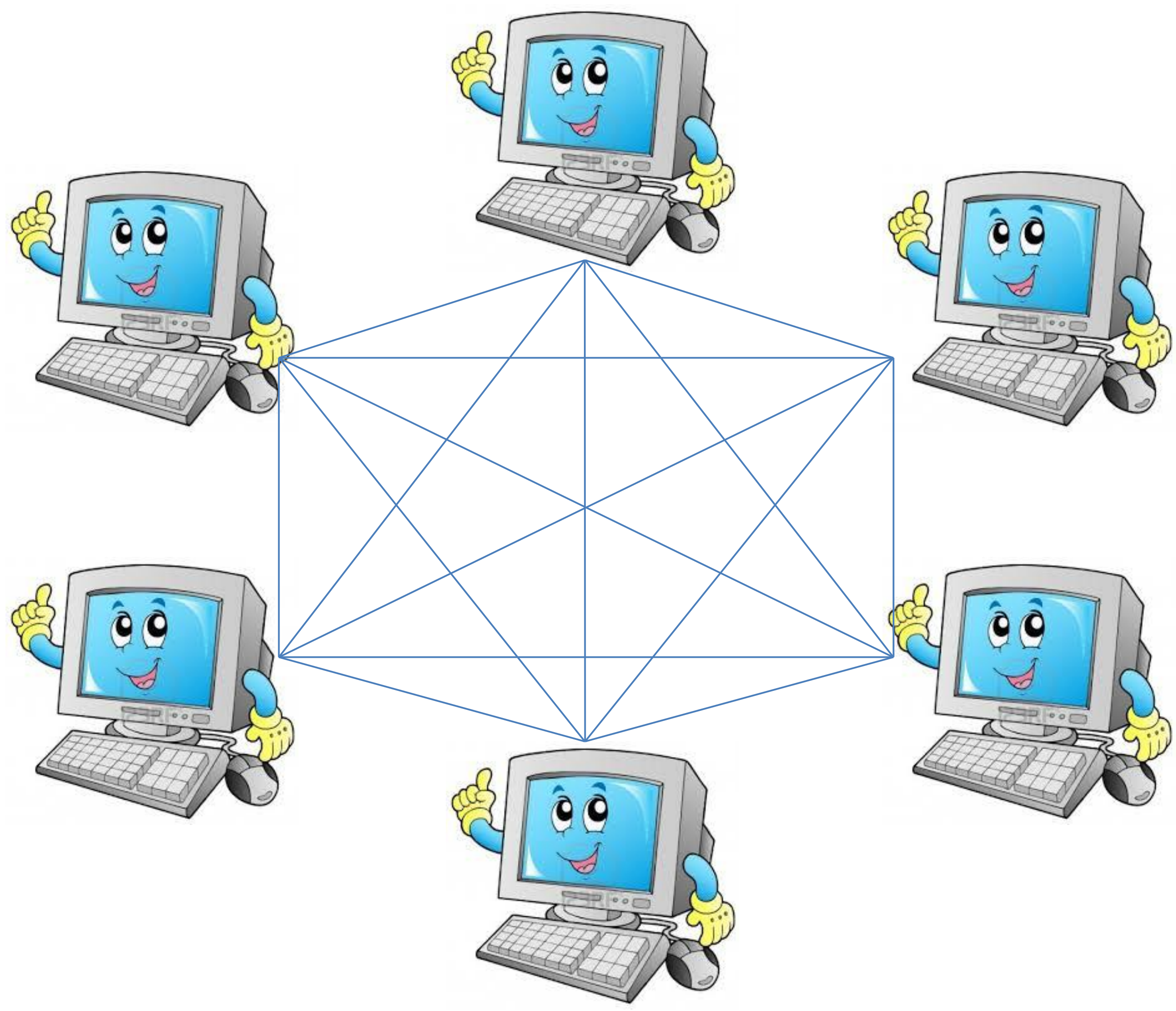
Round-Preserving Parallel Composition of Probabilistic-Termination Cryptographic Protocols

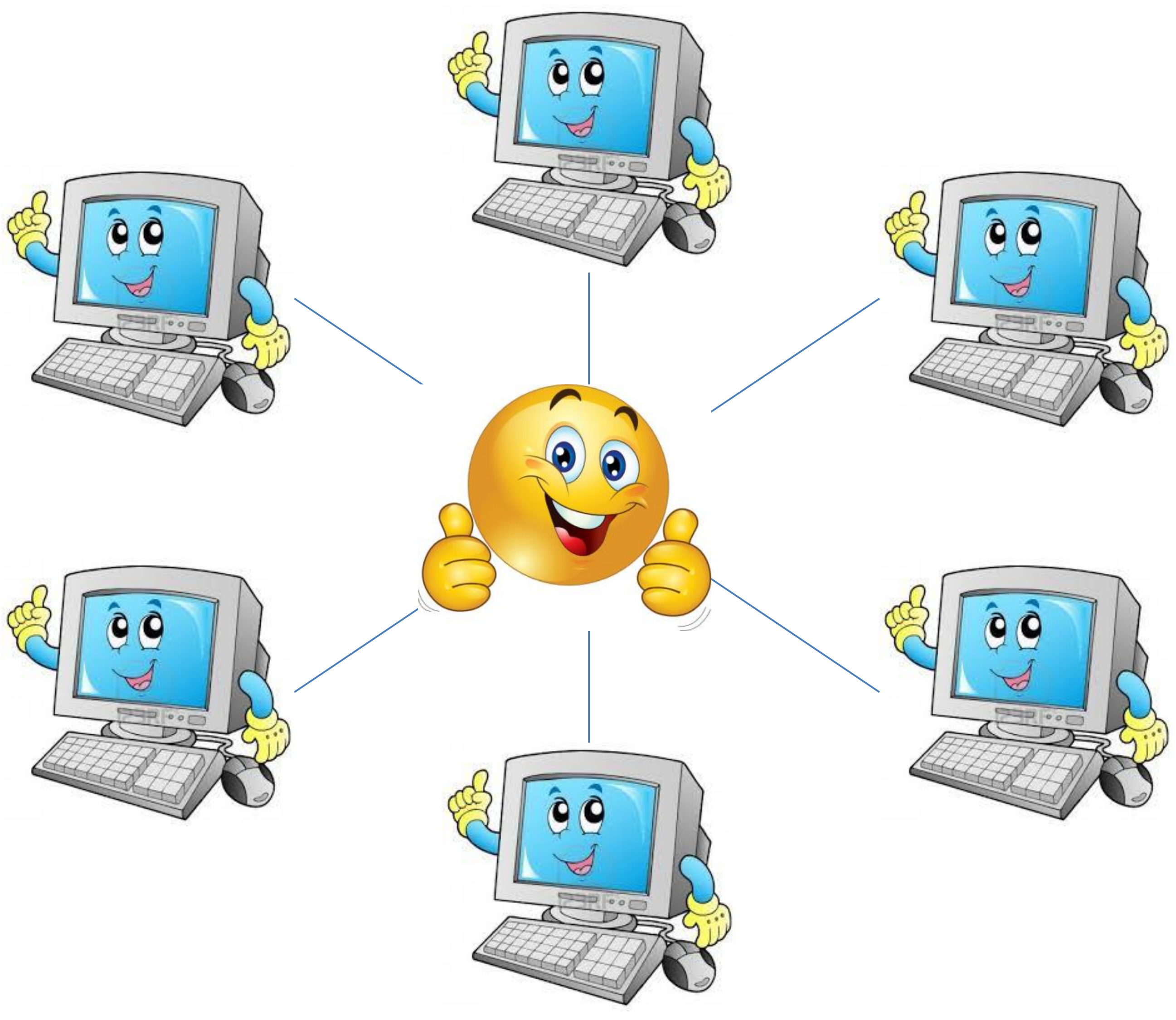
[ICALP'17]

Ran Cohen (TAU)
Sandro Coretti (NYU)
Juan Garay (Yahoo Research)
Vassilis Zikas (RPI)

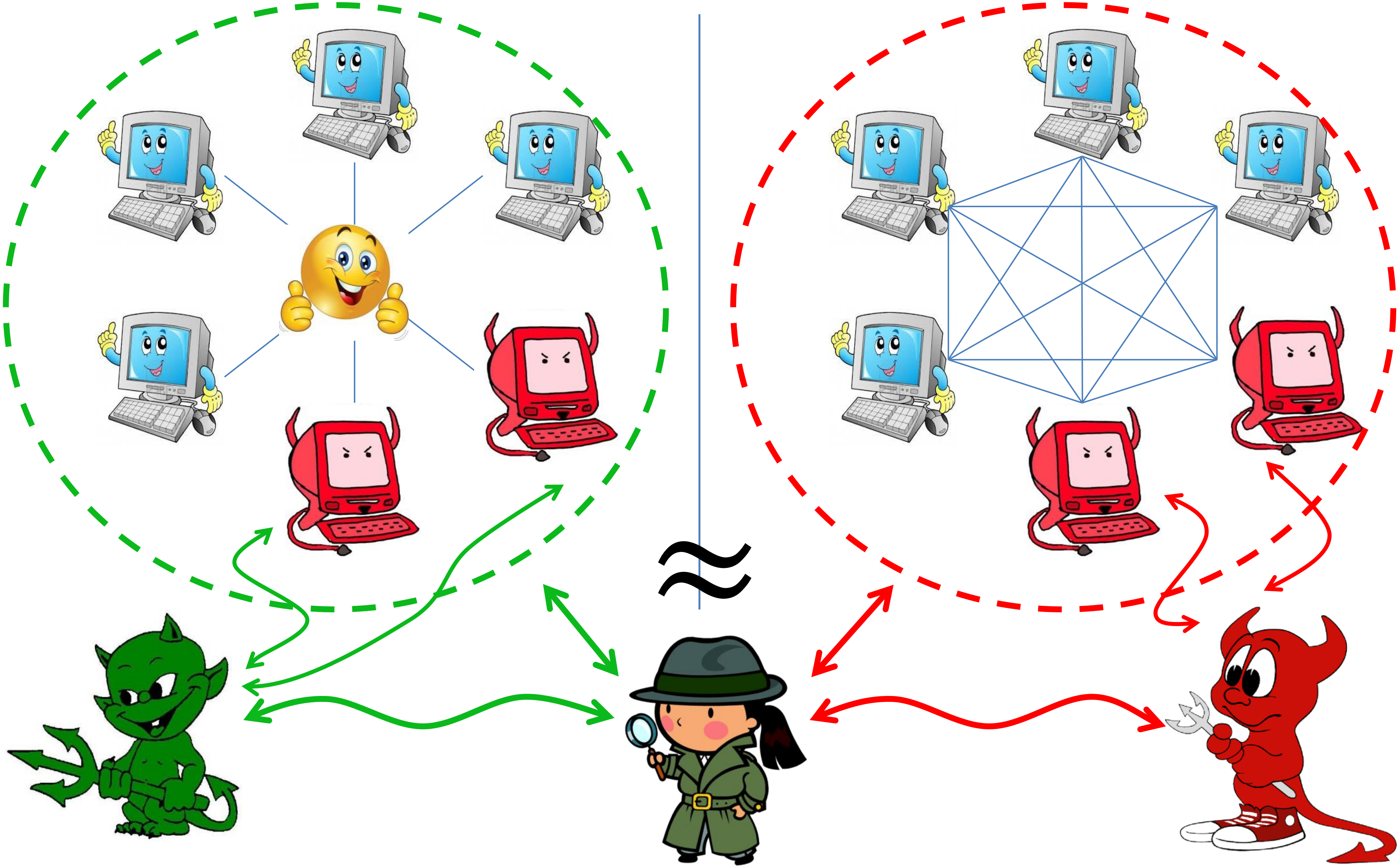
Secure Multiparty Computation



Ideal World



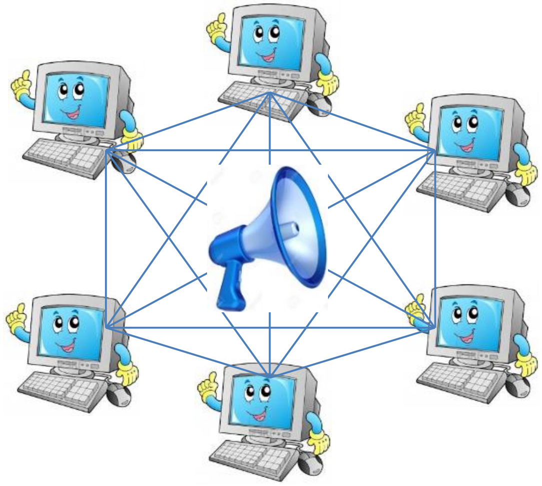
Real/Ideal Paradigm



Broadcast is Good for MPC

Every function f can be computed with guaranteed output delivery (honest majority)

- Round complexity depends only on f (unconditional)
- Constant-round protocols (OWF)
- Optimal three-round protocols (FHE)

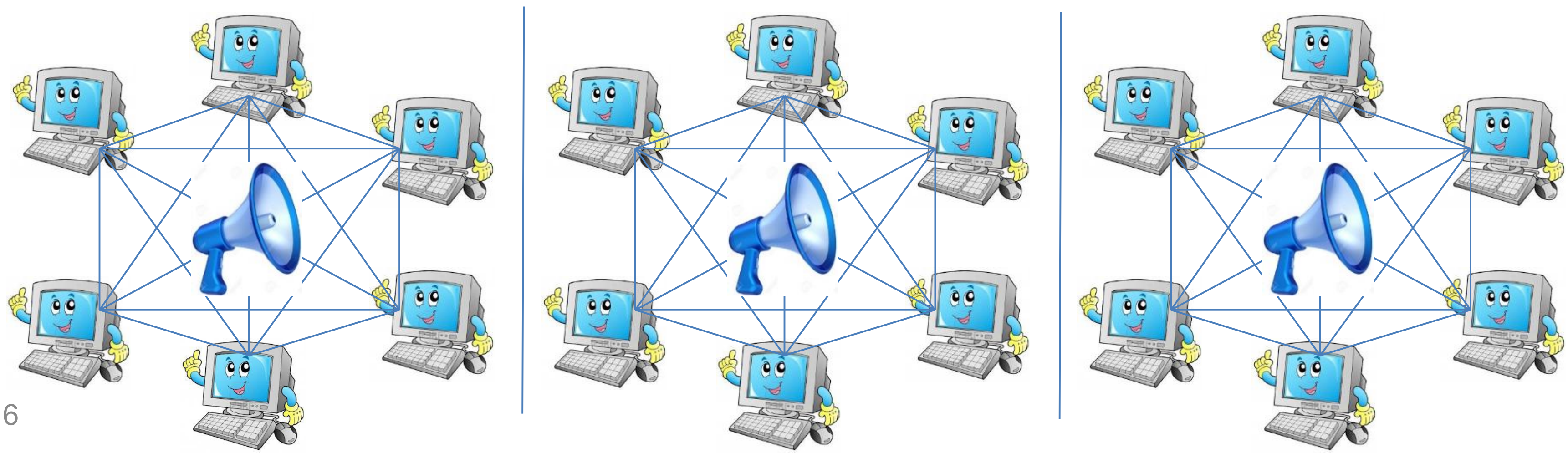


Broadcast is Very Good for MPC

Parallel composition preserves round complexity

If r-round π is secure under parallel composition

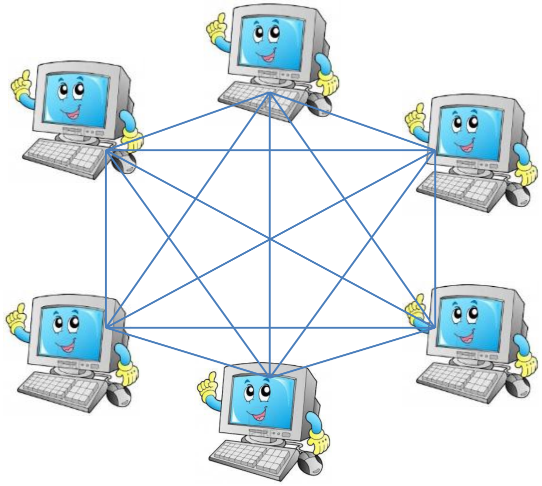
$\Rightarrow$ poly-many parallel executions of π in r rounds



What if Broadcast Doesn't Exist?

Use Broadcast Protocols

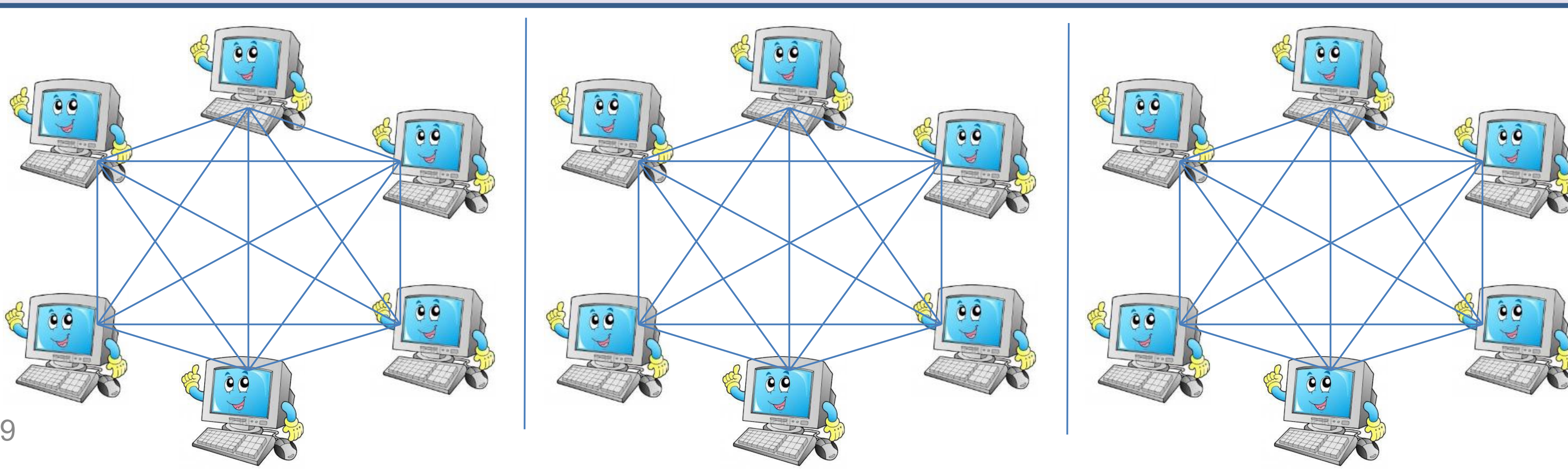
- Trusted setup required for broadcast $t \geq n/3$ (PKI/information-theoretic signatures)
- Some functions can be comp. without setup [**C**-Lindell'14, **C**-Haitner-Omri-Rotem'16]



Termination of Broadcast Protocols

- Protocols with simultaneous termination require $t + 1$ rounds [Fischer-Lynch'82, Dolev-Reischuk-Strong'90]
- Exp. constant round $\Rightarrow$ probabilistic termination [Feldman-Micali'88, Fitzi-Garay'03, Katz-Koo'06, Micali'17]
  - Termination round not a priori known
  - Non-simultaneous termination

Naïve parallel composition **not round preserving**



Naïve Parallel Composition

Protocol with *expected* $O(1)$ rounds (geometric dist.)

$\Rightarrow n$ parallel instances take $\Theta(\log n)$ rounds

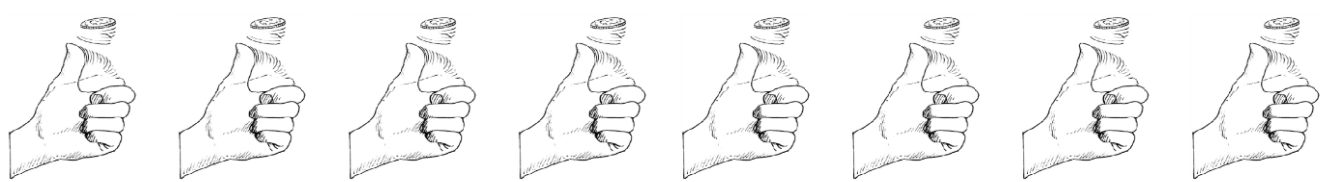
Example: Coin flipping

- Stand-alone coin flip: $\Pr(heads) = 1/2$

  $\Rightarrow$ output is $heads$ in expected 2 rounds

- Flipping in parallel n coins, each coin until $heads$

  $\Rightarrow$ expected $\log n$ rounds



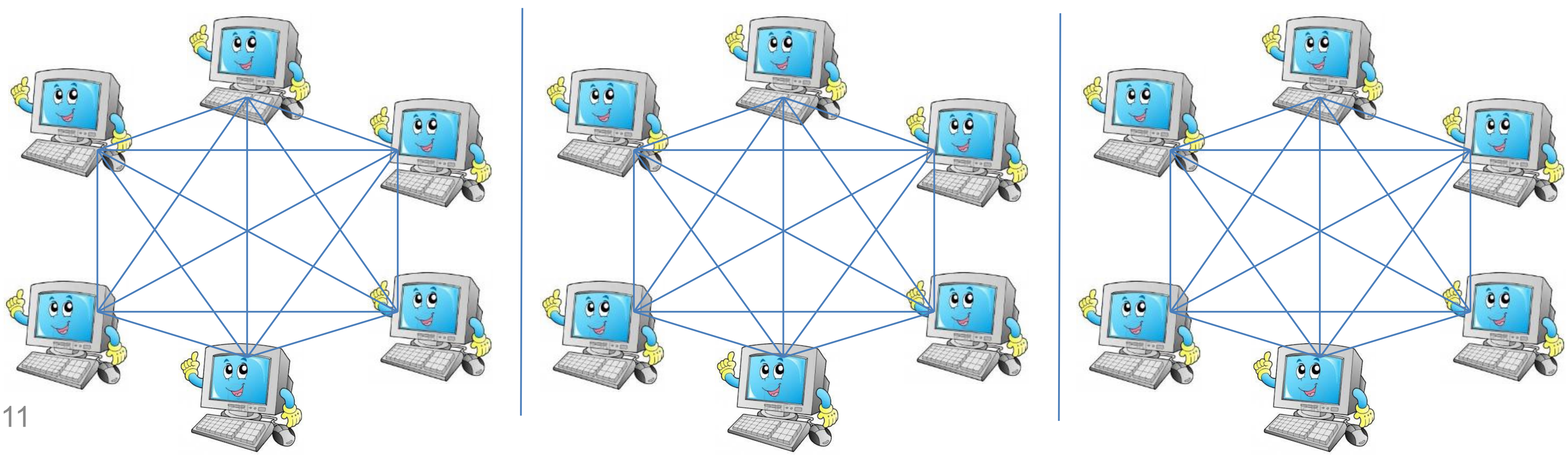
Parallel Composition of Broadcast

- Expected constant round parallel broadcast [BenOr-ElYaniv'03, Fitzi-Garay'03, Katz-Koo'06]
- Composable parallel bcast [**C**-Coretti-Garay-Zikas'16]

⇒ Recipe for MPC:

same exp. round complexity as in broadcast model

1) Construct protocol assuming broadcast channel
2) Instantiate bcast channel using PT parallel bcast



Parallel Composition of Broadcast

- Expected constant round parallel broadcast [BenOr-ElYaniv'03, Fitzi-Garay'03, Katz-Koo'06]
- Composable parallel bcast [**C**-Coretti-Garay-Zikas'16]

⇒ Recipe for MPC:

same exp. round complexity as in broadcast model

1) Construct protocol assuming broadcast channel
2) Instantiate bcast channel using PT parallel bcast

Problem:

Solutions for broadcast crucially rely on its privacy-free nature

The MPC protocol has probabilistic termination

(Naïve parallel composition not round preserving)

Main Question

Can parallel composition of arbitrary PT protocols be round-preserving?

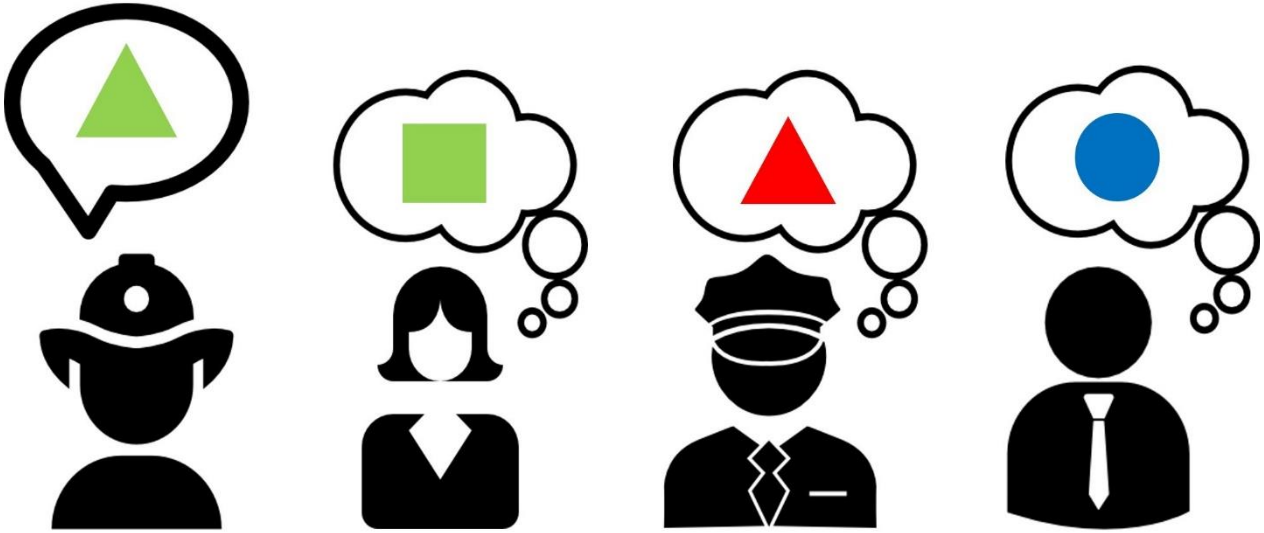
Main Question

Can parallel composition of arbitrary PT protocols be round-preserving? In a black-box way?

BB w.r.t. **functionality** [Rosulek'12, IKPSY'16]

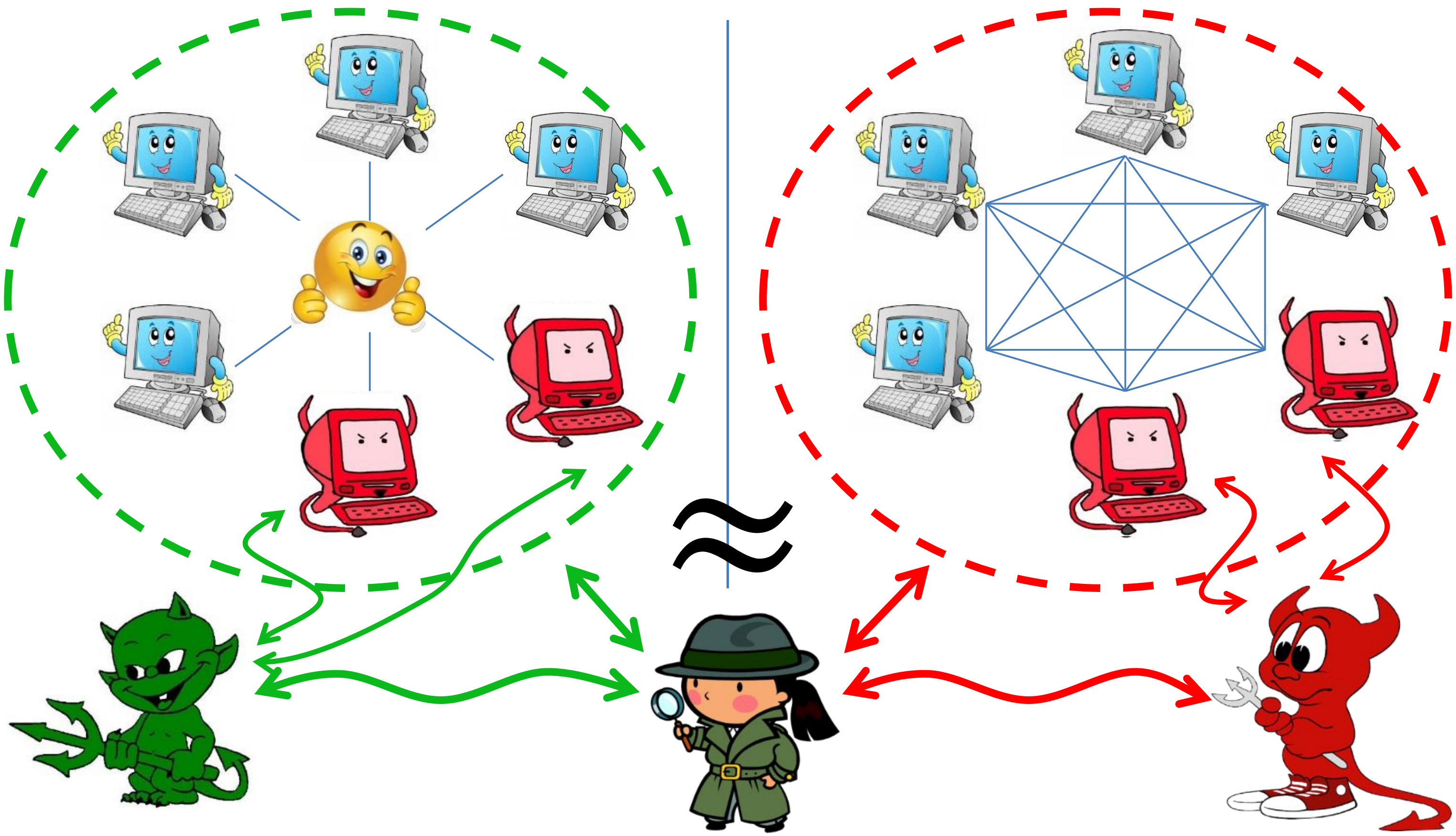
BB w.r.t. **protocol** (next-message function)

Common Terminology



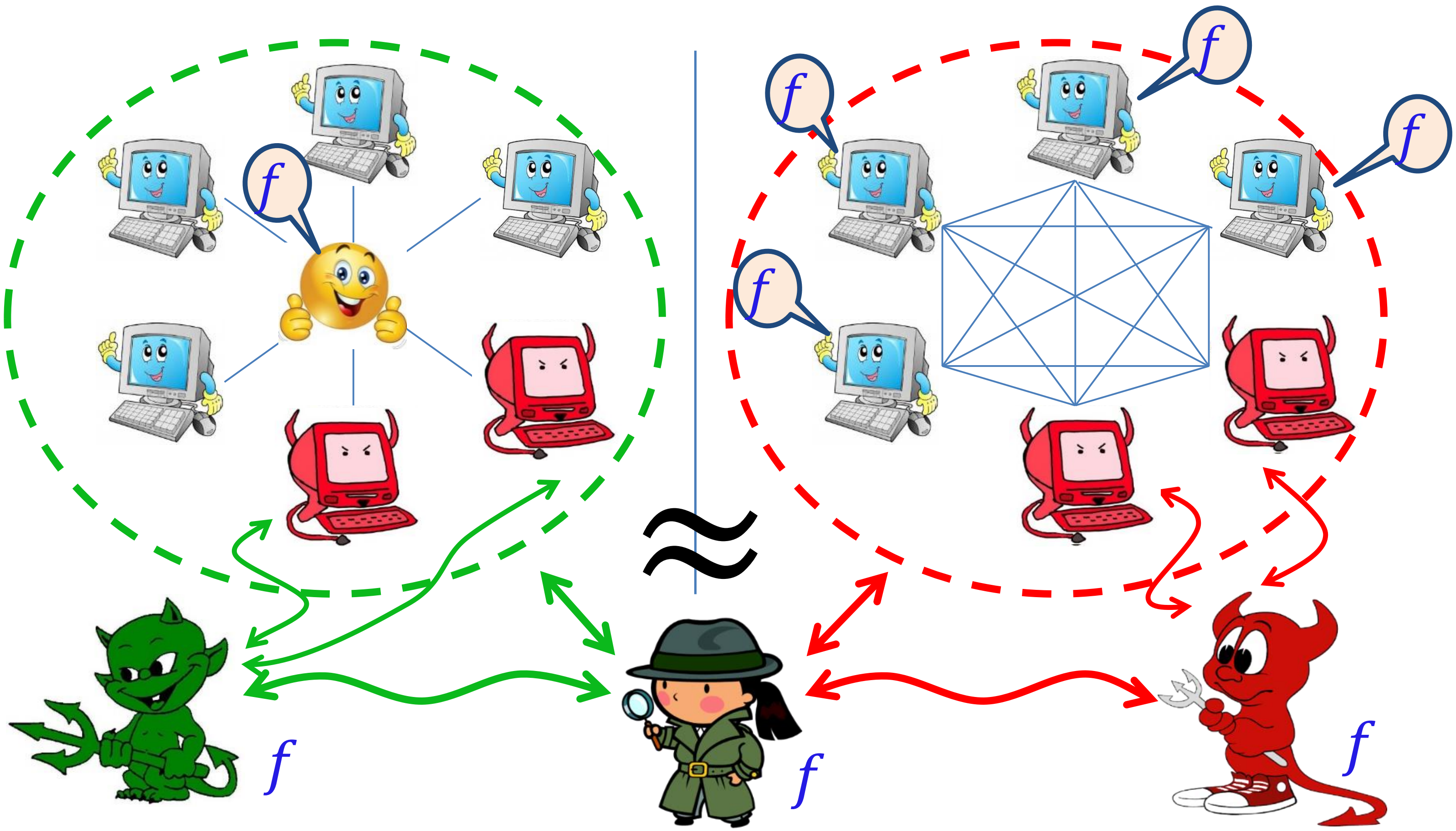
Synchronous MPC [KMTZ'13, CCGZ'16]

- Ideal world captures round complexity of π
- Trusted party samples $r_{term} \leftarrow D = D(\pi)$
- Parties continuously ask for output (receive by r_{term})
- $\mathcal{S}$ can instruct early delivery for specific parties



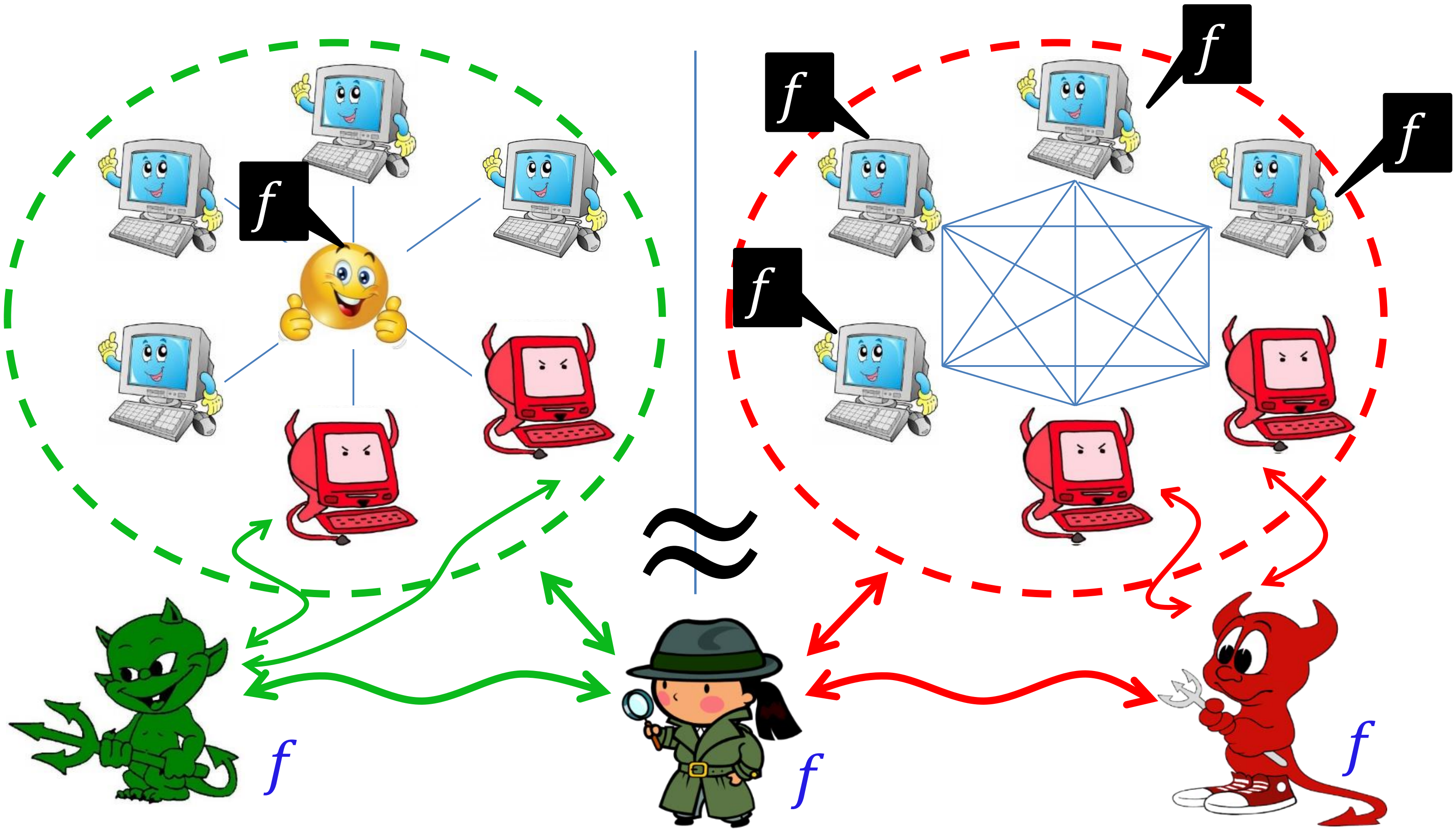
Functionally BB Protocols

- Traditional MPC: all parties know f



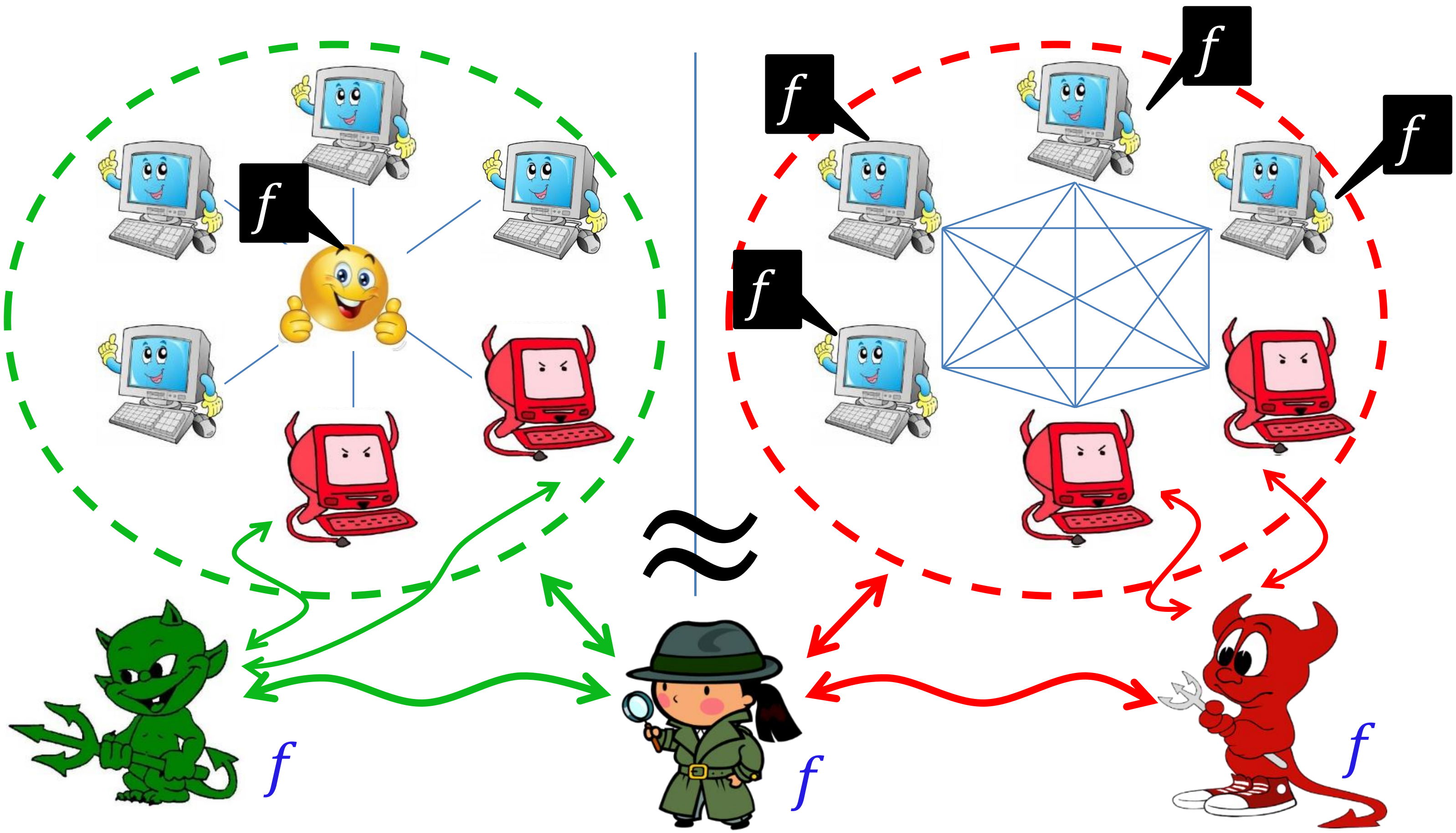
Functionally BB Protocols

- Traditional MPC: all parties know f
- FBB protocol is defined for function class $\mathcal{F} = \{f_1, \dots, f_N\}$
- Parties have oracle access to $f \in \mathcal{F}$ ($\mathcal{Z}, \mathcal{A}, \mathcal{S}$ know f)



Functionally BB Protocols

Protocol π is **FBB protocol** for $\mathcal{F}$
if $\forall f \in \mathcal{F}$ protocol π^f securely computes f



Impossibility of FBB Protocols

Theorem [Ishai-Kushilevitz-Prabhakaran-Sahai-Yu'16]:

$\exists 2$-party function class $\mathcal{F}$ such that **no** FBB protocol computes $\mathcal{F}$ facing semi-honest adversary

Proof intuition:

The function class $\mathcal{F} = \{f_\alpha\}_{\alpha \in \{0,1\}^\kappa}$ defined as

$$f_\alpha(x_1, x_2) = \begin{cases} 1, & x_1 \oplus x_2 = \alpha \\ 0, & x_1 \oplus x_2 \neq \alpha \end{cases}$$

Impossibility of FBB Protocols

- For random α, x_1, x_2 consider protocol π^{f_α}
- Following events occur with negl probability:
  - A party queries f_α with (p, q) s.t. $p \oplus q = \alpha$
  - A party queries f_α with (p, q) s.t. $p \oplus q = x_1 \oplus x_2$

  $\Rightarrow$ All oracle queries in π^{f_α} return 0
- Consider coupled experiment with $\alpha^* = x_1 \oplus x_2$
- For random coins such that events don't occur all oracle queries in $\pi^{f_{\alpha^*}}$ also return 0

$\Rightarrow$ both π^{f_α} and $\pi^{f_{\alpha^*}}$ output the same value

output 0 except negl

output 1 except negl

Parallel Composition of Functions

Given n-party functions $f_1, f_2, \dots, f_m$

denote by $f_1 \parallel f_2 \parallel \cdots \parallel f_m$ the following function:

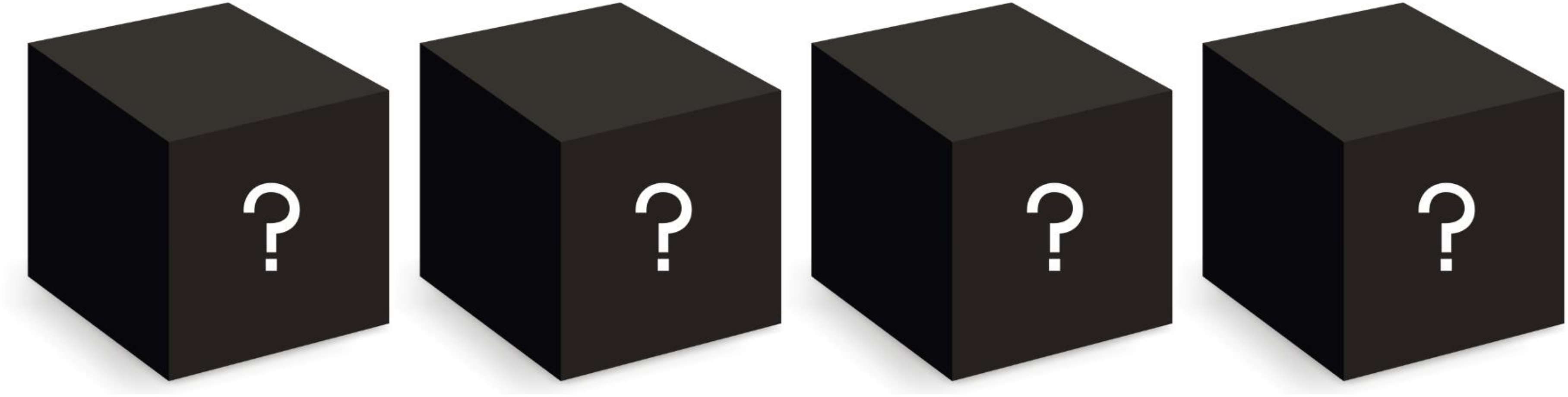
- Each P_i has input $\boldsymbol{x}_i = \left(x_i^1, x_i^2, \dots, x_i^m\right)$
- Output is $\boldsymbol{y} = (y_1, y_2, \dots, y_m)$

$$f_1\left(x_1^1, x_2^1, \dots, x_n^1\right)$$

$$f_2\left(x_1^2, x_2^2, \dots, x_n^2\right)$$

$$f_m(x_1^m, x_2^m, \dots, x_n^m)$$

FBB Parallel Composition

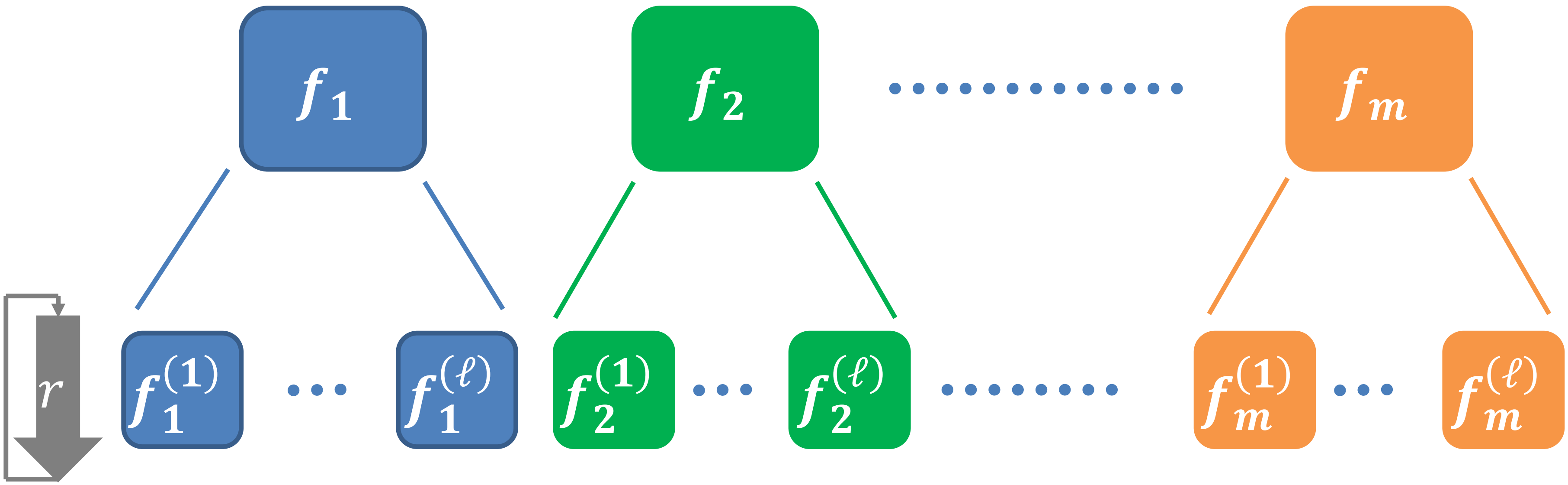


Semi-Honest FBB Protocol

Theorem 1:

- Let $\mathcal{F}_1, \dots, \mathcal{F}_m$ be deterministic function classes
- Consider $(\mathcal{F}_1, \dots, \mathcal{F}_m)$-hybrid model
  that $\forall j$ computes the function $f_j \in \mathcal{F}_j$
  with expected constant round complexity μ
- Then $\exists$ FBB protocol for $\mathcal{F}_1 \parallel \cdots \parallel \mathcal{F}_m$
  with expected constant round complexity

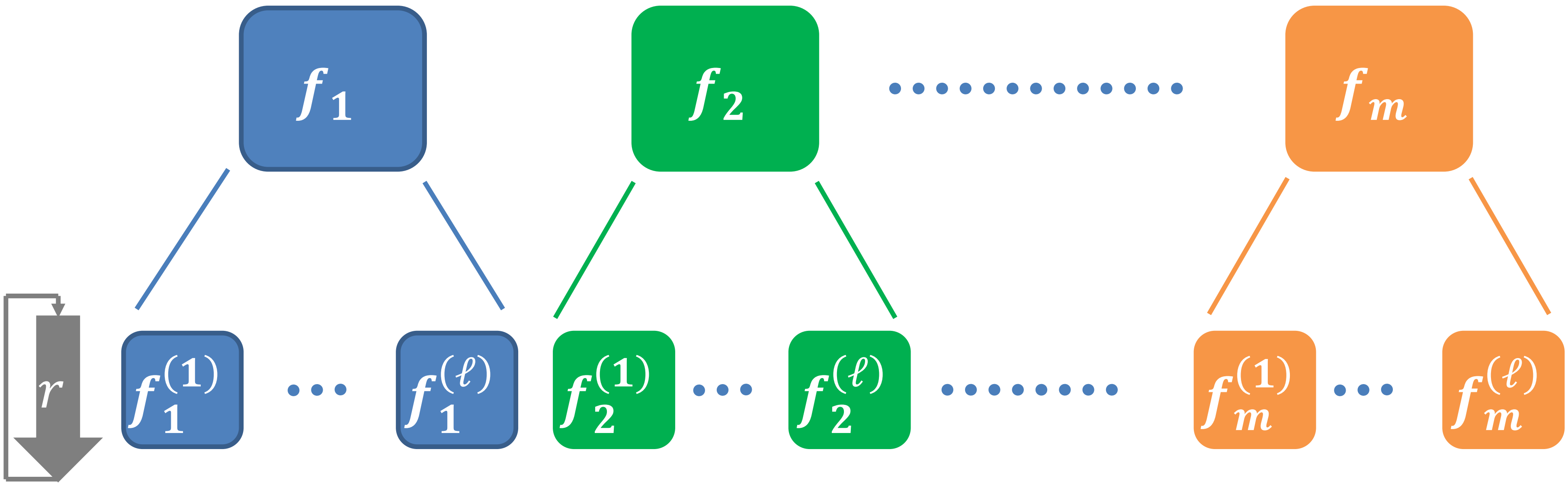
Semi-Honest FBB Protocol



1) Parties invoke ℓ instances of each f_j
2) Each P_i sends x_i^j to all instances of f_j and asks output for r rounds
3) If some P_i received output y_j for each f_j distribute $(y_1, \ldots, y_m)$ and halt, otherwise restart

parameters

Semi-Honest FBB Protocol



Proof intuition:

- ✓ Correctness
- ✓ Privacy: corrupt parties always use the same input values (semi-honest)
- ✓ Round complexity: for $\ell = \Omega(\log m)$ and constant $r > \mu$, the expected number of "restarts" is constant (Markov)

What About Malicious?

- The previous protocol is **not secure** for malicious
- The adversary can send different x_i^j and $\tilde{x}_i^j$ to f_j and learn multiple outputs
- This is inherent for **batched-parallel composition protocols**
  - All parties use original inputs $(x_1^k, \ldots, x_n^k)$ in two calls to the trusted party
  - Possibly in different rounds ρ and ρ'
  - Possibly for computing different f_j and $f_{j'}$

Malicious FBB Protocol

Theorem 2: Let $m = O(\kappa)$
$\exists n$-party function classes $\mathcal{F}_1, \ldots, \mathcal{F}_m$ s.t.
if π computes $\mathcal{F}_1 \parallel \cdots \parallel \mathcal{F}_m$ in $(\mathcal{F}_1, \ldots, \mathcal{F}_m)$-hybrid model (with exp. 2 rounds, geometric dist.)
then, facing a **single** malicious corrupted party:

- π must call each $\mathcal{F}_i$ at least once
- If π is naïve parallel composition
  $\Rightarrow$ not round preserving ($\log \kappa$)
- π is not batched-parallel composition protocol

until some get output

call each $\mathcal{F}_j$ until **all** parties get output

using same inputs in two calls

Proof Intuition

Define $\mathcal{F}_1 = \cdots = \mathcal{F}_m = \{f_\alpha\}_{\alpha \in \{0,1\}^\kappa}$ where

$$f_\alpha(x_1, x_2, \lambda, \ldots, \lambda) = \begin{cases} (x_2, x_1, \alpha, \ldots, \alpha), & x_1 \oplus x_2 = \alpha \\ (0^\kappa, 0^\kappa, \ldots, 0^\kappa), & x_1 \oplus x_2 \neq \alpha \end{cases}$$

- Naïve composition fails for geometric dist.
- No FBB protocol (without invoking trusted party) – extending [IKPSY'16]
- No batched-parallel protocol

See the paper for details

Protocol-BB Parallel Composition

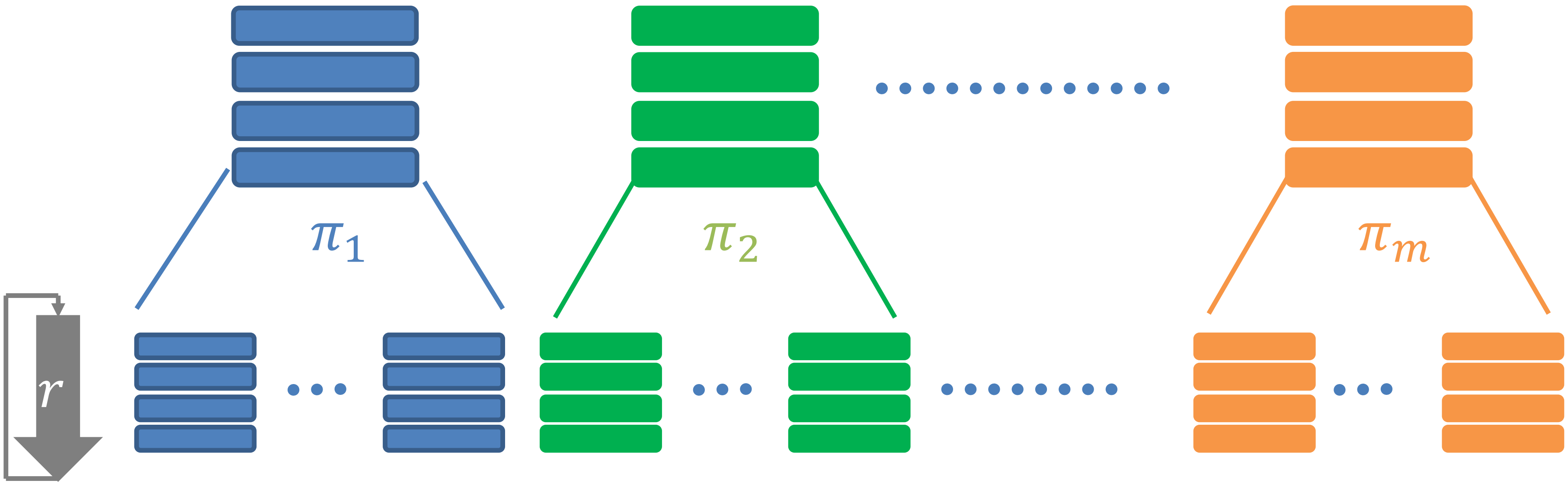
Protocol-BB Parallel Composition

Theorem 3:

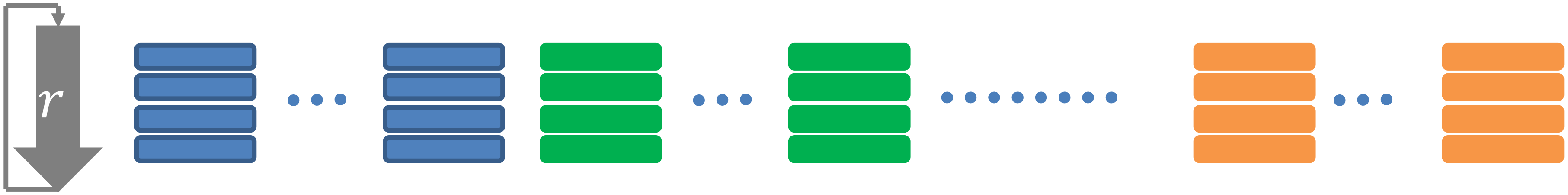
- Let PT protocols $\pi_1, \ldots, \pi_m$ realizing $f_1, \ldots, f_m$
- Then $\pi = \mathrm{compiler}(\pi_1, \ldots, \pi_m)$ realizes $f_1 \parallel \cdots \parallel f_m$
  - ➢ Round preserving $\mathbb{E}(\pi) = O\left(\max_i \mathbb{E}(\pi_i)\right)$
  - ➢ Black-box w.r.t. protocols $\pi_1, \ldots, \pi_m$

The compiler doesn't know the code of π_i
(oracle access to next-message function)

Protocol Compiler

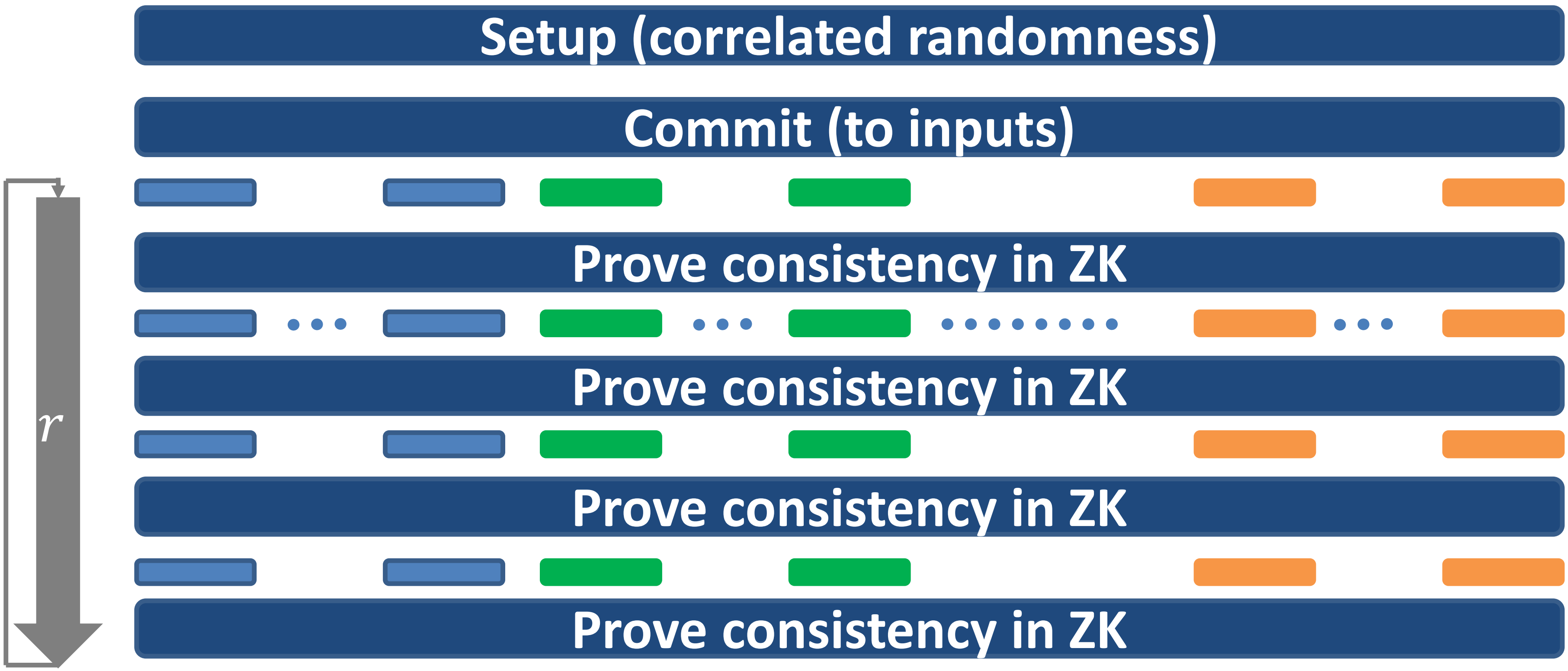


Prevent Multiple Inputs



Use **Setup, Commit, then Prove** functionality with a tweak [Canetti-Lindell-Ostrovsky-Sahai'02] [Ishai-Ostrovsky-Zikas'14]

Prevent Multiple Inputs



Use **Setup, Commit, then Prove** functionality with a tweak [Canetti-Lindell-Ostrovsky-Sahai'02] [Ishai-Ostrovsky-Zikas'14]

Some Challenges

- 1-to-many ZK black-box in $\pi_1, \ldots, \pi_m$ (based [IKOS'07])
  Adjust [IOZ'14] to security without abort ($t < n/2$)
- Recover from invalid ZK proofs without:
  1) Breaching privacy ($\mathcal{A}$ might have learned output)
  2) Blowing up round complexity
- Implement the Setup in constant rounds
  (use only correlated randomness for broadcast)
- Reactive functionalities with probabilistic termination

See the paper for details

Summary

We study parallel composition of PT protocols

Functionally black-box (FBB) protocols

- No round-preserving FBB parallel composition (using known techniques)
- Round-preserving FBB parallel composition with semi-honest security

Black-box w.r.t. protocols

- Round-preserving compiler for parallel composition

Thank You