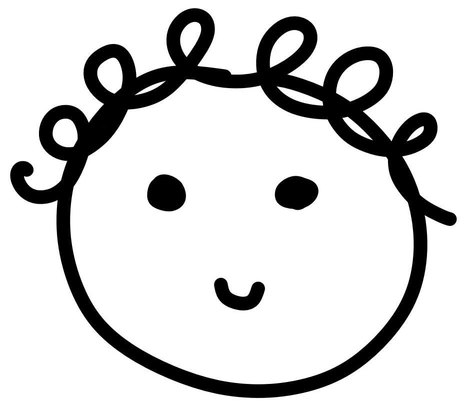
Zero Knowledge Proofs (ZKP) and Secure Multiparty Computation (MPC)

Sophia Yakoubov

Zero Knowledge Proofs (ZKP)
and
Secure Multiparty Computation (MPC)

Sophia Yakoubov

Proof of Sudoku Solvability

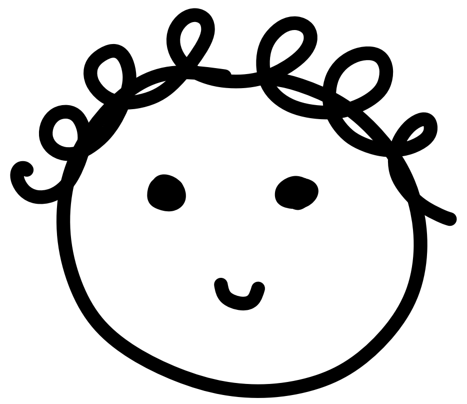


Alice

	7	5		9				6
	2	3		8			4	
8					3			1
5			7		2			
	4		8		6		2	
			9		1			3
9			4					7
	6			7		5	8	
7				1		3	9	

Constraints:
- row contains 1-9
- column contains 1-9
- sub-square contains 1-9

Proof of Sudoku Solvability



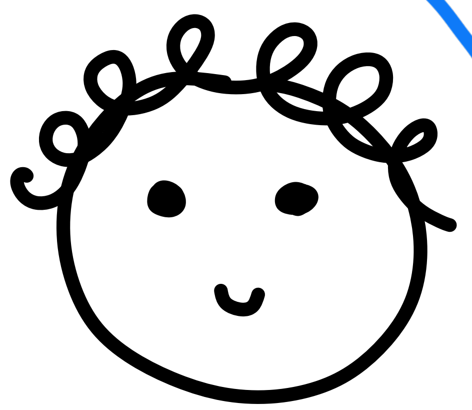
Alice

1	7	5	2	9	4	8	3	6
6	2	3	1	8	7	9	4	5
8	9	4	5	6	3	2	7	1
5	1	9	7	3	2	4	6	8
3	4	7	8	5	6	1	2	9
2	8	6	9	4	1	7	5	3
9	3	8	4	2	5	6	1	7
4	6	1	3	7	9	5	8	2
7	5	2	6	1	8	3	9	4

Constraints:
- row contains 1-9
- column contains 1-9
- sub-square contains 1-9

Proof of Sudoku Solvability

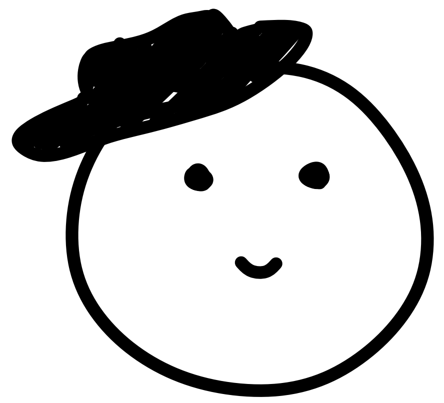
Do I just...
give the answer away?
No! ZKP!



Alice

Can you solve
this sudoku?

	7	5		9				6
	2	3		8			4	
8					3			1
5			7		2			
	4		8		6		2	
			9		1			3
9			4					7
	6			7		5	8	
7				1		3	9	

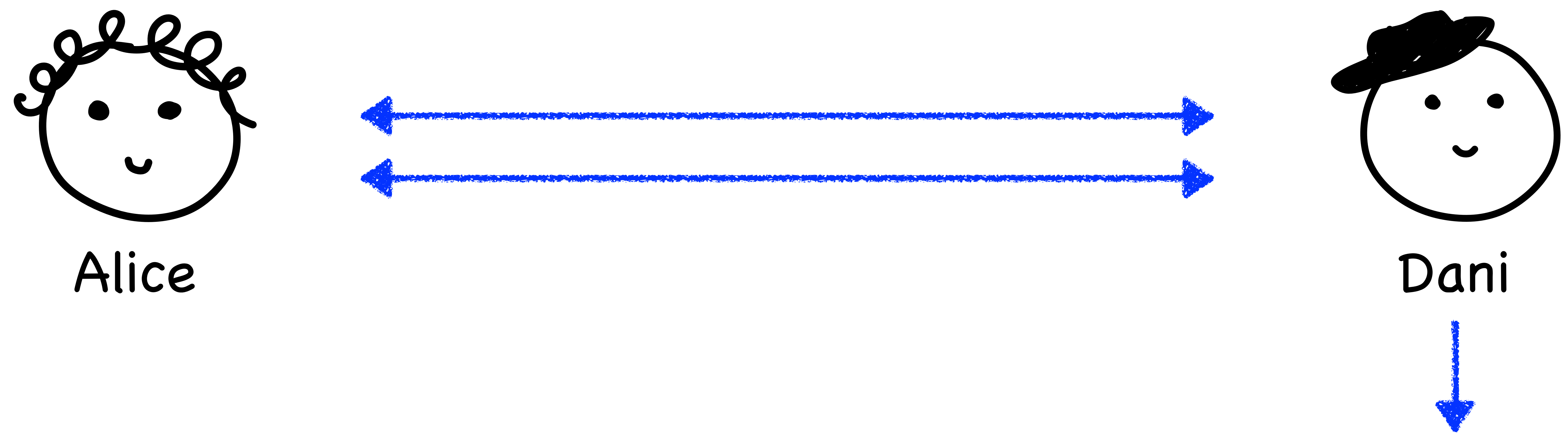


Dani

No – it's unsolvable,
you're messing with me!

Constraints:
- row contains 1-9
- column contains 1-9
- sub-square contains 1-9

Definitions of a Zero Knowledge Proof



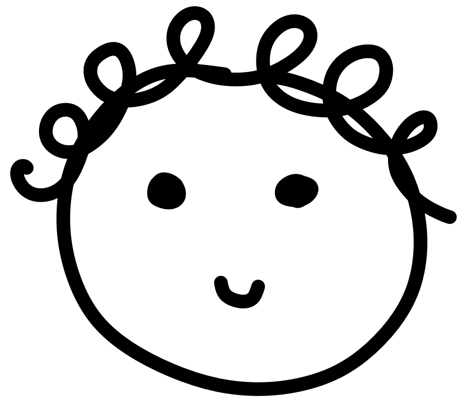
- Completeness: if the statement is true, Dani accept
- Soundness: if the statement is false, Dani rejects, even if Alice cheats
- Zero Knowledge: Dani learns nothing other than the fact that the statement is true

  We formalize this via the existence of a simulator:

  $\forall$ PPT $D^*, \exists$ PPT S s.t. VIEW$(D^*) \equiv S(x)$

Proof of Sudoku Solvability

Goals:
- completeness
- soundness
- ZK



Alice

	7	5		9				6
	2	3		8			4	
8					3			1
5			7		2			
	4		8		6		2	
			9		1			3
9			4					7
	6			7		5	8	
7				1		3	9	

Constraints:
- row contains 1-9
- column contains 1-9
- sub-square contains 1-9

Proof of Sudoku Solvability

Goals:
- completeness
- soundness
- ZK

Alice

random permutation:

1	7	5	2	9	4	8	3	6
6	2	3	1	8	7	9	4	5
8	9	4	5	6	3	2	7	1
5	1	9	7	3	2	4	6	8
3	4	7	8	5	6	1	2	9
2	8	6	9	4	1	7	5	3
9	3	8	4	2	5	6	1	7
4	6	1	3	7	9	5	8	2
7	5	2	6	1	8	3	9	4

1 → 2
2 → 6
3 → 5
4 → 9
5 → 1
6 → 7
7 → 8
8 → 4
9 → 3

2	8	1	6	3	9	4	5	7
7	6	5	2	4	8	3	9	1
4	3	9	1	7	5	6	8	2
1	2	3	8	5	6	9	7	4
5	9	8	4	1	7	2	6	3
6	4	7	3	9	2	8	1	6
3	5	4	9	6	1	7	2	8
9	7	2	5	8	3	1	4	6
8	1	6	7	2	4	5	3	9

if the original sudoku was "valid", so is the new one!

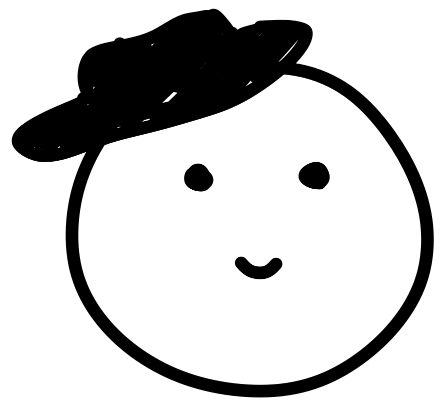
Constraints:
- row contains 1-9
- column contains 1-9
- sub-square contains 1-9

Proof of Sudoku Solvability

Goals:
✓ completeness
✓ soundness
- ZK

Alice

2	8	1	6	3	9	4	5	7
7	6	5	2	4	8	3	9	1
4	3	9	1	7	5	6	8	2
1	2	3	8	5	6	9	7	4
5	9	8	4	1	7	2	6	3
6	4	7	3	9	2	8	1	6
3	5	4	9	6	1	7	2	8
9	7	2	5	8	3	1	4	6
8	1	6	7	2	4	5	3	9



Dani

Constraints:
- row contains 1-9
- column contains 1-9
- sub-square contains 1-9
- "initial conditions": a permutation on 1, ..., 9 maps the original black numbers to the ones here

Q: what does Dani check?

If Alice could fool Dani, she could solve the sudoku!

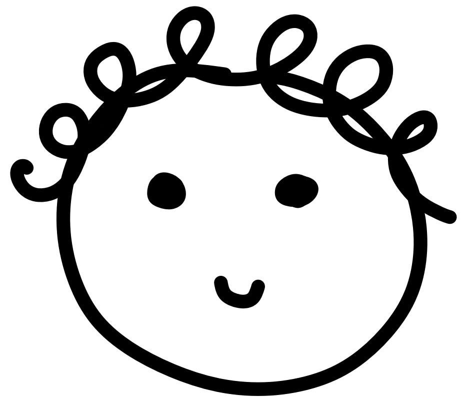
Proof of Sudoku Solvability

Goals:

✓ completeness

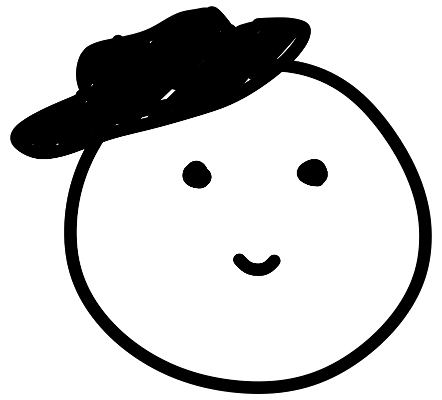
✓ soundness

✗ ZK



Alice

2	8	1	6	3	9	4	5	7
7	6	5	2	4	8	3	9	1
4	3	9	1	7	5	6	8	2
1	2	3	8	5	6	9	7	4
5	9	8	4	1	7	2	6	3
6	4	7	3	9	2	8	1	6
3	5	4	9	6	1	7	2	8
9	7	2	5	8	3	1	4	6
8	1	6	7	2	4	5	3	9



Dani

Constraints:

- row contains 1–9
- column contains 1–9
- sub-square contains 1–9
- "initial conditions": a permutation on 1, …, 9 maps the original black numbers to the ones here

Q: do we get ZK?

A: no! If there existed a simulator that could emulate Dani's view, it could also solve the sudoku.

Proof of Sudoku Solvability

Constraints:
- row contains 1-9
- column contains 1-9
- sub-square contains 1-9
- initial conditions

Q: what does Dani check?

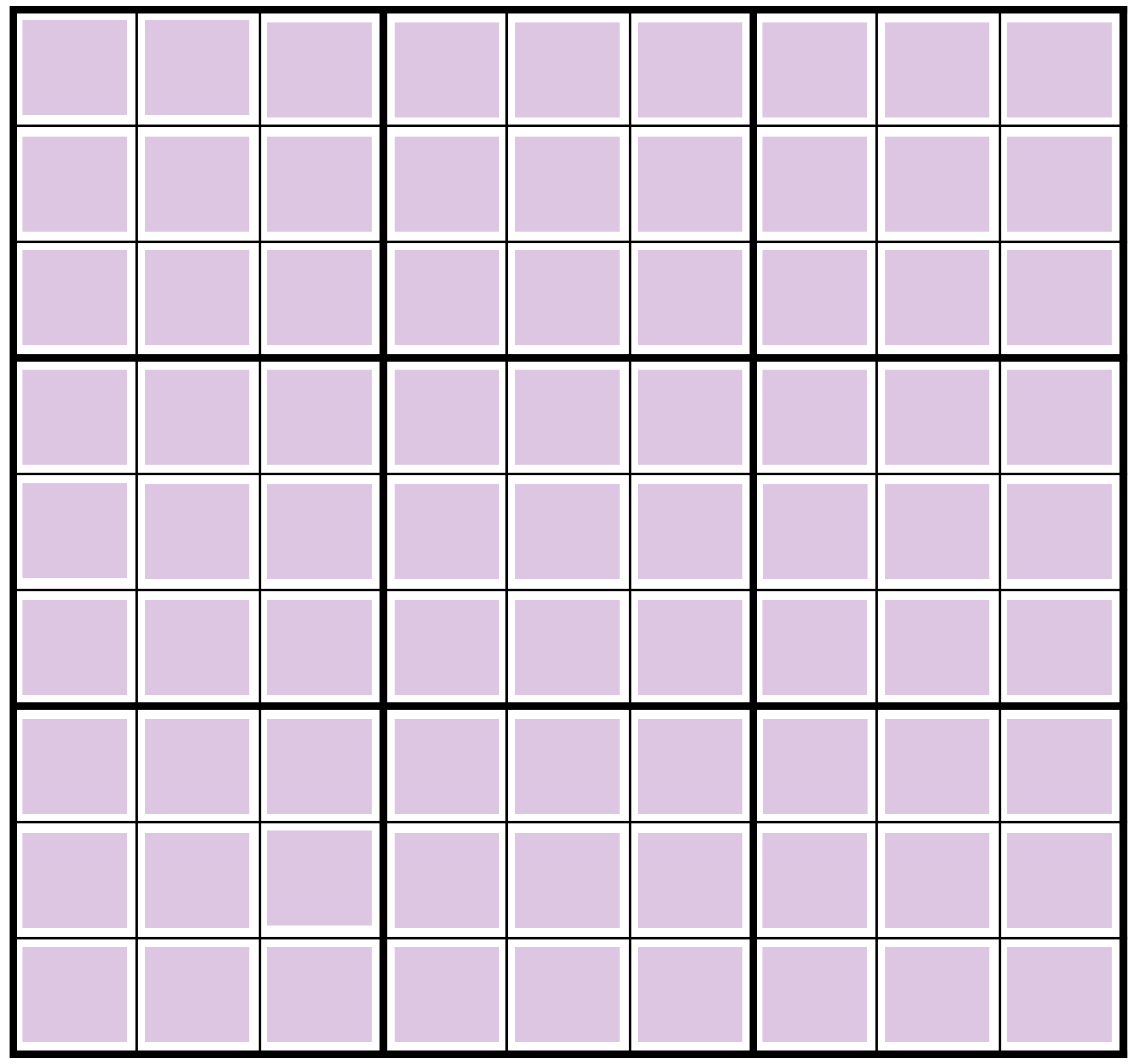
Q: what would the simulator do?

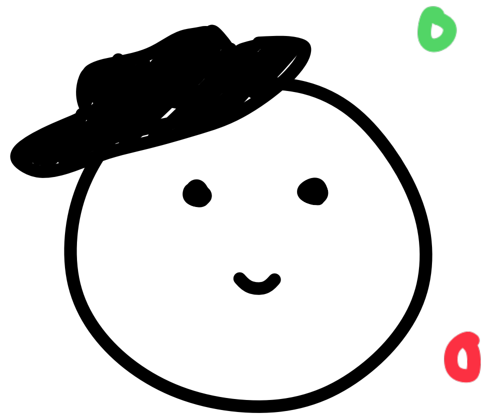
Q: do we get soundness?

Proof of Sudoku Solvability

Goals:
✓ completeness
✓ soundness
✓ ZK


Alice




Dani

I didn't learn anything!

... but if Alice cheated, she might get away with it with prob $\leq 27/28$.

Open constraint i!

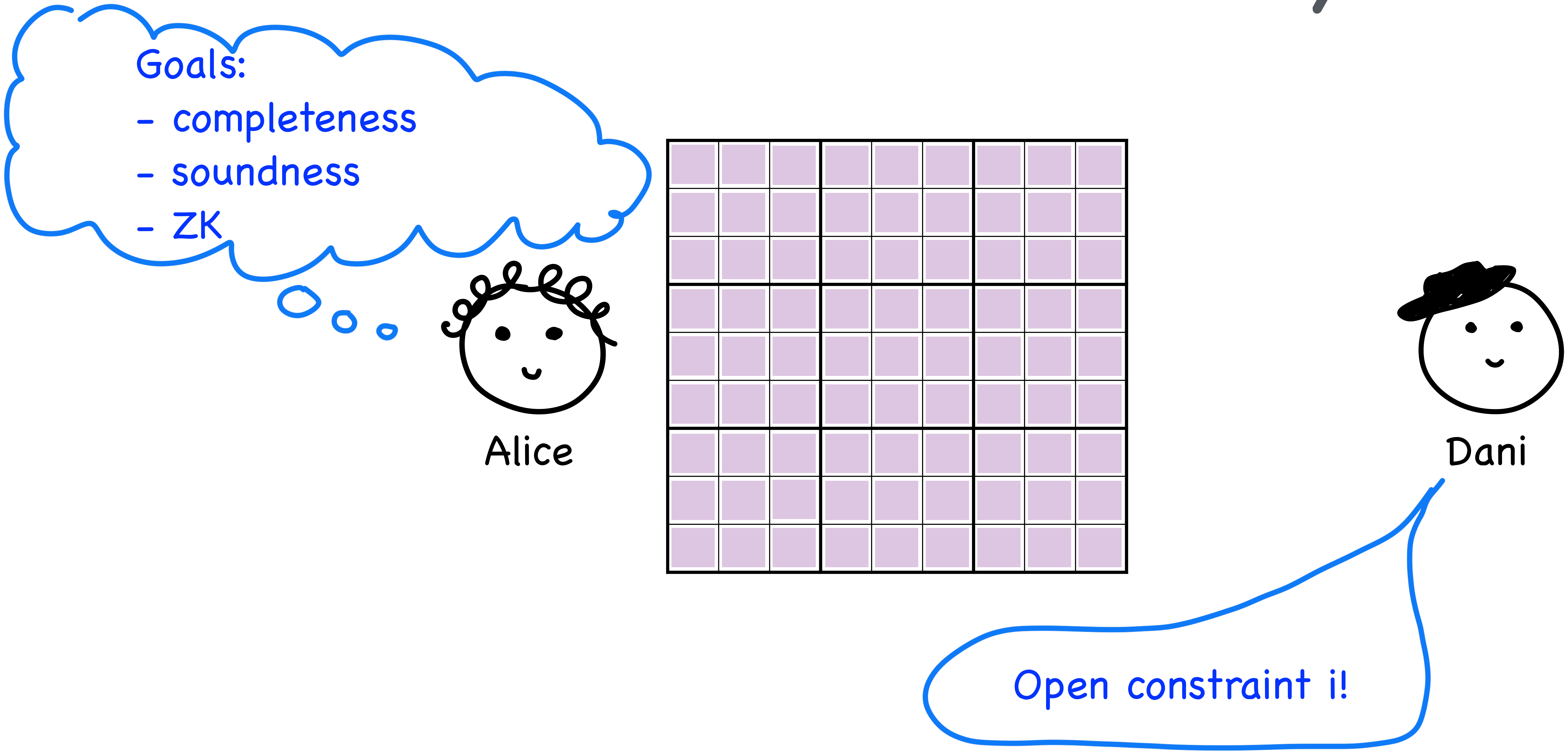
Constraints:
- row contains 1-9
- column contains 1-9
- sub-square contains 1-9
- initial conditions

Q: what would the simulator do?

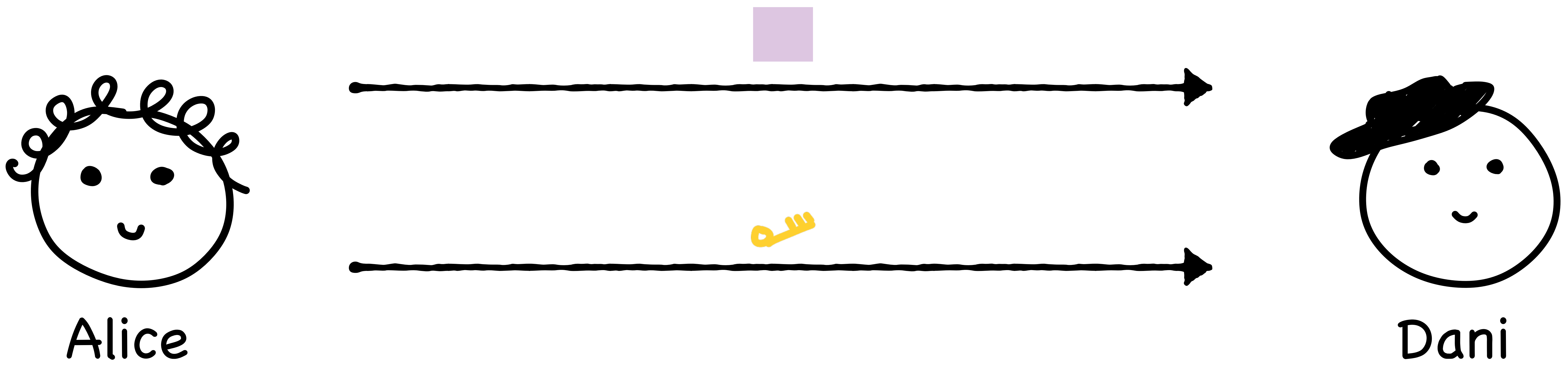
Q: how likely is Alice to get away with it?

Solution: repeat k times, s.t $(27/28)^k$ is small enough.

Proof of Sudoku Solvability... online



Tool: Commitments



Commit(x) $\rightarrow$ (,) Open(,) $\rightarrow x$

Properties:

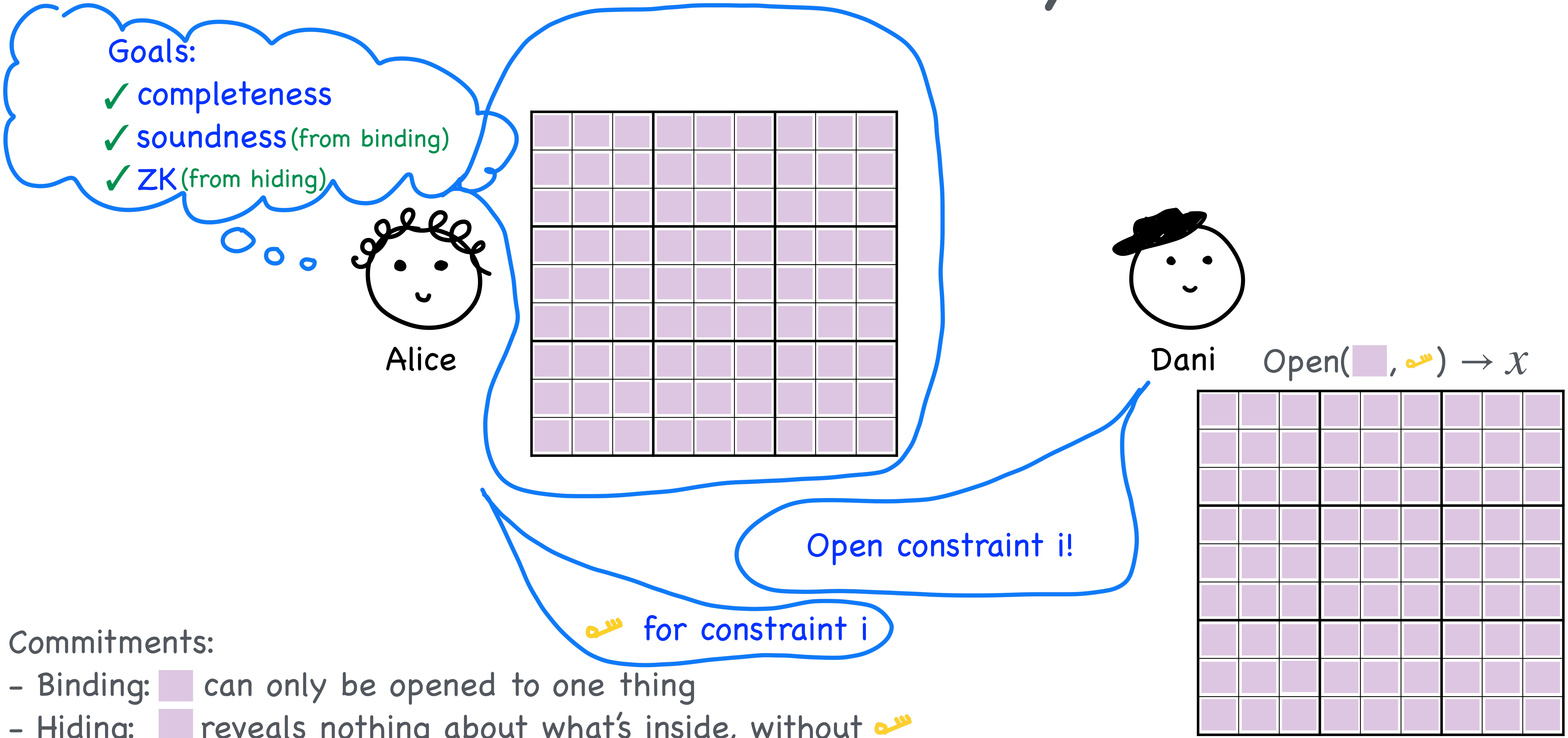
- Hiding: reveals nothing about what's inside, without
- Binding: can only be opened to one thing

Each property can be...

- Perfect (unbreakable even with unlimited resources), or
- Computational (reliant on the hardness of some problem)

Q: how might we build this thing?

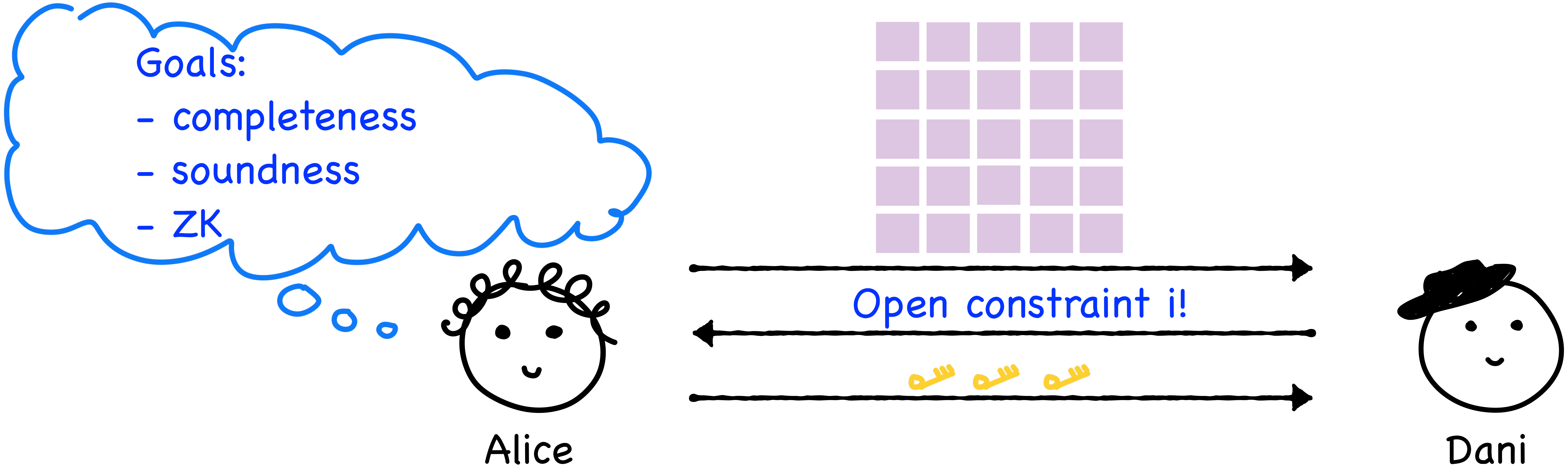
Proof of Sudoku Solvability... online



Commitments:

- Binding: can only be opened to one thing
- Hiding: reveals nothing about what's inside, without

Framework for a ZKP



n constraints on the committed stuff:
- one reveals nothing
- if all of them hold, the statement is true

repeat k times, s.t $(n-1/n)^k$ is small enough.

Back to Reality!

In practice, we want to prove things like...

- Identity, or possession of credentials
- Correct computation
- Generally: knowledge/existence of w s.t. $R(x, w) = 1$

How do we do this?

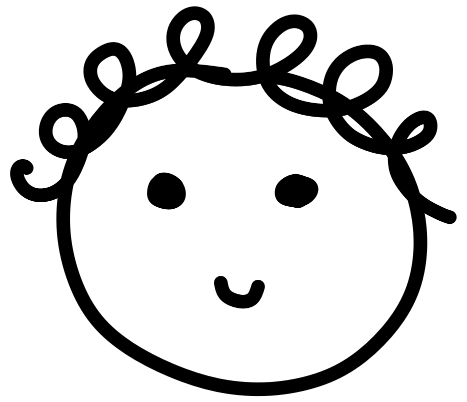
- Sudoku is NP-complete - we can t[illegible]o a sudoku!

inefficient

Zero Knowledge Proofs (ZKP)
and
Secure Multiparty Computation (MPC)

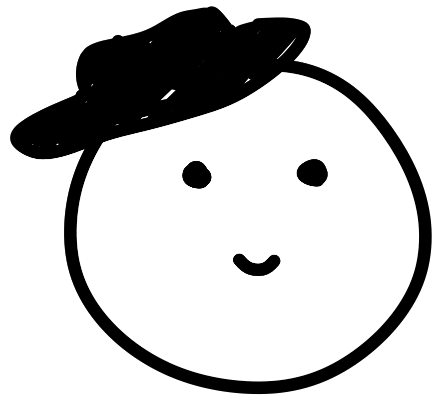
Sophia Yakoubov

Secure Multi-Party Computation (MPC): A First Example



Alice

$x_A \in [1,\ldots,10]$



Dani

$x_D \in [1,\ldots,10]$

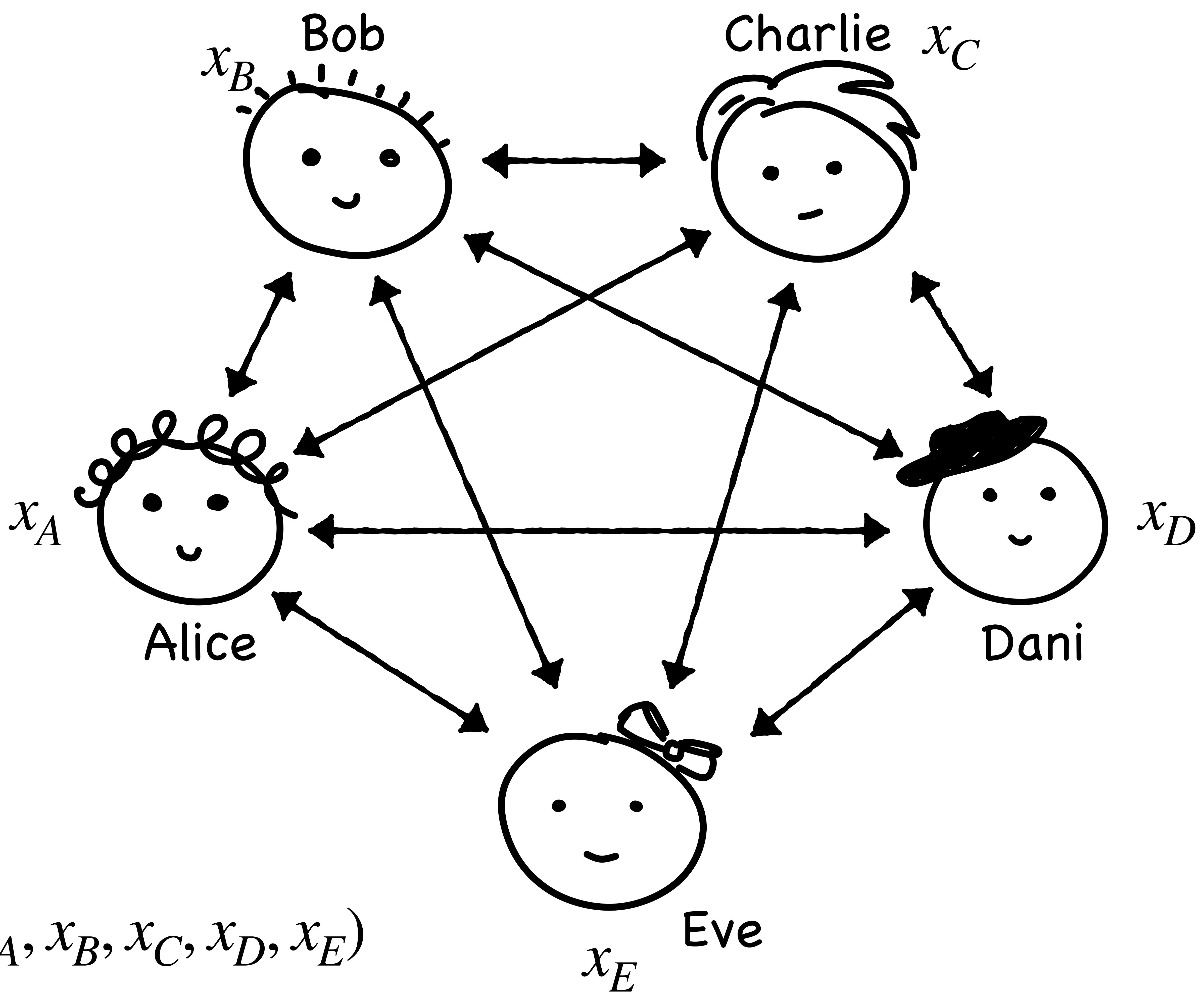
We want privacy:
if $x_A \neq x_D$, that is all they learn

$$f(x_A, x_B) = \begin{cases} 1 & \text{if } x_A = x_B \\ 0 & \text{otherwise} \end{cases}$$

we could ask someone to help us...
but there is no-one we both trust!

Q: can you think of an easy way to do this?

Secure Multi-Party Computation (MPC)

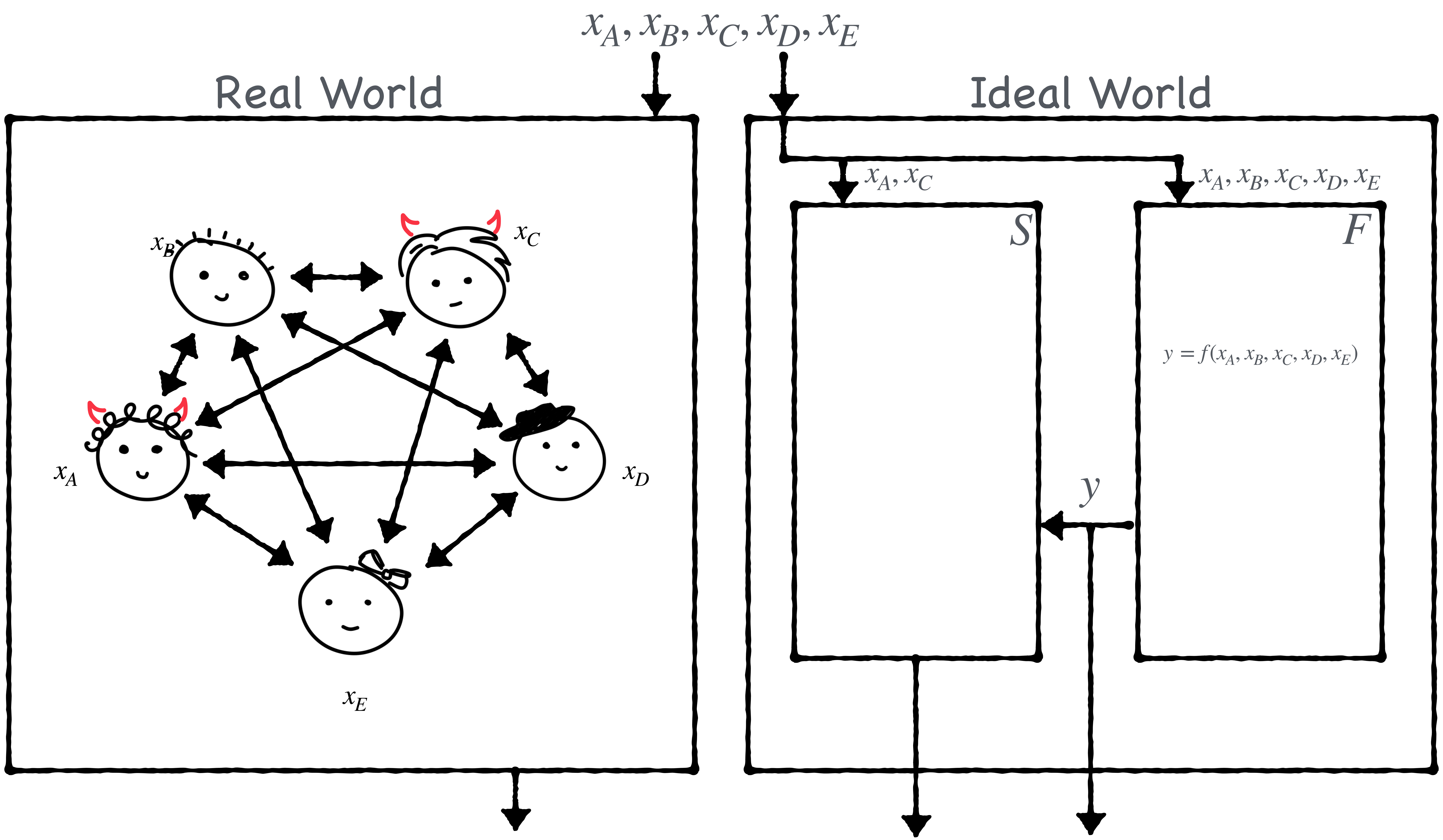


$y = f(x_A, x_B, x_C, x_D, x_E)$

We want:

- correctness
- t-privacy: the combined views of t or fewer participants reveal nothing other than y

Secure Multi-Party Computation (MPC)



$$(\mathrm{REAL}_{A,C}(x_A, x_C), y) \equiv (\mathrm{IDEAL}_{A,C}(x_A, x_C), y)$$

We want:

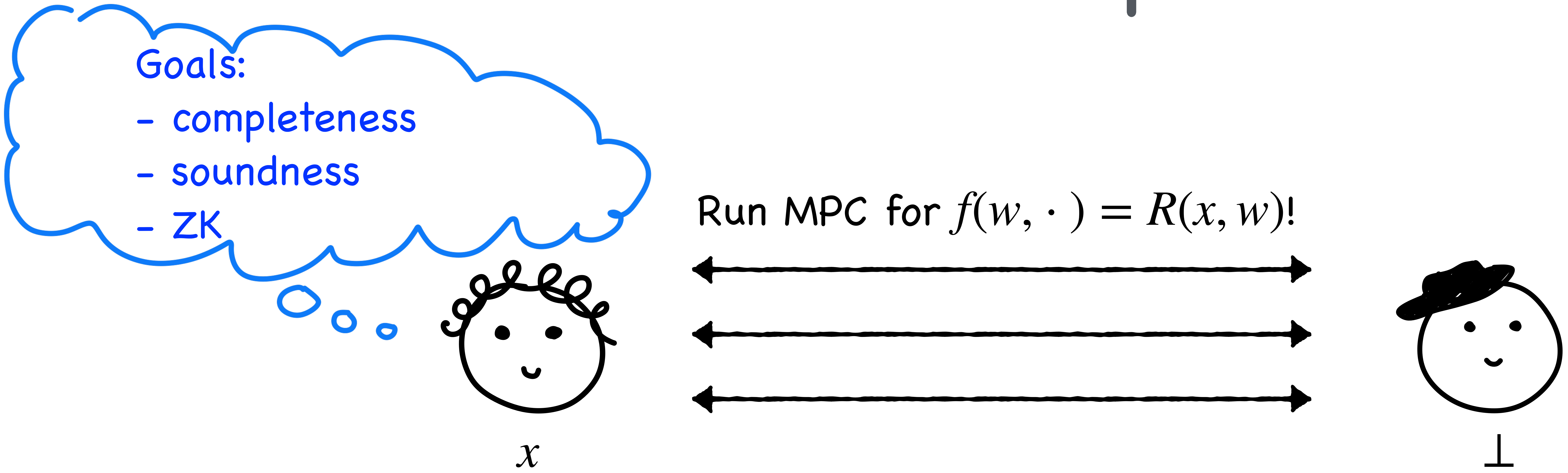
- correctness
- t-privacy: the combined views of t or fewer participants reveal nothing other than y

Zero Knowledge Proofs (ZKP)

$\Uparrow$

Secure Multiparty Computation (MPC)

ZKP from MPC: Attempt 1



	Communication Complexity	Tools
Reduce to Sudoku (or something…)	$poly(k, \lvert R \rvert)$	lightweight (commitments)
Run 2PC	$O(k \lvert R \rvert)$	heavyweight (i.e. "public key" operations)

MPC from Lightweight Tools

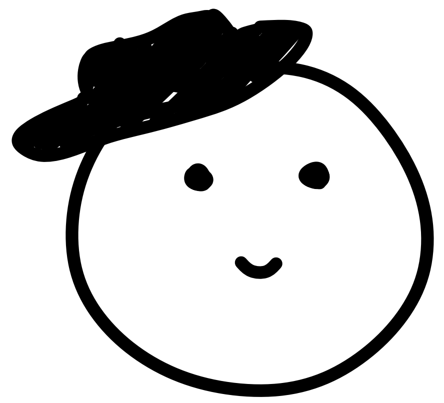
Workarounds:

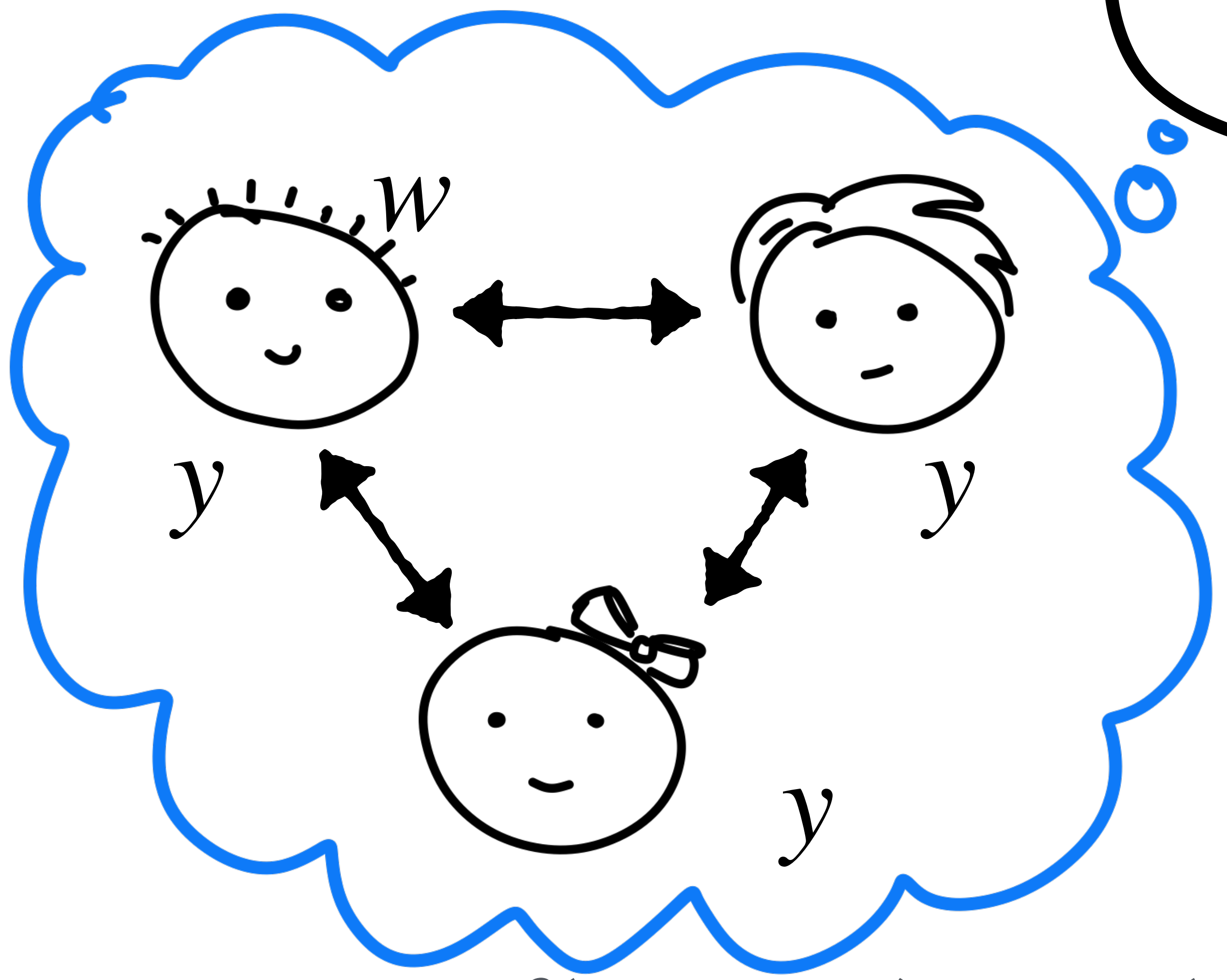
- More participants …
  we can get t-privacy for $t < \frac{n}{2}$ using only lightweight tools
  e.g.: $n = 3, t = 1$
- Correlated randomness

ZKP from MPC: Attempt 2

Goals:
- completeness
- soundness
- ZK

x



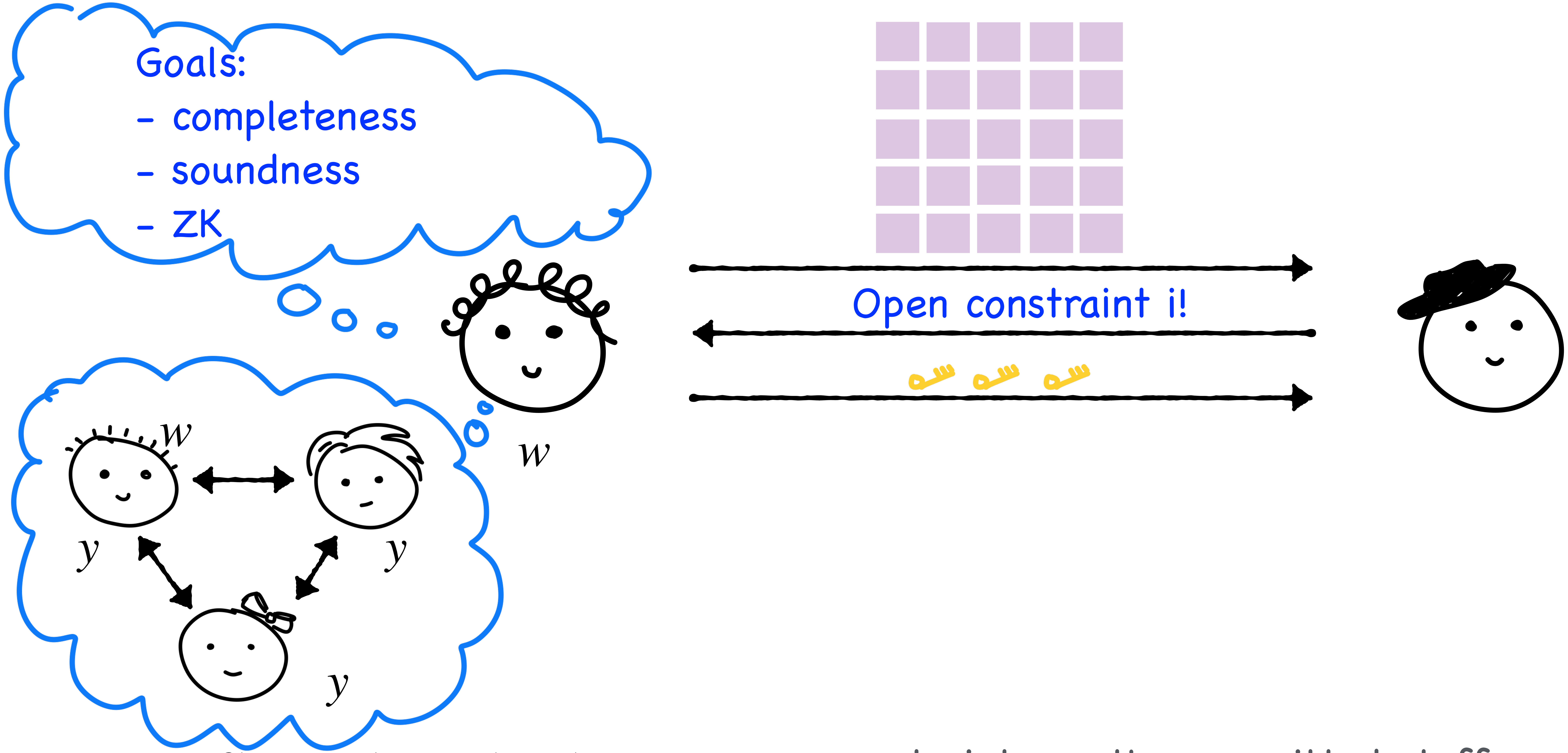


w

MPC for $f(w, \cdot\,, \cdot\,) = R(x, w)$

[Ishai,Kushelevitz,Ostrovsky,Sahai]

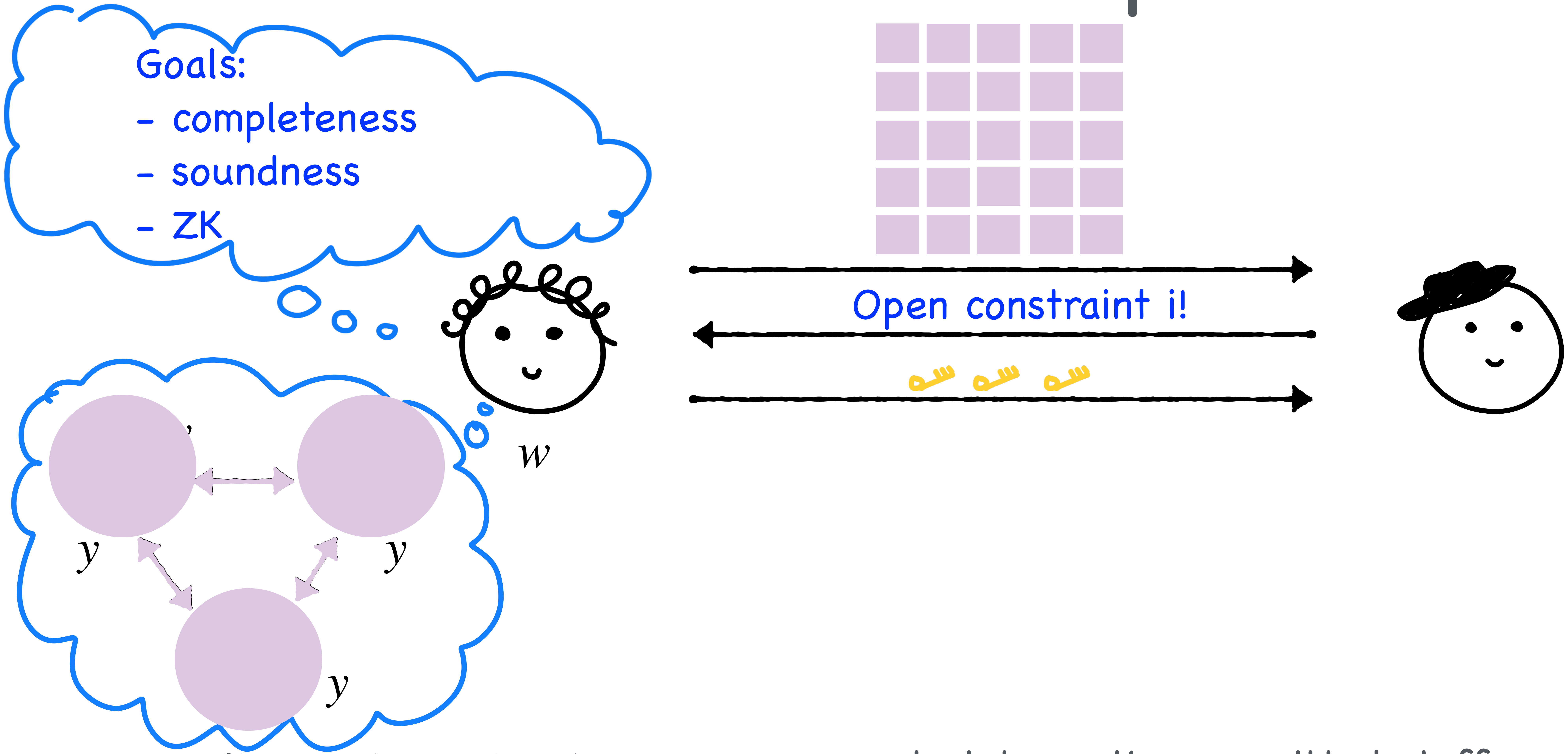
Framework for a ZKP



MPC for $f(w, \cdot, \cdot) = R(x, w)$

n constraints on the committed stuff:
- one reveals nothing
- if all of them hold, the statement is true

ZKP from MPC: Attempt 2



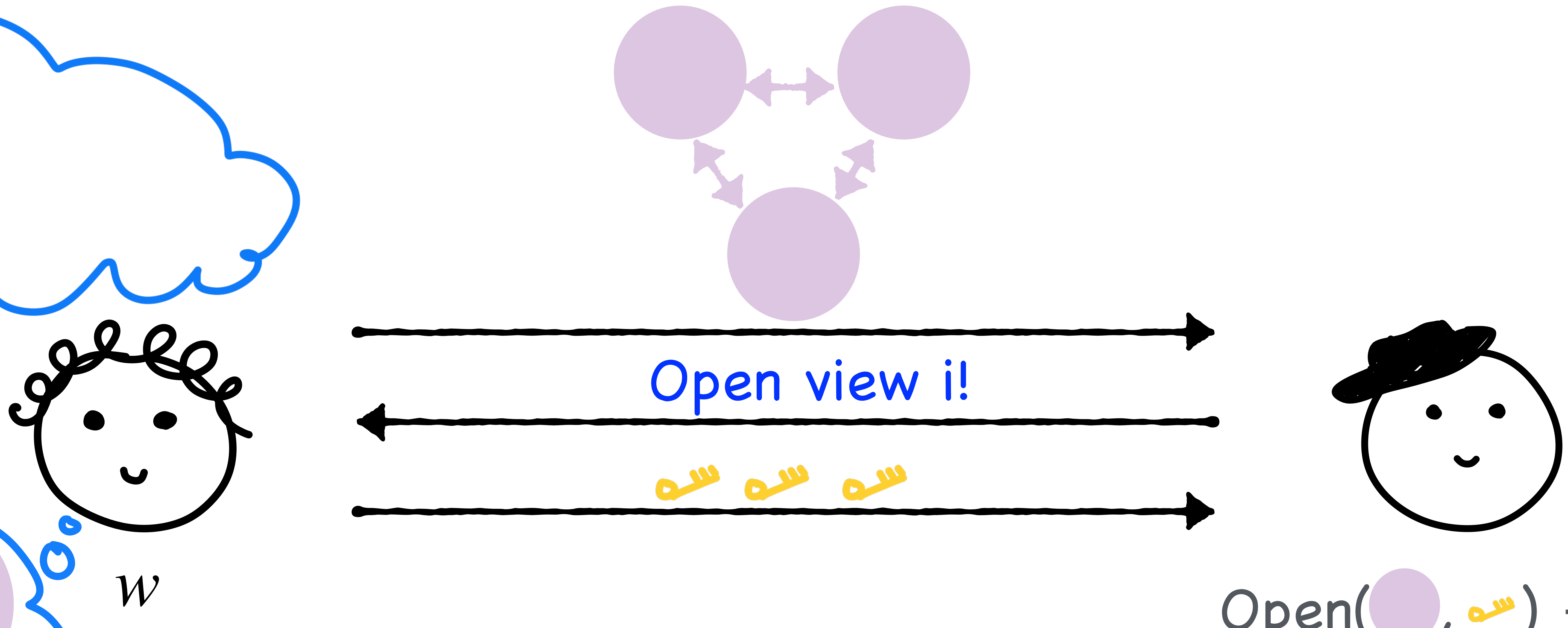
MPC for $f(w, \cdot\,, \cdot\,) = R(x, w)$

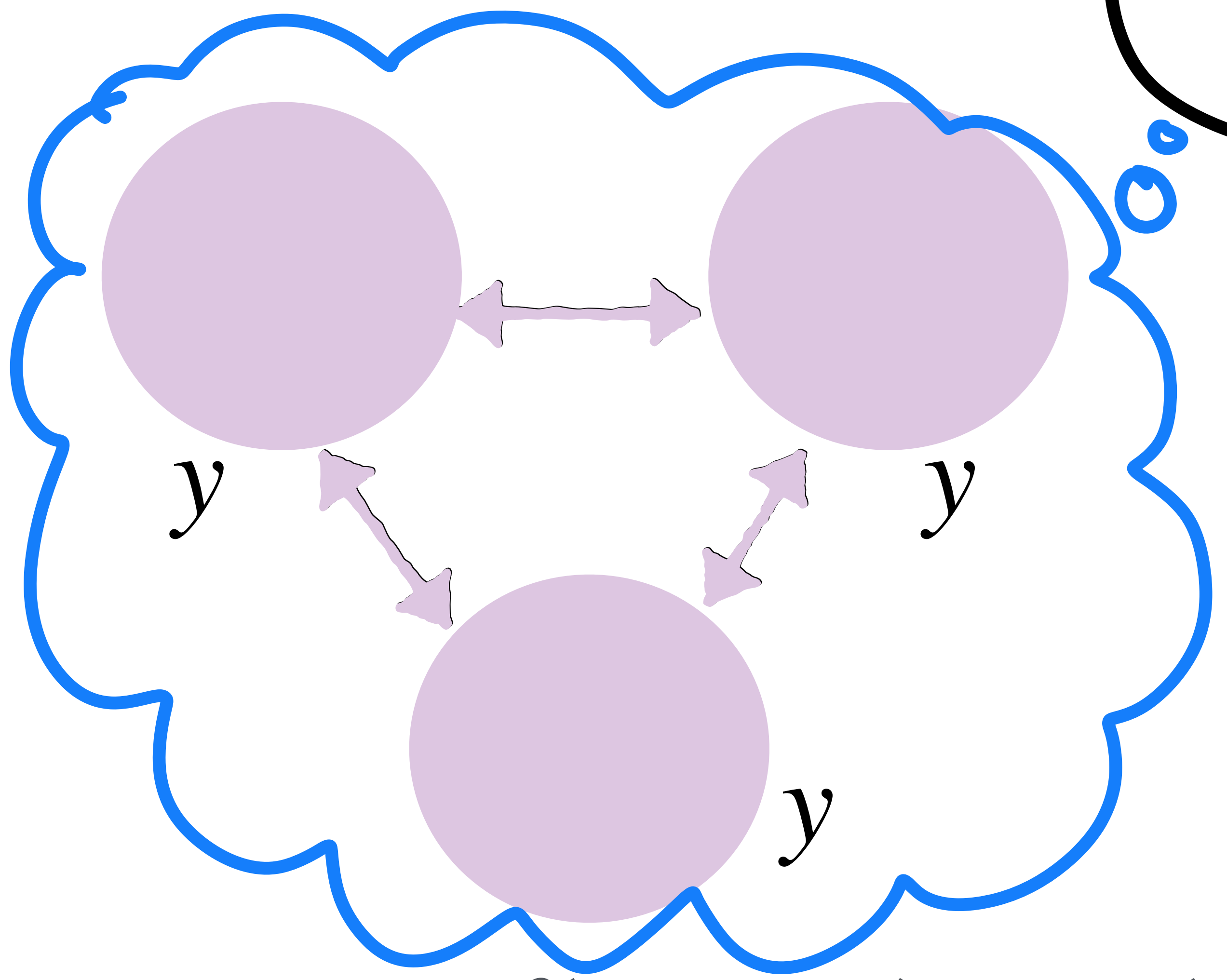
n constraints on the committed stuff:

- one reveals nothing
- if all of them hold, the statement is true

ZKP from MPC: Attempt 2

Goals:
- completeness
- soundness
- ZK





MPC for $f(w, \cdot, \cdot) = R(x, w)$

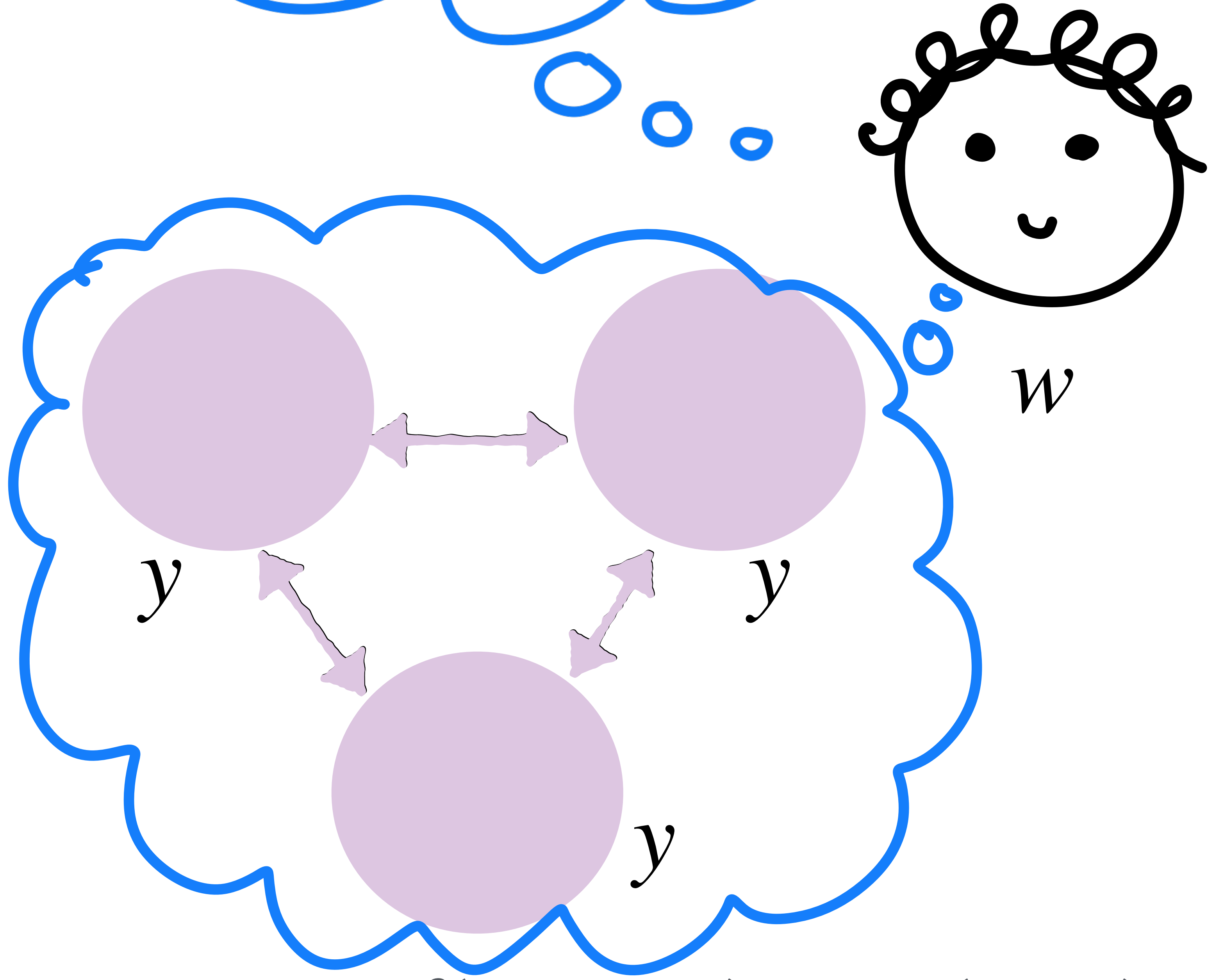
Open(,) → party i's choices
Open(,) → messages
Open(,) → messages

party i did not cheat,
and output is 1

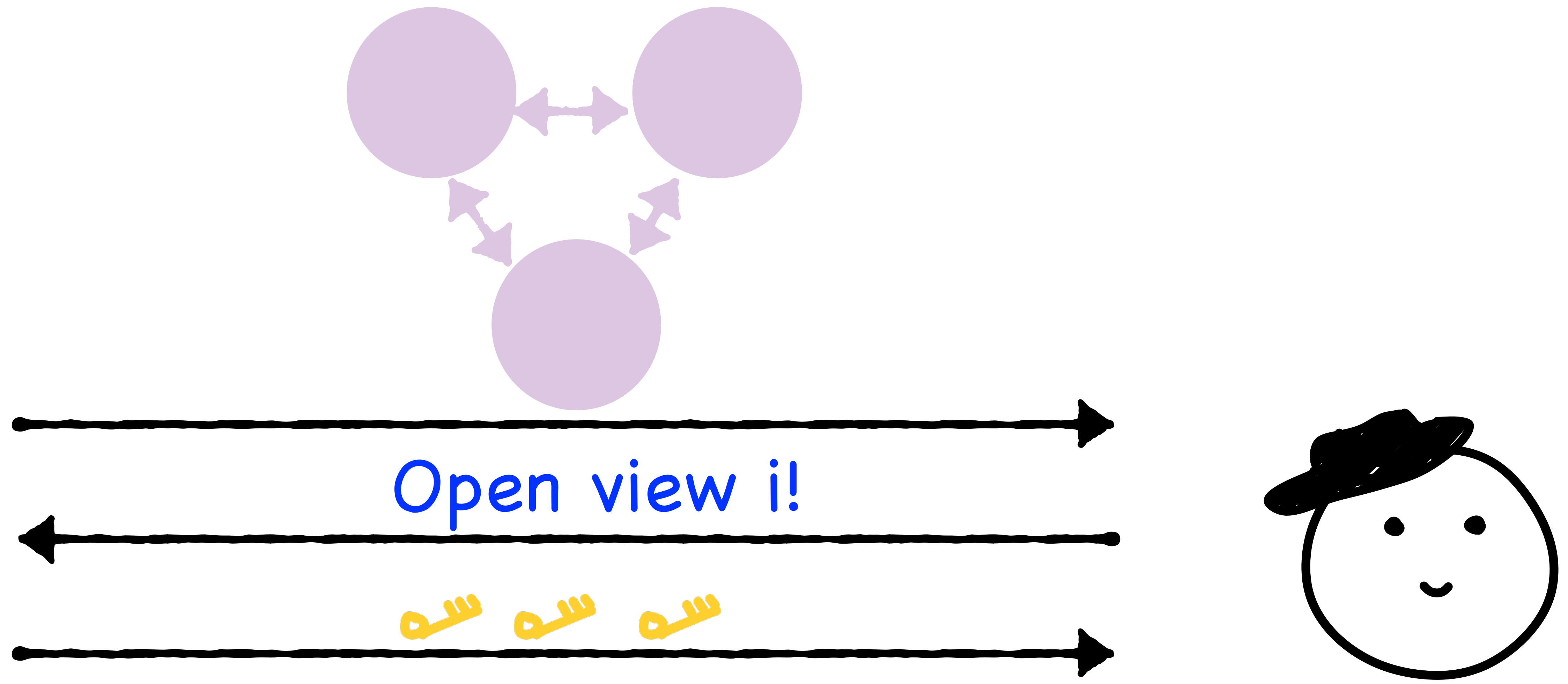
n constraints on the committed stuff:
- one reveals nothing
- if all of them hold, the statement is true

Completeness... follows from MPC correctness

Goals:
- ✓ completeness
- soundness
- ZK



MPC for $f(w, \cdot\,, \cdot\,) = R(x, w)$



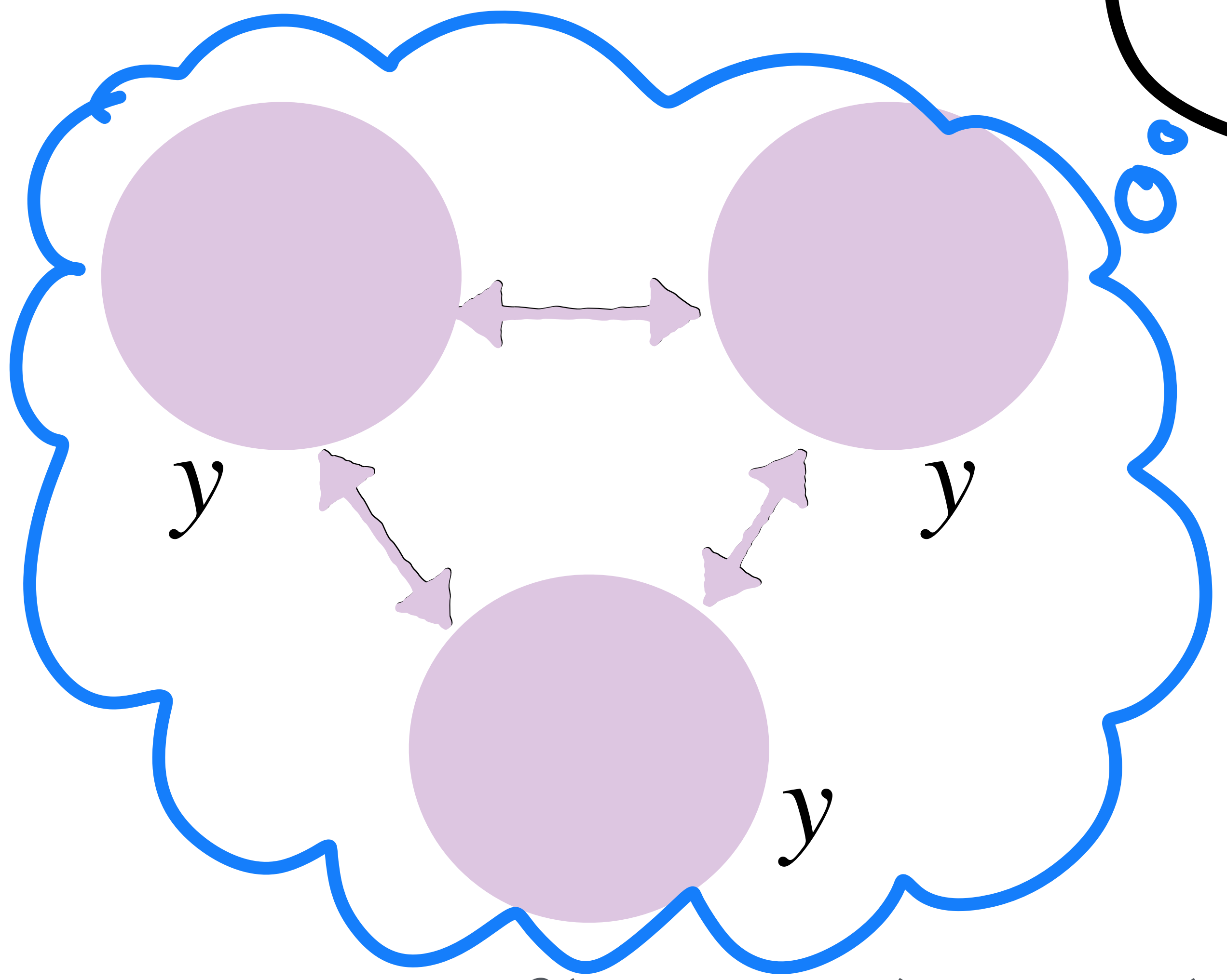
Open(,) → party i's choices
Open(,) → messages
Open(,) → messages

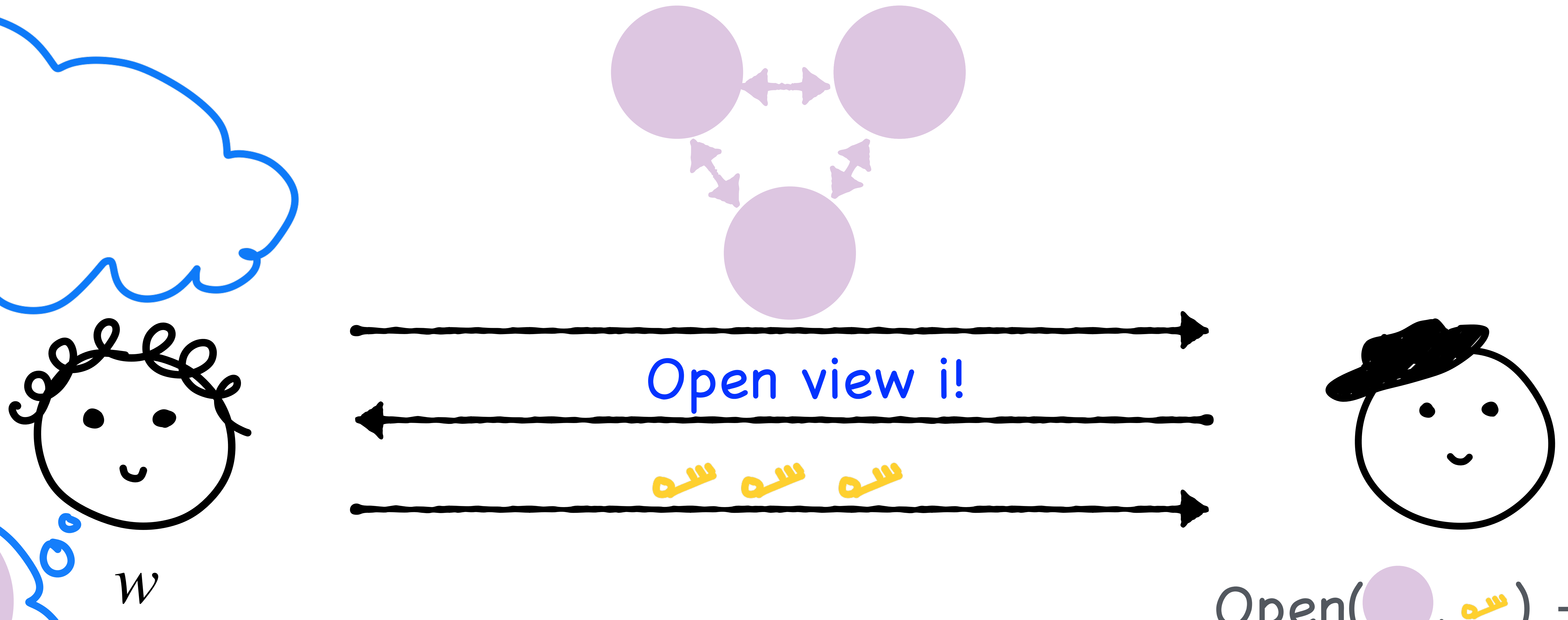
party i did not cheat,
and output is 1

n constraints on the committed stuff:
- one reveals nothing
- if all of them hold, the statement is true

Soundness...

Goals:
✓ completeness
✓ soundness
– ZK





Open(,) → party i's choices
Open(,) → messages
Open(,) → messages

party i did not cheat,
and output is 1

MPC for $f(w, \cdot, \cdot) = R(x, w)$
- 1-privacy

✓ perfect correctness →

n constraints on the committed stuff:
- one reveals nothing

✓ if all of them hold, the statement is true

repeat k times,
s.t $(2/3)^k$ is
small enough.

To convince Dani, Alice must cheat on behalf of at least one party.

Q: are we there yet?

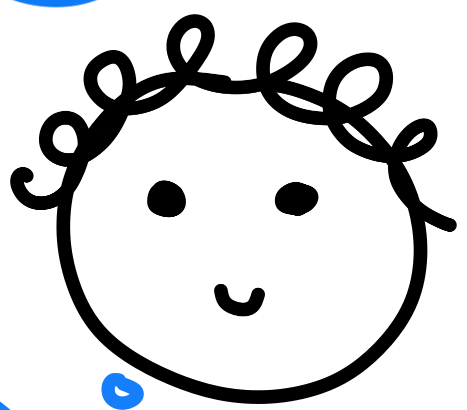
ZK...

Goals:

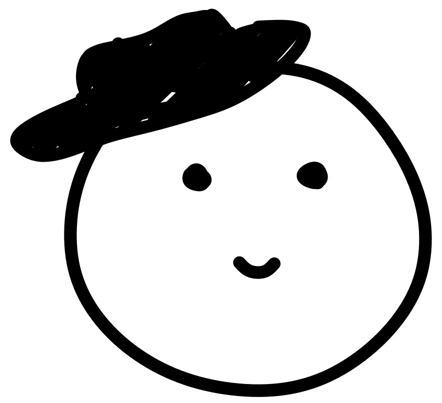
✓ completeness

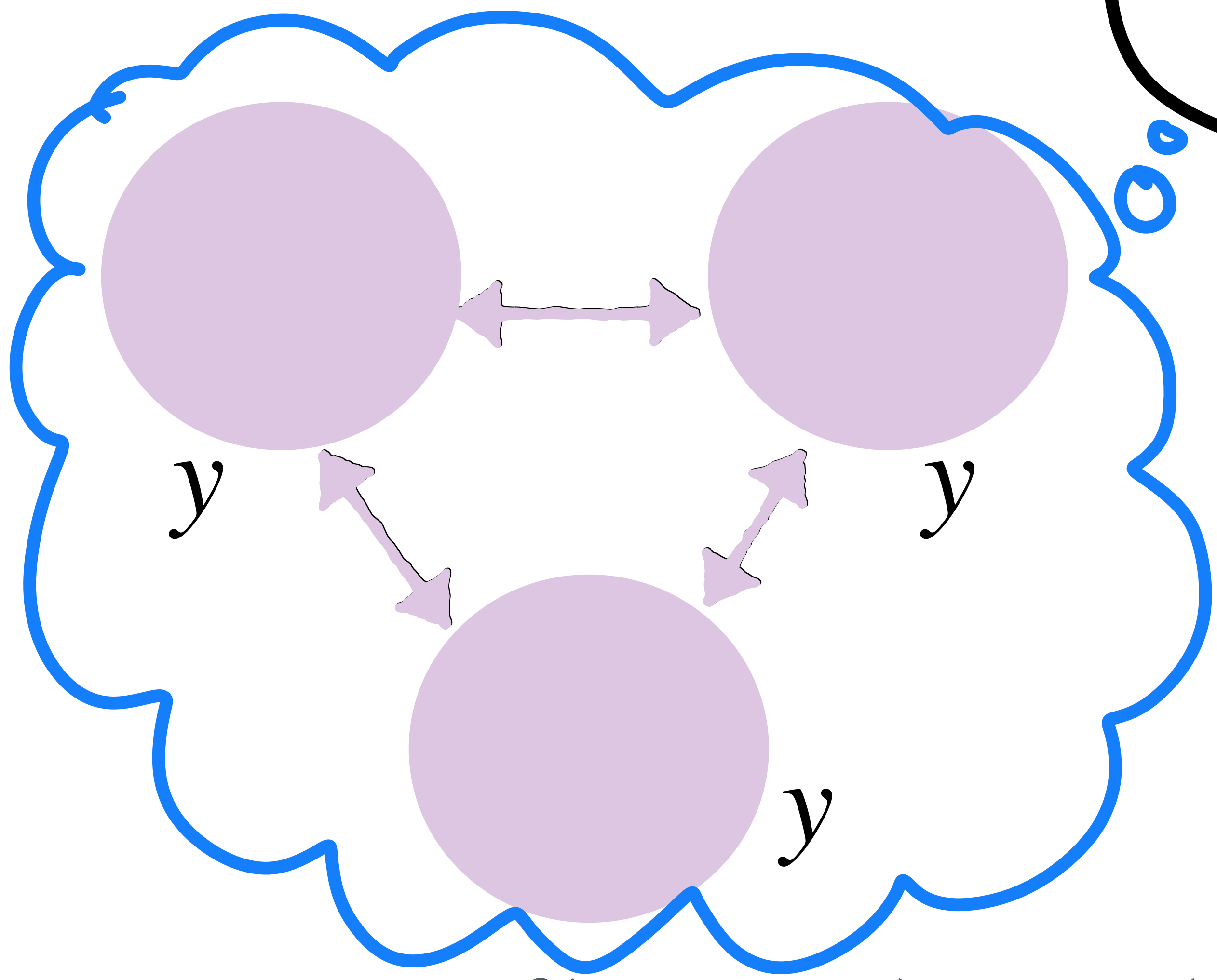
✓ soundness

- ZK



Open view i!





Open(,) → party i's choices

Open(,) → messages

Open(,) → messages

party i did not cheat, and output is 1

MPC for $f(w, \cdot, \cdot) = R(x, w)$

- 1-privacy

✓ perfect correctness

n constraints on the committed stuff:

- one reveals nothing

✓ if all of them hold, the statement is true

repeat k times, s.t $(2/3)^k$ is small enough.

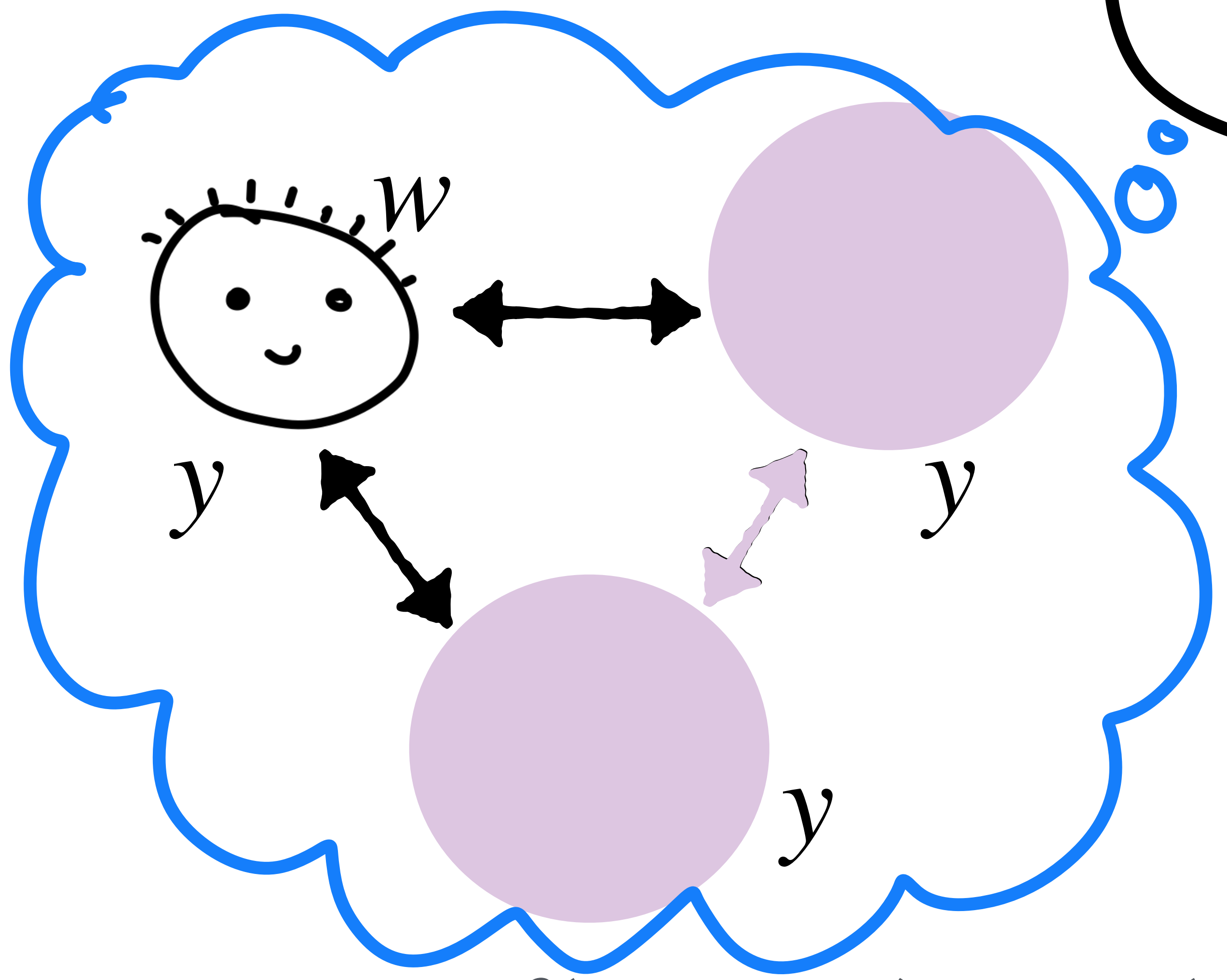
ZK...

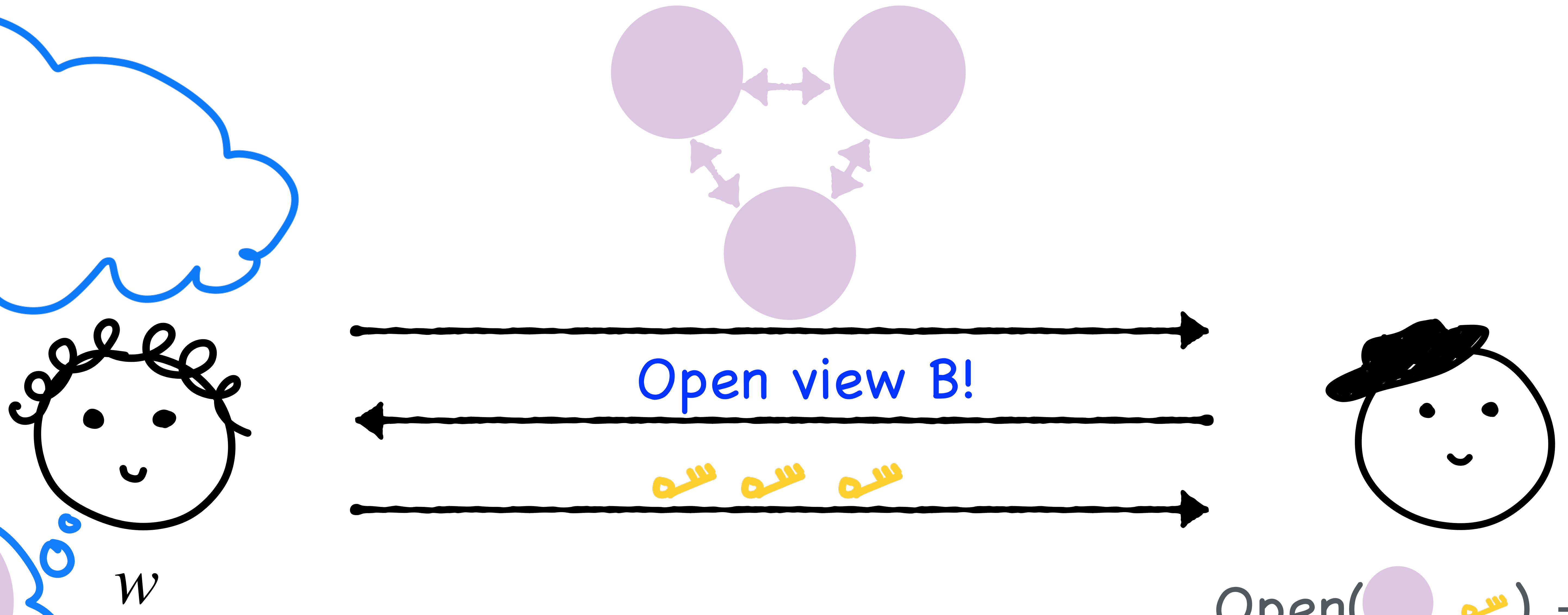
Goals:

✓ completeness

✓ soundness

✗ ZK





Open(,) → party i's choices

Open(,) → messages

Open(,) → messages

MPC for $f(w, \cdot, \cdot) = R(x, w)$

- 1-privacy

✓ perfect correctness

party i did not cheat, and output is 1

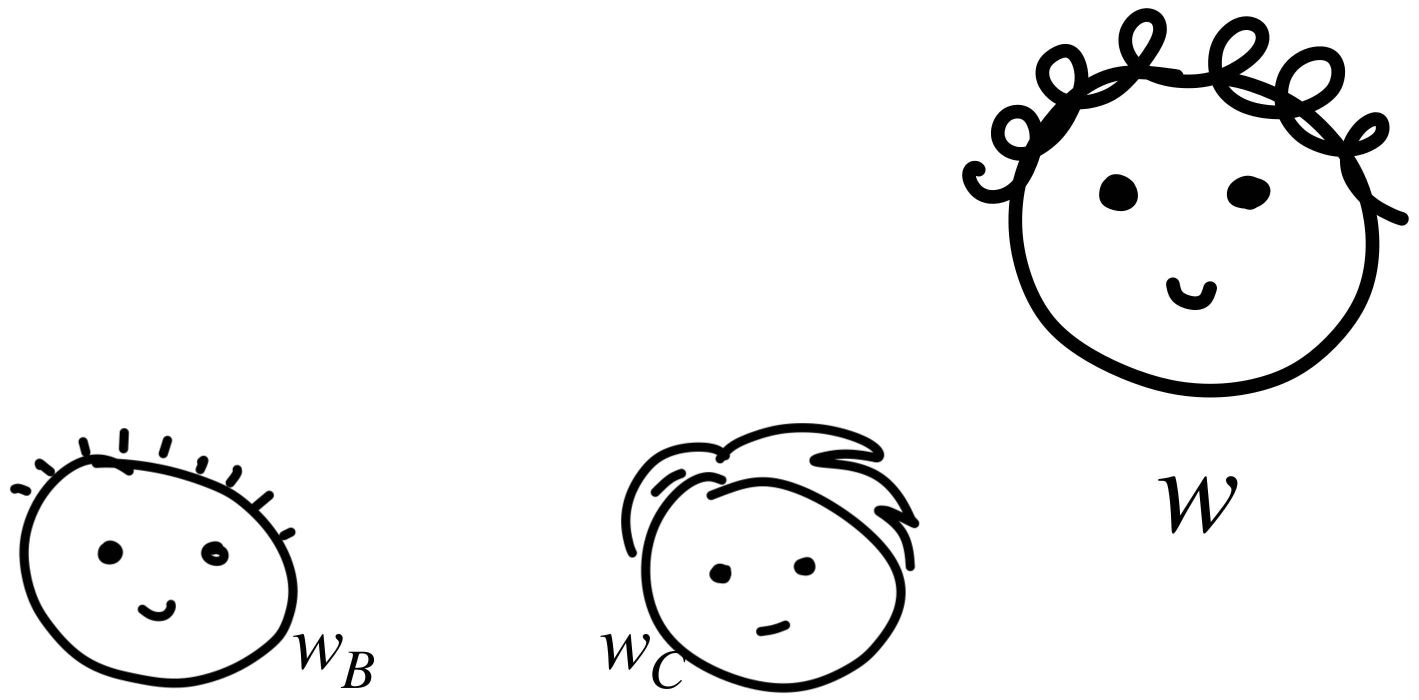
n constraints on the committed stuff:

- one reveals nothing

✓ if all of them hold, the statement is true

repeat k times, s.t $(2/3)^k$ is small enough.

Tool: Secret Sharing

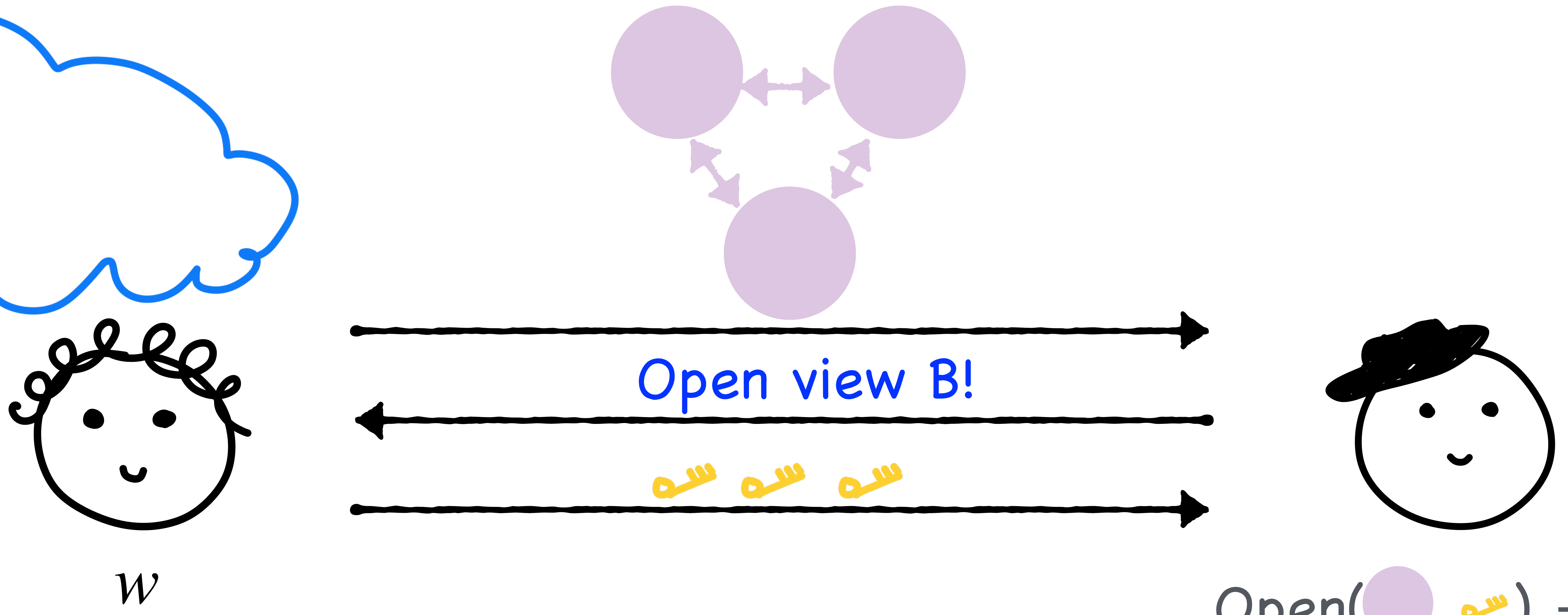


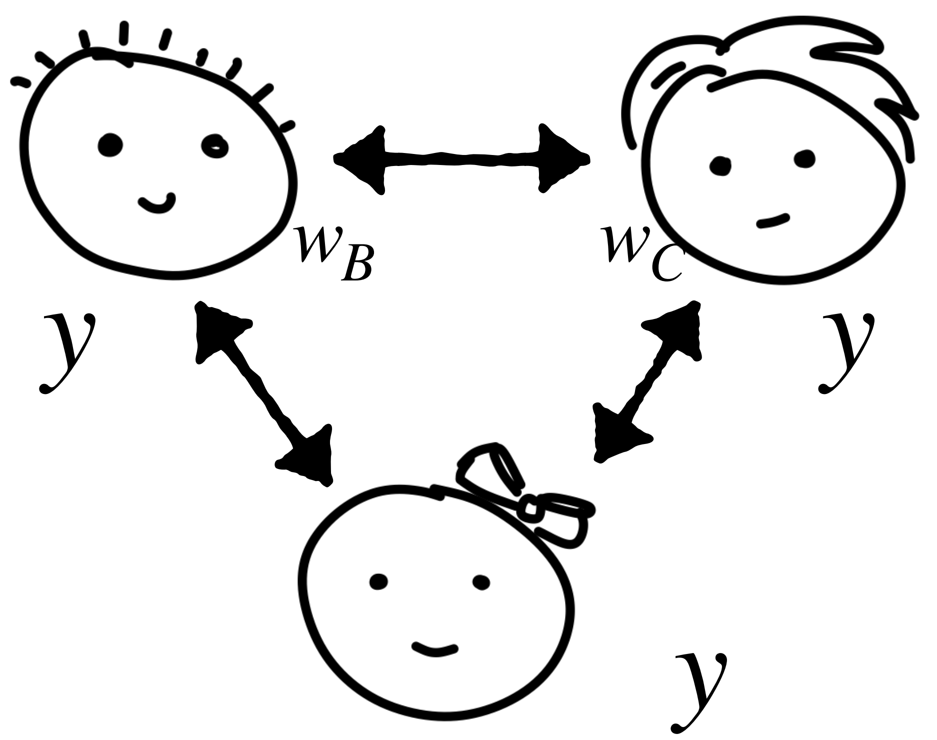
pick random w_B, w_C
s.t. $w = w_B + w_C \bmod m$

privacy: w_B, w_C alone look random
together, they determine w

ZK...

Goals:
- ✓ completeness
- ✓ soundness
- ✓ ZK





Open(,) → party i's choices
Open(,) → messages
Open(,) → messages

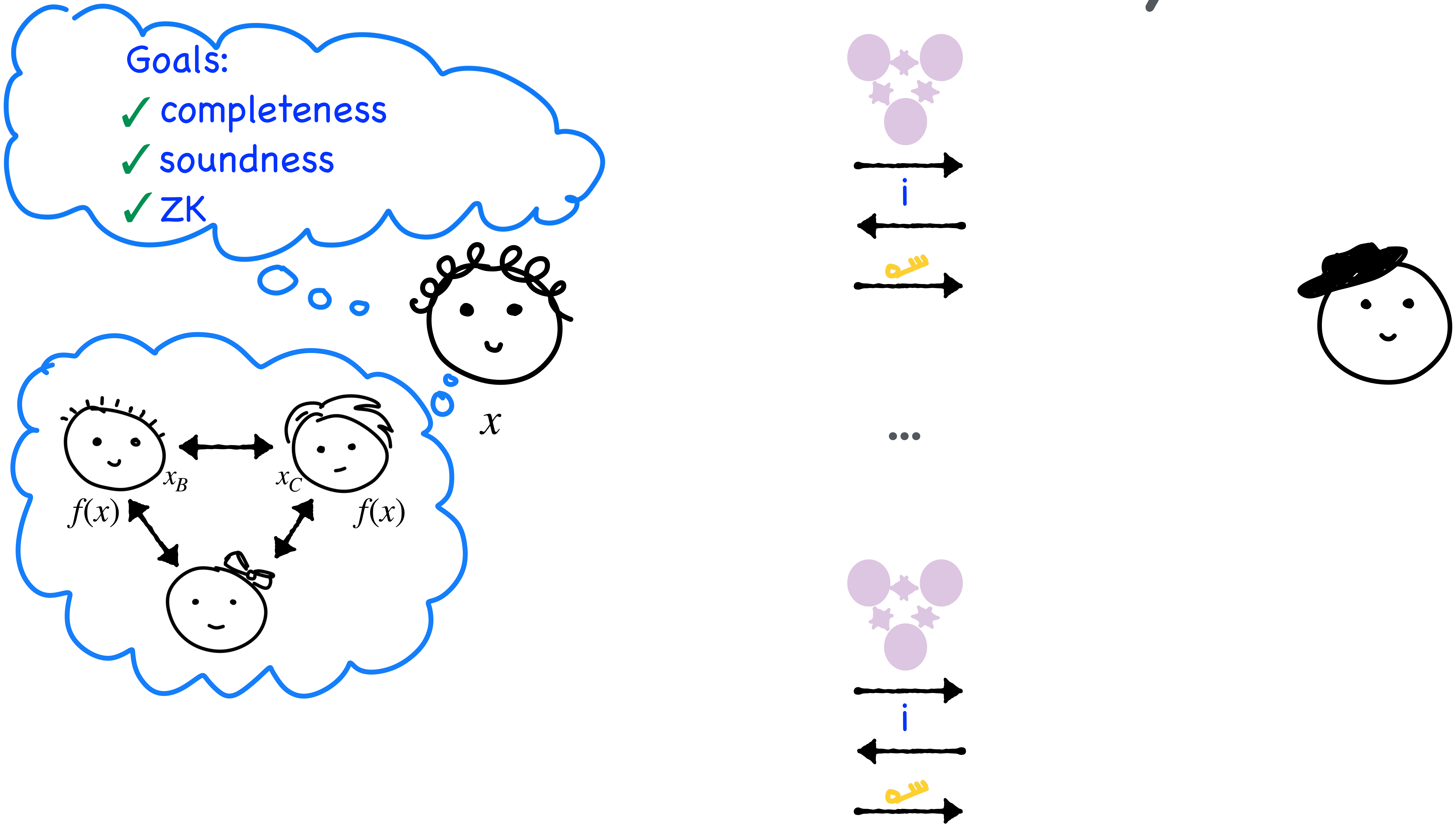
MPC for $f(w_B, w_C, \cdot\,) = R(x, w_B + w_C)$
- ✓ 1-privacy → ✓ one reveals nothing
- ✓ perfect correctness → ✓ if all of them hold, the statement is true

Q: what does the simulator do?

n constraints on the committed stuff:
- ✓ one reveals nothing
- ✓ if all of them hold, the statement is true

repeat k times, s.t $(2/3)^k$ is small enough.

ZKP from MPC: Summary



ZKP from MPC

	Communication Complexity	Tools
Reduce to Sudoku (or something…)	$poly(k, \|R\|)$	lightweight (commitments)
Run MPC	$O(k\|R\|)$	heavyweight (i.e. "public key" operations)
Run MPC in the Head	$O(k\|VIEW\|) = O(k\|R\|)$	

ZKP from MPC

	Communication Complexity	Tools
Reduce to Sudoku (or something…)	$poly(k, \|R\|)$	lightweight (commitments)
Run MPC	$O(k\|R\|)$	heavyweight (i.e. "public key" operations)
Run MPC in the Head	$O(k\|VIEW\|) = O(k\|R\|)$	lightweight (commitments)