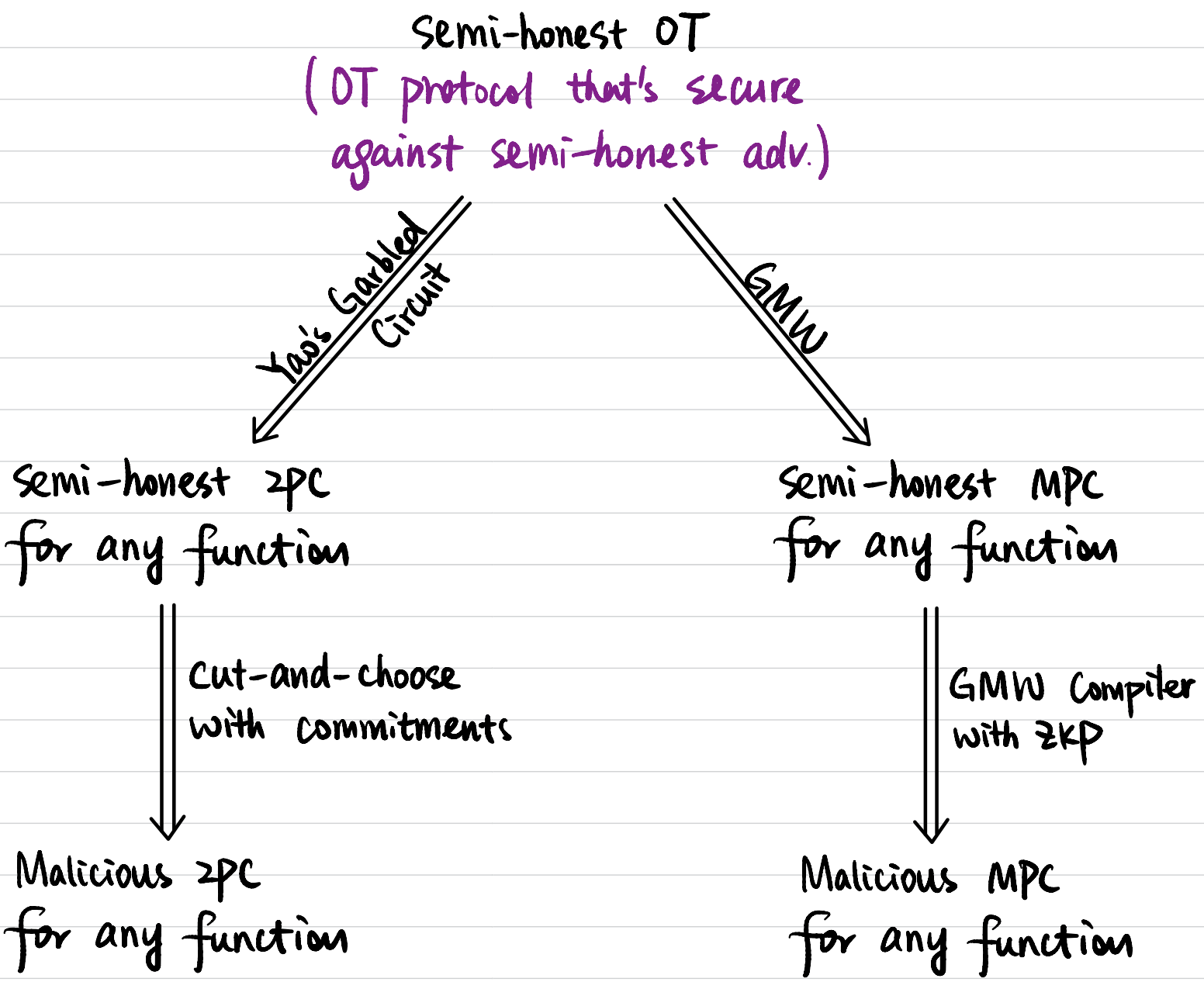
CSCI 1515 Applied Cryptography

This Lecture:

- Yao's Garbled Circuits
- Construction of Oblivious Transfer

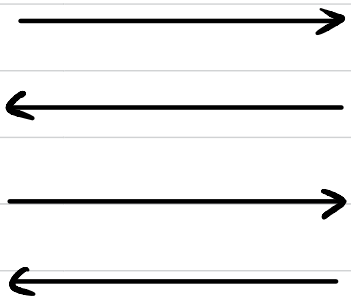

Semi-honest OT
(OT protocol that's secure against semi-honest adv.)
Yao's Garbled Circuit
GMW
Semi-honest 2PC for any function
Semi-honest MPC for any function
Cut-and-choose with commitments
GMW Compiler with ZKP
Malicious 2PC for any function
Malicious MPC for any function

Oblivious Transfer (OT)

Sender

Receiver

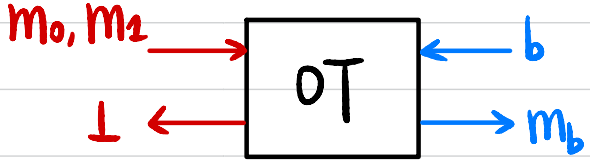
Input: $m_0, m_1 \in \{0,1\}^\ell$



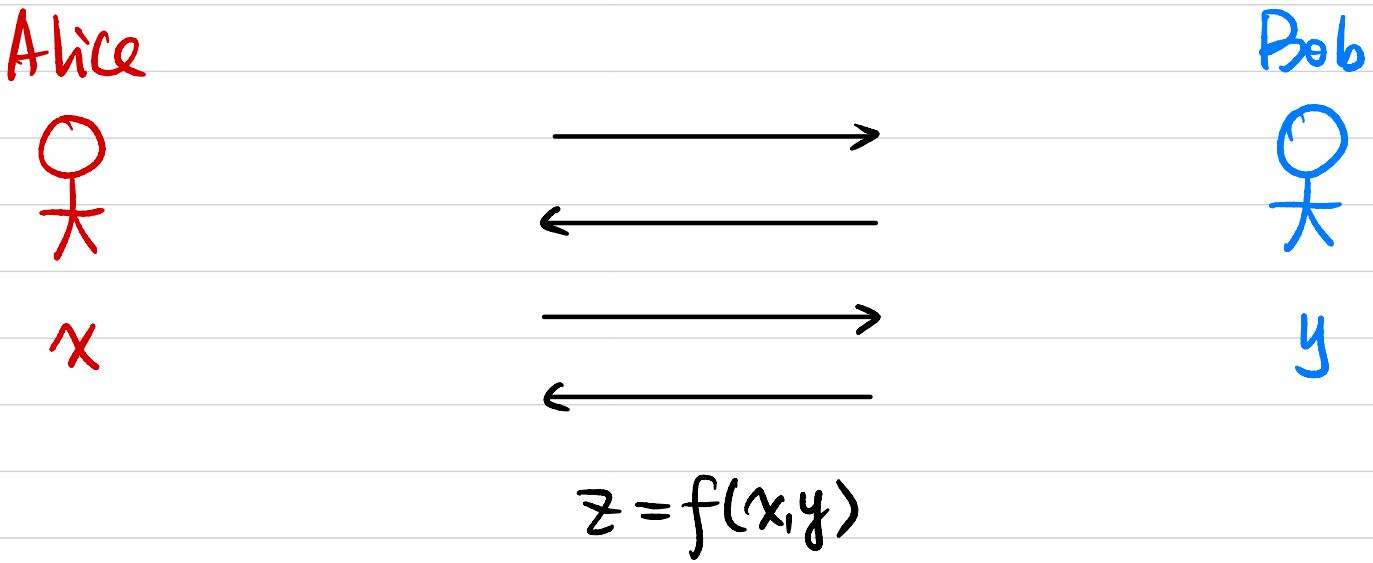
Input: $b \in \{0,1\}$

Output: $\perp$

Output: m_b



Secure Two-Party Computation (2PC)



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:

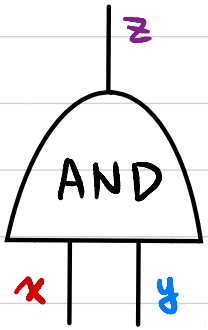
  Can deviate arbitrarily from the protocol description.

Example: Private Dating

Alice

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

$f(x,y) = x \wedge y$



Bob

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

Truth Table:

$x=0$	$y=0$	$\Rightarrow$	$z=0$
$x=0$	$y=1$	$\Rightarrow$	$z=0$
$x=1$	$y=0$	$\Rightarrow$	$z=0$
$x=1$	$y=1$	$\Rightarrow$	$z=1$

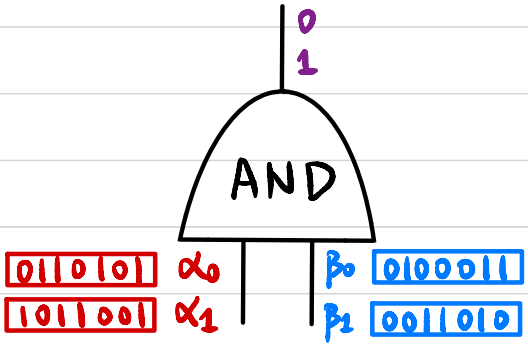
Example: Private Dating

Garbler

Evaluator

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

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



$\alpha_0, \alpha_1, \beta_0, \beta_1 \xleftarrow{\$} \{0,1\}^\lambda$

(labels)

Garbled Gate

$\text{Enc}_{\alpha_0}(\text{Enc}_{\beta_0}(0))$

$\text{Enc}_{\alpha_0}(\text{Enc}_{\beta_1}(0))$

$\text{Enc}_{\alpha_1}(\text{Enc}_{\beta_0}(0))$

$\text{Enc}_{\alpha_1}(\text{Enc}_{\beta_1}(1))$

Garbled Truth Table:

α_0 β_0 $\Rightarrow$ 0

α_0 β_1 $\Rightarrow$ 0

α_1 β_0 $\Rightarrow$ 0

α_1 β_1 $\Rightarrow$ 1

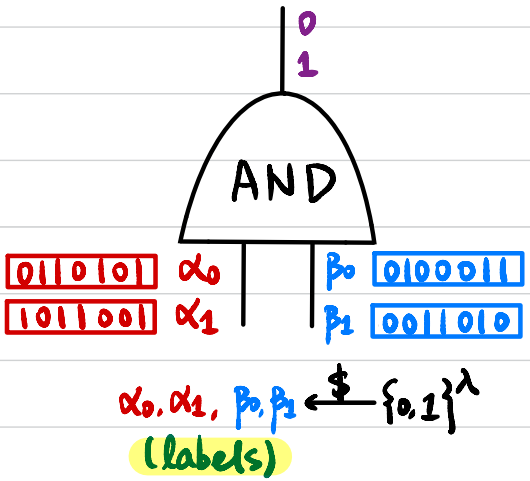
Evalator only gets α_x & β_y

Example: Private Dating

Evalator only gets α_x & β_y

Garbler

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



$\alpha_0, \alpha_1, \beta_0, \beta_1 \xleftarrow{\$} \{0,1\}^\lambda$

(labels)

Garbled Gate

$Enc_{\alpha_0}(Enc_{\beta_0}(0))$

$Enc_{\alpha_0}(Enc_{\beta_1}(0))$

$Enc_{\alpha_1}(Enc_{\beta_0}(0))$

$Enc_{\alpha_1}(Enc_{\beta_1}(1))$

Input label for x: α_x

Evaluator

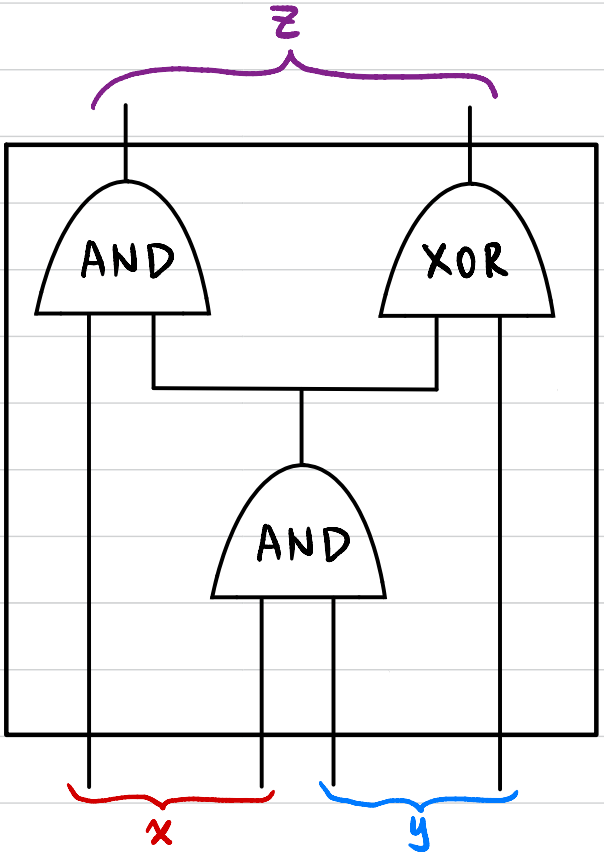
$y \in \{0,1\}$

Input label for y (β_y)?

Arbitrary Function → Represent it as a Boolean circuit

Alice

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



Bob

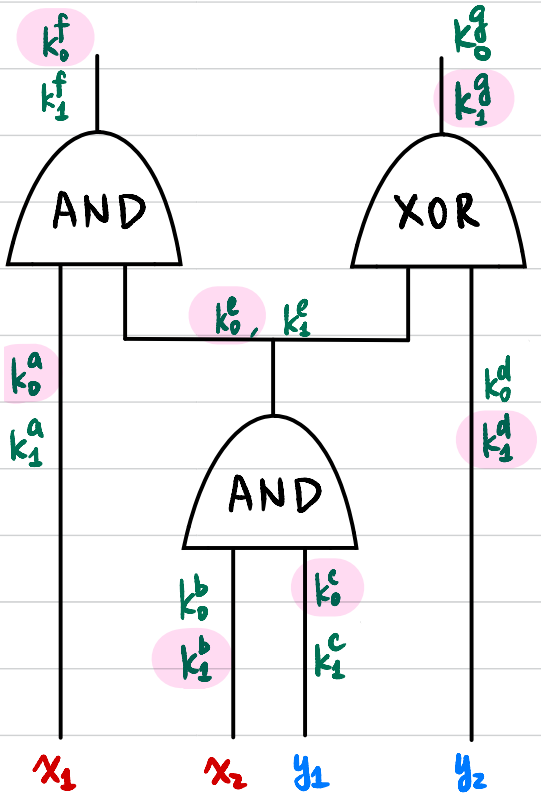
$y \in \{0,1\}^2$

$$z = f(x,y)$$

Yao's Garbled Circuit

Evalator gets one label per wire

$$\mathrm{Enc}_{k_0^a}(\mathrm{Enc}_{k_0^e}(k_0^f))$$
$$\mathrm{Enc}_{k_0^a}(\mathrm{Enc}_{k_1^e}(k_0^f))$$
$$\mathrm{Enc}_{k_1^a}(\mathrm{Enc}_{k_0^e}(k_0^f))$$
$$\mathrm{Enc}_{k_1^a}(\mathrm{Enc}_{k_1^e}(k_1^f))$$



$$\mathrm{Enc}_{k_0^e}(\mathrm{Enc}_{k_0^d}(k_0^g))$$
$$\mathrm{Enc}_{k_0^e}(\mathrm{Enc}_{k_1^d}(k_1^g))$$
$$\mathrm{Enc}_{k_1^e}(\mathrm{Enc}_{k_0^d}(k_1^g))$$
$$\mathrm{Enc}_{k_1^e}(\mathrm{Enc}_{k_1^d}(k_0^g))$$

Each label $\xleftarrow{\$} \{0,1\}^\lambda$

$$\mathrm{Enc}_{k_0^b}(\mathrm{Enc}_{k_0^c}(k_0^e))$$
$$\mathrm{Enc}_{k_0^b}(\mathrm{Enc}_{k_1^c}(k_0^e))$$
$$\mathrm{Enc}_{k_1^b}(\mathrm{Enc}_{k_0^c}(k_0^e))$$
$$\mathrm{Enc}_{k_1^b}(\mathrm{Enc}_{k_1^c}(k_1^e))$$

Secure 2PC

Evalator gets one label per wire

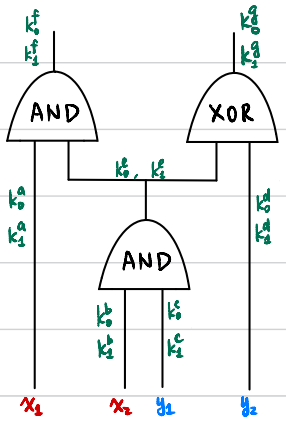
Alice (Garbler)

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

Bob (Evaluator)

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

$\mathrm{Enc}_{k_0^f}(\mathrm{Enc}_{k_0^e}(k_0^f))$
$\mathrm{Enc}_{k_0^f}(\mathrm{Enc}_{k_1^e}(k_0^f))$
$\mathrm{Enc}_{k_1^f}(\mathrm{Enc}_{k_0^e}(k_0^f))$
$\mathrm{Enc}_{k_1^f}(\mathrm{Enc}_{k_1^e}(k_1^f))$



$\mathrm{Enc}_{k_0^g}(\mathrm{Enc}_{k_0^d}(k_0^g))$
$\mathrm{Enc}_{k_0^g}(\mathrm{Enc}_{k_1^d}(k_1^g))$
$\mathrm{Enc}_{k_1^g}(\mathrm{Enc}_{k_0^d}(k_0^g))$
$\mathrm{Enc}_{k_1^g}(\mathrm{Enc}_{k_1^d}(k_0^g))$

Garbled Circuit
(Garbled Gates)

Input labels for x

Input labels for y?

$\mathrm{Enc}_{k_0^b}(\mathrm{Enc}_{k_0^c}(k_0^e))$
$\mathrm{Enc}_{k_0^b}(\mathrm{Enc}_{k_1^c}(k_0^e))$
$\mathrm{Enc}_{k_1^b}(\mathrm{Enc}_{k_0^c}(k_0^e))$
$\mathrm{Enc}_{k_1^b}(\mathrm{Enc}_{k_1^c}(k_1^e))$

- Output labels?
  - ① $k_0^f \to 0$ $k_1^f \to 1$
  - ② Bob sends output labels back to Alice
- How to decide which ciphertext to decrypt?

Oblivious Transfer (OT)

Sender

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

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

$A = g^a$ →

$k_0 := H(B^a)$

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

$ct_0 \leftarrow Enc_{k_0}(m_0)$

$ct_1 \leftarrow Enc_{k_1}(m_1)$ →

Receiver

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

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

← $B = g^b \cdot A^c$

Output:

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

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

Why is it correct?

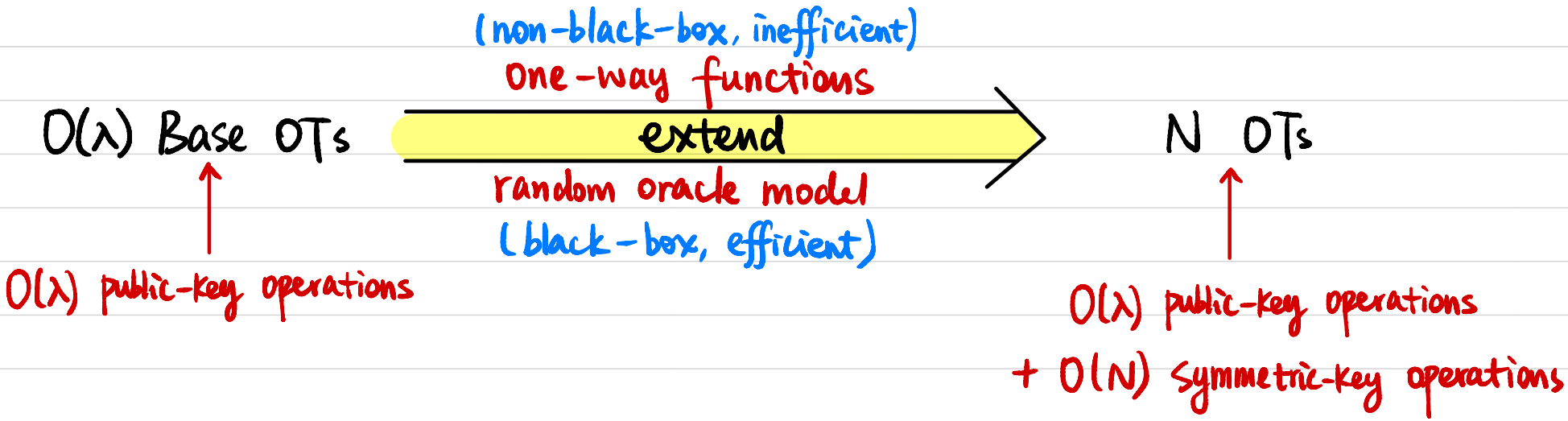
Why is it secure against semi-honest Sender?

Why is it secure against semi-honest Receiver?

OT Extension

Can we construct OT from symmetric-key primitives only?

Unlikely! (theoretical impossibility)



Putting it All Together: Semi-Honest 2PC

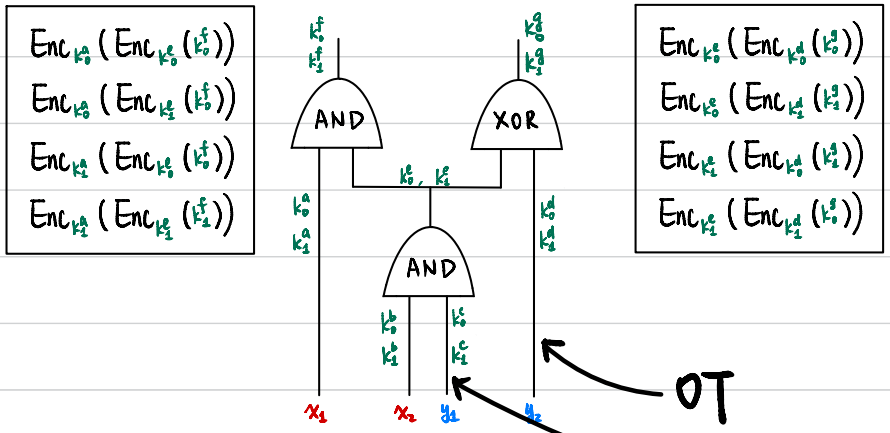
What could go wrong against malicious adversaries?

Alice (Garbler)

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

Bob (Evaluator)

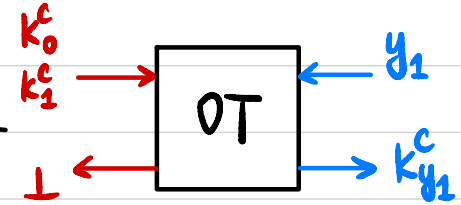
$y \in \{0,1\}^2$



Garbled Circuit (Garbled Gates)

Input labels for x

Input labels for y: OT



Output labels k^f, k^g

Output z

Shuffle

$H(k_0^b \| k_0^c) \oplus (k_0^e \| 0 \cdots 0) \rightarrow$ garbage

$H(k_0^b \| k_1^c) \oplus (k_0^e \| 0 \cdots 0) \rightarrow$ garbage

$H(k_1^b \| k_0^c) \oplus (k_0^e \| 0 \cdots 0) \rightarrow k_0^e \| 0 \cdots 0$

$H(k_1^b \| k_1^c) \oplus (k_1^e \| 0 \cdots 0) \rightarrow$ garbage

256 128 128

Communication cost?

Computation cost?