# SEC595
# Encrypted Computing

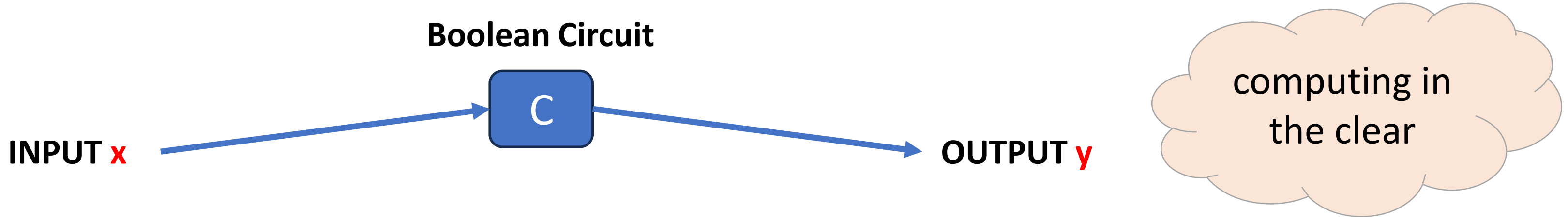
## Lecture 18: Garbled Circuit

***Acknowledgment**: Content is based on the slides developed by Dr. Ahmed Almulhem in COE426*

# Garbled Circuit

- A “**garbled**” version of a Boolean circuit
  - Also known as **encrypted** circuit, or **scrambled** circuit

- Overview

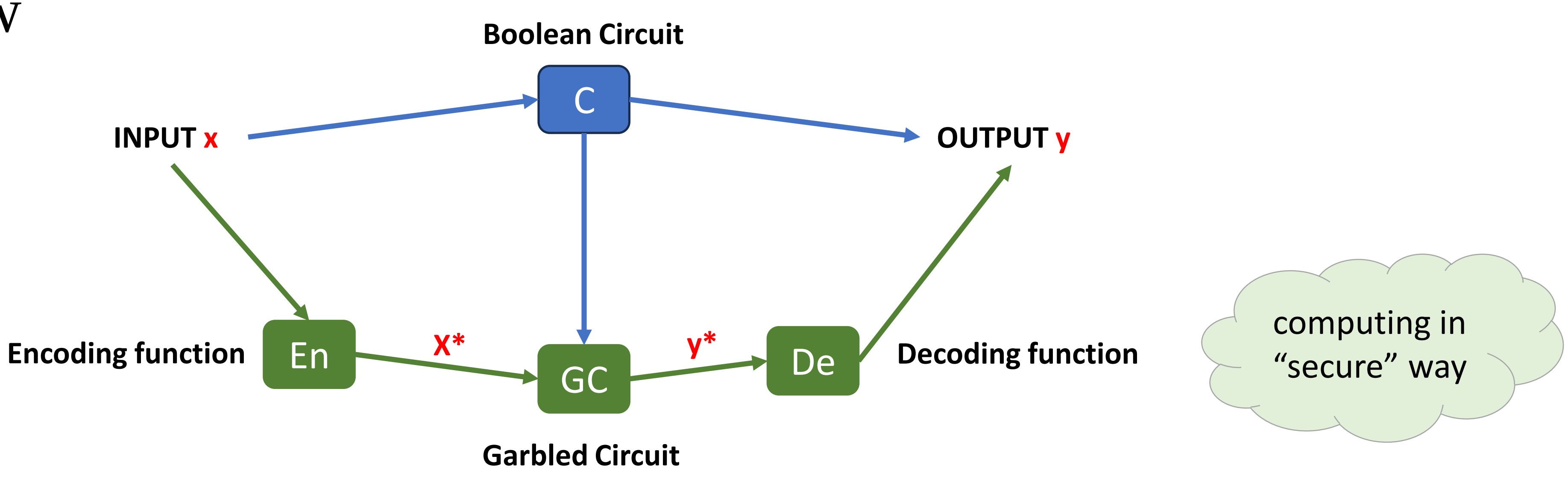



Bellare, Mihir, Viet Tung Hoang, and Phillip Rogaway. "Foundations of garbled circuits." *Proceedings of the 2012 ACM conference on Computer and communications security*. 2012.

# Garbled Circuit

- A “**garbled**” version of a Boolean circuit
  - Also known as **encrypted** circuit, and **scrambled** circuit

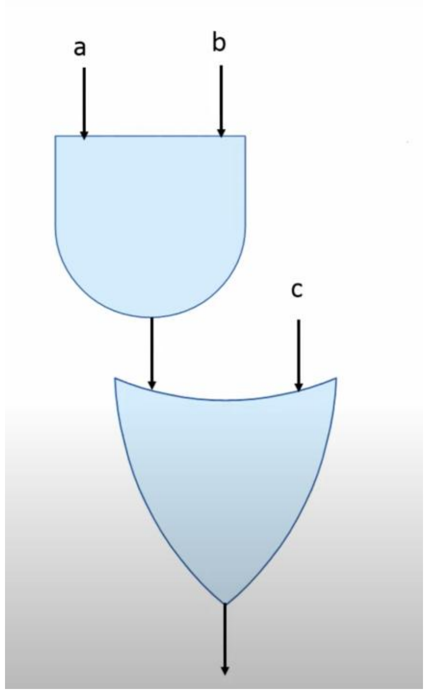
- Overview




Bellare, Mihir, Viet Tung Hoang, and Phillip Rogaway. "Foundations of garbled circuits." *Proceedings of the 2012 ACM conference on Computer and communications security*. 2012.

# Yao’s Garbling Scheme

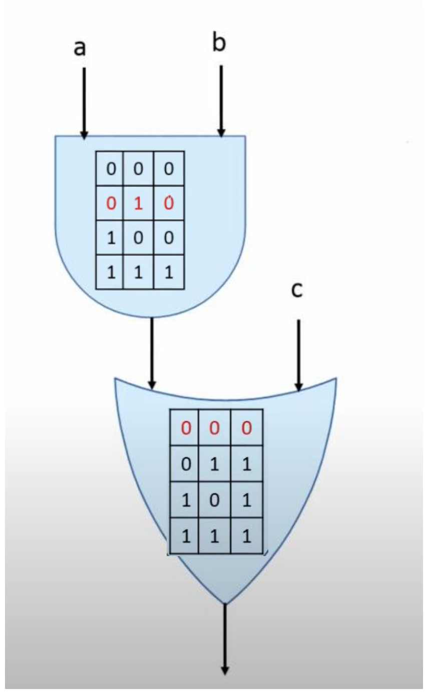
Computation in Clear




src: Secure Computation (Online Course)

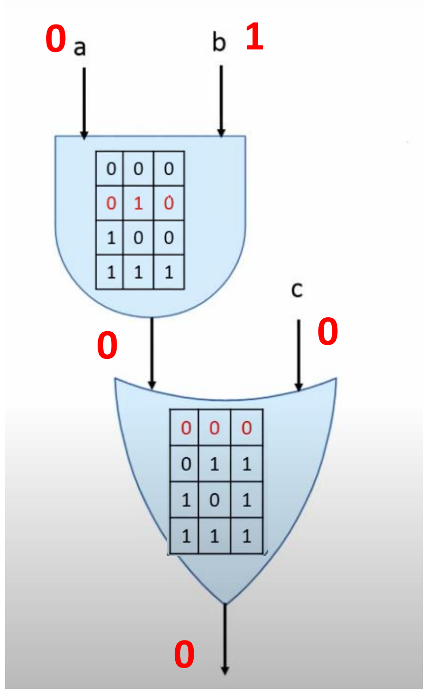
# Yao’s Garbling Scheme

Computation in Clear



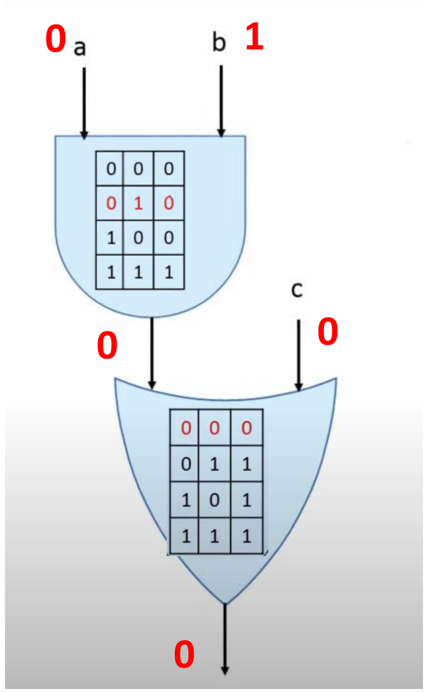
# Yao's Garbling Scheme

Computation in Clear

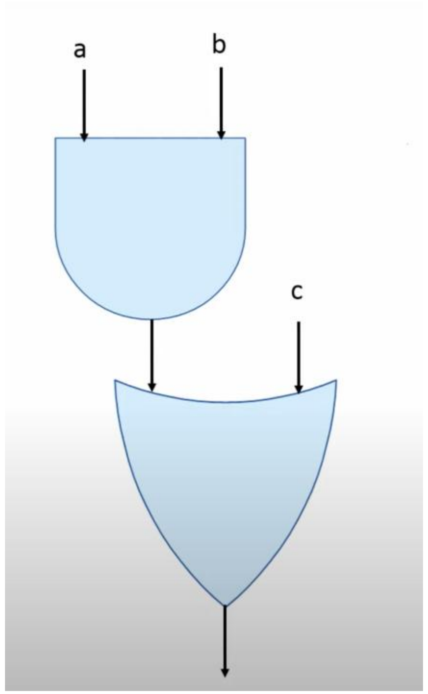


# Yao’s Garbling Scheme

Computation in Clear

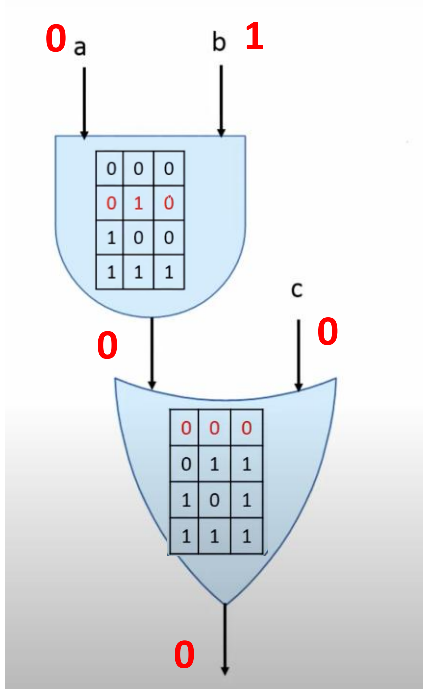



Garbled Computation



# Yao’s Garbling Scheme

Computation in Clear



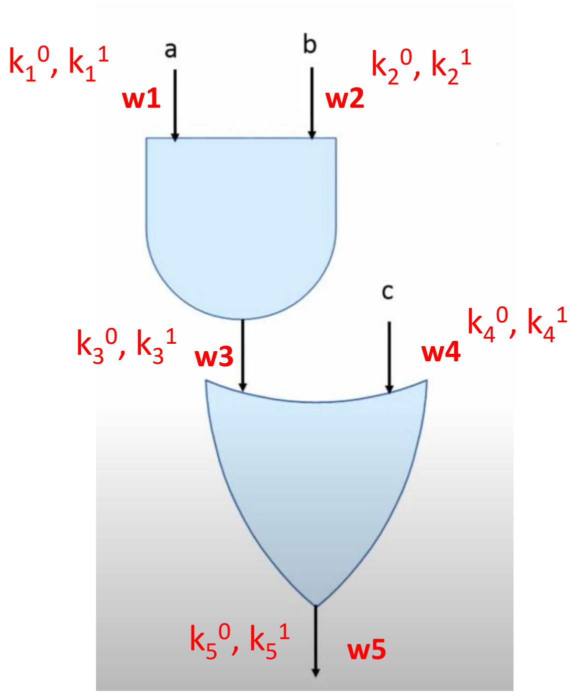

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**Garbling Phase:**

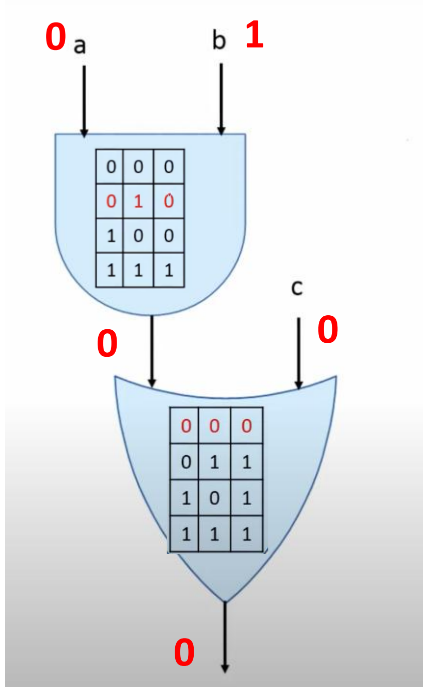
1. **Garbling Wires:**
Assign two keys for each wire – e.g w1 -> ($k_1^0$, $k_1^1$)

Garbled Computation



# Yao’s Garbling Scheme

Computation in Clear

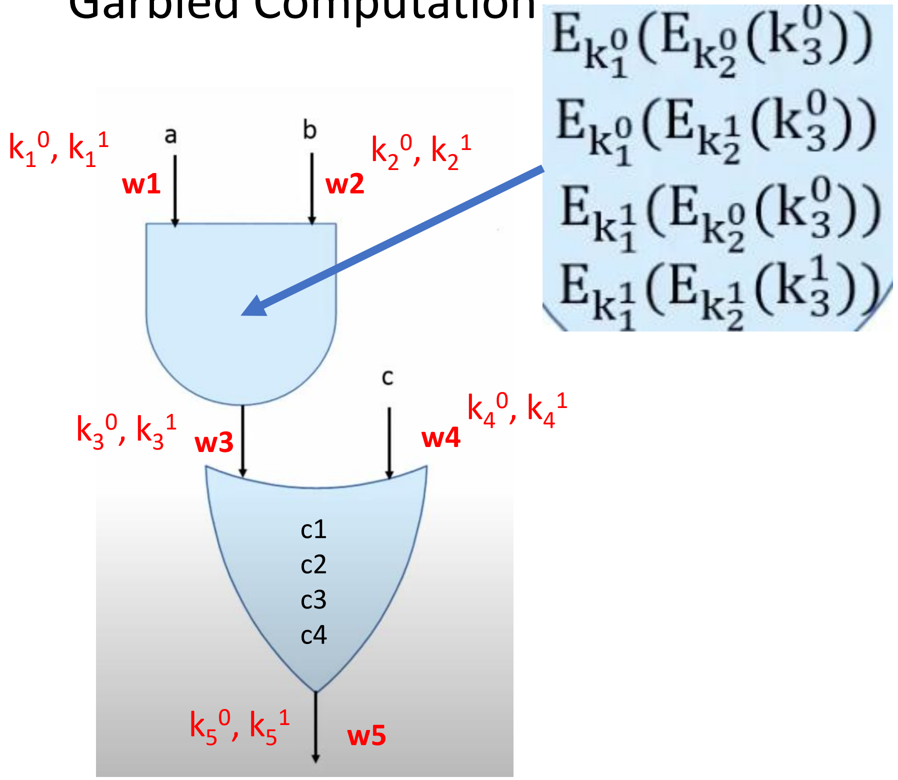



**<u>Garbling Phase:</u>**

1. **Garbling Wires:**
Assign two keys for each wire – e.g w1 -> ($k_1^0$, $k_1^1$)

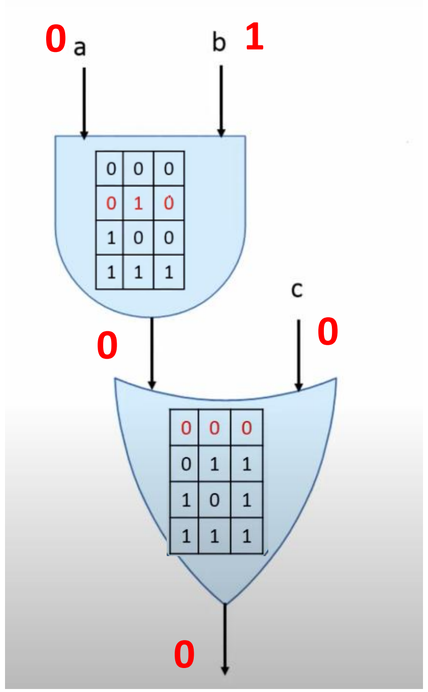
2. **Garbling Gates:**
- Use double encryption
- Shuffle rows

Garbled Computation



# Yao's Garbling Scheme

Computation in Clear




**<u>Garbling Phase:</u>**

1. **Garbling Wires:**
Assign two keys for each wire – e.g w1 -> ($k_1^0$, $k_1^1$)

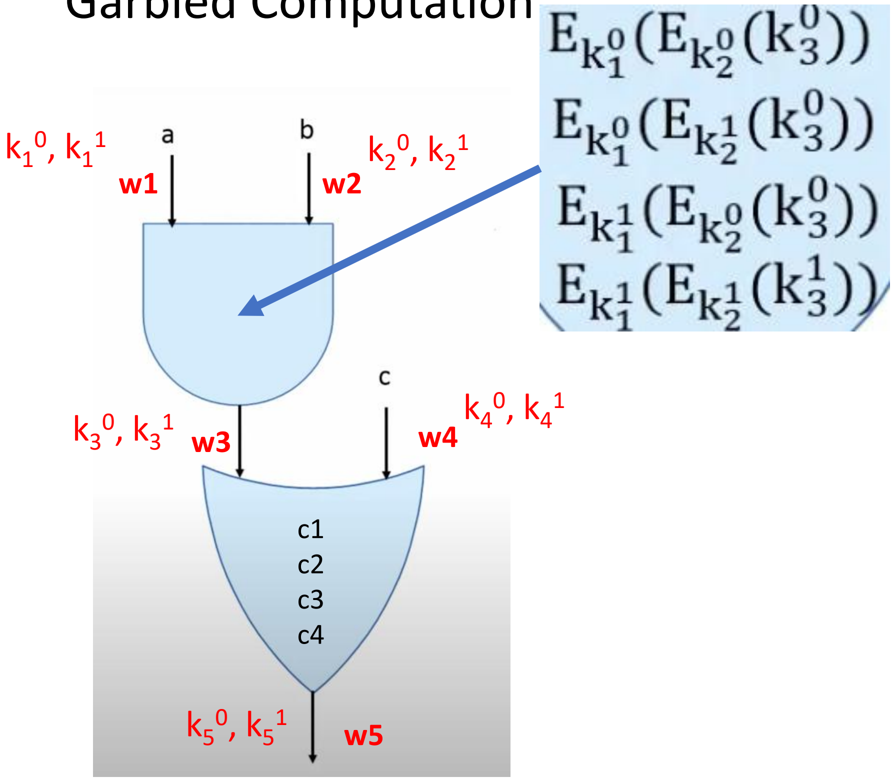
2. **Garbling Gates:**
 - Use double encryption
 - Shuffle rows

**<u>Evaluation Phase:</u>**

1. Use <u>unlabeled</u> keys corresponding to actual inputs

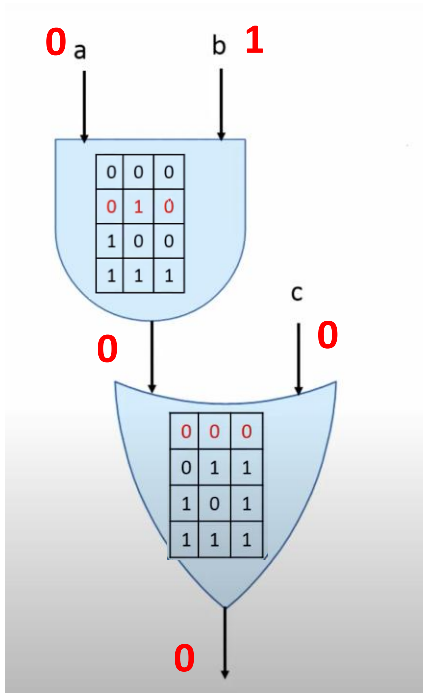
2. Evaluate in topological order

Garbled Computation



# Yao's Garbling Scheme

Computation in Clear




**Garbling Phase:**

1. **Garbling Wires:**
Assign two keys for each wire – e.g w1 -> ($k_1^0$, $k_1^1$)

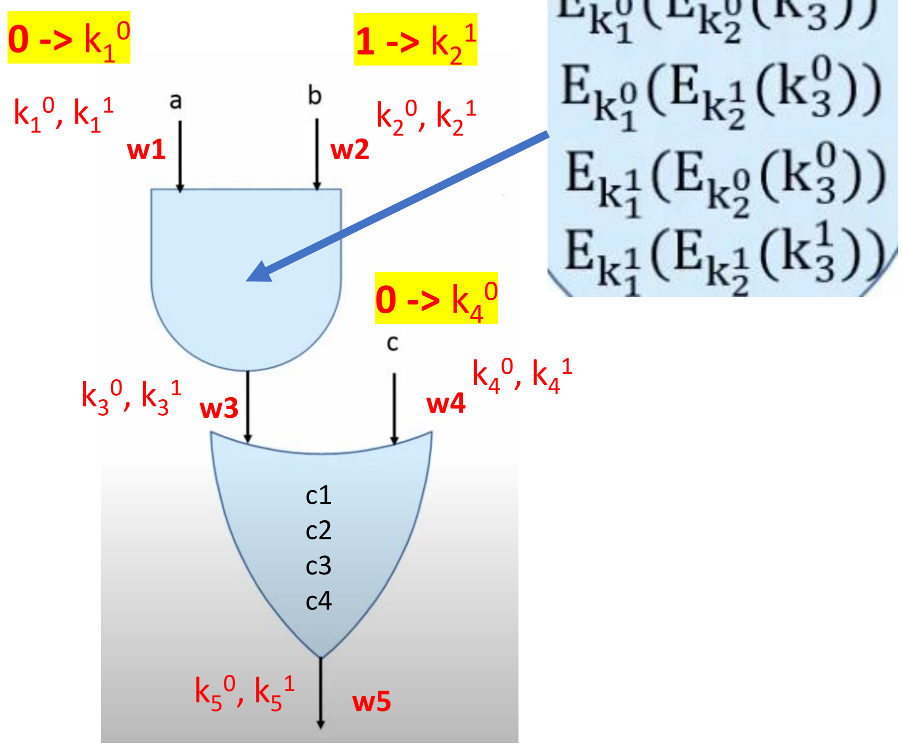
2. **Garbling Gates:**
 **-** Use double encryption
**-** Shuffle rows

**Evaluation Phase:**

1. Use unlabeled keys corresponding to actual inputs

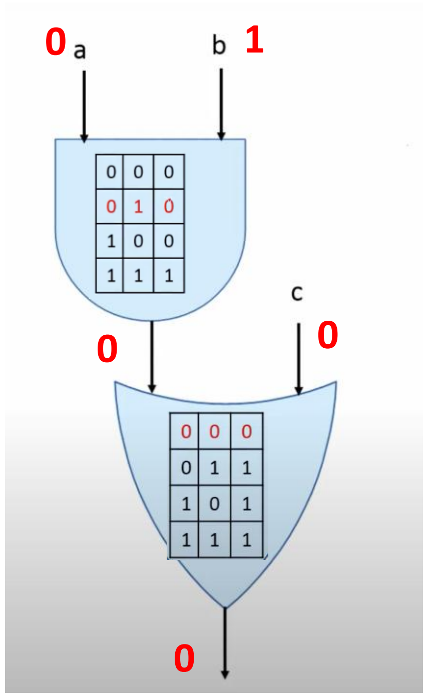
2. Evaluate in topological order

Garbled Computation



# Yao's Garbling Scheme

Computation in Clear




**<u>Garbling Phase:</u>**

1. **Garbling Wires:**
Assign two keys for each wire – e.g w1 -> ($k_1^0$, $k_1^1$)

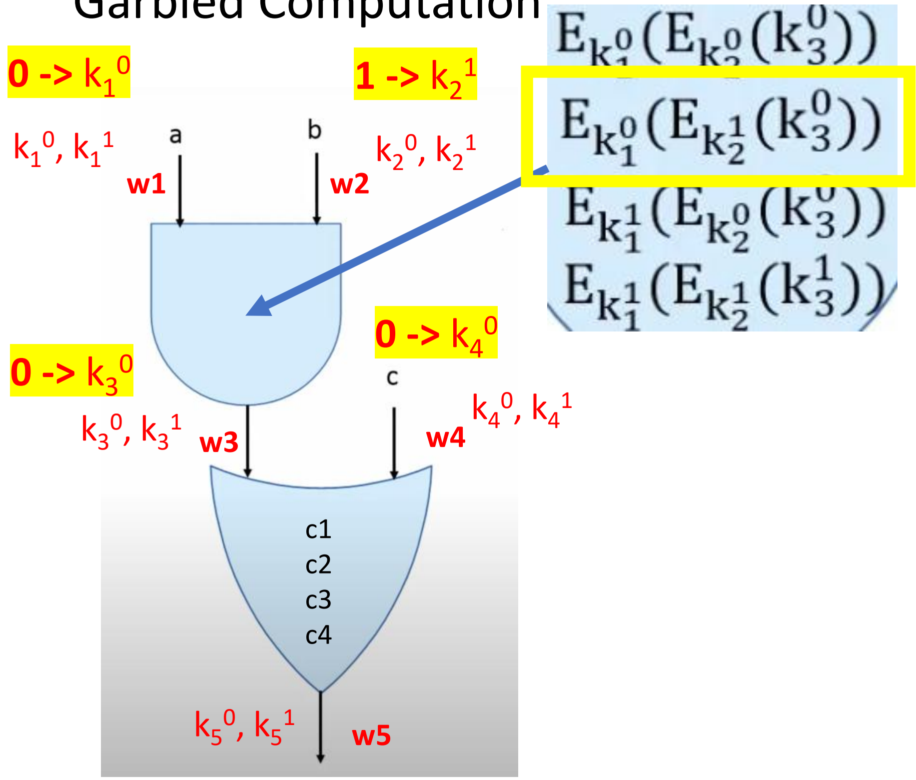
2. **Garbling Gates:**
 **-** Use double encryption
**-** Shuffle rows

**<u>Evaluation Phase:</u>**

1. Use <u>unlabeled</u> keys corresponding to actual inputs

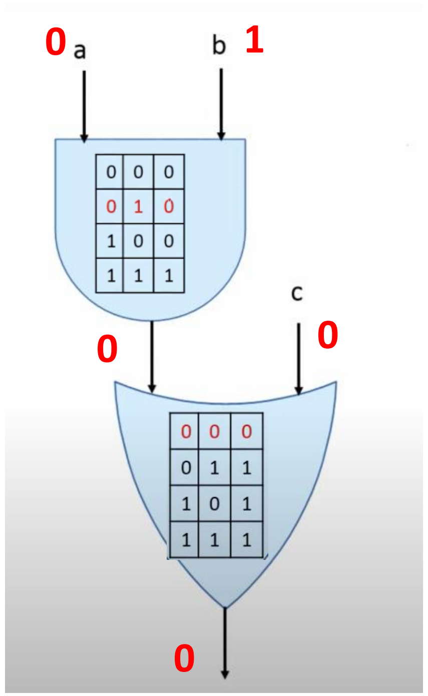
2. Evaluate in topological order

Garbled Computation



# Yao’s Garbling Scheme

Computation in Clear




**Garbling Phase:**

1. **Garbling Wires:**
Assign two keys for each wire – e.g w1 -> ($k_1^0$, $k_1^1$)

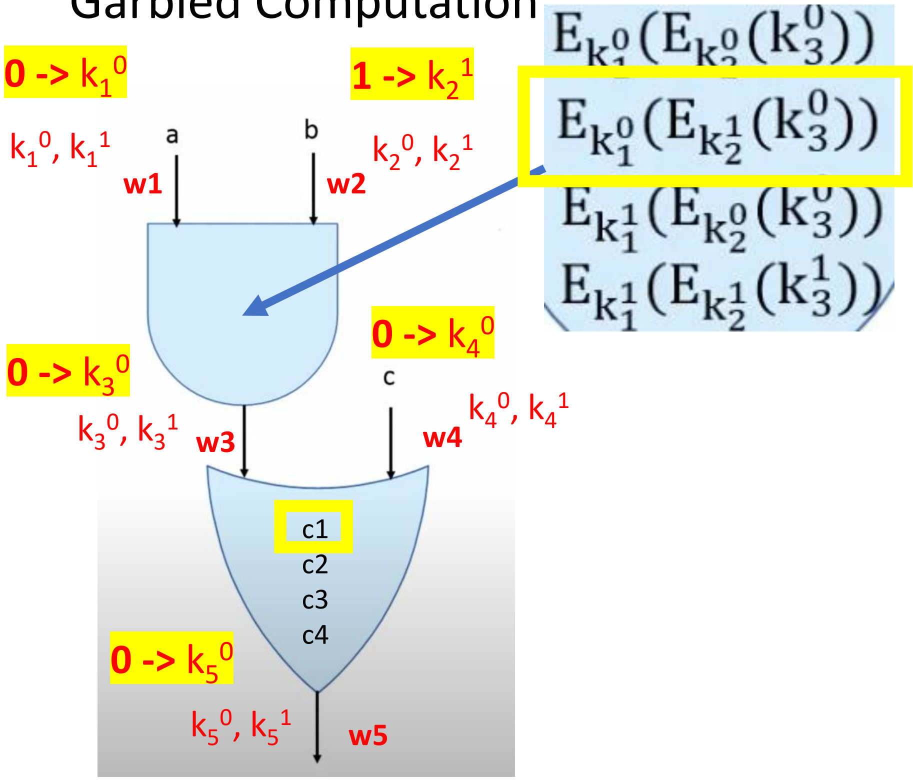
2. **Garbling Gates:**
- Use double encryption
- Shuffle rows

**Evaluation Phase:**

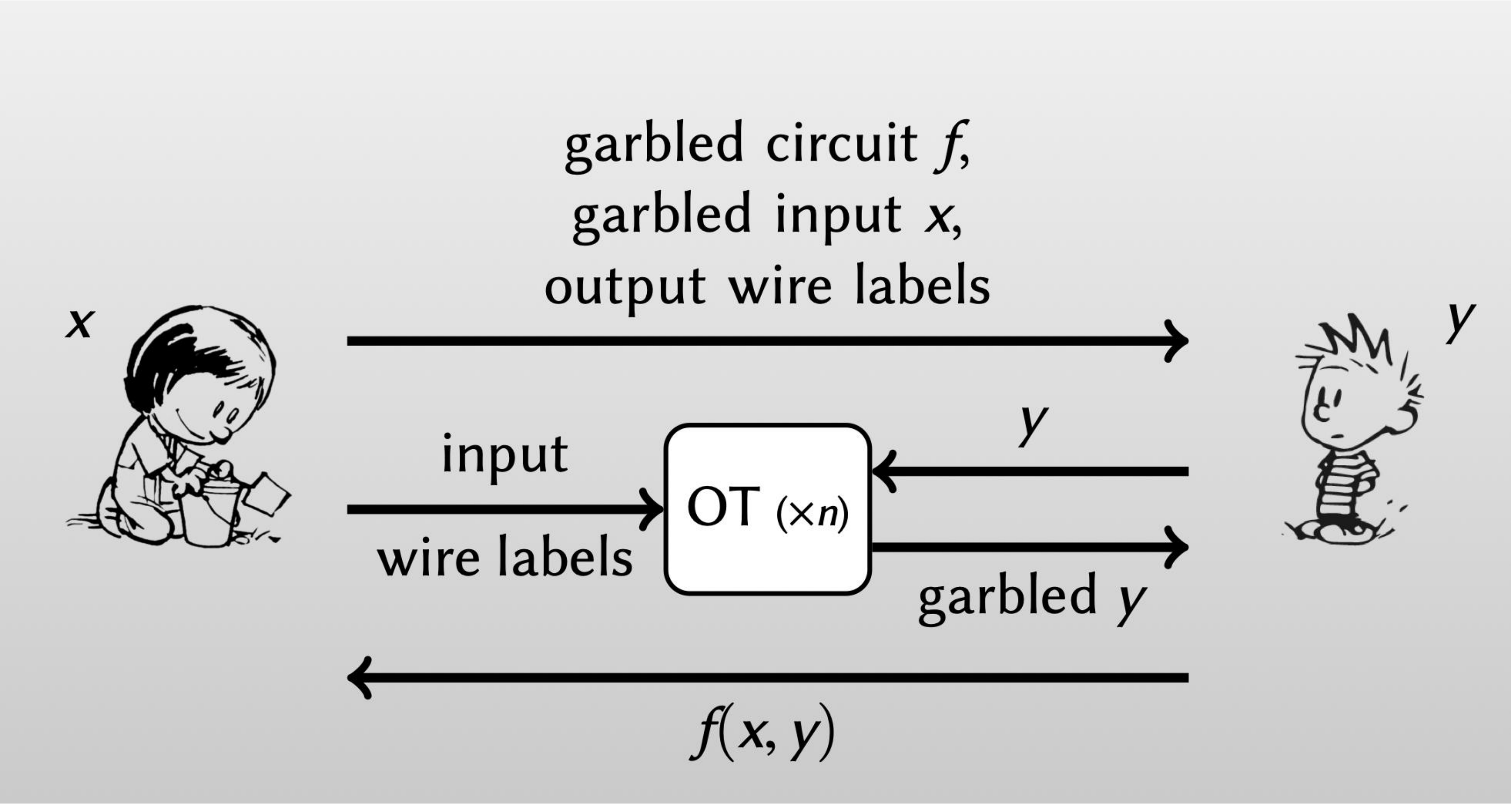
1. Use unlabeled keys corresponding to actual inputs

2. Evaluate in topological order

Garbled Computation

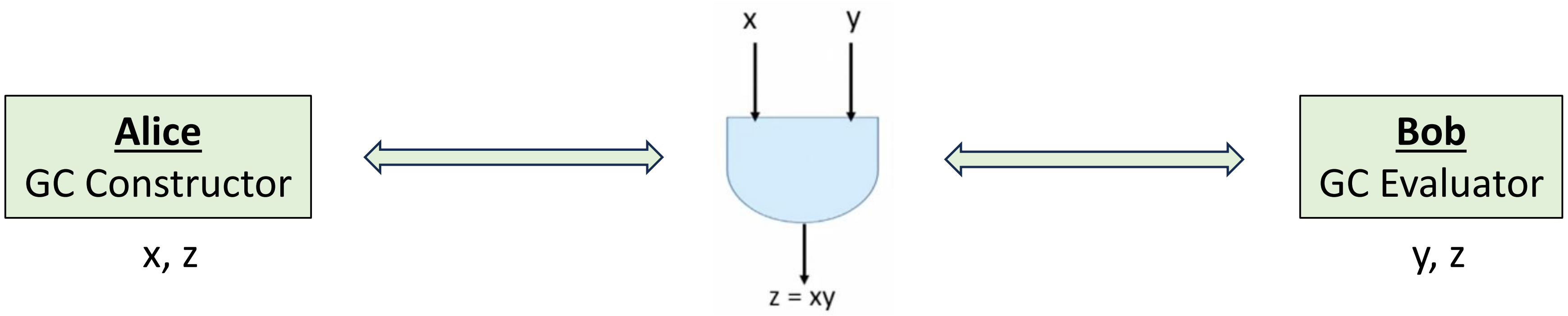


# Yao’s 2-PC Protocol




Src: https://web.engr.oregonstate.edu/~rosulekm/cryptabit/1-overview.pdf

# Yao’s 2-PC Protocol

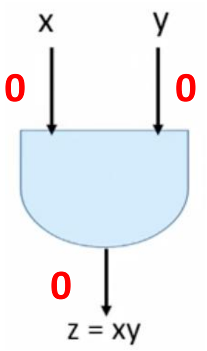



src: [Secure Computation (Online Course)]

# Yao's 2-PC Protocol

**Alice**
GC Constructor

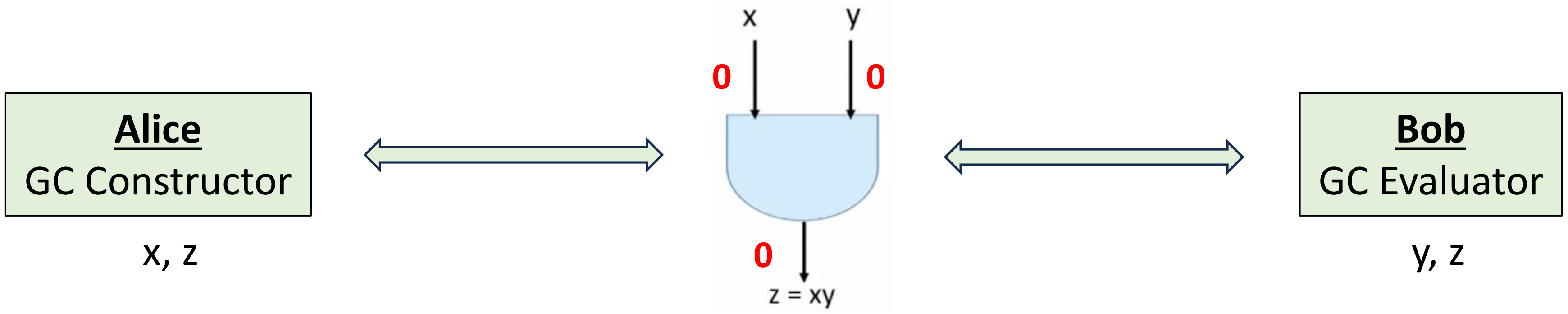
x, z

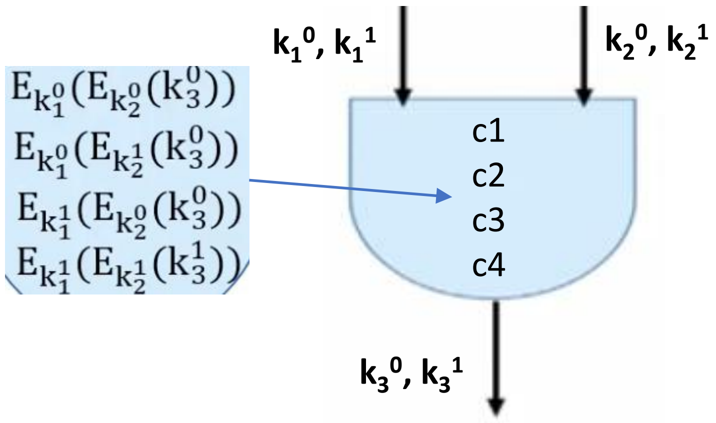



**Bob**
GC Evaluator

y, z

# Yao's 2-PC Protocol



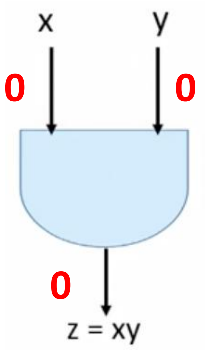




- 6 random keys
- 4 ciphertexts – generated through double-encryption

# Yao’s 2-PC Protocol

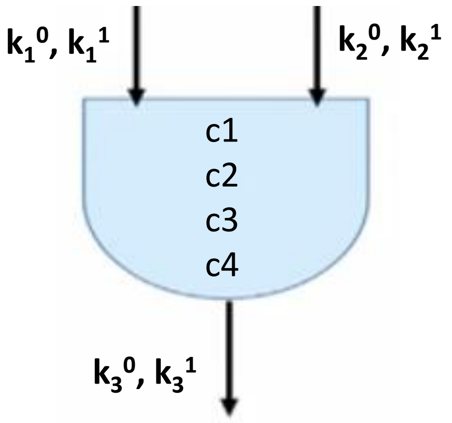
**Alice**
GC Constructor

x, z




**Bob**
GC Evaluator

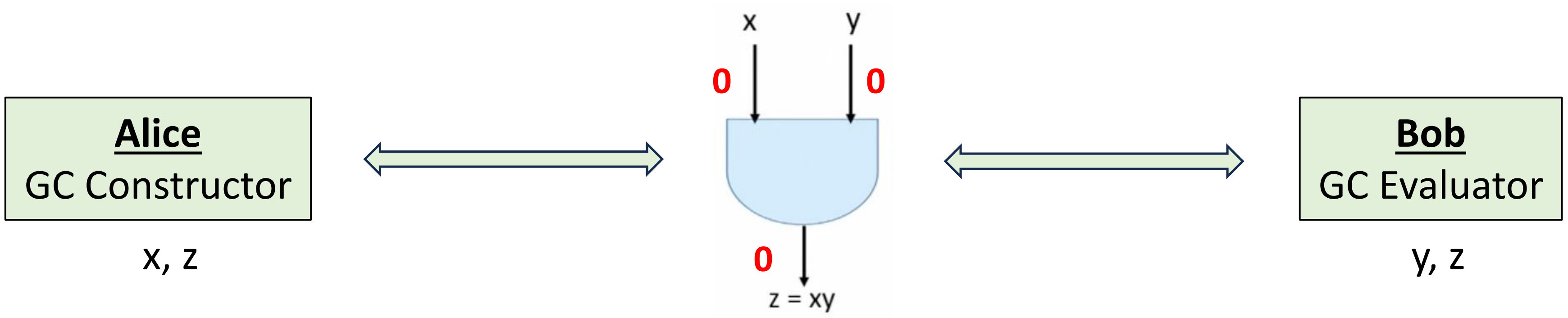
y, z

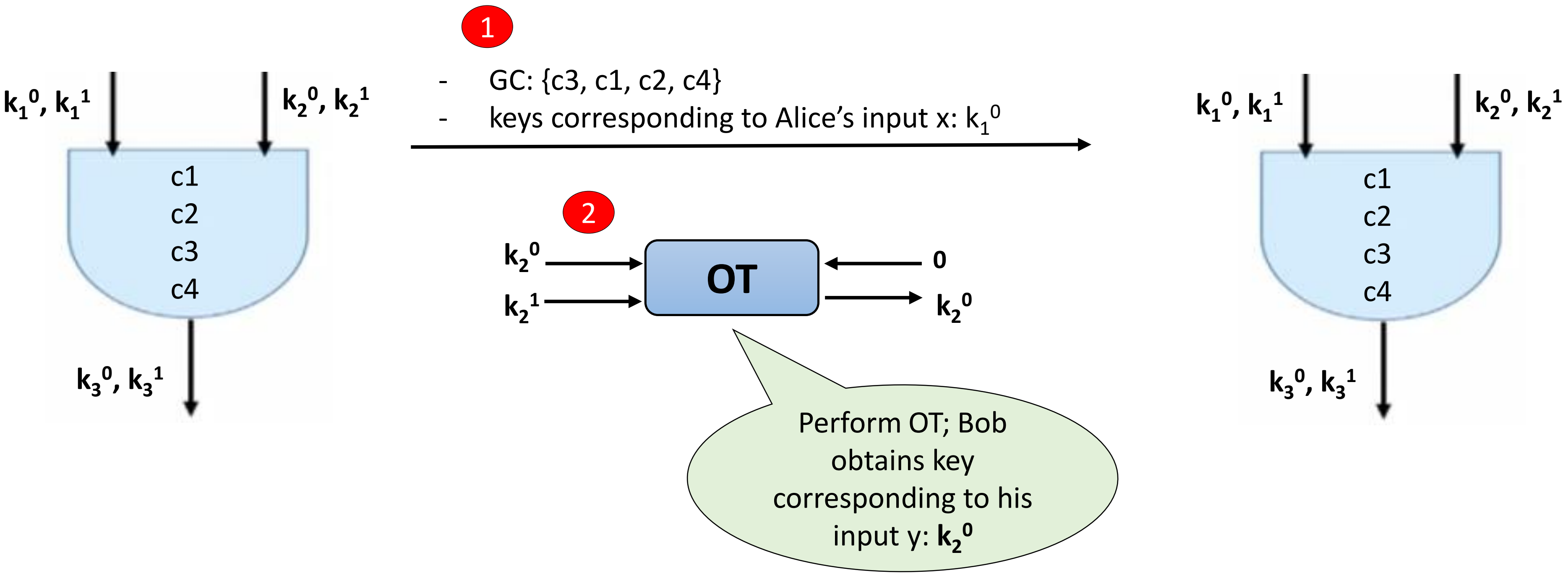



1

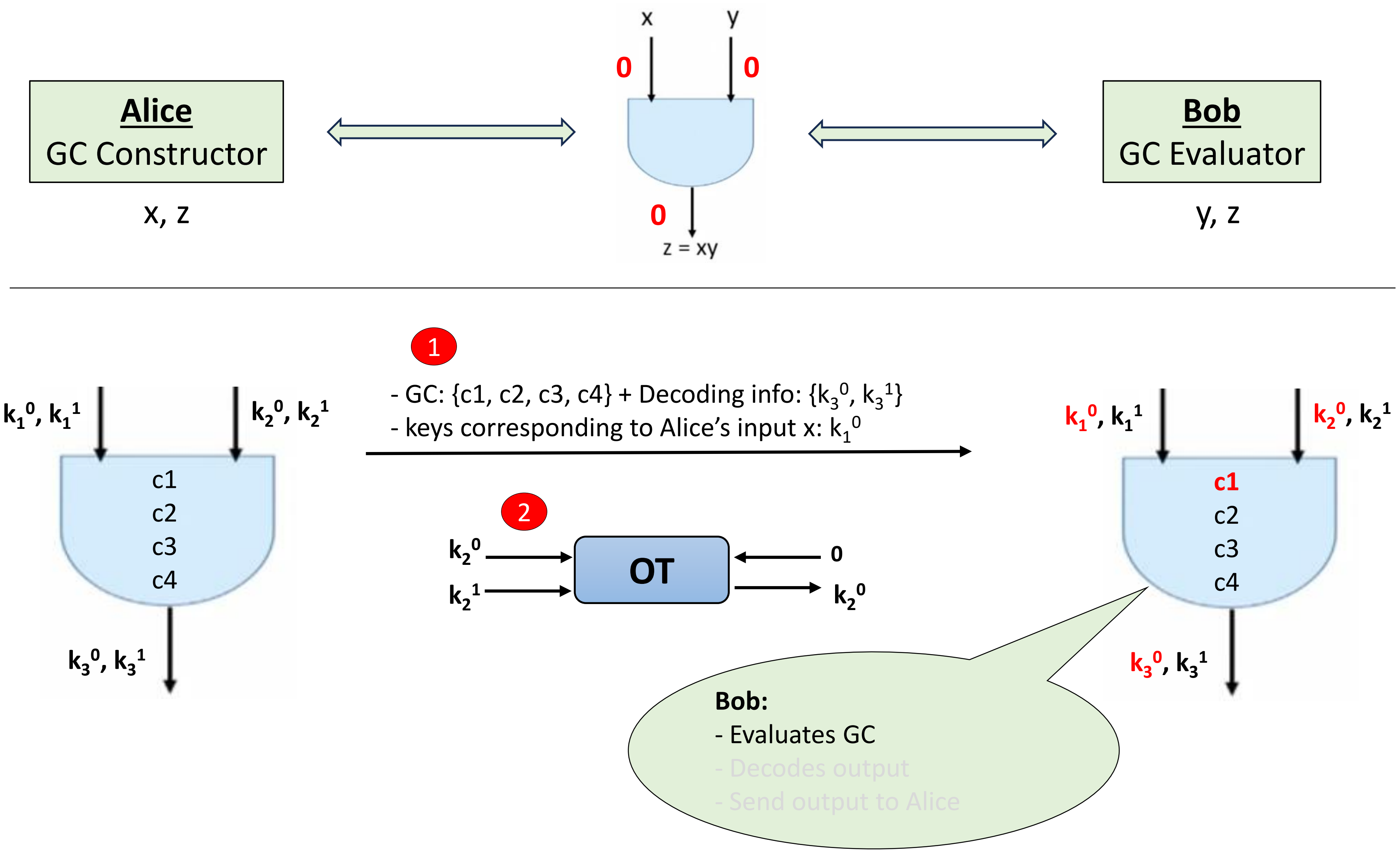
- GC: {c3, c1, c2, c4}}
- keys corresponding to Alice’s input x: $k_1^0$

# Yao’s 2-PC Protocol

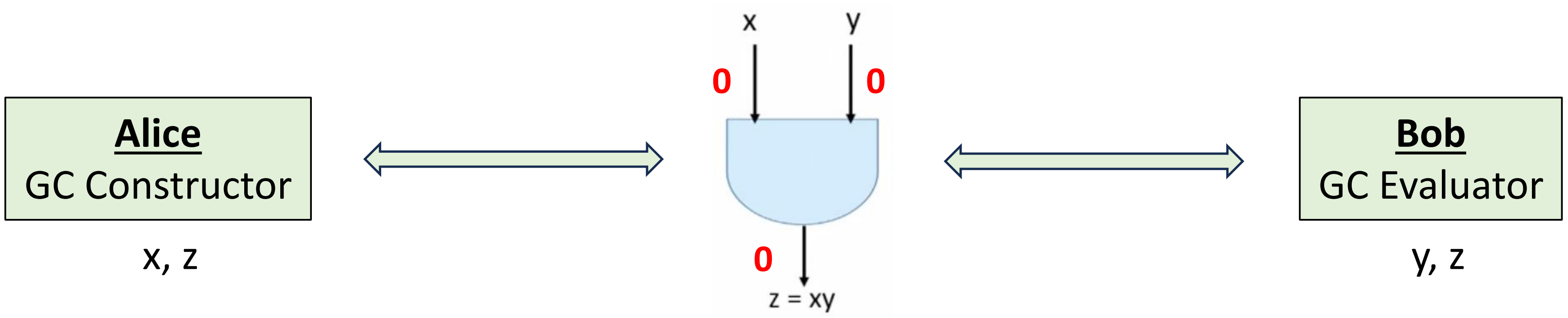


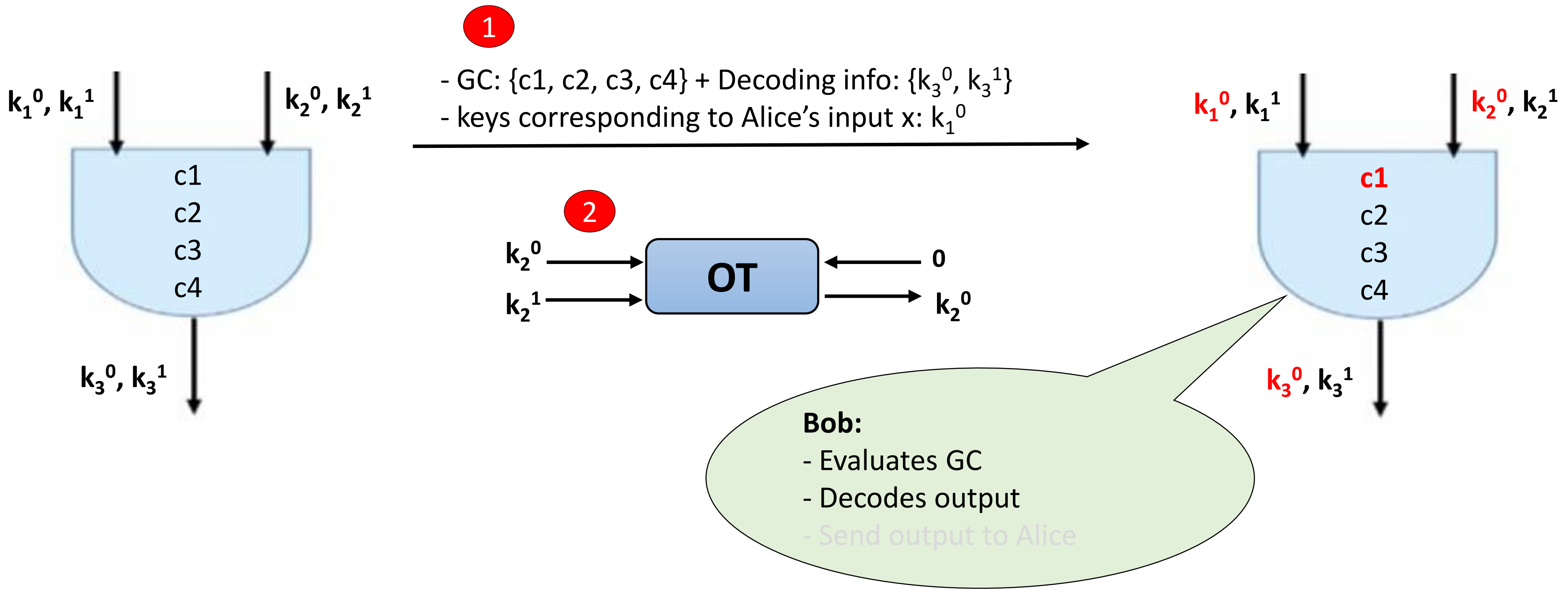



# Yao’s 2-PC Protocol

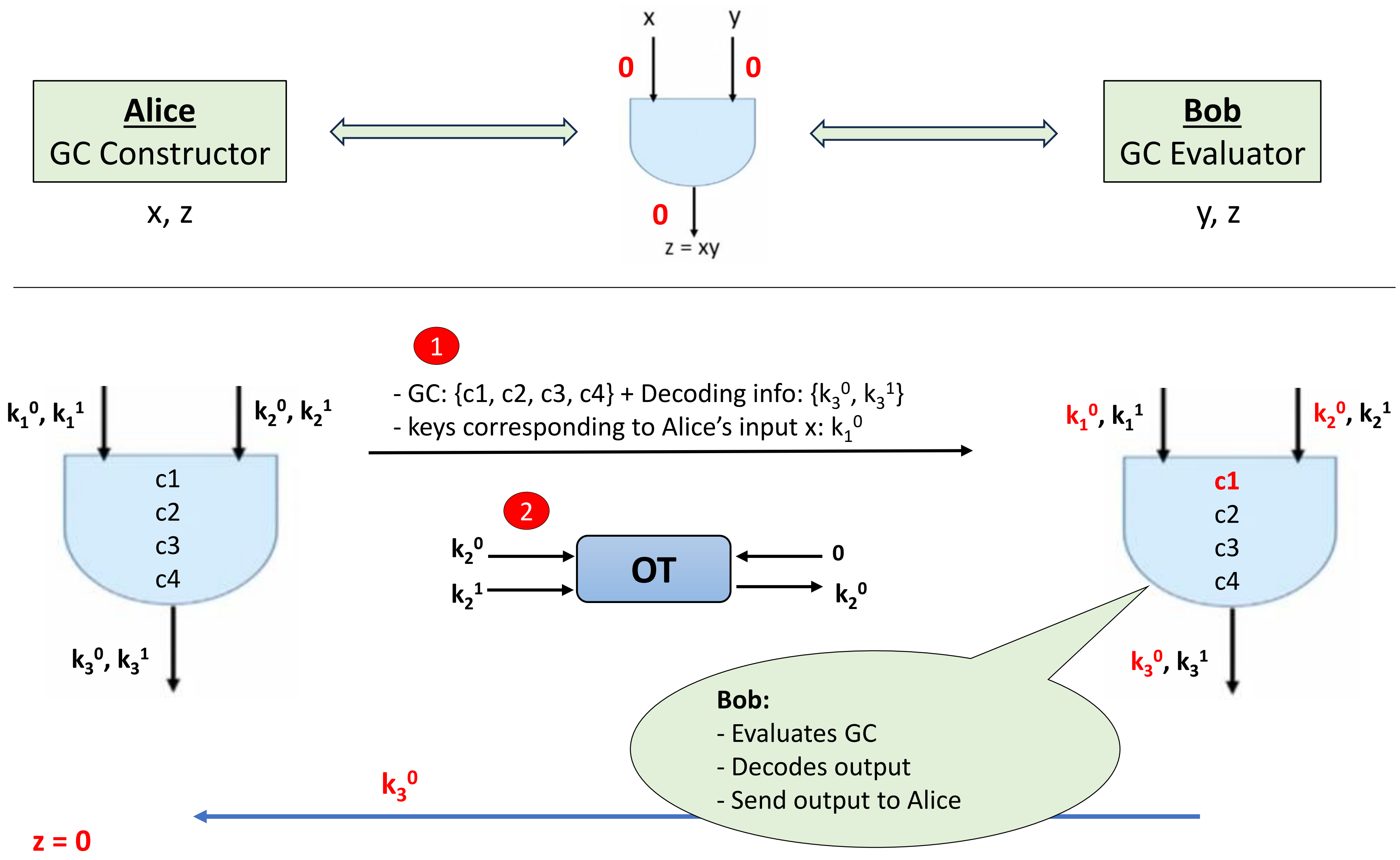


# Yao's 2-PC Protocol



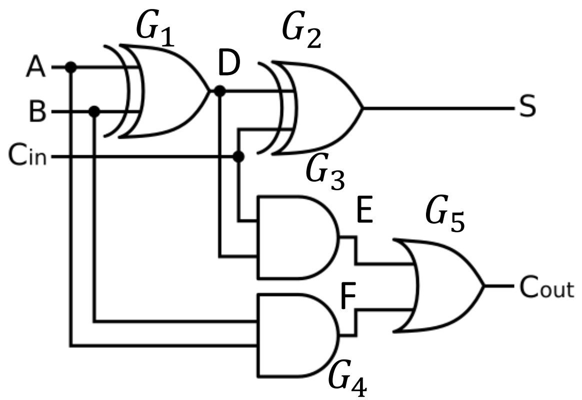
# Yao's 2-PC Protocol



# Yao's 2-PC Protocol (Complex function)

**Alice**
GC Constructor

Alice's input:
$A = 1$




Assume $C_{in} = 0$

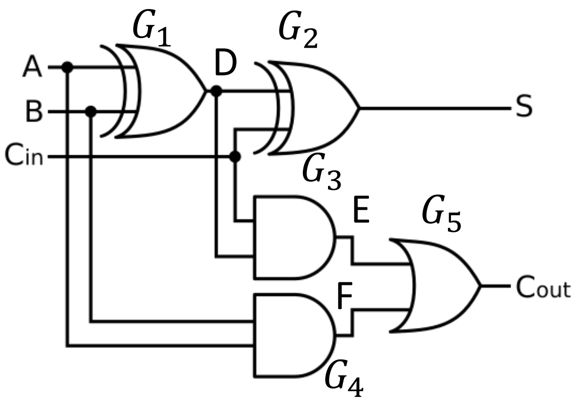
**Bob**
GC Evaluator

Bob's input:
$B = 0$

# Yao's 2-PC Protocol (Complex function)

**Alice**
GC Constructor

Alice's input:
$A = 1$




Assume $C_{in} = 0$

**Bob**
GC Evaluator

Bob's input:
$B = 0$

1

Generate keys:

$$(A = 0, k_{A_0}), (A = 1, k_{A_1})$$
$$(B = 0, k_{B_0}), (B = 1, k_{B_1})$$
$$(C_{in} = 0, k_{C_{in_0}}), (C_{in} = 1, k_{C_{in_1}})$$
$$(D = 0, k_{D_0}), (D = 1, k_{D_1})$$
$$(E = 0, k_{E_0}), (E = 1, k_{E_1})$$
$$(F = 0, k_{F_0}), (F = 1, k_{F_1})$$
$$(S = 0, k_{S_0}), (S = 1, k_{S_1})$$
$$(C_{out} = 0, k_{C_{out_0}}), (C_{out} = 1, k_{C_{out_1}})$$

Truth tables for all gates $\mathcal{G}$

$G_1$

$$E_{k_{A_0}}\left(E_{k_{B_0}}(k_{D_0})\right)$$
$$E_{k_{A_0}}\left(E_{k_{B_1}}(k_{D_1})\right)$$
$$E_{k_{A_1}}\left(E_{k_{B_0}}(k_{D_1})\right)$$
$$E_{k_{A_1}}\left(E_{k_{B_1}}(k_{D_0})\right)$$

$G_2$

$$E_{k_{D_0}}\left(E_{C_{in_0}}(k_{S_0})\right)$$
$$E_{k_{D_0}}\left(E_{C_{in_1}}(k_{S_1})\right)$$
$$E_{k_{D_1}}\left(E_{C_{in_0}}(k_{S_1})\right)$$
$$E_{k_{D_1}}\left(E_{C_{in_1}}(k_{S_0})\right)$$

$G_3$

$$E_{k_{D_0}}\left(E_{C_{in_0}}(k_{E_0})\right)$$
$$E_{k_{D_0}}\left(E_{C_{in_1}}(k_{E_0})\right)$$
$$E_{k_{D_1}}\left(E_{C_{in_0}}(k_{E_0})\right)$$
$$E_{k_{D_1}}\left(E_{C_{in_1}}(k_{E_1})\right)$$

$G_4$

$$E_{k_{A_0}}\left(E_{C_{B_0}}(k_{F_0})\right)$$
$$E_{k_{A_0}}\left(E_{C_{B_1}}(k_{F_0})\right)$$
$$E_{k_{A_1}}\left(E_{C_{B_0}}(k_{F_0})\right)$$
$$E_{k_{A_1}}\left(E_{C_{B_1}}(k_{F_1})\right)$$

$G_5$

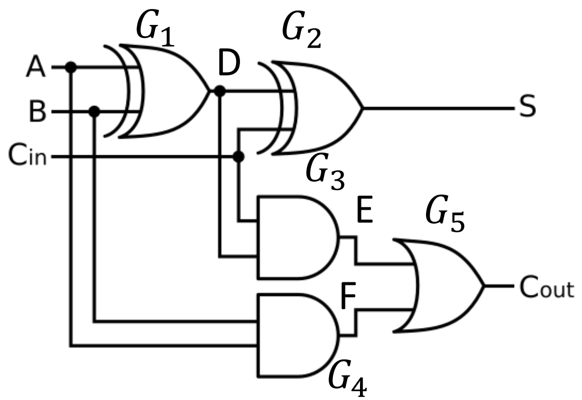
$$E_{k_{F_0}}\left(E_{k_{E_0}}\left(k_{Cout_0}\right)\right)$$
$$E_{k_{F_0}}\left(E_{k_{E_1}}\left(k_{Cout_1}\right)\right)$$
$$E_{k_{F_1}}\left(E_{k_{E_0}}\left(k_{Cout_1}\right)\right)$$
$$E_{k_{F_1}}\left(E_{k_{E_1}}\left(k_{Cout_1}\right)\right)$$

Manifest for the digital circuit $\mathcal{C}$

$$G_1(A, B) = D$$
$$G_2(D, C_{in}) = S$$
$$G_3(D, C_{in}) = E$$
$$G_4(A, B) = F$$
$$G_5(E, F) = C_{out}$$

# Yao's 2-PC Protocol (Complex function)

**Alice**
GC Constructor




**Bob**
GC Evaluator

Alice's input:
$A = 1$

Assume $C_{in} = 0$

Bob's input:
$B = 0$

2

$$\mathcal{G} = \{G_1, G_2, G_3, G_4, G_5\}, \mathcal{C}, (A, k_{A_1}), (C_{in} k_{C_{in_0}})$$

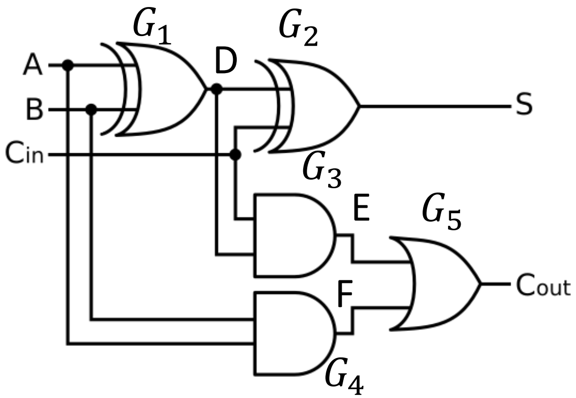
Generate keys:

$(A = 0, k_{A_0}), (A = 1, k_{A_1})$
$(B = 0, k_{B_0}), (B = 1, k_{B_1})$
$(C_{in} = 0, k_{C_{in_0}}), (C_{in} = 1, k_{C_{in_1}})$
$(D = 0, k_{D_0}), (D = 1, k_{D_1})$
$(E = 0, k_{E_0}), (E = 1, k_{E_1})$
$(F = 0, k_{F_0}), (F = 1, k_{F_1})$
$(S = 0, k_{S_0}), (S = 1, k_{S_1})$
$(C_{out} = 0, k_{C_{out_0}}), (C_{out} = 1, k_{C_{out_1}})$

# Yao's 2-PC Protocol (Complex function)

**Alice**
GC Constructor

Alice's input:
$A = 1$




Assume $C_{in} = 0$

**Bob**
GC Evaluator

Bob's input:
$B = 0$

$G_1(A, B) = D$
$G_2(D, C_{in}) = S$
$G_3(D, C_{in}) = E$
$G_4(A, B) = F$
$G_5(E, F) = C_{out}$

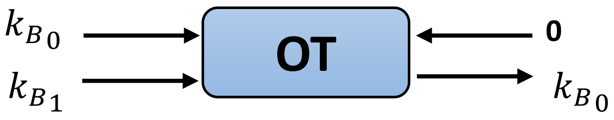
---

$$\mathcal{G} = \{G_1, G_2, G_3, G_4, G_5\}, \mathcal{C}, (A, k_{A_1}), (C_{in} k_{C_{in_0}})$$

Generate keys:

$(A = 0, k_{A_0}), (A = 1, k_{A_1})$
$(B = 0, k_{B_0}), (B = 1, k_{B_1})$
$(C_{in} = 0, k_{C_{in_0}}), (C_{in} = 1, k_{C_{in_1}})$
$(D = 0, k_{D_0}), (D = 1, k_{D_1})$
$(E = 0, k_{E_0}), (E = 1, k_{E_1})$
$(F = 0, k_{F_0}), (F = 1, k_{F_1})$
$(S = 0, k_{S_0}), (S = 1, k_{S_1})$
$(C_{out} = 0, k_{C_{out_0}}), (C_{out} = 1, k_{C_{out_1}})$

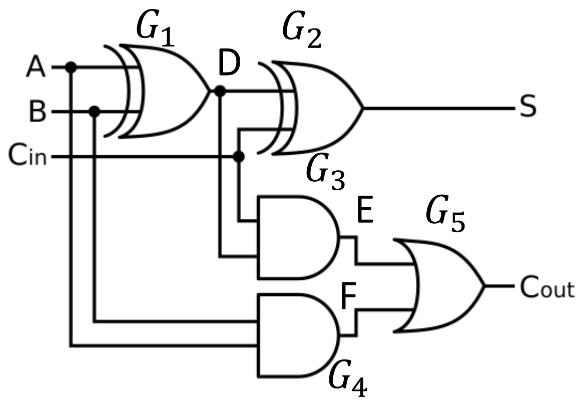
3



# Yao's 2-PC Protocol (Complex function)

**Alice**
GC Constructor

Alice's input:
$A = 1$




Assume $C_{in} = 0$

**Bob**
GC Evaluator

Bob's input:
$B = 0$

$G_1(A, B) = D$
$G_2(D, C_{in}) = S$
$G_3(D, C_{in}) = E$
$G_4(A, B) = F$
$G_5(E, F) = C_{out}$

---

$$\mathcal{G} = \{G_1, G_2, G_3, G_4, G_5\}, \mathcal{C}, (A, k_{A_1}), (C_{in} k_{C_{in_0}}), k_{B_0}$$

4

$G_1$

$E_{k_{A_0}}\left(E_{k_{B_0}}(k_{D_0})\right)$

$E_{k_{A_0}}\left(E_{k_{B_1}}(k_{D_1})\right)$

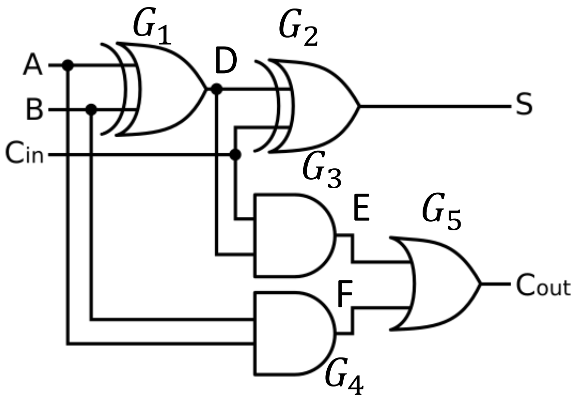
$E_{k_{A_1}}\left(E_{k_{B_0}}(k_{D_1})\right)$ -> $k_{D_1}$

$E_{k_{A_1}}\left(E_{k_{B_1}}(k_{D_0})\right)$

# Yao's 2-PC Protocol (Complex function)

**Alice**
GC Constructor

⟺




⟺

**Bob**
GC Evaluator

Alice's input:
$A = 1$

Assume $C_{in} = 0$

Bob's input:
$B = 0$

$G_1(A, B) = D$
$G_2(D, C_{in}) = S$
$G_3(D, C_{in}) = E$
$G_4(A, B) = F$
$G_5(E, F) = C_{out}$

$$\mathcal{G} = \{G_1, G_2, G_3, G_4, G_5\}, \mathcal{C}, (A, k_{A_1}), (C_{in} k_{C_{in_0}}), k_{B_0}$$

4

$G_1$

$E_{k_{A_0}}\left(E_{k_{B_0}}(k_{D_0})\right)$
$E_{k_{A_0}}\left(E_{k_{B_1}}(k_{D_1})\right)$
$E_{k_{A_1}}\left(E_{k_{B_0}}(k_{D_1})\right)$ -> $k_{D_1}$
$E_{k_{A_1}}\left(E_{k_{B_1}}(k_{D_0})\right)$

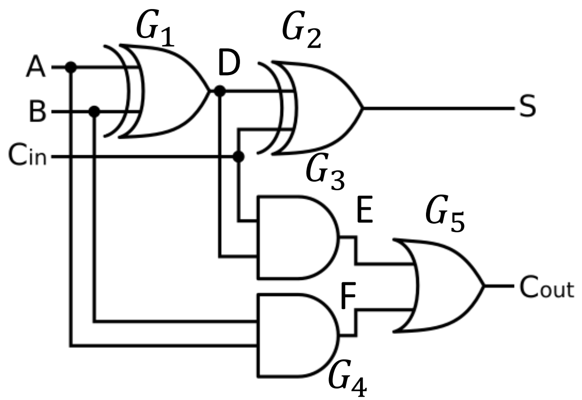
$G_2$

$E_{k_{D_0}}\left(E_{C_{in_0}}(k_{S_0})\right)$
$E_{k_{D_0}}\left(E_{C_{in_1}}(k_{S_1})\right)$
$E_{k_{D_1}}\left(E_{C_{in_0}}(k_{S_1})\right)$ -> $k_{S_1}$
$E_{k_{D_1}}\left(E_{C_{in_1}}(k_{S_0})\right)$

# Yao's 2-PC Protocol (Complex function)

**Alice**
GC Constructor

Alice's input:
$A = 1$




Assume $C_{in} = 0$

**Bob**
GC Evaluator

Bob's input:
$B = 0$

$G_1(A,B) = D$
$G_2(D,C_{in}) = S$
$G_3(D,C_{in}) = E$
$G_4(A,B) = F$
$G_5(E,F) = C_{out}$

$$\mathcal{G} = \{G_1, G_2, G_3, G_4, G_5\}, \mathcal{C}, (A, k_{A_1}), (C_{in} k_{C_{in_0}}), k_{B\,0}$$

4

$G_1$

$E_{k_{A_0}}\left(E_{k_{B_0}}(k_{D\,0})\right)$
$E_{k_{A_0}}\left(E_{k_{B_1}}(k_{D\,1})\right)$
$E_{k_{A_1}}\left(E_{k_{B_0}}(k_{D\,1})\right)$ -> $k_{D\,1}$
$E_{k_{A_1}}\left(E_{k_{B_1}}(k_{D\,0})\right)$

$G_2$

$E_{k_{D_0}}\left(E_{C_{in_0}}(k_{S\,0})\right)$
$E_{k_{D_0}}\left(E_{C_{in_1}}(k_{S\,1})\right)$
$E_{k_{D_1}}\left(E_{C_{in_0}}(k_{S\,1})\right)$ -> $k_{S\,1}$
$E_{k_{D_1}}\left(E_{C_{in_1}}(k_{S\,0})\right)$

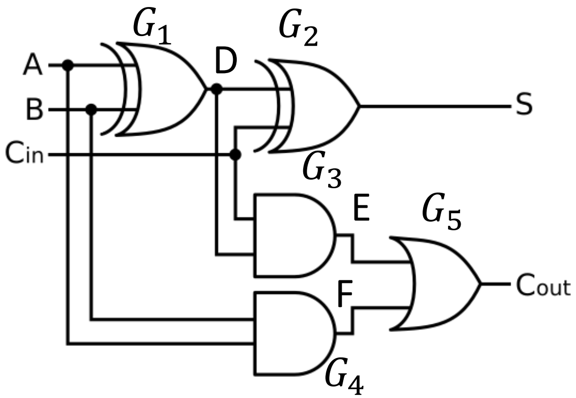
$G_3$

$E_{k_{D_0}}\left(E_{C_{in_0}}(k_{E\,0})\right)$
$E_{k_{D_0}}\left(E_{C_{in_1}}(k_{E\,0})\right)$
$E_{k_{D_1}}\left(E_{C_{in_0}}(k_{E\,0})\right)$ -> $k_{E\,0}$
$E_{k_{D_1}}\left(E_{C_{in_1}}(k_{E\,1})\right)$

# Yao's 2-PC Protocol (Complex function)

**Alice**
GC Constructor

⟺




⟺

**Bob**
GC Evaluator

Alice's input:
$A = 1$

Assume $C_{in} = 0$

Bob's input:
$B = 0$

$G_1(A, B) = D$
$G_2(D, C_{in}) = S$
$G_3(D, C_{in}) = E$
$G_4(A, B) = F$
$G_5(E, F) = C_{out}$

---

$$\mathcal{G} = \{G_1, G_2, G_3, G_4, G_5\}, \mathcal{C}, (A, k_{A_1}), (C_{in} k_{C_{in_0}}), k_{B_0}$$

4

$G_1$

$E_{k_{A_0}}\left(E_{k_{B_0}}(k_{D_0})\right)$
$E_{k_{A_0}}\left(E_{k_{B_1}}(k_{D_1})\right)$
$E_{k_{A_1}}\left(E_{k_{B_0}}(k_{D_1})\right)$ -> $k_{D_1}$
$E_{k_{A_1}}\left(E_{k_{B_1}}(k_{D_0})\right)$

$G_2$

$E_{k_{D_0}}\left(E_{C_{in_0}}(k_{S_0})\right)$
$E_{k_{D_0}}\left(E_{C_{in_1}}(k_{S_1})\right)$
$E_{k_{D_1}}\left(E_{C_{in_0}}(k_{S_1})\right)$ -> $k_{S_1}$
$E_{k_{D_1}}\left(E_{C_{in_1}}(k_{S_0})\right)$

$G_3$

$E_{k_{D_0}}\left(E_{C_{in_0}}(k_{E_0})\right)$
$E_{k_{D_0}}\left(E_{C_{in_1}}(k_{E_0})\right)$
$E_{k_{D_1}}\left(E_{C_{in_0}}(k_{E_0})\right)$ -> $k_{E_0}$
$E_{k_{D_1}}\left(E_{C_{in_1}}(k_{E_1})\right)$

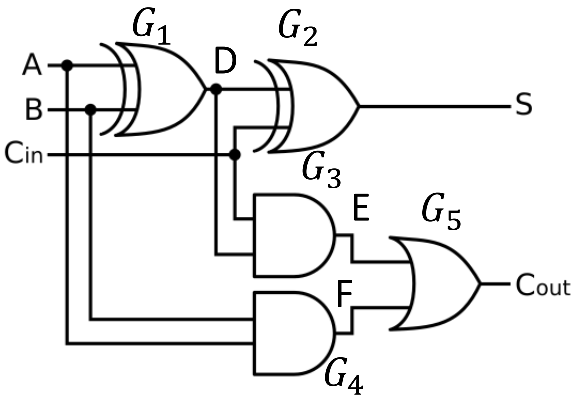
$G_4$

$E_{k_{A_0}}\left(E_{C_{B_0}}(k_{F_0})\right)$
$E_{k_{A_0}}\left(E_{C_{B_1}}(k_{F_0})\right)$
$E_{k_{A_1}}\left(E_{C_{B_0}}(k_{F_0})\right)$ -> $k_{F_0}$
$E_{k_{A_1}}\left(E_{C_{B_1}}(k_{F_1})\right)$

# Yao's 2-PC Protocol (Complex function)

**Alice**
GC Constructor

Alice's input:
$A = 1$




Assume $C_{in} = 0$

**Bob**
GC Evaluator

Bob's input:
$B = 0$

$G_1(A, B) = D$
$G_2(D, C_{in}) = S$
$G_3(D, C_{in}) = E$
$G_4(A, B) = F$
$G_5(E, F) = C_{out}$

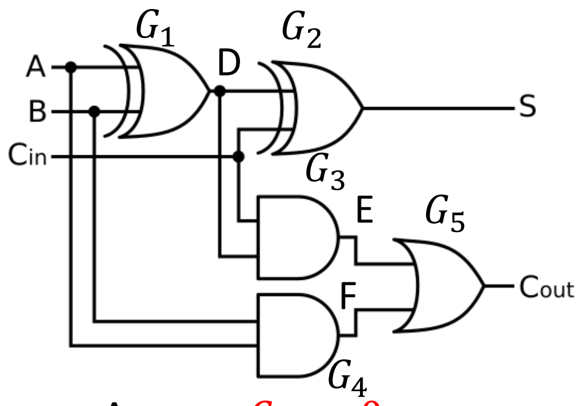
$$\mathcal{G} = \{G_1, G_2, G_3, G_4, G_5\}, \mathcal{C}, (A, k_{A_1}), (C_{in} k_{C_{in_0}}), k_{B_0}$$

4

| $G_1$ | $G_2$ | $G_3$ | $G_4$ | $G_5$ |
|---|---|---|---|---|
| $E_{k_{A_0}}(E_{k_{B_0}}(k_{D_0}))$ | $E_{k_{D_0}}(E_{C_{in_0}}(k_{S_0}))$ | $E_{k_{D_0}}(E_{C_{in_0}}(k_{E_0}))$ | $E_{k_{A_0}}(E_{C_{B_0}}(k_{F_0}))$ | $E_{k_{F_0}}(E_{k_{E_0}}(k_{C_{out_0}}))$ -> $k_{C_{out_0}}$ |
| $E_{k_{A_0}}(E_{k_{B_1}}(k_{D_1}))$ | $E_{k_{D_0}}(E_{C_{in_1}}(k_{S_1}))$ | $E_{k_{D_0}}(E_{C_{in_1}}(k_{E_0}))$ | $E_{k_{A_0}}(E_{C_{B_1}}(k_{F_0}))$ | $E_{k_{F_0}}(E_{k_{E_1}}(k_{C_{out_1}}))$ |
| $E_{k_{A_1}}(E_{k_{B_0}}(k_{D_1}))$ -> $k_{D_1}$ | $E_{k_{D_1}}(E_{C_{in_0}}(k_{S_1}))$ -> $k_{S_1}$ | $E_{k_{D_1}}(E_{C_{in_0}}(k_{E_0}))$ -> $k_{E_0}$ | $E_{k_{A_1}}(E_{C_{B_0}}(k_{F_0}))$ -> $k_{F_0}$ | $E_{k_{F_1}}(E_{k_{E_0}}(k_{C_{out_1}}))$ |
| $E_{k_{A_1}}(E_{k_{B_1}}(k_{D_0}))$ | $E_{k_{D_1}}(E_{C_{in_1}}(k_{S_0}))$ | $E_{k_{D_1}}(E_{C_{in_1}}(k_{E_1}))$ | $E_{k_{A_1}}(E_{C_{B_1}}(k_{F_1}))$ | $E_{k_{F_1}}(E_{k_{E_1}}(k_{C_{out_1}}))$ |

# Yao's 2-PC Protocol (Complex function)

**Alice**
GC Constructor




**Bob**
GC Evaluator

Alice's input:
$A = 1$

Assume $C_{in} = 0$

Bob's input:
$B = 0$

$G_1(A, B) = D$
$G_2(D, C_{in}) = S$
$G_3(D, C_{in}) = E$
$G_4(A, B) = F$
$G_5(E, F) = C_{out}$

$$\mathcal{G} = \{G_1, G_2, G_3, G_4, G_5\}, \mathcal{C}, (A, k_{A_1}), (C_{in} k_{C_{in_0}}), k_{B_0}$$

Generate keys:

$(A = 0, k_{A_0}), (A = 1, k_{A_1})$
$(B = 0, k_{B_0}), (B = 1, k_{B_1})$
$(C_{in} = 0, k_{C_{in_0}}), (C_{in} = 1, k_{C_{in_1}})$
$(D = 0, k_{D_0}), (D = 1, k_{D_1})$
$(E = 0, k_{E_0}), (E = 1, k_{E_1})$
$(F = 0, k_{F_0}), (F = 1, k_{F_1})$
$(S = 0, k_{S_0}), (S = 1, k_{S_1})$
$(C_{out} = 0, k_{C_{out_0}}), (C_{out} = 1, k_{C_{out_1}})$

5

$$\mathcal{O} = \{k_{S_1}, k_{C_{out_0}}\}$$

# References

- Secure Computation (Online Course)
- https://web.engr.oregonstate.edu/~rosulekm/