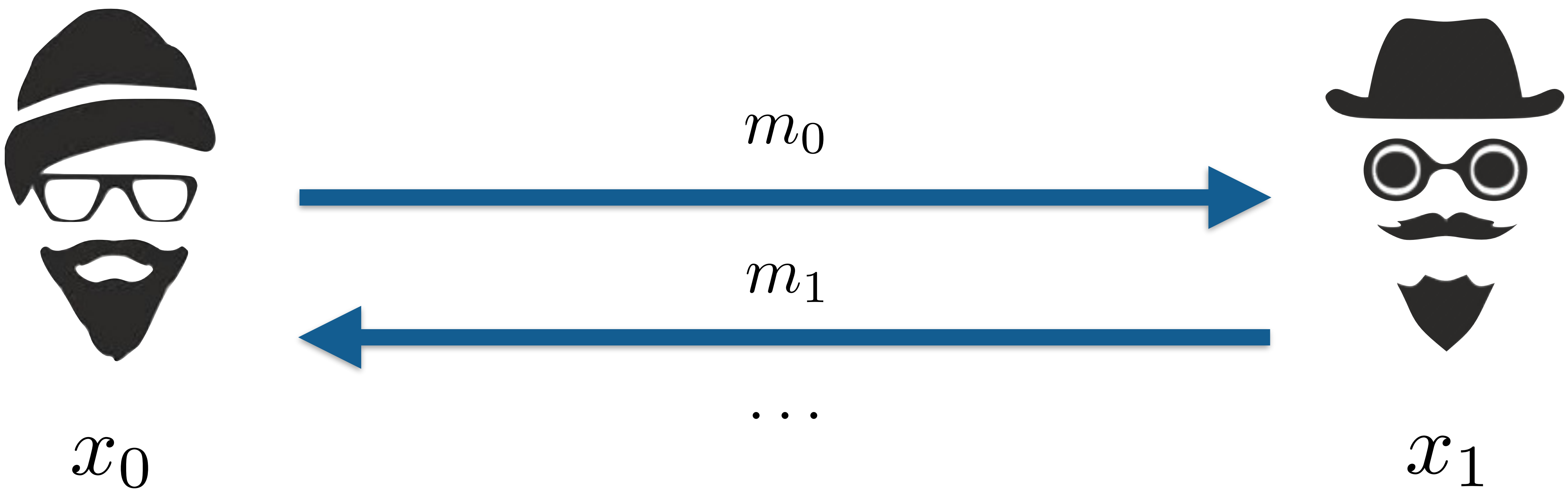
New Protocols for Secure Equality Test and Comparison

Geoffroy Couteau

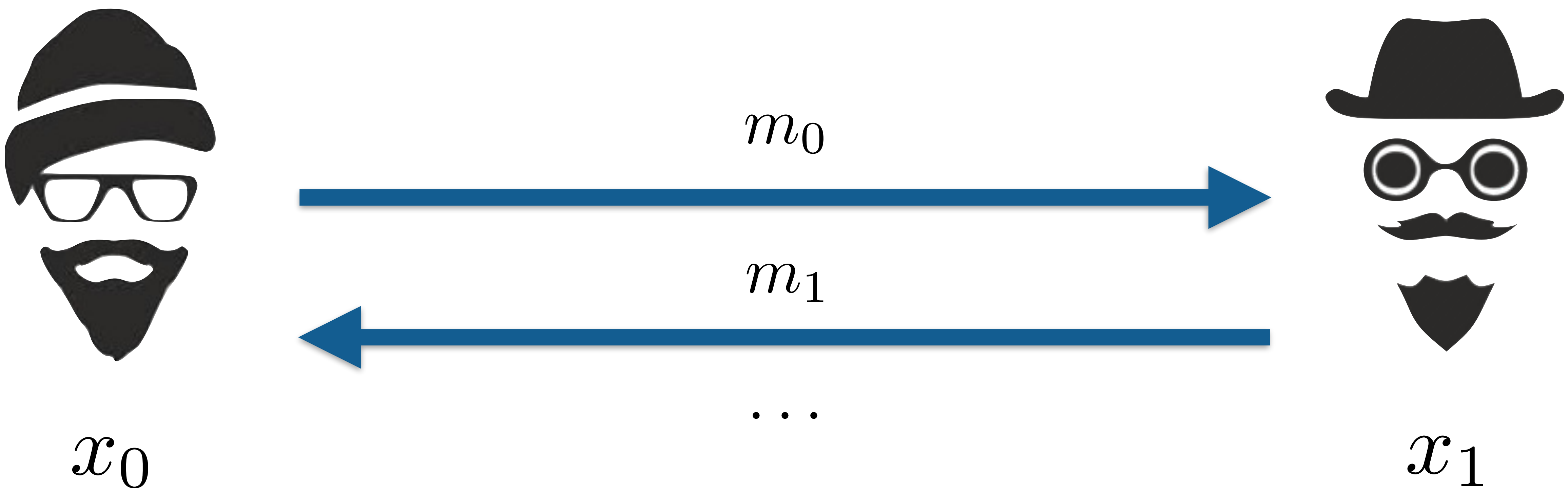
Karlsruher Institut für Technologie

Secure Computation



$y_0 = f_0(x_0, x_1)$ $y_1 = f_1(x_0, x_1)$

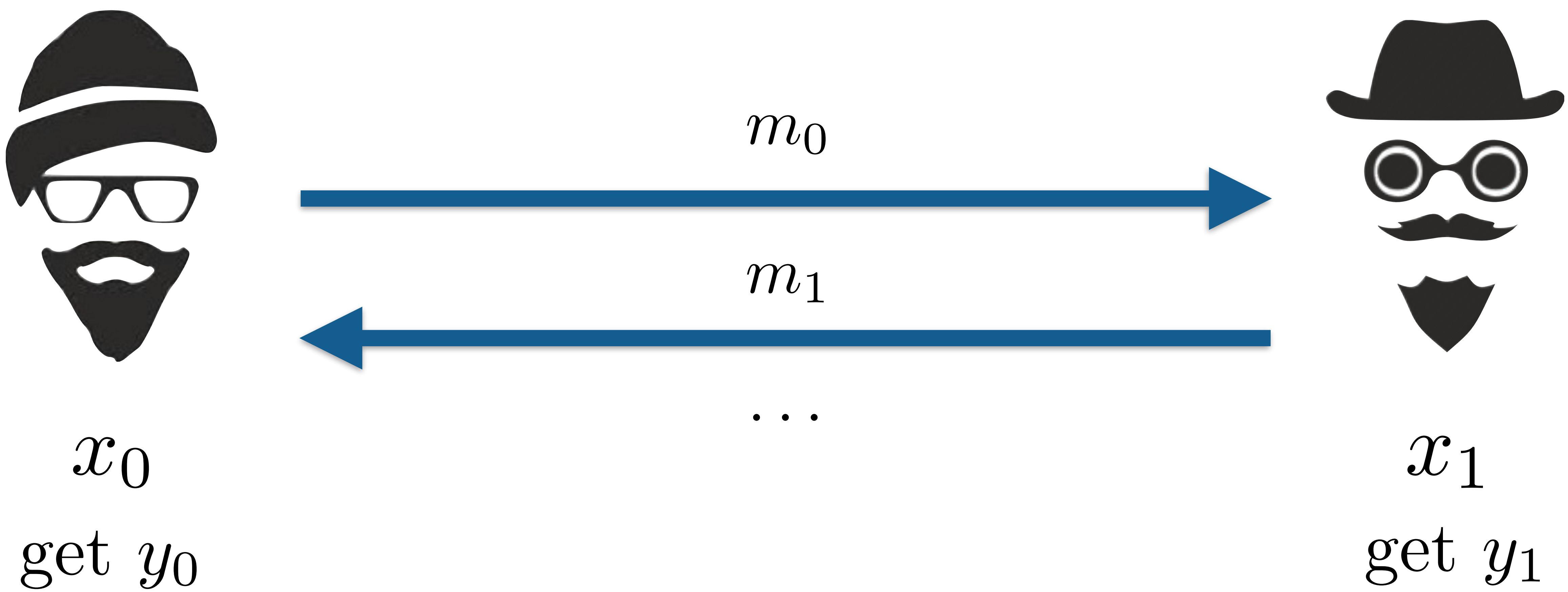
Secure Computation



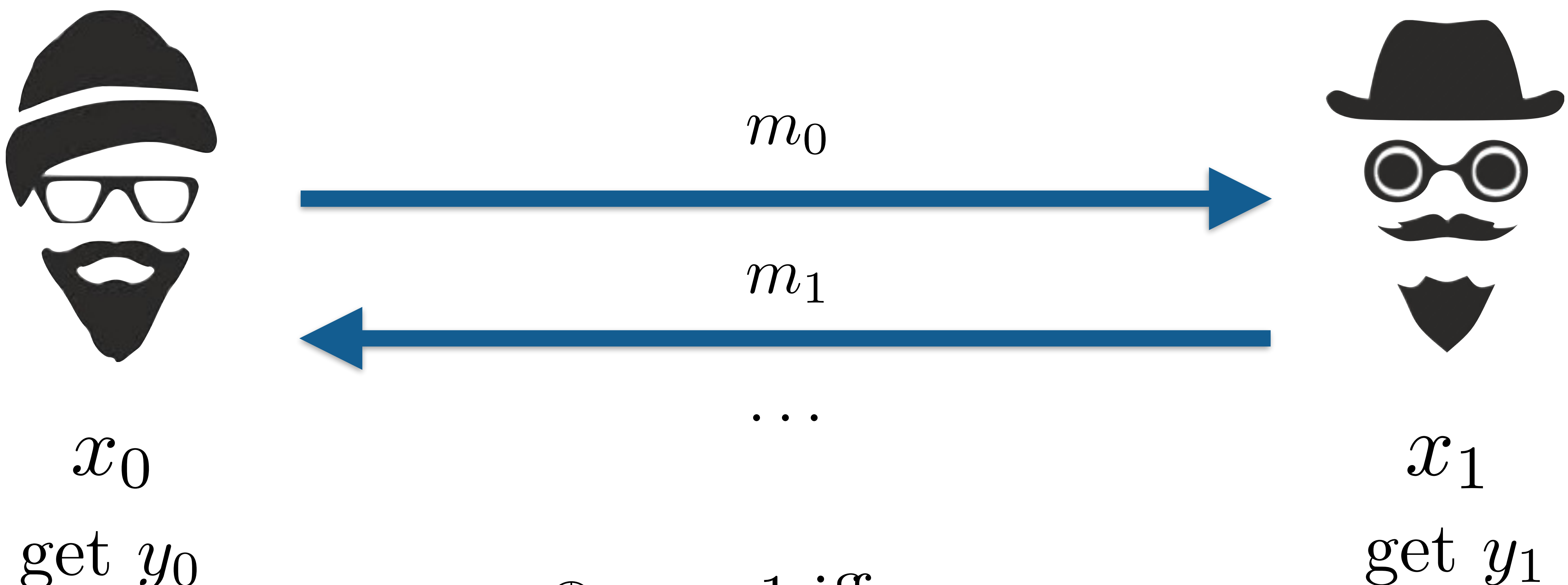
$y_0 = f_0(x_0, x_1)$ $y_1 = f_1(x_0, x_1)$

- Correctness: the parties learn the correct output
- Privacy: the parties learn nothing more than the output

Equality Test & Comparison

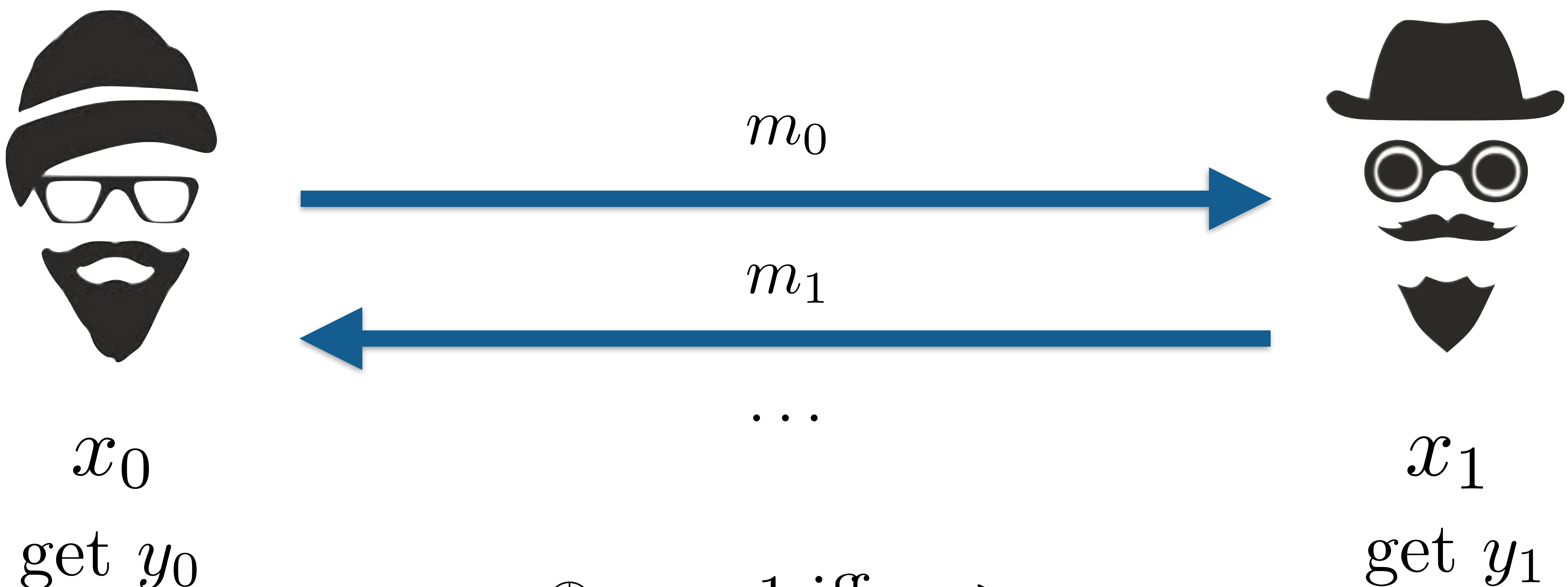


Equality Test & Comparison



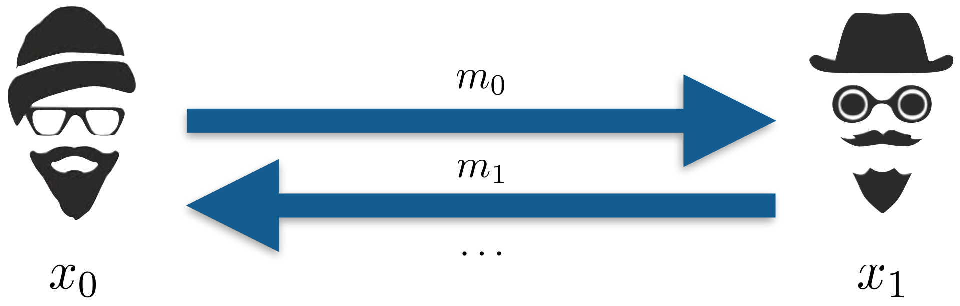
$$y_0 \oplus y_1 = 1 \text{ iff } x_0 = x_1$$

Equality Test & Comparison

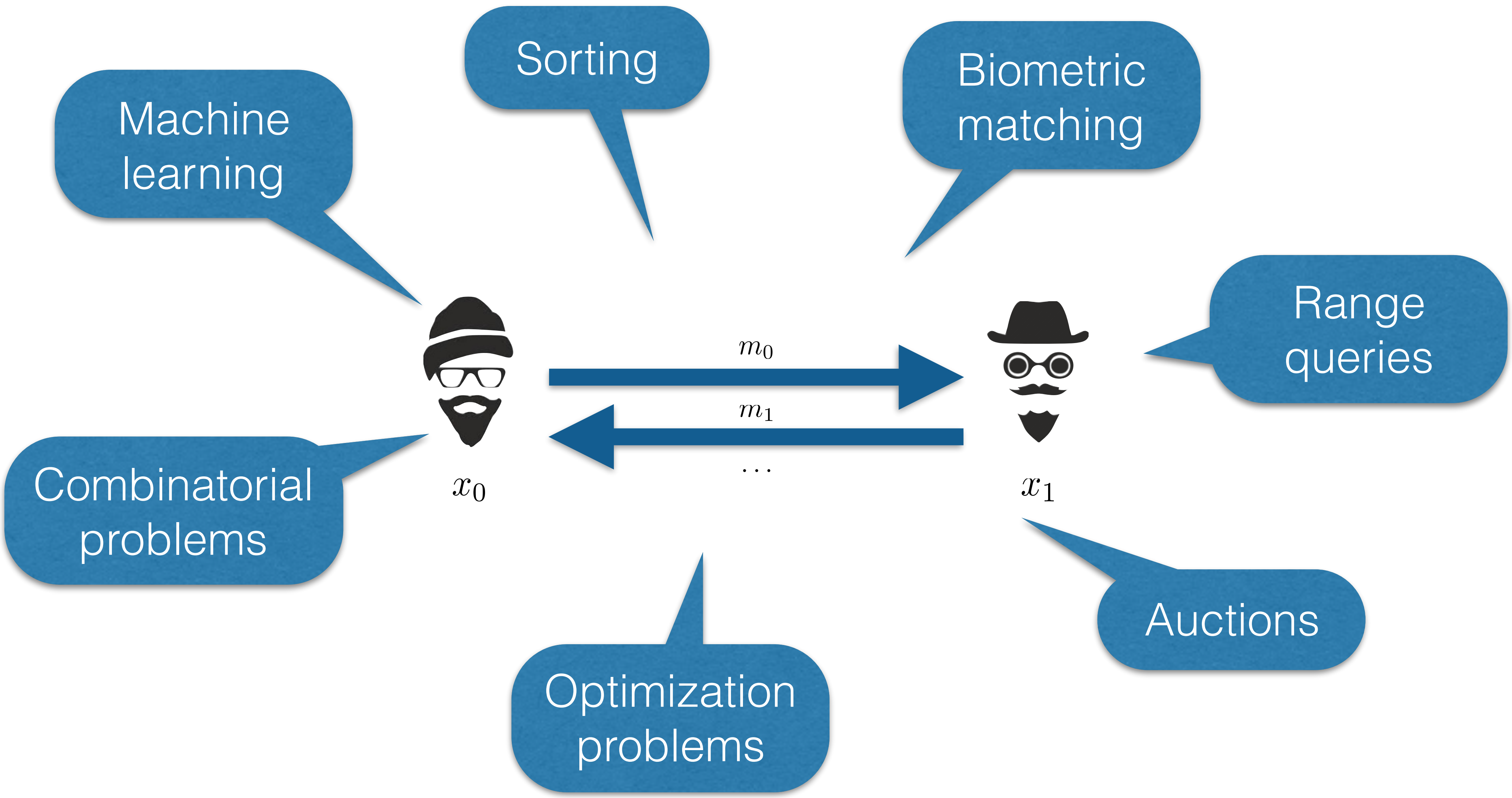


$$y_0 \oplus y_1 = 1 \text{ iff } x_0 > x_1$$

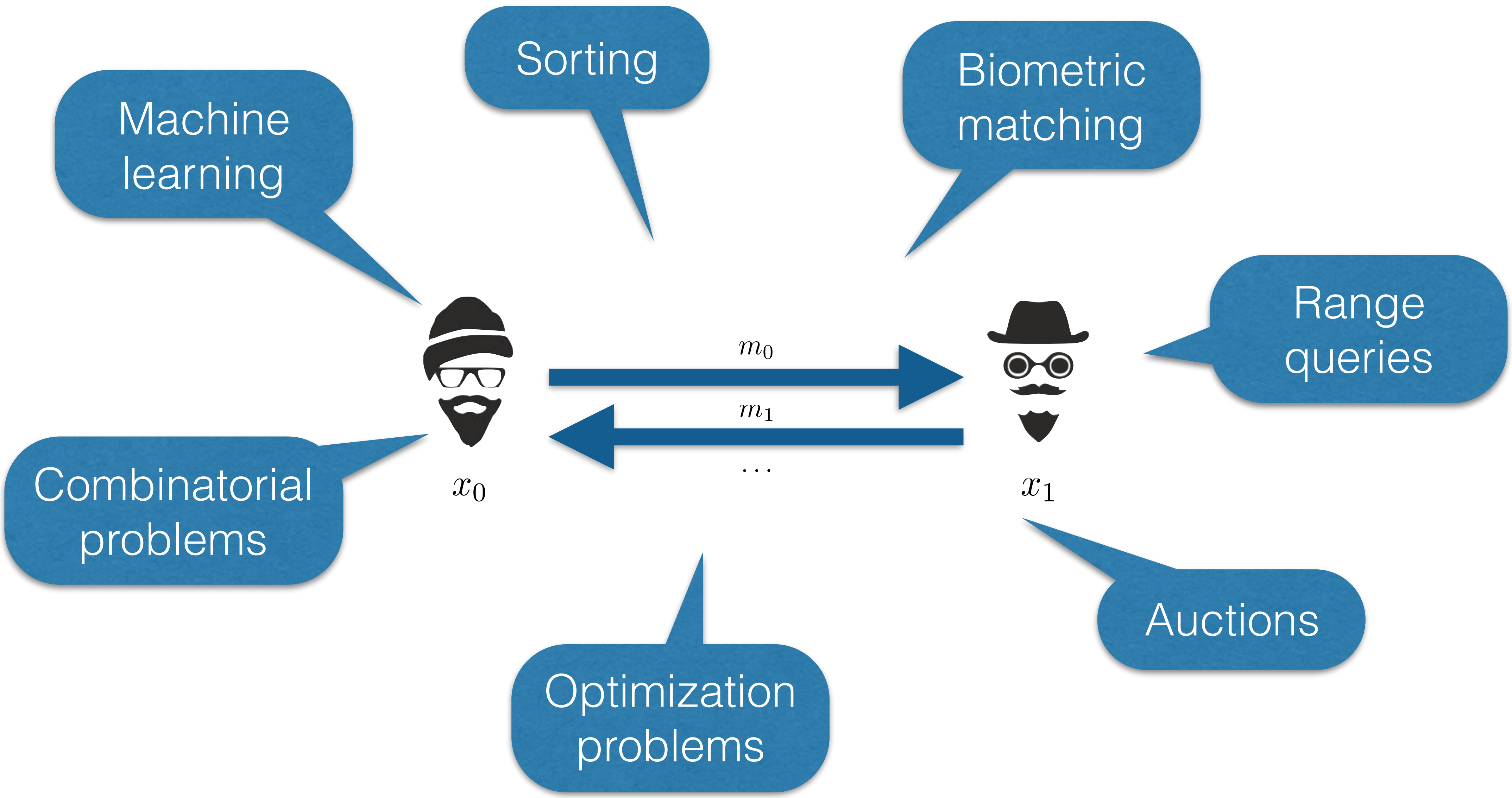
Equality Test & Comparison



Equality Test & Comparison



Equality Test & Comparison



This work: new protocols from OT, with preprocessing

Oblivious Transfer

Oblivious Transfer

Quick facts about OT:

- OT extension makes OT cheap (3 hash/OT)
- OT can be 'packed' for short messages
- OT can be efficiently obtained from random OT

Equality Test

$x =$ | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |

$\oplus$

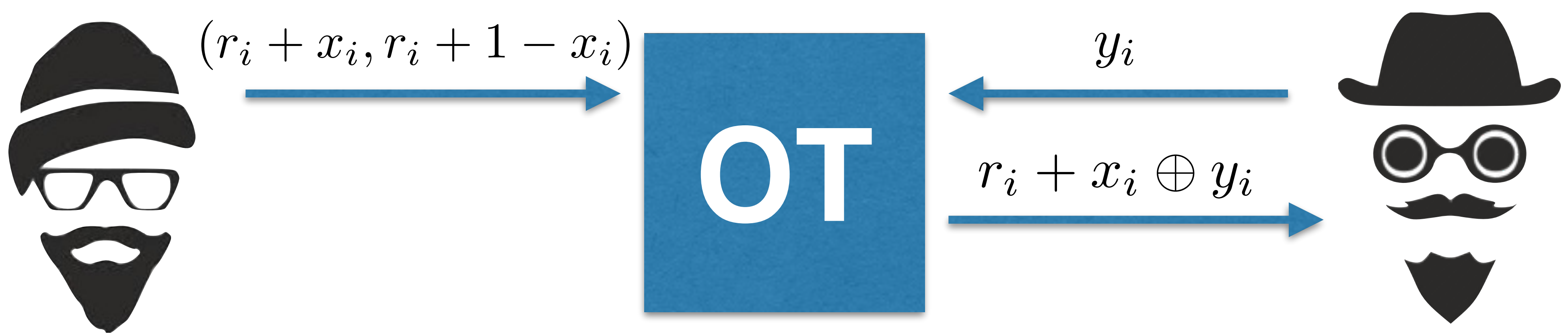
$y =$ | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |

$$(x_i \oplus y_i)_{i \leq \ell}$$

$$(x = y) \iff \sum_{i=1}^{\ell} x_i \oplus y_i = 0 \bmod \ell + 1$$

Equality Test

$$(x = y) \iff \sum_{i=1}^{\ell} x_i \oplus y_i = 0 \bmod \ell + 1$$

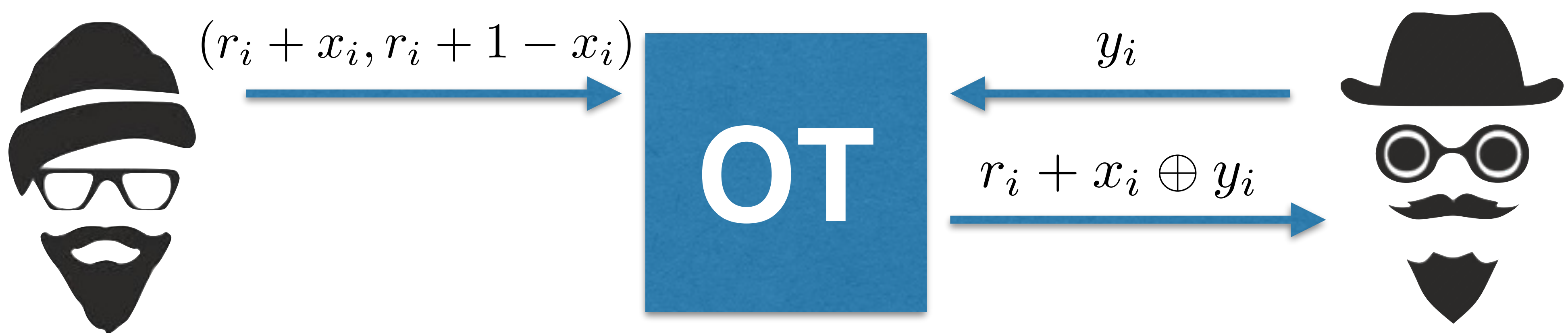


picks $r_i \leftarrow \mathbb{Z}_{\ell+1}$

gets $s_i = r_i + x_i \oplus y_i \bmod \ell + 1$

Equality Test

$$(x = y) \iff \sum_{i=1}^{\ell} x_i \oplus y_i = 0 \bmod \ell + 1$$



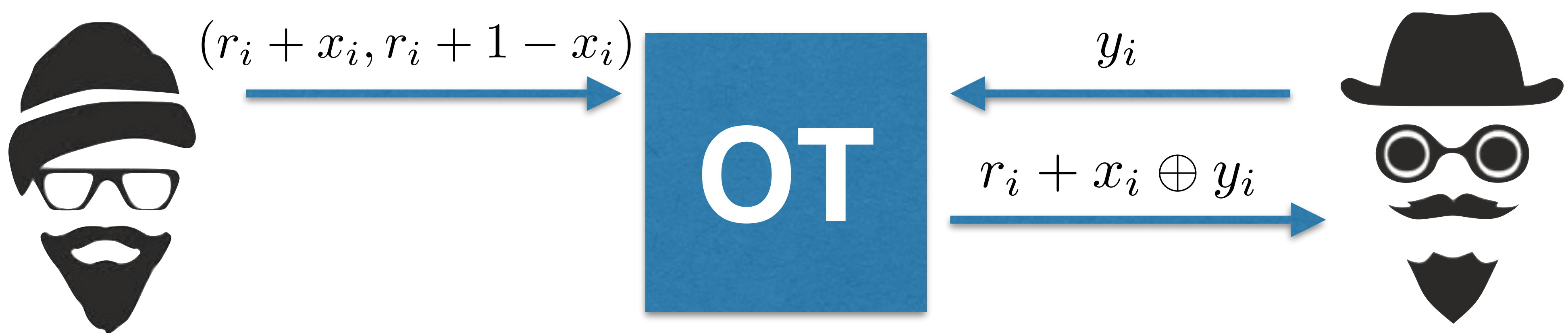
picks $r_i \leftarrow \mathbb{Z}_{\ell+1}$

gets $s_i = r_i + x_i \oplus y_i \bmod \ell + 1$

$$(x = y) \iff \sum_{i=1}^{\ell} r_i = \sum_{i=1}^{\ell} s_i \bmod \ell + 1$$

Equality Test

$$(x = y) \iff \sum_{i=1}^{\ell} x_i \oplus y_i = 0 \bmod \ell + 1$$



picks $r_i \leftarrow \mathbb{Z}_{\ell+1}$

gets $s_i = r_i + x_i \oplus y_i \bmod \ell + 1$

$$(x = y) \iff \sum_{i=1}^{\ell} r_i = \sum_{i=1}^{\ell} s_i \bmod \ell + 1$$

---

sets $x' \leftarrow \sum_{i=1}^{\ell} r_i$

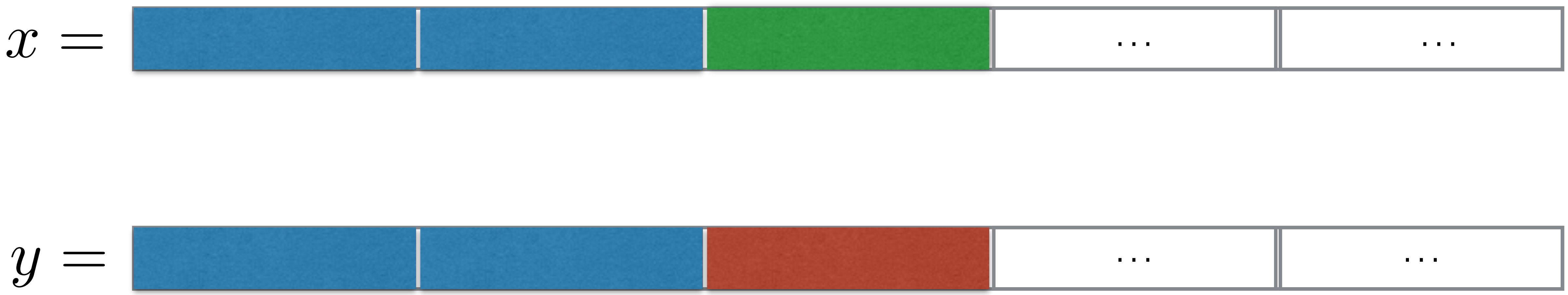
sets $y' \leftarrow \sum_{i=1}^{\ell} s_i$

Equality Test

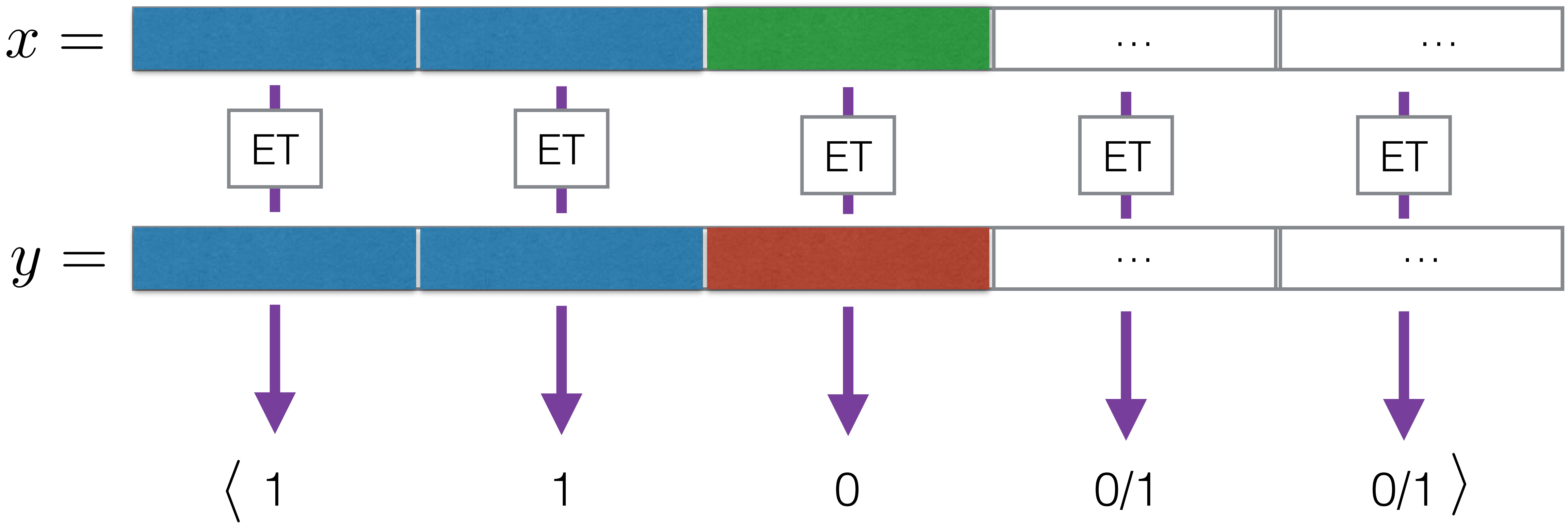
- Number of rounds: $log^* \kappa$
- Uses only small-string OT
- Can be efficiently preprocessed:

  - run the protocol on random inputs (r_0, s_0)
  - store the intermediate values (r_i, s_i)
  - exchange the $(x_i \oplus r_i, y_i \oplus s_i)_i$ and use the OT to ROT reduction

Comparison

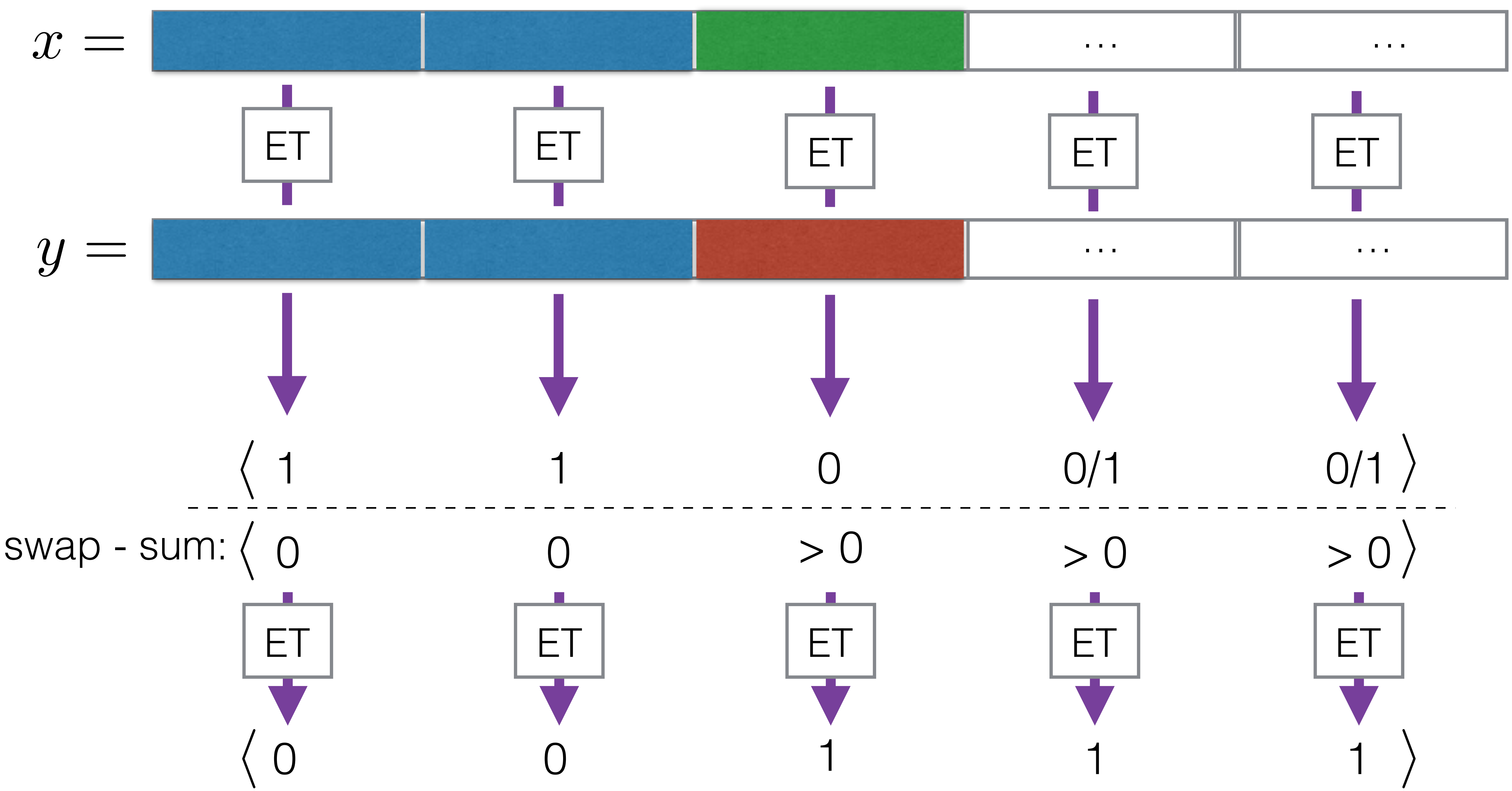


Comparison

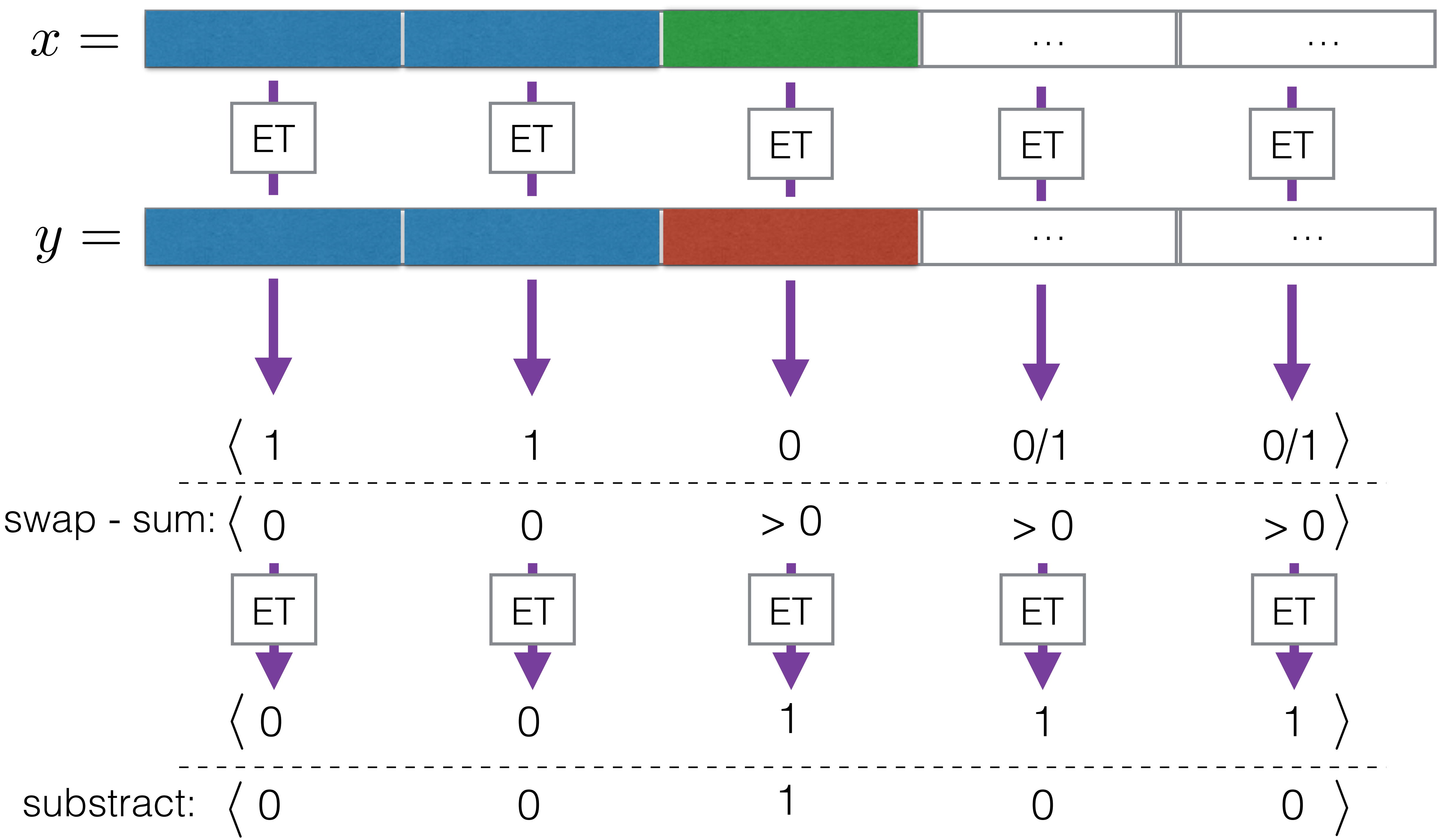


$x =$
$\cdots$
$\cdots$
ET
ET
ET
ET
ET
$y =$
$\cdots$
$\cdots$
$\langle$ 1
1
0
0/1
0/1 $\rangle$

Comparison

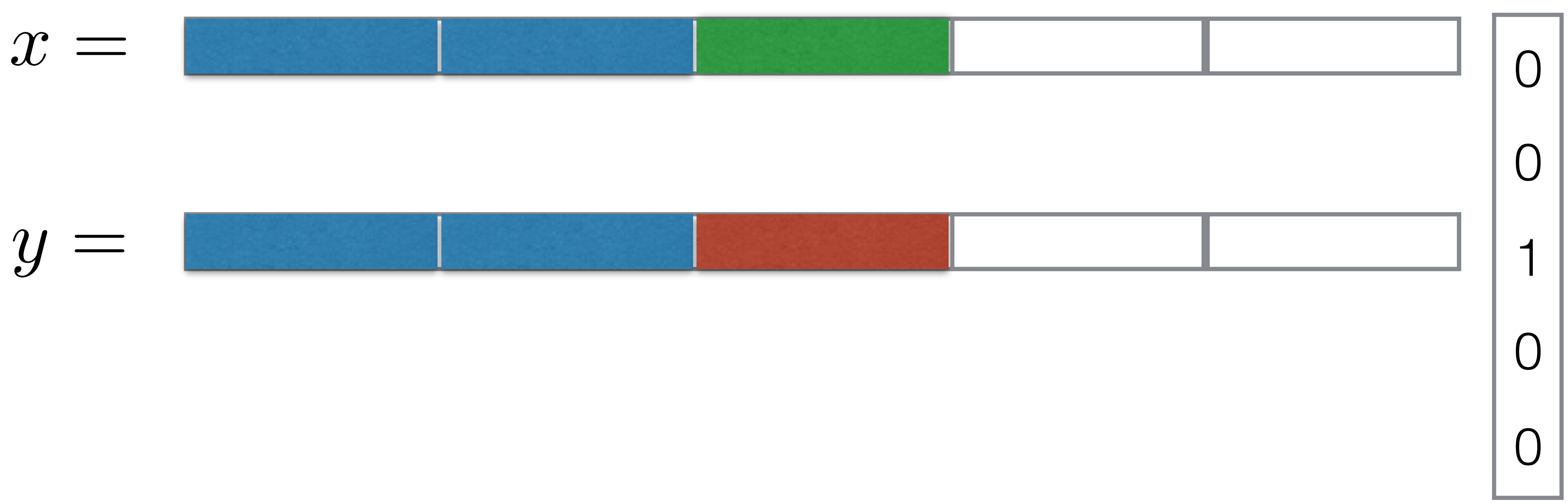


Comparison


$x =$
$\cdots$
$\cdots$
ET
ET
ET
ET
ET
$y =$
$\cdots$
$\cdots$
$\langle$ 1 1 0 0/1 0/1 $\rangle$
swap - sum: $\langle$ 0 0 > 0 > 0 > 0 $\rangle$
ET
ET
ET
ET
ET
$\langle$ 0 0 1 1 1 $\rangle$
substract: $\langle$ 0 0 1 0 0 $\rangle$

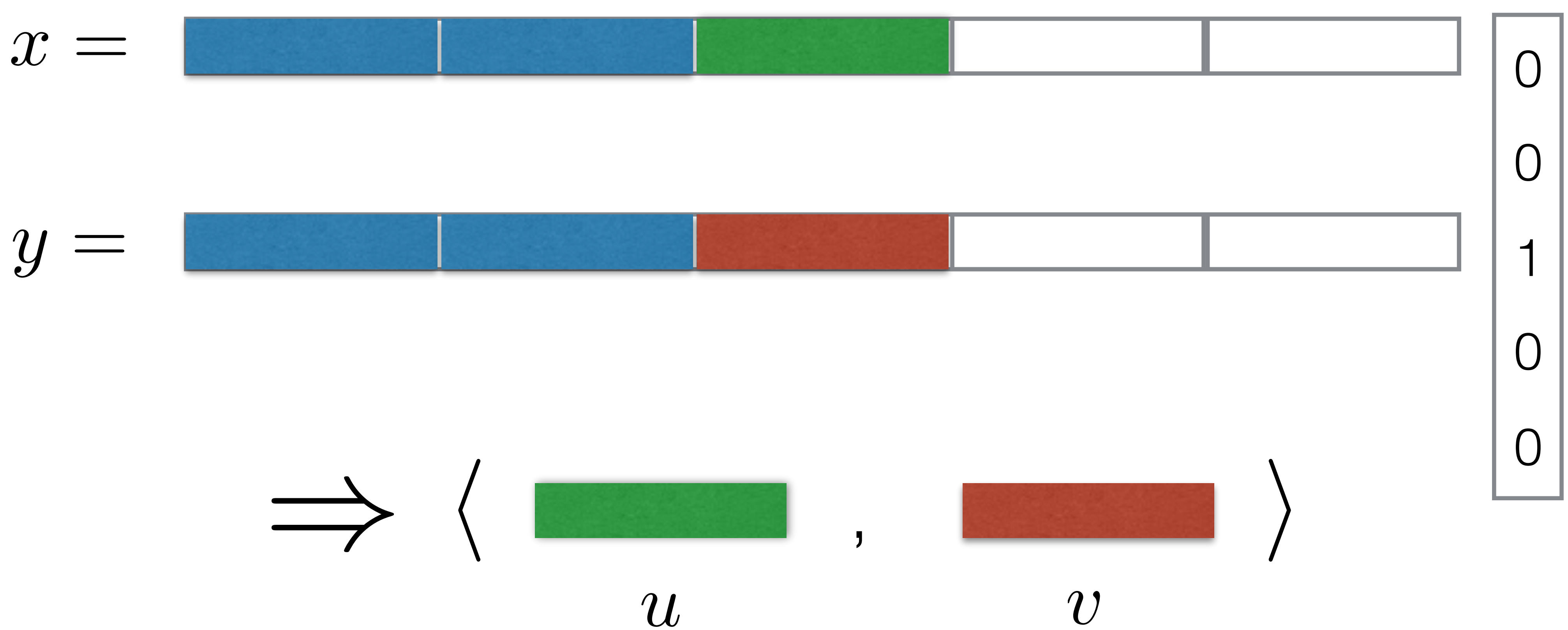
Comparison

Inner product:



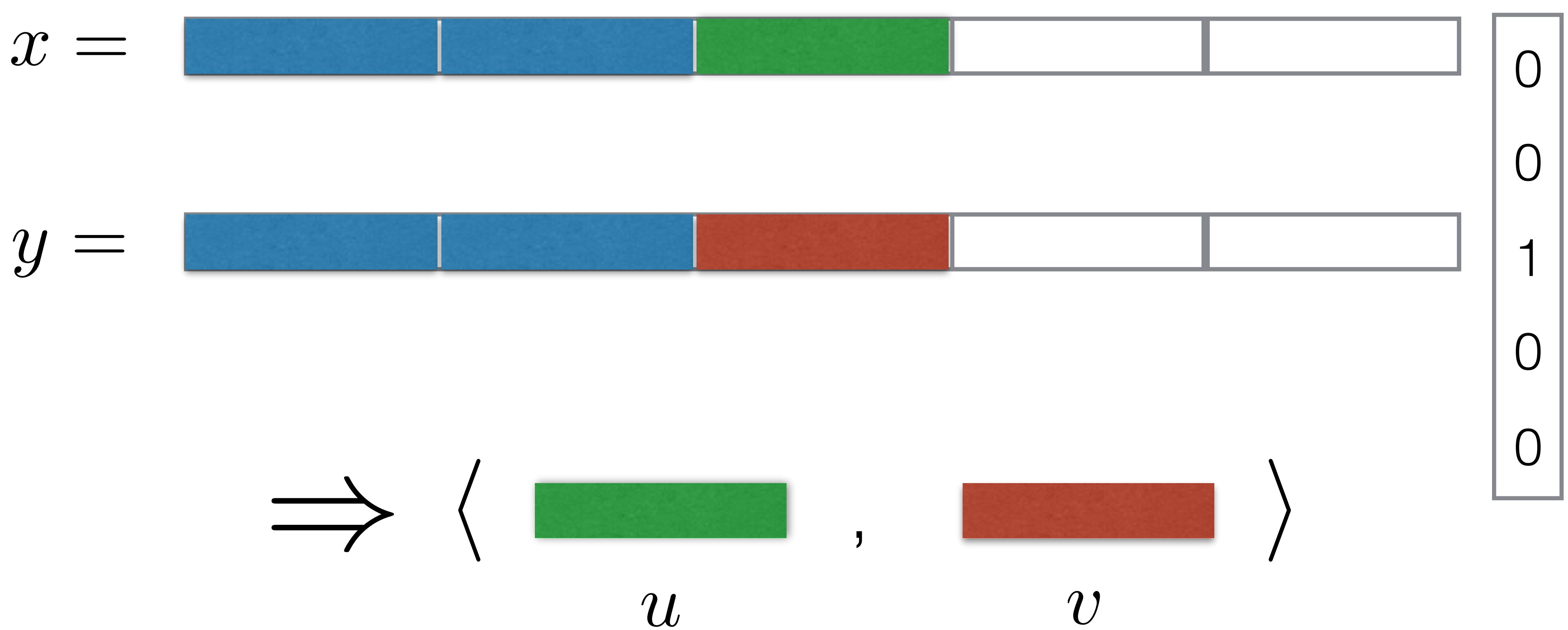
Comparison

Inner product:



Comparison

Inner product:



Lemma: assuming shares over $\mathbb{Z}_t$, $u \neq v$, and $|u|, |v| \leq t/2$, Alice and Bob can locally compute respective values x', y' such that $u \leq v$ iff $x' \leq y'$.

Comparison

- The full protocol has $O(\log\log\ell)$ rounds
- It can be interfaced with other existing protocols
- The communication is asymptotically optimal, $O(\ell)$
- The online phase is extremely efficient

Thank you for your attention

Questions?