{i=1}^{n} b_i = b$

**WANT:** $\{c_i\}$ s.t. $\bigoplus_{i=1}^{n} c_i = c = a \cdot b$

$c_i = ?$

$$a \cdot b = \left(\sum_{i=1}^{n} a_i\right) \cdot \left(\sum_{i=1}^{n} b_i\right) \mod 2$$

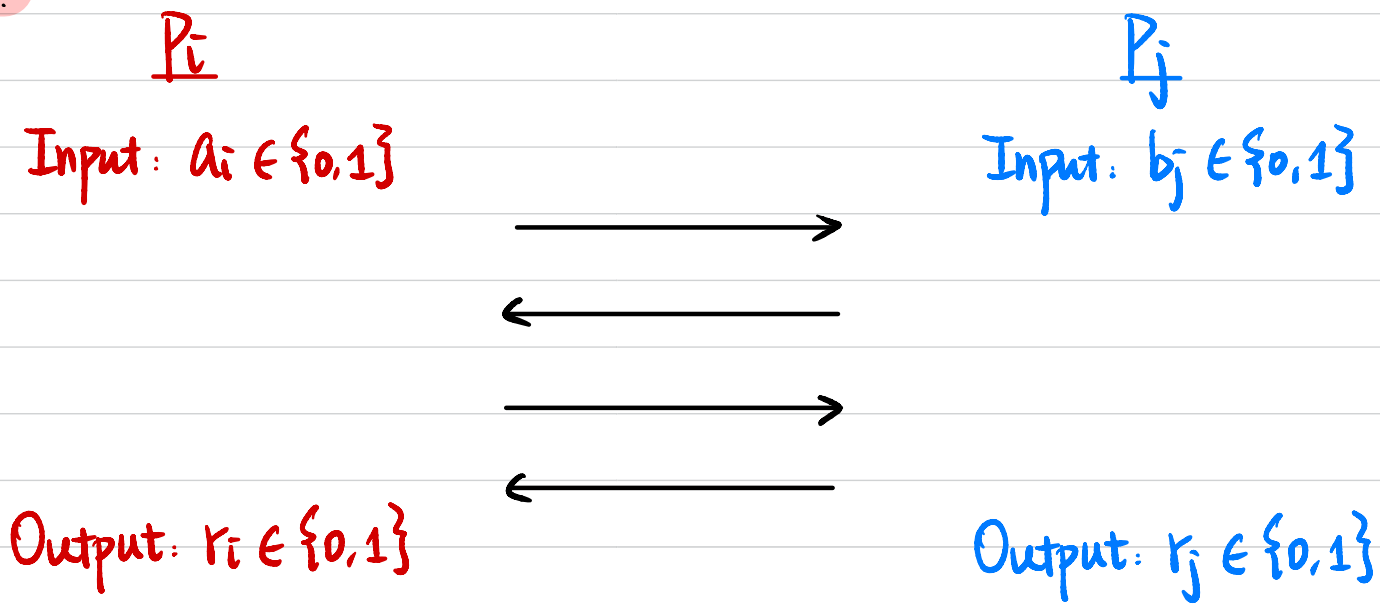
$$= \left(\sum_{i=1}^{n} a_i \cdot b_i\right) + \left(\sum_{i \neq j} a_i \cdot b_j\right) \mod 2$$

$a_i \cdot b_i$: $P_i$ locally

$a_i \cdot b_j$: ?

# MPC for any function with $t \leq n-1$ (GMW)

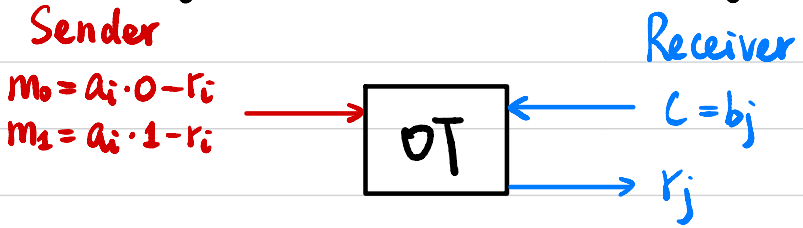
**Reshare:**



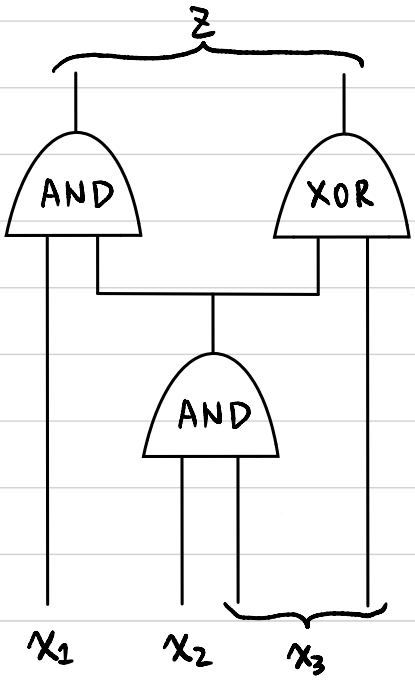

**WANT:** Random $r_i, r_j \in \{0,1\}$ s.t. $r_i + r_j = a_i \cdot b_j \pmod 2$

1) $P_i$ randomly samples $r_i \xleftarrow{\$} \{0,1\}$

2) How to let $P_j$ learn $r_j$ s.t. $r_i + r_j = a_i \cdot b_j \pmod 2$ ?



# MPC for any function with $t \leq n-1$ (GMW)




Each party $P_i$ holds a random share $v_i^w \in \{0,1\}$ s.t. $\bigoplus_{i=1}^{n} v_i^w = v^w$

**Inputs:**

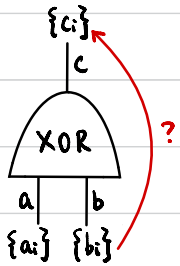
For each input wire $w$:

If it's from party $P_k$ with input value $v^w \in \{0,1\}$.

$P_k$ randomly samples $v_i^w \xleftarrow{\$} \{0,1\}$ s.t. $\bigoplus_{i=1}^{n} v_i^w = v^w$

↳ Sends $v_i^w$ to party $P_i$.

**XOR gates:**

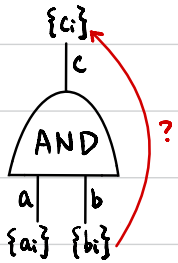



GIVEN: $\bigoplus_{i=1}^{n} a_i = a$ $\bigoplus_{i=1}^{n} b_i = b$

WANT: $\{c_i\}$ s.t. $\bigoplus_{i=1}^{n} c_i = c = a \oplus b$

$c_i = a_i \oplus b_i$

**AND gates:**




GIVEN: $\bigoplus_{i=1}^{n} a_i = a$ $\bigoplus_{i=1}^{n} b_i = b$

WANT: $\{c_i\}$ s.t. $\bigoplus_{i=1}^{n} c_i = c = a \cdot b$

$c_i = ?$

$$a \cdot b = \left(\sum_{i=1}^{n} a_i\right) \cdot \left(\sum_{i=1}^{n} b_i\right) \mod 2$$

$$= \left(\sum_{i=1}^{n} a_i \cdot b_i\right) + \left(\sum_{i \neq j} a_i \cdot b_j\right) \mod 2$$

$P_i$ locally — Reshare

**Outputs:**

For each output wire $w$:

Each party $P_i$ holds a random share $v_i^w \in \{0,1\}$

↳ Sends $v_i^w$ to all parties

Each party computes the value $v^w = \bigoplus_{i=1}^{n} v_i^w$

# MPC for any function with $t \le n-1$ (GMW)

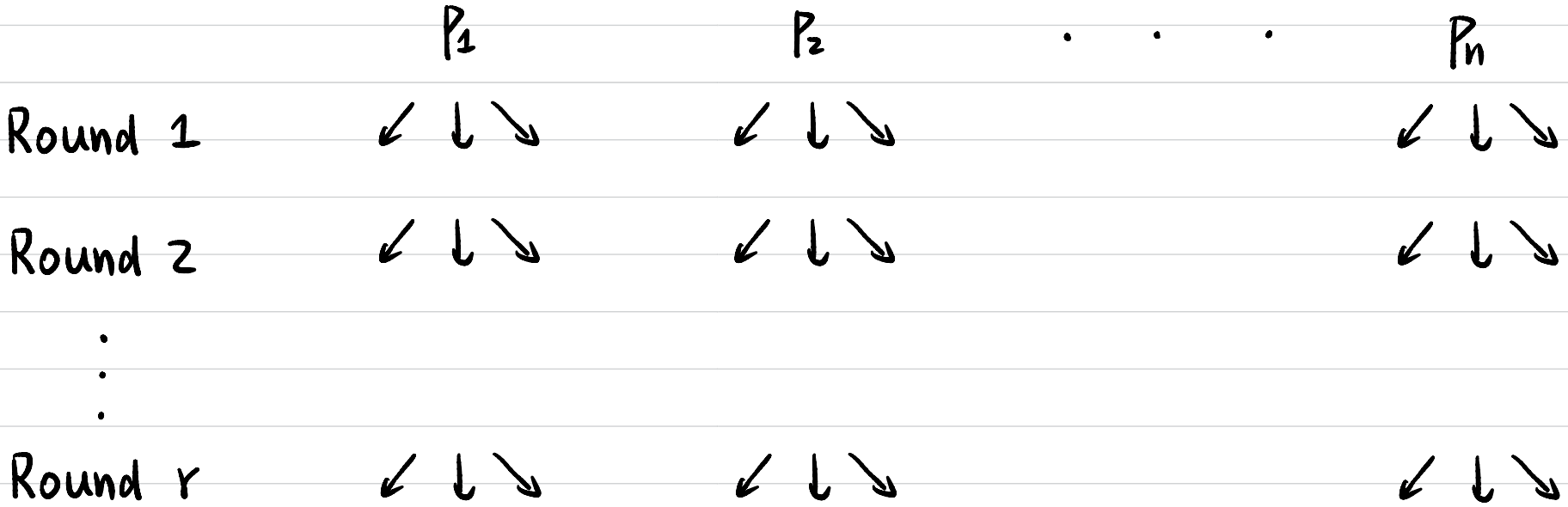
**Computational Complexity?**

$O(\#\text{AND} \cdot n)$ for each party

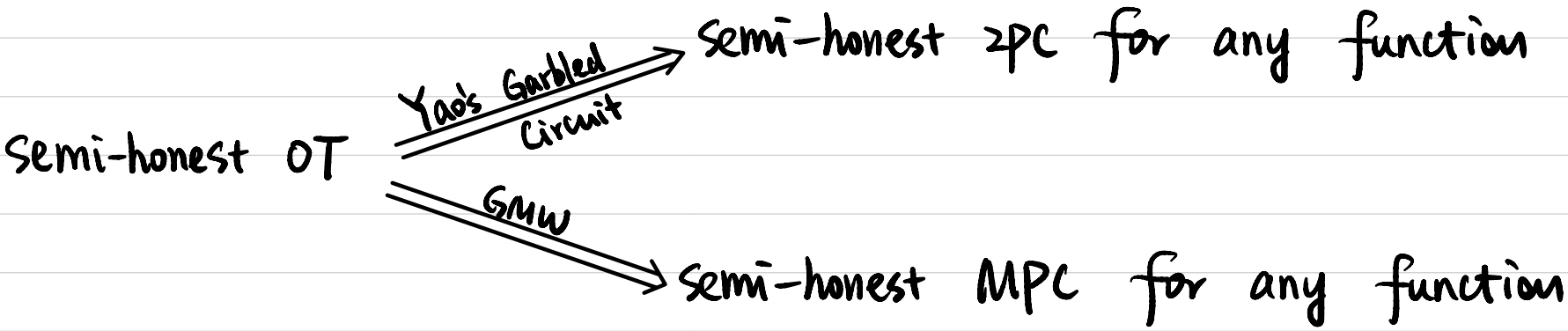
**Communication Complexity?**

$O(\#\text{AND} \cdot n^2)$ in total

**Round Complexity?** $O(\text{depth of AND gates})$




**What could go wrong against malicious adversaries?**




How to compare ?

Yao's Garbled Circuit

+ malicious security lower overhead

+ # OTs : # inputs

Goldreich-Micali-Wigderson

- # OTs : #AND · $n^2$