Secure Multiparty Computation

CS/ECE 407

Today's objectives

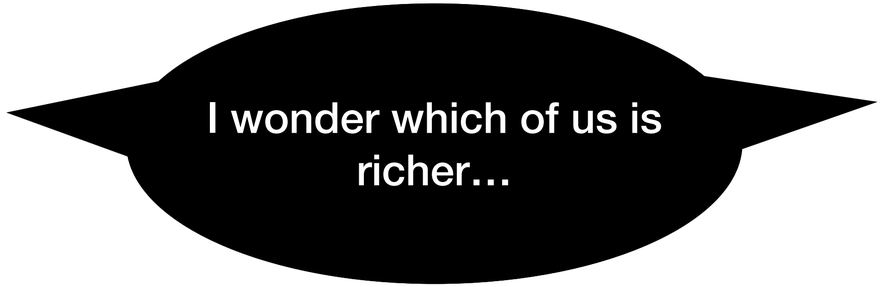
Introduce the notion of a secure computation

Define and construct oblivious transfer

Connect secure computation with secret sharing

Construct a general-purpose MPC protocol

Yao’s Millionaire Problem



$13m

$45m

Yao’s Millionaire Problem

I wonder which of us is richer…

$13m

You tell me how much money you have, and I’ll tell you the answer!

$45m

Yao's Millionaire Problem

I wonder which of us is richer…

$13m

No way! You tell me how much money *you* have!

You tell me how much money you have, and I'll tell you the answer!

$45m

Yao’s Millionaire Problem

$13m

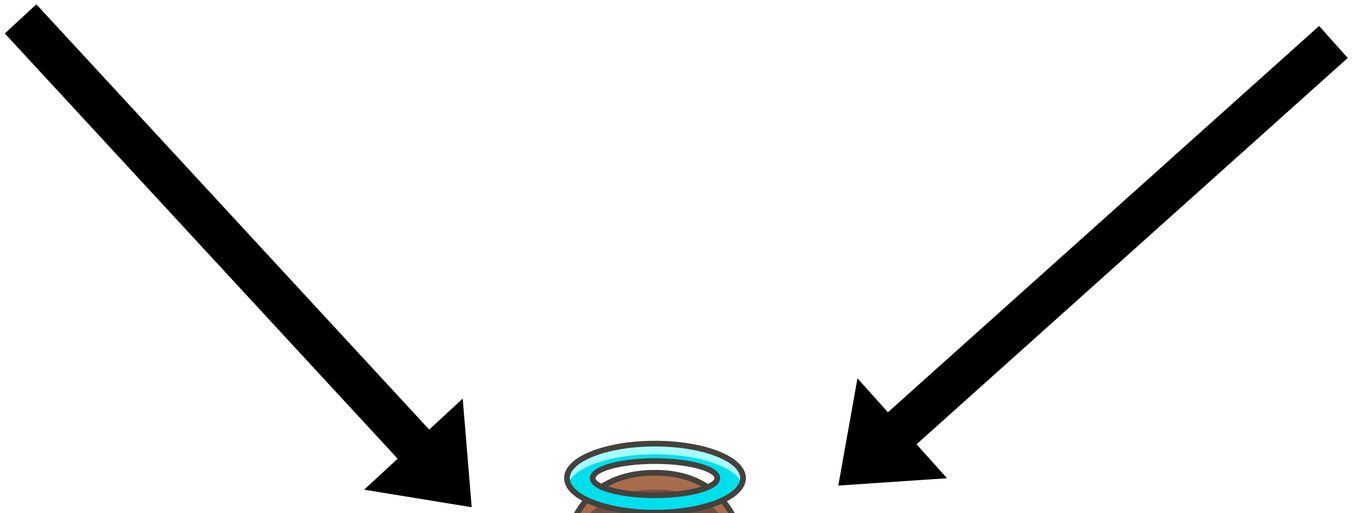
$45m

Incorruptible,
trusted third party

Yao's Millionaire Problem

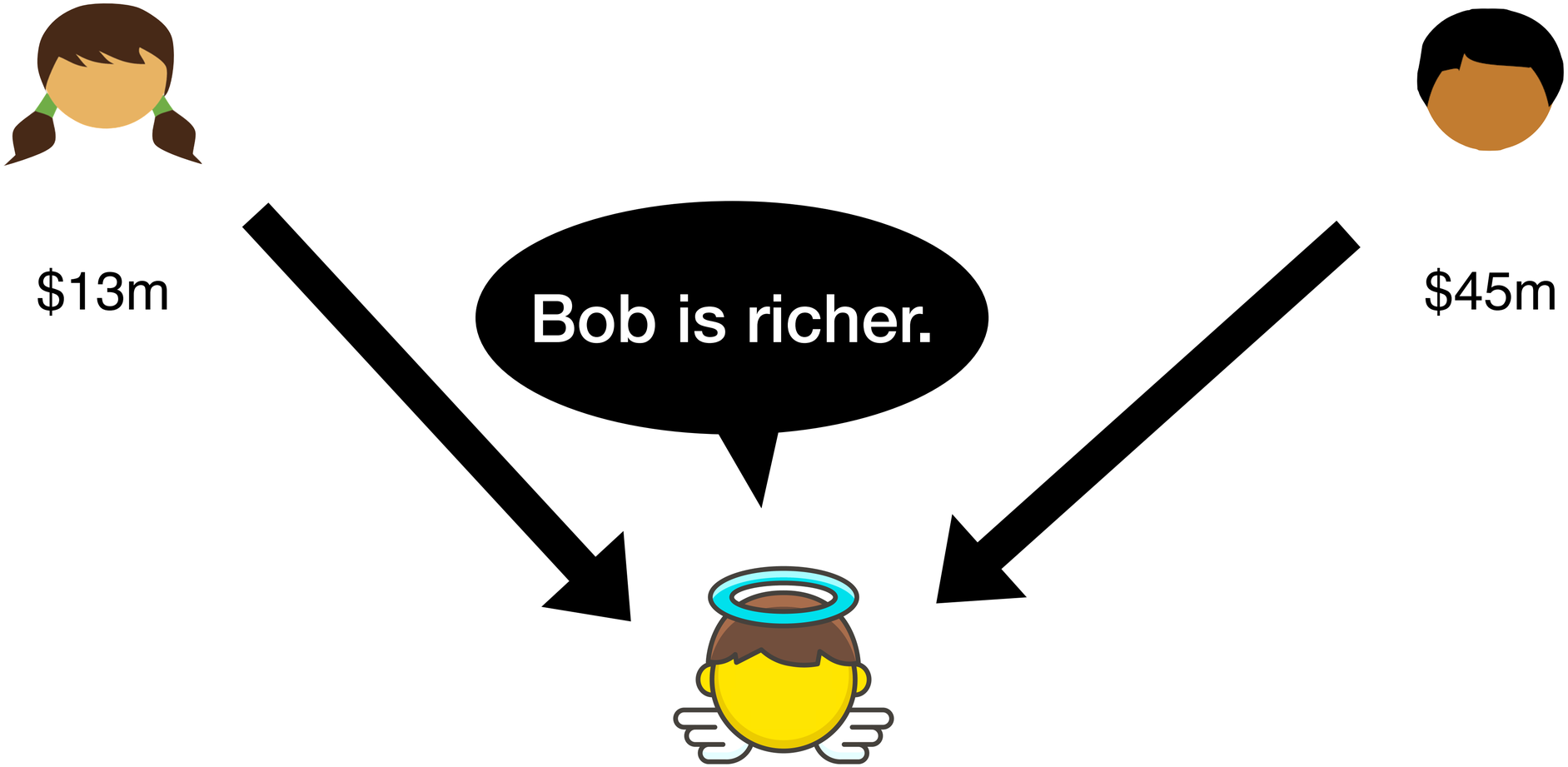
$13m

$45m

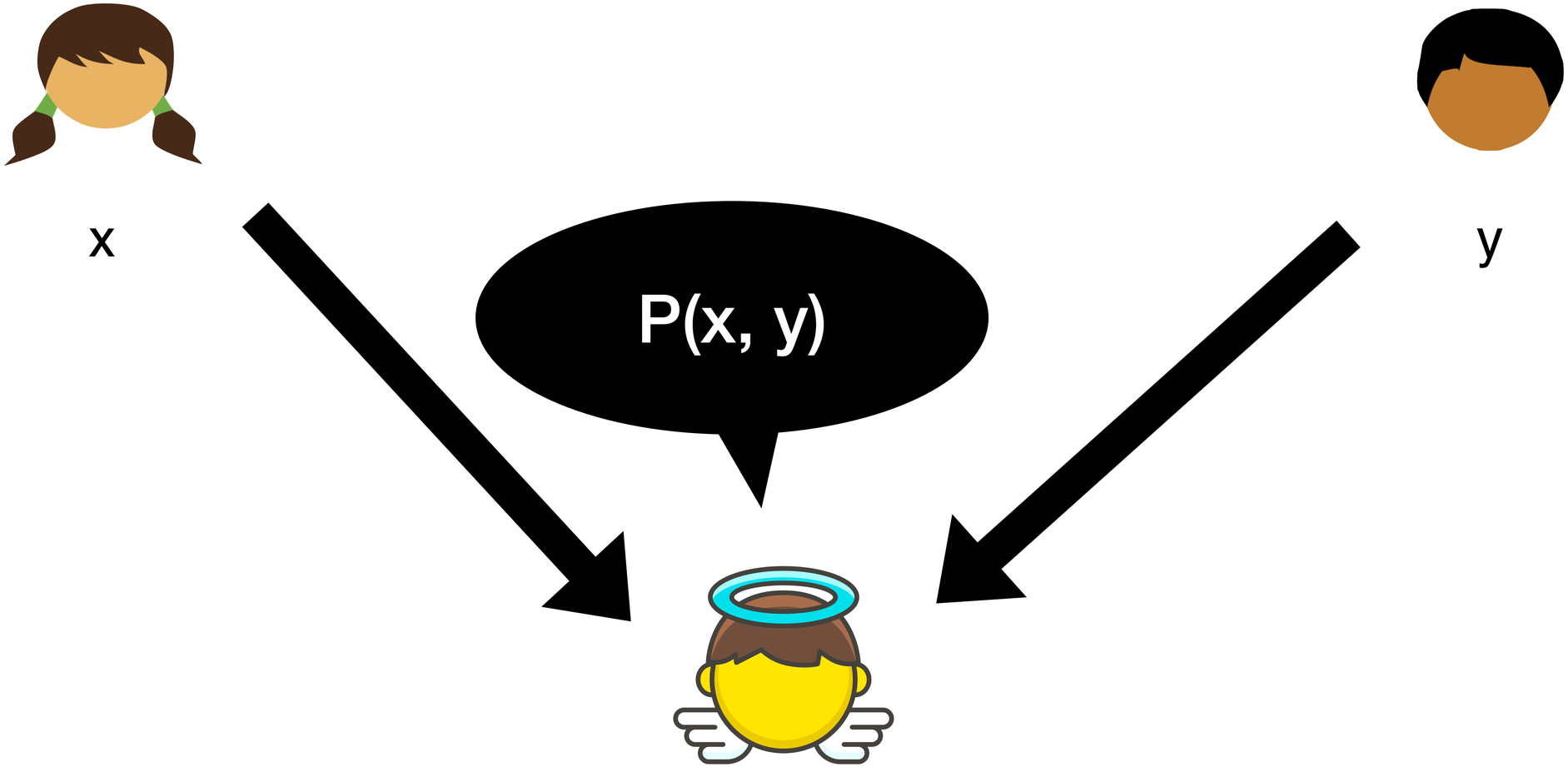


Incorruptible,
trusted third party

Yao's Millionaire Problem



MPC



MPC

x

y

P(x, y)

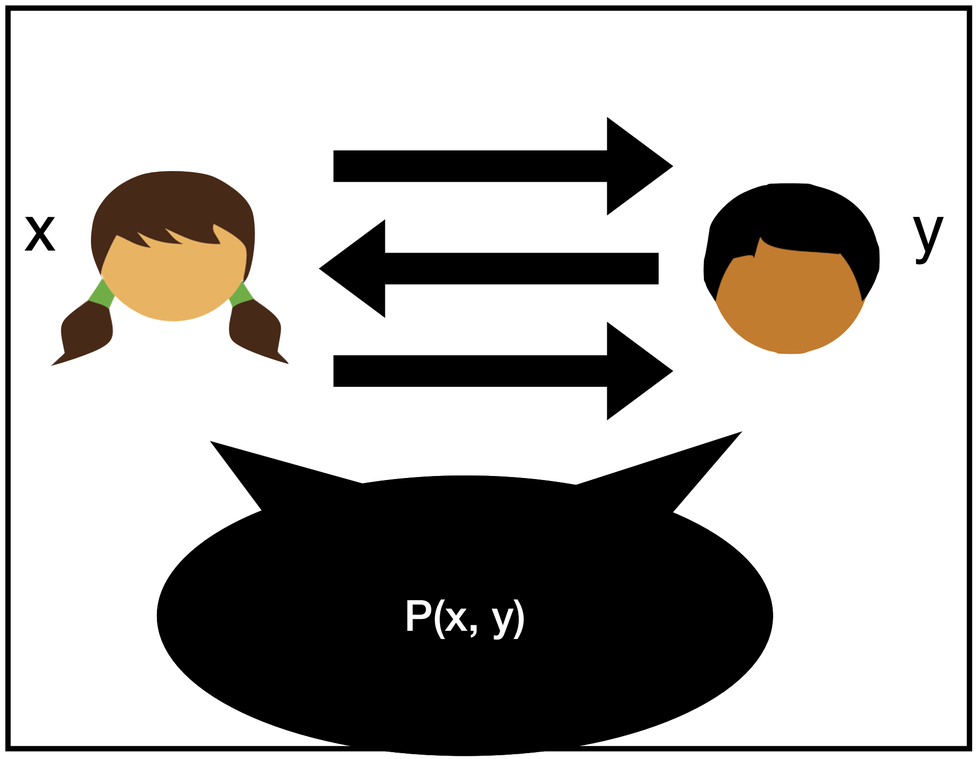
Incorruptible,
trusted third party

We have protocols for any program P

Privacy-preserving

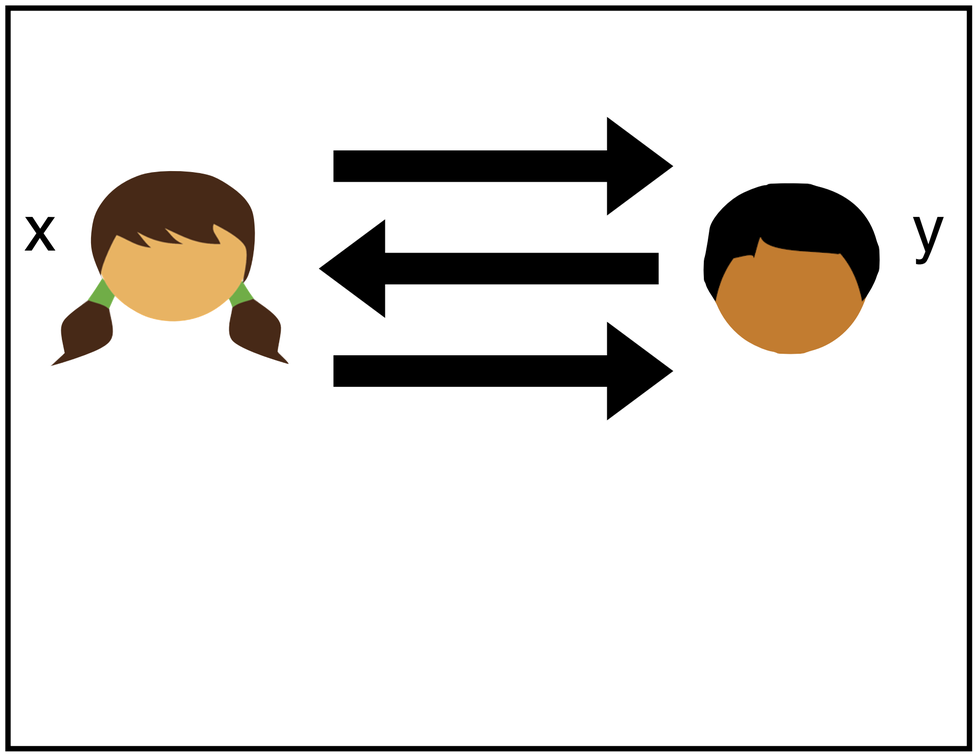
- auctions
- analytics
- contact discovery
- fraud detection
- …

MPC informally

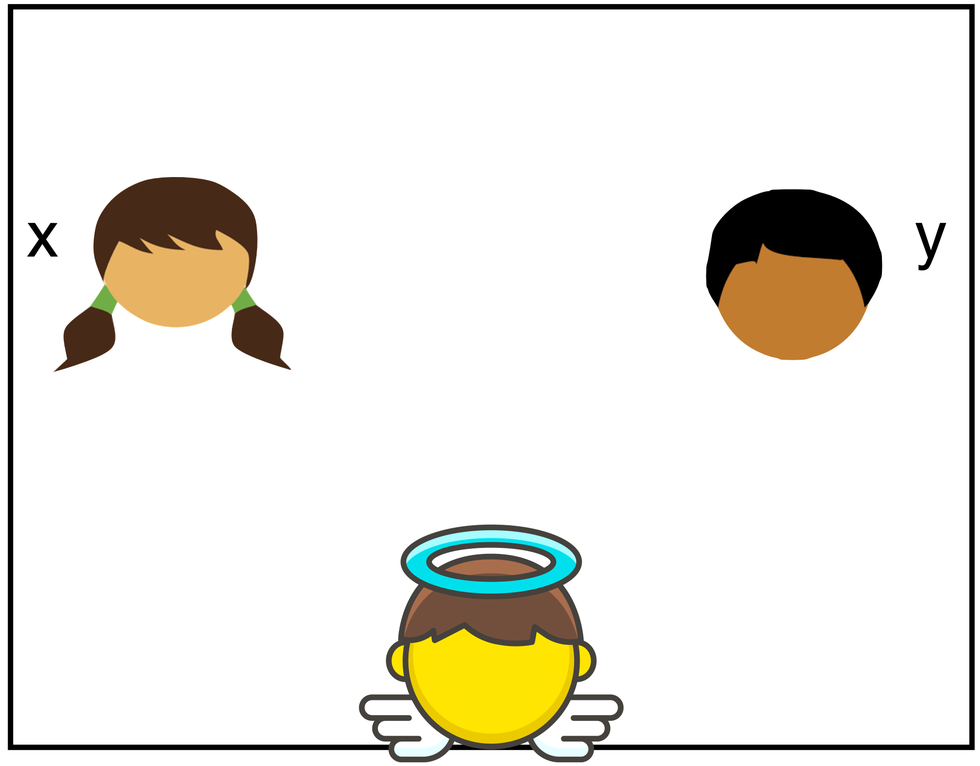


≈

MPC informally

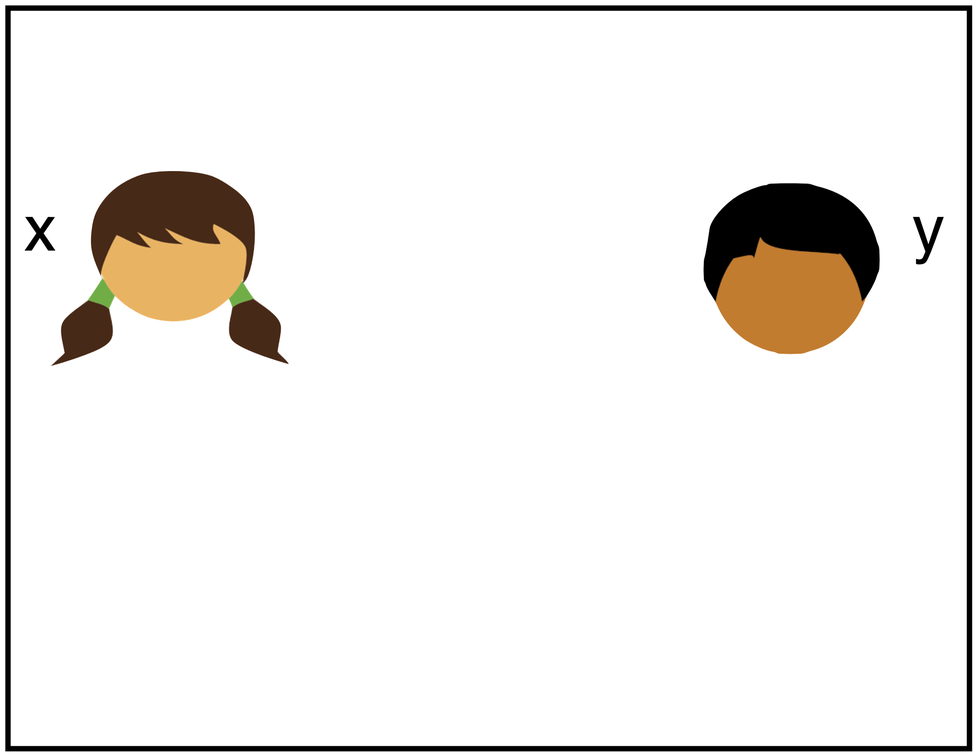


≈



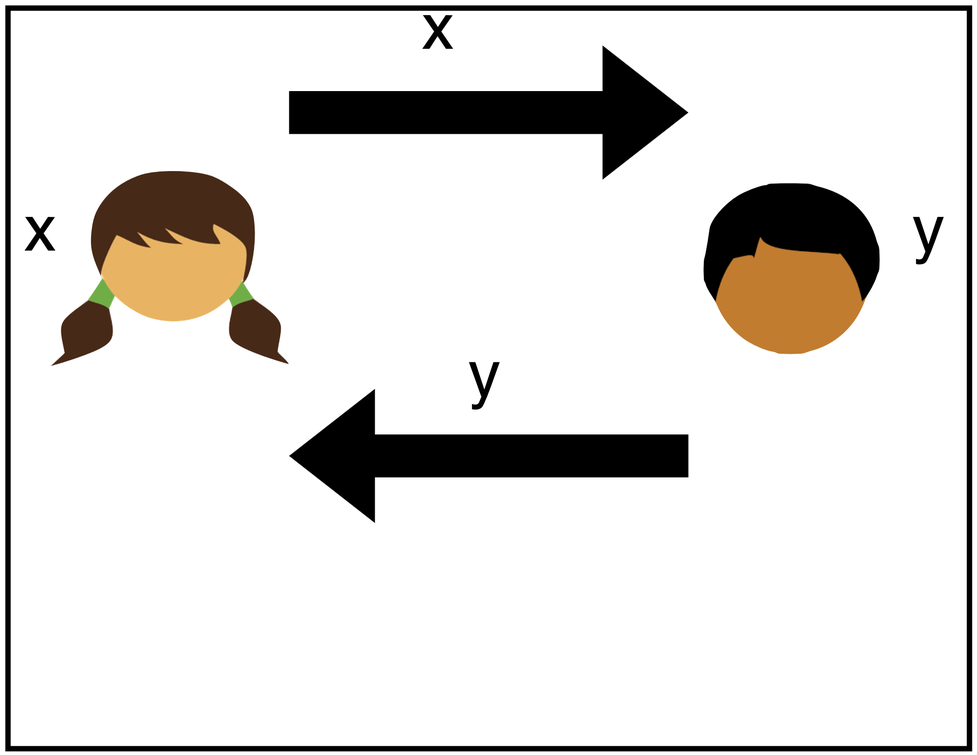
First semi-honest secure protocol

$$P(x, y) = x \oplus y$$



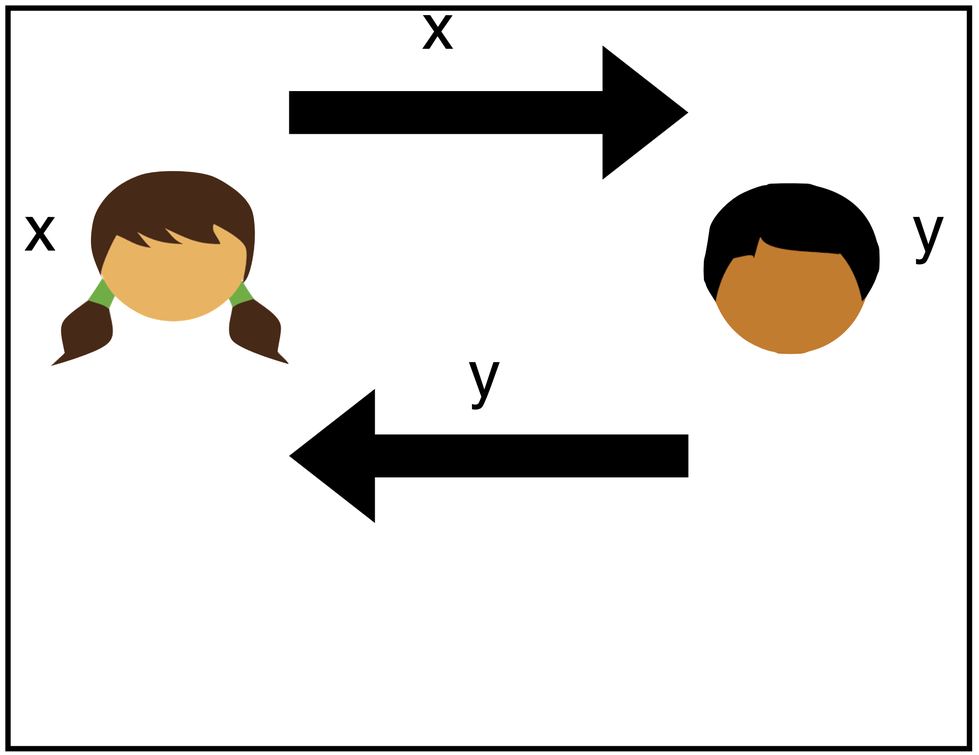
First semi-honest secure protocol

$$P(x, y) = x \oplus y$$

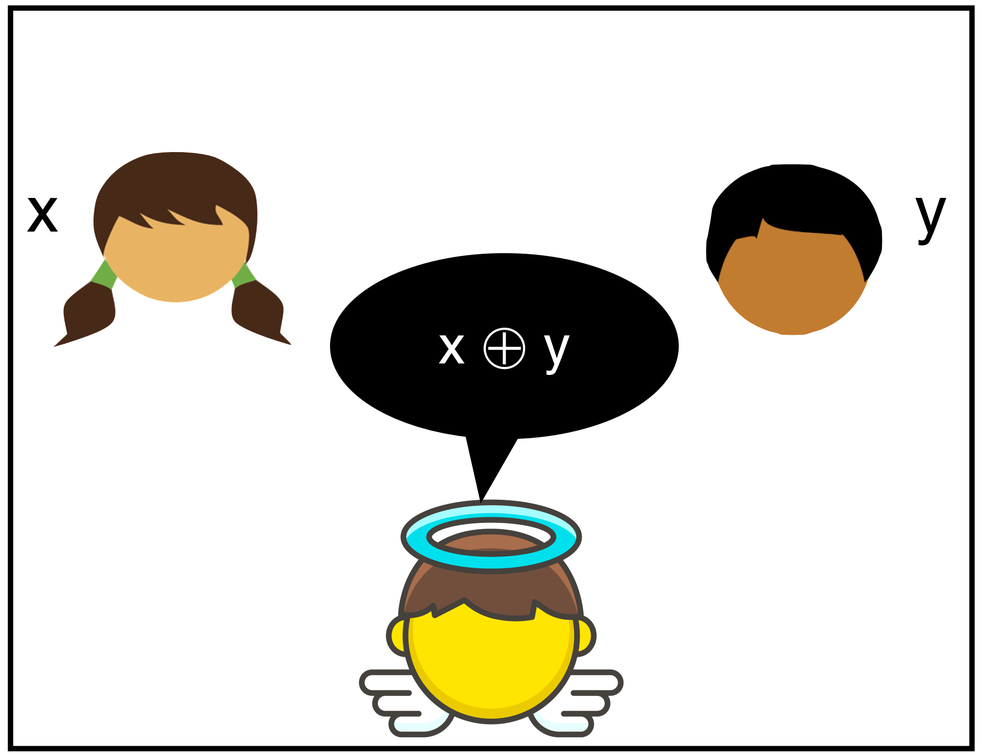


First semi-honest secure protocol

$$P(x, y) = x \oplus y$$



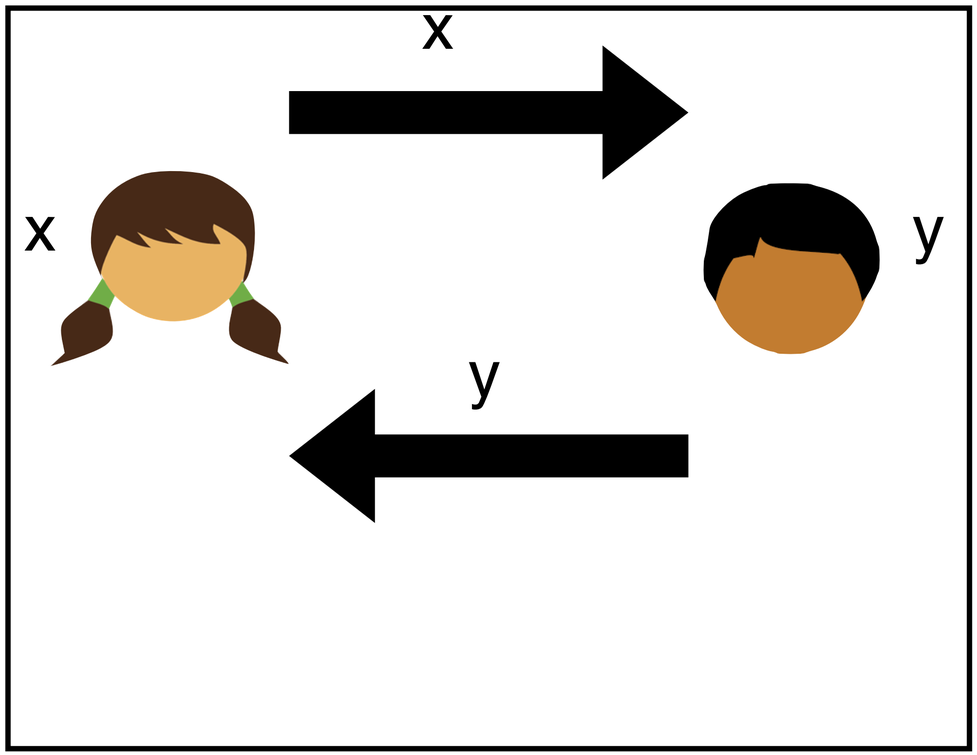
≈



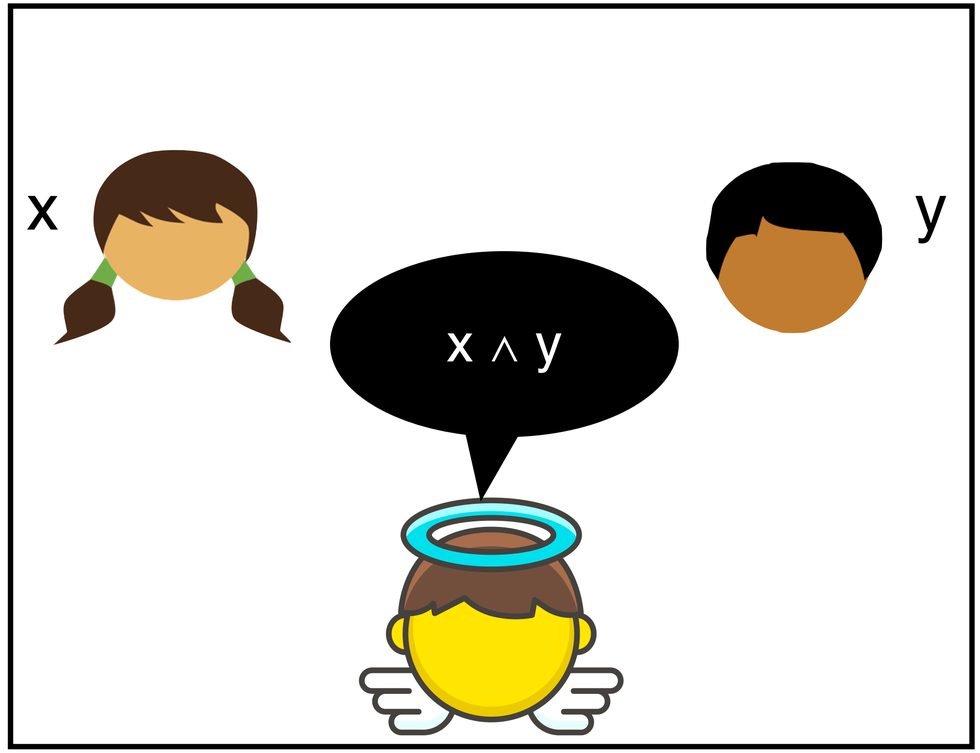
*The protocol is secure if everything Alice and Bob see in the real world can be **simulated** from what they see in the ideal world*

Is this secure?

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

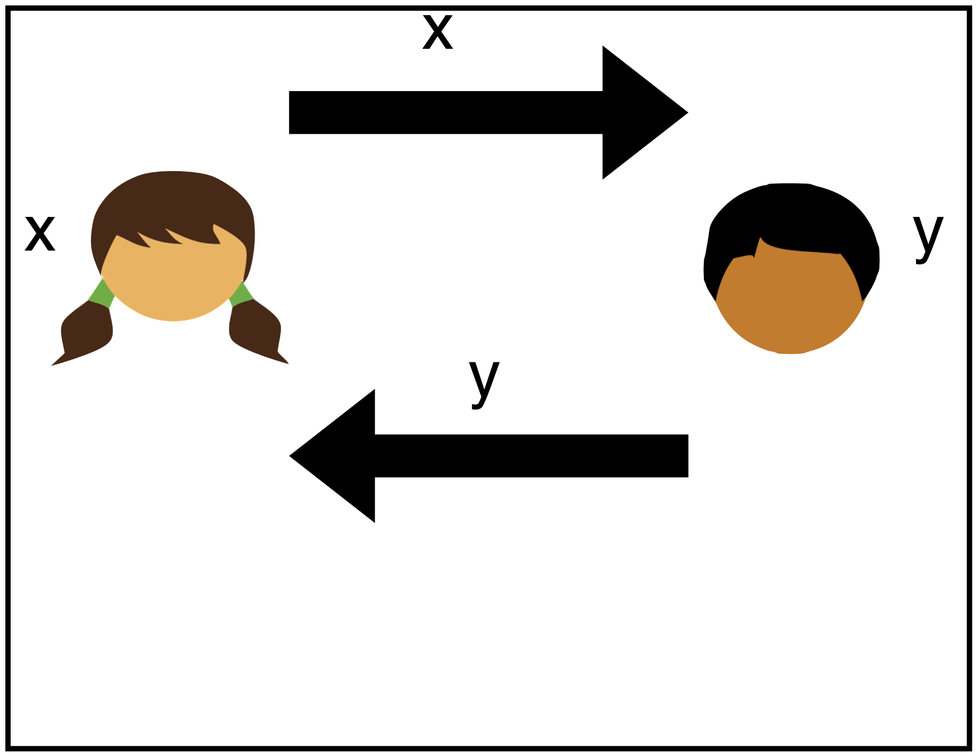


≈

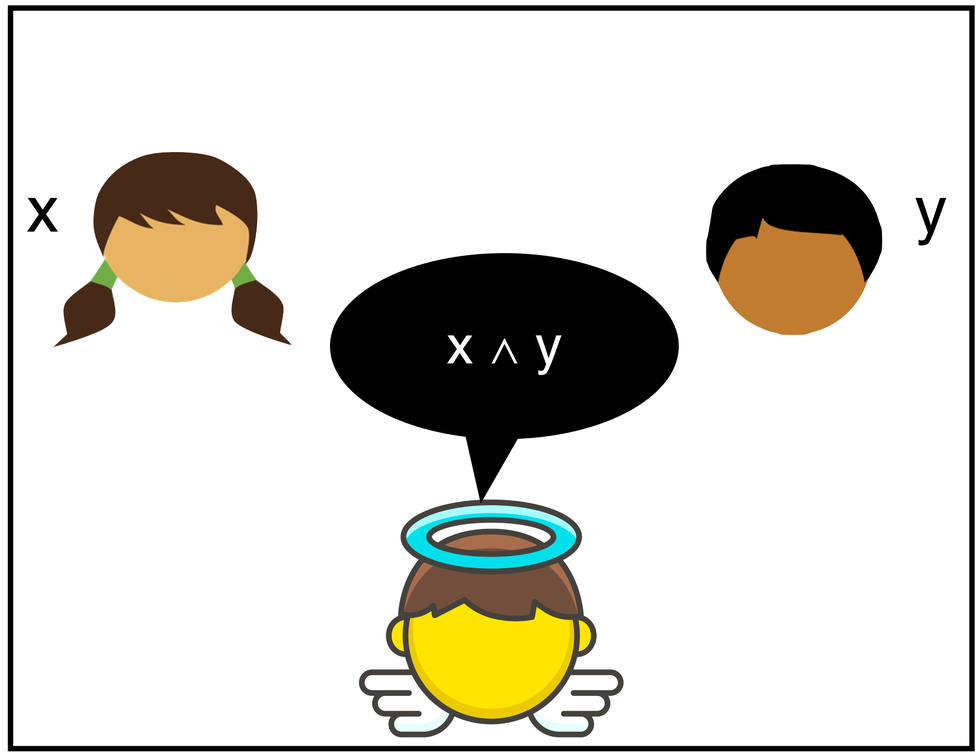


Is this secure?

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



≈



Oblivious Transfer

Sender

Receiver

Oblivious Transfer

m_0, m_1

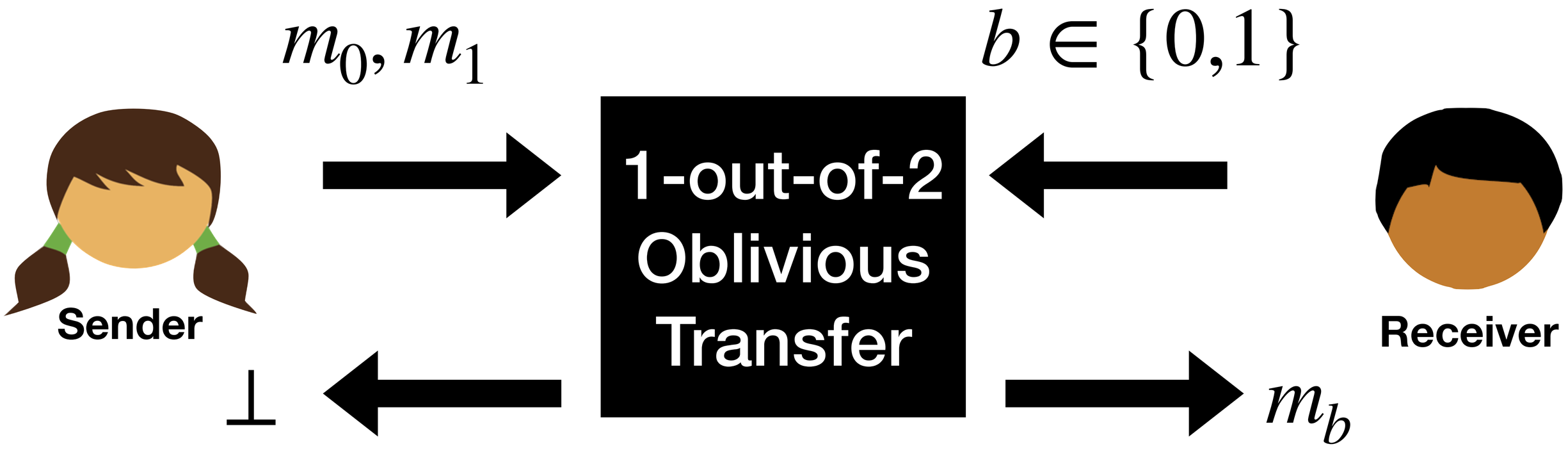
Sender

1-out-of-2 Oblivious Transfer

Receiver

Oblivious Transfer

Oblivious Transfer



Oblivious Transfer from PKE

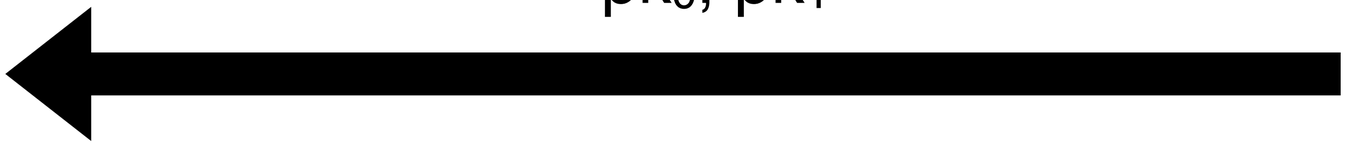
(with ability to sample public key)

Sender

m_0, m_1

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

pk_0, pk_1



$Enc(pk_0, m_0)$ $Enc(pk_1, m_1)$

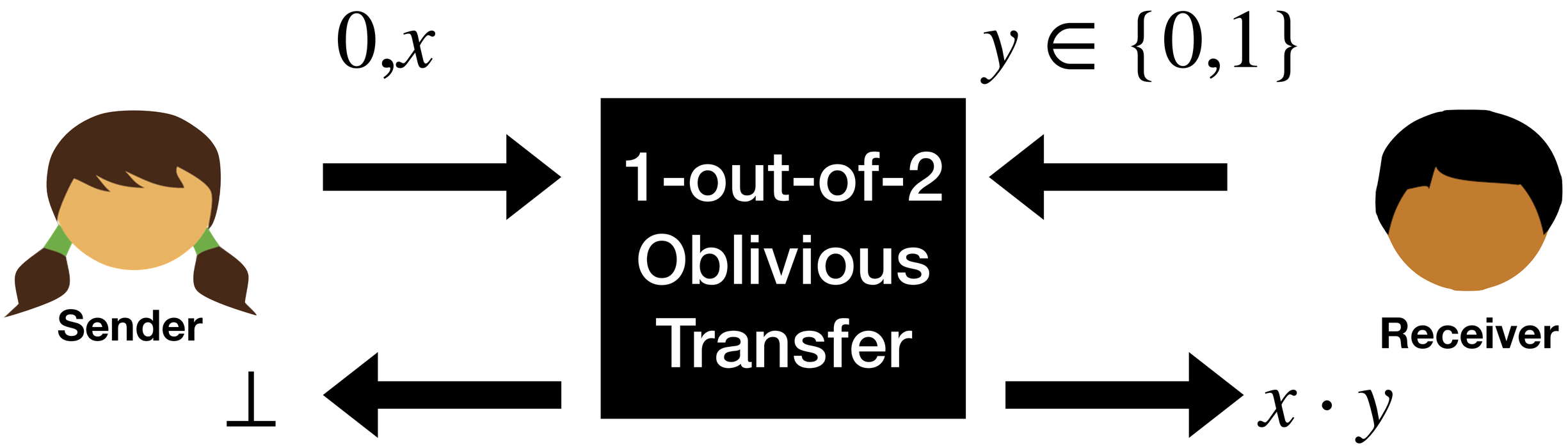
Receiver

$sk_b, pk_b \leftarrow KeyGen()$

$pk_{1-b} \leftarrow\$ PK$

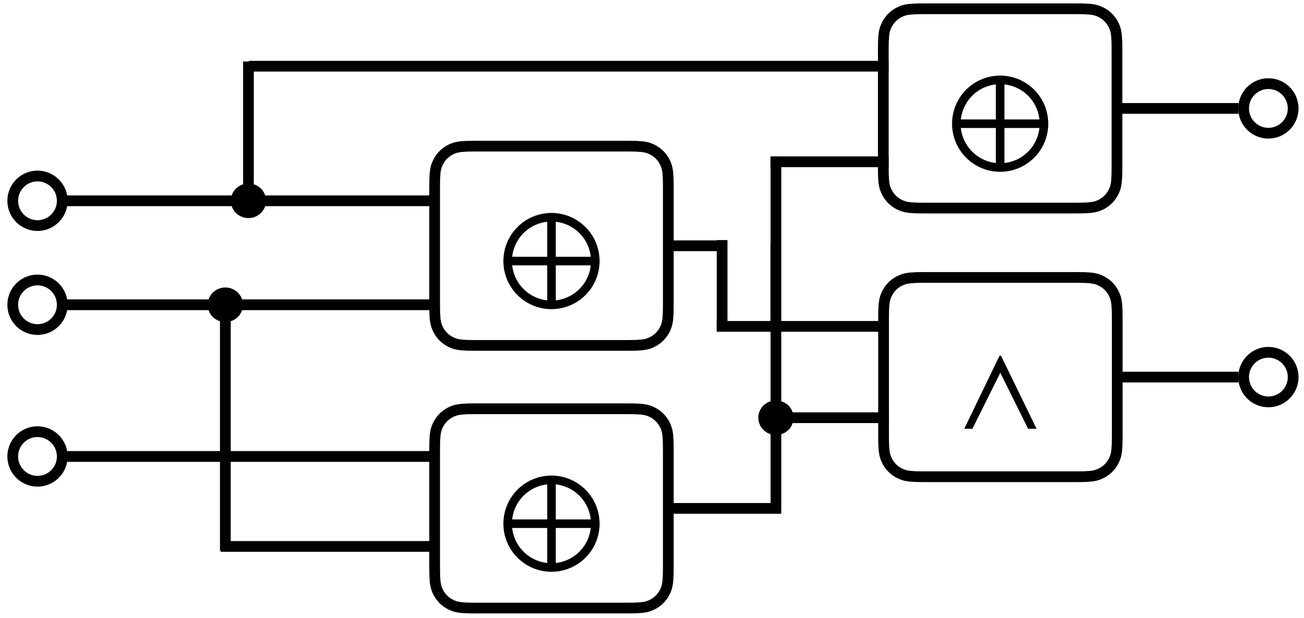
$Dec(sk_b, Enc(pk_b, m_b)) = m_b$

Secure AND


$0, x$
$y \in \{0,1\}$
1-out-of-2 Oblivious Transfer
Sender
Receiver
$\perp$
$x \cdot y$

Sketch of how to run any program P securely

“Compute on secret shares of data”



Today's objectives

Introduce the notion of a secure computation

Define and construct oblivious transfer

Connect secure computation with secret sharing

Construct a general-purpose MPC protocol