Asymptotically Optimal Adaptive Asynchronous Common Coin and DKG with Silent Setup

Hanwen Feng
hanwen.feng@sydney.edu.au

Qiang Tang
qiang.tang@sydney.edu.au

THE UNIVERSITY OF SYDNEY

-- The Problem

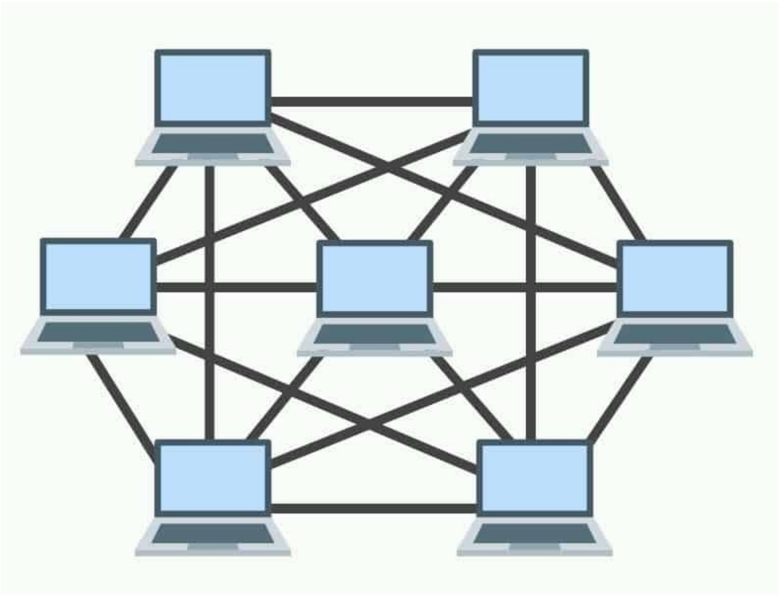
-- The Challenges

-- Our Contributions

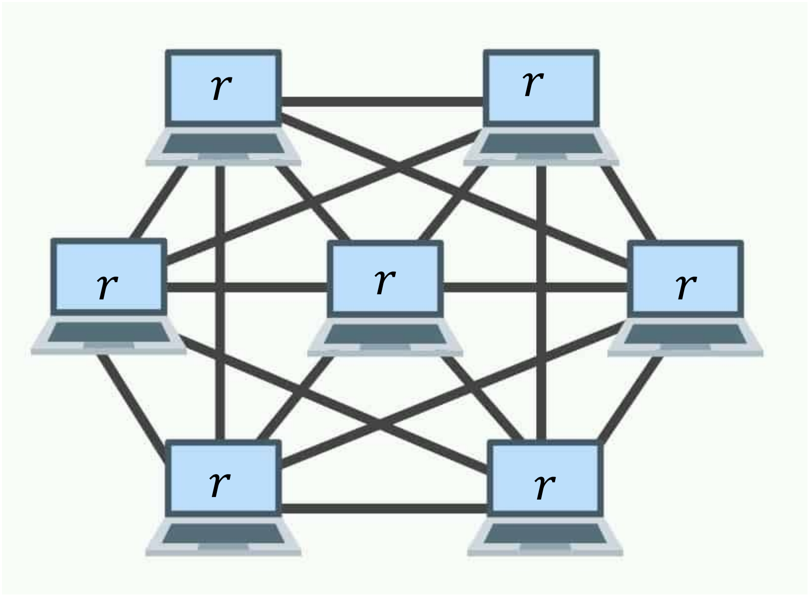
---- Asymptotically Optimal Construction

---- New Framework for Analyzing Specific Composition

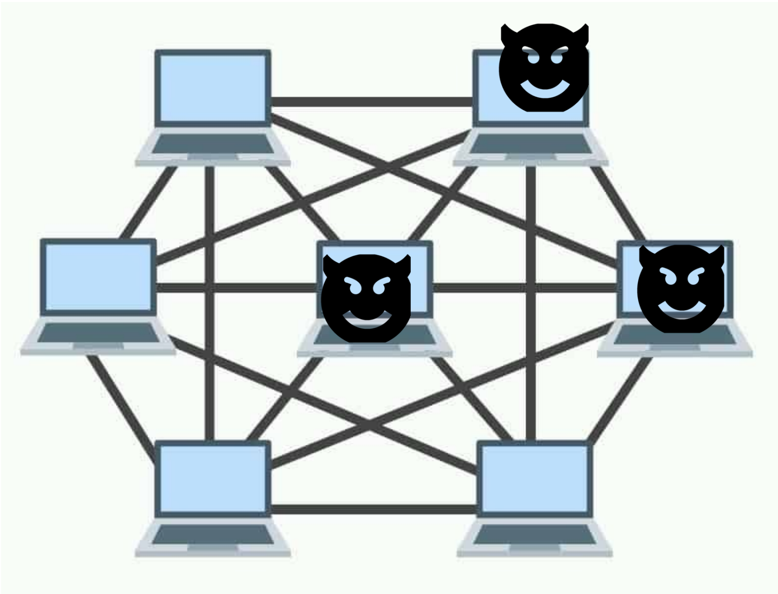
Common Coin Protocol



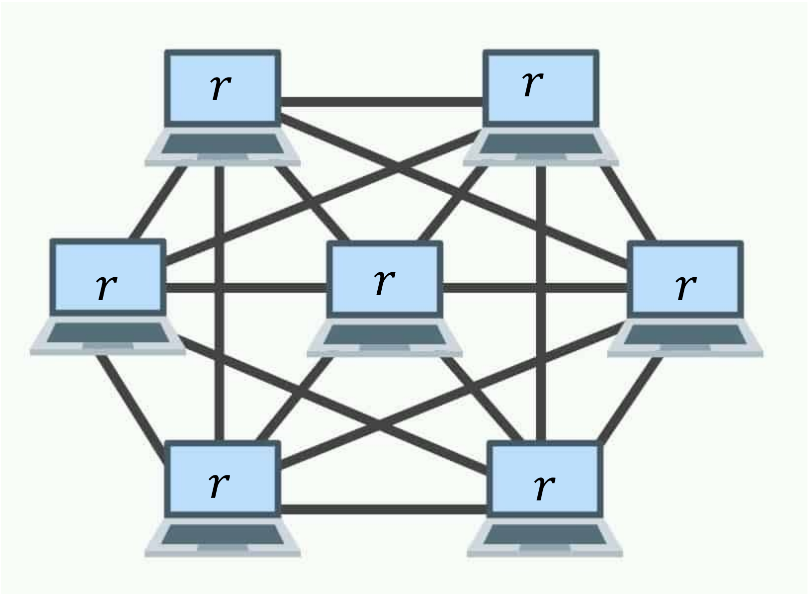
Common coin protocol



Common Coin Protocol



Common coin protocol



When at most t out of n nodes are corrupted,

Termination All nodes can output a value

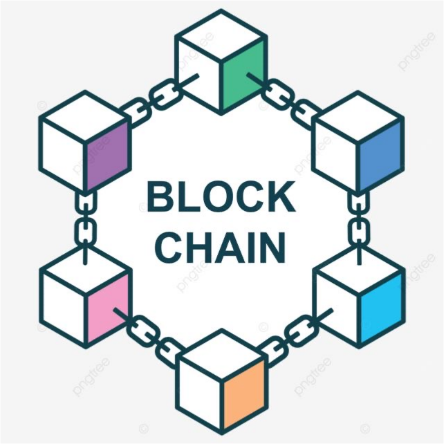
Agreement All nodes output the same value

Unpredictable No one knows r in advance

Bias-resistance r is nearly uniformly distributed

Where it matters

Blockchains



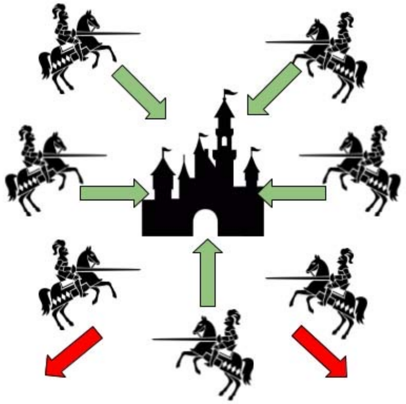
TorProject.org

Anonymous communication

Gambling

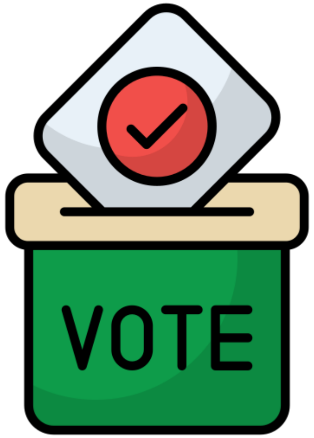
...

Byzantine
Consensus



Problem: 5 generals say to "attack" but 2 are traitors and say to "retreat."

Voting
Sortition
Audition



Lotteries

All applications where randomness is a public interest

Secure Common Coin Protocols Are Expensive

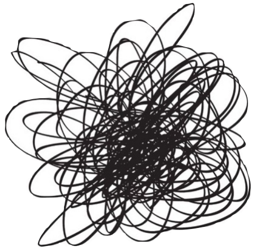
Network Model

Fully Connected Network: Every pair of nodes is connected via an authenticated channel.

Asynchronous
Network

$send\ m$ → Eventually receive m

- ✓ Better capture the real network
- ✓ Asynchronous protocols are easier to implement

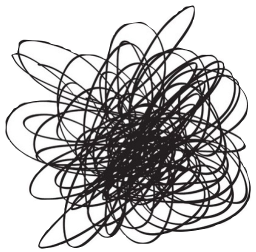


Early theory results

Information-theoretical Setting

$O(n^7)$ communication complexity

n is the network size



Early theory results
Information-theoretical Setting
$O(n^7)$ communication complexity
n is the network size

Resurging interest due to blockchain applications

Kokoris-Kogias et al., CCS'20

Computational Setting
$O(n^4)$ communication complexity
$O(n)$ round complexity

Abraham et al. PODC'21, CRYPTO'23

Freitas et al. DISC'22

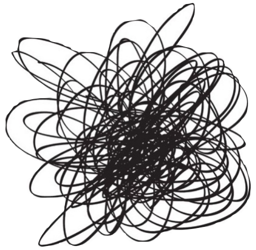
Gao et al. ICDCS '22

Bandarupalli et al. CCS'24

Das et al. CCS '21

Das et al. CCS '24

They all require $\Omega(n^3)$ communication complexity
$\Omega(1)$ round complexity



Early theory results
Information-theoretical Setting
$O(n^7)$ communication complexity
n is the network size

Resurging interest due to blockchain applications

Kokoris-Kogias et al., CCS'20

Computational Setting
$O(n^4)$ communication complexity
$O(n)$ round complexity

Abraham et al. PODC'21, CRYPTO'23

Freitas et al. DISC'22

Gao et al. ICDCS '22

Bandarupalli et al. CCS'24

Das et al. CCS '21

Das et al. CCS '24

They all require $\Omega(n^3)$ communication complexity
$\Omega(1)$ round complexity

Ideally, the communication cost of adaptively secure protocols can be as small as $O(n^2)$

Can we design an asynchronous common coin protocol with optimal communication complexity?

At the same time, preserve other optimal metrics:
(1) $O(1)$ rounds
(2) Tolerate up to 33% Byzantine nodes (optimal for all asynchronous consensus)
I.e., a network with $n = 3f + 1$ nodes, up to f are corrupted.

-- The Problem

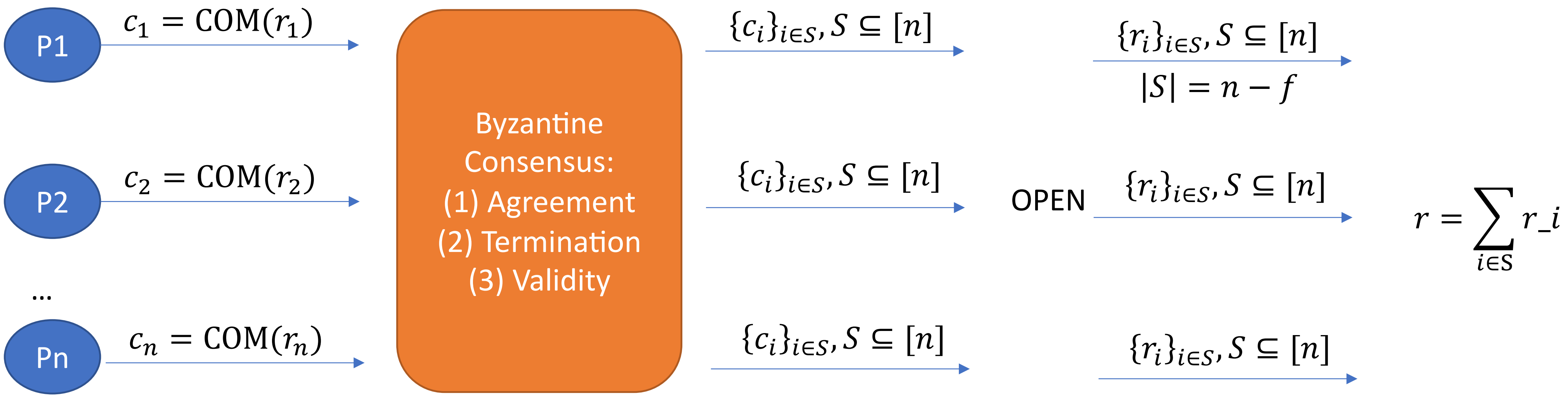
-- The Challenges

-- Our Contributions

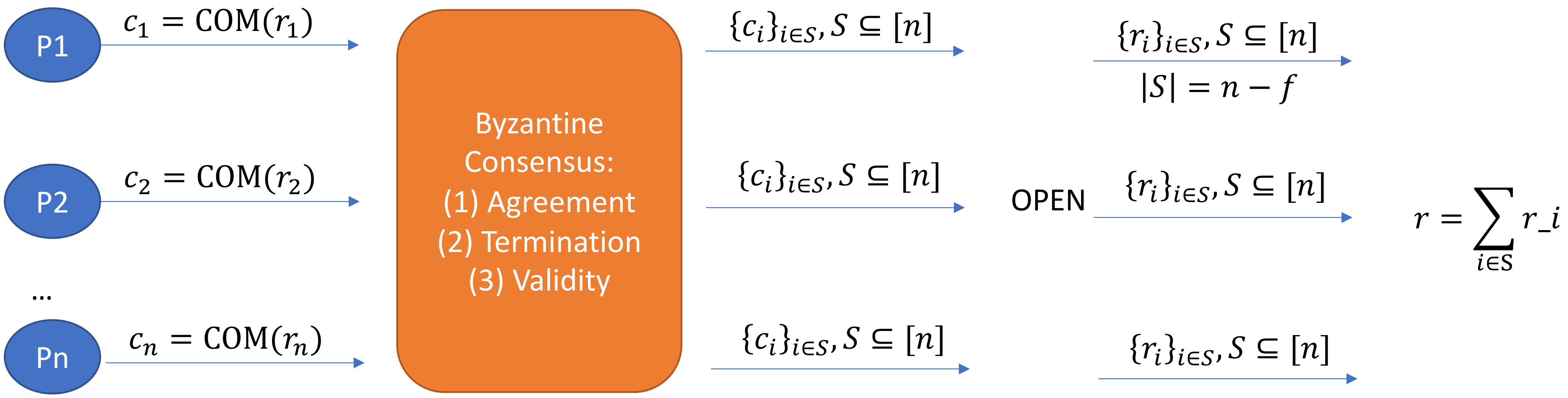
---- Asymptotically Optimal Construction

---- New Framework for Analyzing Specific Composition

Commit-Agree-Reveal Paradigm



Commit-Agree-Reveal Paradigm



Commitment

- Hiding and binding.
- "Forced Opening": The network can open a commitment without the committer
- Examples: Verifiable secret sharing, Timed commitment, etc.

Commitment
with “Forced Opening”

Common Coin

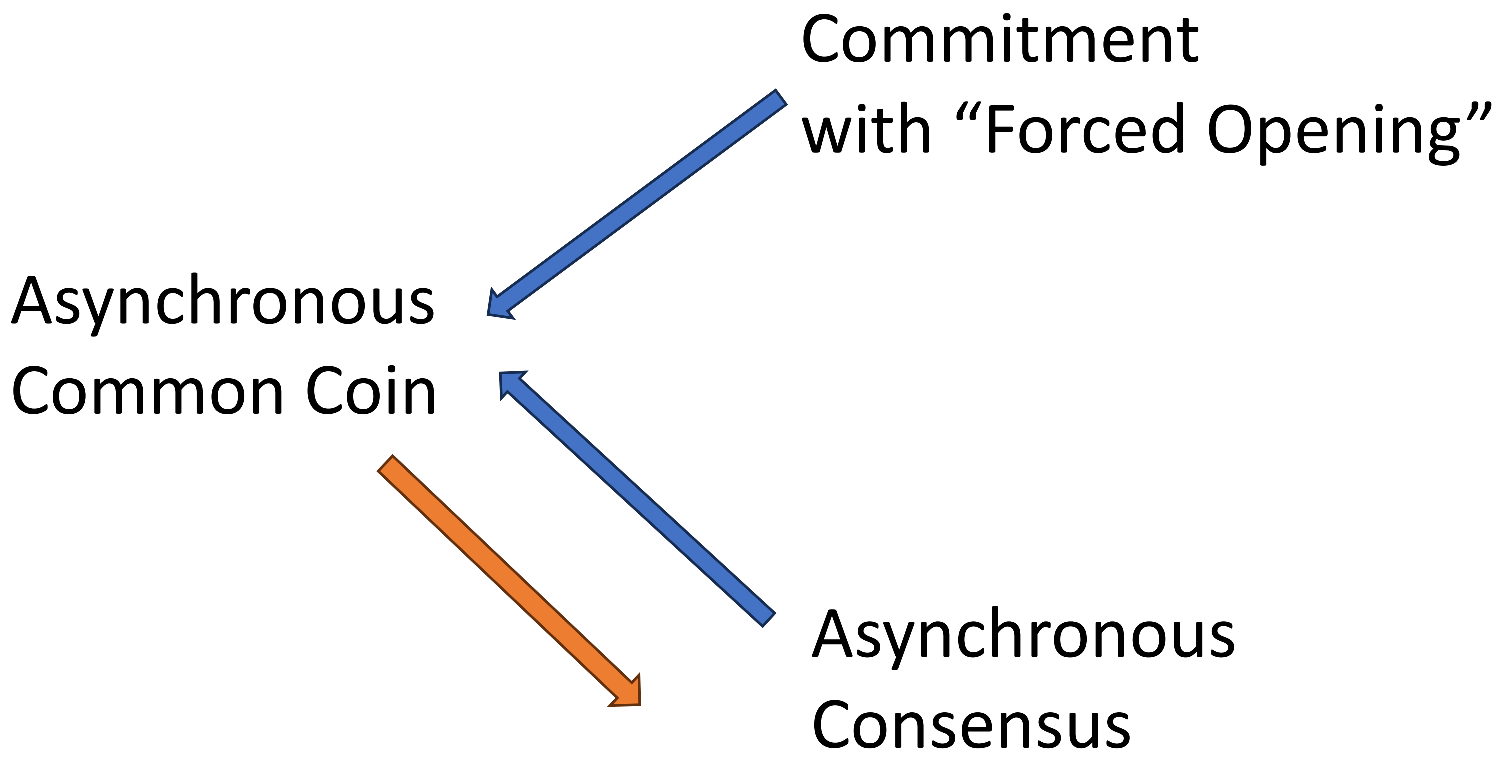
Consensus

Fischer-Lynch-Paterson Impossibility [J’ACM 1985]:

Achieving consensus in an asynchronous network is impossible when at least one node may crash and a deterministic algorithm is used.

The most popular design is to employ a common coin for randomizing the protocol.

Asynchronous Consensus ⟵ Asynchronous Common Coin



We must avoid this circularity

A less standard randomization technique is used:

Asynchronous Consensus ← Asynchronous Common Coin

When at most t out of n nodes are corrupted,

always hold (except with a negligible probability):

Termination

Agreement

Unpredictable

Bias-resistance

A less standard randomization technique is used:

Asynchronous Consensus ← Asynchronous Common Coin

When at most t out of n nodes are corrupted, always hold (except with a negligible probability):

Termination

Agreement

Unpredictable

Bias-resistance

Asynchronous Consensus ← Asynchronous Weak Common Coin

✓ Termination

Only hold with a constant probability $0 < \phi < 1$ — Agreement, Unpredictable, Bias-resistance

Agreement

Unpredictable

Bias-resistance

Commitment
with "Forced Opening"

Asynchronous
Consensus

Asynchronous
Weak
Common Coin

When at most t out of n nodes are corrupted,

- ✓ Termination
- Agreement
- Unpredictable
- Bias-resistance

Only hold with a constant probability $0 < \phi < 1$ (applies to Agreement, Unpredictable, Bias-resistance)

Information Gather
(A weak form
of consensus)

Can be deterministic!
So we break the circularity.

(2) Requires $O(n^3)$ communication cost

Commitment with "Forced Opening"

(3) Requires $O(n^3)$ communication cost

Asynchronous Consensus

Asynchronous Weak Common Coin

(1) Requires $O(n^3)$ communication cost

Information Gather (A weak form of consensus)

All existing asynchronous common coin protocols require $O(n^3)$ communication costs

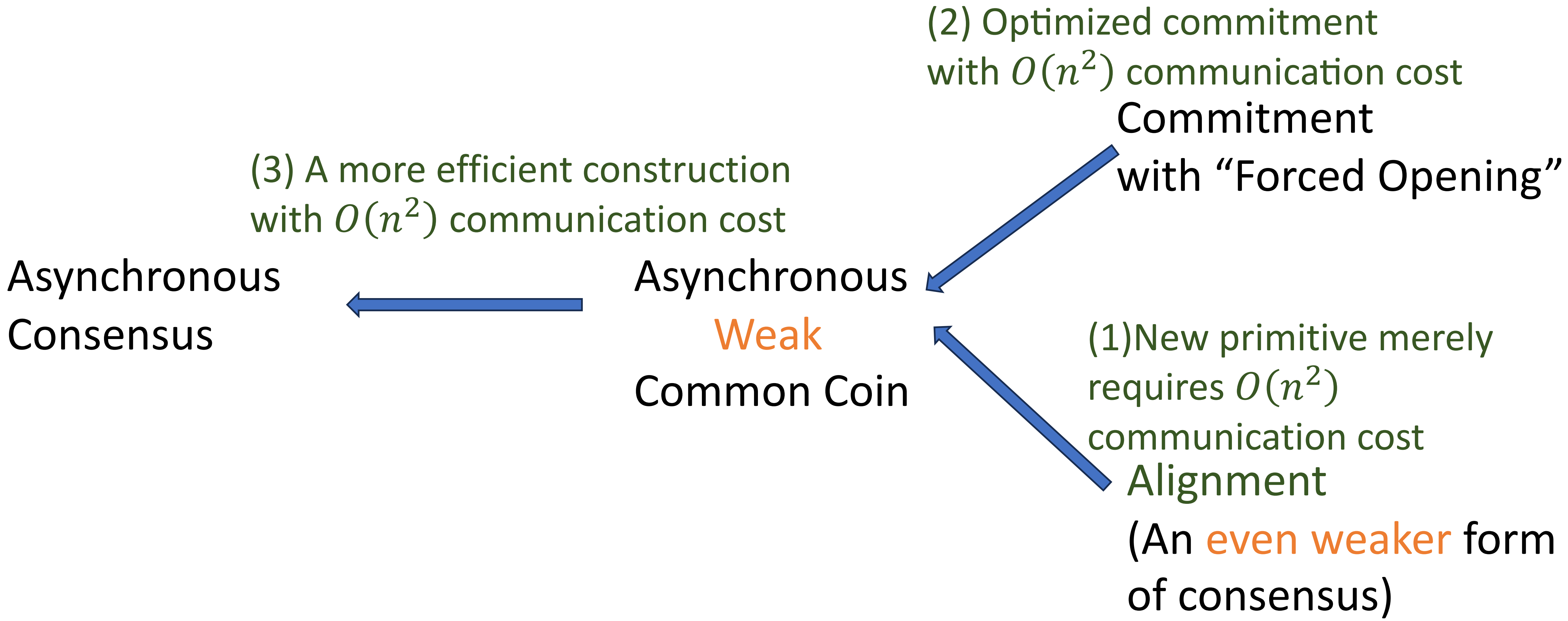
-- The Problem

-- The Challenges

-- Our Contributions

---- Asymptotically Optimal Construction

---- New Framework for Analyzing Specific Composition



We achieve $O(n^2)$ communication complexity by conquering all three parts of challenges

(2) Optimized commitment with $O(n^2)$ communication cost

Commitment with “Forced Opening”

(3) A more efficient construction with $O(n^2)$ communication cost

Asynchronous Consensus

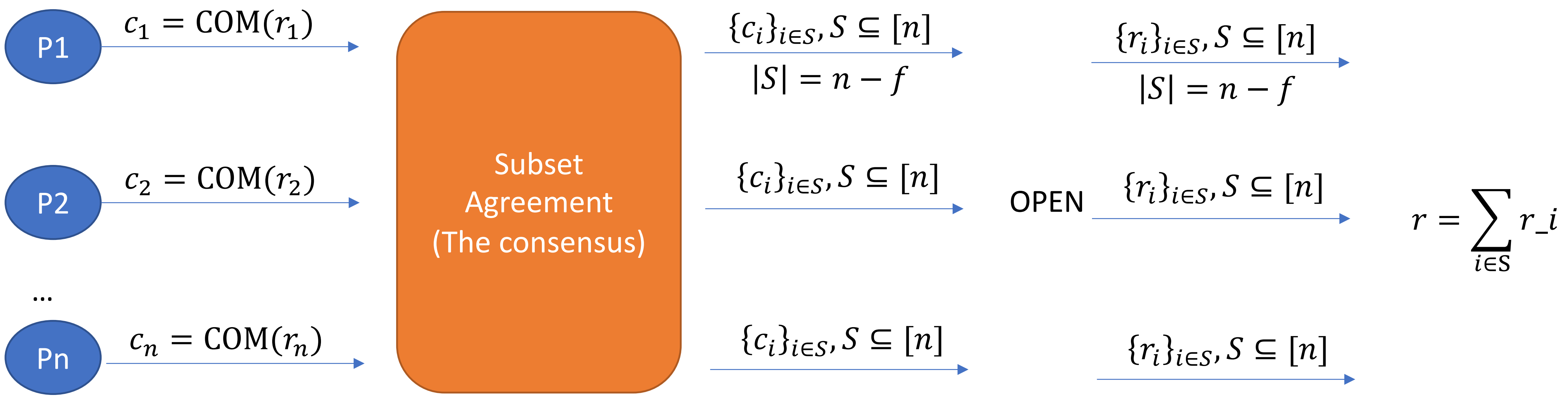
Asynchronous Weak Common Coin

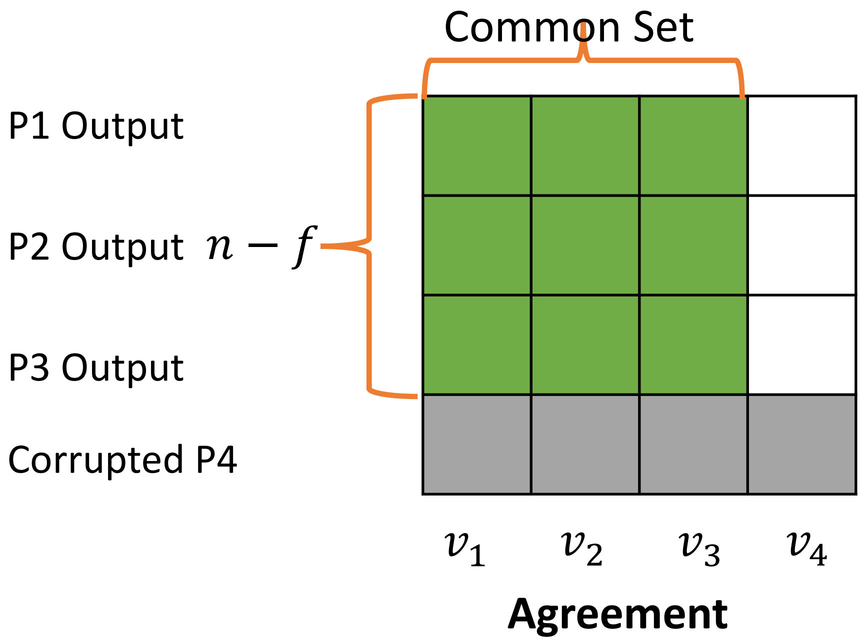
(1)New primitive merely requires $O(n^2)$ communication cost

Alignment (An even weaker form of consensus)

We achieve $O(n^2)$ communication complexity by conquering all three parts of challenges

Asynchronous Common Coin

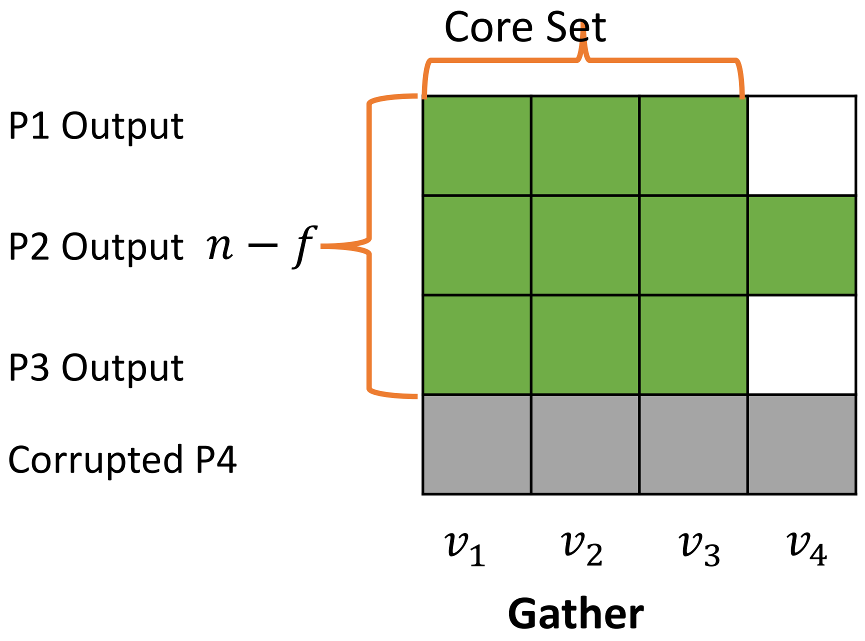




f is the maximum number of nodes an adversary can corrupt.

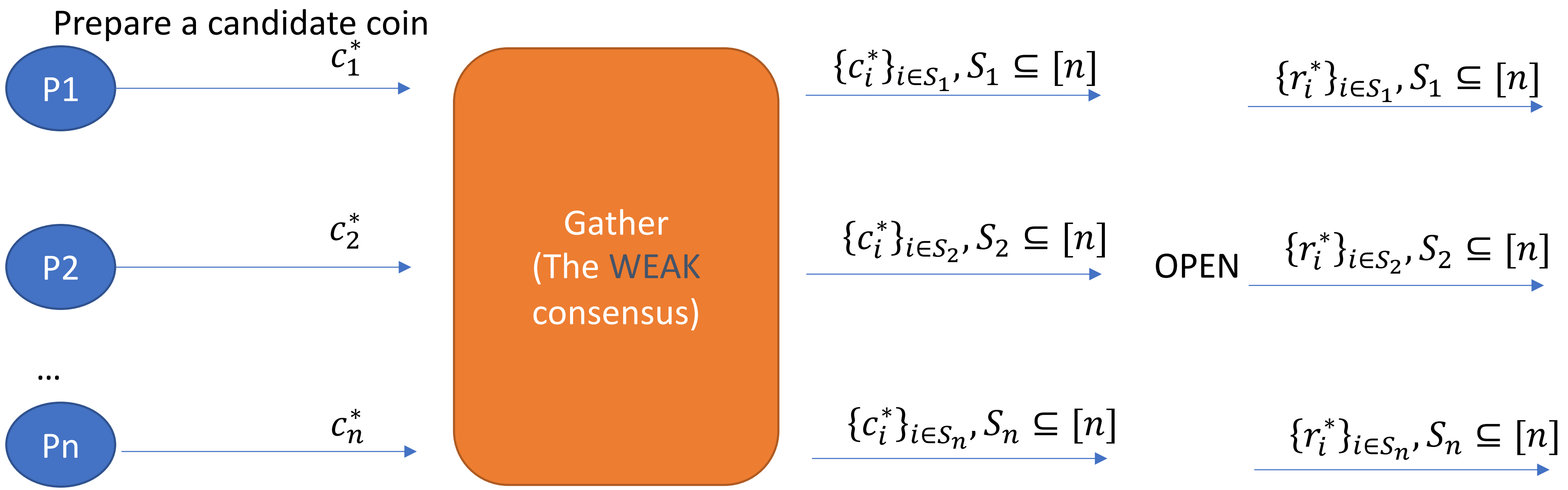
Asynchronous Weak Common Coin (existing approach)

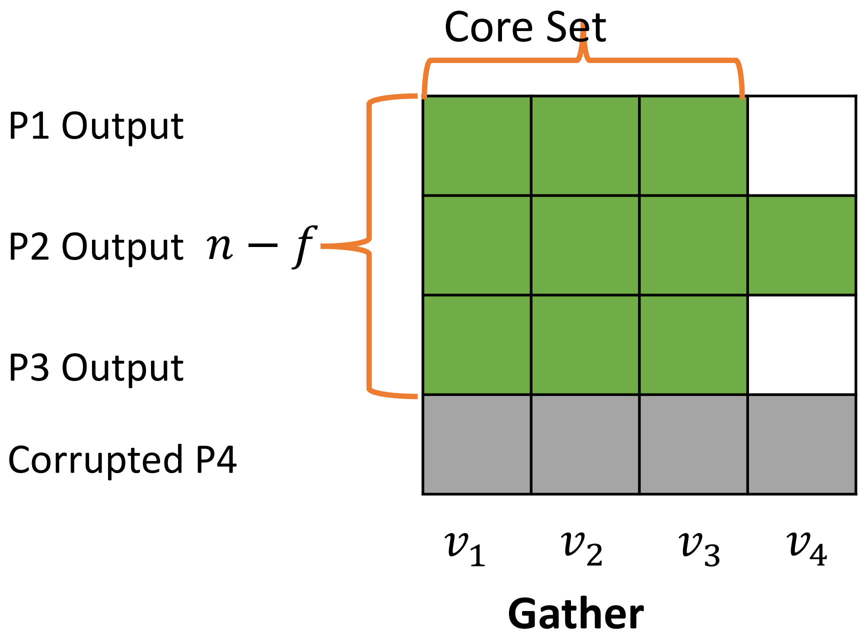
Gather
(The WEAK
consensus)



Outputs can be different, but must share the same core set

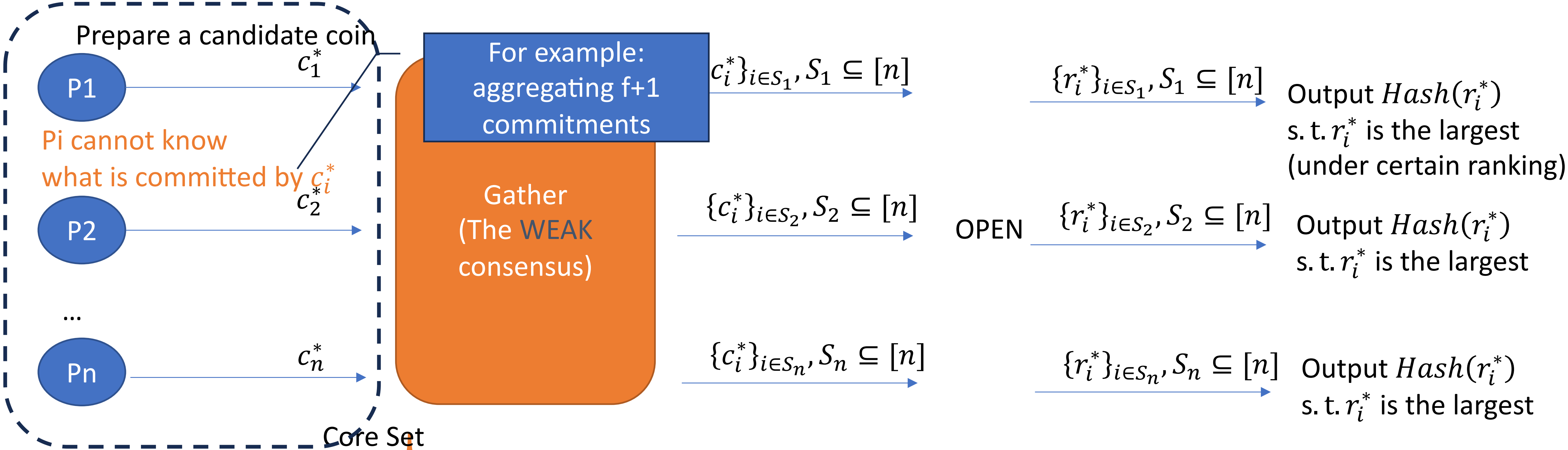
Asynchronous Weak Common Coin (existing approach)

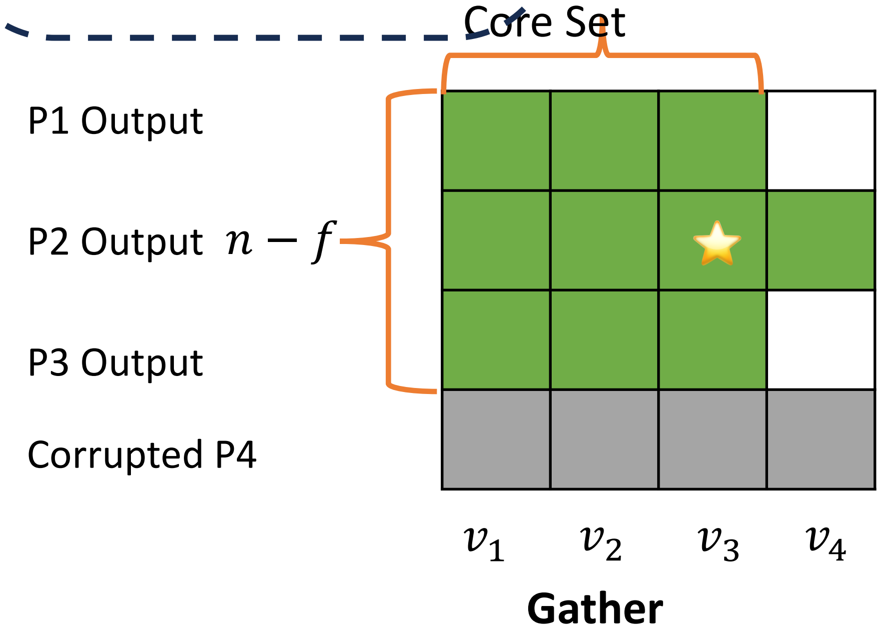




Outputs can be different, but must share the same core set

Asynchronous Weak Common Coin (existing approach)

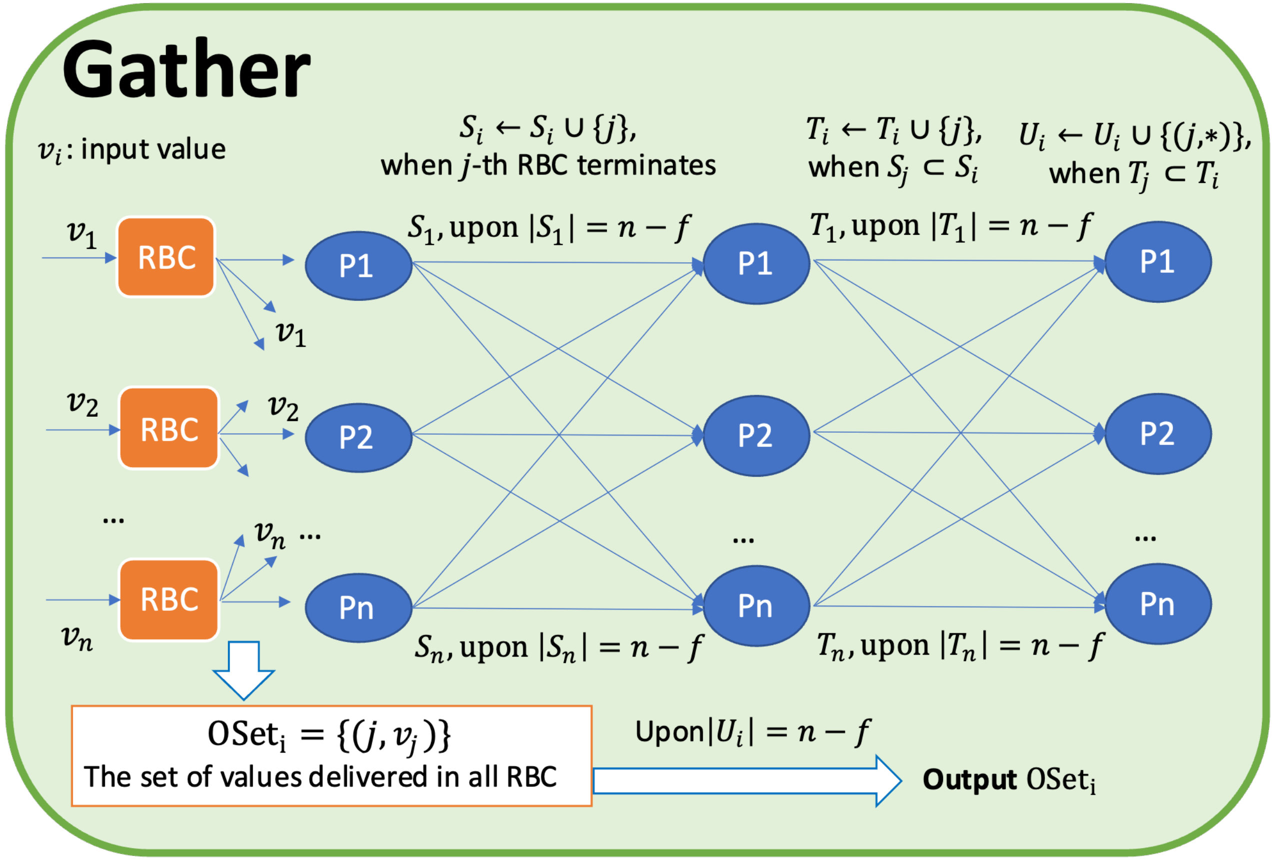




Outputs can be different, but must share the same core set

$$\Pr[\text{larges value} \in \text{core set}] = \frac{n-f}{n}$$

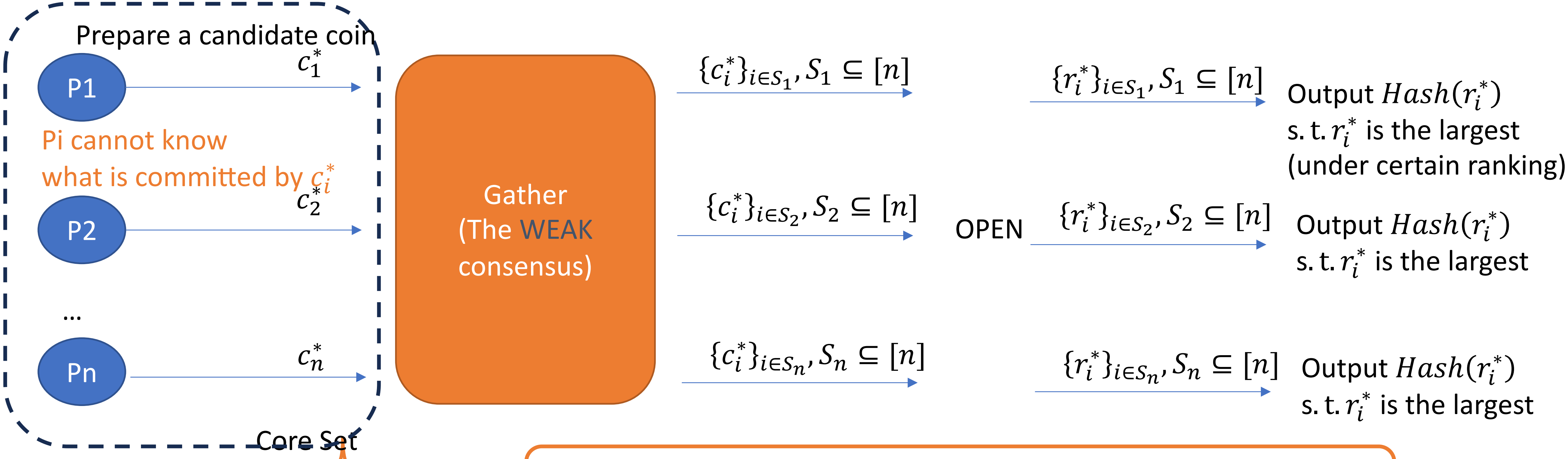
Gather is expensive…

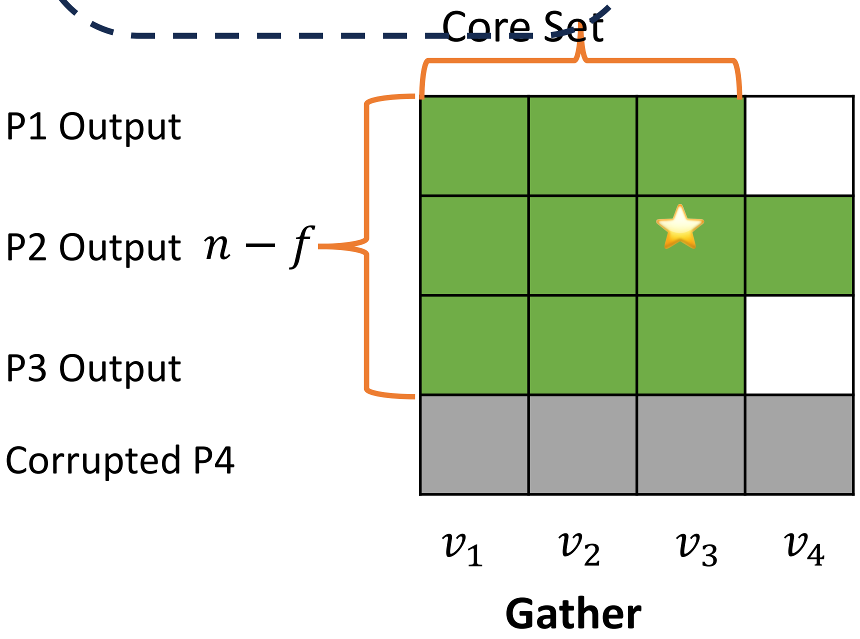


When a node outputs a subset, it needs to make sure a core set will appear in everyone else's output sets.

It seems to require all nodes to advice the others what they plan to output, necessitating $O(n^3)$-bit communication costs.

Do we really need Gather?



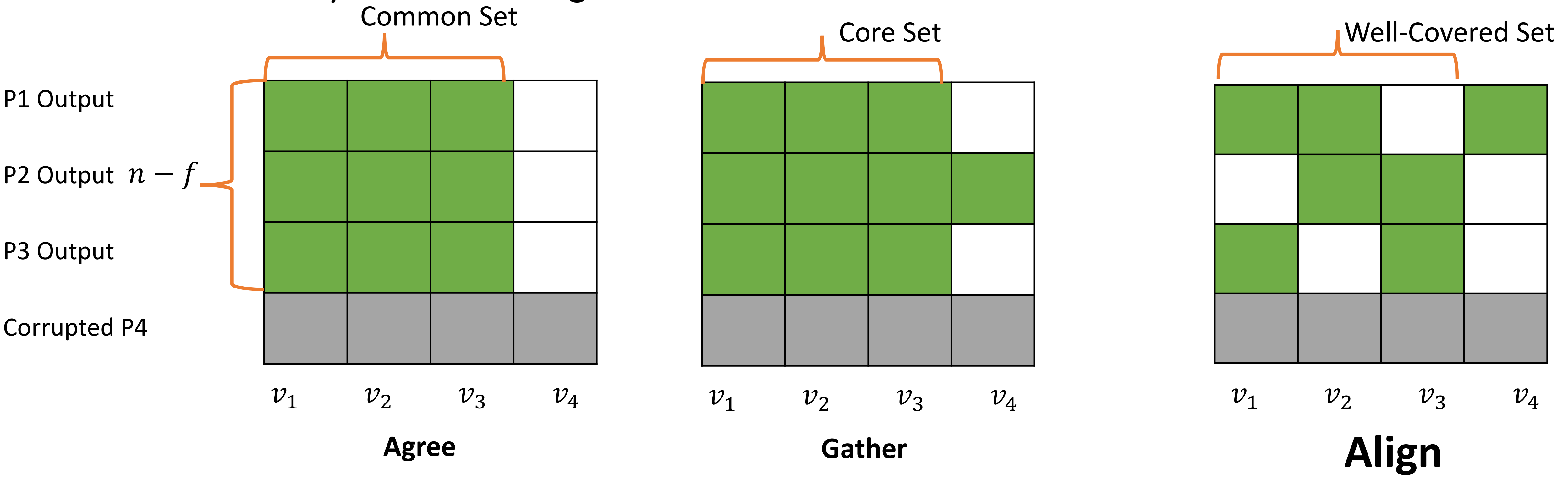


Our goal: with a constant probability, everyone sees the same largest value.

Still achievable with two relaxations:

- Only f+1 nodes initially see the largest value. They can help others.
- Core set is never needed.

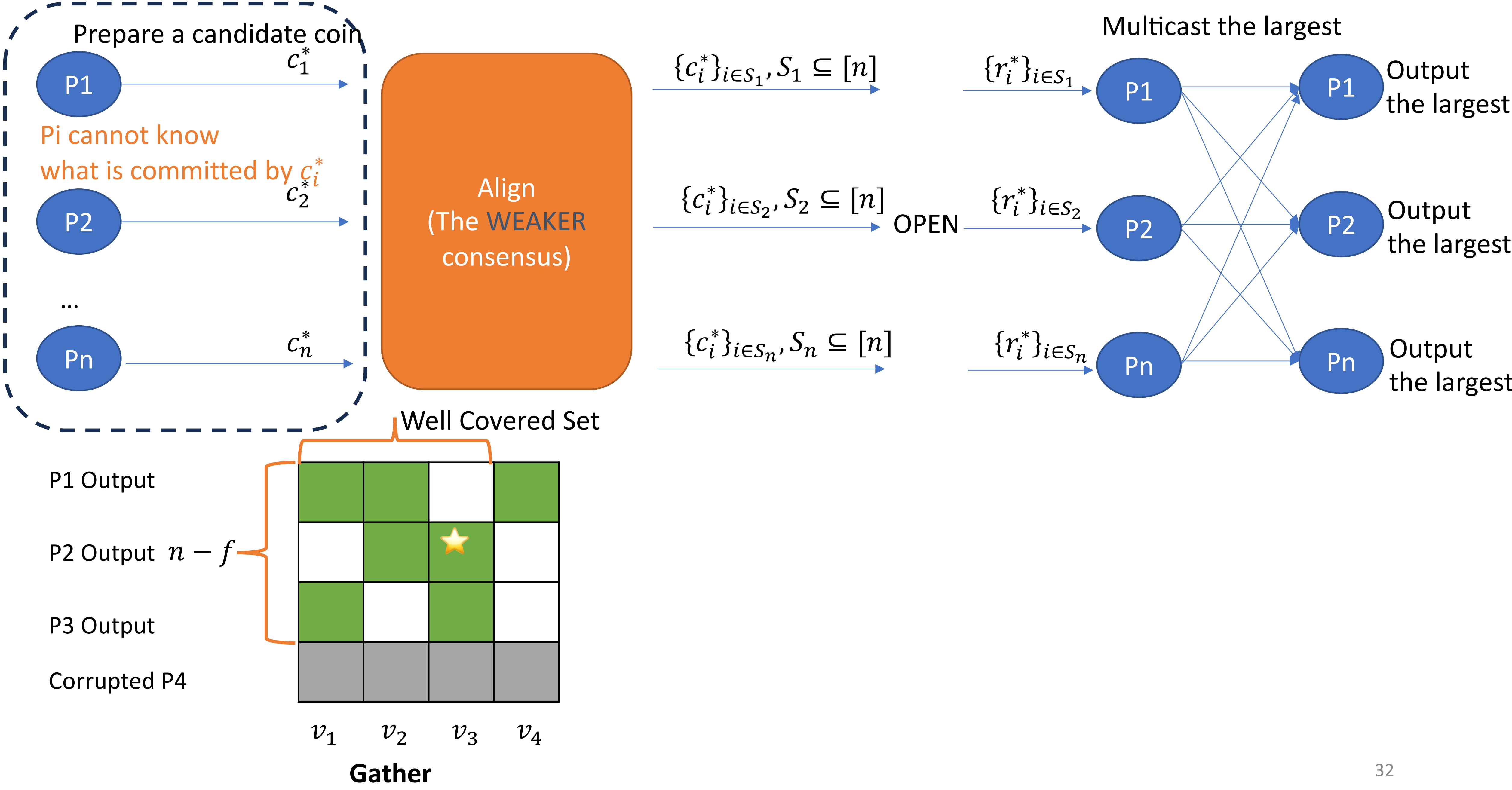
We introduced Asynchronous Alignment



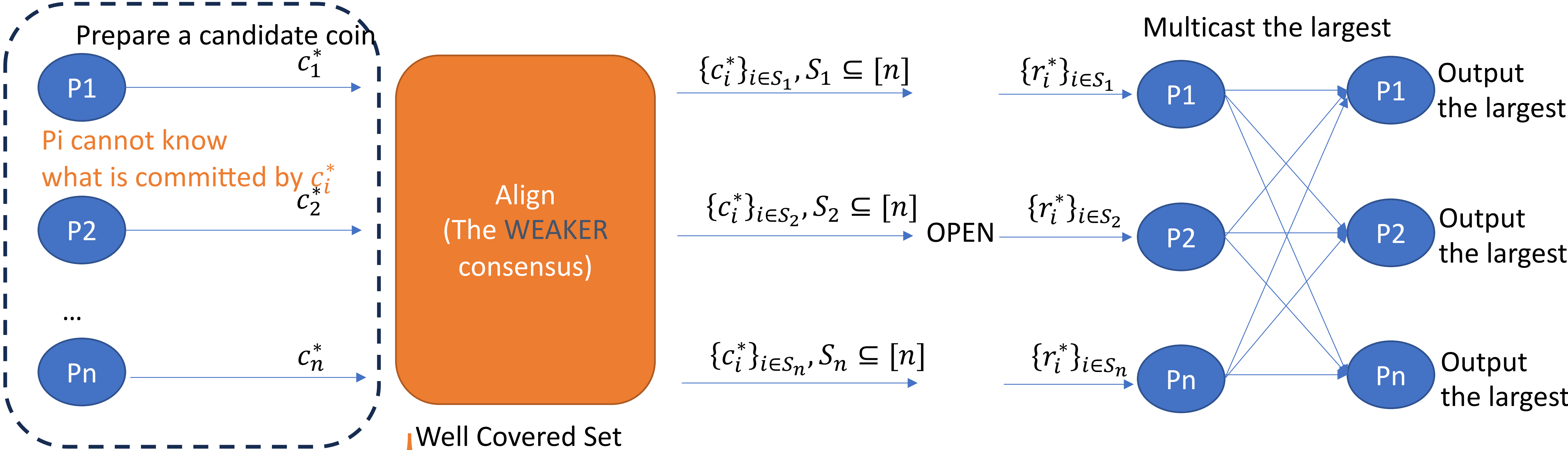
There exists a core set of n-f elements, so that the whole core set is outputted by all honest nodes

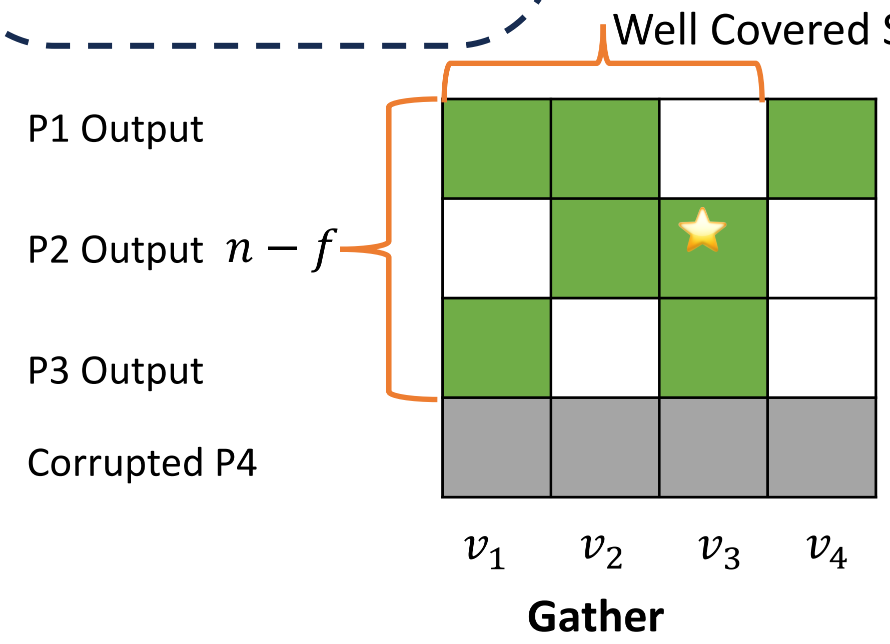
There exists a well-covered set of n-f elements, so that every element in the set is outputted by f+1 honest nodes

Asynchronous Weak Common Coin (Our Design)



Asynchronous Weak Common Coin (Our Design)

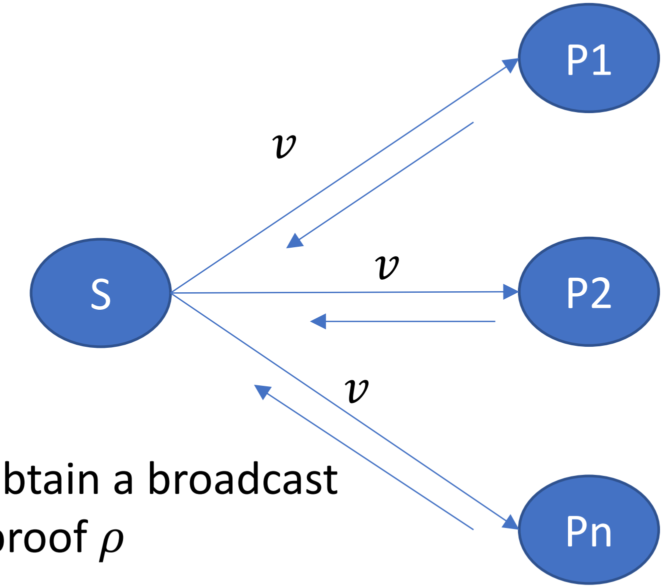




- With a probability of $\frac{n-f}{n}$, the largest value r_i^* appears in the well-covered set.
- In this case, f+1 honest nodes can decide this r_i^* as their largest value.
- In the next round, every node can receive at least $n-f=2f+1$ messages, with at least one carrying r_i^*.
- So all honest nodes can output the same $Hash(r_i^*)$

Asynchronous Alignment with $O(n^2)$ communication complexity

Provable broadcast (PB)



Obtain a broadcast proof ρ

1. The sender broadcasts v to the network

2. The receivers echo a receipt to the sender

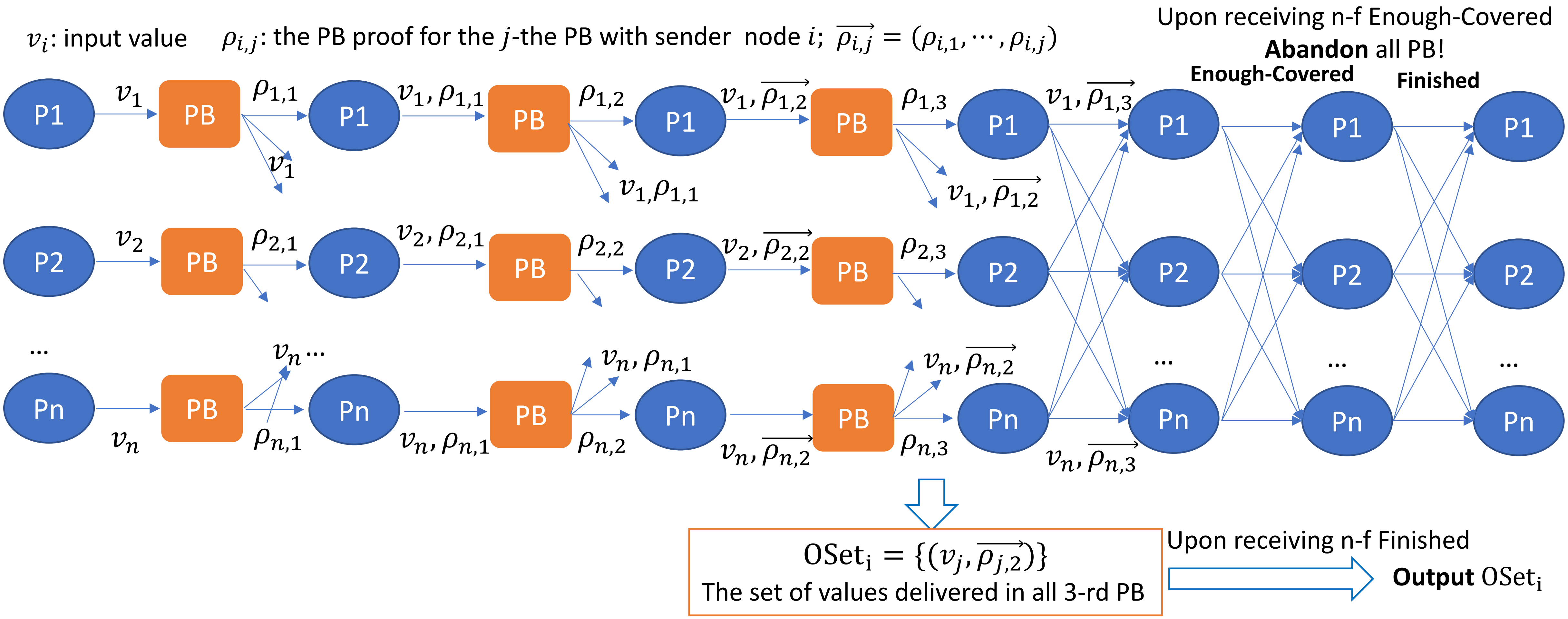
3. The sender can form a proof ρ based on the receipts

Communication cost: $O(n|v|)$,
with silent-setup threshold signature[Garg et al. Oakland'24, Das et al., CCS'23]

Security:
(1) only one value v can have a valid receipt in each instance
(2) The existence of ρ suggests that at least $n - 2f = f + 1$ honest nodes have received the value

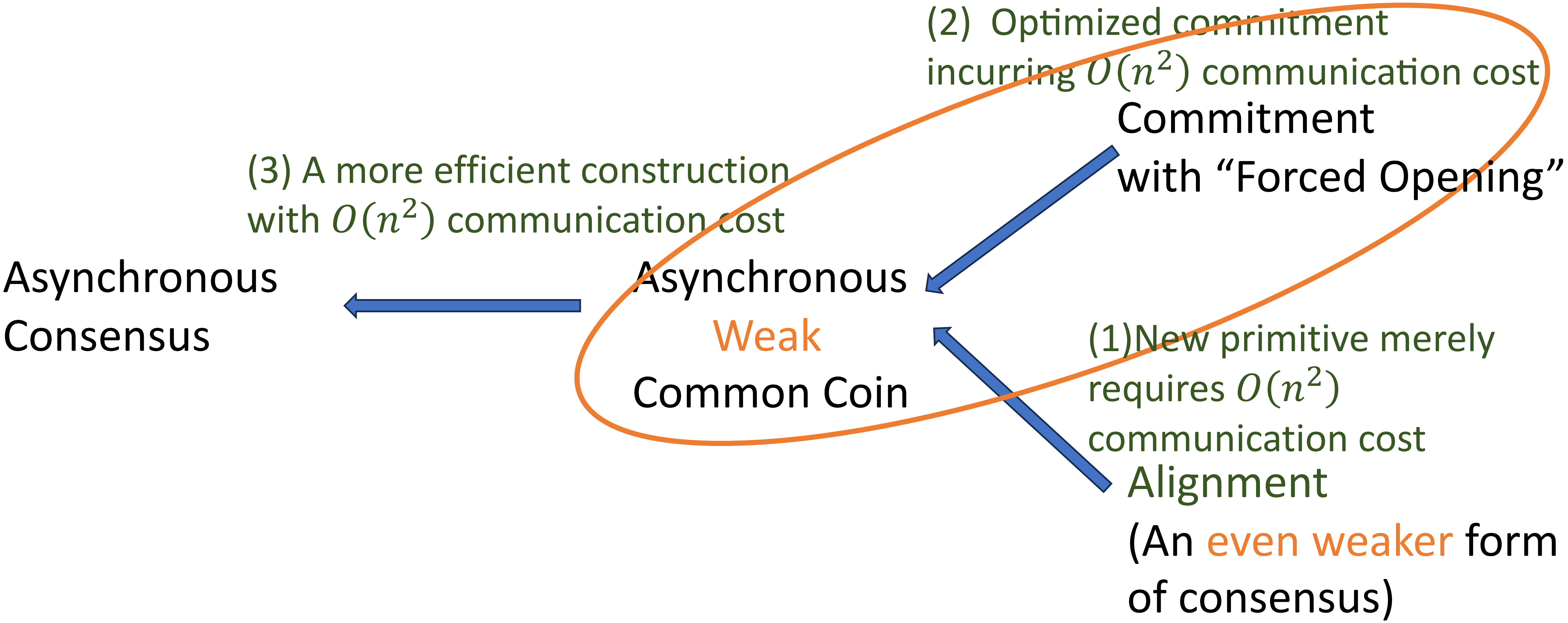
Asynchronous Alignment with $O(n^2)$ communication complexity

v_i: input value $\rho_{i,j}$: the PB proof for the j-the PB with sender node i; $\overrightarrow{\rho_{i,j}} = (\rho_{i,1}, \cdots, \rho_{i,j})$



Intuition:

An honest node decides to output when it received $n - f$ values with valid proofs, which suggests that those values have been received by at least $f + 1$ honest nodes. These values can define a well-covered set.


(2) Optimized commitment incurring $O(n^2)$ communication cost
Commitment with "Forced Opening"
(3) A more efficient construction with $O(n^2)$ communication cost
Asynchronous Consensus
Asynchronous Weak Common Coin
(1)New primitive merely requires $O(n^2)$ communication cost
Alignment
(An even weaker form of consensus)

Optimized Commitment from Silent-Setup Threshold Encryption

Existing instantiation: publicly verifiable secret sharing.

- $O(n\lambda)$ –sized commitment

New tool: Silent-setup threshold encryption (Garg et al., CRYPTO 2024)

- An $O(\lambda)$-sized ciphertext as the commitment.

Optimized Commitment from Silent-Setup Threshold Encryption

Existing instantiation: publicly verifiable secret sharing.

- $O(n\lambda)$ –sized commitment

New tool: Silent-setup threshold encryption (Garg et al., CRYPTO 2024)

- An $O(\lambda)$-sized ciphertext as the commitment.

Gaps:

- Ciphertexts cannot be aggregated, so we may need $f+1$ ciphertexts as a candidate coin
- Opening $O(n)$ ciphertexts may incur $O(n^3\lambda)$ comm. cost

Optimized Commitment from Silent-Setup Threshold Encryption

Existing instantiation: publicly verifiable secret sharing.

- $O(n\lambda)$ –sized commitment

New tool: Silent-setup threshold encryption (Garg et al., CRYPTO 2024)

- An $O(\lambda)$-sized ciphertext as the commitment.

Gaps:

- ➢ Ciphertexts cannot be aggregated, so we may need $f+1$ ciphertexts as a candidate coin
- ➢ Opening $O(n)$ ciphertexts may incur $O(n^3\lambda)$ comm. cost

Silent-Setup Threshold Encryption with **Tag homomorphism**

- Ciphertexts with the same tag can be aggregated
- Ciphertexts with the same tag can be decrypted with the same key

(2) Optimized commitment incurring $O(n^2)$ communication cost

Commitment with "Forced Opening"

(3) A more efficient construction with $O(n^2)$ communication cost

Asynchronous Consensus

Asynchronous Weak Common Coin

(1)New primitive merely requires $O(n^2)$ communication cost

Alignment (An even weaker form of consensus)

Leader Election with Quadratic Communication

- Agreement; Termination;
- Elect a good leader with constant probability

Asynchronous Consensus ← $O(n^2)$ — Asynchronous Leader Election ← $O(n^3)$ — Asynchronous Weak Common Coin

[Abraham et al. PODC'19]
[Lu et al., PODC'20]

[Gao et al., ICDCS'22]

Leader Election with Quadratic Communication

- Agreement; Termination;
- Elect a good leader with constant probability

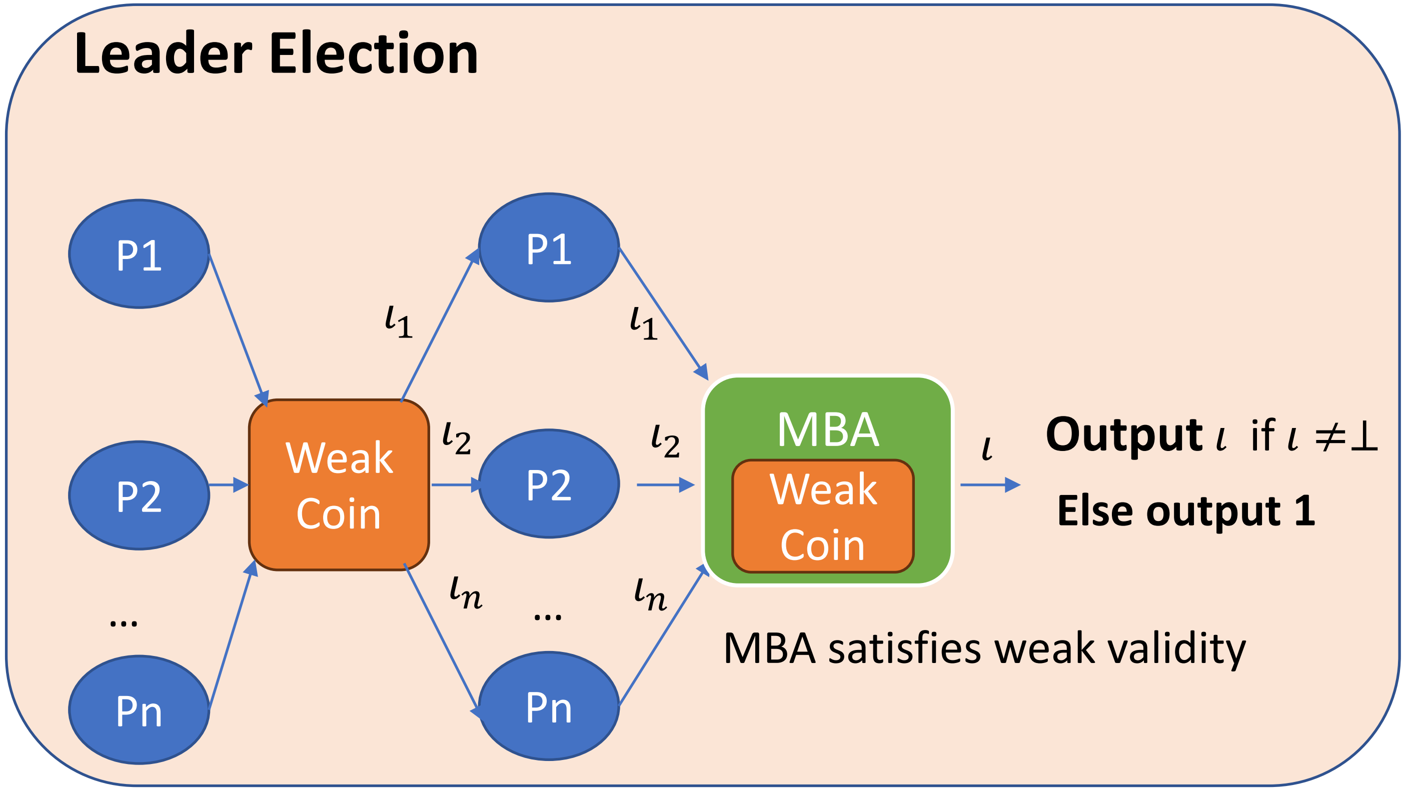
Asynchronous Consensus ← $O(n^2)$ ← Asynchronous Leader Election ← $O(n^3)$ ← Asynchronous **Weak** Common Coin

[Abraham et al. PODC'19]
[Lu et al., PODC'20]

[Gao et al., ICDCS'22]

$O(n^2)$ ←

Our new reduction



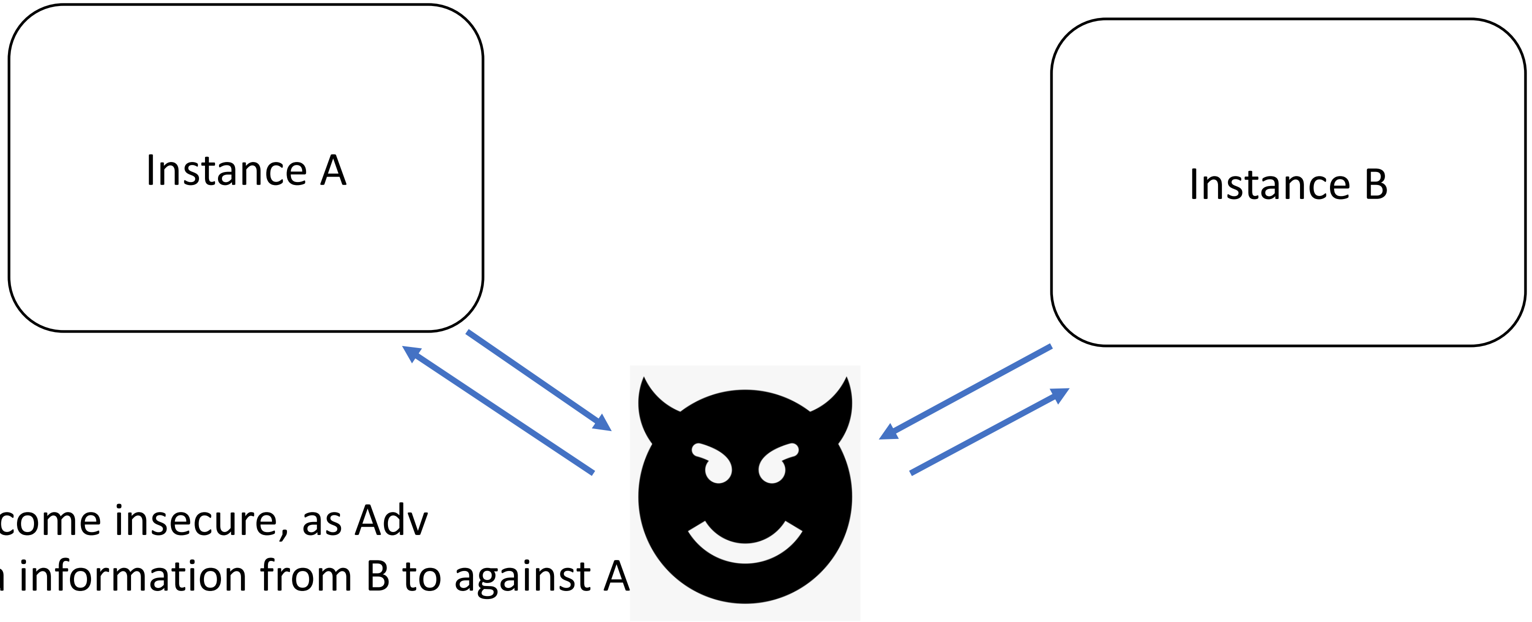
-- The Problem

-- The Challenges

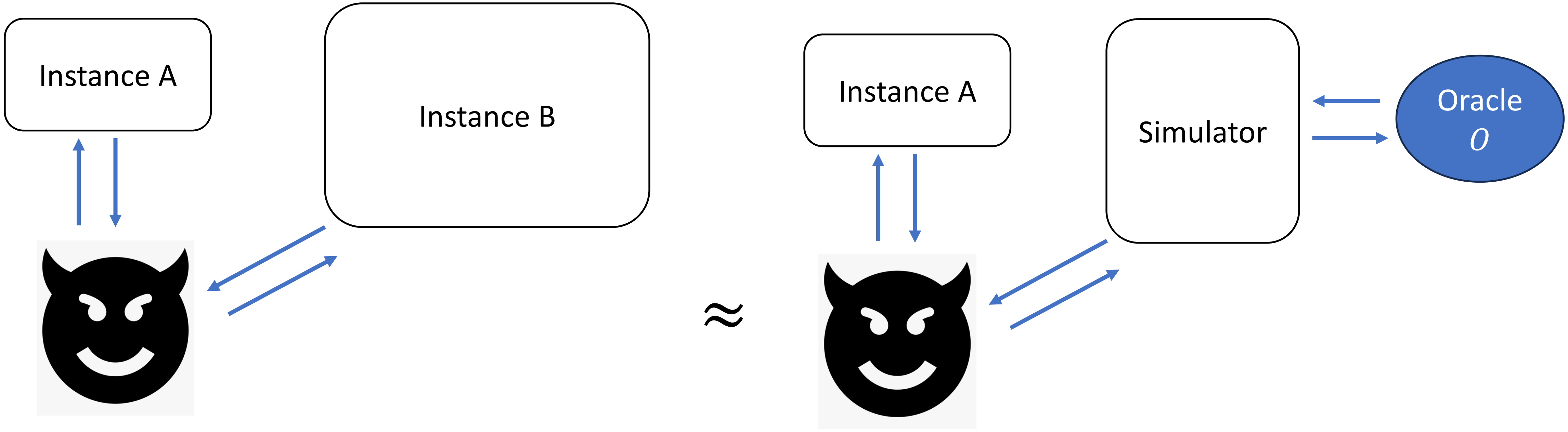
-- Our Contributions

---- Asymptotically Optimal Construction

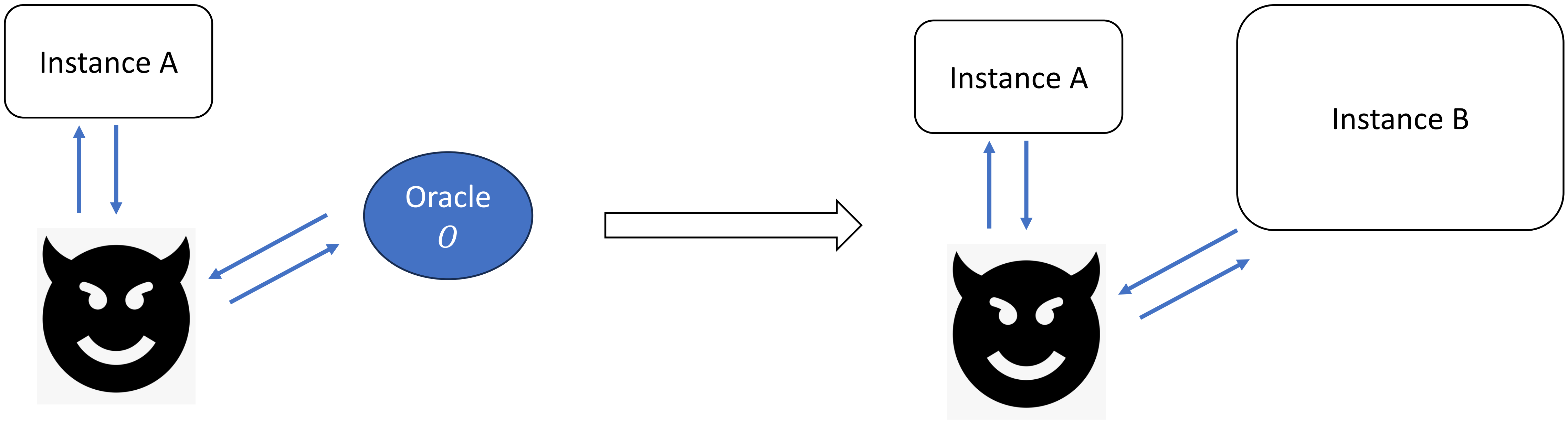
---- New Framework for Analyzing Specific Composition



Instance A may become insecure, as Adv
may leverage extra information from B to against A



Instance B can be emulated with O



If instance A is secure even when Adv has access to O,
then instance A remains secure when it is composed with instance B

In our coin protocols, the oracles are simply a signing oracle and a decryption oracle.

The position of this work*

	Communication Complexity	Round Complexity	Setup Assumptions
CKS[PODC'00]	$O(n^2)$	$O(1)$	Private Setup (basically a trusted party has created a coin in the setup)
KMS[CCS'20]	$O(n^4)$	$O(n)$	PKI(Public Key Infrastructure)
DYX+[IEEE SP'22]	$O(n^3)$	$O(\log n)$	PKI
AJM+[PODC'21]	$O(n^3)$	$O(1)$	PKI
AJM+[CRYPTO'23]	$O(n^3)$	$O(1)$	CRS(Common Reference String)&PKI
This work*	$O(n^2)$	$O(1)$	CRS&PKI

Silent Setups (KMS[CCS'20] through This work*)

Direction implications

The first batch of silent-setup quadratic-communication asynchronous consensus protocols

The first silent-setup quadratic-communication asynchronous distributed key generation

- By instantiating the framework in [Eprint:2025/149]

Future Questions

Q: Truly practical asynchronous coin?

Post-quantum Secure Asynchronous Coin?

Thanks!

Hanwen Feng
hanwen.feng@sydney.edu.au

Qiang Tang
qiang.tang@sydney.edu.au

THE UNIVERSITY OF SYDNEY