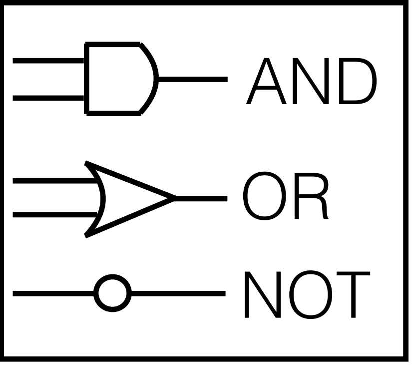
Secure Computation - II

(Garbled Circuits)

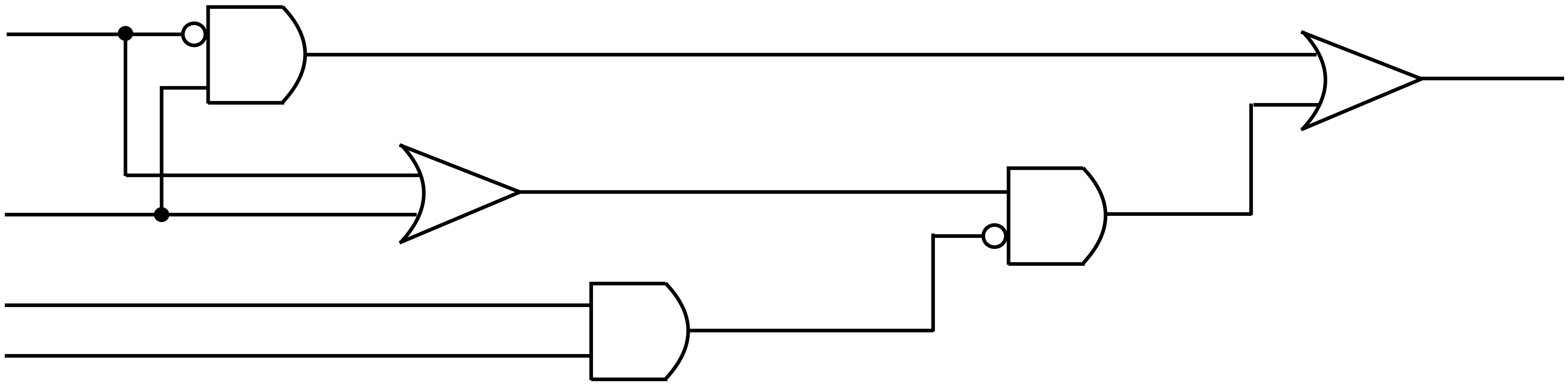
Lecture 14

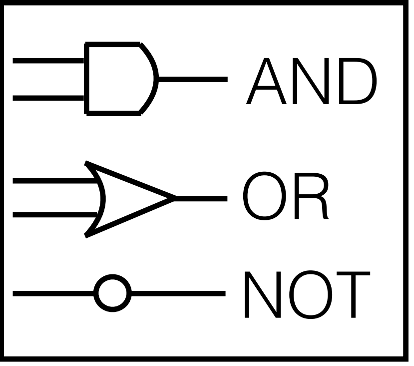
Secure Two-Party Computation from Garbled Circuits



Garbled Circuits

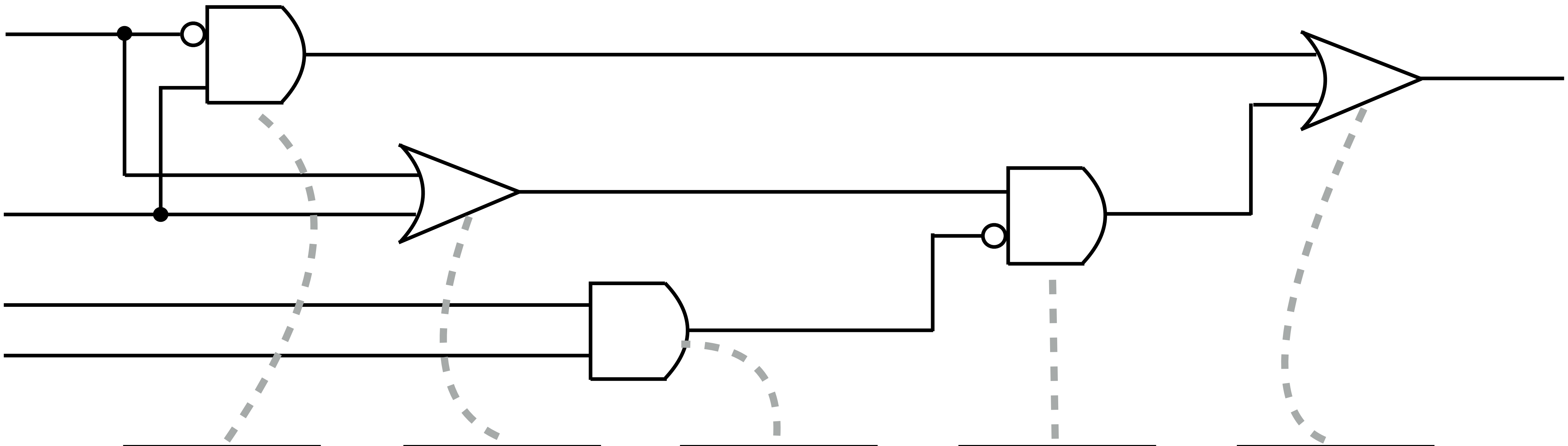
[Yao86]





Garbled Circuits

[Yao86]



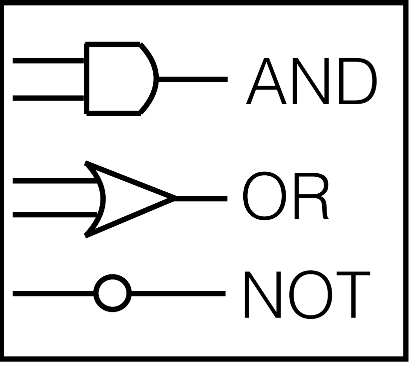
0	0	0
0	1	1
1	0	0
1	1	0

0	0	0
0	1	1
1	0	1
1	1	1

0	0	0
0	1	0
1	0	0
1	1	1

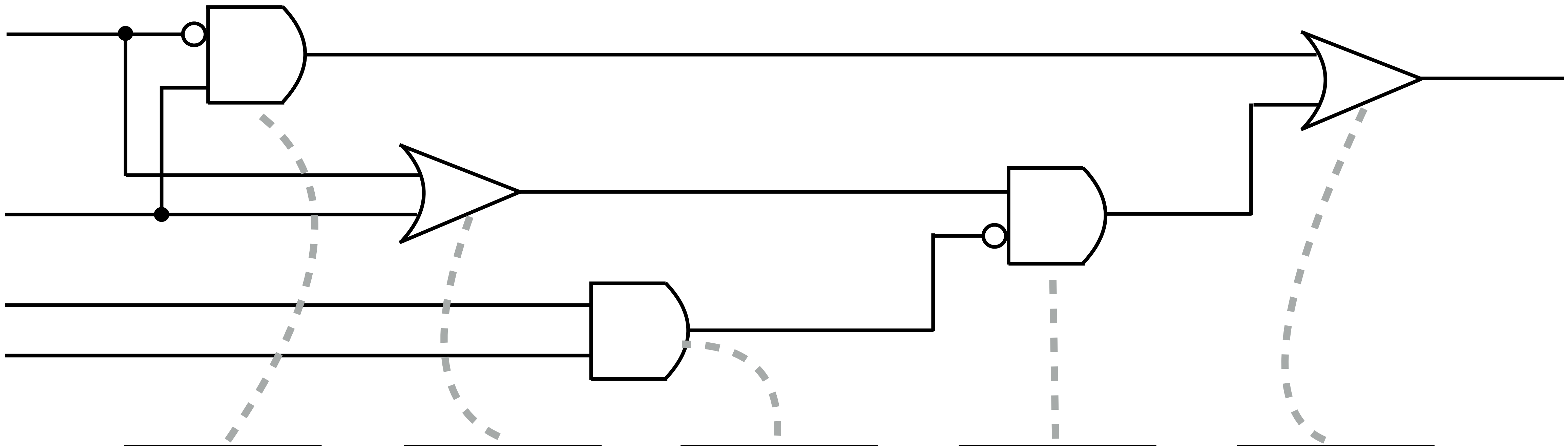
0	0	0
0	1	0
1	0	1
1	1	0

0	0	0
0	1	1
1	0	1
1	1	1



Garbled Circuits

[Yao86]



0	0	0
0	1	1
1	0	0
1	1	0

0	0	0
0	1	1
1	0	1
1	1	1

0	0	0
0	1	0
1	0	0
1	1	1

0	0	0
0	1	0
1	0	1
1	1	0

0	0	0
0	1	1
1	0	1
1	1	1

Goal: “Garble” (Circuit=C,input=x) s.t. (C_{garble},x_{garble}) only reveals $C(x)$

Syntax

- **Two algorithms**: $(Garble, Eval)$
  - $Garble(C)$ outputs $(C_{\text{garble}}, x_{\text{garble}})$
  - $Eval(C_{\text{garble}}, x_{\text{garble}})$ outputs a value z

Syntax

- **Two algorithms**: ($Garble$, $Eval$)
  - $Garble(C)$ outputs $(C_{\text{garble}}, x_{\text{garble}})$
  - $Eval(C_{\text{garble}}, x_{\text{garble}})$ outputs a value z
- Correctness:

For every (C,x),

$$\mathbf{Pr}[C(x) = Eval(C_{\text{garble}}, x_{\text{garble}}) \mid (C_{\text{garble}}, x_{\text{garble}}) = Garble(C,x)] = 1\text{-negl}(n)$$

Security

There exists a PPT simulator $\mathbf{S}$ s.t. for every (C,x),

$$(C_{\text{garble}}, x_{\text{garble}}) \sim \mathbf{S}(1^n, C, C(x))$$

where $(C_{\text{garble}}, x_{\text{garble}}) = Garble(C,x)$

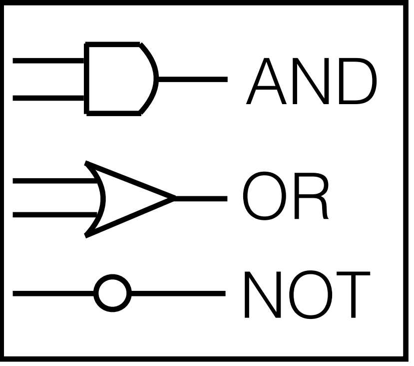
Security

There exists a PPT simulator $\mathbf{S}$ s.t. for every (C,x),

$$(C_{\text{garble}},x_{\text{garble}}) \sim \mathbf{S}(1^n,C,C(x))$$

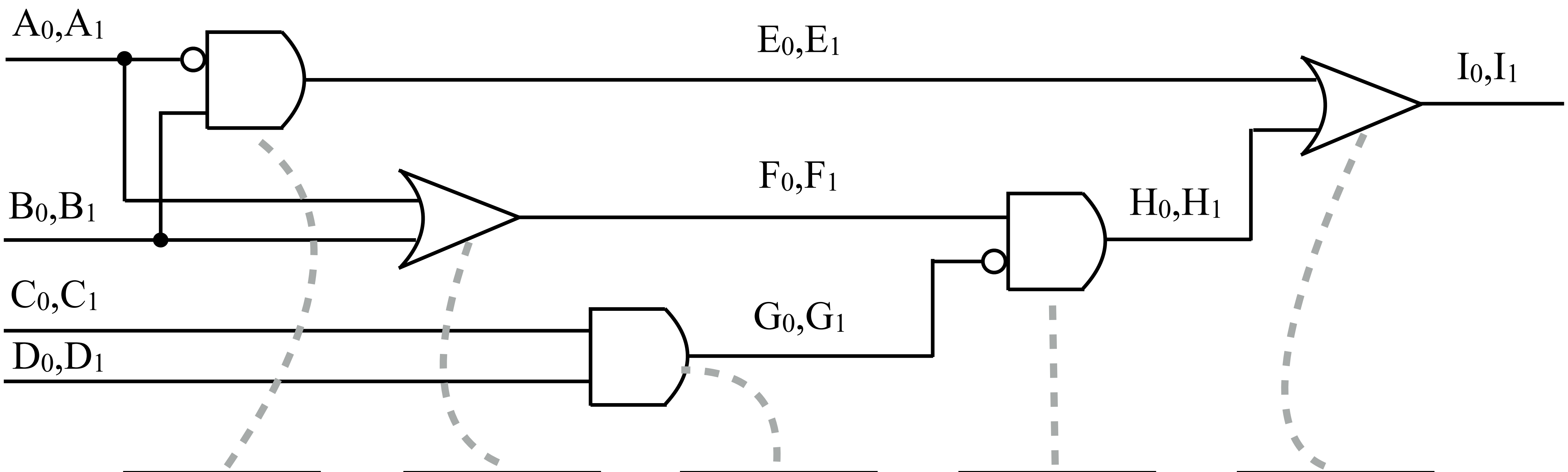
where $(C_{\text{garble}},x_{\text{garble}}) = Garble(C,x)$

- Hiding C: Use universal circuit and pass C as input to the universal circuit



Garbled Circuits

[Yao86]



0	0	0
0	1	1
1	0	0
1	1	0

0	0	0
0	1	1
1	0	1
1	1	1

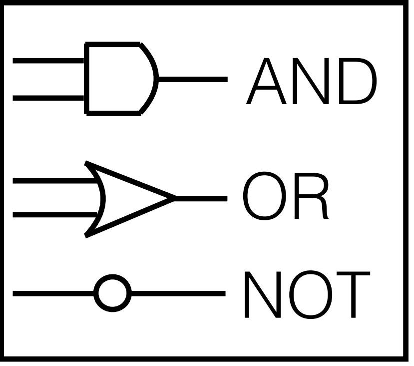
0	0	0
0	1	0
1	0	0
1	1	1

0	0	0
0	1	0
1	0	1
1	1	0

0	0	0
0	1	1
1	0	1
1	1	1

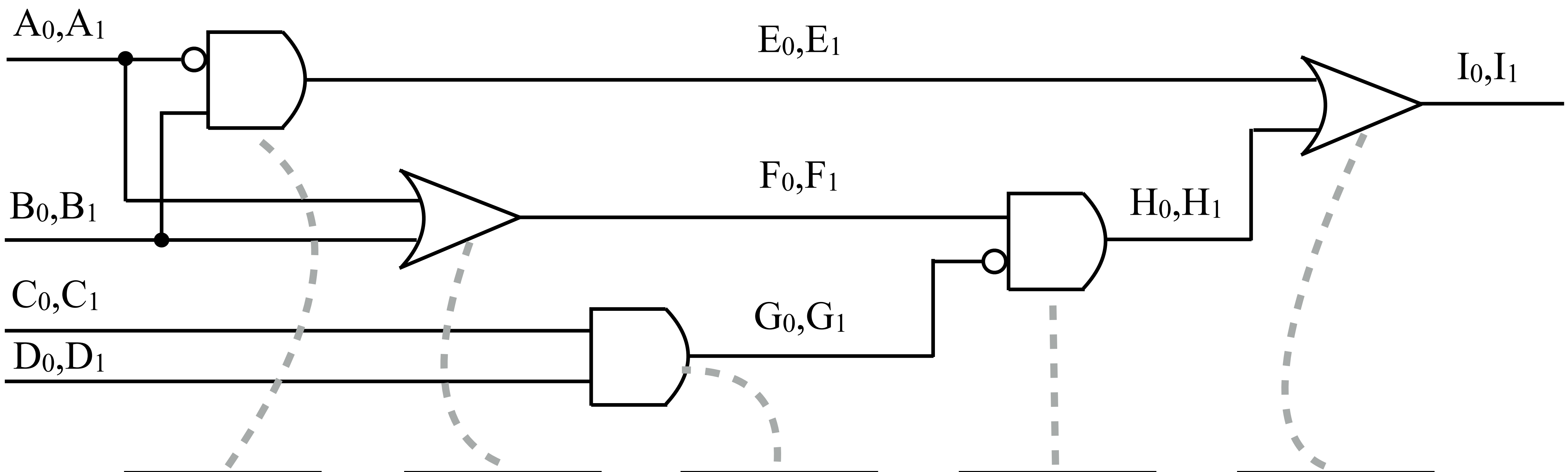
Garble(C,x):

- Pick random **labels** W_0,W_1 for each wire



Garbled Circuits

[Yao86]



A_0	B_0	E_0
A_0	B_1	E_1
A_1	B_0	E_0
A_1	B_1	E_0

A_0	B_0	F_0
A_0	B_1	F_1
A_1	B_0	F_1
A_1	B_1	F_1

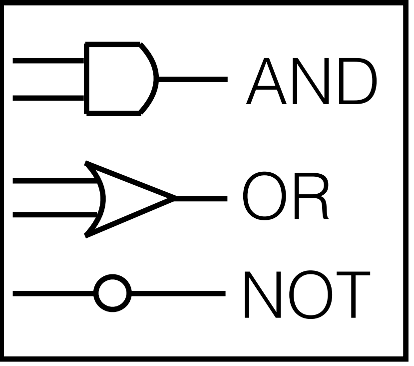
C_0	D_0	G_0
C_0	D_1	G_0
C_1	D_0	G_0
C_1	D_1	G_1

F_0	G_0	H_0
F_0	G_1	H_0
F_1	G_0	H_1
F_1	G_1	H_0

E_0	H_0	I_0
E_0	H_1	I_1
E_1	H_0	I_1
E_1	H_1	I_1

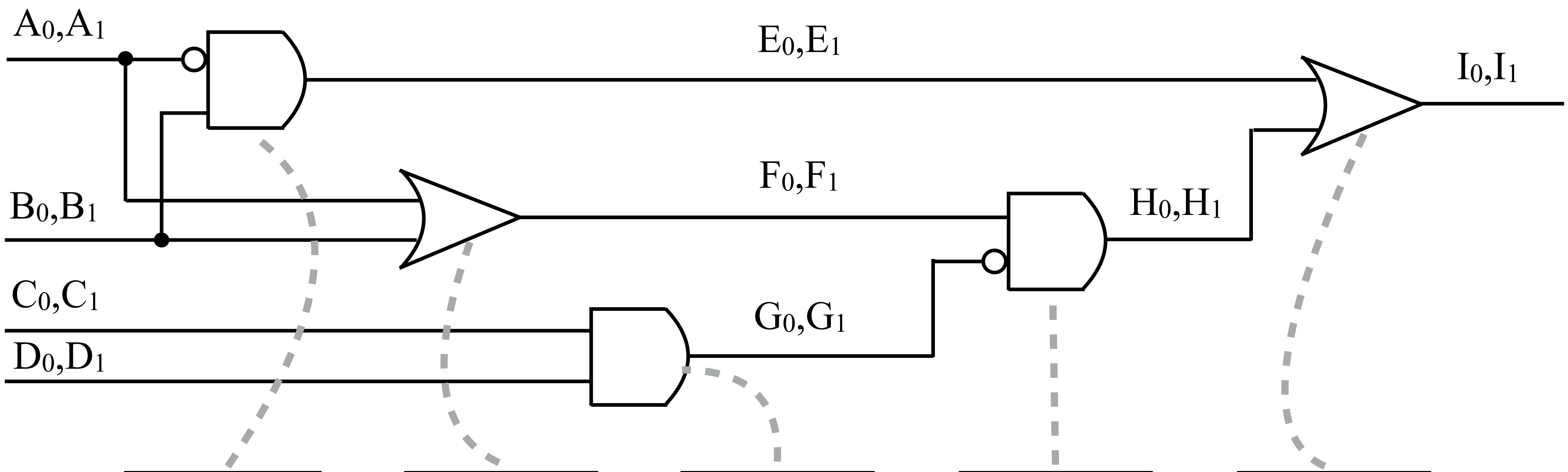
Garble(C,x):

- Pick random **labels** W_0,W_1 for each wire



Garbled Circuits

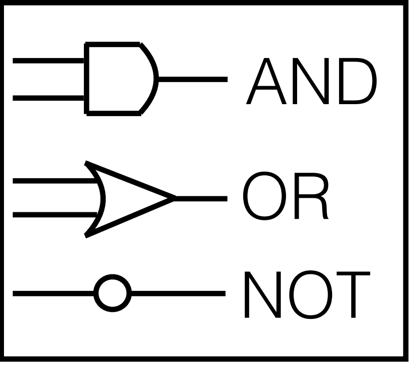
[Yao86]



$\mathbb{E}_{A0,B0}(E_0)$	$\mathbb{E}_{A0,B0}(F_0)$	$\mathbb{E}_{C0,D0}(G_0)$	$\mathbb{E}_{F0,G0}(H_0)$	$\mathbb{E}_{E0,H0}(I_0)$
$\mathbb{E}_{A0,B1}(E_1)$	$\mathbb{E}_{A0,B1}(F_1)$	$\mathbb{E}_{C0,D1}(G_0)$	$\mathbb{E}_{F0,G1}(H_0)$	$\mathbb{E}_{E0,H1}(I_1)$
$\mathbb{E}_{A1,B0}(E_0)$	$\mathbb{E}_{A1,B0}(F_1)$	$\mathbb{E}_{C1,D0}(G_0)$	$\mathbb{E}_{F1,G0}(H_1)$	$\mathbb{E}_{E1,H0}(I_1)$
$\mathbb{E}_{A1,B1}(E_0)$	$\mathbb{E}_{A1,B1}(F_1)$	$\mathbb{E}_{C1,D1}(G_1)$	$\mathbb{E}_{F1,G1}(H_0)$	$\mathbb{E}_{E1,H1}(I_1)$

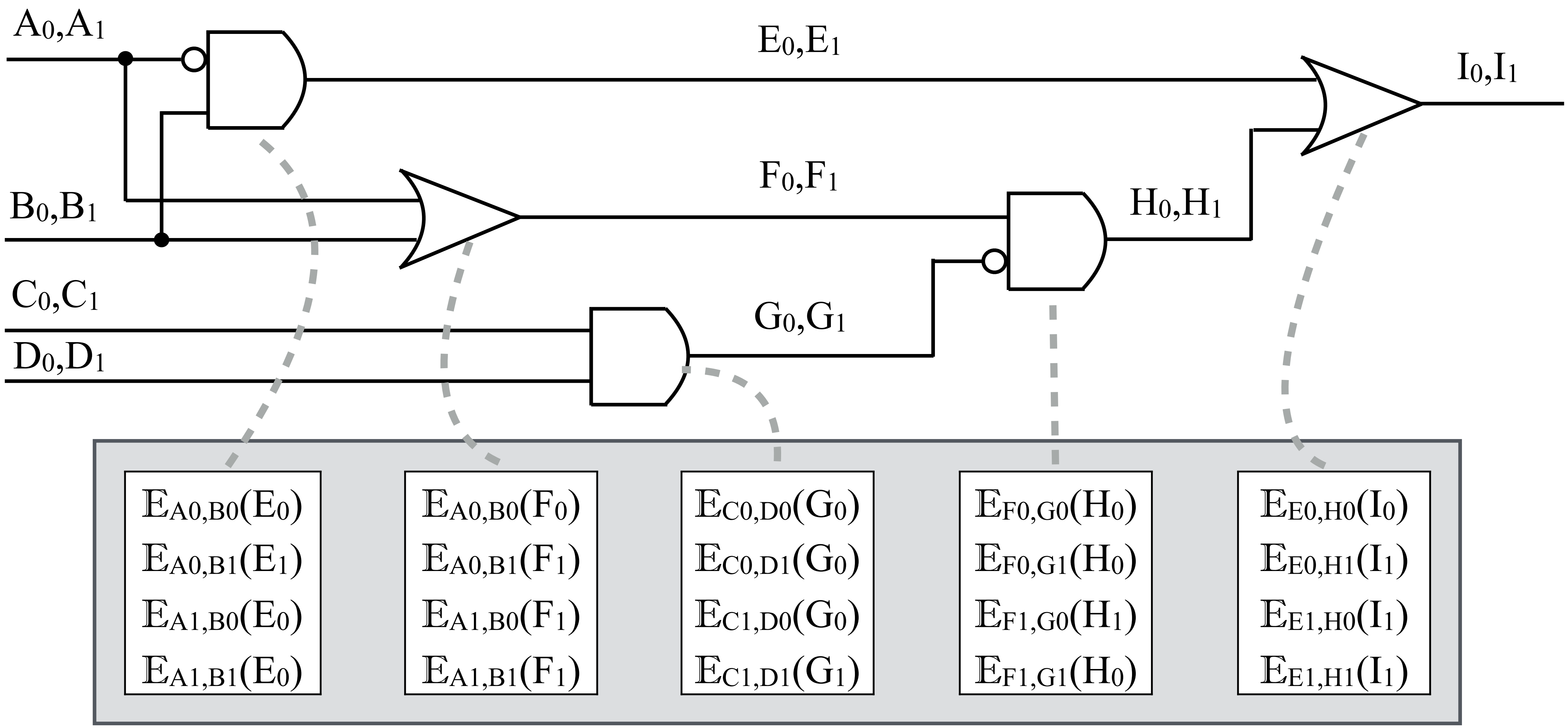
Garble(C,x):

- Pick random **labels** W_0,W_1 for each wire
- “Encrypt” truth table of each gate



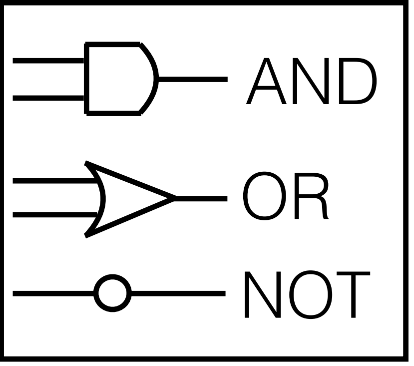
Garbled Circuits

[Yao86]



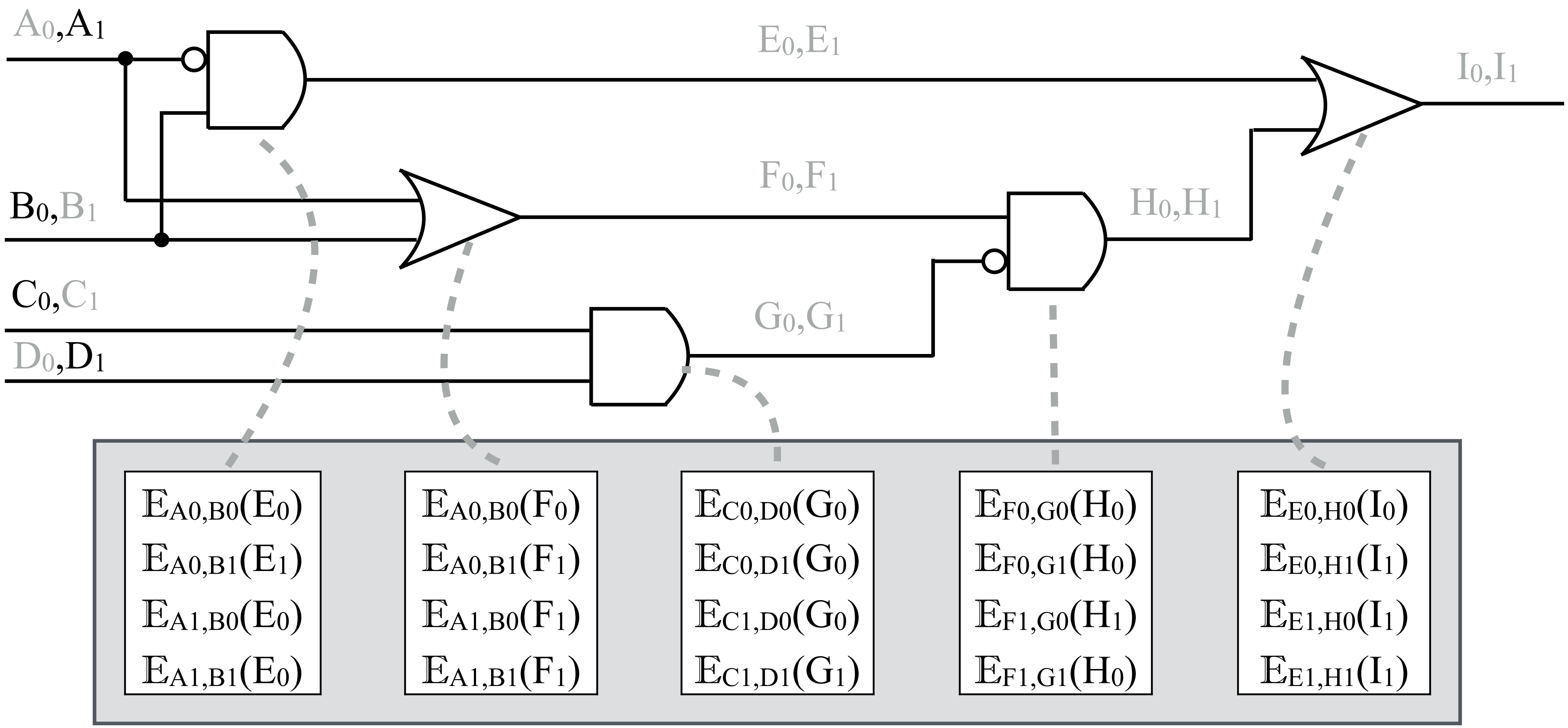
Garble(C,x):

- Pick random **labels** W_0,W_1 for each wire
- "Encrypt" truth table of each gate
- **Garbled Circuit** C_{garble}= all encrypted gates



Garbled Circuits

[Yao86]

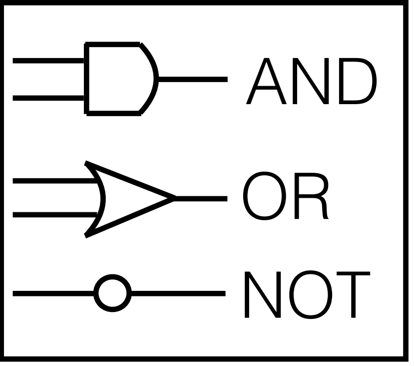


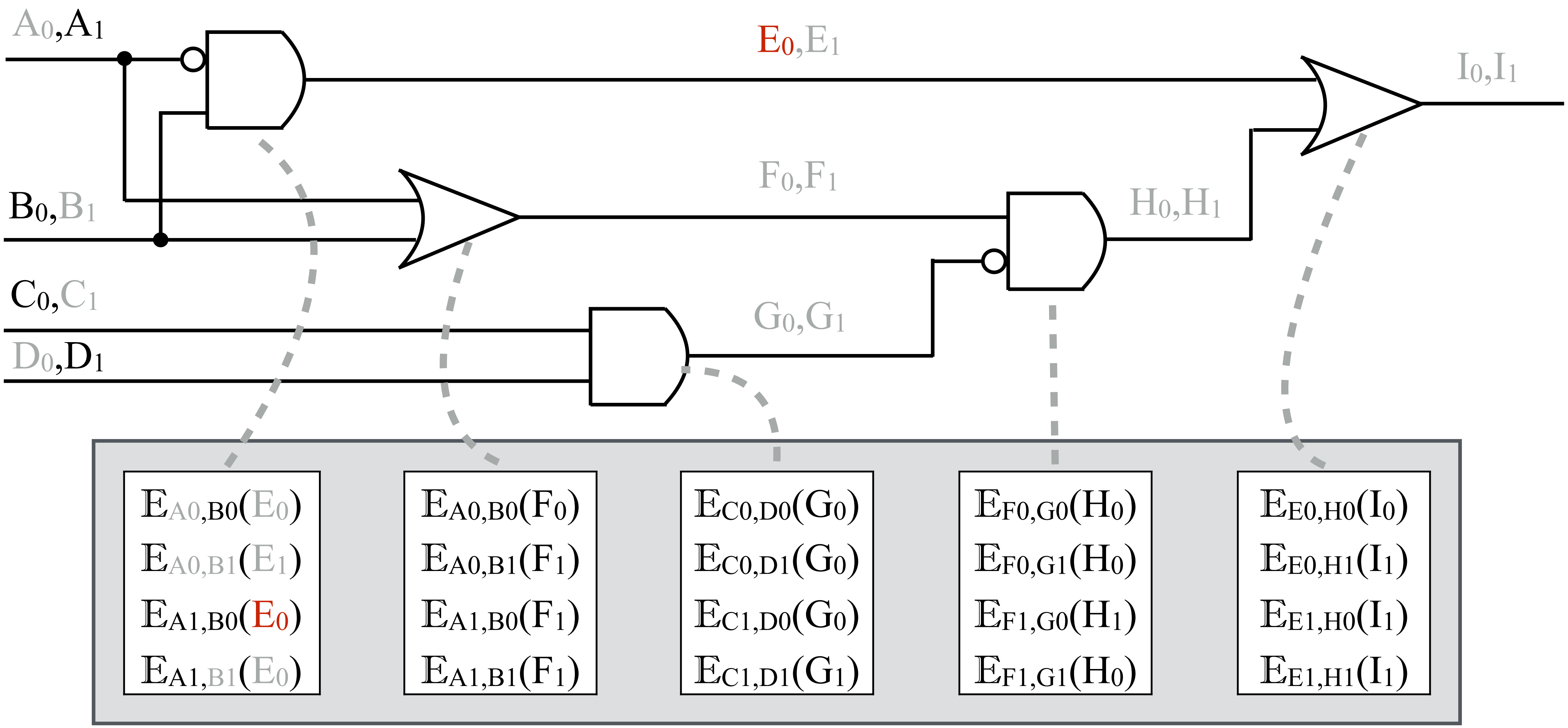
Garble(C,x):

- Pick random **labels** W_0,W_1 for each wire
- “Encrypt” truth table of each gate
- **Garbled Circuit** C_{garble}= all encrypted gates
- **Garbled Input** x_{garble}= one label per wire

Garbled Circuits

[Yao86]



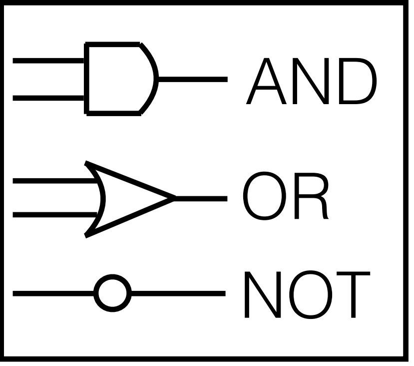


Garble(C,x):

- Pick random **labels** W_0,W_1 for each wire
- “Encrypt” truth table of each gate
- **Garbled Circuit** C_{garble}= all encrypted gates
- **Garbled Input** x_{garble}= one label per wire

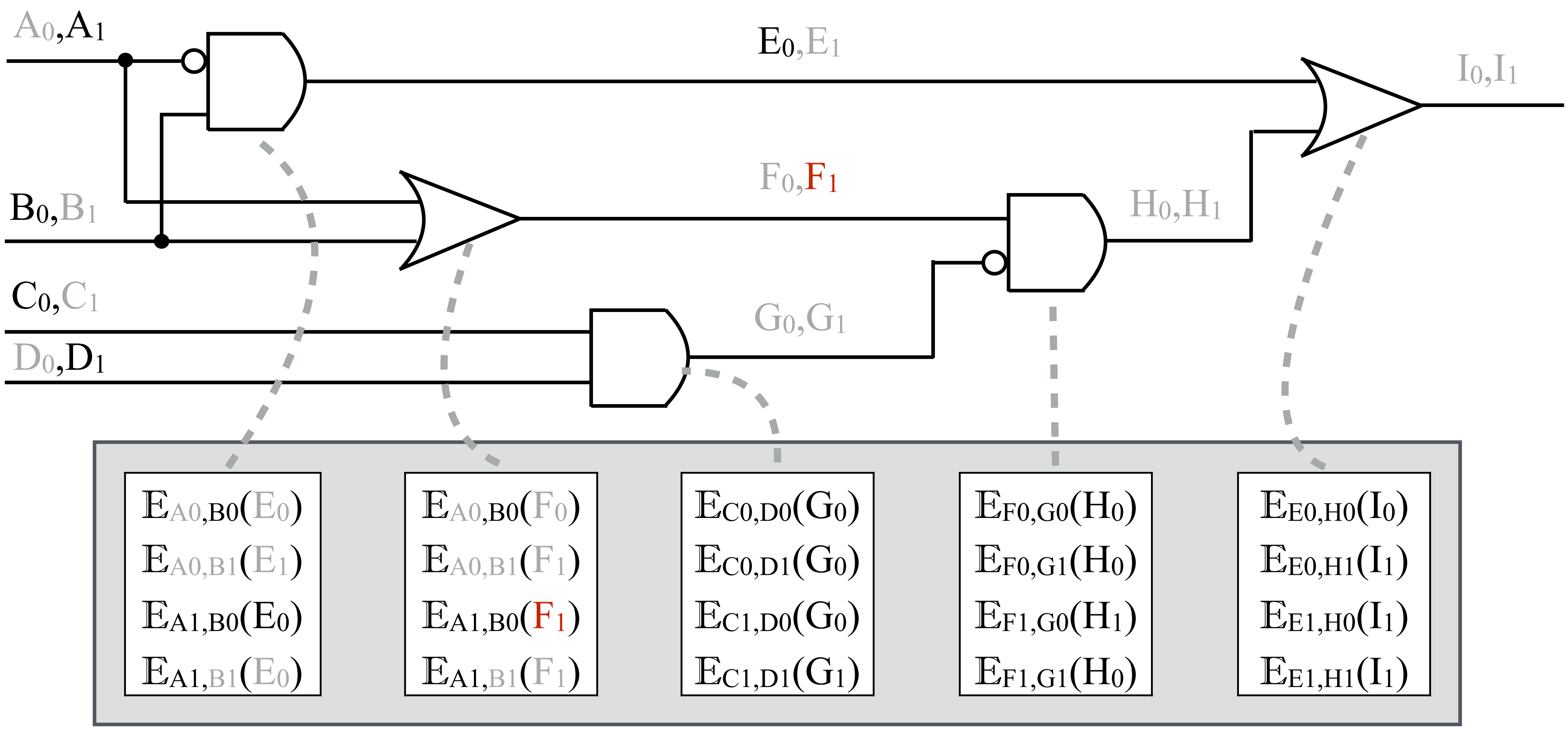
Eval(C_{garble},x_{garble}):

- Only one ciphertext per gate is decryptable



Garbled Circuits

[Yao86]

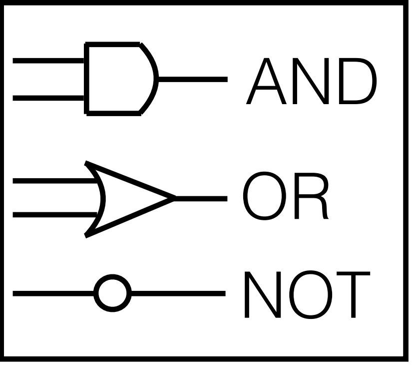


Garble(C,x):

- Pick random **labels** W_0,W_1 for each wire
- "Encrypt" truth table of each gate
- **Garbled Circuit** C_{garble}= all encrypted gates
- **Garbled Input** x_{garble}= one label per wire

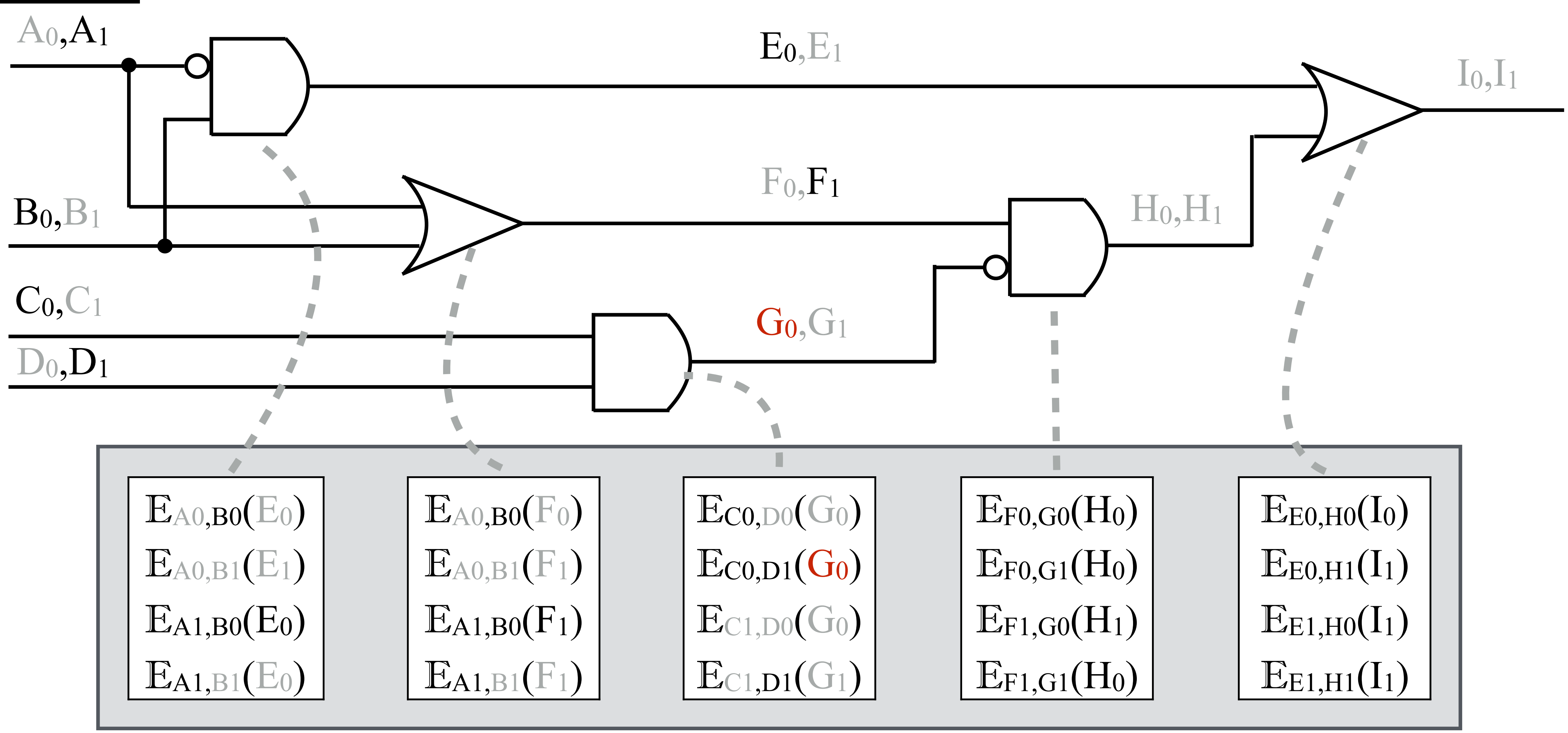
Eval(C_{garble},x_{garble}):

- Only one ciphertext per gate is decryptable
- Result of decryption = value on outgoing wire



Garbled Circuits

[Yao86]

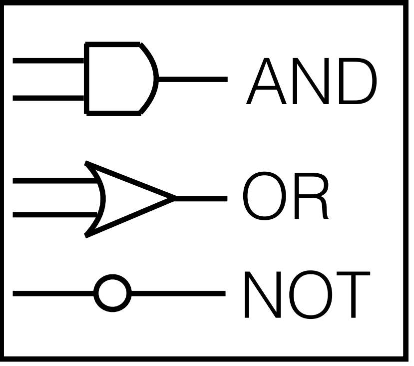


Garble(C,x):

- Pick random **labels** W_0,W_1 for each wire
- “Encrypt” truth table of each gate
- **Garbled Circuit** C_{garble}= all encrypted gates
- **Garbled Input** x_{garble}= one label per wire

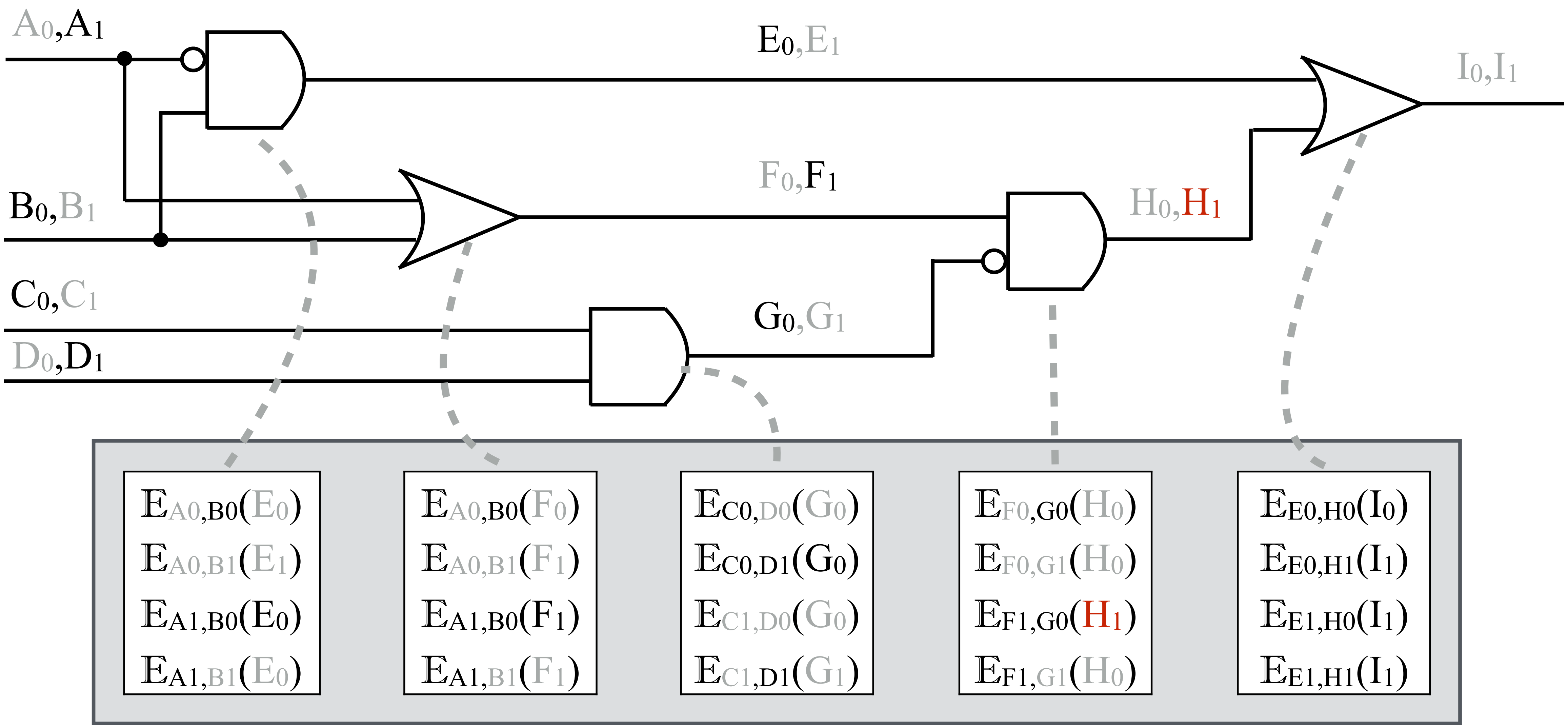
Eval(C_{garble},x_{garble}):

- Only one ciphertext per gate is decryptable
- Result of decryption = value on outgoing wire



Garbled Circuits

[Yao86]

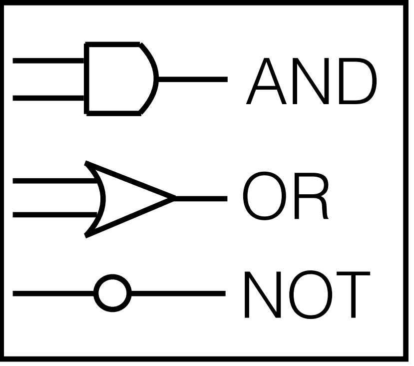


Garble(C,x):

- Pick random **labels** W_0, W_1 for each wire
- “Encrypt” truth table of each gate
- **Garbled Circuit** C_{garble}= all encrypted gates
- **Garbled Input** x_{garble}= one label per wire

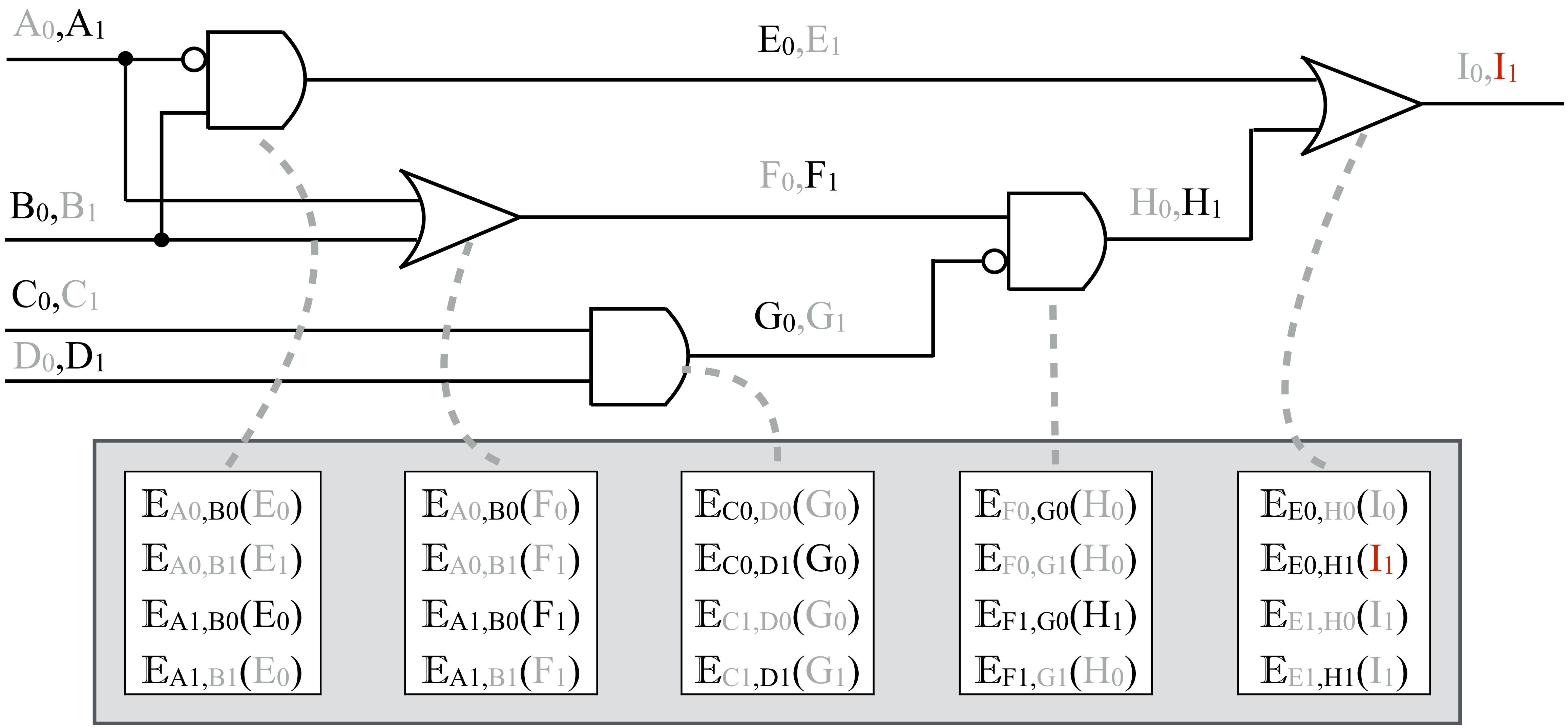
Eval(C_{garble},x_{garble}):

- Only one ciphertext per gate is decryptable
- Result of decryption = value on outgoing wire



Garbled Circuits

[Yao86]



Garble(C,x):

- Pick random **labels** W_0,W_1 for each wire
- "Encrypt" truth table of each gate
- **Garbled Circuit** C_{garble}= all encrypted gates
- **Garbled Input** x_{garble}= one label per wire

Eval(C_{garble},x_{garble}):

- Only one ciphertext per gate is decryptable
- Result of decryption = value on outgoing wire

The finer details

Privacy (intuition):

- For each wire (including input wires), adversary only sees **one** label W_b

The finer details

Privacy (intuition):

- For each wire (including input wires), adversary only sees **one** label W_b
- The 4 entries in each encrypted table are in random order

The finer details

Privacy (intuition):

- For each wire (including input wires), adversary only sees **one** label W_b
- The 4 entries in each encrypted table are in random order
- Adversary tries to decrypt each entry. Only one decryption succeeds.

The finer details

Privacy (intuition):

- For each wire (including input wires), adversary only sees **one** label W_b
- The 4 entries in each encrypted table are in random order
- Adversary tries to decrypt each entry. Only one decryption succeeds.
- Adversary has no idea whether $b=0$ or $b=1$ for any label W_b

The finer details

Privacy (intuition):

- For each wire (including input wires), adversary only sees **one** label W_b
- The 4 entries in each encrypted table are in random order
- Adversary tries to decrypt each entry. Only one decryption succeeds.
- Adversary has no idea whether $b=0$ or $b=1$ for any label W_b

Interpreting the output:

- For every output wire, reveal the mappings (b, W_b)

Secure Computation from Garbled Circuits

Goal: Compute $f(x,y)$

x

y

Secure Computation from Garbled Circuits

Goal: Compute $f(x,y)$

1. Garbled circuit f_{garble}
2. Garbled input x_{garble}

x

y

Secure Computation from Garbled Circuits

Goal: Compute $f(x,y)$

1. Garbled circuit f_{garble}
2. Garbled input x_{garble}

x

y

Problem: How to transmit y_{garble} ?

Secure Computation from Garbled Circuits

Goal: Compute $f(x,y)$

1. Garbled circuit f_{garble}
2. Garbled input x_{garble}

x

y

y

All Labels for 2nd input

y_{garble}

Secure Computation from Garbled Circuits

Goal: Compute $f(x,y)$

1. Garbled circuit f_{garble}
2. Garbled input x_{garble}

x

y

y

All Labels for 2nd input

y_{garble}

Want:

- Alice learns nothing about y

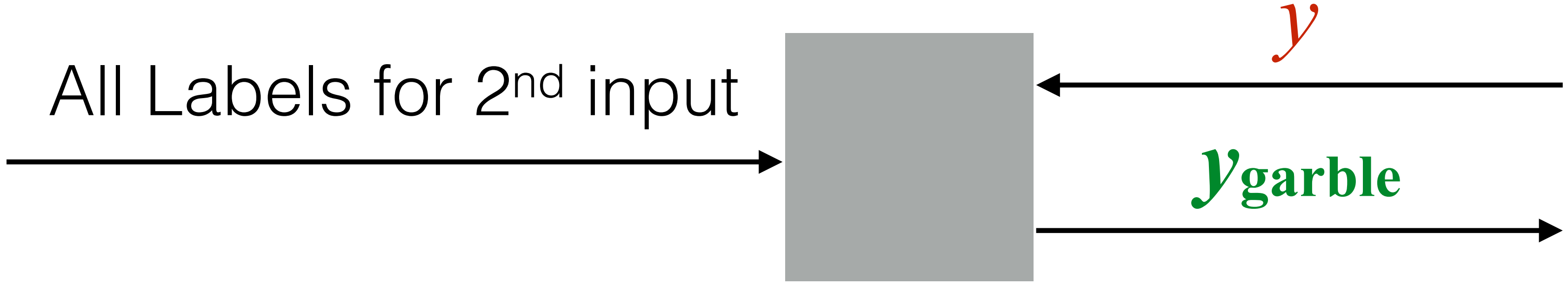
Secure Computation from Garbled Circuits

<u>Goal</u>: Compute $f(x,y)$

1. Garbled circuit f_{garble}
2. Garbled input x_{garble}

x

y

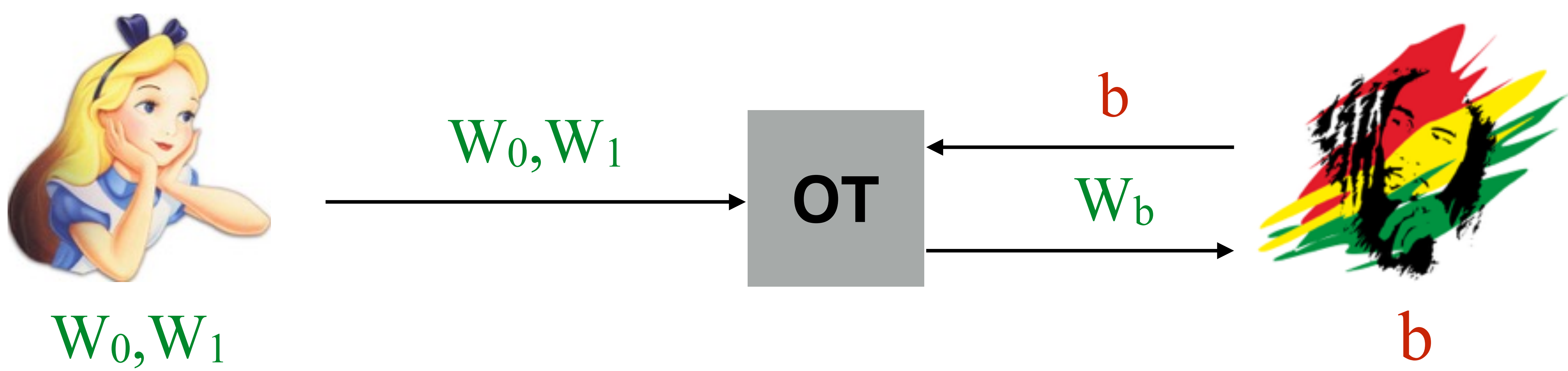


Want:

- Alice learns nothing about y
- Bob does not learn the other labels

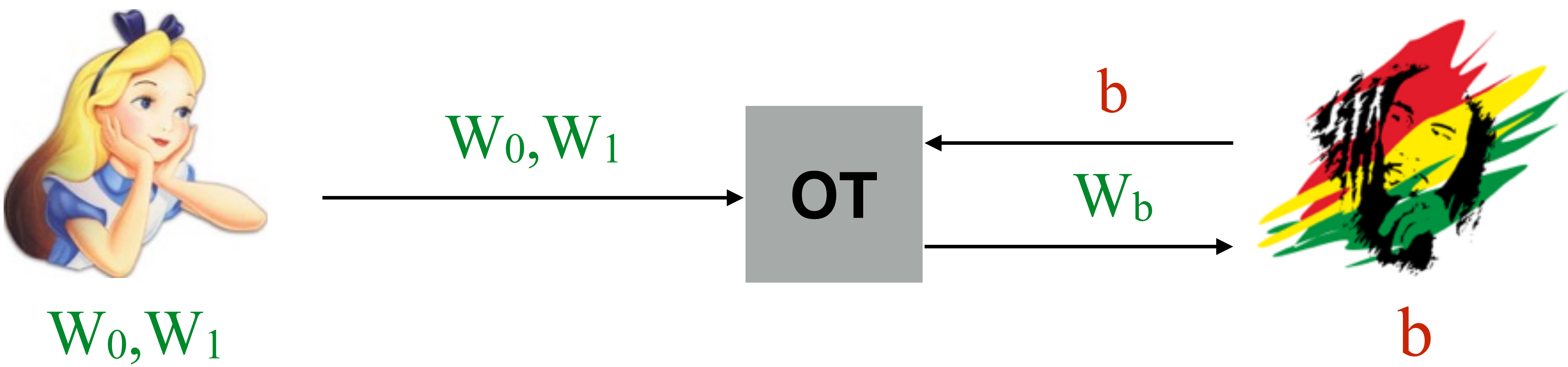
Oblivious Transfer

[Rabin81,Even-Goldreich-Lempel-85]



Oblivious Transfer

[Rabin81,Even-Goldreich-Lempel-85]



Want:

- Alice does not learn b
- Bob does not learn W_{1-b}