Probabilistic Termination and Composability of Cryptographic Protocols

[Crypto '16]

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

Motivation

Given: Protocol with *expected* $O(1)$ running time
(geometric distribution)

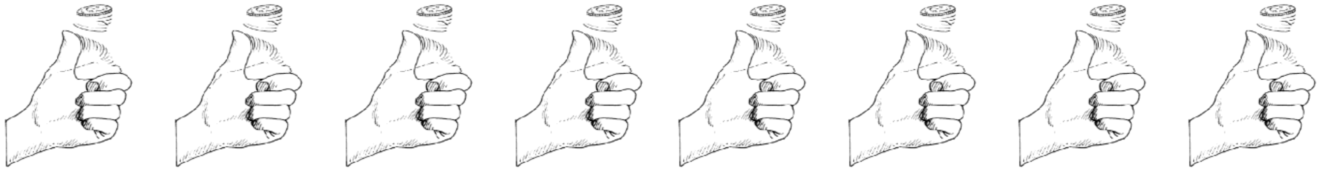
What's the expected running time of n parallel instances?

$\Theta(\log n)$ rounds

<u>Example:</u> Coin flipping

- Stand-alone coin flip: $\Pr(heads) = 1/2$
  Output is $heads$ in expected 2 rounds

- Flipping in parallel n coins, each coin until $heads$
  Expected $\log n$ rounds



Motivation (2)

- Most secure protocols assume a broadcast channel
- Fast broadcast protocols run in expected $O(1)$ time
  - Parallel executions no longer constant
  - Probabilistic termination round
  - Non-simultaneous termination
- **Composable security**: we want security of broadcast to hold in arbitrary protocols/networks/environments
  - Not guaranteed by known solutions
- How to simulate probabilistic termination?

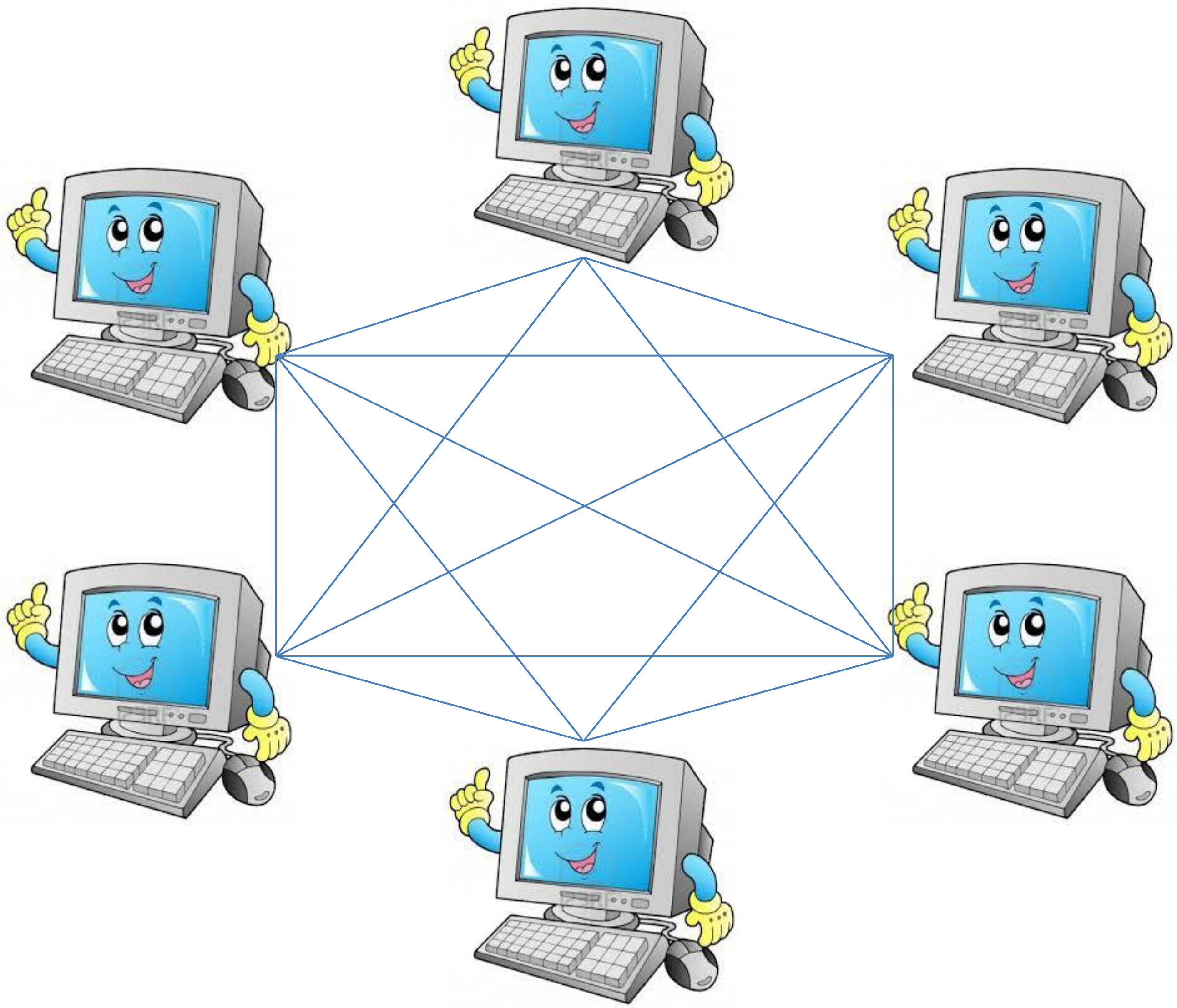
This Work

We study universal composability of cryptographic protocols with *probabilistic termination*

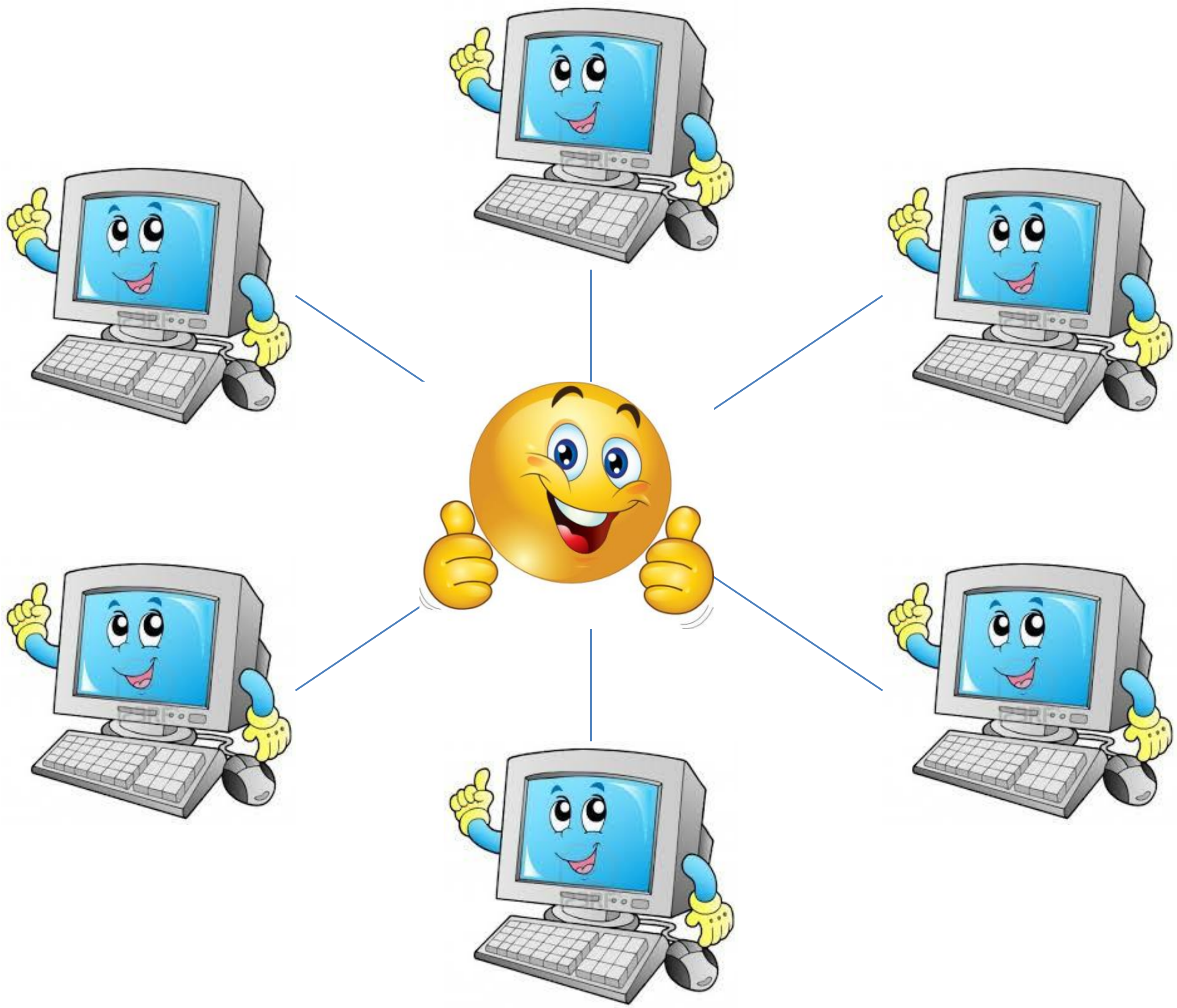
- Framework
  - ➢ Design and analyze simple protocols in modular composition fashion
  - ➢ Compiler to UC protocols with same expected round complexity
- Applications
  Perfect, adaptively secure protocols in the P2P model
  1) Byzantine agreement with expected $O(1)$ rounds
  2) Parallel broadcast with expected $O(1)$ rounds
  3) SFE with expected $O(d)$ rounds

d= depth of the circuit

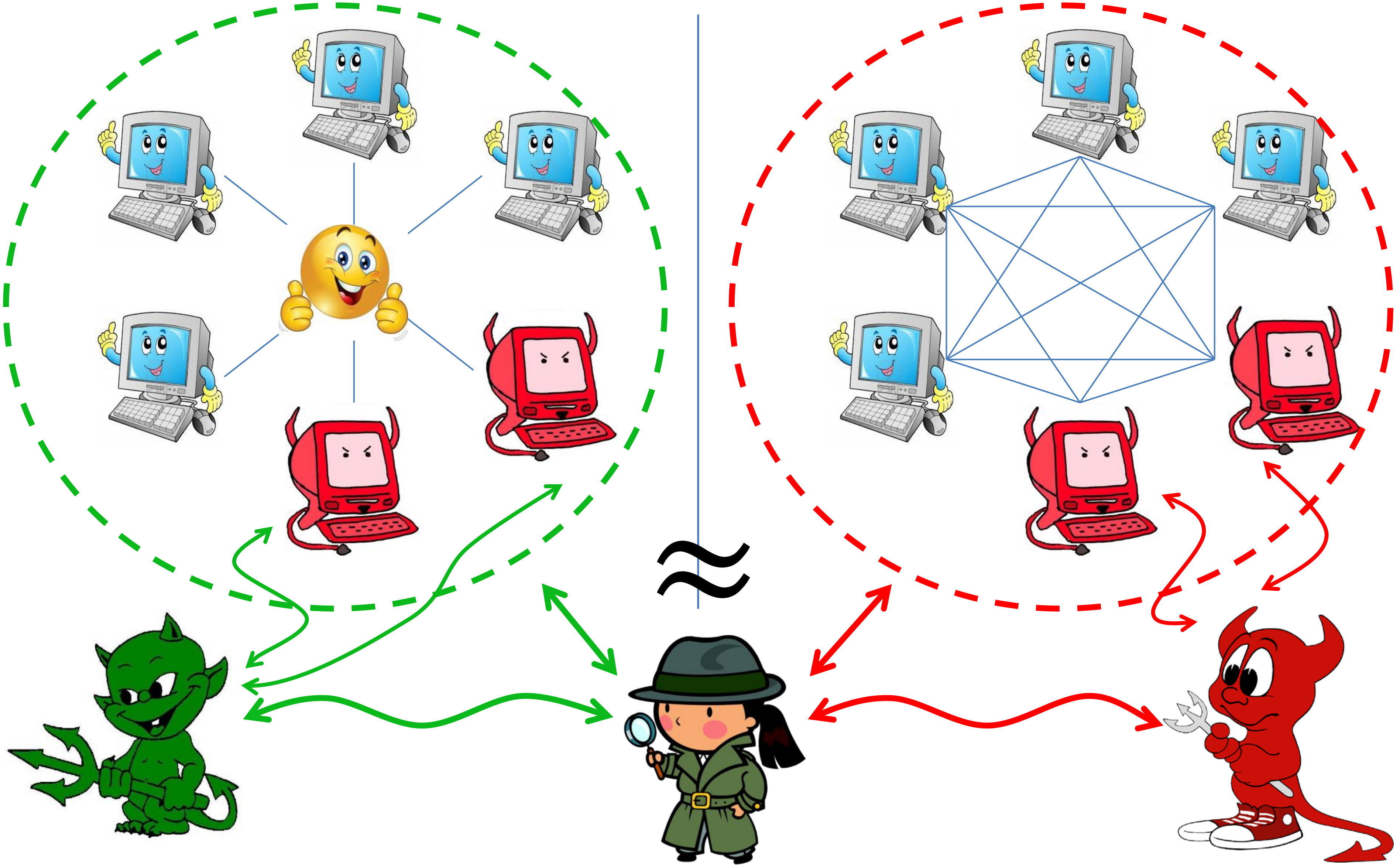
Secure Multiparty Computation (MPC)



Ideal World/“Functionality”

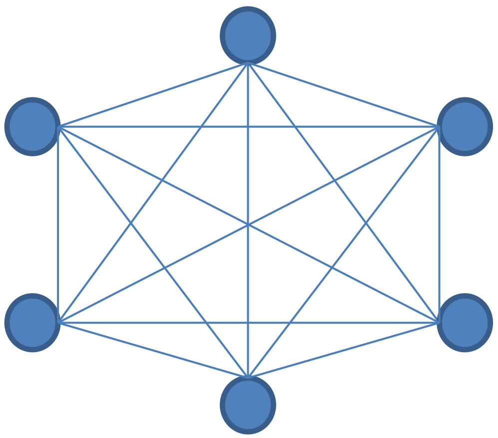


Simulation-based Security



Communication Models

- Point-to-point model
  - Secure (private) channels between the parties (*Secure Message Transmission*)



- Broadcast model
  - Additional *broadcast channel*

- Synchronous communication
  - Protocol proceeds in rounds
    - Bounded delay
    - Global clock

Feasibility of MPC with Broadcast

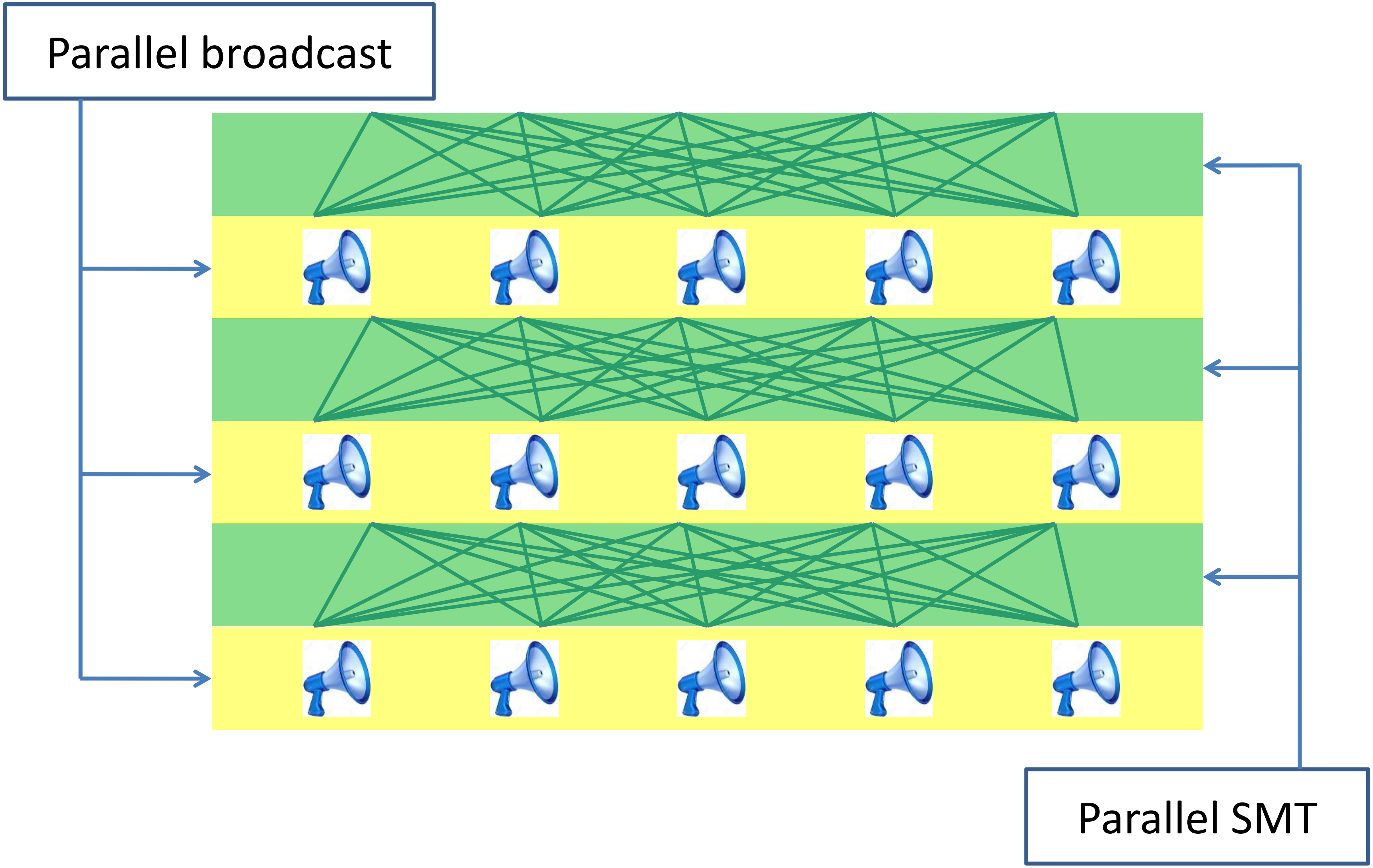
- Classical results [BGW'88] [CCD'88]
  - Perfect, adaptively secure for $t < n/3$
  - Concurrently composable
  - $O(d)$ rounds, $O(d)$ broadcasts

d= depth of the circuit

- Improving broadcast-round complexity
  - $O(d)$ rounds, 1 broadcast [Katz, Koo'07]

Can we get same security and efficiency in the point-to-point model (without broadcast)?

Protocols with Broadcast



Instantiating Broadcast Channel

Byzantine agreement (BA)

Each P_i has input x_i

- **Agreement**: all honest parties output the same value
- **Validity**: if all honest parties have the same input x, the common output is x

BA to broadcast (honest majority)

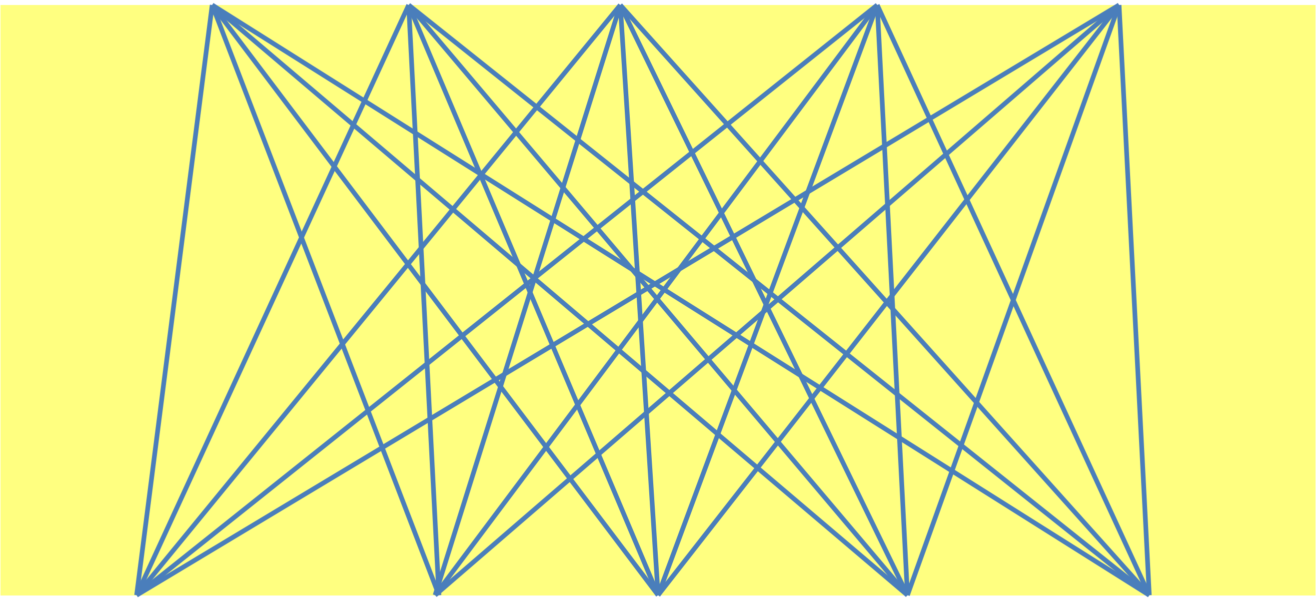
- The sender sends x to all parties
- All parties run BA on these values

Deterministic BA/Broadcast Protocols

- Deterministic Termination (DT) – single & known output round
- Perfect and adaptive security for $t < n/3$
  [BGP'89] [GM'93] [HZ'10]
- Concurrently composable
- Require $O(n)$ rounds – this is inherent [Fischer, Lynch'82]

Deterministic BA/Broadcast Protocols

- Deterministic Termination (DT) – single & known output round
- Perfect and adaptive security for $t < n/3$ [BGP'89] [GM'93] [HZ'10]
- Concurrently composable
- Require $O(n)$ rounds – this is inherent [Fischer, Lynch'82]

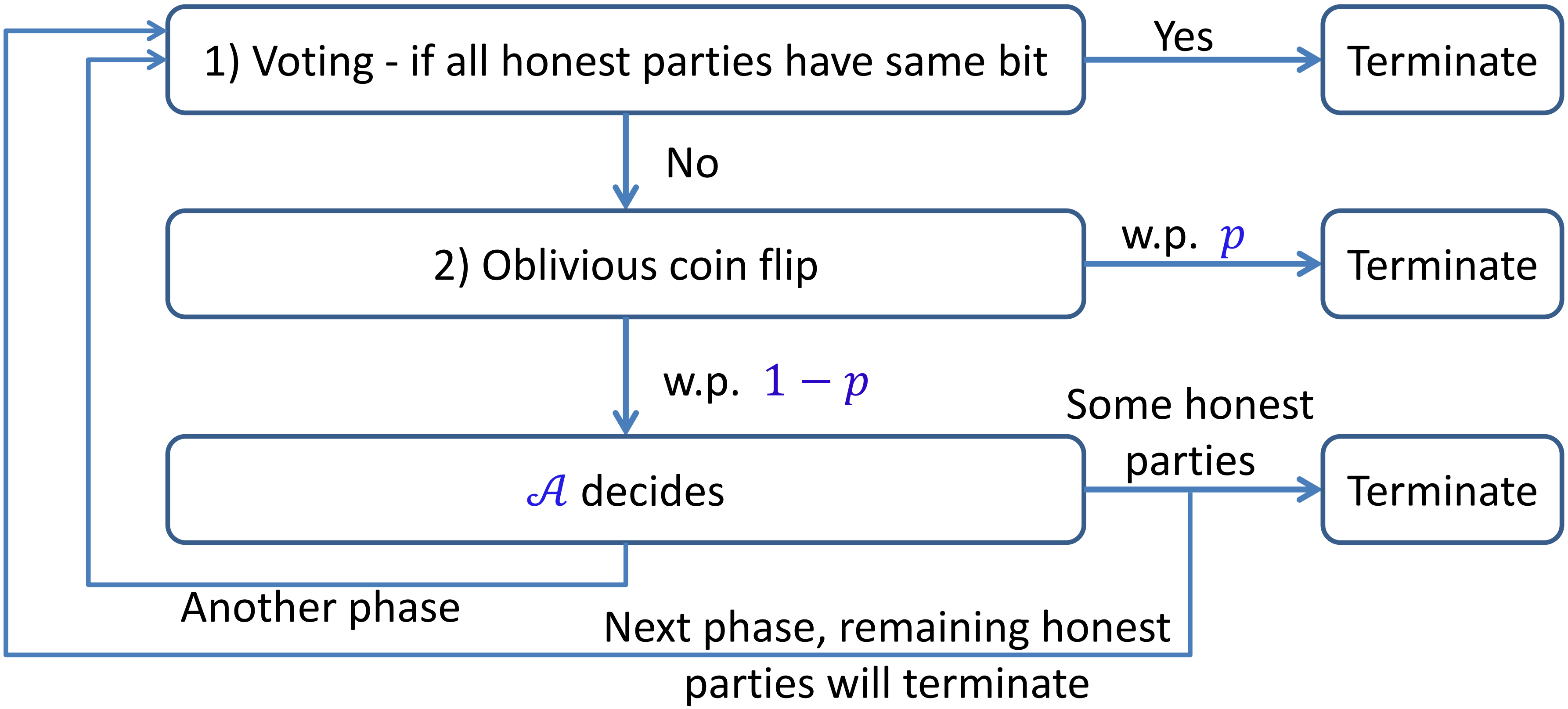


Randomized BA/Broadcast Protocols

Randomization can help [Ben-Or'83] [Rabin'83]

Binary BA protocol [Feldman, Micali'88] (simplified: [FG'03] [KK'06])

- Proceeds in phases until termination
- In each phase each party has a bit (initially its input)



Randomized BA/Broadcast Protocols (2)

- [FM'88] has *Probabilistic Termination* (PT):
  - Termination round not a priori known
  - No simultaneous termination:
    honest parties might terminate at different rounds
    This is inherent [Dolev, Reischuk, Strong'90]
  - Expected $O(1)$ rounds
  - All honest parties terminate within c rounds (constant)
- Extends to multi-valued BA [Turpin, Coan'84]
  - Two additional rounds
- Perfect security [Goldreich, Petrank'90]
  - Best of both worlds
- Variant for parallel broadcast [Ben-Or, El-Yaniv'03]

What’s Missing?

- All PT broadcast protocols are proven secure using a game-based definition (no composition guarantees)
- Composition follows from simulation-based proofs
- [KMTZ’13] defined a UC-based framework for synchronous DT protocols
  - Subtleties of PT protocols are not captured by [KMTZ’13]

We introduce a framework for designing and analyzing synchronous PT protocols

Rest of the Talk

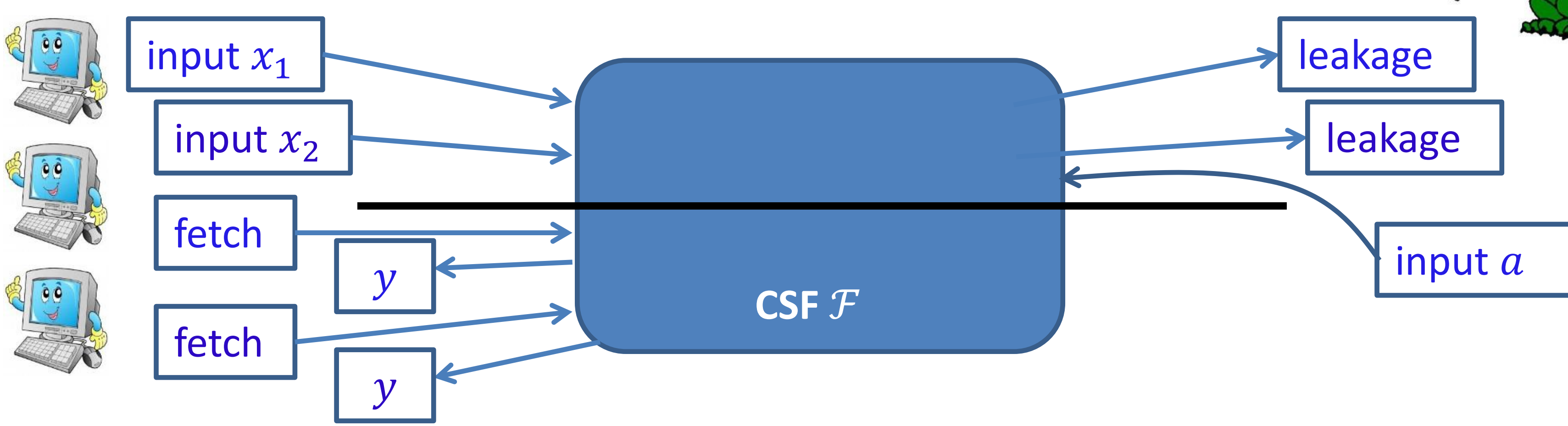
1. The Framework, Part I: Probabilistic Termination
   - Define PT functionalities
   - Construct PT protocols when parties start at same time
2. The Framework, Part II: Non-Simultaneous Start
   - Composition theorem
   - Construct PT protocols without simultaneous start
3. Applications

The Framework

Part I: Probabilistic Termination

Canonical Synchronous Functionality (CSF)

- Separate the function from the round structure
- A CSF consists of input round and output round
- Parametrized by
  - (Randomized) function $f(x_1, \dots, x_n, a)$
  - Leakage function $l(x_1, \dots, x_n)$



CSF Examples

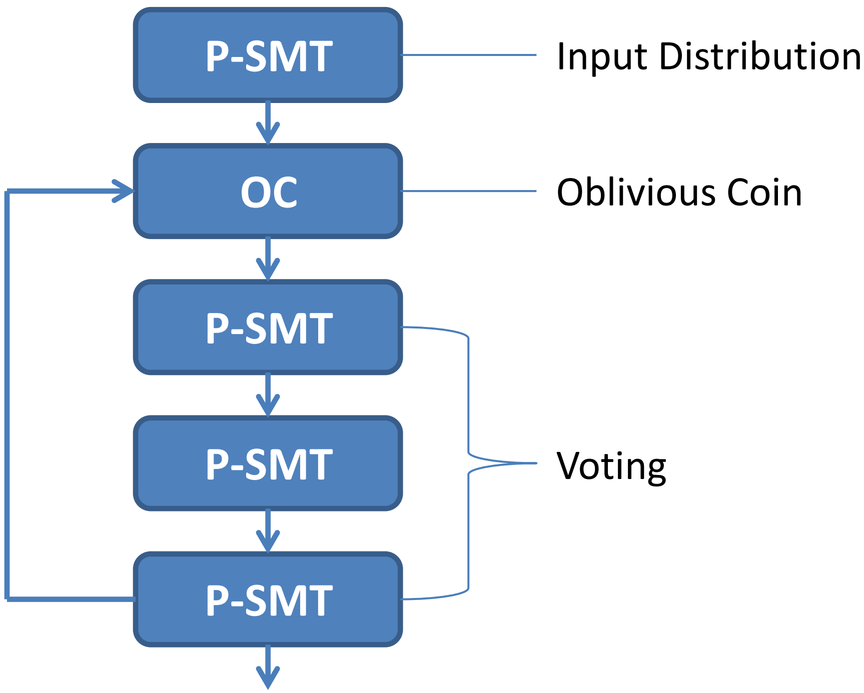
- **SMT**: P_i sends x_i to P_j
  - $f(x_1, \dots, x_n, a) = (y_1, \dots, y_n)$, s.t. $y_j = x_i$ and $y_k = \lambda$ $(k \neq j)$
  - $l(x_1, \dots, x_n) = \begin{cases} |x_i| \text{ if } P_j \text{ honest} \\ x_i \quad \text{if } P_j \text{ corrupted} \end{cases}$
- **Broadcast**: P_i broadcasts x_i
  - $f(x_1, \dots, x_n, a) = (x_i, \dots, x_i)$
  - $l(x_1, \dots, x_n) = |x_i|$

Parallel versions

- **SFE**: parties compute a function g
  - $f(x_1, \dots, x_n, a) = g(x_1, \dots, x_n)$
  - $l(x_1, \dots, x_n) = (|x_1|, \dots, |x_n|)$
- **BA**:
  - $f(x_1, \dots, x_n, a) = \begin{cases} y \text{ if at least } n - t \text{ inputs are } y \\ a \text{ otherwise} \end{cases}$
  - $l(x_1, \dots, x_n) = (x_1, \dots, x_n)$

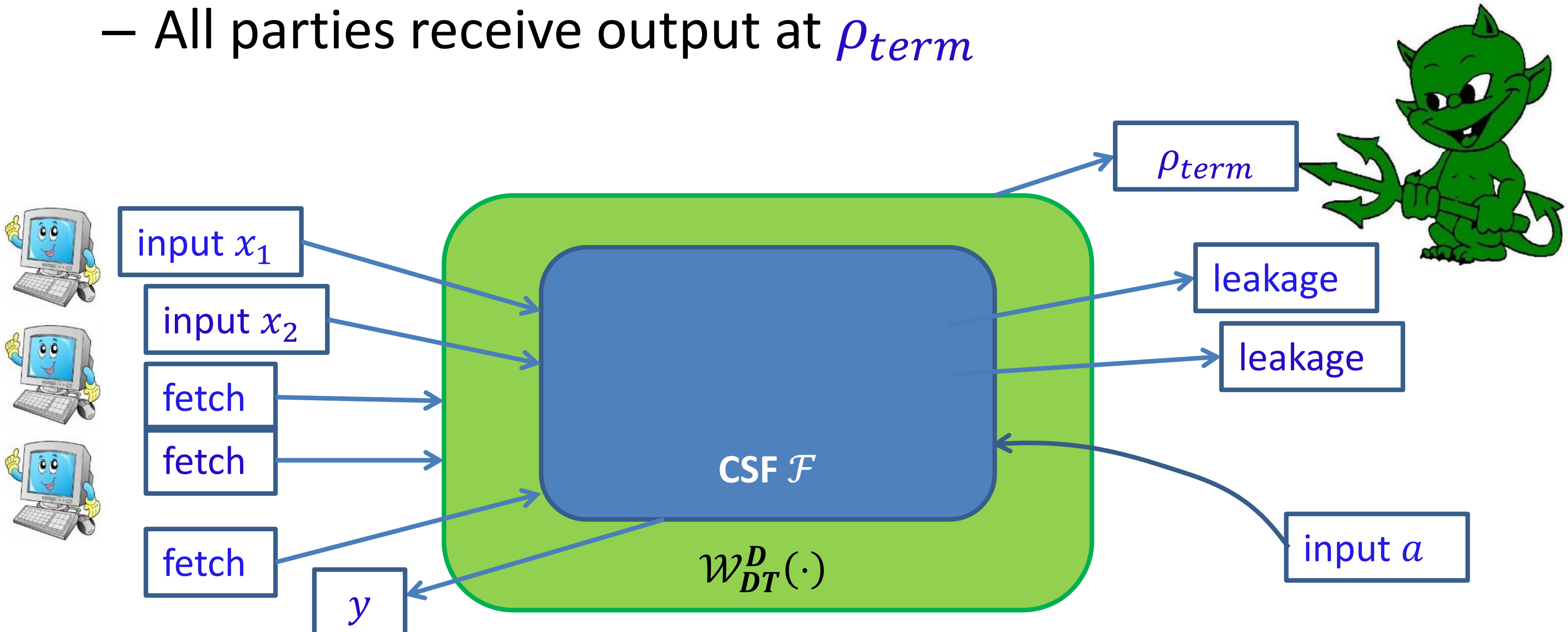
Synchronous Normal Form (SNF)

- SNF protocol:
  - All parties are synchronized throughout the protocol
  - All hybrids are (2-round) CSFs
  - In each round exactly one ideal functionality is called (as in [Canetti'00])
- Example: Protocol π_{RBA} (based on [FM'87])



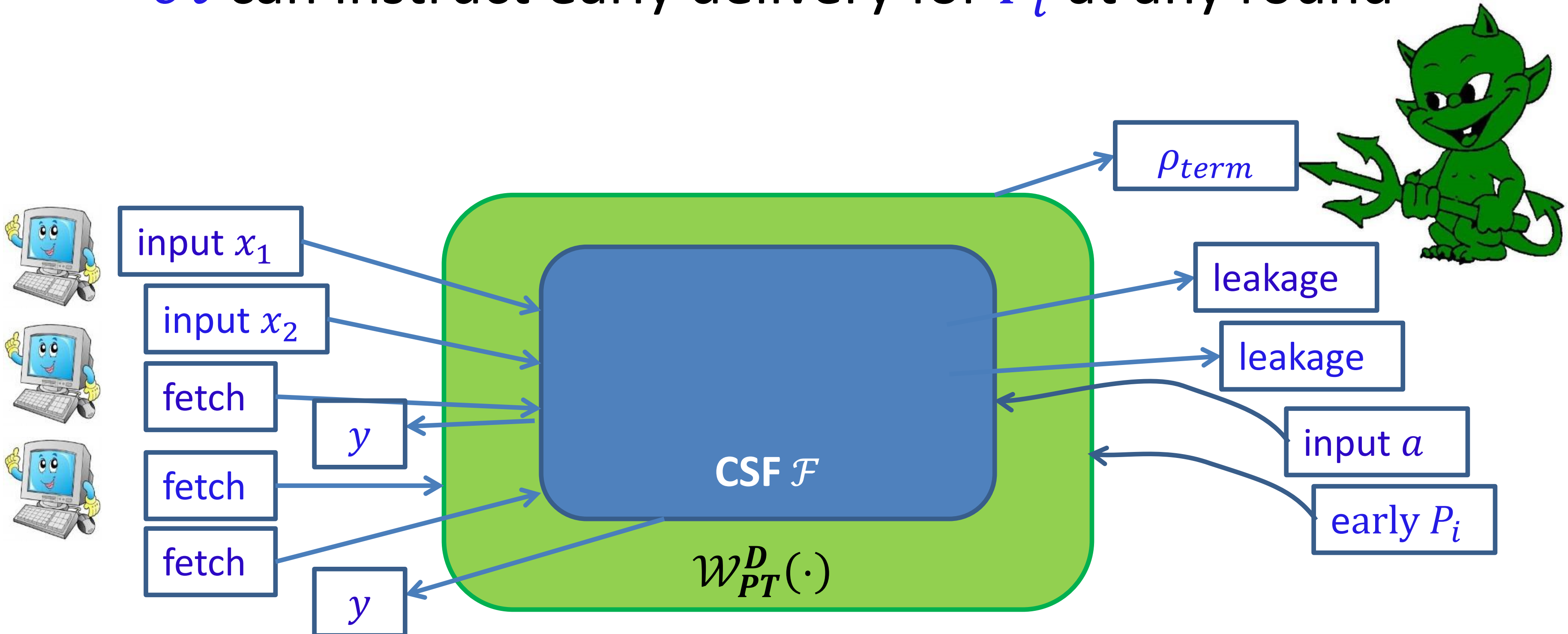
Extending Rounds (DT)

- Most functionalities cannot be implemented by two-round protocols
- Wrap the CSFs with *round-extension* wrappers
  - Sample a termination round $\rho_{term} \leftarrow D$
  - All parties receive output at ρ_{term}



Extending Rounds (PT)

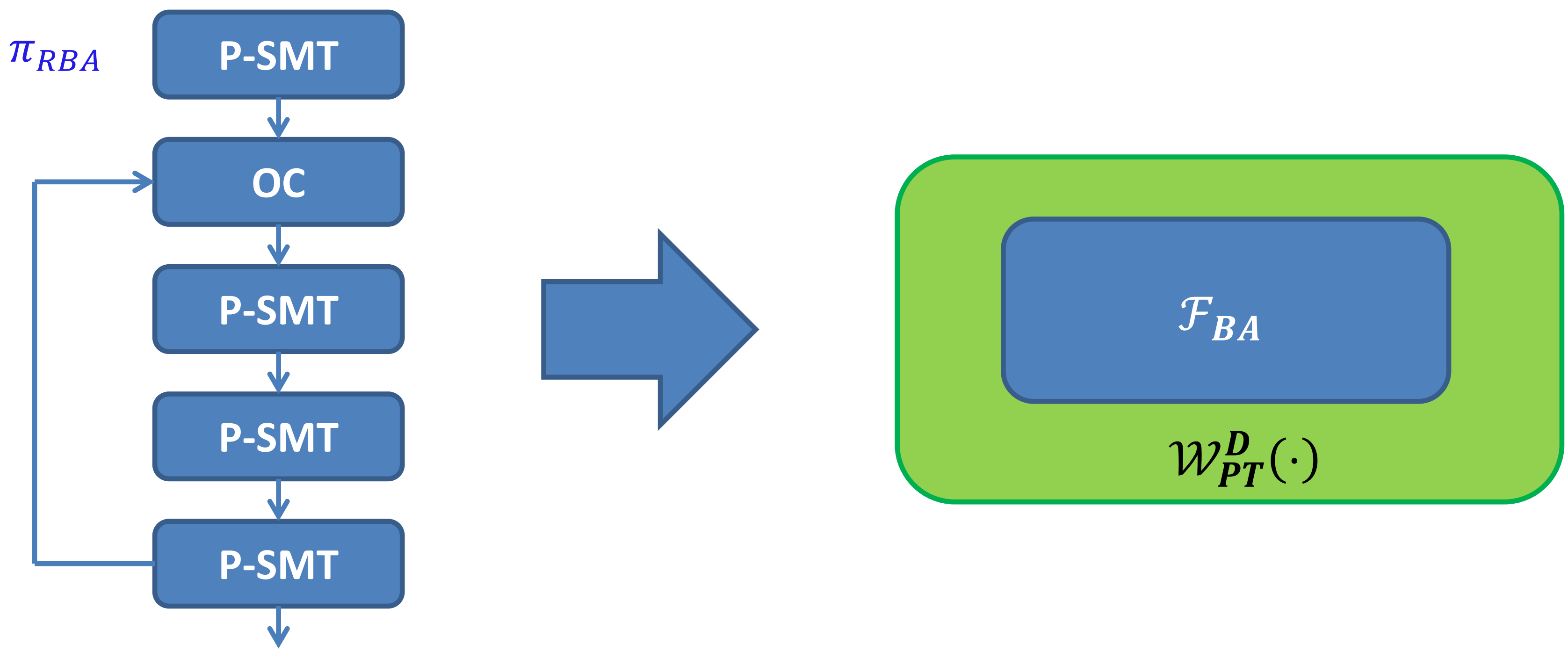
- PT: ρ_{term} is an upper bound
  - Sample a termination round $\rho_{term} \leftarrow D$
  - All parties receive output *by* ρ_{term}
  - $\mathcal{A}$ can instruct early delivery for P_i at any round



Where Do We Stand?

Thm: Protocol π_{RBA} implements $\mathcal{W}_{PT}^{D}(\mathcal{F}_{BA})$ in the $(\mathcal{F}_{PSMT}, \mathcal{F}_{OC})$-hybrid model, for $t < n/3$,

assuming all parties start at the same round



The Framework

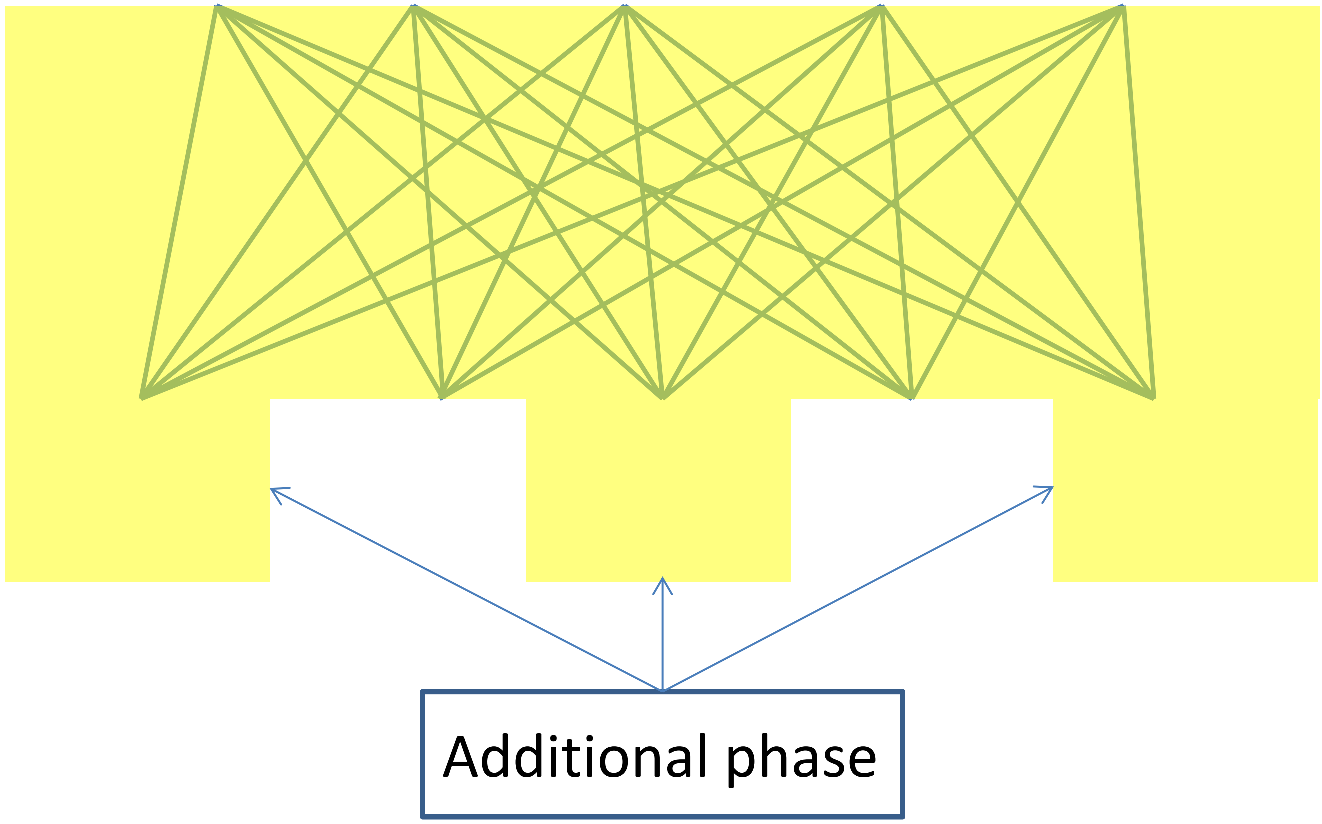
Part II: Non-Simultaneous Start

Sequential Composition

Fast parties start new execution **before** slow parties finished previous execution

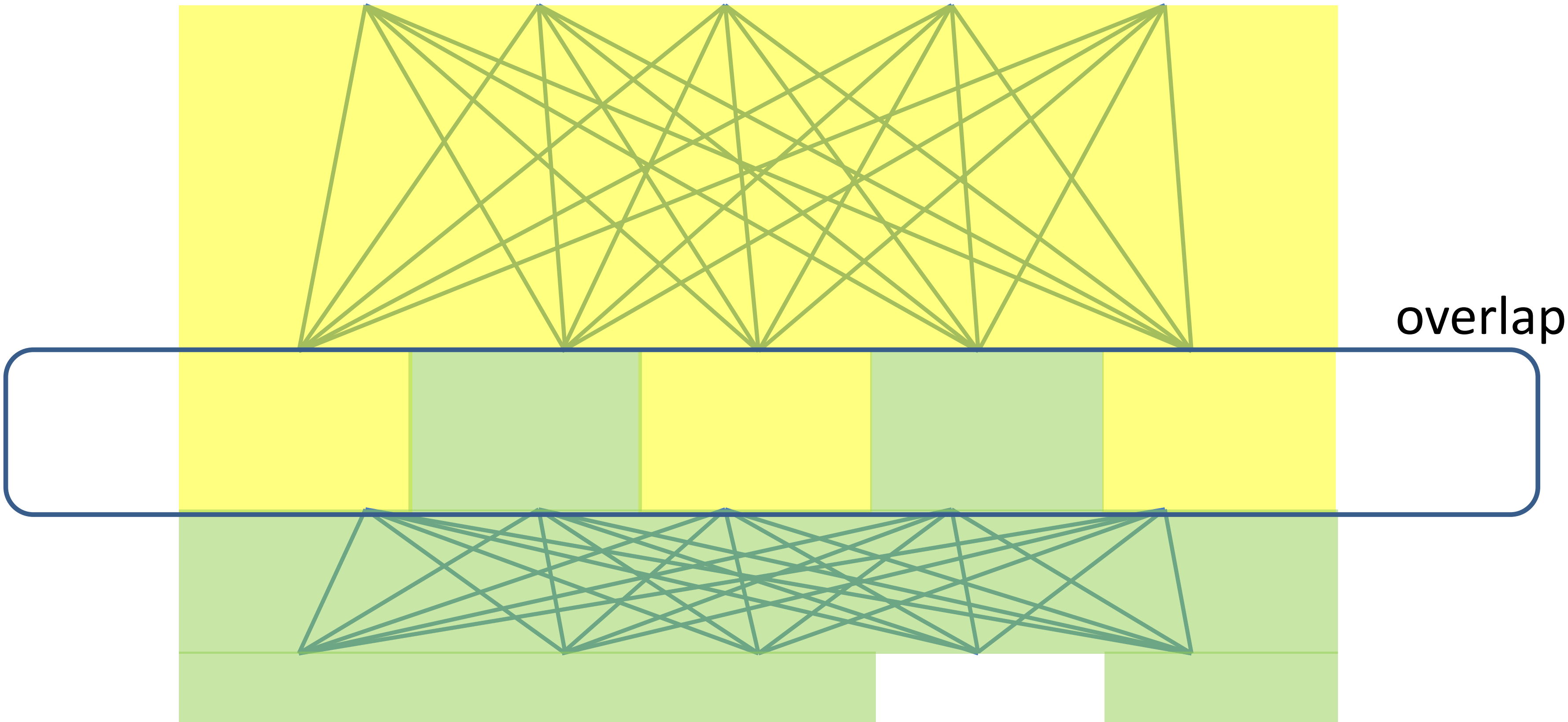
Sequential Composition

Fast parties start new execution **before** slow parties finished previous execution



Sequential Composition

Fast parties start new execution **before** slow parties finished previous execution



Sequential Composition (2)

Goal: ℓ sequential executions of expected $O(1)$ rounds protocols in expected $O(\ell)$ rounds

- Previous solutions [LLR'02] [BE'03] [KK'06]
  - Specific to broadcast
  - Game-based proofs (no composable security)
- We introduce generic compiler for SNF protocols
  - Non-simultaneous start
    - "slack" parameter $c \geq 0$: parties start within $c + 1$ rounds
  - Parties not synchronized (concurrent calls to hybrids)
  - Same round complexity

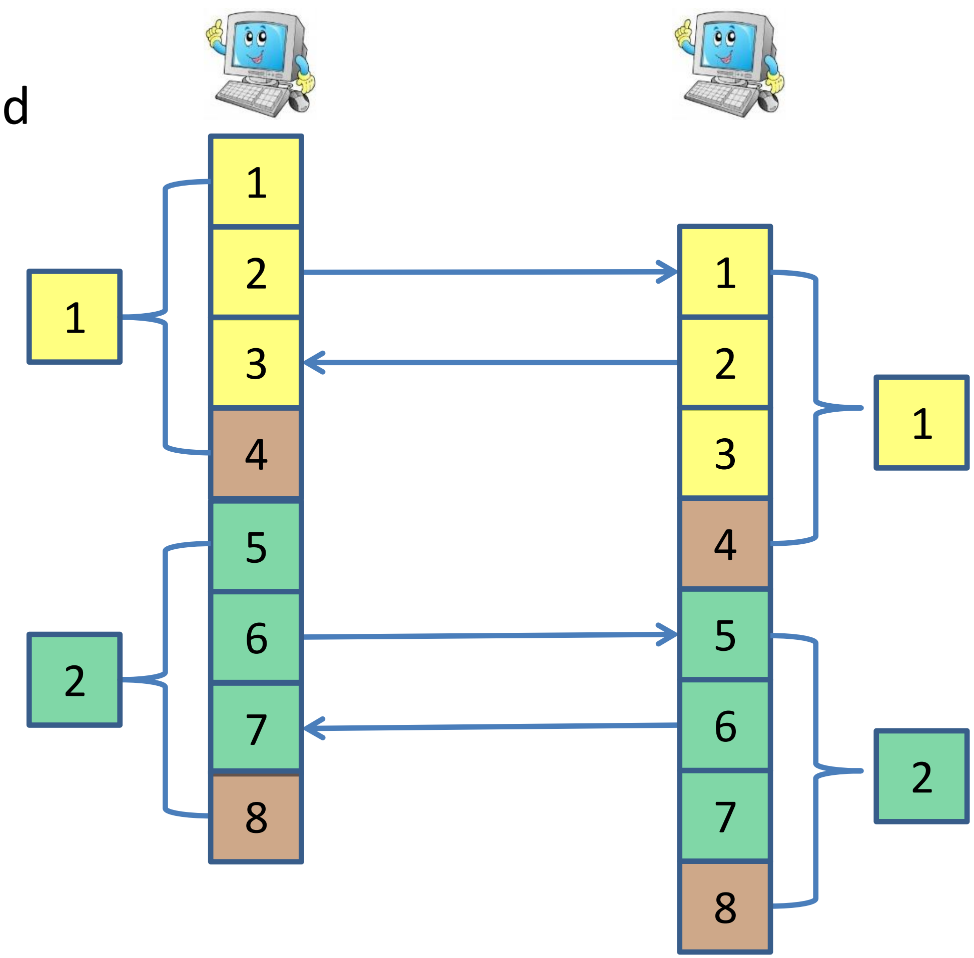
Non-Simultaneous Start

Main idea: add "dummy" rounds to make overlap meaningless

Extend each round to $3c+1$:

- $2c+1$ rounds: listen
  - Round $c+1$: listen & send
- c rounds : wait (without listening)

Example: PSMT ($c = 1$)



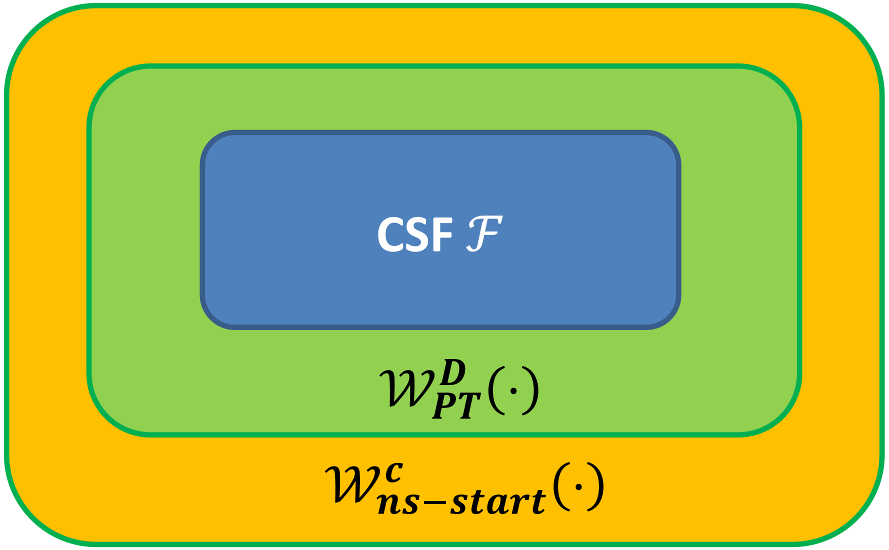
Concurrent Composition

- Each party proceeds in a locally sequential manner
- Round r messages **after** round $r-1$ **before** round $r+1$

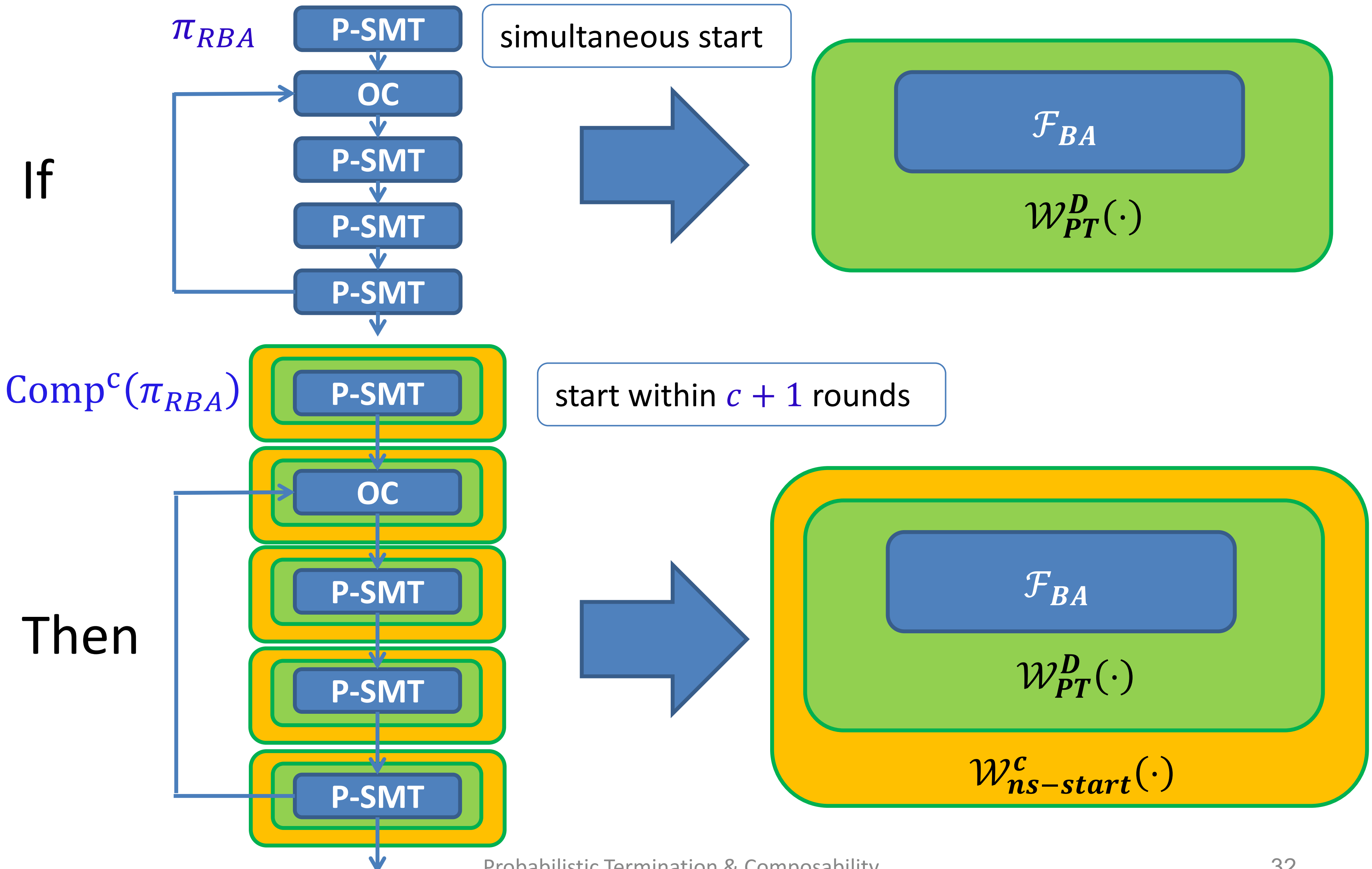
Slack Reduction

- PT hybrids might introduce additional slack
  $\Rightarrow$ rounds might blow-up
- Use slack-reduction techniques [Bracha'84]
  - Upon receiving output v, send (ok, v) to all the parties
  - Upon receiving $t+1$ messages (ok, v), accepts v
  - Upon receiving $n-t$ messages (ok, v), terminates
- Applies to public-output functionalities

Non-Simultaneous start wrapper



Composition Theorem (Illustrated)



Applications
(see the paper for more)

Parallel Broadcast

- n parallel runs of [FM'88] $\Rightarrow$ exp. $\Theta(\log n)$ rounds
- Prior constant-round solutions [BE'03] [FG'03] [KK'06] implement unfair parallel broadcast

[HZ'10] $\mathcal{A}$ can corrupt senders **based on their inputs** and replace the messages

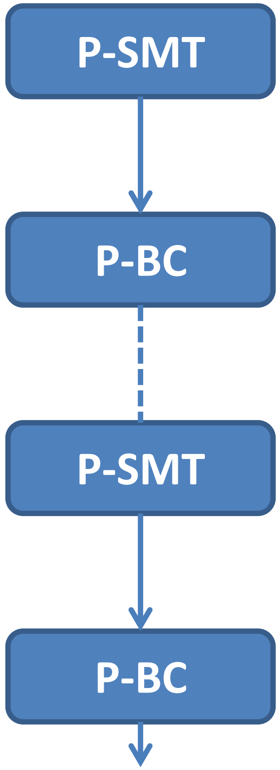
- We show how to get parallel broadcast from unfair parallel broadcast using secret sharing (not VSS)
- **Thm:** Let $c \geq 0$. $\mathcal{W}_{ns-start}^{c}\left(\mathcal{W}_{PT}^{D}(\mathcal{F}_{PBC})\right)$ can be realized in $\mathcal{F}_{PSMT}$-hybrid in expected $O(1)$ rounds, assuming all parties start within $c+1$ rounds

SFE with Expected $O(d)$ Rounds

Protocol [BGW'88] realizes $\mathcal{W}_{PT}^{D}(\mathcal{F}_{SFE})$ in $(\mathcal{F}_{PSMT}, \mathcal{F}_{PBC})$-hybrid in $O(d)$ rounds, assuming all parties start at same round

Thm: Let $c \geq 0$

$\mathcal{W}_{ns-start}^{c}\left(\mathcal{W}_{PT}^{D}(\mathcal{F}_{SFE})\right)$ can be realized in $\mathcal{F}_{PSMT}$-hybrid in expected $O(d)$ rounds, assuming all parties start within $c+1$ rounds



Summary

We study universal composability of cryptographic protocols with *probabilistic termination*

- Framework
  - Design and analyze simple protocols in modular composition fashion
  - Compile to UC protocols with same expected round complexity
- Perfect, adaptively secure protocols in the P2P model
  1) BA with expected $O(1)$ rounds
  2) Parallel broadcast with expected $O(1)$ rounds
  3) SFE with expected $O(d)$ rounds

Thank You