Special Topics in Cryptography

Mohammad Mahmoody

Logistics

- Most submitted PS3. If you have not you will get delay, but email it to me ASAP.
- Deadline for project reports/drafts + slides : This Thursday 5pm. There will be a collab post for it.
- I will announce the order of presentations. So your talk could be on any of the remaining days.
- You are all anticipated to participate in each others’ presentations.

Last time

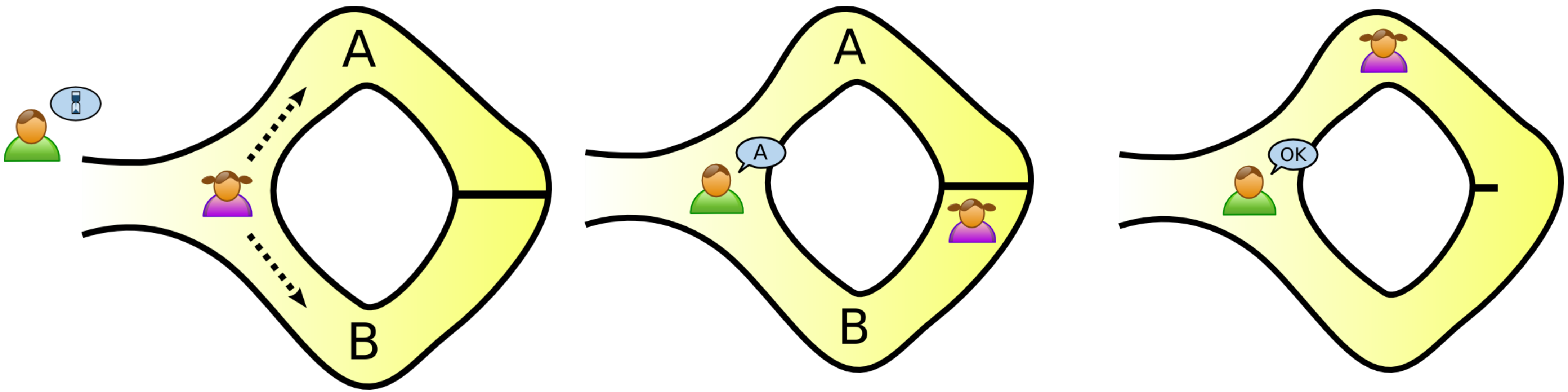
- Zero Knowledge Proofs

Today

- Secure computation

Can we ever prove we know something without revealing the details of the secret?

- Alice knows a magic word to open the door inside the cave:



NP vs P — poly-time solvable

Formal Definition of Zero Knowledge Proofs

- Suppose $L \in \mathbf{NP}$ meaning, there is poly-time verifier $V(\cdot,\cdot)$ such that $x \in L \Leftrightarrow \exists\, w, V(x,w) = 1$

$L = \{ G \mid G \text{ is 3-colorable} \}$

$V(G, C) :=$ go over all $i, j \in G$ where $i \sim j$ make sure $c(i) \neq c(j)$

C: mapping from $V(G) \to \{1,2,3\}$

- Examples: $L' = \{ G \mid G \text{ is Hamiltonian} \}$

$V(G, \pi)$: make sure $i_j \sim i_{j+1} \pmod n$

witness $(i_1, i_2, \ldots, i_n)$

0: if Prover & Verifier honest $\Pr[\text{verifier accept}] \approx 1$

- An "interactive" protocol between a "prover" P and a "verifier" V:
  1. ($x \notin L$) Is **sound** if: for all even malicious prover P^* : $\Pr[V(x) = 1] \leq \text{negl}(n)$
  2. ($x \in L$) Is zero-knowledge if: for all even malicious verifier V^* $\exists\, S$ such that $S(x) \approx \text{view}(V)$ in interaction with P on input x

Poly-time. poly-time Simulator

Suppose we deal with lang $L: \{N \mid N = p \cdot q\}$

$V(N, w) =$ tel & make sure $N = p \cdot q$

w = (p, q)

GMW
- Goldreich
- Micali
- Wigderson

$\exists$ algorithm $T_L(N) \to G$ such that $N \in L$ iff G is 3-Colorable

& $S_L(w) \to C$ such that if N is $p \cdot q \to C$ is a 3-Coloring for G.

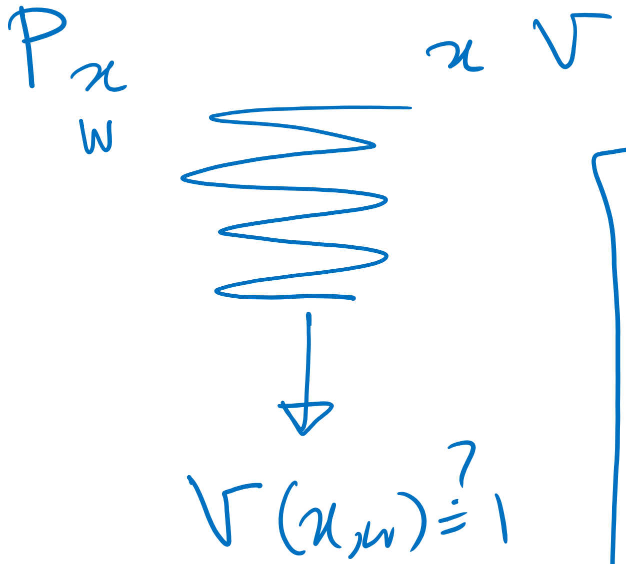
w = $p \cdot q$

Prove — claim $N \in L$ — Ver

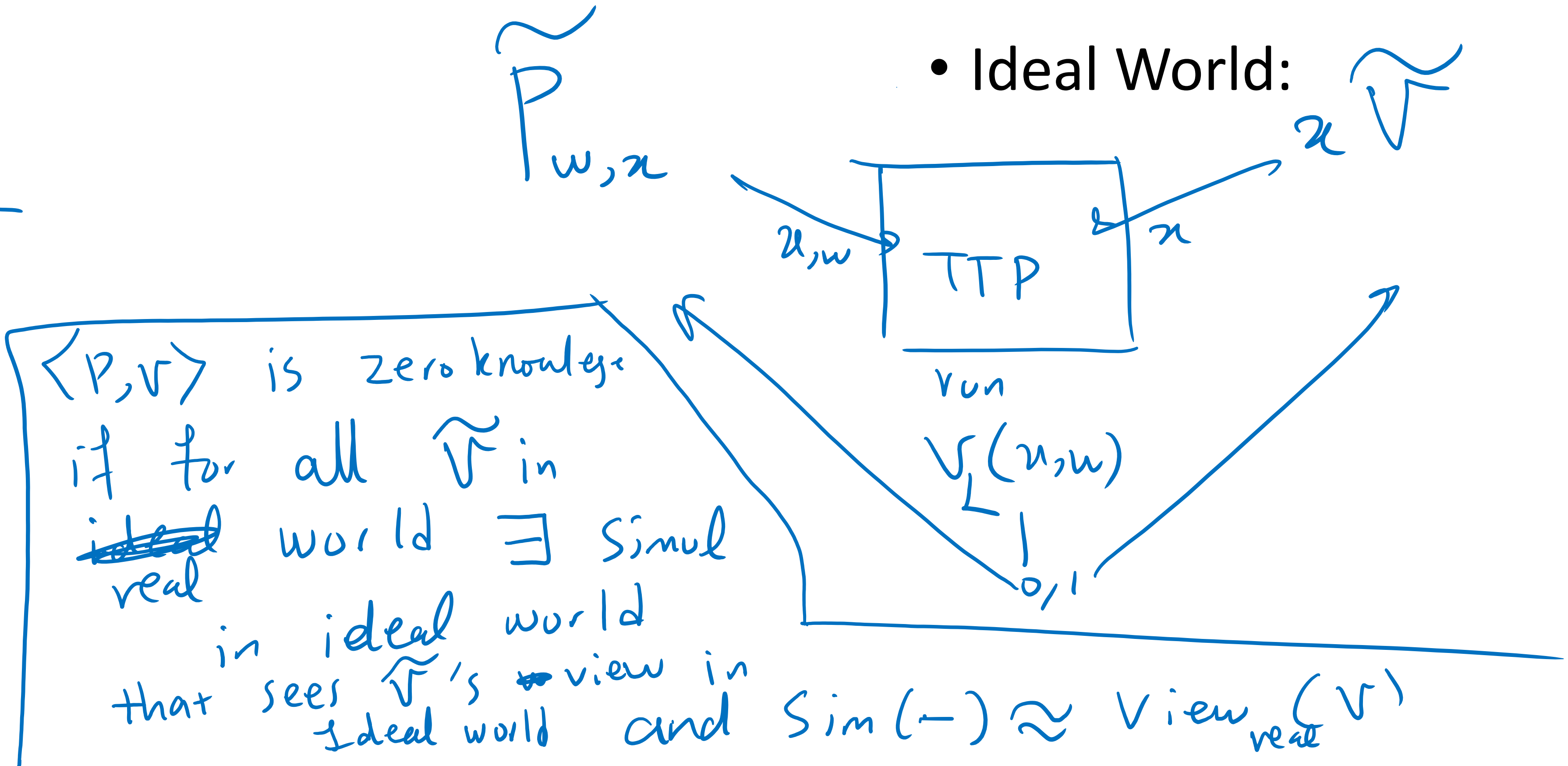
instead Prove (in ZK) $T_L(N) = G \in$ 3-Colorable

Another way to see these two properties

- Using a “trusted third party”.
  - Real World:



  - Ideal World:



Secure Multiparty Computation

Yao's Billionaires Problem: Who has more money?

x

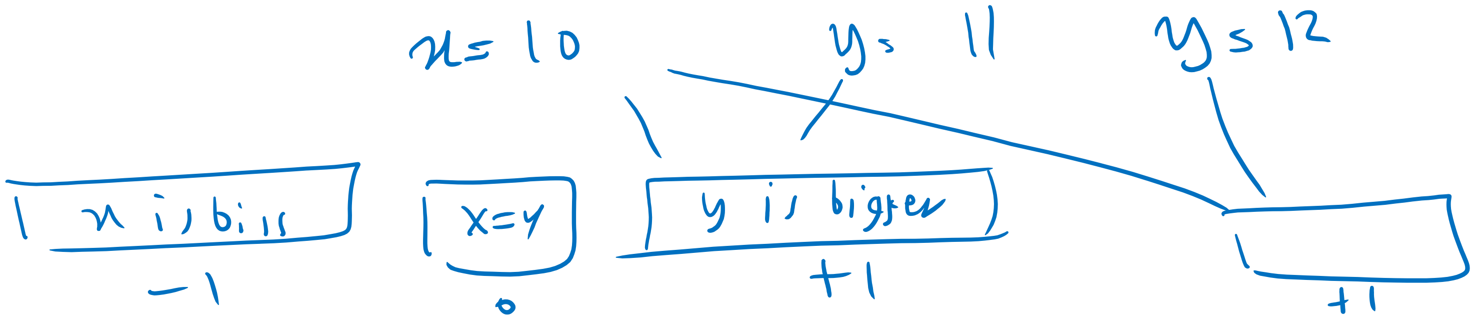
$x > y$

$y > x$

$x = y$

y

In General



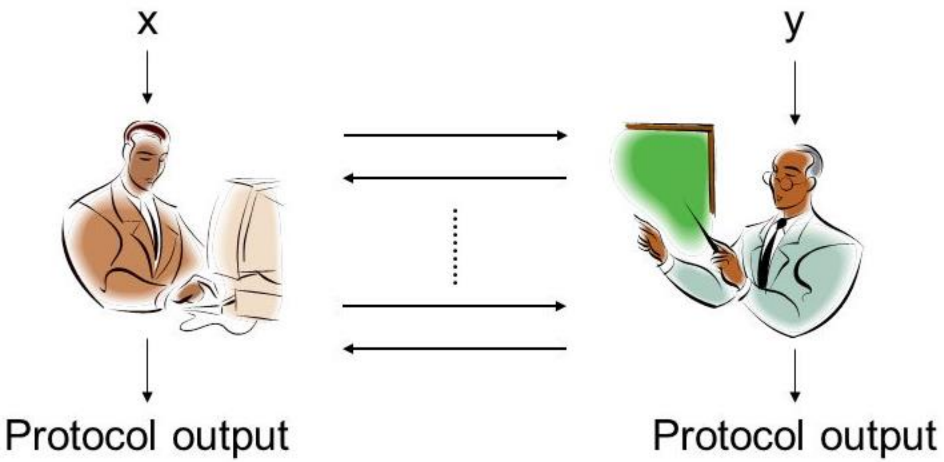
- Parties $P_1, \dots, P_m$ want to compute $f(x_1, \dots, x_m)$ “securely” :
- Party P_i has input x_i an would learn $f(x_1, \dots, x_m)$
- Nobody should learn beyond what they would from the output.

- Security Models:

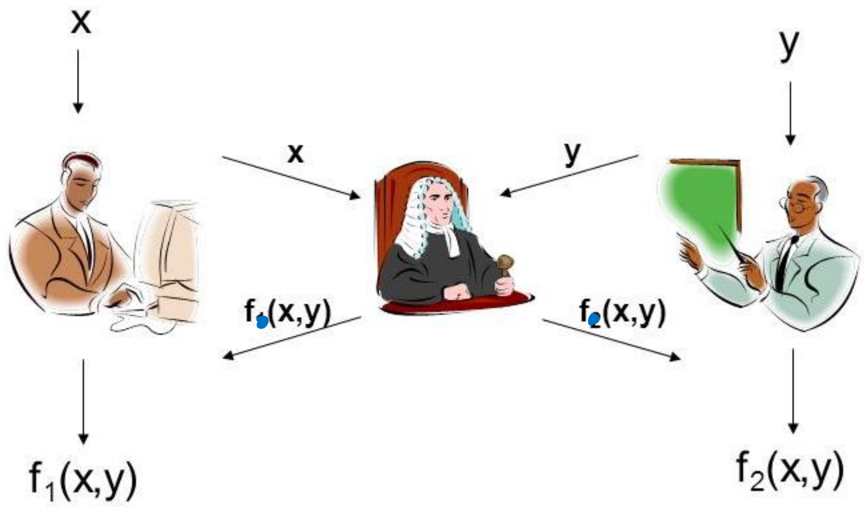
1. **Semi-honest** (aka honest-but-curious) : cheating party follows the protocol, but at the end tries to extract information.
2. **Malicious**: cheating party might deviate from protocol completely.

How to define security in general?

- Real Model:



- Ideal Model:



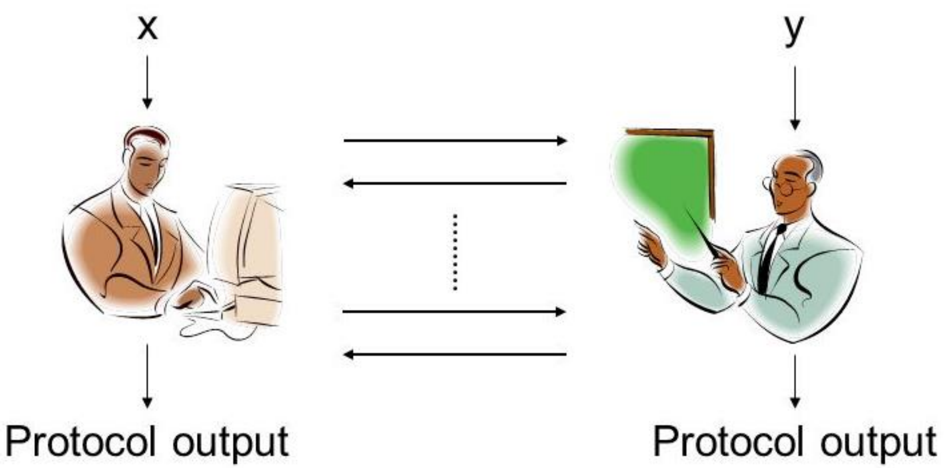
$f(x,y)$

Security for P_1: $\exists$ $Sim_2(\cdot)$ that takes the view of P_2 in Ideal world and simulates view P_2 in Real

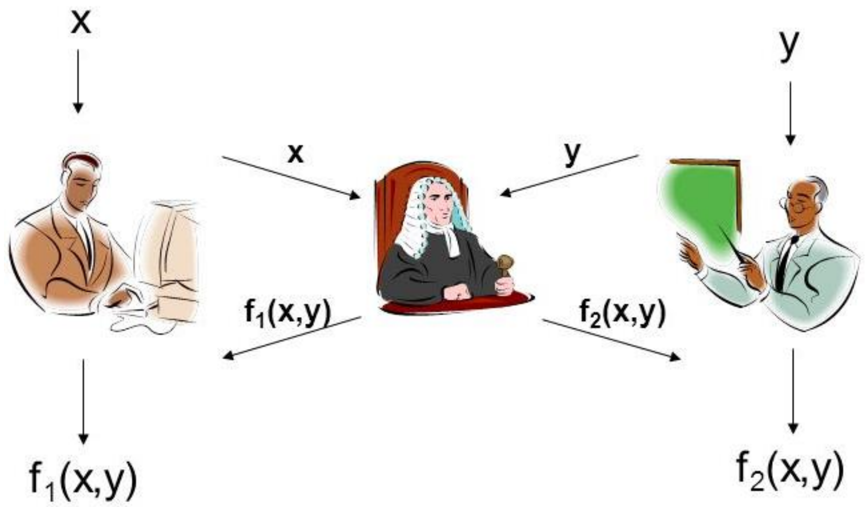
Security for P_2 . . .

How about fully malicious attackers (who might change their inputs)?

- Real Model:



- Ideal Model:



- Ideal model does not allow changing the inputs after they are 'sent'

Oblivious Transfer: a "complete" functionality

$f_1(x, b) = \bot$

interested in

$x_b = f_2(x, b)$

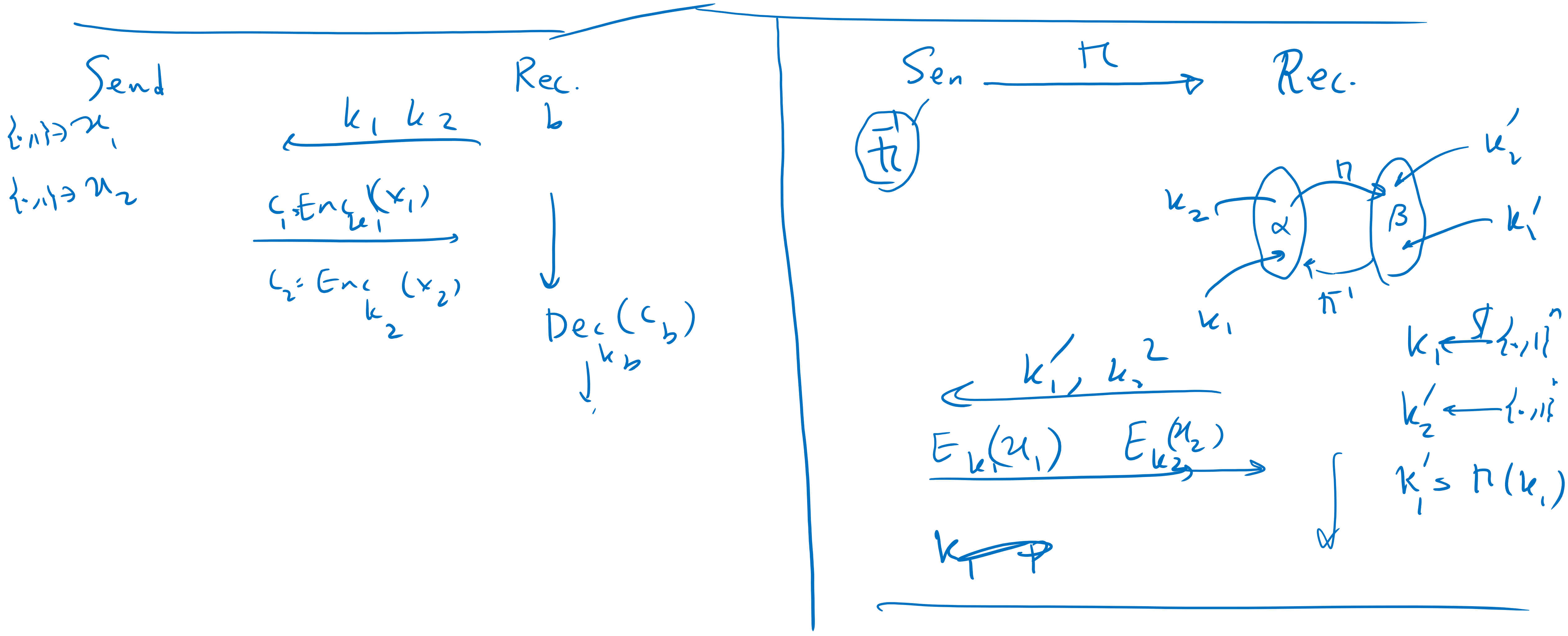
Security for Alice:

Bob is reading only one of x_1, x_2

Security for Bob:

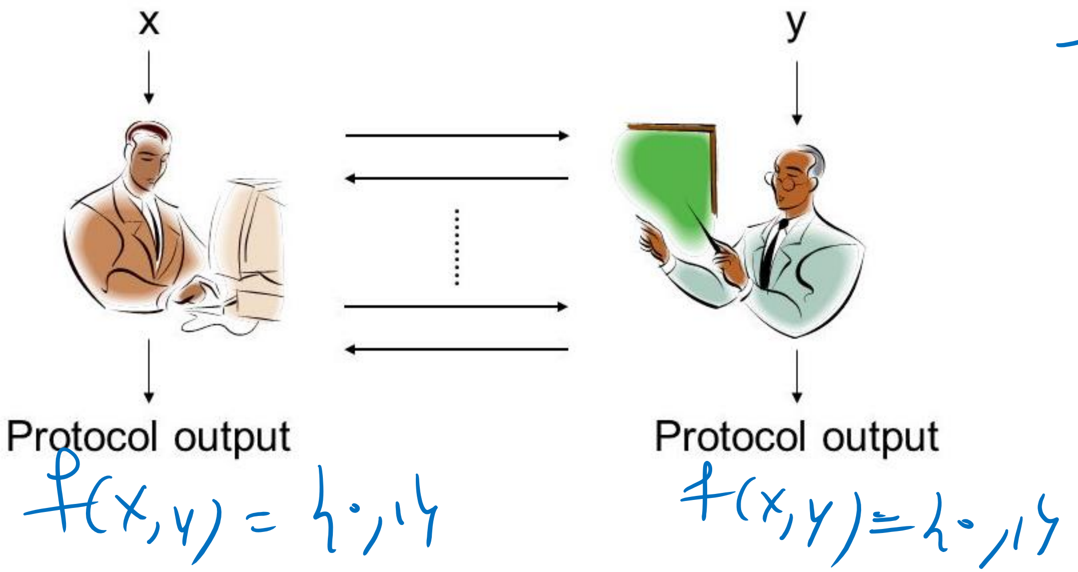
Alice does NOT know $\underline{b}$

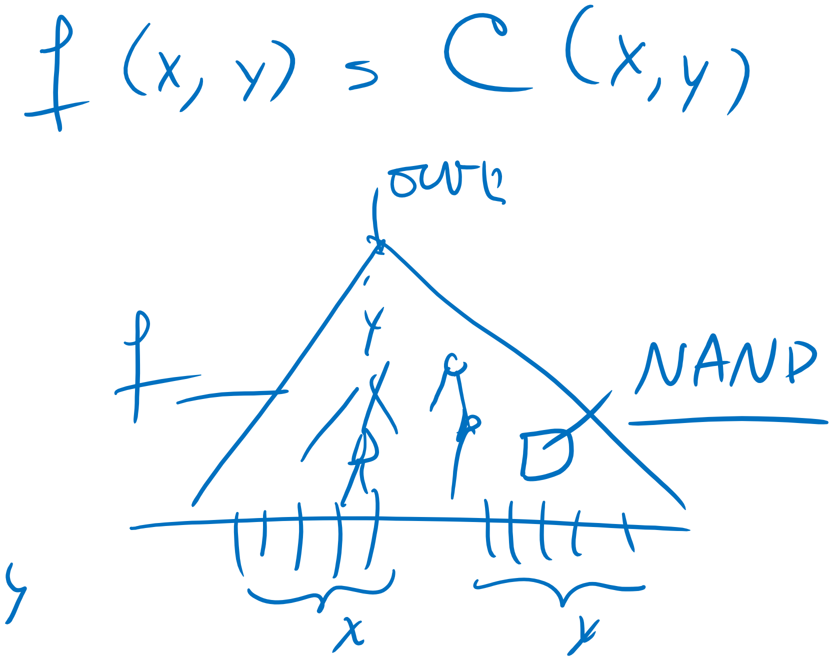
Semi-Honest OT from Trapdoor Permutations



Using OT to get 2 party secure computation

Recall: Secure Function Evaluation

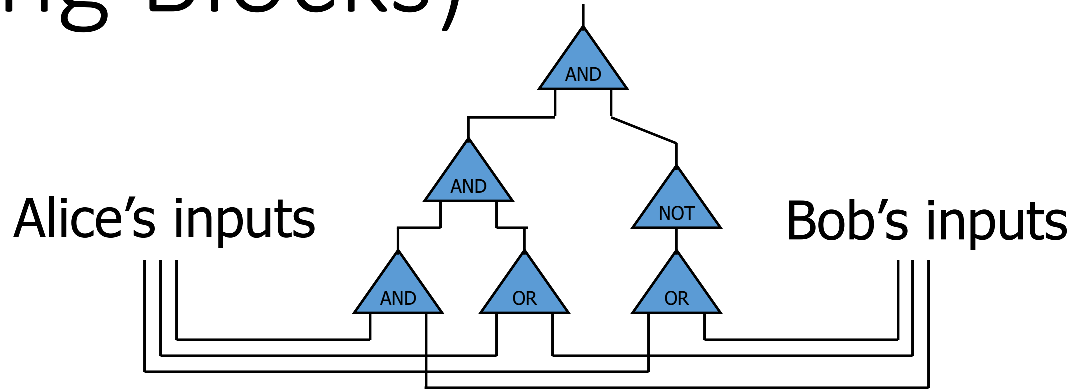


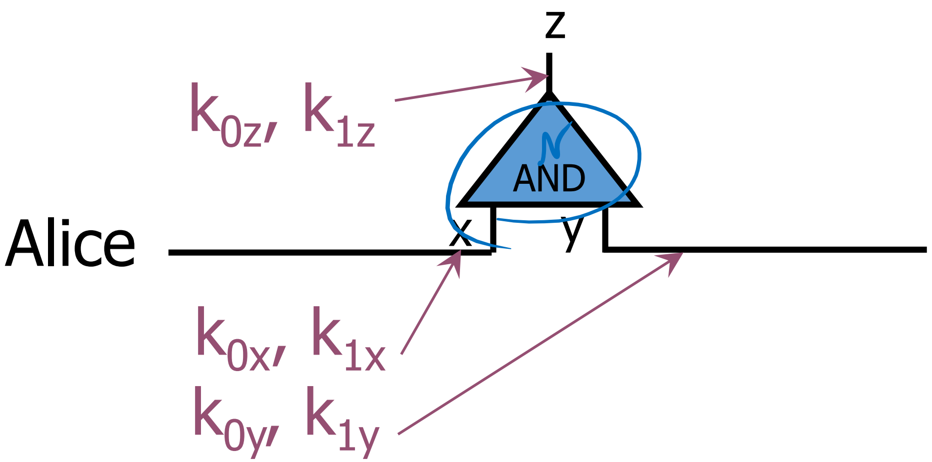


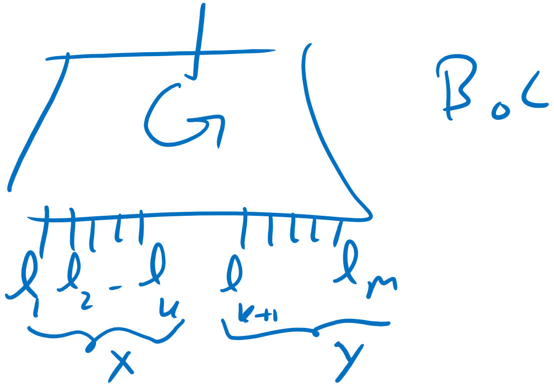
- Protocol's output: f(x,y) where function f is known to both parties.

Yao’s Solution: Garbling of circuits: (Using OT and SKE as building Blocks)

1. Alice writes f as a circuit C
2. Convert C into a “garbled” version G where:
   - G “hides” the computation and only reveals the output.
   - Bob can plug in his input only with Alice’s help.







3. Alice sends G (and related keys) to Bob
4. Bob gets right keys for his own inputs **using OT protocol.**
5. Bob “executes” the circuit and sends the answer back.

Garbling truth table of NAND gate

"Truth table"

x	y	z = NAND(x,y)
0	0	1
0	1	1
1	0	1
1	1	0

k_{0x} k_{0y} k_{0z}

k_{1x} k_{1y} k_{1z}

$E_{0x} E_{0y}(k_{1z})$

$E_{0x} E_{1y}(k_{1z})$

$E_{1x} E_{0y}(k_{1z})$

$E_{1x} E_{1y}(k_{0z})$

want: if somebody know:

$(k_{0x}\ k_{1y}) \to k_{1z}$

$E_{1y}(Enc_{0x}(k_{1z}))$

① use some enc that returns $\perp$ on wrong rows.

Yao’s garbled circuit

- The basic form is only semi-honest secure

- Can be made maliciously secure:
- inefficiently: using ZK proofs
- Efficiently: using “cut and choose”