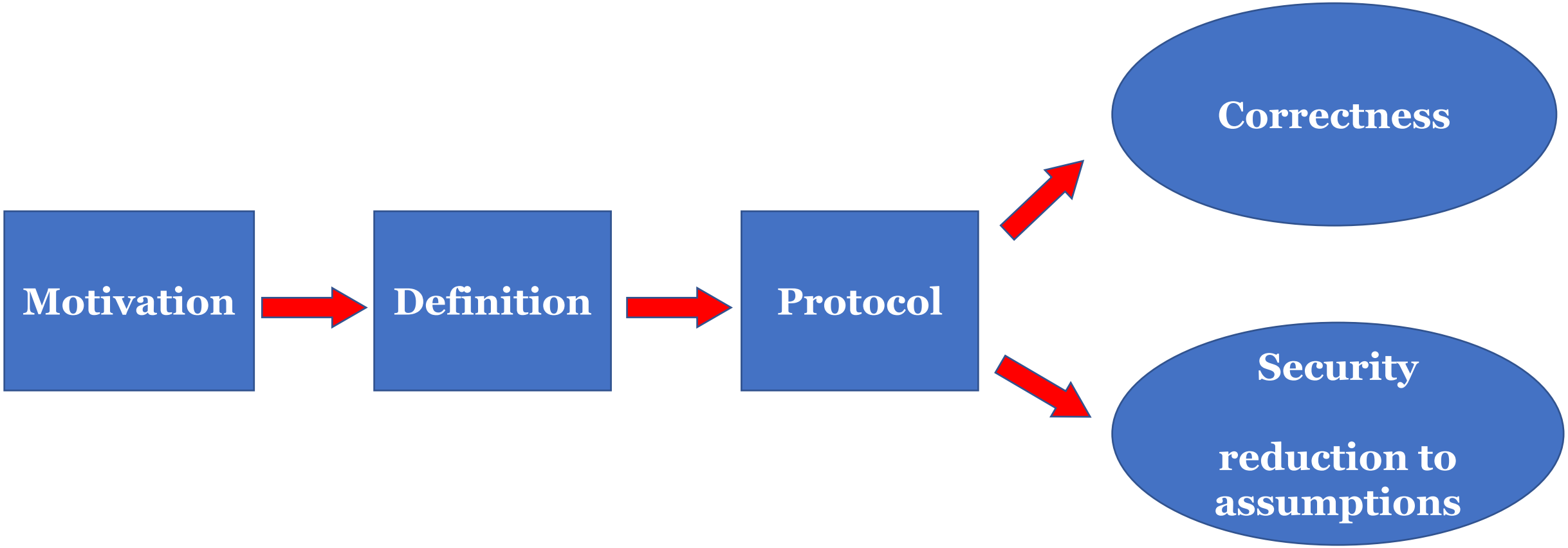
Secure Multi-Party Computation

Cryptography



I don't want to talk to you anymore.

What do you do after work?

Ideal scenario

Let’s run a cryptographic protöcöl.

We both like sports!

What do you do after work?

Set A

Set B

Set A∩B

Other elements remain secret to each other

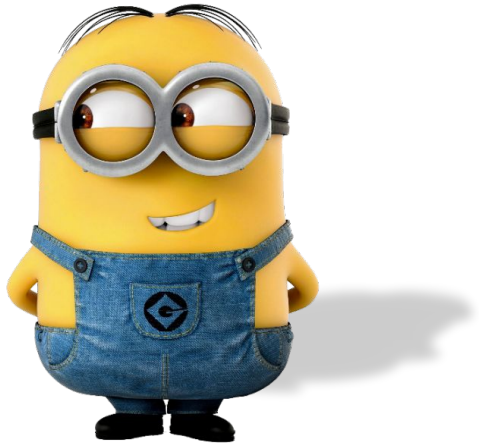
Secure two-party computation

f

x

$f(x, y)$

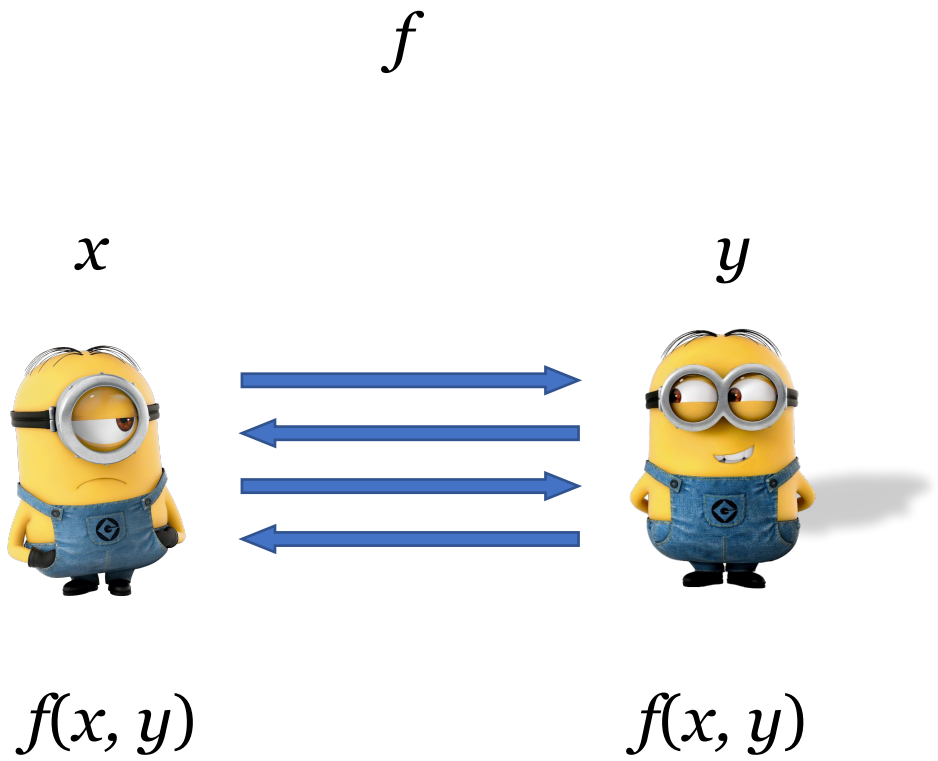
y

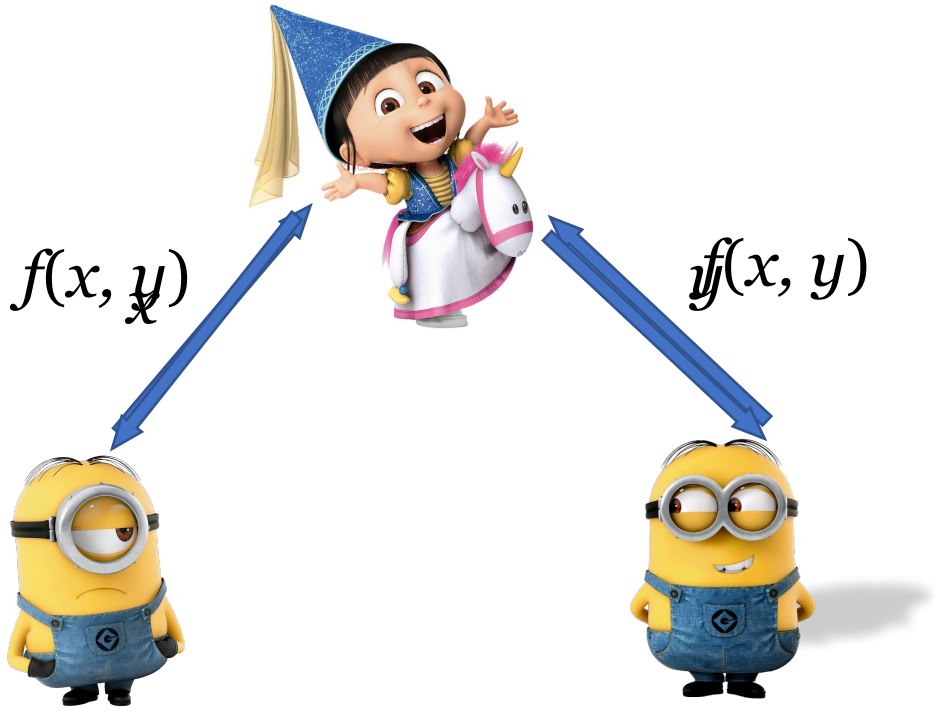


$f(x, y)$

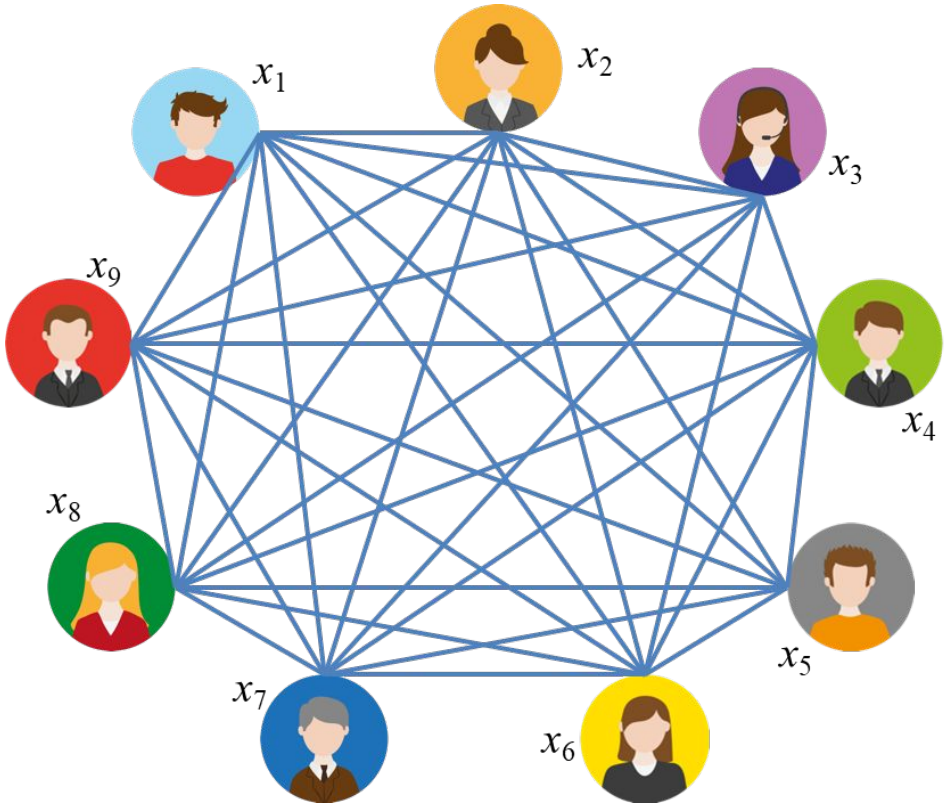
x and y remain secret

Secure two-party computation

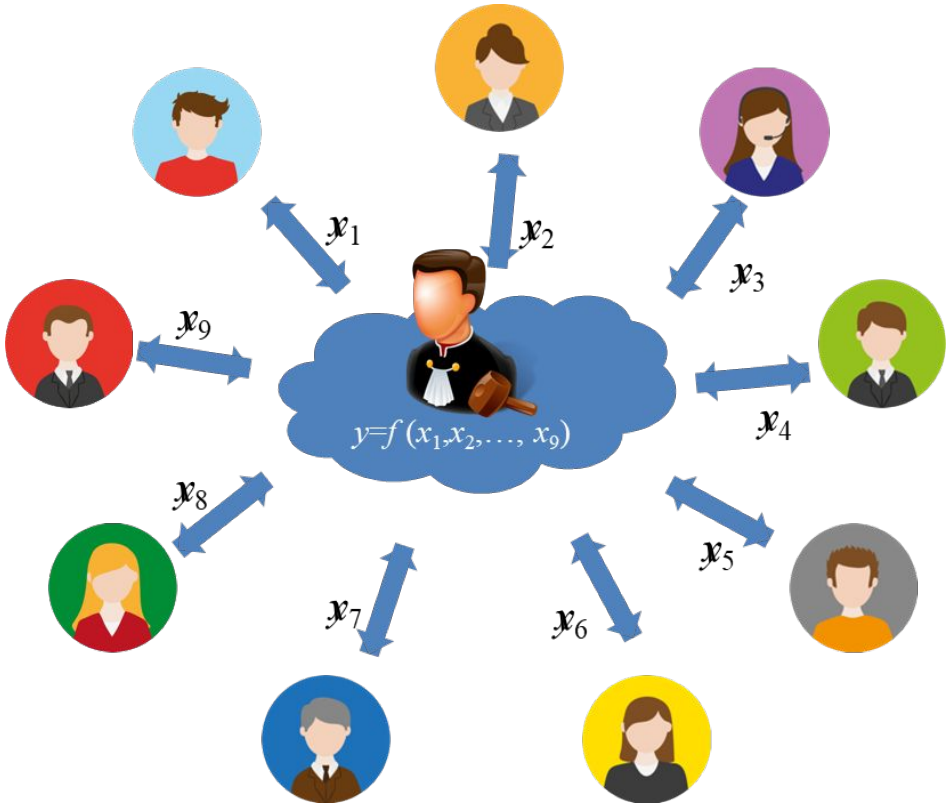




Secure multi-party computation (MPC)



$y=f(x_1,x_2,\ldots,x_9)$



Motivations: old days

- Millionaire problem
- Mental poker

Yao, Andrew Chi-Chih,
“Protocols for Secure Computations”,
Foundations of Computer Science (FOCS) 1982

Motivations: now

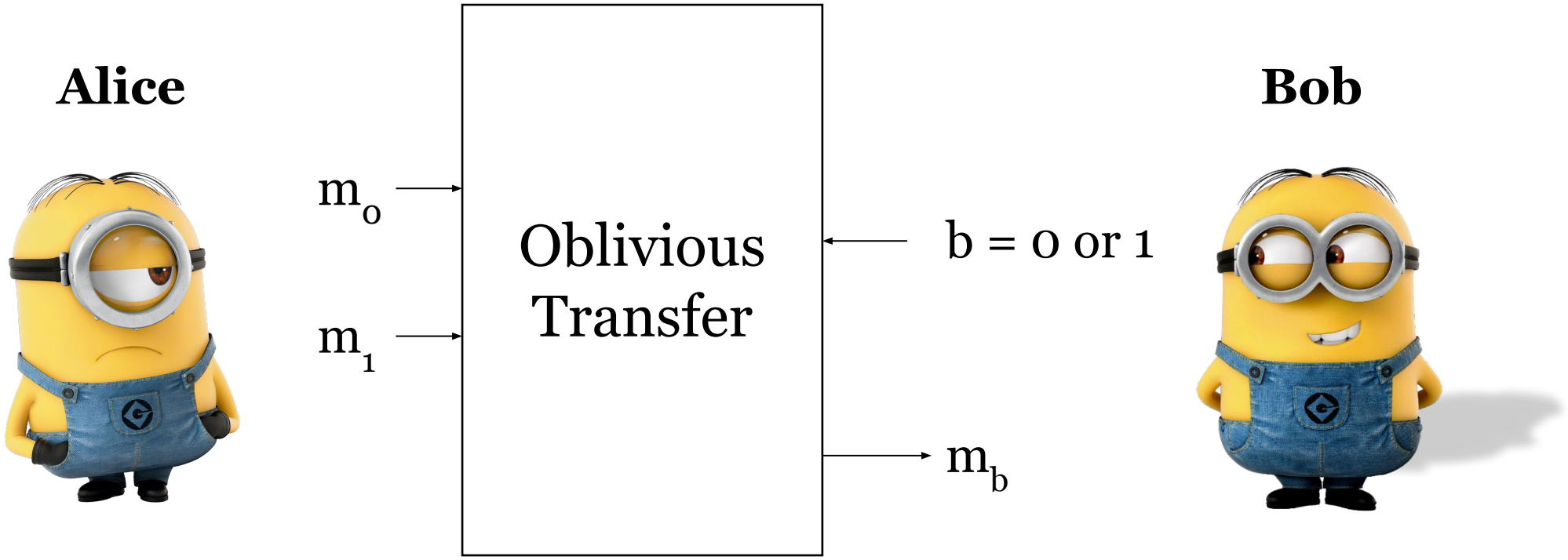
- Auction
- Voting
- Privacy-preserving data mining/machine learning
- Distributed encryptions/signatures

Existing solutions

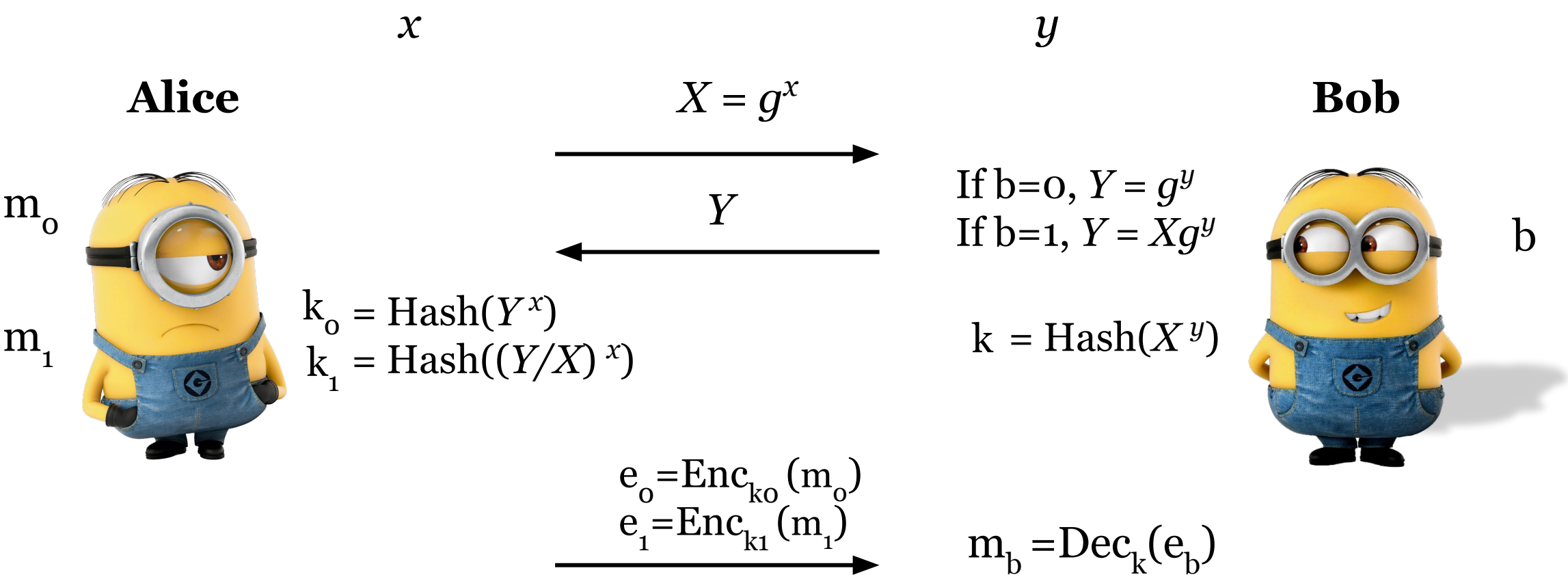
- Yao’s garbled circuit

- GMW protocol

Building block: oblivious transfer (OT)



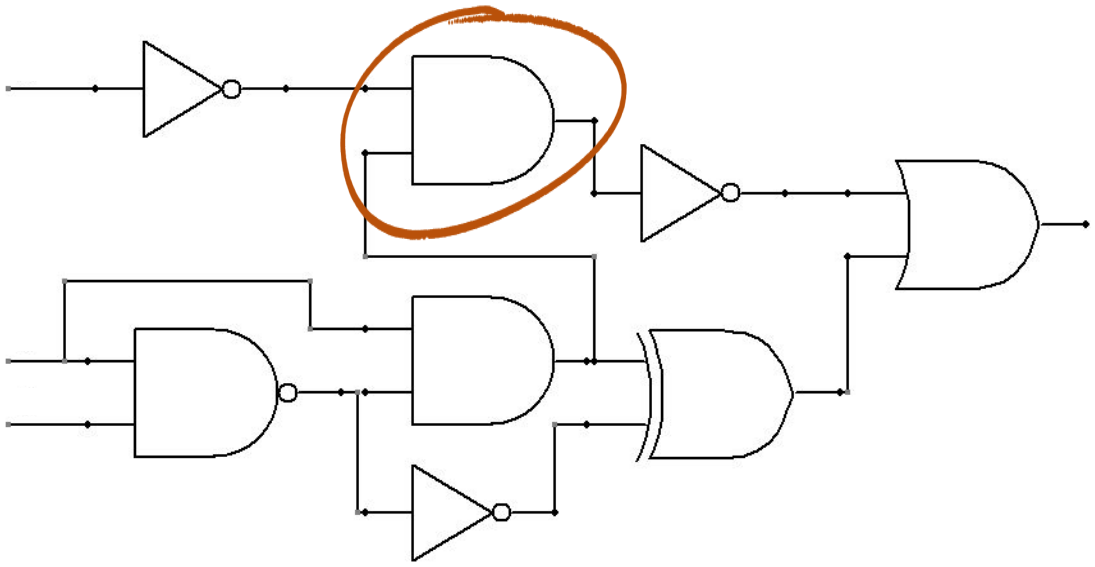
Building block: oblivious transfer (OT)



Yao’s garbled circuit

Yao's garbled circuit

- What is a (Boolean) circuit?



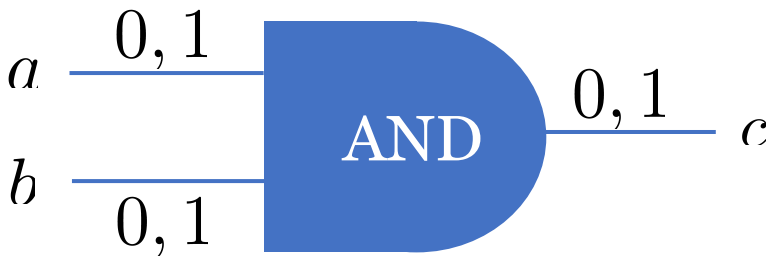
Wire a	Wire b	Wire c
0	0	0
0	1	0
1	0	0
1	1	1

Yao’s garbled circuit

- Why circuit?
  - Captures all functions
  - Gate-by-gate paradigm

Yao's garbled circuit

- How to garble a circuit?



Wire a	Wire b	Wire c
0	0	0
0	1	0
1	0	0
1	1	1

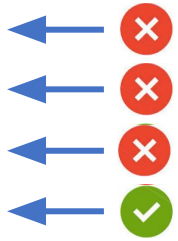
Yao's garbled circuit

- How to garble a circuit?

$A_1, B_0 \rightarrow C_0$

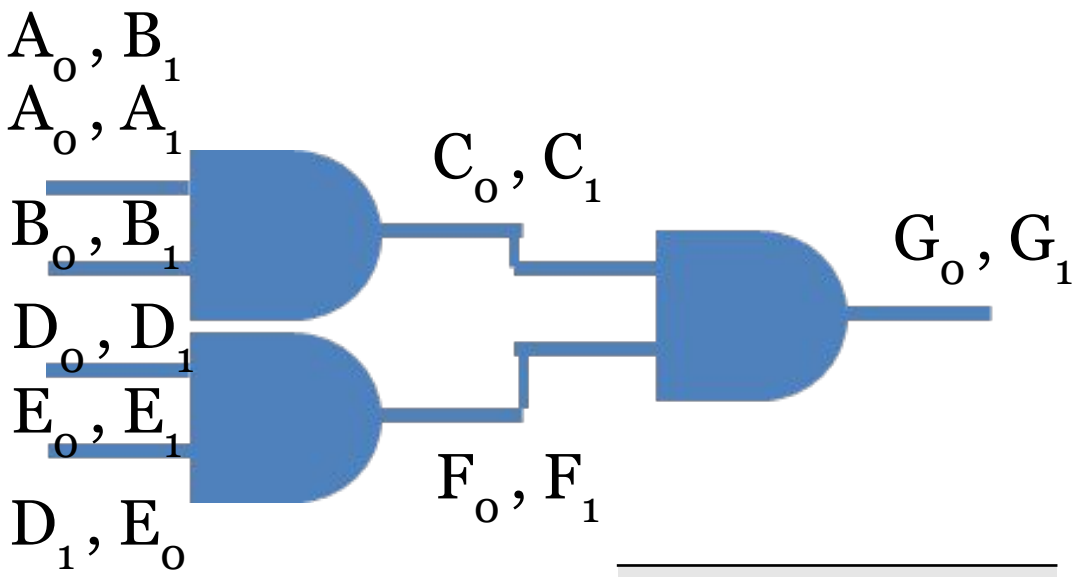
$A_1, B_1 \rightarrow C_1$

Wire a	Wire b	Wire c
0	0	0
0	1	0
1	0	0
1	1	1



Yao's garbled circuit

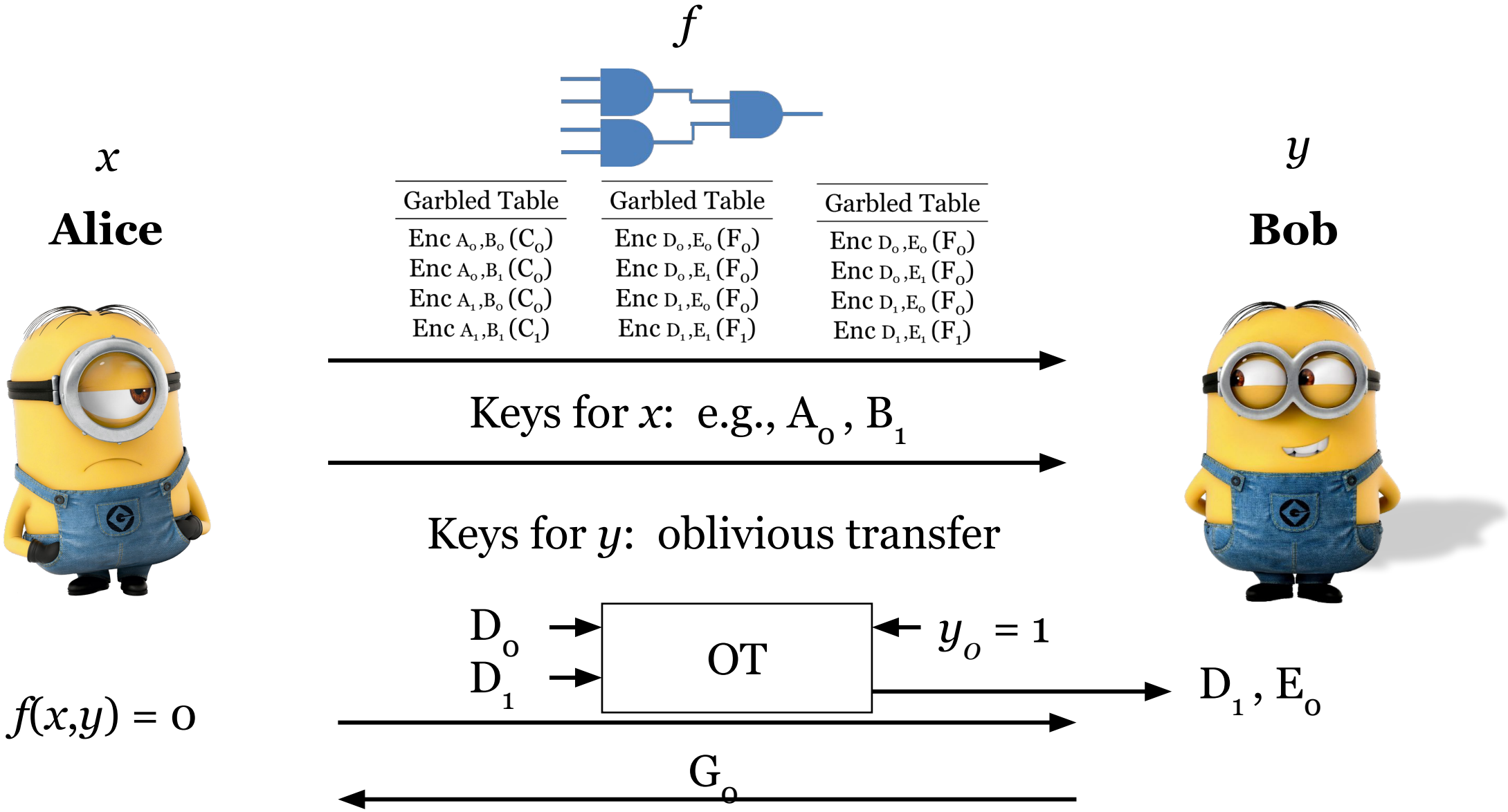
Garbled Table
$Enc_{A_0,B_0}(C_0)$
$Enc_{A_0,B_1}(C_0)$ ✓
$Enc_{A_1,B_0}(C_0)$
$Enc_{A_1,B_1}(C_1)$



Garbled Table
$Enc_{D_0,E_0}(F_0)$
$Enc_{D_0,E_1}(F_0)$
$Enc_{D_1,E_0}(F_0)$ ✓
$Enc_{D_1,E_1}(F_1)$

Garbled Table
$Enc_{C_0,F_0}(G_0)$ ✓
$Enc_{C_0,F_1}(G_0)$
$Enc_{C_1,F_0}(G_0)$
$Enc_{C_1,F_1}(G_1)$

Putting everything together



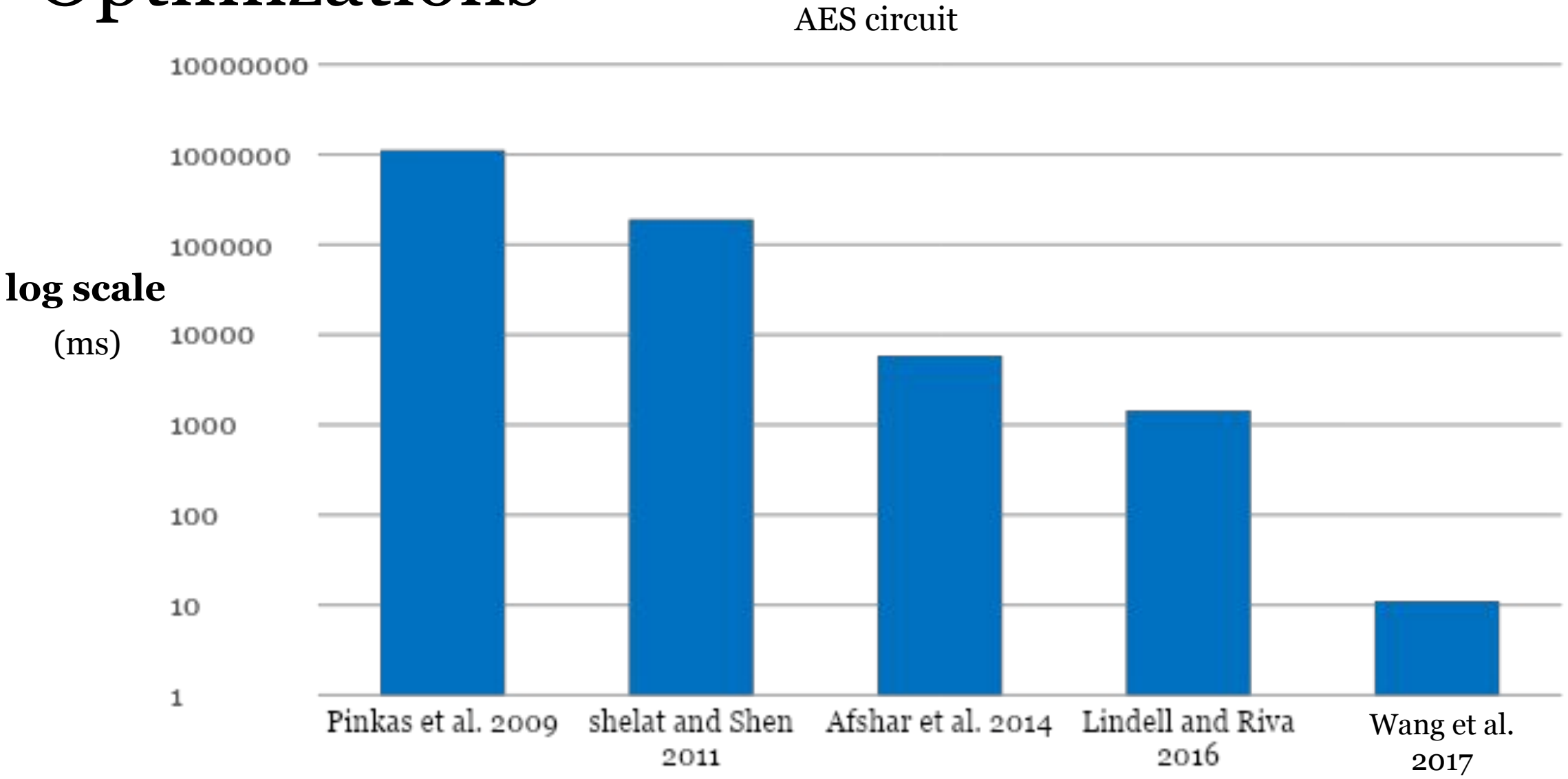
Properties of Yao's garbled circuit

- Constant round

- Boolean circuits

  Why not larger domain?

Optimizations



Point and permute

Garbled Table
$\text{Enc}_{A_0,B_0}(C_0)$
$\text{Enc}_{A_0,B_1}(C_0)$
$\text{Enc}_{A_1,B_0}(C_0)$
$\text{Enc}_{A_1,B_1}(C_1)$

Solution: append the “select bit” to the keys

Row reduction

Garbled Table
$\text{Hash}(A_0 \| \| B_0) \oplus (C_0)$
$\text{Hash}(A_1 \| \| B_0) \oplus (C_0)$
$\text{Hash}(A_0 \| \| B_1) \oplus (C_0)$
$\text{Hash}(A_1 \| \| B_1) \oplus (C_1)$

$A_1, B_0 \rightarrow C_0$

$$C_0 = \text{Hash}(A_0 \| \| B_0)$$

Free XOR

XOR + AND is universal: construct NAND!!!

Garbled Table
$\text{Enc}_{A_0,B_0}(C_0)$
$\text{Enc}_{A_0,B_1}(C_0)$
$\text{Enc}_{A_1,B_0}(C_0)$
$\text{Enc}_{A_1,B_1}(C_1)$

A_0, A_1 $\quad$ $A, A \oplus R$

B_0, B_1 $\quad$ $B, B \oplus R$

Half gates

- 1 input known to 1 party
  - Sender (garbler) knows 1 input
  - Receiver (evaluator) knows 1 input

- $c = a \wedge b = a \wedge (r \oplus r \oplus b) = a \wedge r \oplus a \wedge (r \oplus b)$