From Fairness to Full Security in Multiparty Computation

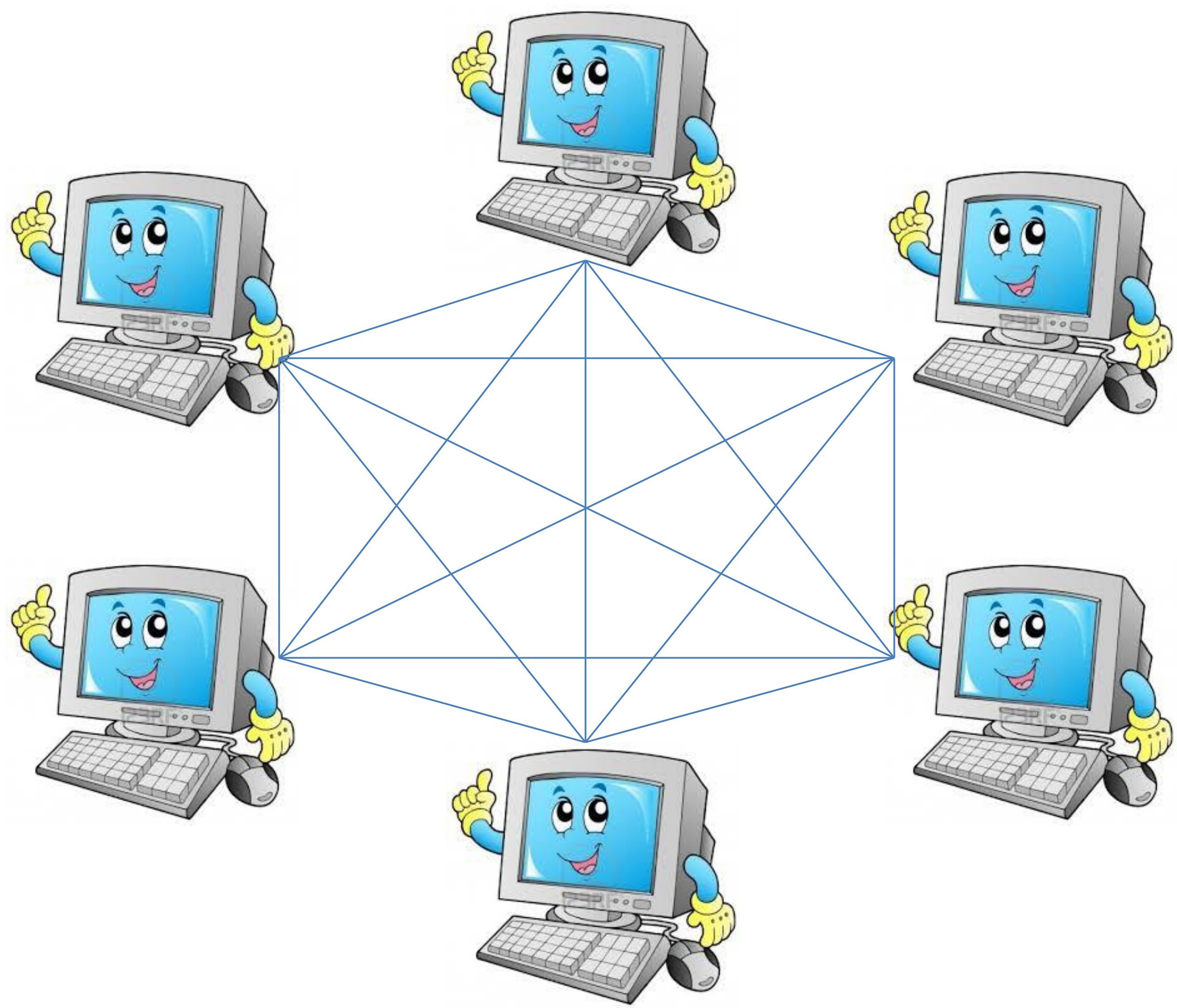
Ran Cohen (MIT & NEU)
Iftach Haitner (TAU)
Eran Omri (Ariel University)
Lior Rotem (HUJI)

Information Sharing

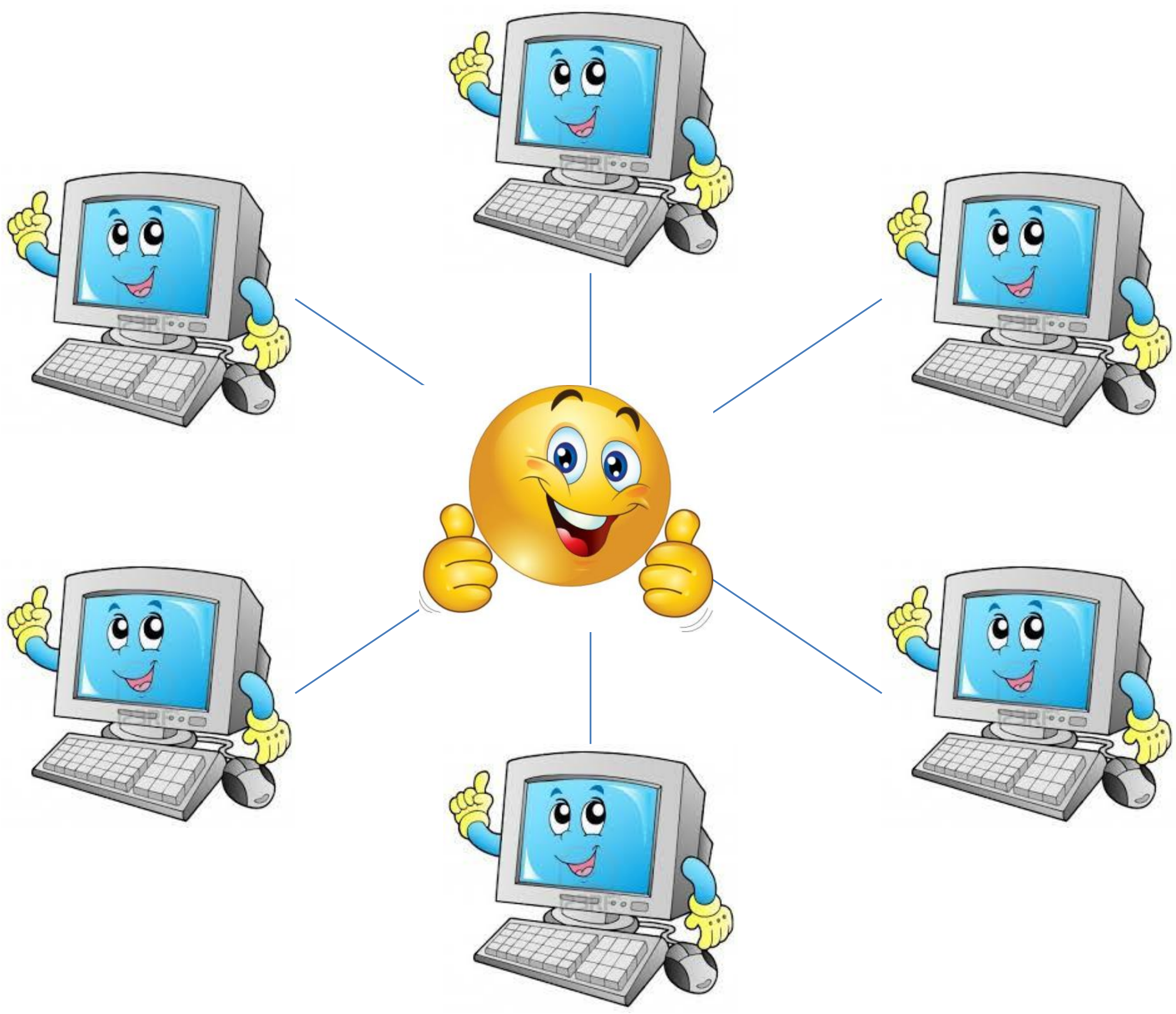
- A **terrorist threat** over the world
- Several **intelligence agencies** try to stop it
- Each agency has **secret data** – can't stop attack alone
- If the agencies **join forces** – they can stop the attack
- The terrorists have **double agents** in some agencies

Can the attack be stopped in time?

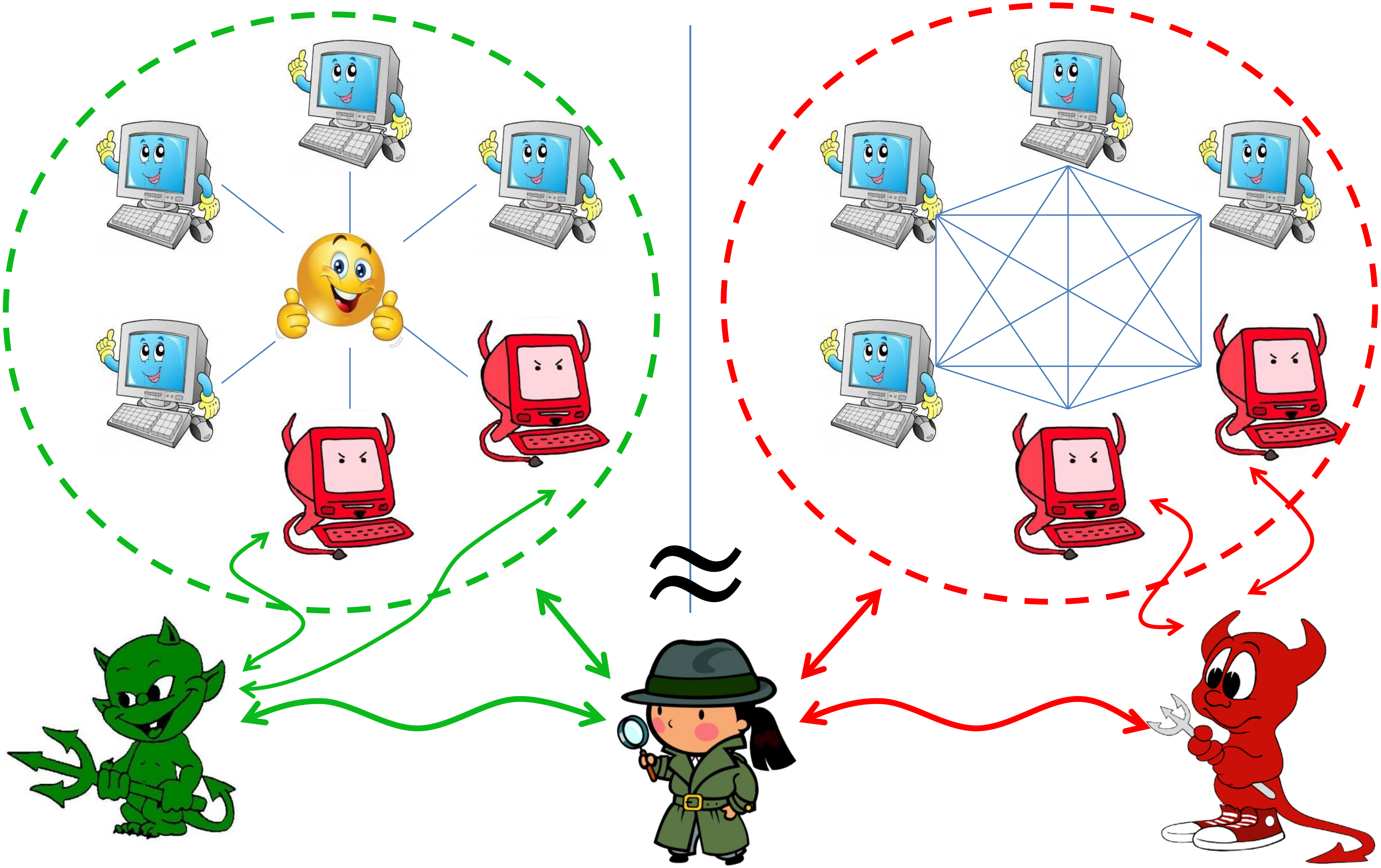
Secure Multiparty Computation



Ideal World

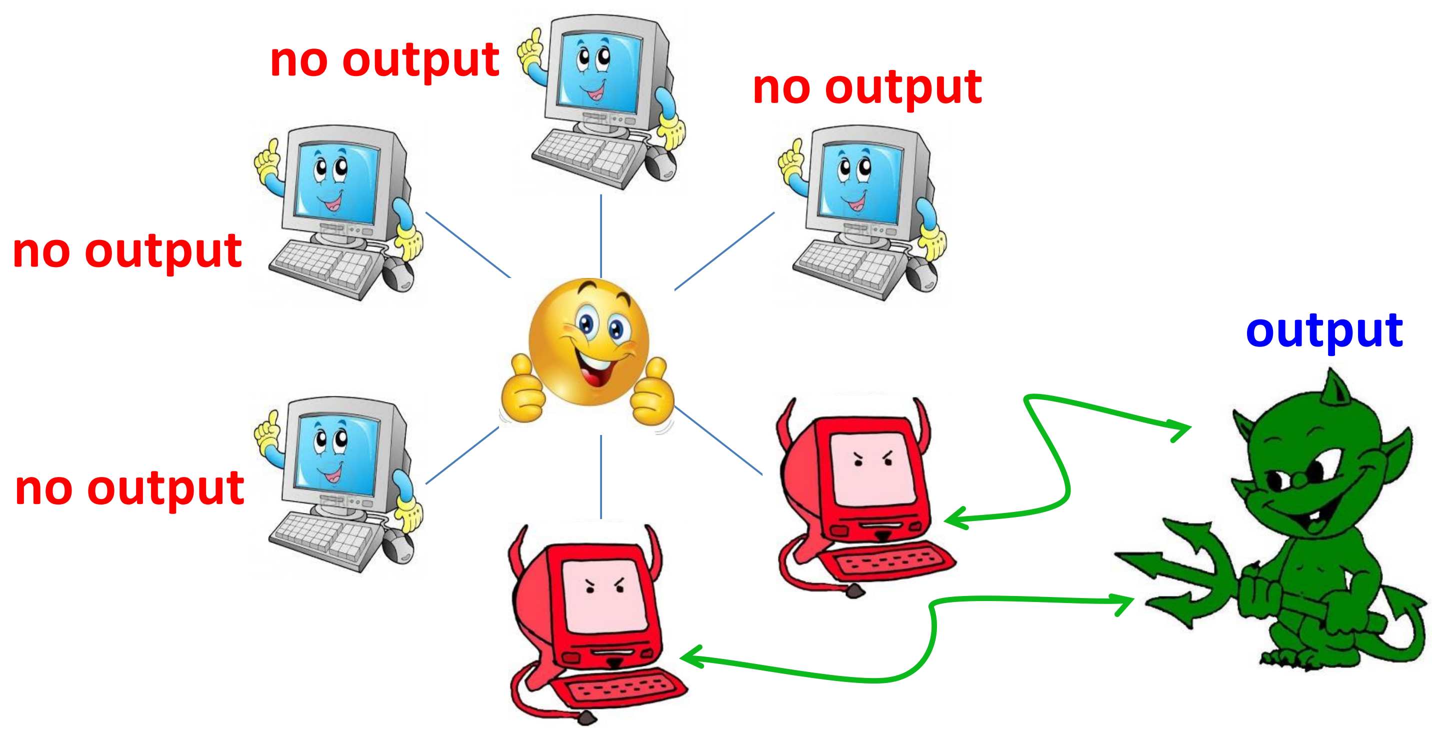


Security Definition



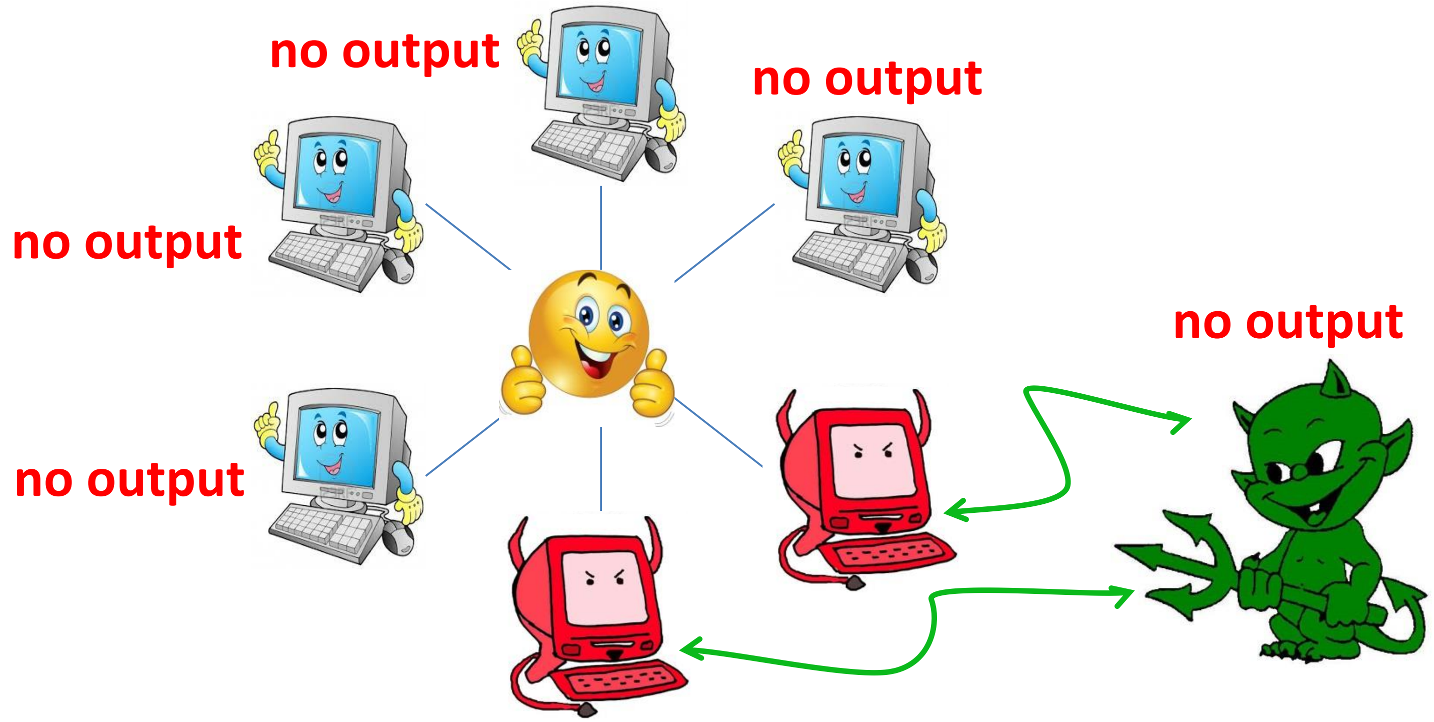
Notions of Security

- Security with abort: **abort after** obtaining output



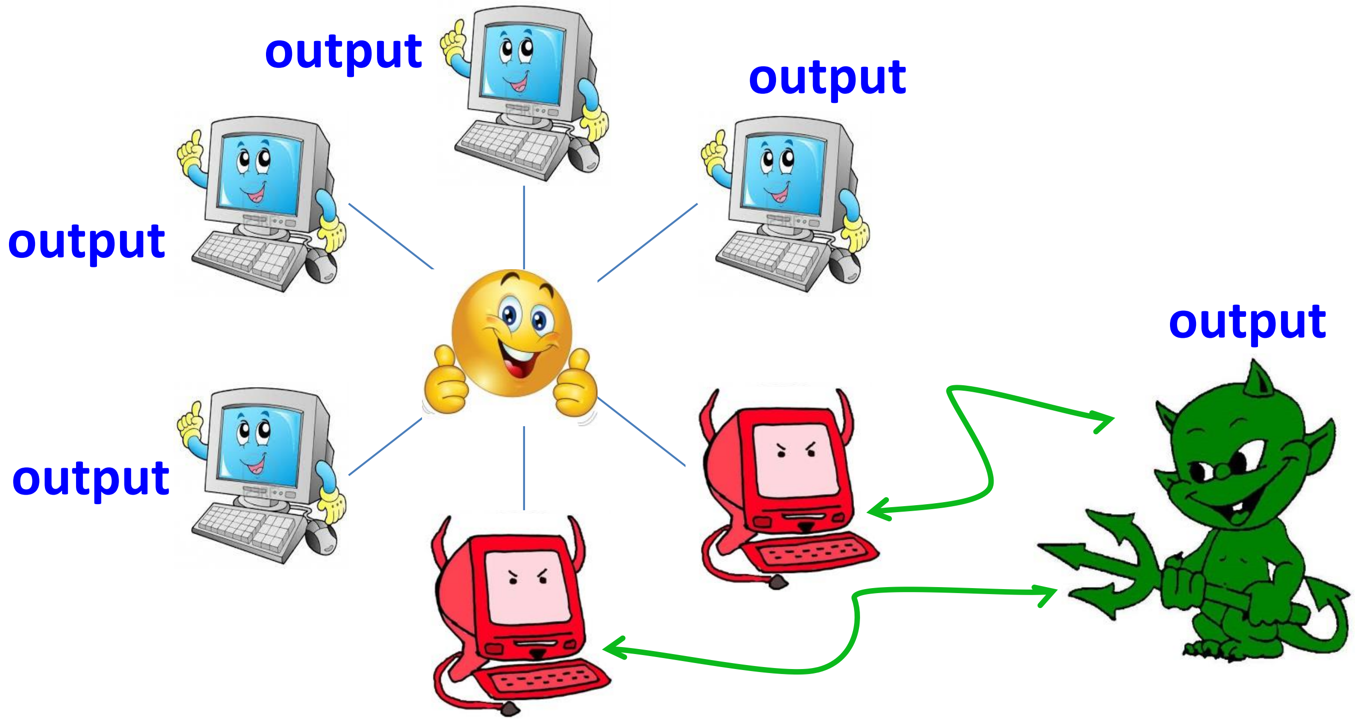
Notions of Security

- Security with abort: **abort after** obtaining output
- Fairness: **abort before** obtaining output



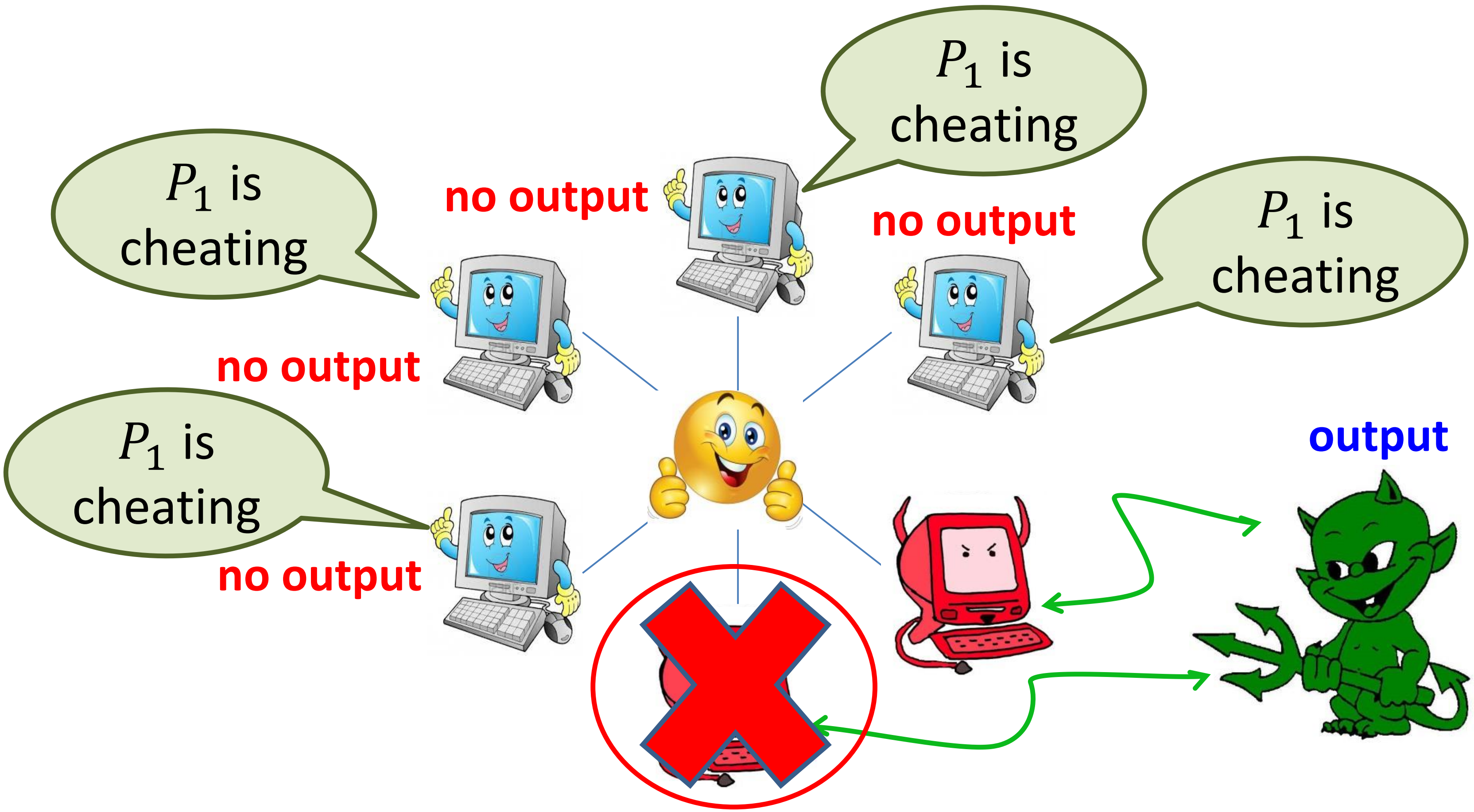
Notions of Security

- Security with abort: **abort after** obtaining output
- Fairness: **abort before** obtaining output
- Full security (guaranteed output delivery): **no abort**



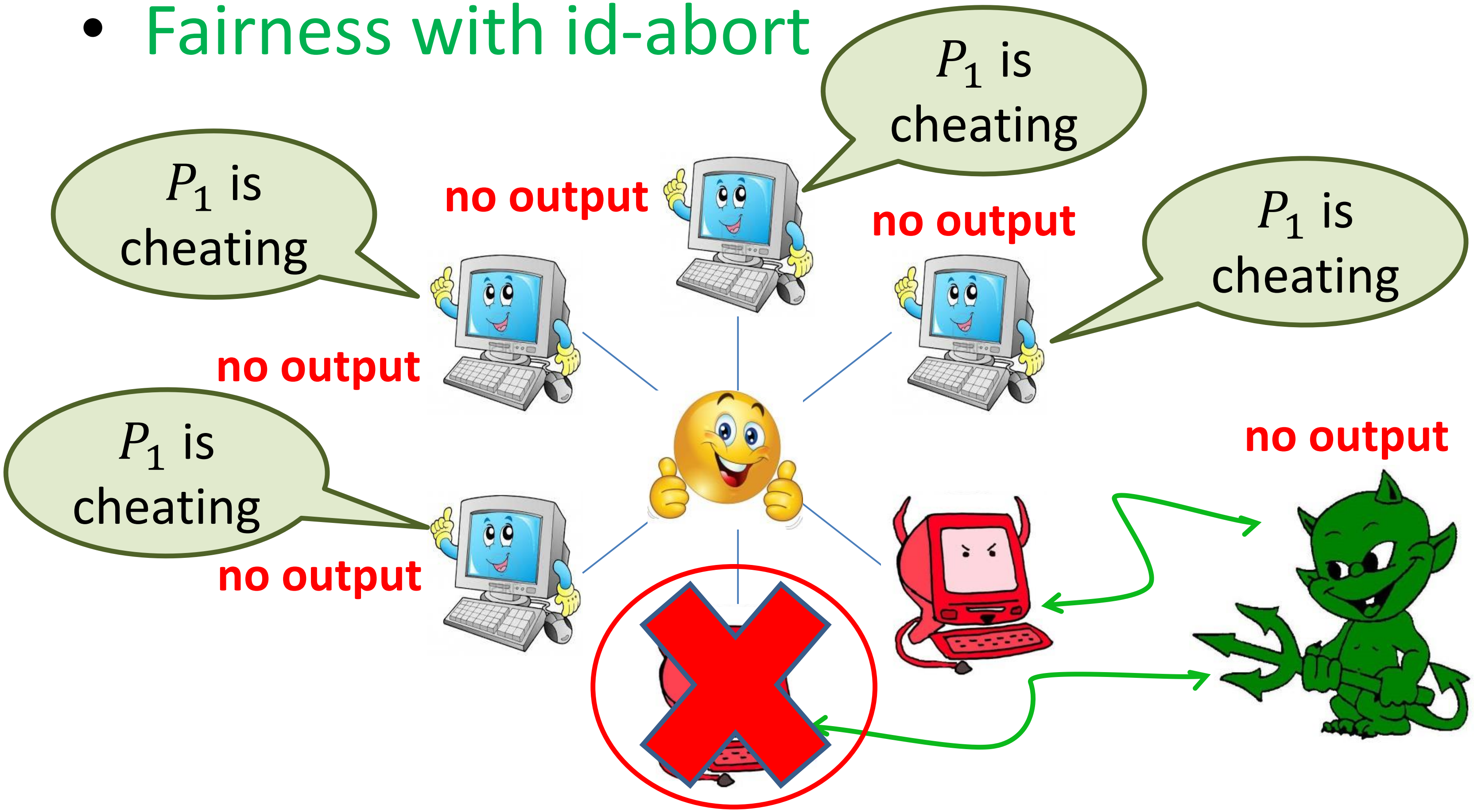
Identifiable Abort

- Security with id-abort: honest parties identify a corrupted party in case of abort



Identifiable Abort

- Security with id-abort: honest parties identify a corrupted party in case of abort
- Fairness with id-abort



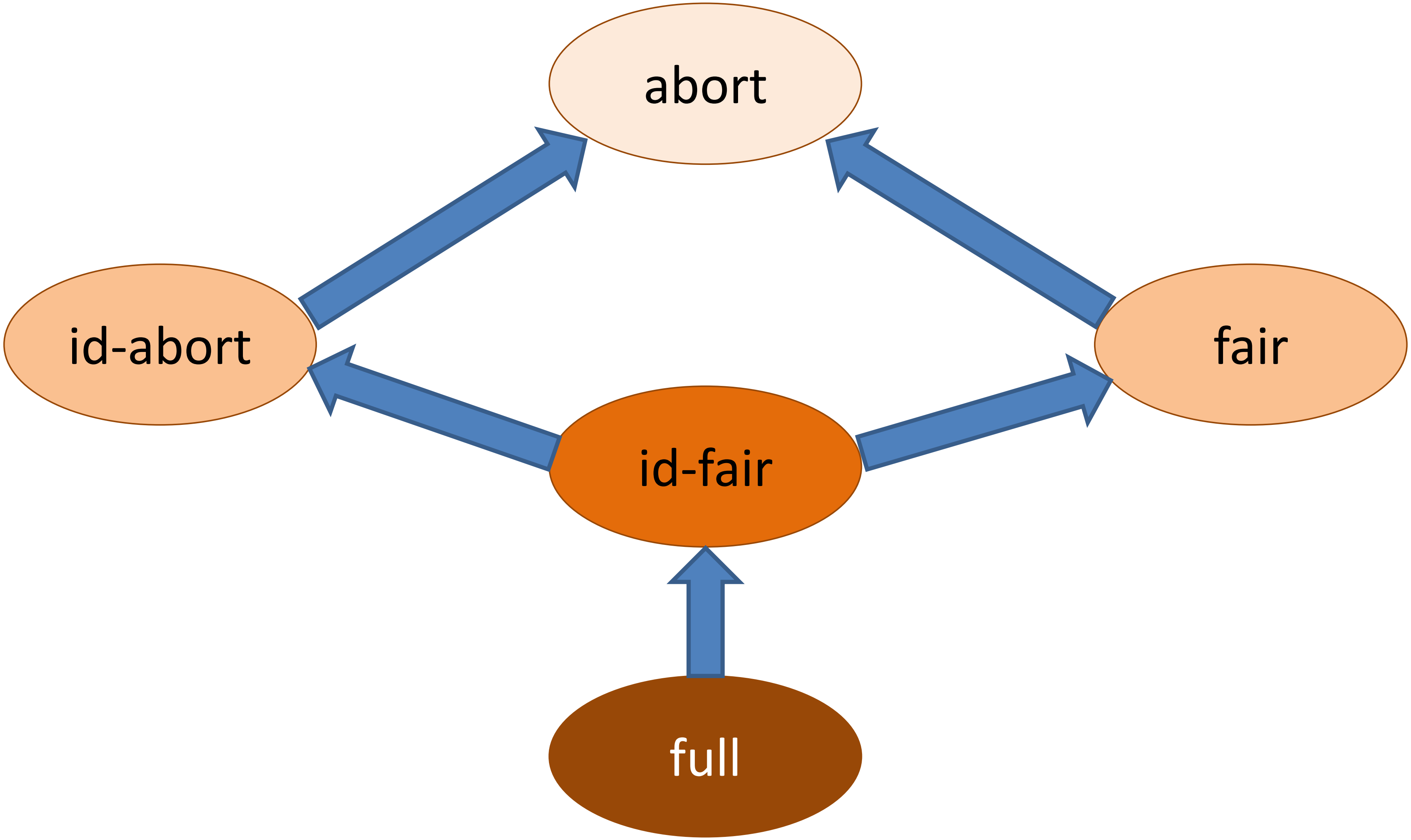
Known Results (w/o setup)

	Broadcast	**Point-to-Point**
$t < n/3$		✓ $\forall f$ full security [BGW'88, CCD'88]
$t < n/2$	✓ $\forall f$ full security [RB'89, CDDHR'99]	✗ $\exists f$ without full security [PSL'80, CL'14] ✗ $\exists f$ without id-abort [CL'14] ✓ $\forall f$ fairness [FGMR'02]
$t < n$ (*) assuming OT	✗ $\exists f$ without fairness [Cleve'86] ✓ $\forall f$ id-abort [GMW'87] ✓ $\exists f$ with full security [Gordon,Katz'09]	✗ $\exists f$ without fairness [Cleve'86] ✓ $\forall f$ security with abort [FGHHS'02] ✓ $\exists f$ with full security [CL'14, CHOR'16] ✓ $\exists f$ fair without full [CL'14]

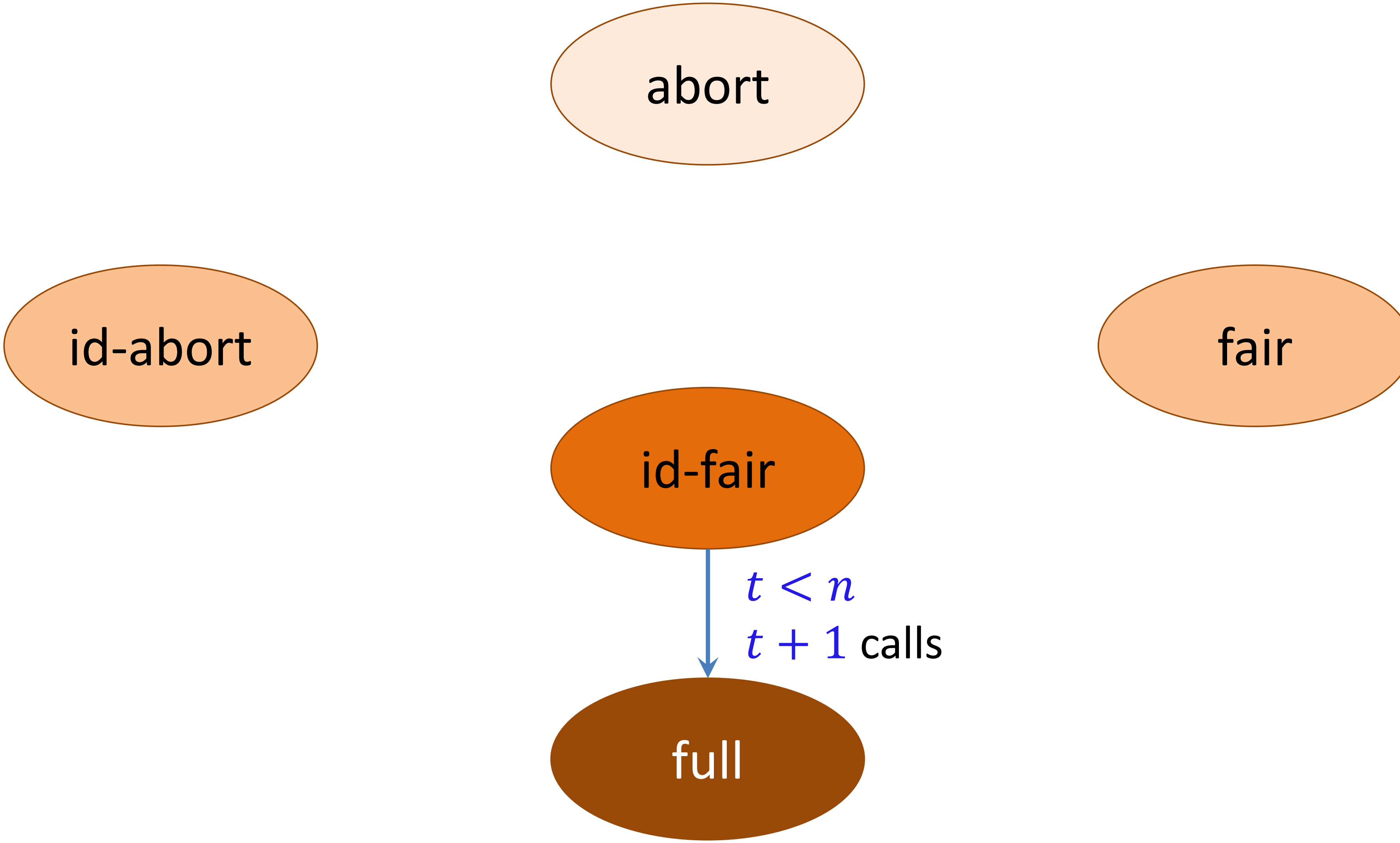
Known Results (w/o setup)

	Broadcast	Point-to-Point
$t < n/3$	✓ $\forall f$ full security [RB’89, CDDHR’99]	✓ $\forall f$ full security [BGW’88, CCD’88]
$t < n/2$		✗ $\exists f$ without full security [PSL’80,**C**L’14] ✗ $\exists f$ without id-abort [**C**L’14] ✓ $\forall f$ fairness [FGMR’02]
$t < n$ (*) assuming OT	✗ $\exists f$ without fairness [Cleve’86] ✓ $\forall f$ id-abort [GMW’87] ✓ $\exists f$ with full security [Gordon,Katz’09]	✗ $\exists f$ without fairness [Cleve’86] ✓ $\forall f$ security with abort [FGHHS’02] ✓ $\exists f$ with full security [**C**L’14, **C**HOR’16] ✓ $\exists f$ fair without full [**C**L’14]

Security Hierarchy



Security Hierarchy

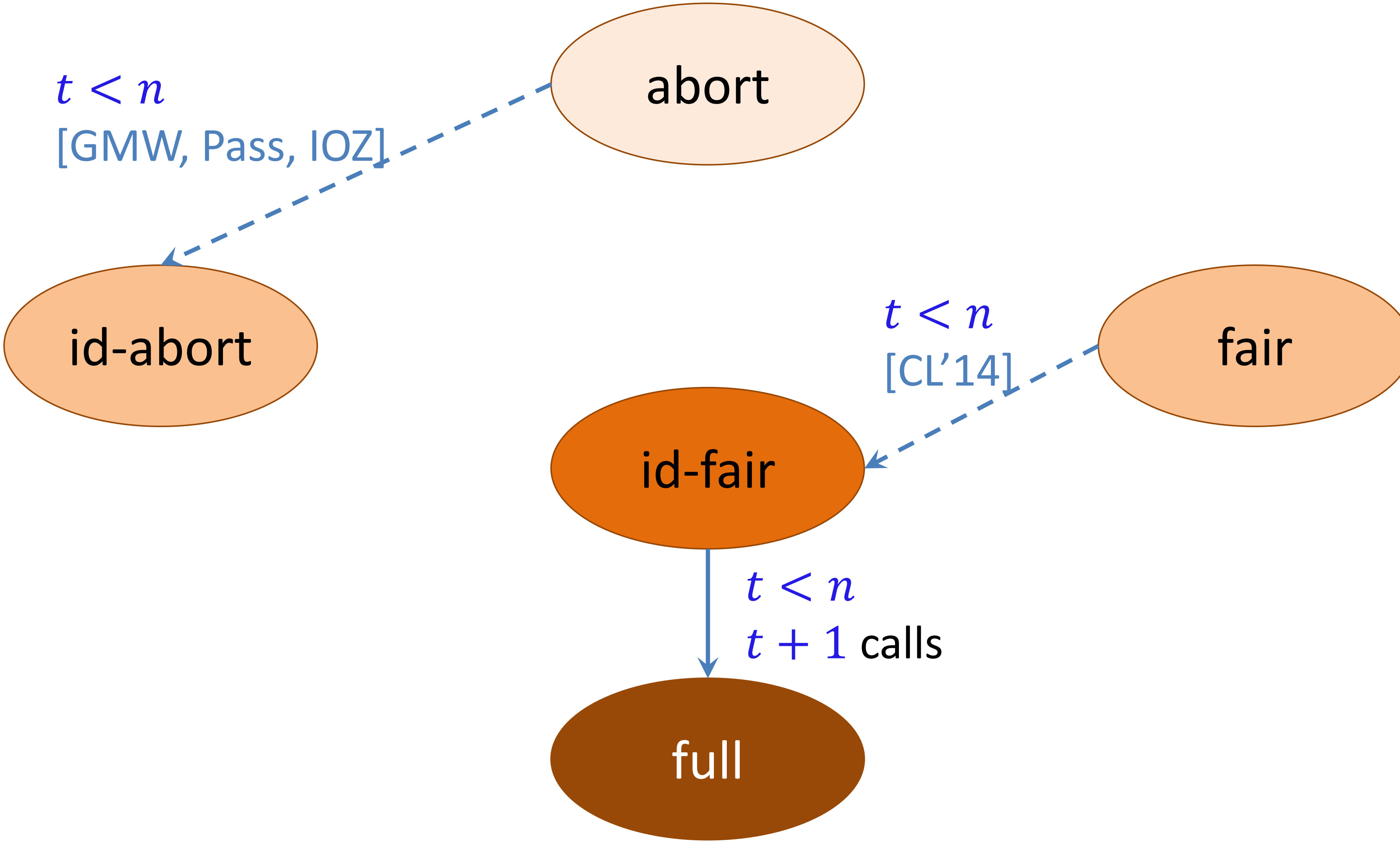


Id-Fair to Full Security ($t < n$)

Player-Elimination Technique

- Execute $t + 1$ times
  - Compute f with fairness & id-abort
  - If obtained output, halt
  - Otherwise, eliminate identified corrupted party

Security Hierarchy



Abort to Id-Abort ($t < n$)

GMW Paradigm

– Generate committed randomness (augmented CF)

– Commit to input

– Prove honest behavior in zero knowledge

[GMW'87]	[Pass'04]	[Ishai,Ostrovsky,Zikas'14]
OWF	TDP & CRH	Information theoretic (correlated randomness)
$O(n)$ rounds	$O(1)$ rounds	$O(1)$ rounds

[**C**,Lindell'14] fair to id-fair

Abort to Id-Abort ($t < n$)

GMW Paradigm

– Generate committed randomness (augmented CF)

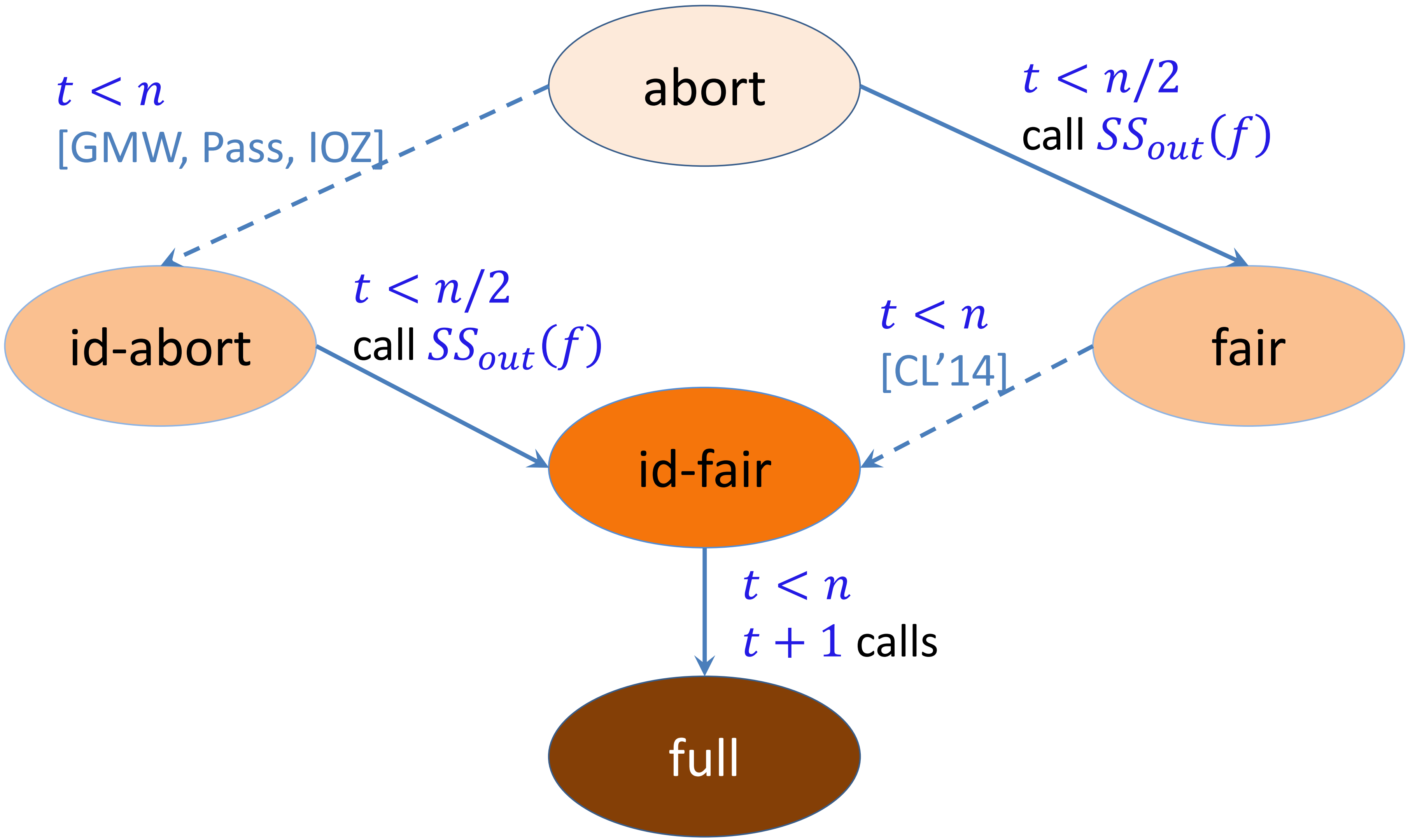
– Commit to input

– Prove honest behavior in zero knowledge

[GMW'87]	[Pass'04]	[Ishai,Ostrovsky,Zikas'14]
OWF	TDP & CRH	Information theoretic (correlated randomness)
$O(n)$ rounds	$O(1)$ rounds	$O(1)$ rounds

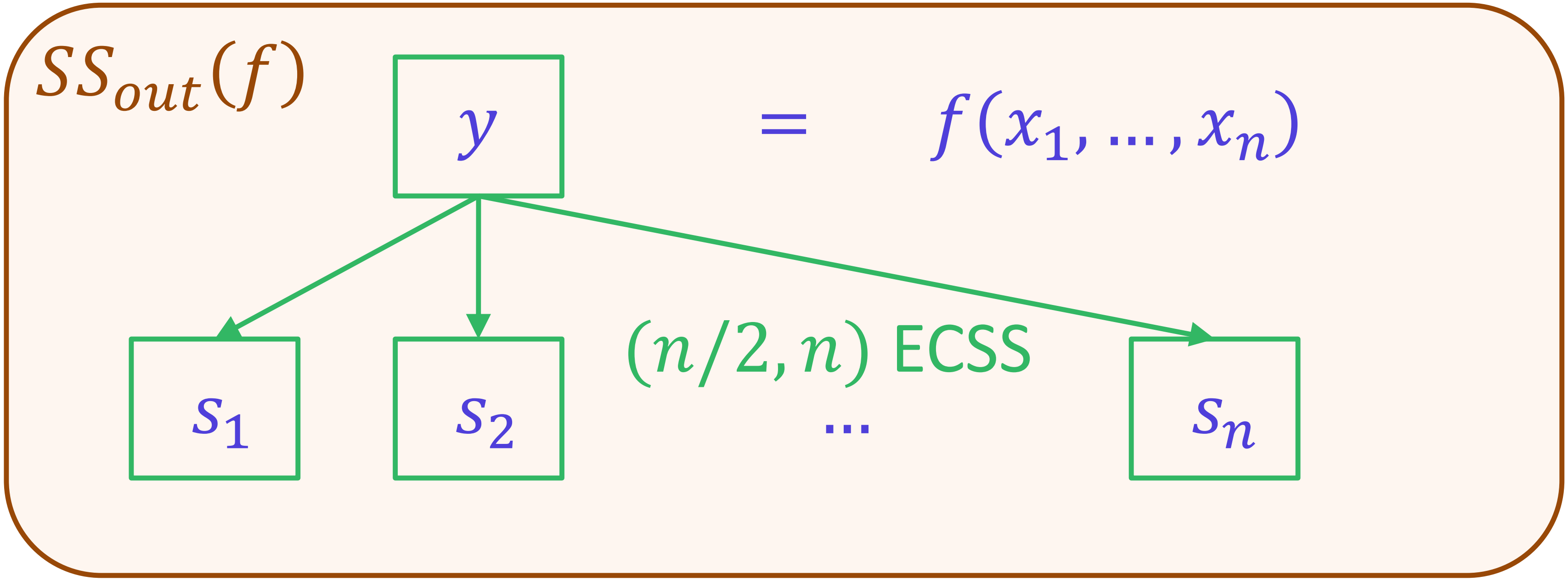
[**C**,Lindell'14] fair to id-fair

Security Hierarchy

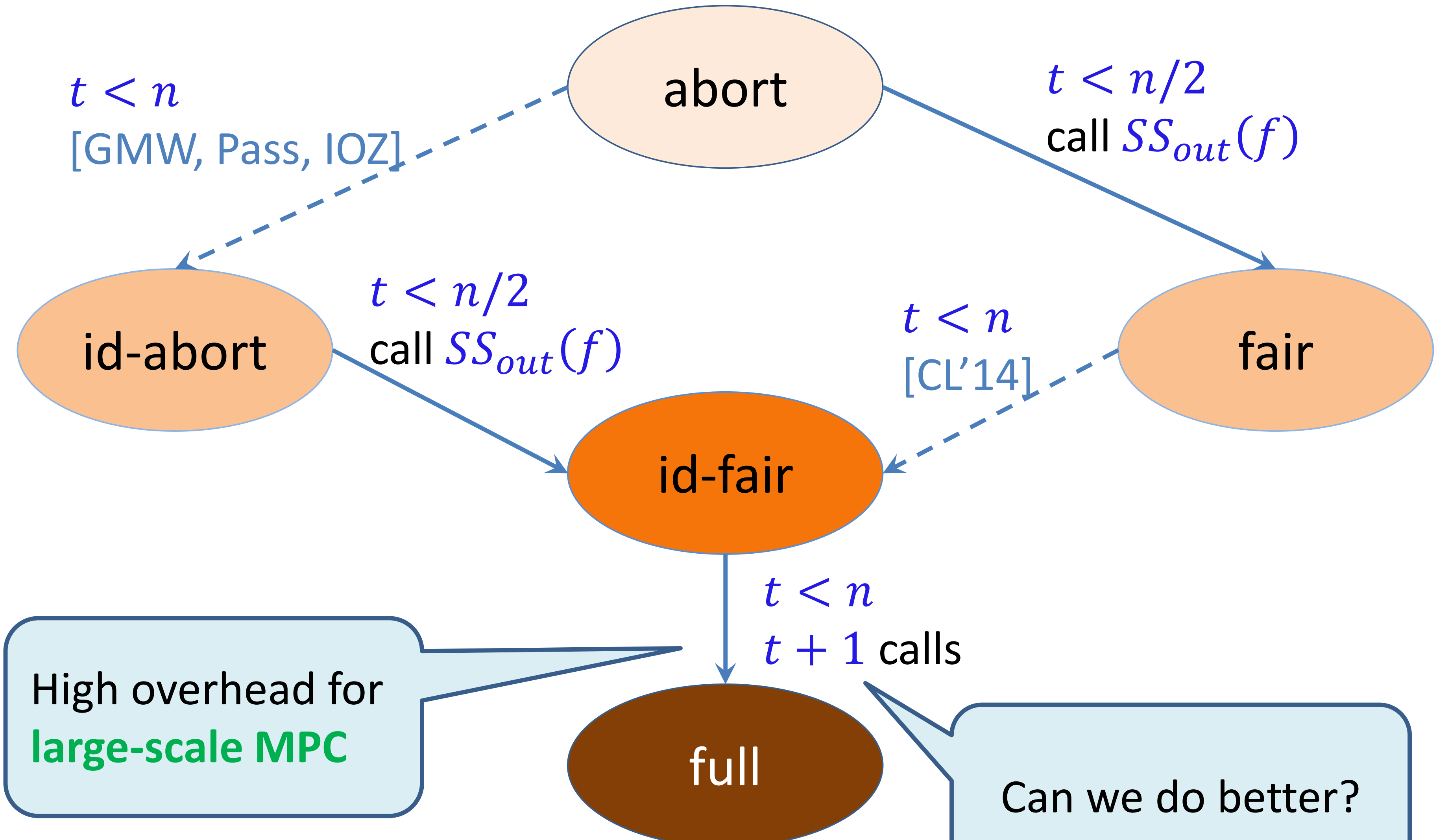


Abort to Fairness ($t < n/2$)

- Main tool: Error-Correcting Secret Sharing
  - $(s_1, \dots, s_n) \leftarrow \mathrm{Share}(s)$
  - Any set of t shares is independent of s
  - $s \leftarrow \mathrm{Recon}(s_1, \dots, s_n)$, even if t shares are incorrect
- Security with abort of $SS_{out}(f) \Rightarrow$ Fairness of f



Security Hierarchy



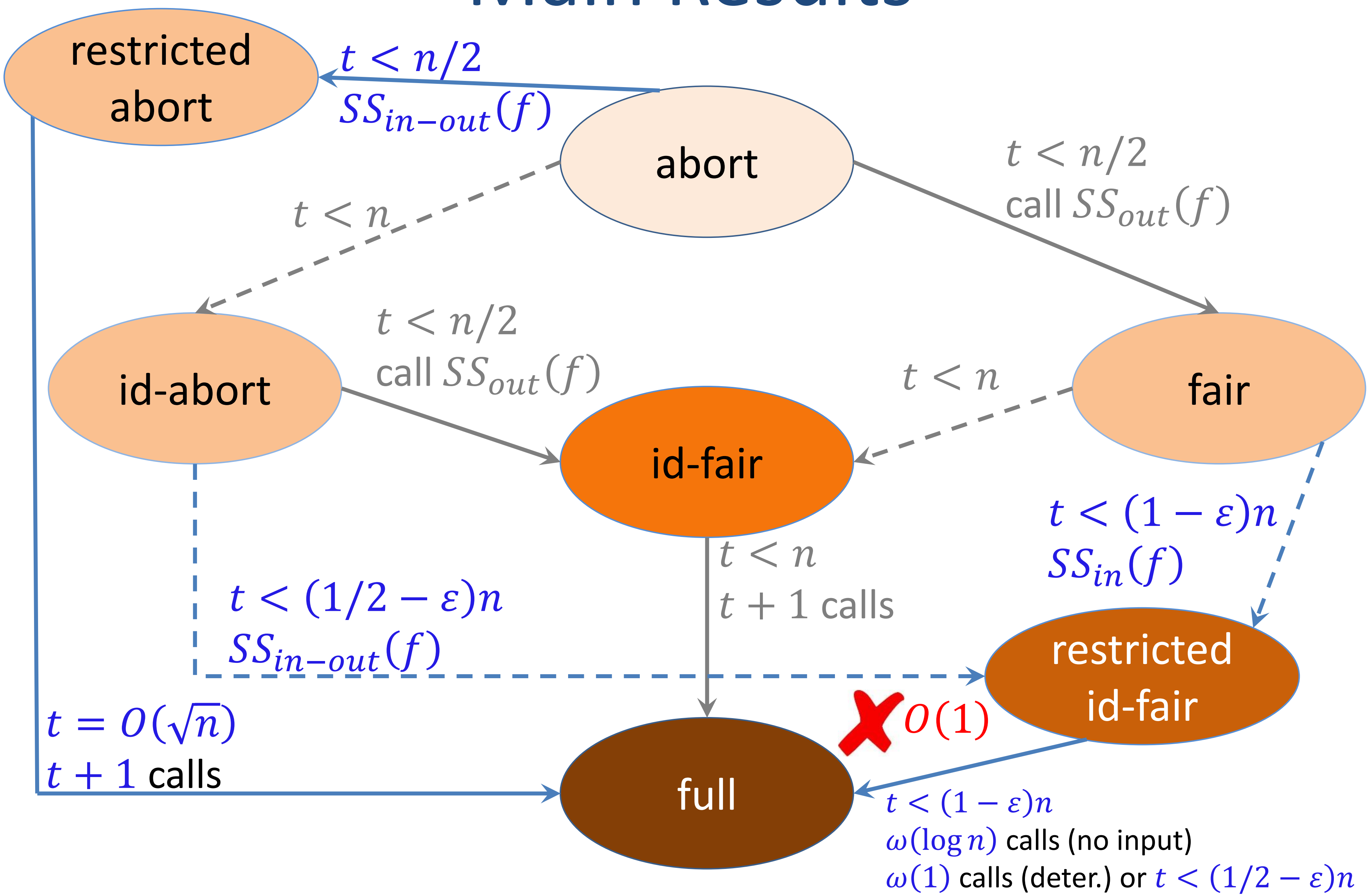
Main Question

The setting:

- Large-scale MPC
- Constant fraction of honest parties
  $t = \beta n$ for $0 < \beta < 1$

What is the cost (rounds) of transforming fair computation to fully secure computation?

Main Results



Rest of the talk

- **Randomized functionalities without inputs**
  - Fair to full in $\omega(\log n)$ rounds
  - Application: coin-flipping protocols
- **Functionalities with inputs**
  - Fair to full in $\omega(1)$ rounds
  - Application: multiparty Boolean OR
- **Lower bound**
  - No fair to full in $O(1)$ rounds

Randomized Functionalities Without Input

Thm1: Fairness to Full security (No Input)

- Let f be a no-input function
  - f^n is the n-party version (n copies of the output)
  - $n' = \omega(\log n)$
  - $t = \beta n$ and $t' = \beta' n'$ where $0 < \beta < \beta' < 1$
- If $f^{n'}$ is t'-comp. w/ fairness in r' rounds, then f^n is t-comp. w/ full security in $O(t' \cdot r')$ rounds

π comp. f^n
$r = O(t' \cdot r')$-round
Fully secure for t corrupt

π' comp. $f^{n'}$
r'-round
Fair for t' corrupt

Application: Coin Flipping

δ-bias coin flipping: the common output is δ-close to uniformly random bit, facing t corruptions

[Cleve'86]	δ-bias CF requires $\Omega(1/\delta)$ rounds

Application: Coin Flipping

δ-bias coin flipping: the common output is δ-close to uniformly random bit, facing t corruptions

[Cleve'86]	δ-bias CF requires $\Omega(1/\delta)$ rounds	
[ABCGM'85]	$t < n$	$O(t^2/\delta^2)$ rounds

Application: Coin Flipping

δ-bias coin flipping: the common output is δ-close to uniformly random bit, facing t corruptions

[Cleve'86]	δ-bias CF requires $\Omega(1/\delta)$ rounds	
[ABCGM'85]	$t < n$	$O(t^2/\delta^2)$ rounds
[MNS'09] [BOO'10] [HT'14] [AO'16] [BHLT'17]	$n < \log\log(1/\delta)$	$\tilde{O}(1/\delta)$ rounds

Application: Coin Flipping

δ-bias coin flipping: the common output is δ-close to uniformly random bit, facing t corruptions

[Cleve'86]	δ-bias CF requires $\Omega(1/\delta)$ rounds	
[ABCGM'85]	$t < n$	$\mathrm{O}(t^2/\delta^2)$ rounds
[MNS'09] [BOO'10] [HT'14] [AO'16] [BHLT'17]	$n < \log\log(1/\delta)$	$\tilde{\mathrm{O}}(1/\delta)$ rounds
[BOO'10]	$t = \beta n$, $1/2 < \beta < 1$	$O(n + 1/\delta^2)$ rounds

Application: Coin Flipping

δ-bias coin flipping: the common output is δ-close to uniformly random bit, facing t corruptions

[Cleve'86]	δ-bias CF requires $\Omega(1/\delta)$ rounds	
[ABCGM'85]	$t < n$	$O(t^2/\delta^2)$ rounds
[MNS'09] [BOO'10] [HT'14] [AO'16] [BHLT'17]	$n < \log\log(1/\delta)$	$\tilde{O}(1/\delta)$ rounds
[BOO'10]	$t = \beta n$, $1/2 < \beta < 1$	$O(n + 1/\delta^2)$ rounds
This work	$t = \beta n$, $1/2 < \beta < 1$	$O(\boldsymbol{\log(n)\log^*(n)} + 1/\delta^2)$

Main Idea

Restricting the adversary's ability to abort

1) Define restricted id-abort

2) Fairness & restricted id-abort $\Rightarrow$ full security

3) Fairness $\Rightarrow$ fairness & restricted id-abort

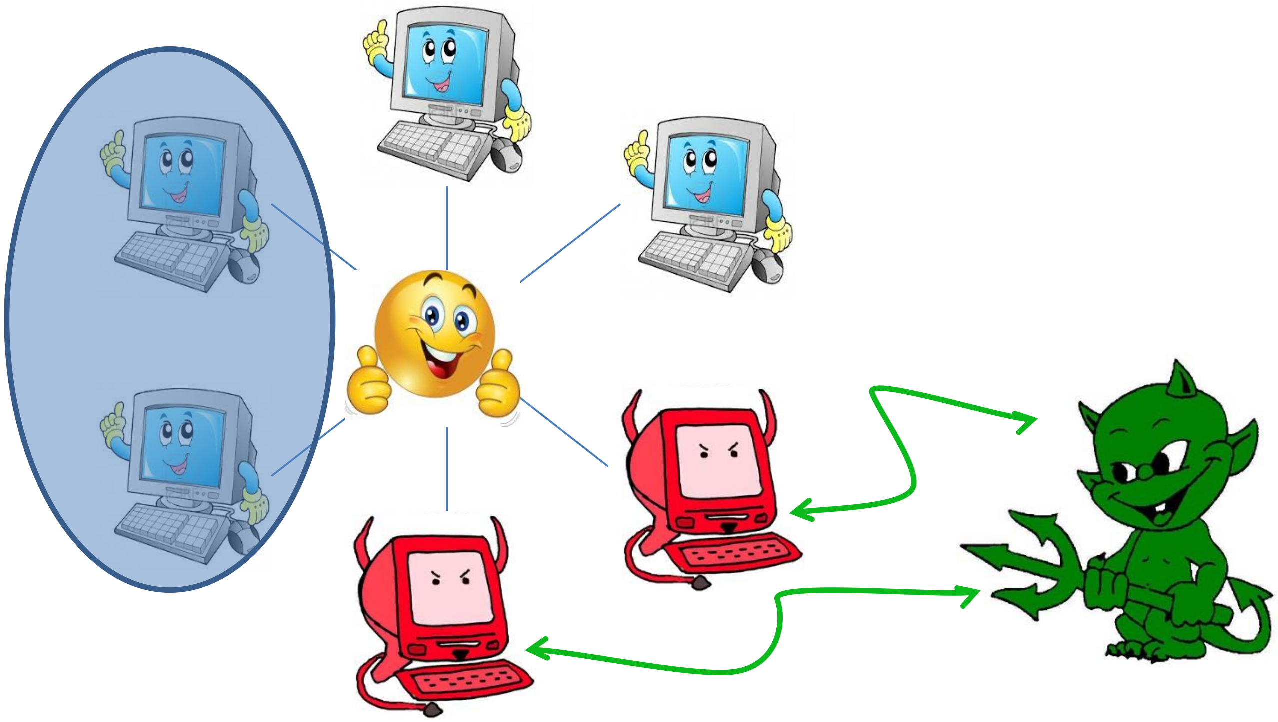
π comp. f^n
$r = O(t' \cdot r')$-round
Fully secure for t corrupt

π' comp. $f^{n'}$
r'-round
Fair for t' corrupt

Restricted Id-Abort

A designated subset of the parties $\mathcal{C}$ (committee)

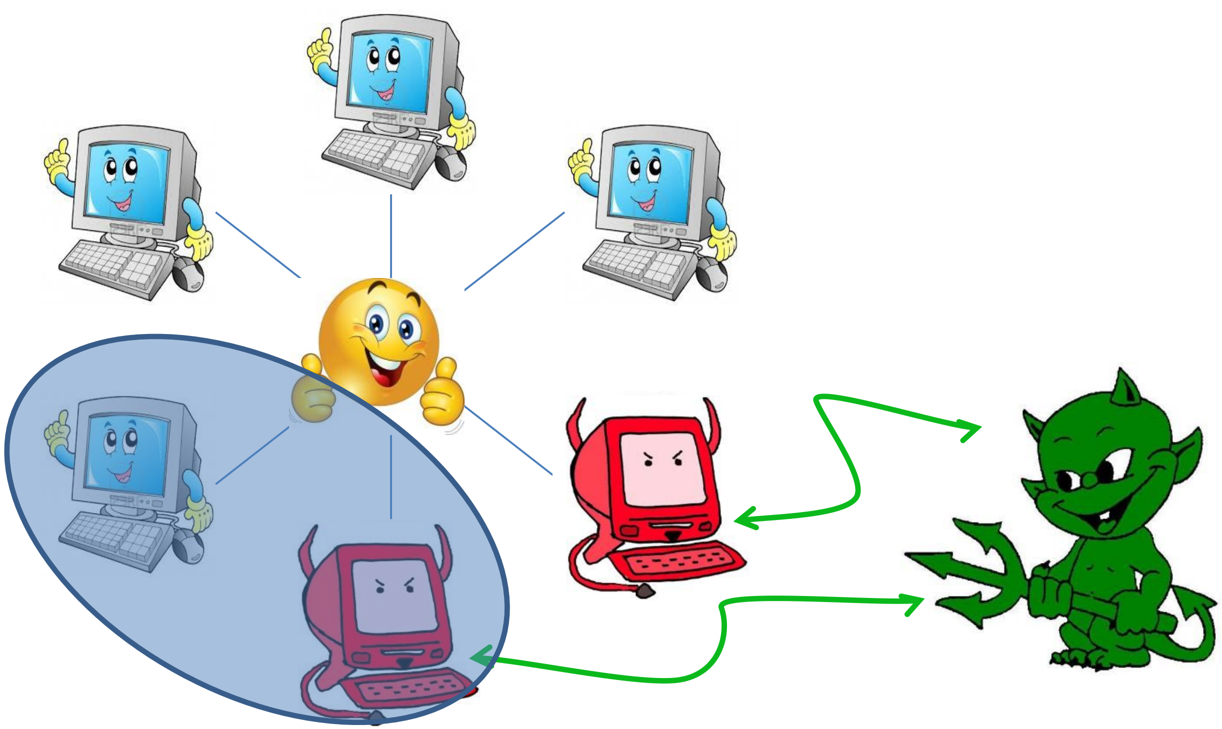
- If $\mathcal{C}$ is fully honest: no abort



Restricted Id-Abort

A designated subset of the parties $\mathcal{C}$ (committee)

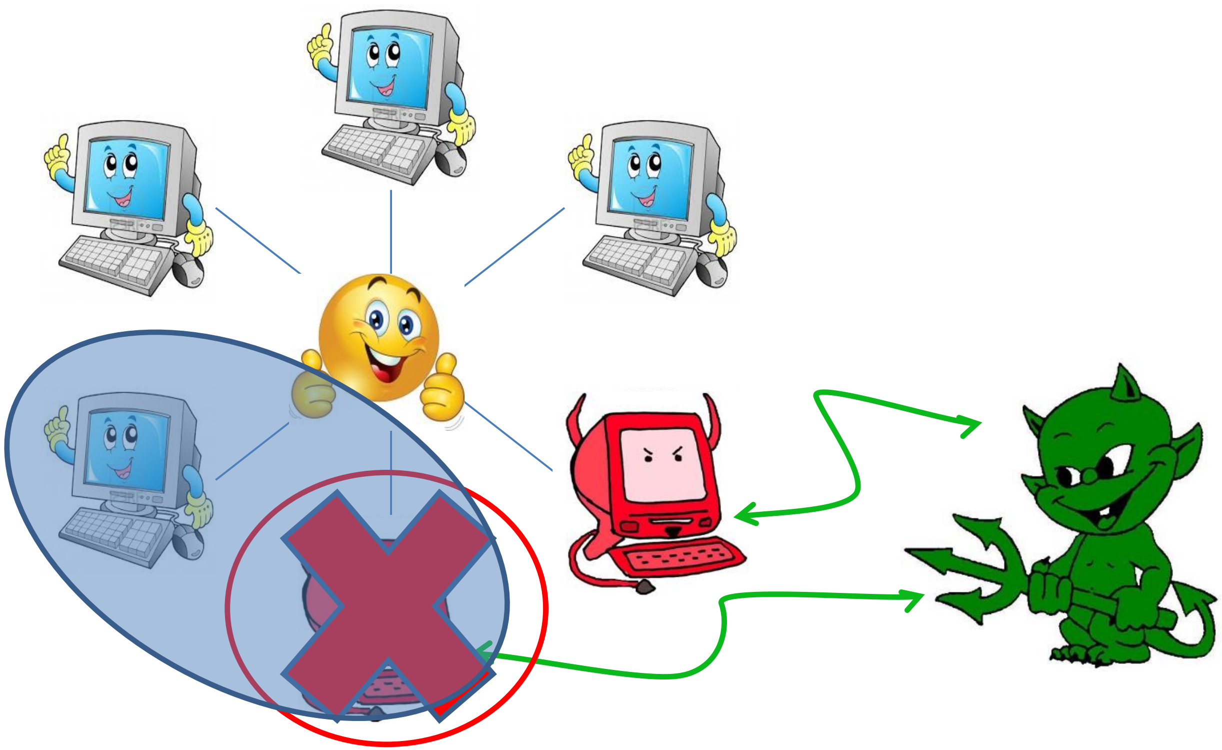
- If $\mathcal{C}$ is fully honest: no abort
- If $\mathcal{C}$ has corrupted party: id-abort in $\mathcal{C}$



Restricted Id-Abort

A designated subset of the parties $\mathcal{C}$ (committee)

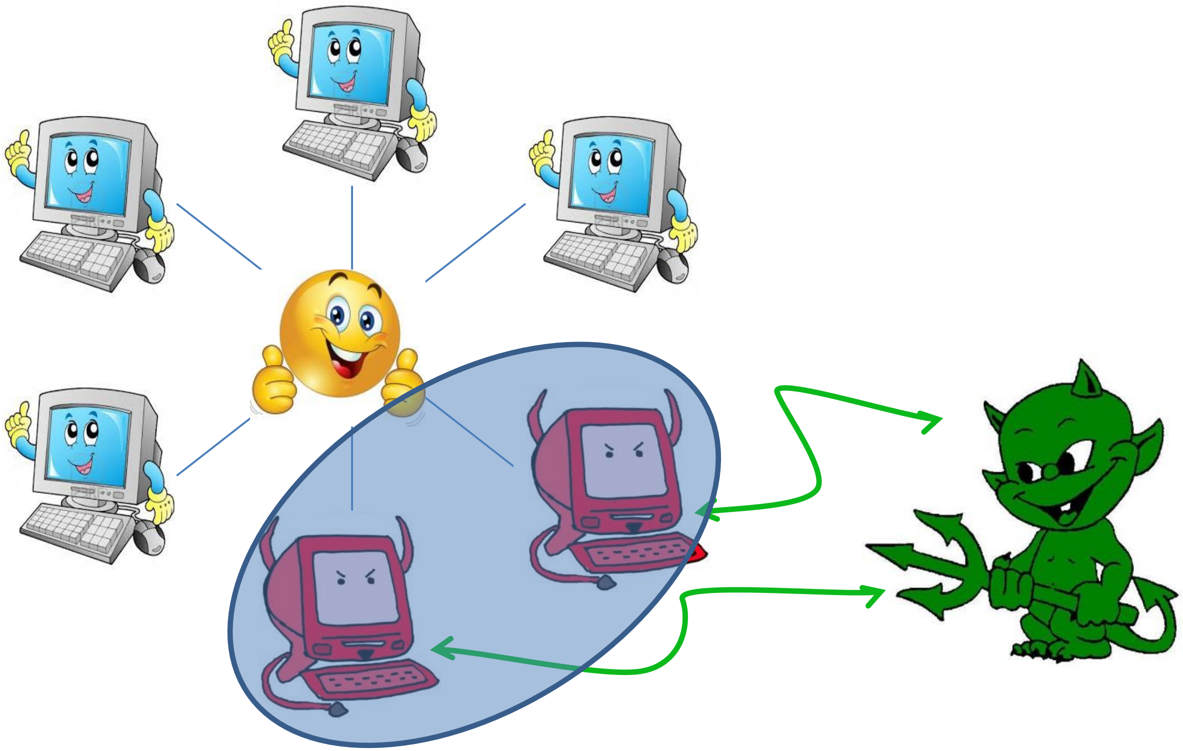
- If $\mathcal{C}$ is fully honest: no abort
- If $\mathcal{C}$ has corrupted party: id-abort in $\mathcal{C}$



Restricted Id-Abort

A designated subset of the parties $\mathcal{C}$ (committee)

- If $\mathcal{C}$ is fully honest: no abort
- If $\mathcal{C}$ has corrupted party: id-abort in $\mathcal{C}$
- If $\mathcal{C}$ is fully corrupted: adversary determines the output

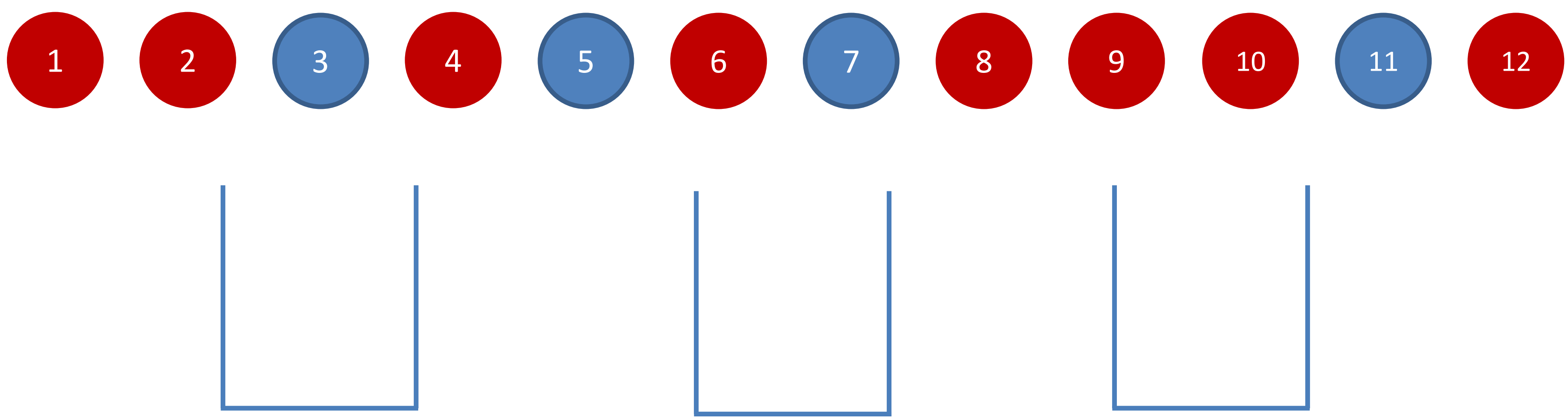


Restricted Id-Fair to Full

1) **Committee election** [Feige's lightest-bin protocol]

Elect committee $\mathcal{C}$ of size $n' = \omega(\log n)$

$\mathcal{C}$ has at most $(\beta + \varepsilon)n'$ corrupted parties, except negl prob



Restricted Id-Fair to Full

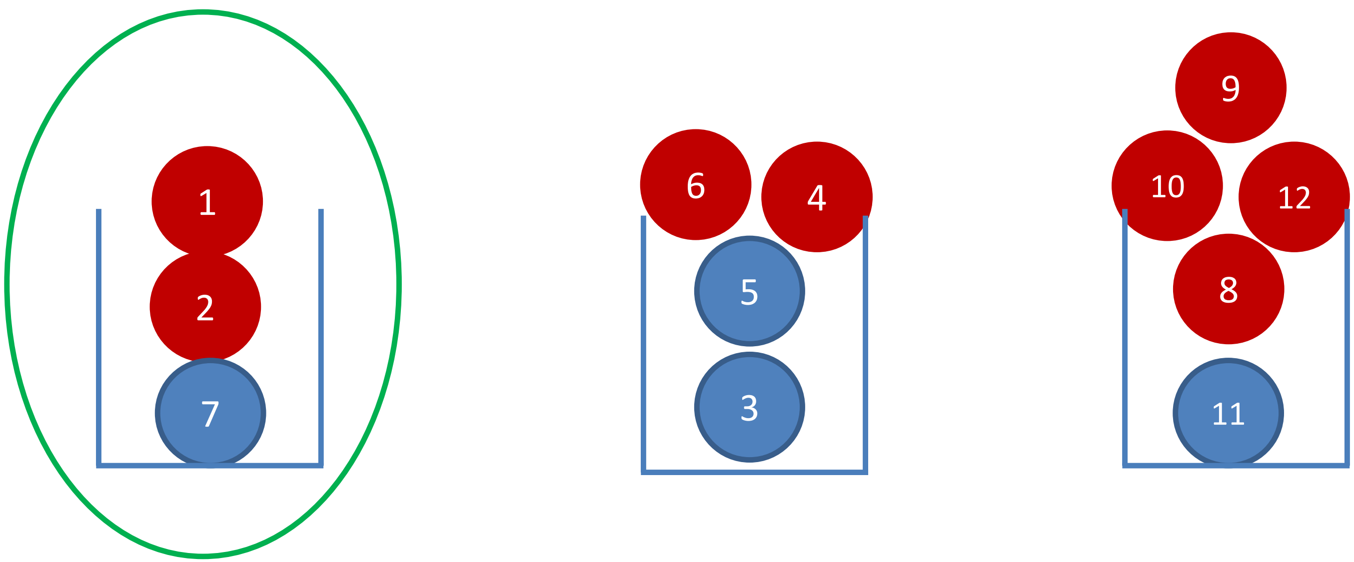
1) **Committee election** [Feige's lightest-bin protocol]

   Elect committee $\mathcal{C}$ of size $n' = \omega(\log n)$

   $\mathcal{C}$ has at most $(\beta + \varepsilon)n'$ corrupted parties, except negl prob

2) **Player elimination**

   $(\boldsymbol{\beta} + \boldsymbol{\varepsilon})\boldsymbol{n}' + \mathbf{1}$ iterations of f with fairness & $\mathcal{C}$-id-abort

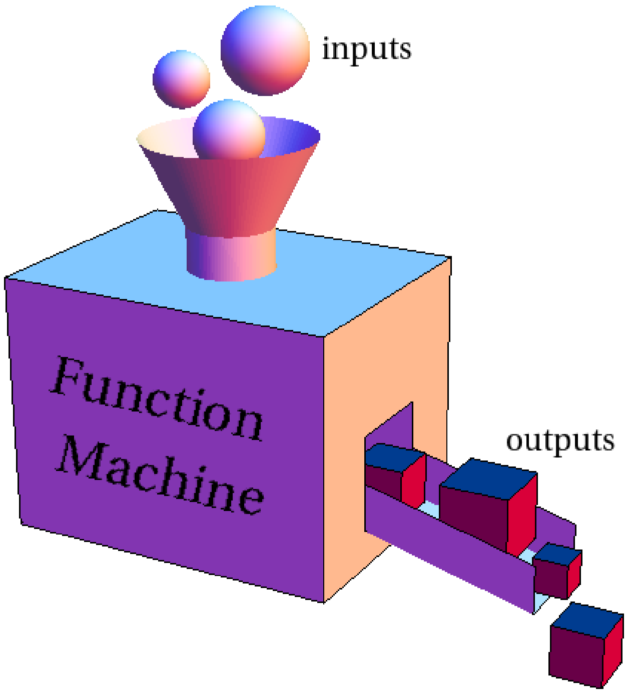


Obtaining Restricted Id-Fair

Committee members compute over broadcast:

1) Augmented coin flipping, security with id-abort
2) The function $f^{n'}$, fairness with id-abort
3) Broadcast output and prove correctness [Pass'04]

Functions With Input



Thm 2: Functions With Input

Let f be a n-party function, let $t = \beta n$, and let $n' = \omega(\log n)$

If $SS_{in}(f)$ is $(n' - 1)$-computed w/ fairness in parallel in r rounds, then f is t-computed w/ full security in $O(r \cdot \log^* n)$ rounds

any $\omega(1)$ funtion

Application: Boolean OR

$$f(x_1, \ldots, x_n) = x_1 \vee \cdots \vee x_n$$

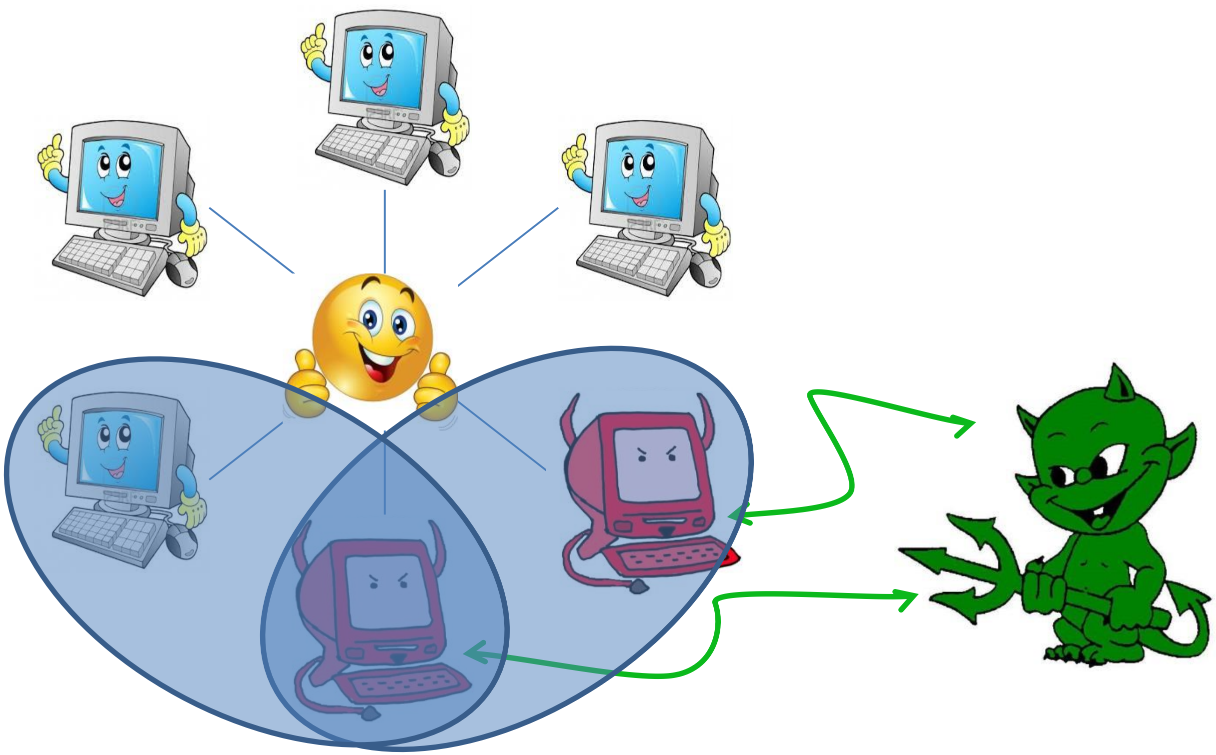
- [Gordon,Katz'09] Fully secure Boolean OR facing $t < n$ with $\mathrm{O}(n)$ rounds

- **This work:** Fully secure Boolean OR facing $t = \beta n$ with $\mathrm{O}(\boldsymbol{\log^* n})$ rounds

Restricted Id-Abort (With Input)

Multiple committees $\mathcal{C}_1, \ldots, \mathcal{C}_\ell$

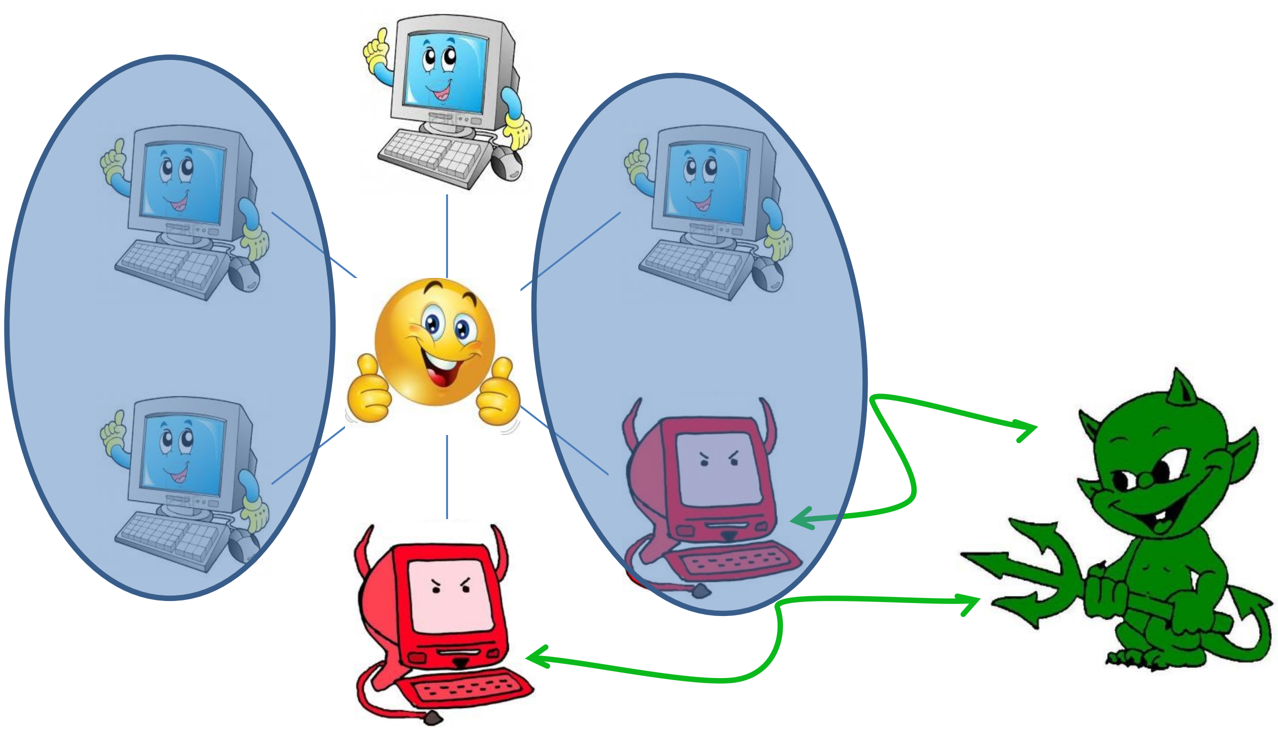
- If $\exists$ fully corrupted $\mathcal{C}_i$: $\mathcal{A}$ lerans all inputs & determines output



Restricted Id-Abort (With Input)

Multiple committees $\mathcal{C}_1, \dots, \mathcal{C}_\ell$

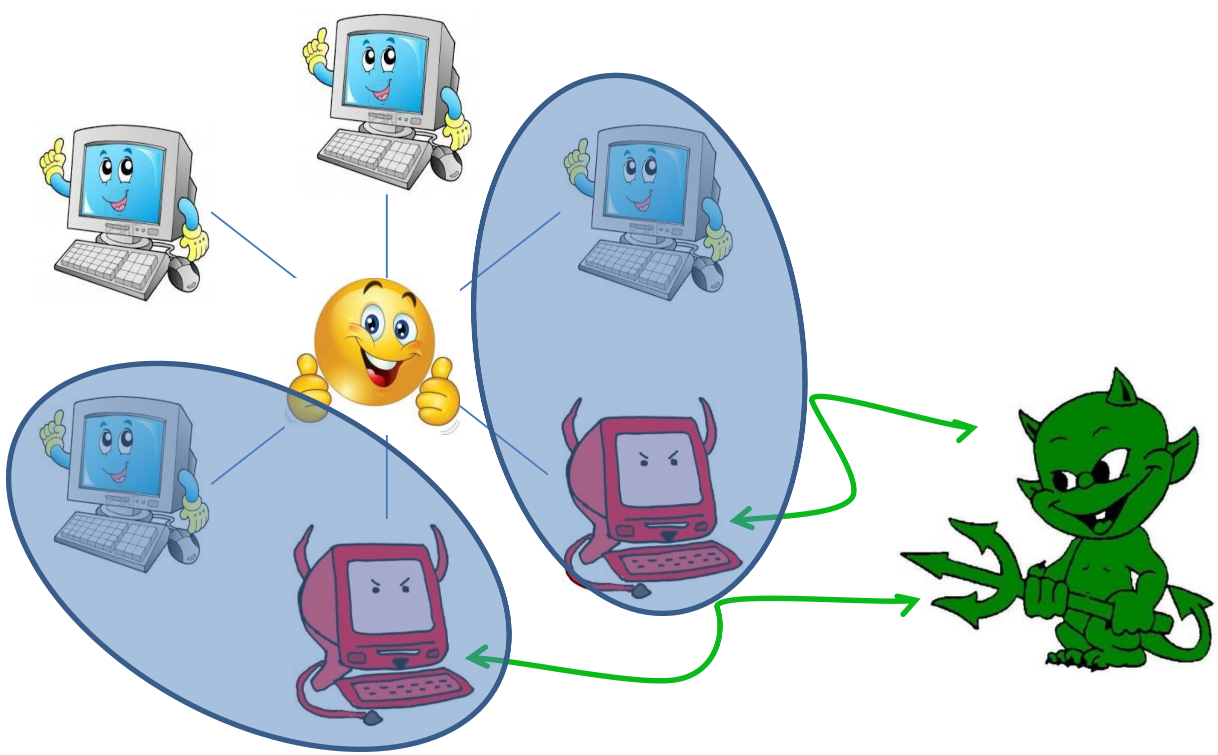
- If $\exists$ fully corrupted $\mathcal{C}_i$: $\mathcal{A}$ lerans all inputs & determines output
- If $\exists$ fully honest $\mathcal{C}_i$: no abort



Restricted Id-Abort (With Input)

Multiple committees $\mathcal{C}_1, \ldots, \mathcal{C}_\ell$

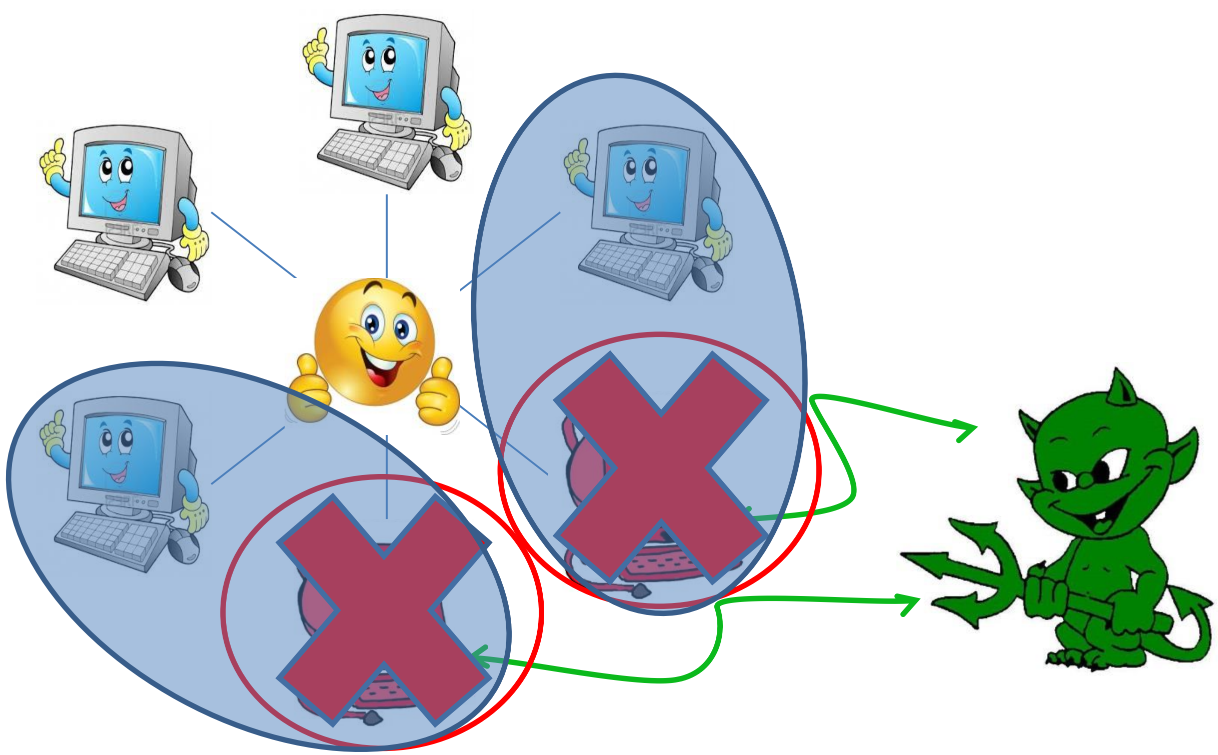
- If $\exists$ fully corrupted $\mathcal{C}_i$: $\mathcal{A}$ lerans all inputs & determines output
- If $\exists$ fully honest $\mathcal{C}_i$: no abort
- Otherwise : abort by identifying corrupted party in **every** $\mathcal{C}_i$



Restricted Id-Abort (With Input)

Multiple committees $\mathcal{C}_1, \ldots, \mathcal{C}_\ell$

- If $\exists$ fully corrupted $\mathcal{C}_i$: $\mathcal{A}$ lerans all inputs & determines output
- If $\exists$ fully honest $\mathcal{C}_i$: no abort
- Otherwise : abort by identifying corrupted party in **every** $\mathcal{C}_i$



Restricted Id-Fair to Full in $\omega(1)$

1) **Committee election**
   Elect committee $\mathcal{C}$ of size $m = \omega(\log n)$
2) **Fix sub-committees**
   All subsets $\mathcal{C}_1, \ldots, \mathcal{C}_\ell \subseteq \mathcal{C}$ of size $n' = m - n''$
3) **Player elimination**
   Compute f with fairness & $(\mathcal{C}_1, \ldots, \mathcal{C}_\ell)$-id-abort

Lemma: Let $\varphi(n) \in \omega(1)$

For $m = \log n \cdot \varphi(n)$ and $n'' = \log n / \varphi(n)$

- No $\mathcal{C}_i$ is fully corrupted (except negl. probability)
- There are poly-many $\mathcal{C}_i$'s
- if $\mathcal{A}$ aborts, n'' parties are identified

$\Rightarrow$ Full security in $m/n'' = \varphi(n)^2$ iterations

Obtaining Restricted Id-Fair

Problem:

How to send inputs to committee

Solution:

Each party n'-out-of-n' secret shares its input

Another Problem:

Bad committee members might change shares

Solution:

Functionality $SS_{in}(f)$ will verify shares

Doesn't follow from fairness

More Problems:

- Identify corrupted members before learning output
- Corrupted committee members don't blame honest

Computing Over Shared Inputs

Each party P_i:

1) Compute $x_i = s_1 \oplus \cdots \oplus s_{n'}$
2) $\forall j \in [n']$ broadcast $c_j = \mathrm{Com}(s_j; r_j)$
3) $\forall j \in [n']$ broadcast $\mathrm{Enc}_{pk_j}(s_j, r_j)$
4) Prove honest behavior

Perfectly binding

Each committee member $\widetilde{P}_j$:

1) Obtain relevant decommitments
2) Use the decommitments as inputs to $SS_{in}(f)$

The Functionality $SS_{in}(f)$

Parameters: commitments sent by the parties

Input: $\forall j \in [n']$, n-vector of decommitments

Verify all commitments open properly

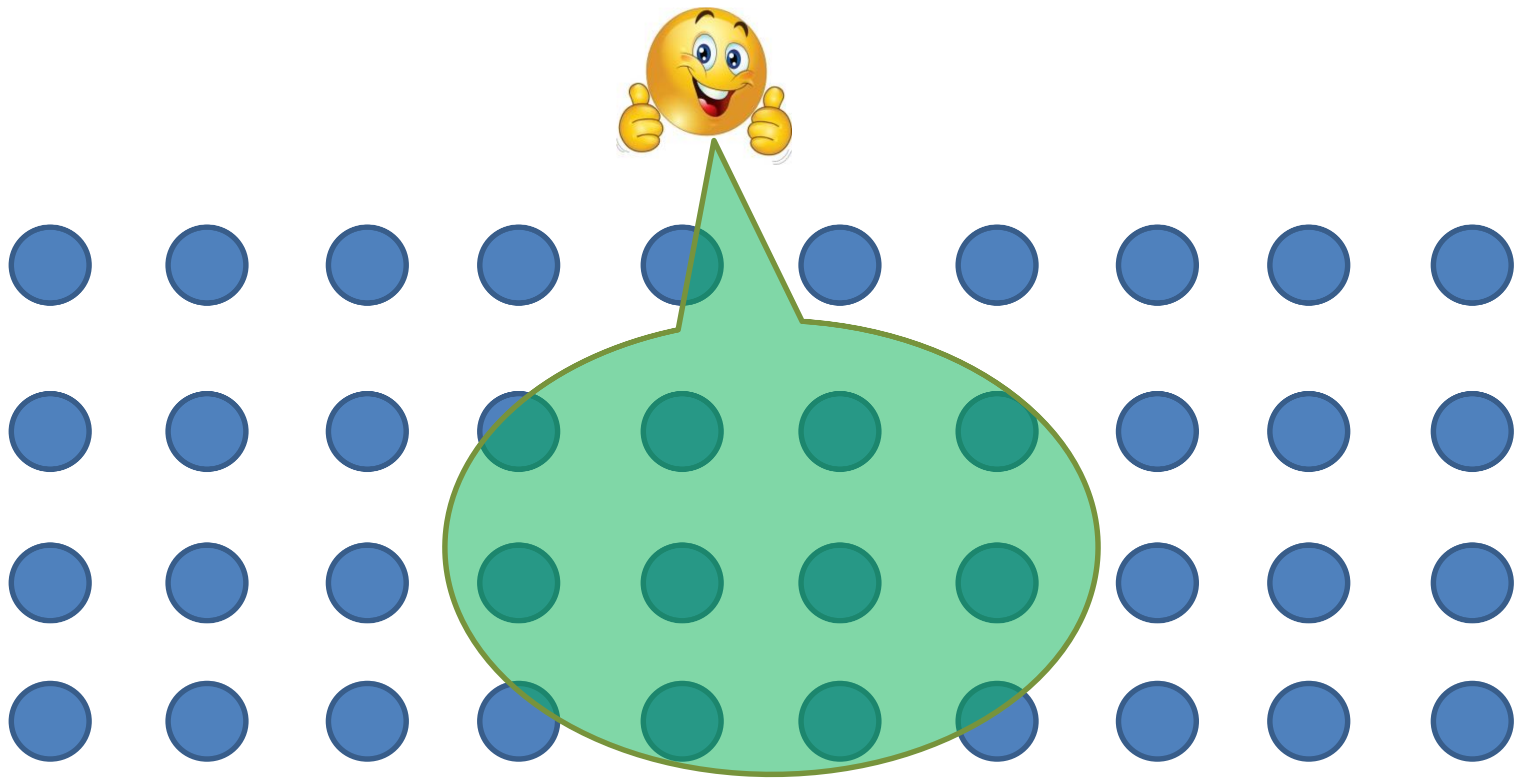
- If $\exists j \in [n']$ that doesn't open the commitment
  - Output $(\perp, j)$
- If all commitments open
  - Reconstruct $x_1, \dots, x_n$
  - Output $y = f(x_1, \dots, x_n)$

Lower Bound

The Setting (1)

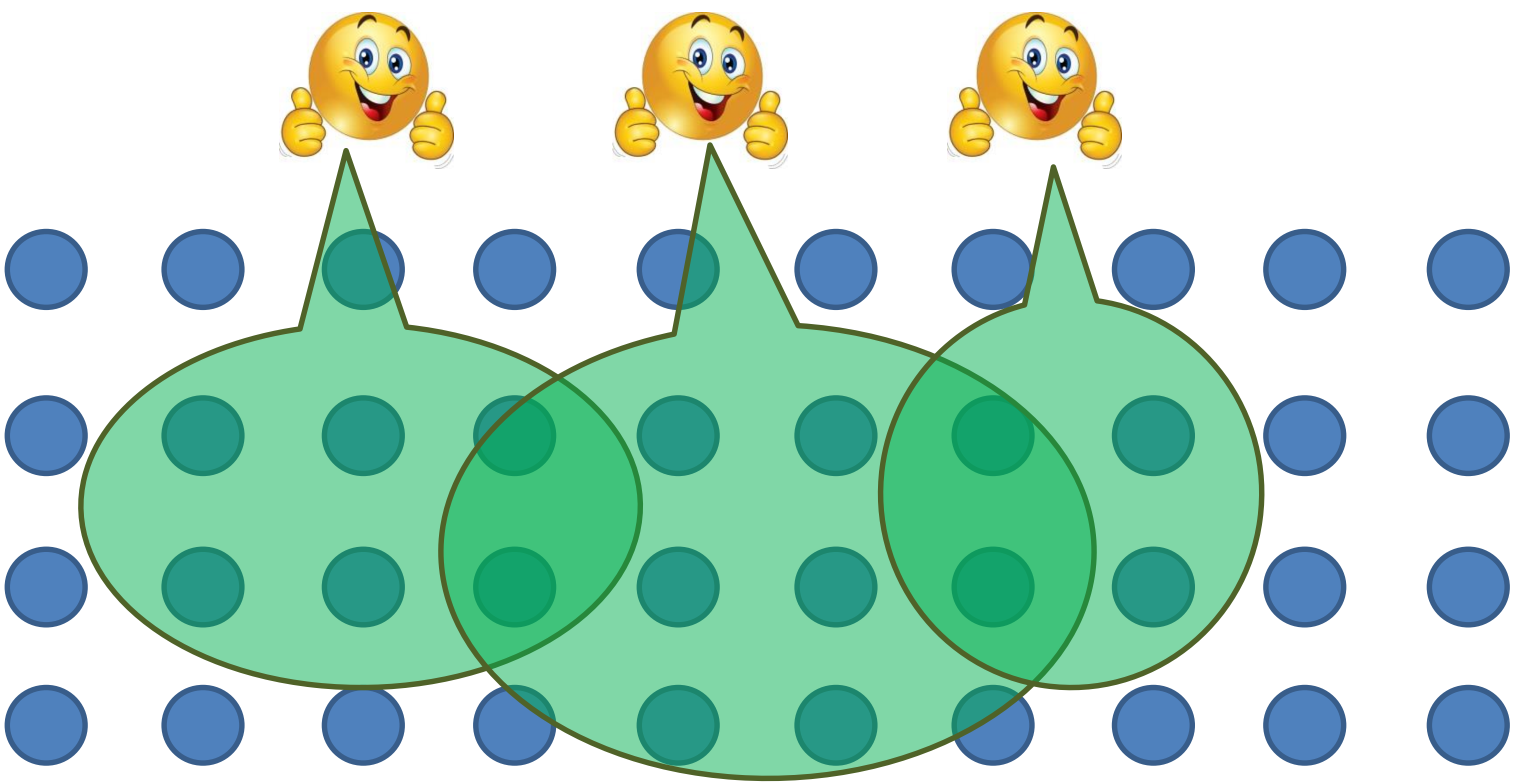
Fully secure coin-flipping protocol

Hybrid: a TTP computes CF with fairness and restricted id-abort, for any $\mathcal{C} \subseteq [n]$



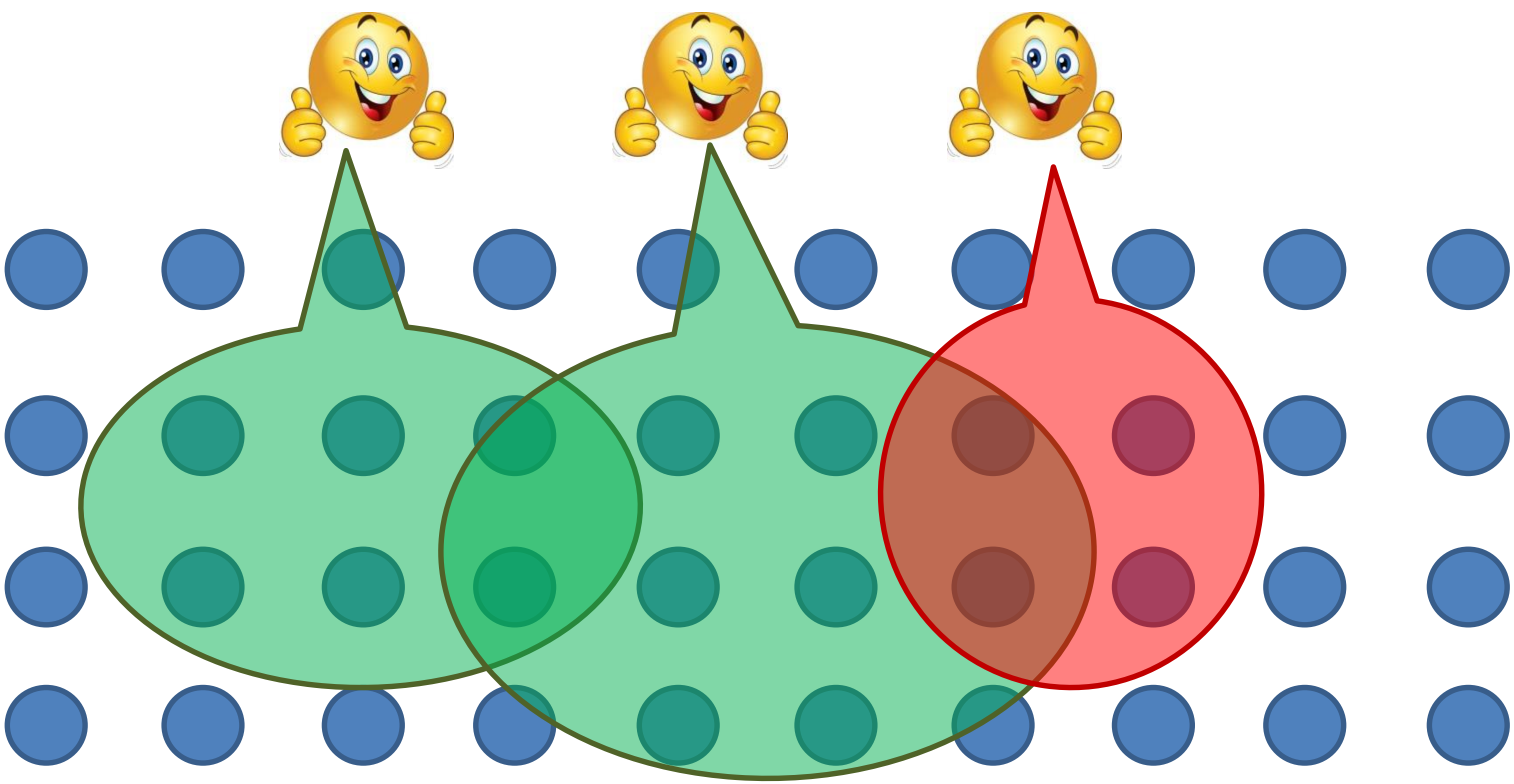
The Setting (2)

Parallel calls: parties can invoke TTP in parallel for different committees $\mathcal{C}_1, \ldots, \mathcal{C}_\ell \subseteq [n]$ at the same functionality round



The Setting (3)

Rushing: if $\exists \mathcal{C}_i$ that is fully corrupted, $\mathcal{A}$ decides to abort $\mathcal{C}_i$ after seeing the output of all other computations in the round



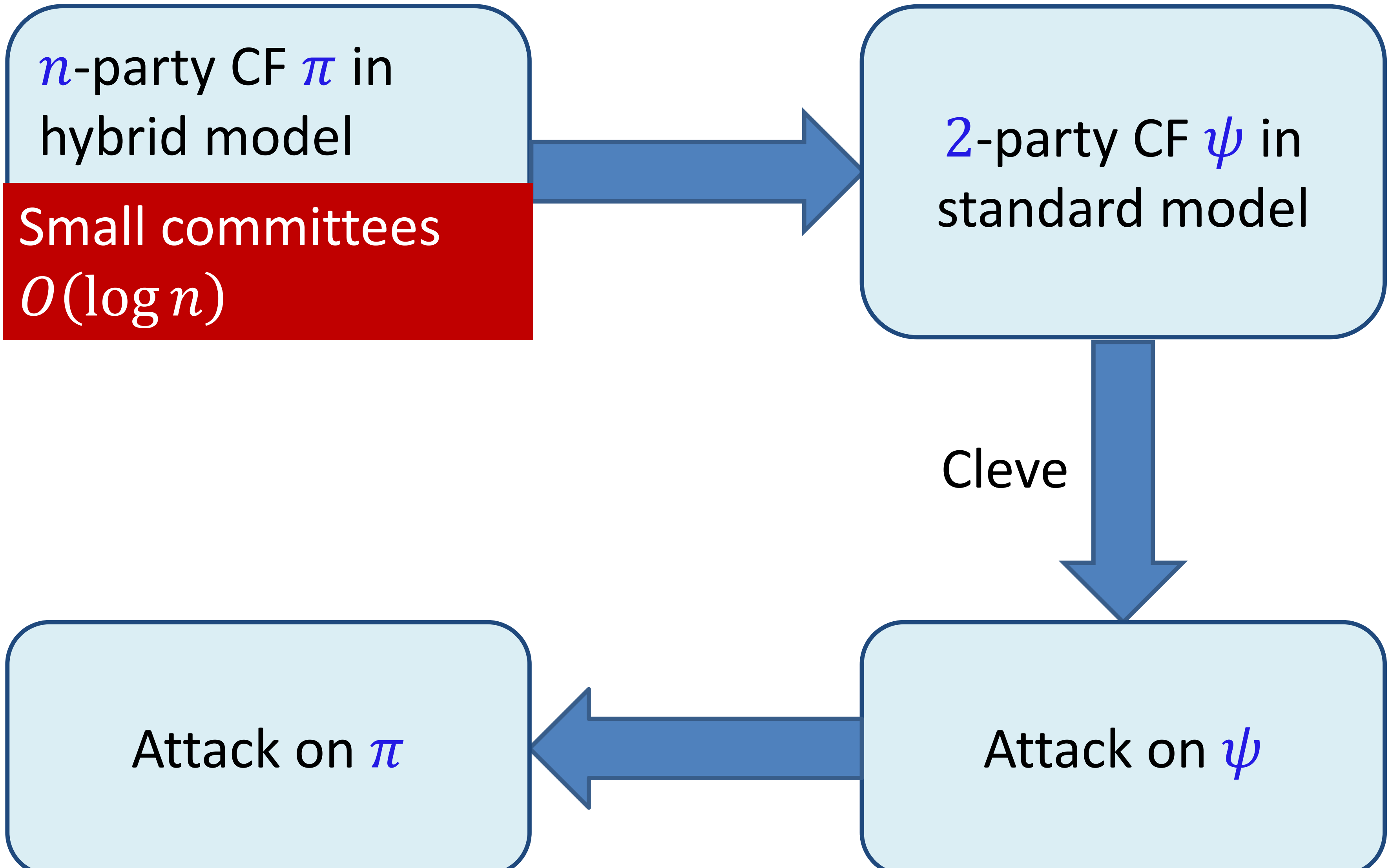
Thm 3: The Lower Bound

Let π be a coin-flipping with a constant number of functionality rounds, and let $1/2 < \beta < 1$

Then, $\exists$ PPT fail-stop adversary that by corrupting $\beta \cdot n$ parties, can bias the output of π

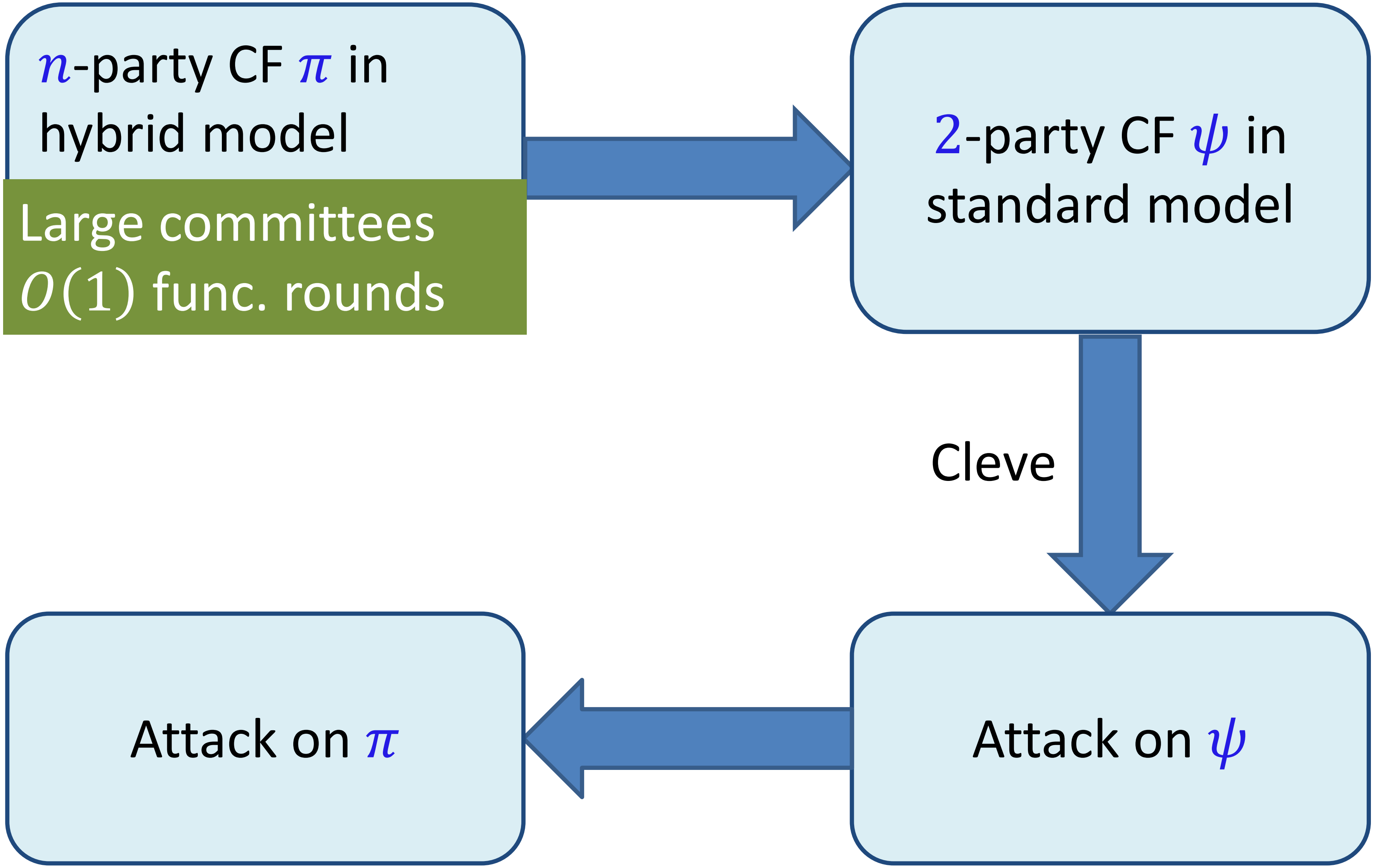
Thm 1: $\exists$ CF in this model (using $\omega(\log n)$ rounds)

Proof Idea



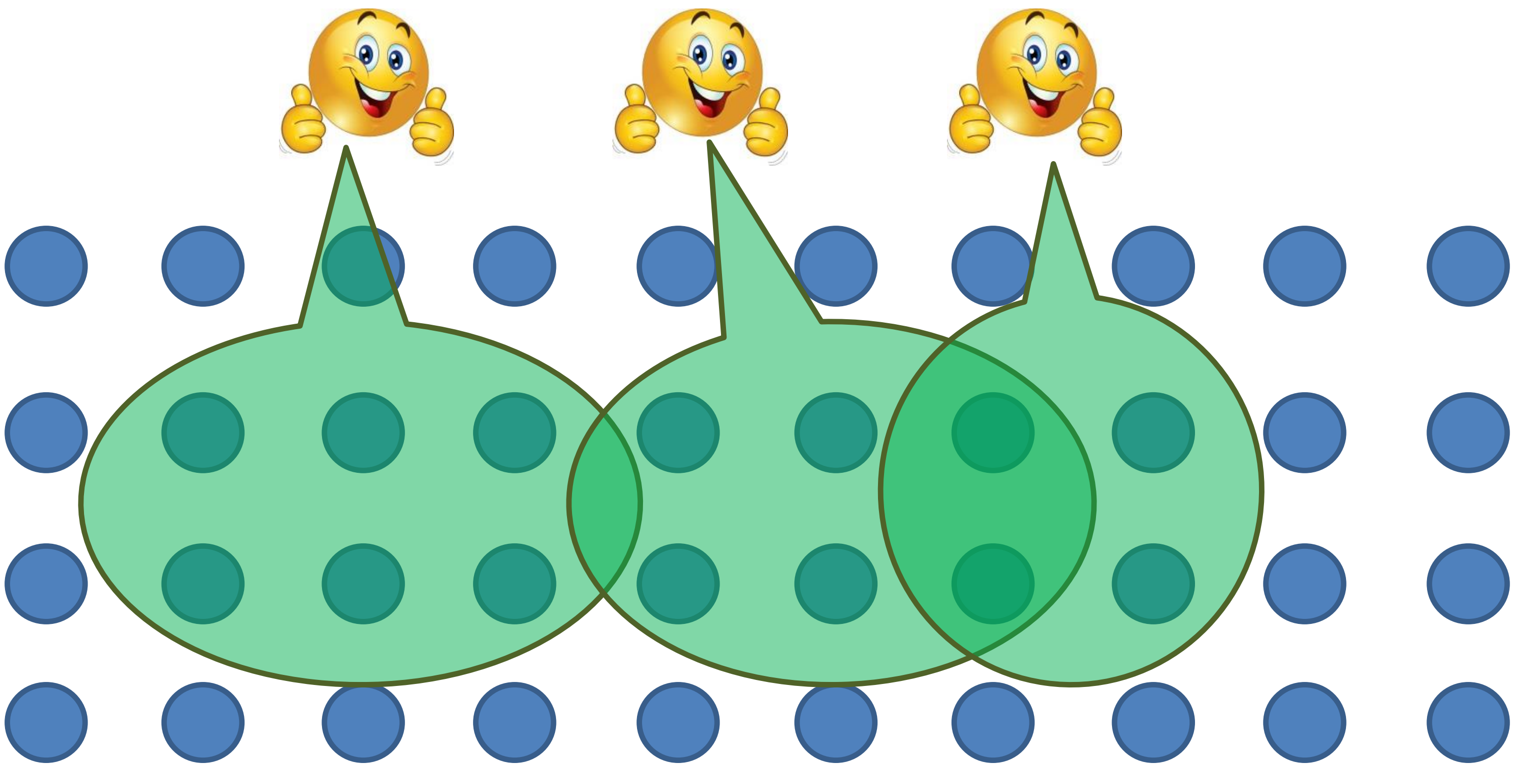
n-party CF π in hybrid model
Small committees $O(\log n)$
2-party CF ψ in standard model
Cleve
Attack on ψ
Attack on π

Proof Idea


n-party CF π in hybrid model
Large committees
$O(1)$ func. rounds
2-party CF ψ in standard model
Cleve
Attack on ψ
Attack on π

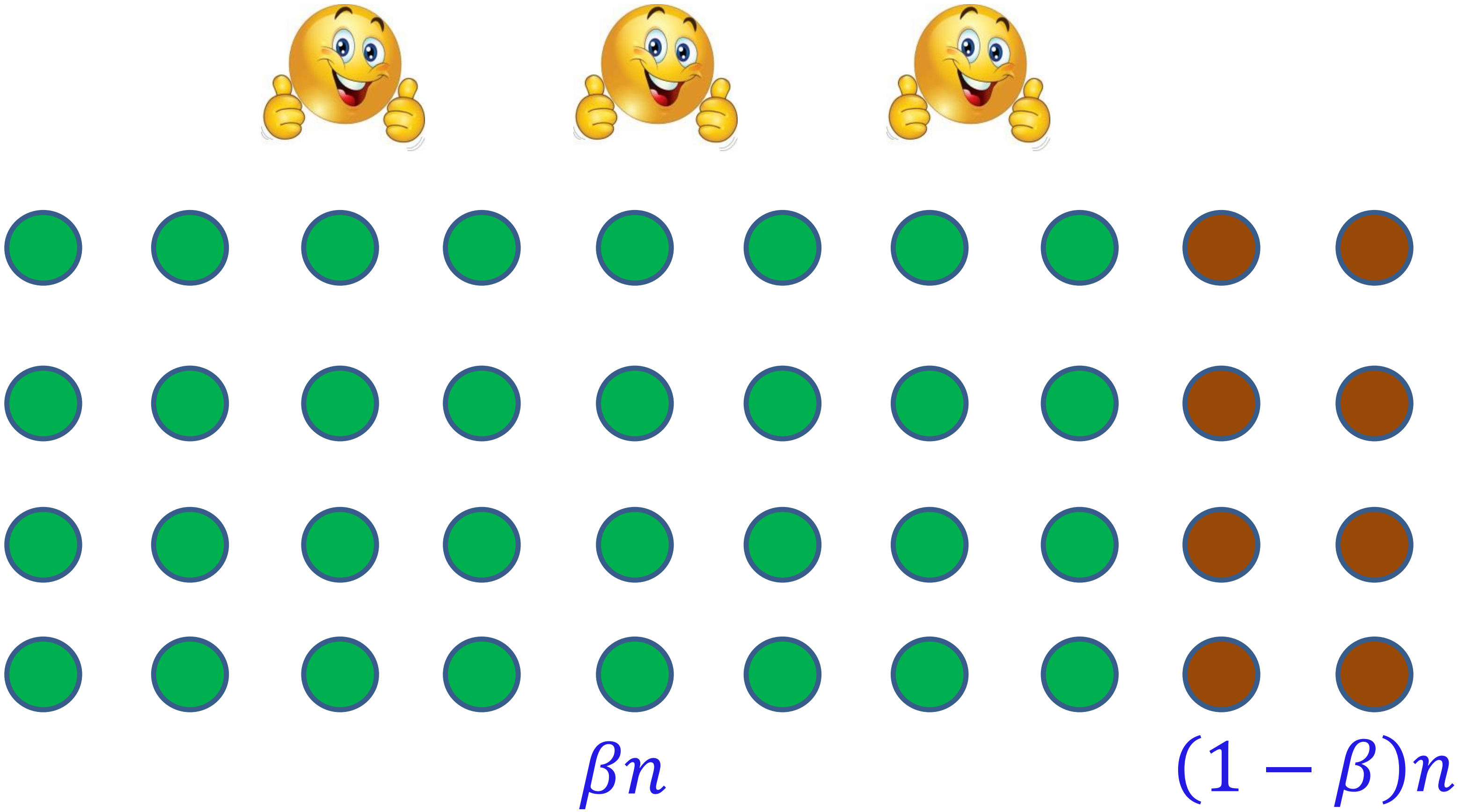
Case I : No Large Committees

All committees have size at most $c \cdot \log n$



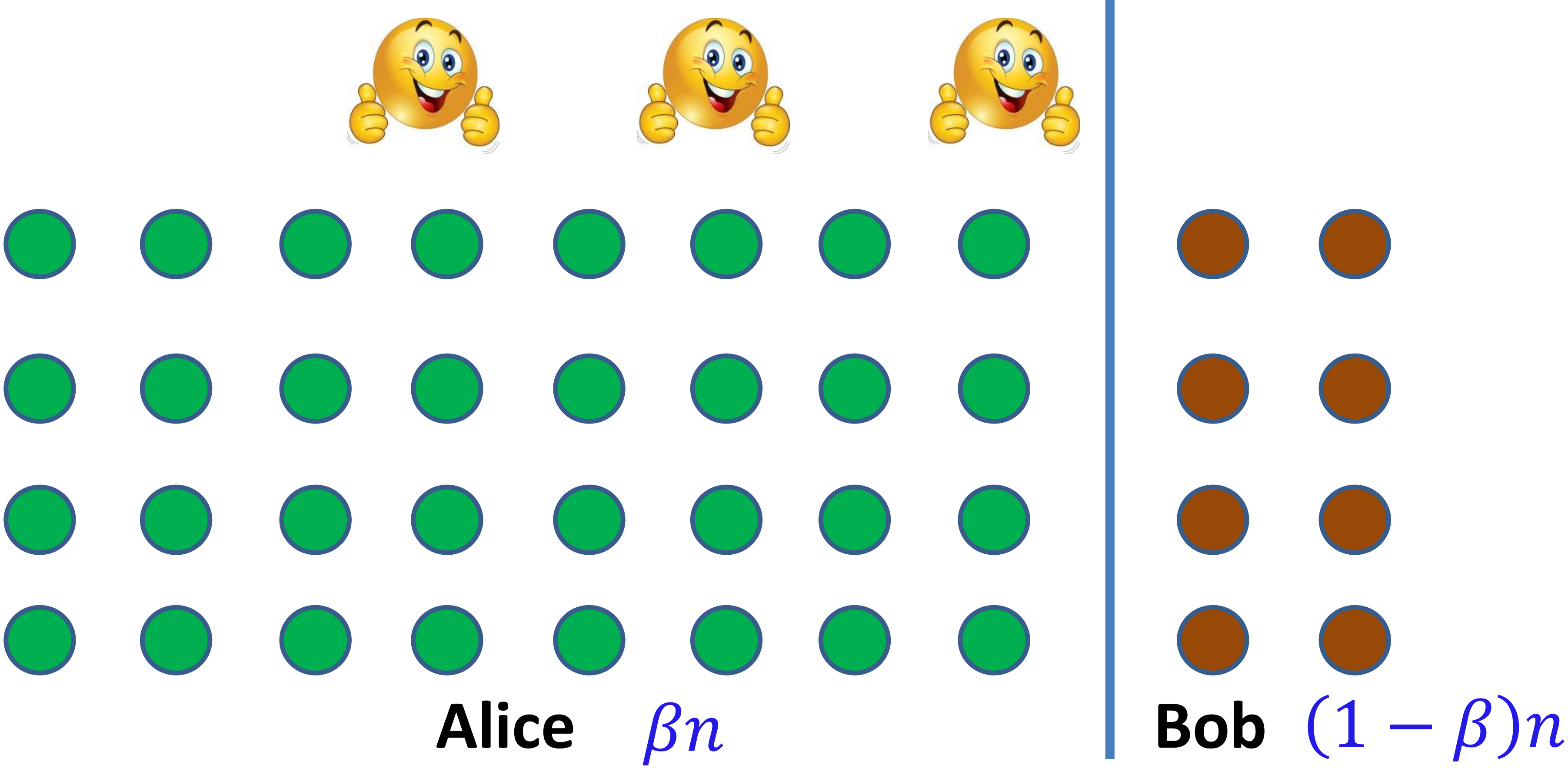
Case I : 2-Party Coin Flipping

- Split the parties to 2 sets



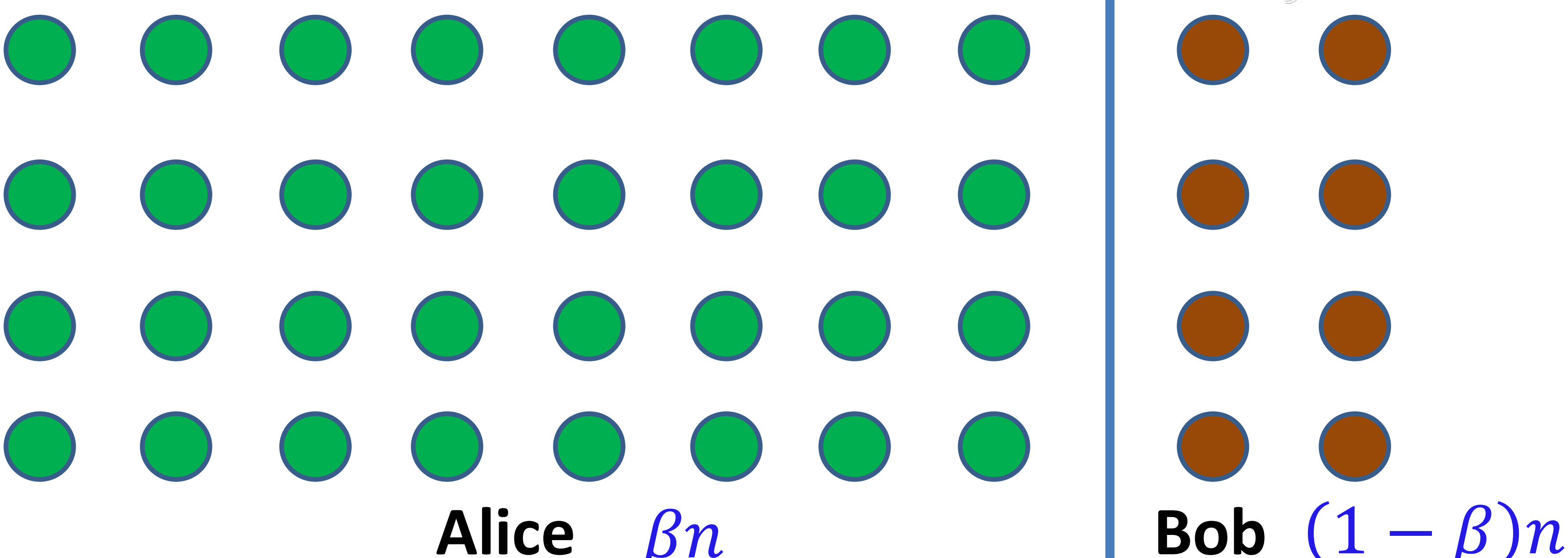
Case I : 2-Party Coin Flipping

- Split the parties to 2 sets
- Alice controls one set, Bob the other



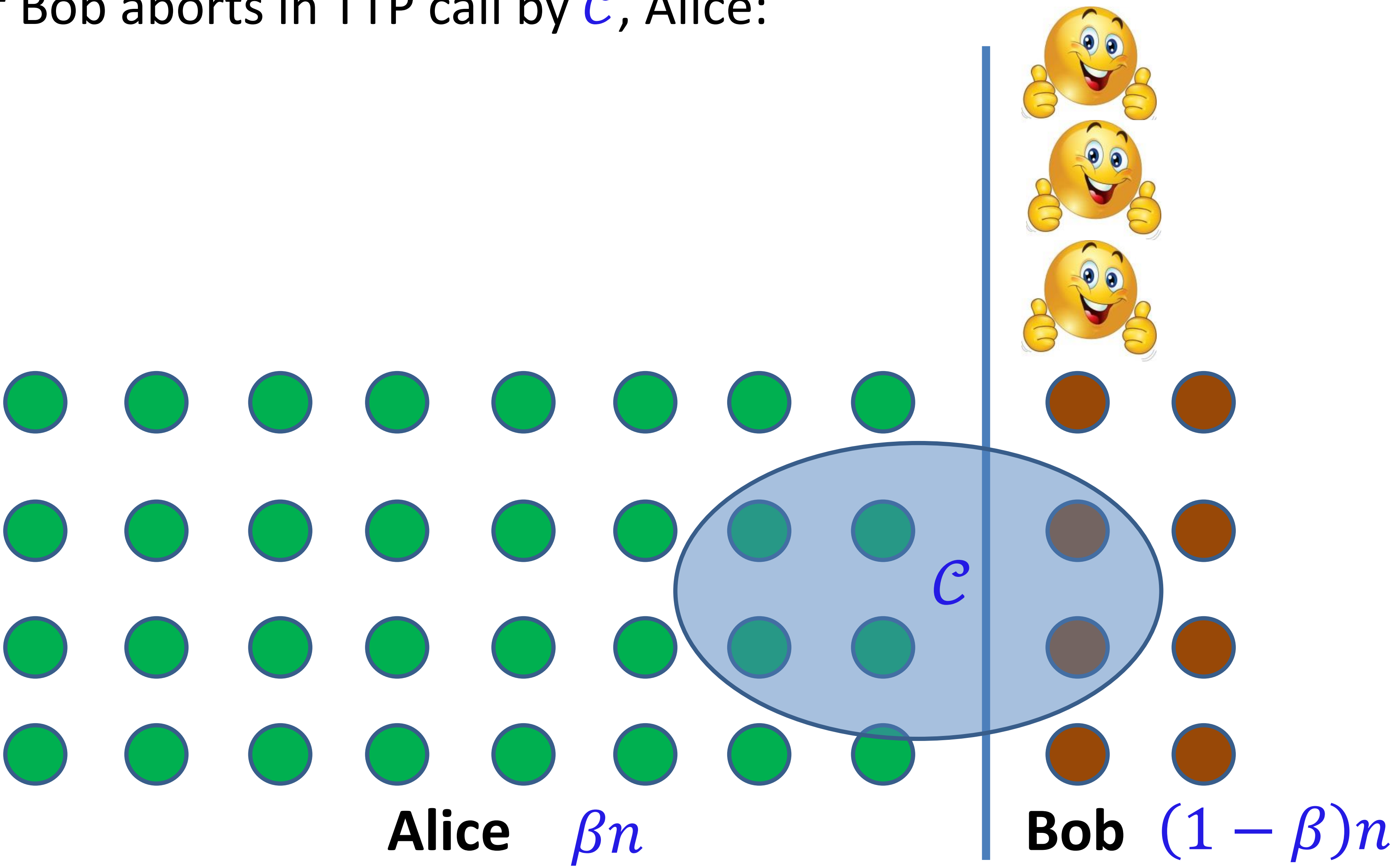
Case I : 2-Party Coin Flipping

- Split the parties to 2 sets
- Alice controls one set, Bob the other
- Bob controls trusted party



Case I : Dealing with Abort

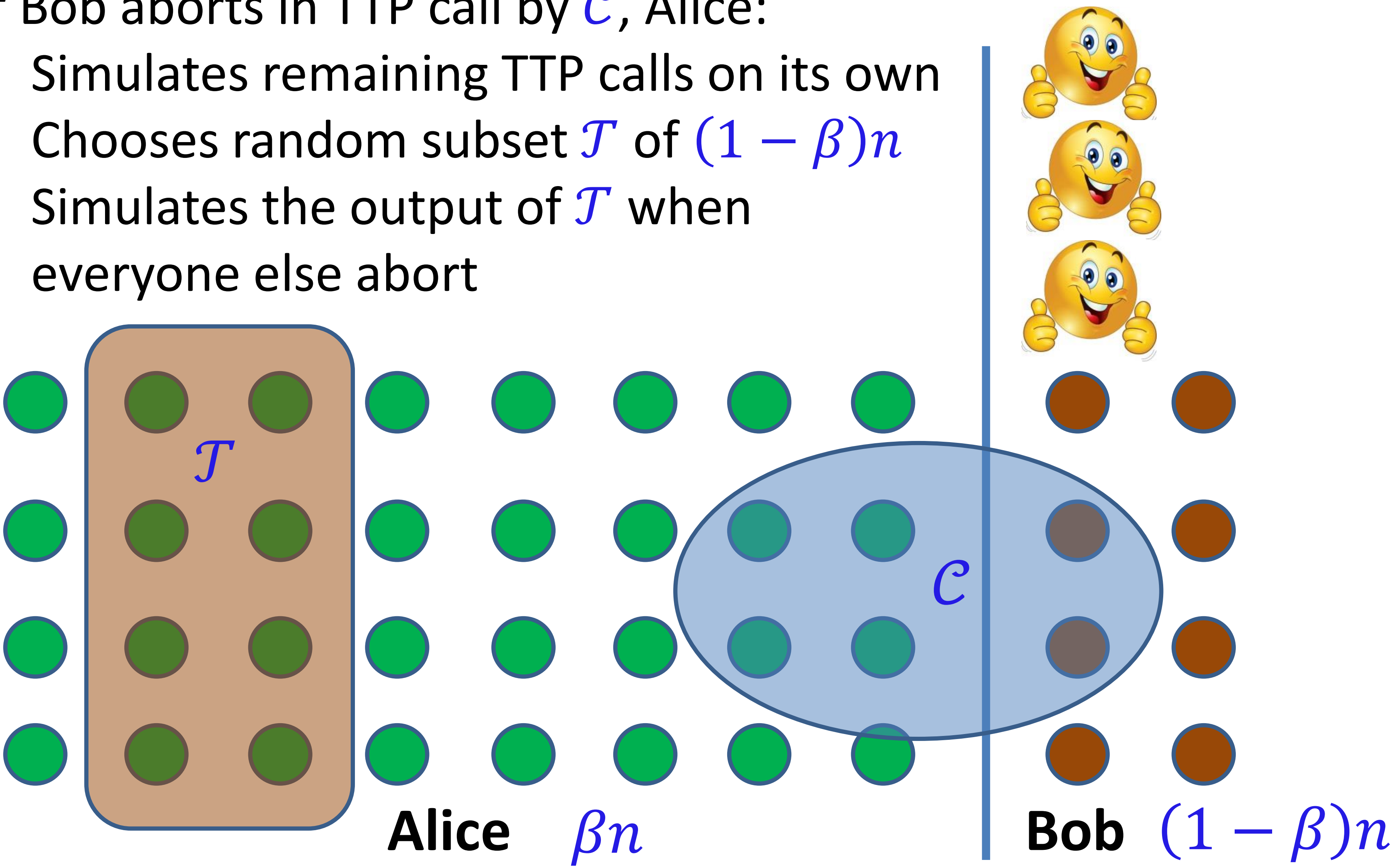
If Bob aborts in TTP call by $\mathcal{C}$, Alice:



Case I : Dealing with Abort

If Bob aborts in TTP call by $\mathcal{C}$, Alice:

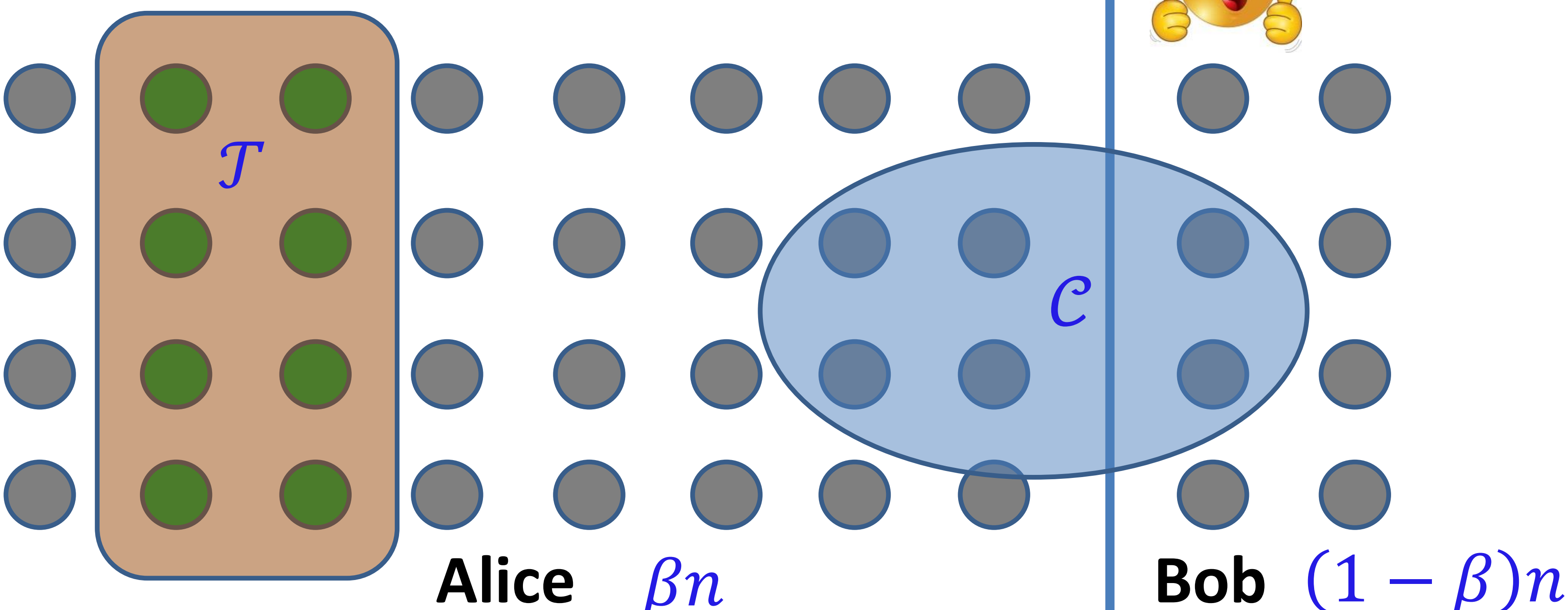
- Simulates remaining TTP calls on its own
- Chooses random subset $\mathcal{T}$ of $(1-\beta)n$
- Simulates the output of $\mathcal{T}$ when everyone else abort



Case I : Dealing with Abort

If Bob aborts in TTP call by $\mathcal{C}$, Alice:

- Simulates remaining TTP calls on its own
- Chooses random subset $\mathcal{T}$ of $(1-\beta)n$
- Simulates the output of $\mathcal{T}$ when everyone else abort

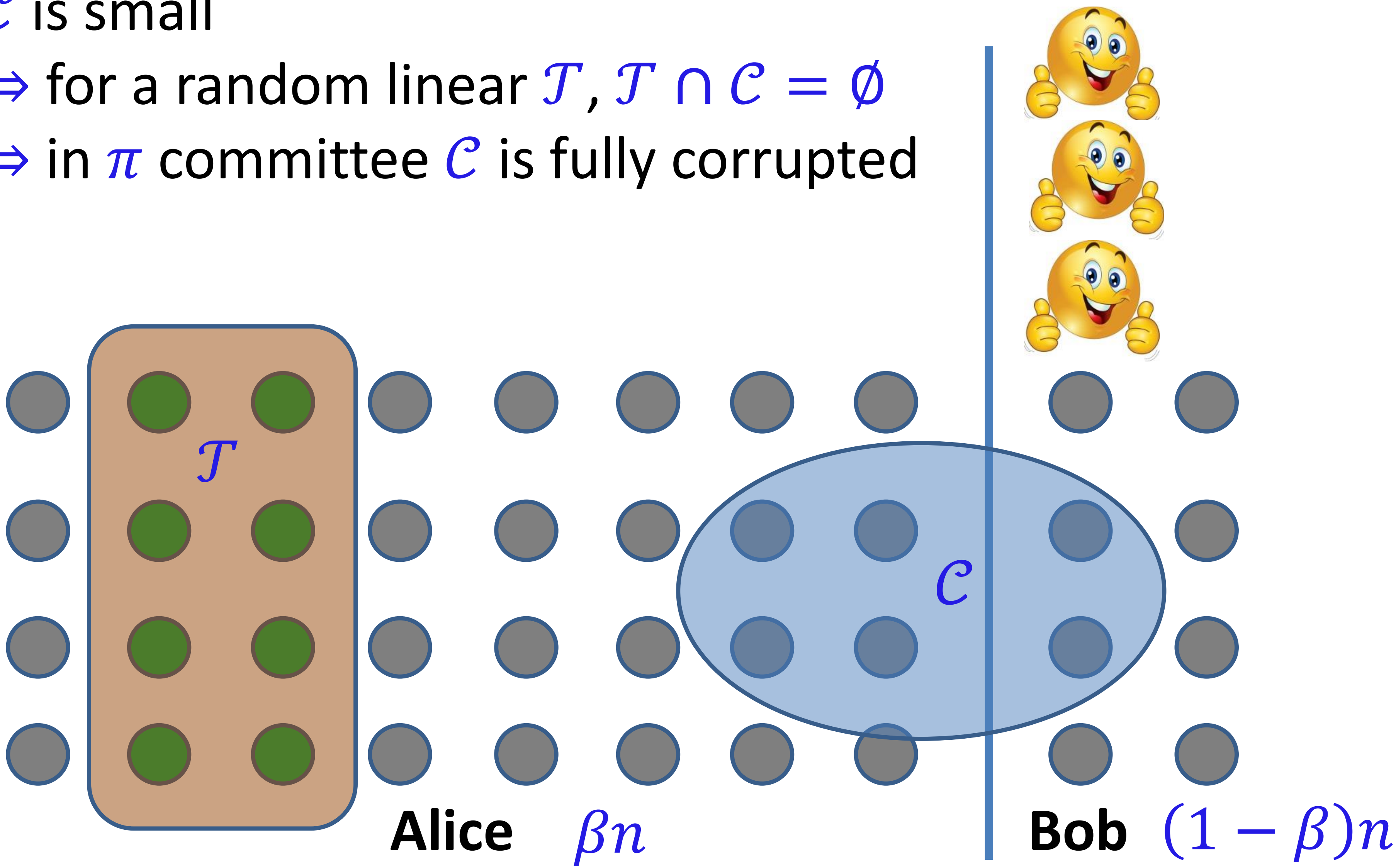


Case I : Dealing with Abort

$\mathcal{C}$ is small

$\Rightarrow$ for a random linear $\mathcal{T}$, $\mathcal{T} \cap \mathcal{C} = \emptyset$

$\Rightarrow$ in π committee $\mathcal{C}$ is fully corrupted



Case II : Arbitrary Committees

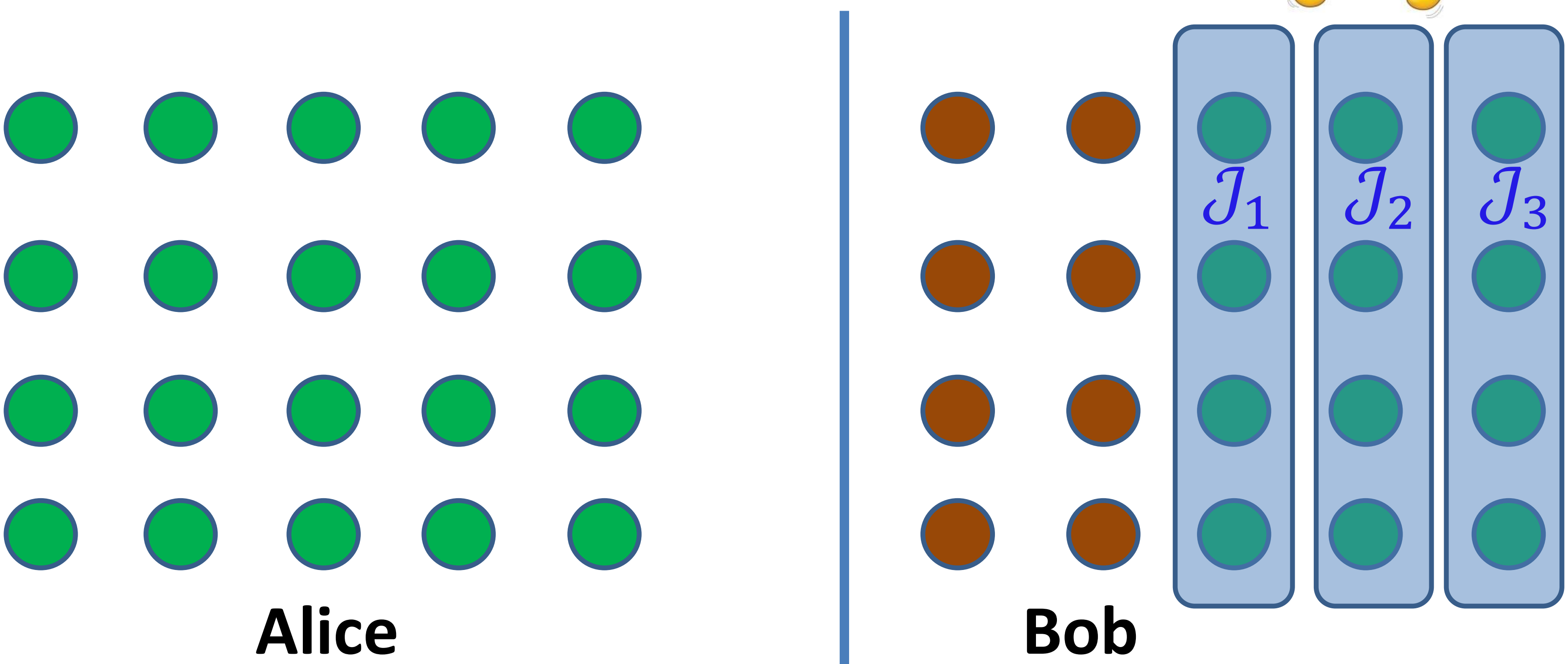
Main idea:

- The adversary aborts all large committees
- Reduces to the no-large committees case

- For random disjoint linear subsets $\mathcal{J}_1, \ldots, \mathcal{J}_s$, all large committees in round i intersect $\mathcal{J}_i$ (whp)
- The adversary has "budget" only for a constant number of rounds

Case II : 2-Party Coin Flipping

- Bob controls the subsets $\mathcal{J}_1, \dots, \mathcal{J}_s$
- Emulates TTP in the i'th round only for committees $\mathcal{C}$ s.t. $\mathcal{C} \cap \mathcal{J}_i = \emptyset$



Summary

What did we see

- Fair to Full, $t = \beta n$, no input, $\omega(\log n)$
- Fair to Full, $t = \beta n$, with input, $\omega(1)$
- No Fair to Full coin flipping, $t = \beta n$, $O(1)$

What didn't we see

- Fair to Full, $t = \beta n$, HM, $\omega(1)$ - BB & info-theoretic
- Abort to Full, $t = O(\sqrt{n})$, no identifiability

What's open

- No input, gap between feasibility $\omega(\log n)$ and lower bound $O(1)$

Thank You