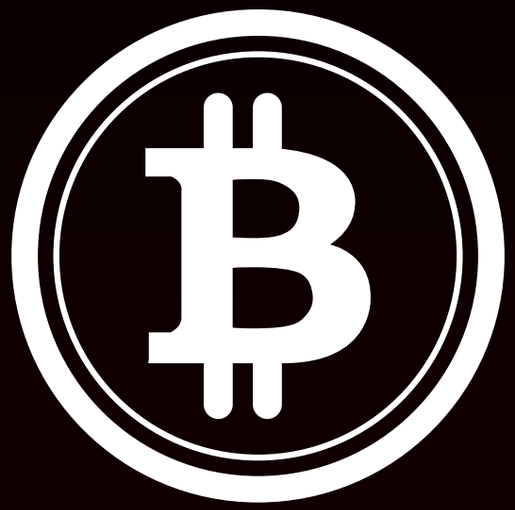
Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol

Aggelos Kiayias

Based joint work with
Alexander Russell
Bernardo David
Roman Oliynykov

INPUT I OUTPUT

Bitcoin



a remarkable solution

but to what problem?

Towards a Science of Blockchain Systems

distinguish

protocol solutions (like the bitcoin protocol)

from

objectives, (e.g., robust transaction ledger)

re-evaluate objectives & hypotheses

protocols

explore the protocol design space

assumptions

???

Analysing the Bitcoin Backbone

[Garay, **K**, Leonardos, 2014, http://eprint.iacr.org/2014/765]

bitcoin backbone = abstraction of "core" bitcoin protocol

robust transaction ledger = persistence / liveness

backbone protocol

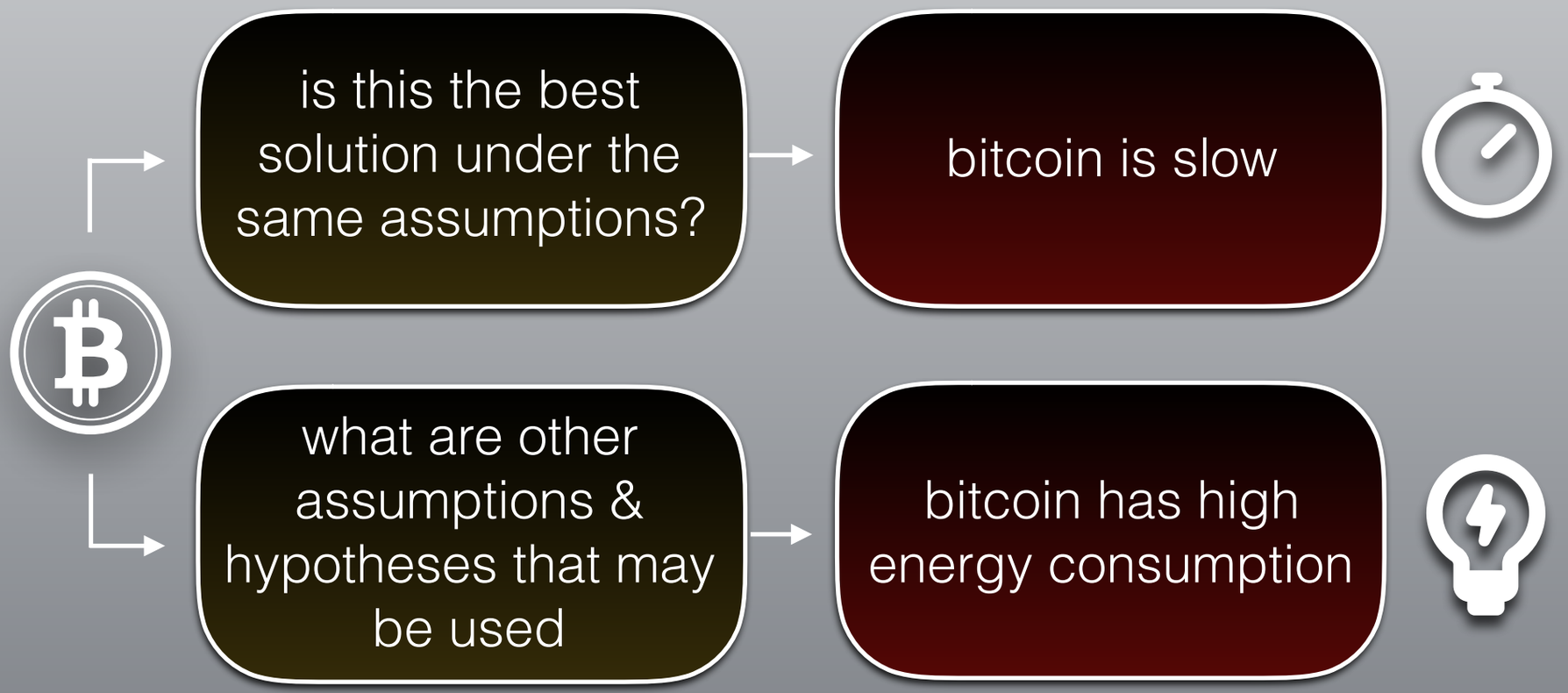
[GKL] provides proof of security in static model

PoW/ Random Oracle

is this the best solution under the same assumptions?

what are other assumptions & hypotheses that may be used

Protocol Design Challenges



VISA

~2000 tps

PayPal

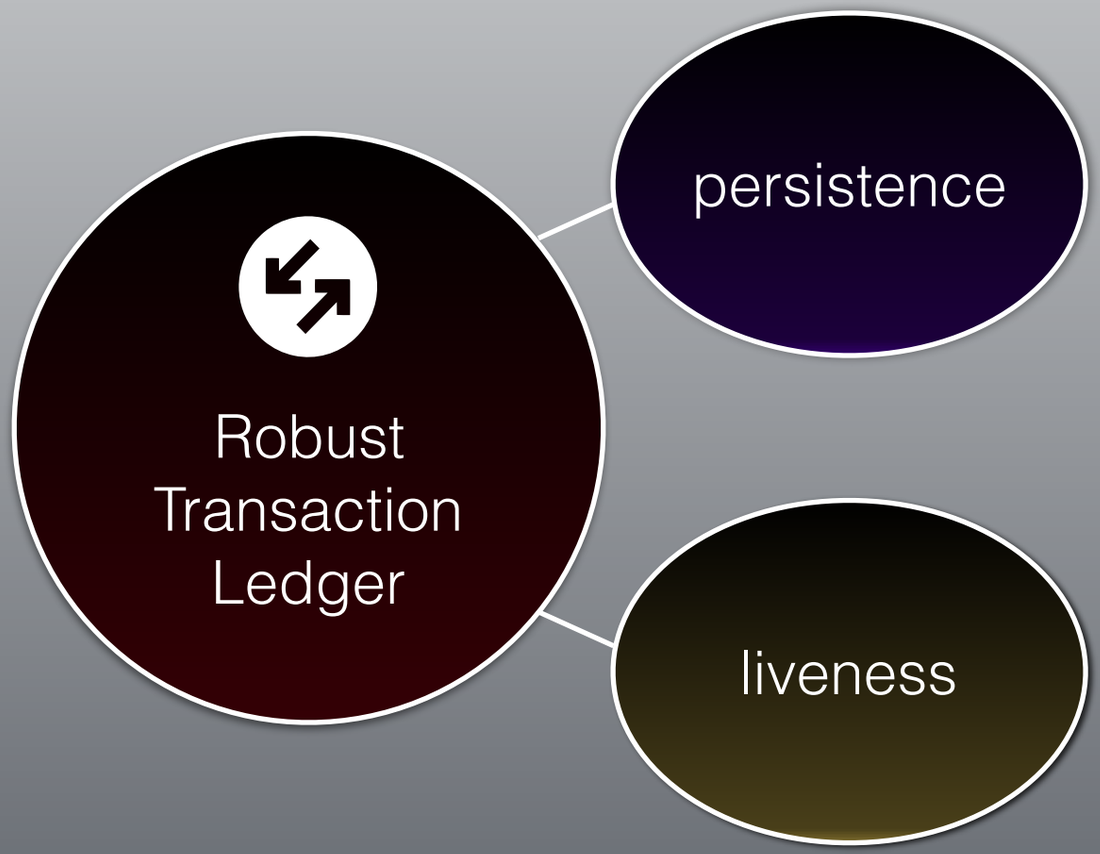
~100 tps

~7-8 tps

350MW

Robust Transaction Ledger

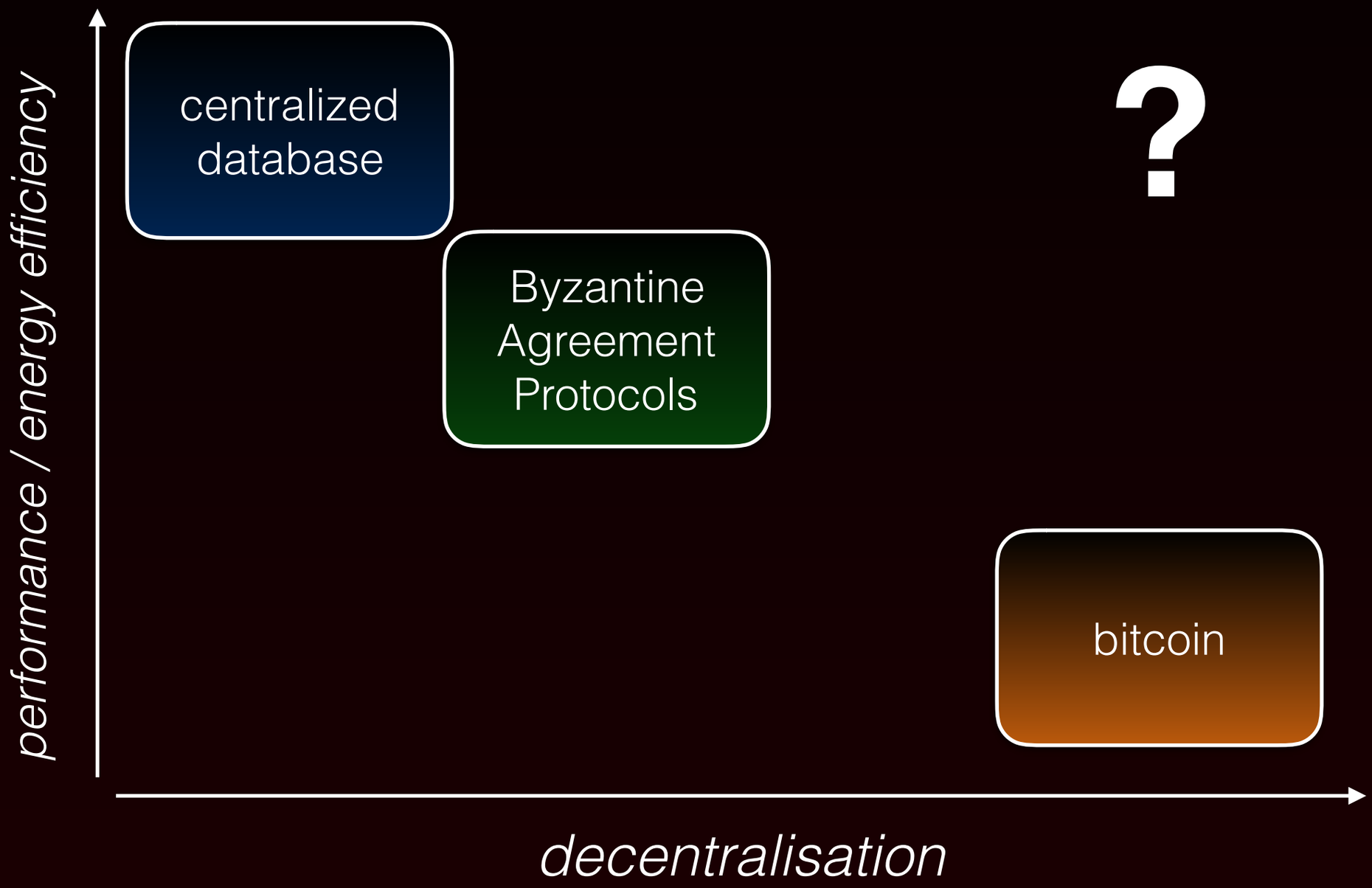
What are the alternative ways to meet the main objectives?



once a **tx** is confirmed by a node, any other node that reports it will agree with its placement in the ledger

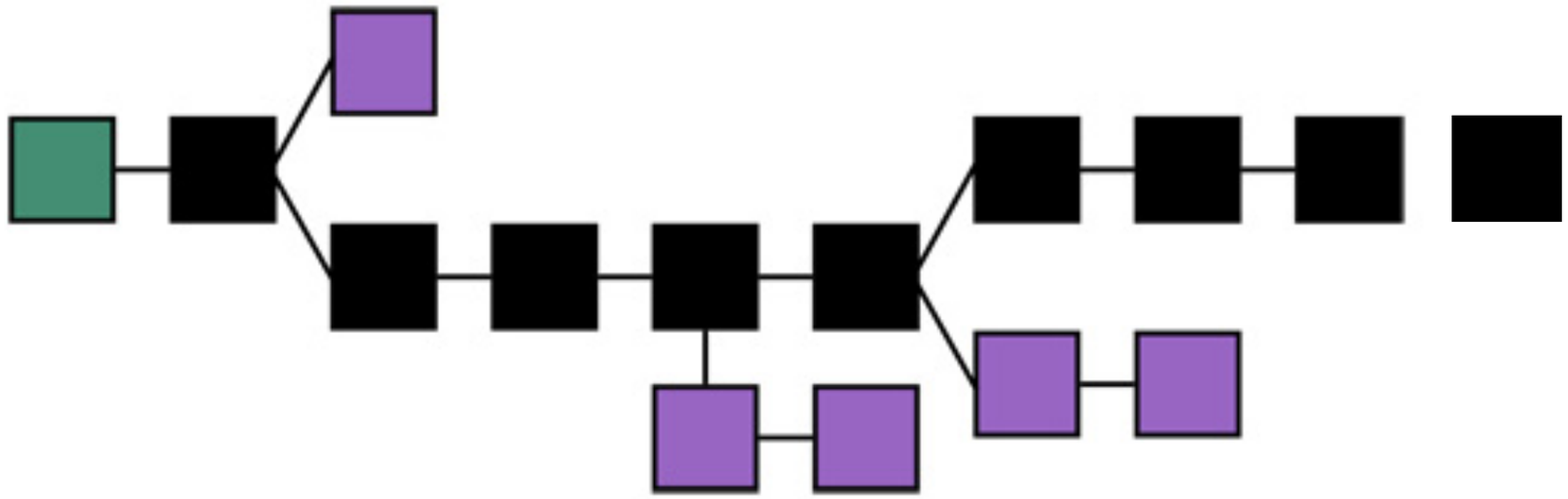
broadcasting a **tx** to the network will result to it being confirmed by the nodes

How to implement a Robust Transaction Ledger



Proof of Stake Motivation

generating the next block in bitcoin is like an election

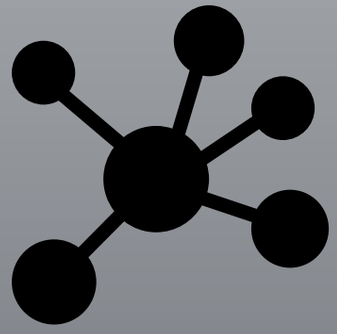


A miner is elected with probability proportional to its hashing power.

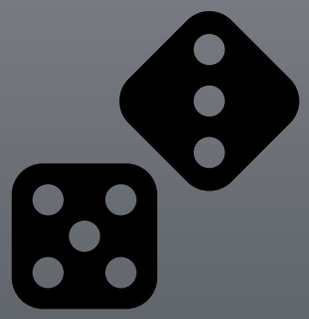
Collisions may occur but they can be resolved by the longest chain rule

Proof of Stake

Use stake instead of hashing power.

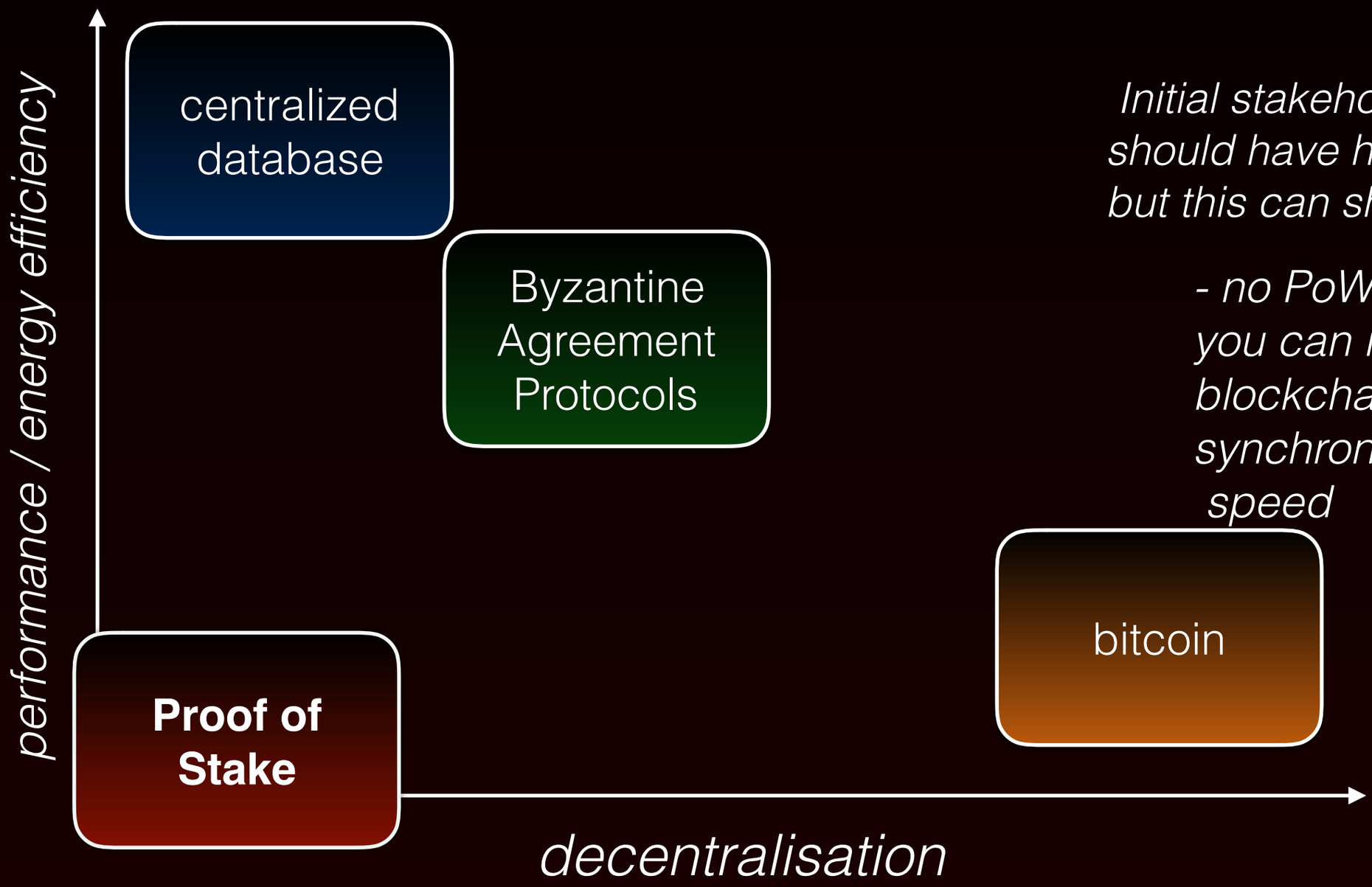


Define the set of miners to be the set of all stakeholders, as reported in the ledger.



Use a **randomised process** that takes the current stake into account to elect the next miner eligible to produce a block.

How to implement a Robust Transaction Ledger



Initial stakeholder distribution should have honest majority, but this can shift over time

- no PoW barrier - you can run the blockchain at maximum synchronization speed

PoS Based Cryptocurrencies

- Nxt
- Blackcoin
- Peercoin (PPCoin)
- Neucoin
- Many others…

PoS Design Ideas (1)

- PeerCoin, NXT
  - Eligibility to issue a block is based on a hash value that depends on current chain
  - Level of stake of stakeholder calibrates eligibility so that, e.g., higher stake results in more frequent eligibility.

PoS Design Ideas (2)

- [BentovGabizonMizrahi16] attempt a more principled approach as follows:
  - Stakeholders are elected based on their stake.
  - Collective coin flipping is used to seed the stakeholder distribution.

PoS woes

- **Grinding attacks.** The adversary may try to bias the random election process in its favor.

- **Nothing-at-stake.** the adversary may try multiple alternative histories (even from any point in the past), thus, simple "*longest chain wins*" is meaningless assuming stake shifts over time.

- **Circularity**. even if coin flipping is used to inject fresh randomness, it can be proven secure assuming there is agreement between the participants. Given that the blockchain is used for agreement, how we can avoid circularity in the security argument?

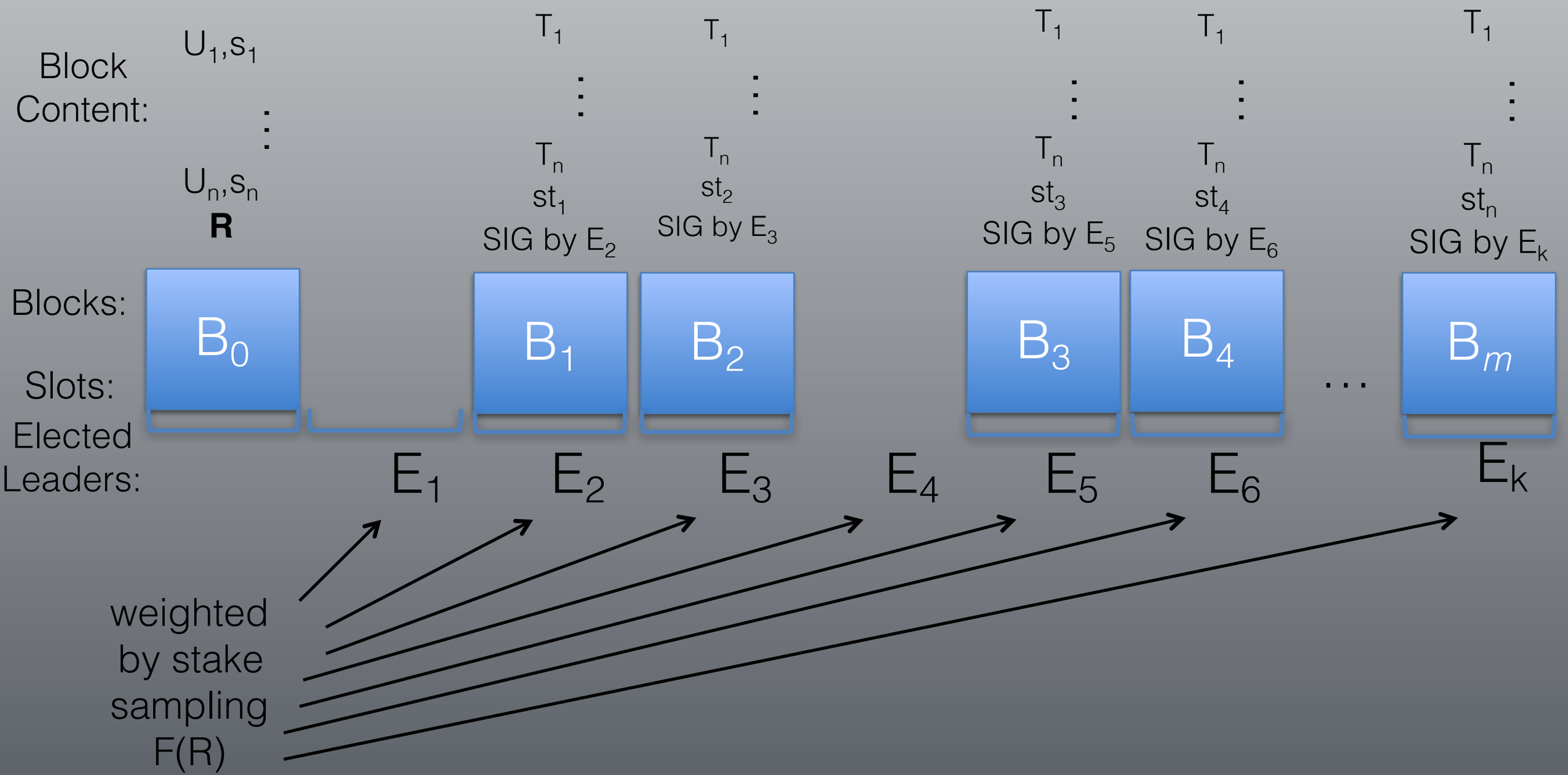
Our Contributions

- **Formalisation:**
  - Modeling the PoS design challenge.

- **Construction:**
  - Ouroboros: A PoS-based Robust Transaction Ledger.

- **Proof strategy:**
  - Show agreement works for a small interval via a combinatorial argument for static stake. Then, exploit this short agreement opportunity to run an MPC protocol that will be used to bootstrap the process.

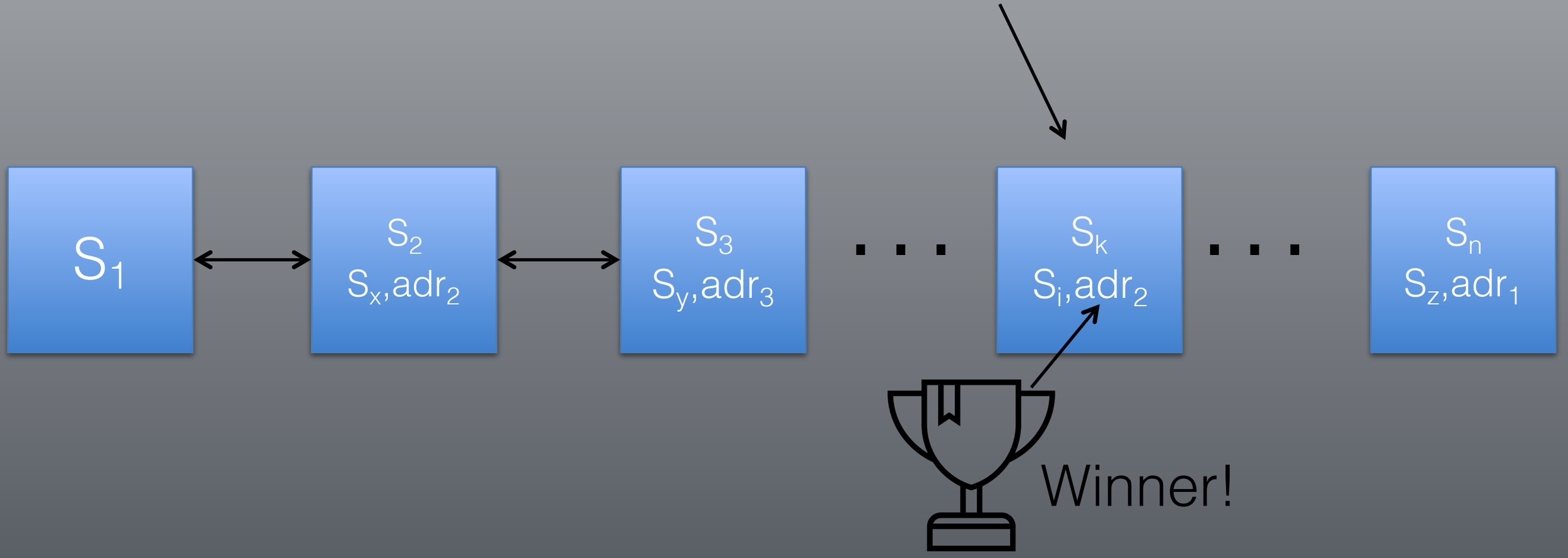
Ouroboros : Static Stake



Weighted-by-stake Sampling

Stakeholder

F(seed) ⟶ **i**



Security Properties

- Common Prefix:

$$\forall r_1, r_2, (r_1 \leq r_2), P_1, P_2, \text{ with } \mathcal{C}_1, \mathcal{C}_2 : \mathcal{C}_1^{\lceil k} \preceq \mathcal{C}_2$$

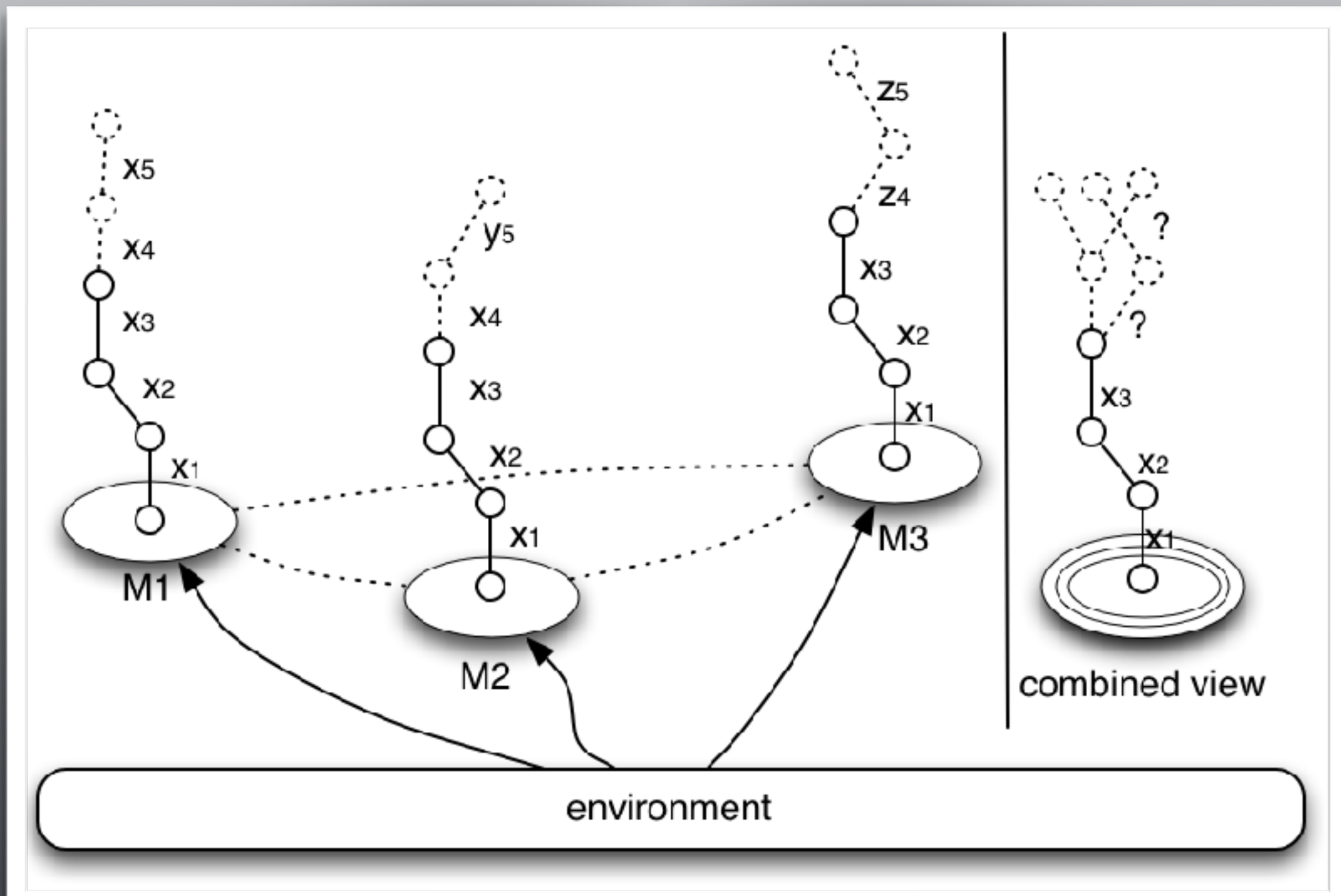
- Chain Quality:

Parameters $\mu \in (0,1), k \in \mathbb{N}$
The proportion of blocks in any k-long subsequence produced by the adversary is less than μk

- Chain Growth:

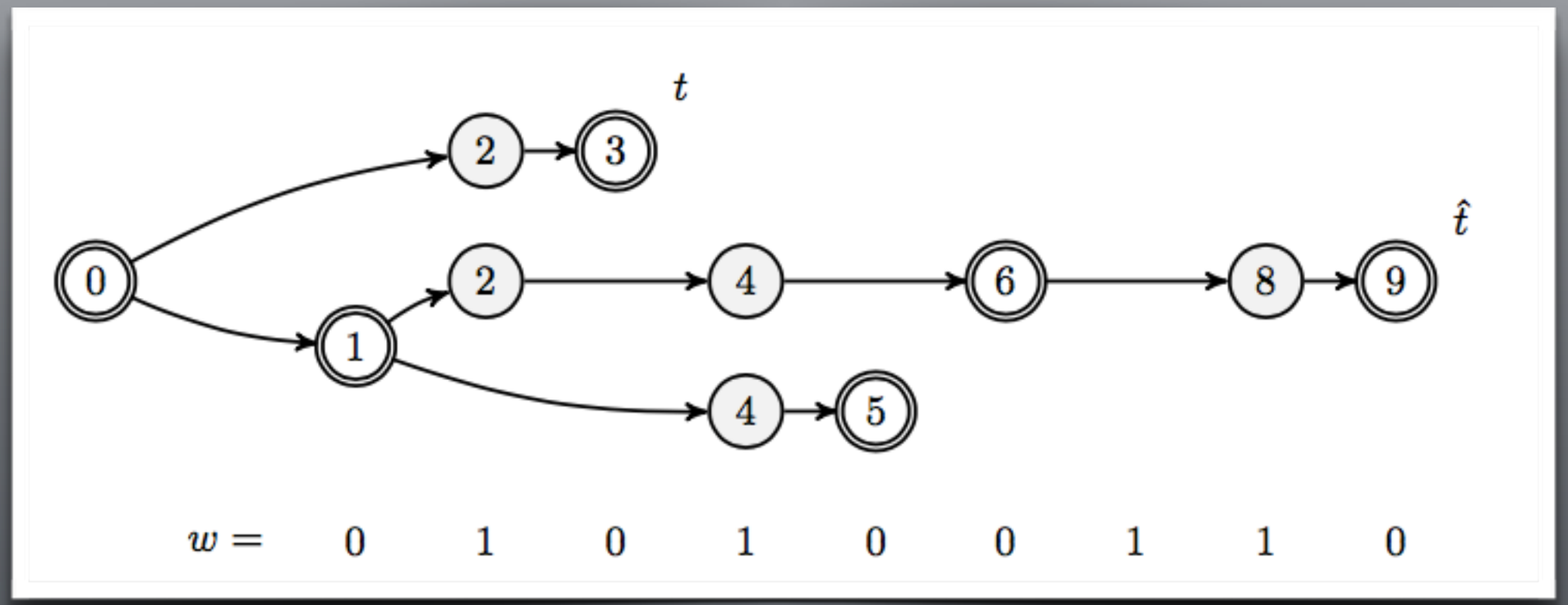
Parameters $\tau \in (0,1), s \in \mathbb{N}$
$\forall r_1, r_2$ honest player P with chains $\mathcal{C}_1, \mathcal{C}_2$
$r_2 - r_1 \geq s \implies |\mathcal{C}_2| - |\mathcal{C}_1| \geq \tau s$

Common Prefix: will honest players converge?



“Forkable” Strings

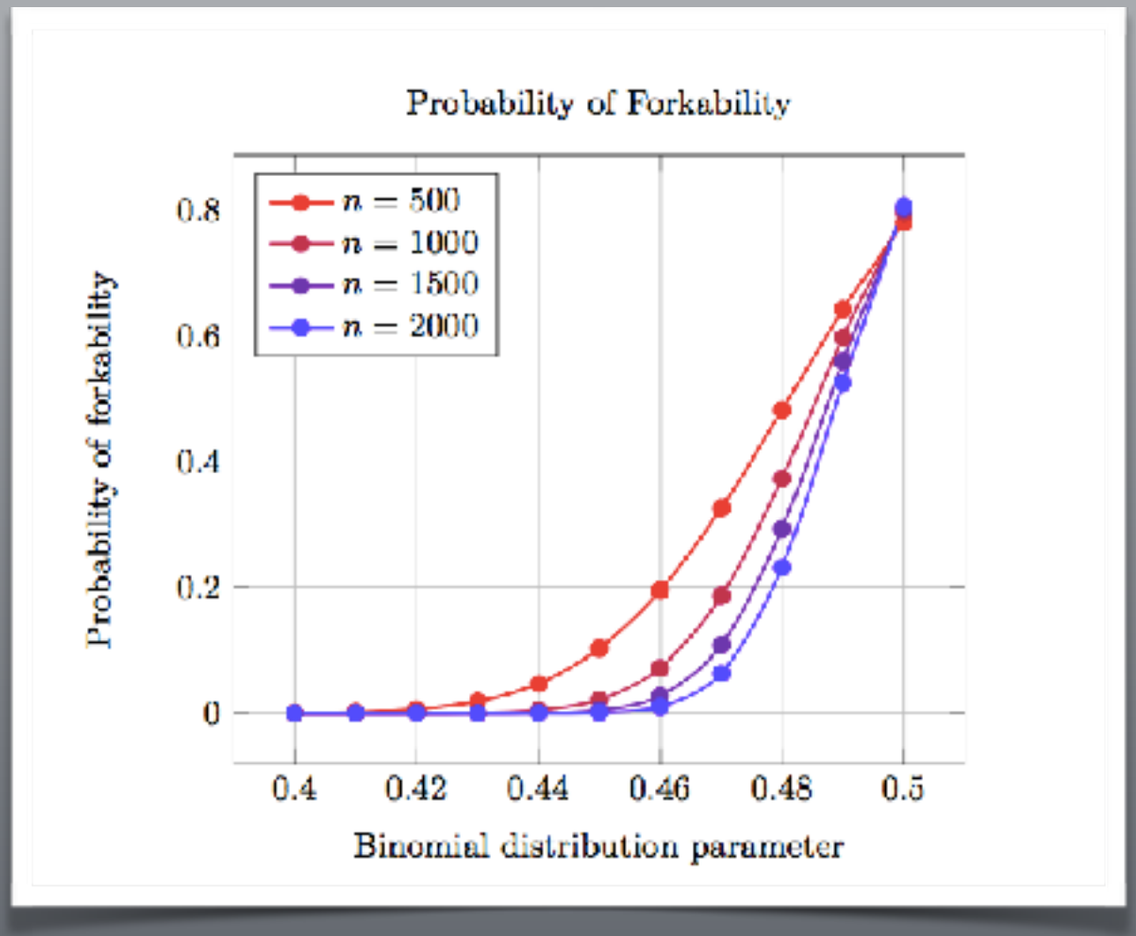
$$w \in \{0,1\}^* \qquad w_i = \begin{cases} 0 & \textit{i}\text{-th slot belongs to an honest party} \\ 1 & \textit{i}\text{-th slot belongs to a malicious coalition} \end{cases}$$



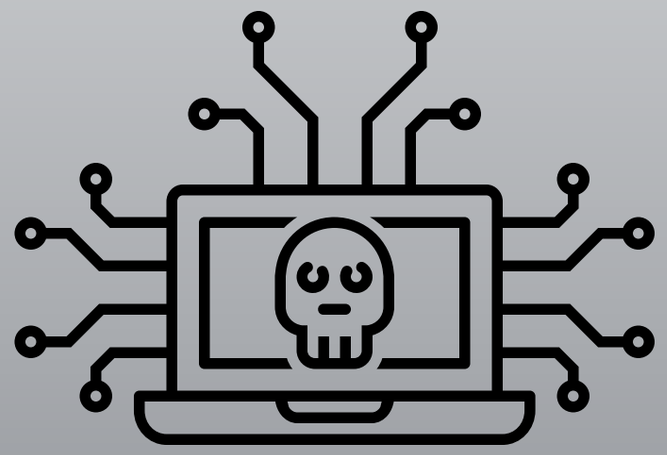
Forkable Density

Theorem. (1) There are no forkable strings of length *n* of Hamming weight ratio less than 1/3

(2) The density of forkable strings drops exponentially in *n*, $2^{-\Theta(\sqrt{n})}$ assuming (1-ε)/2 Hamming Weight ratio.



Covert Adversaries



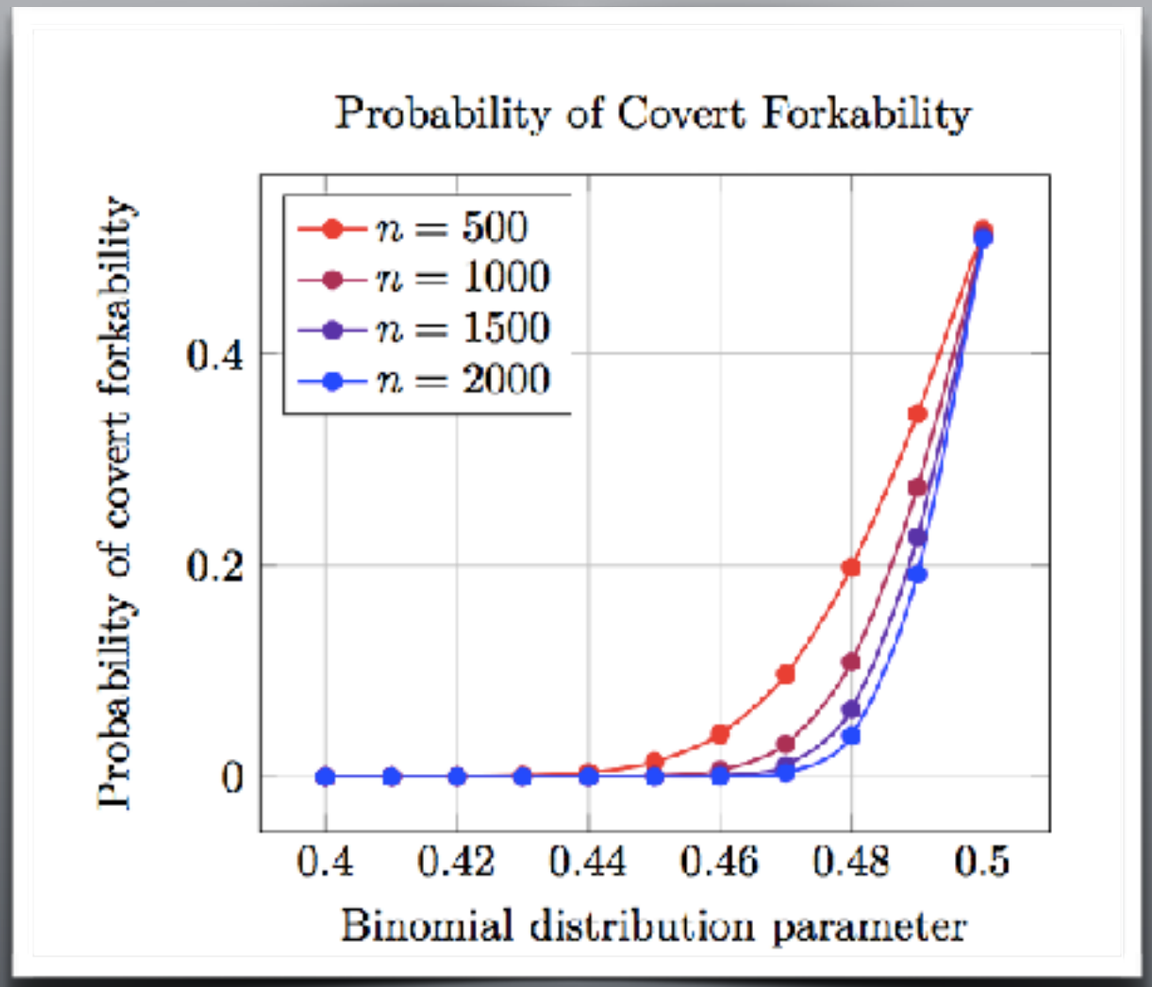
The forking attacks include strategies that sign on the same slot twice.

This is not “deniable”

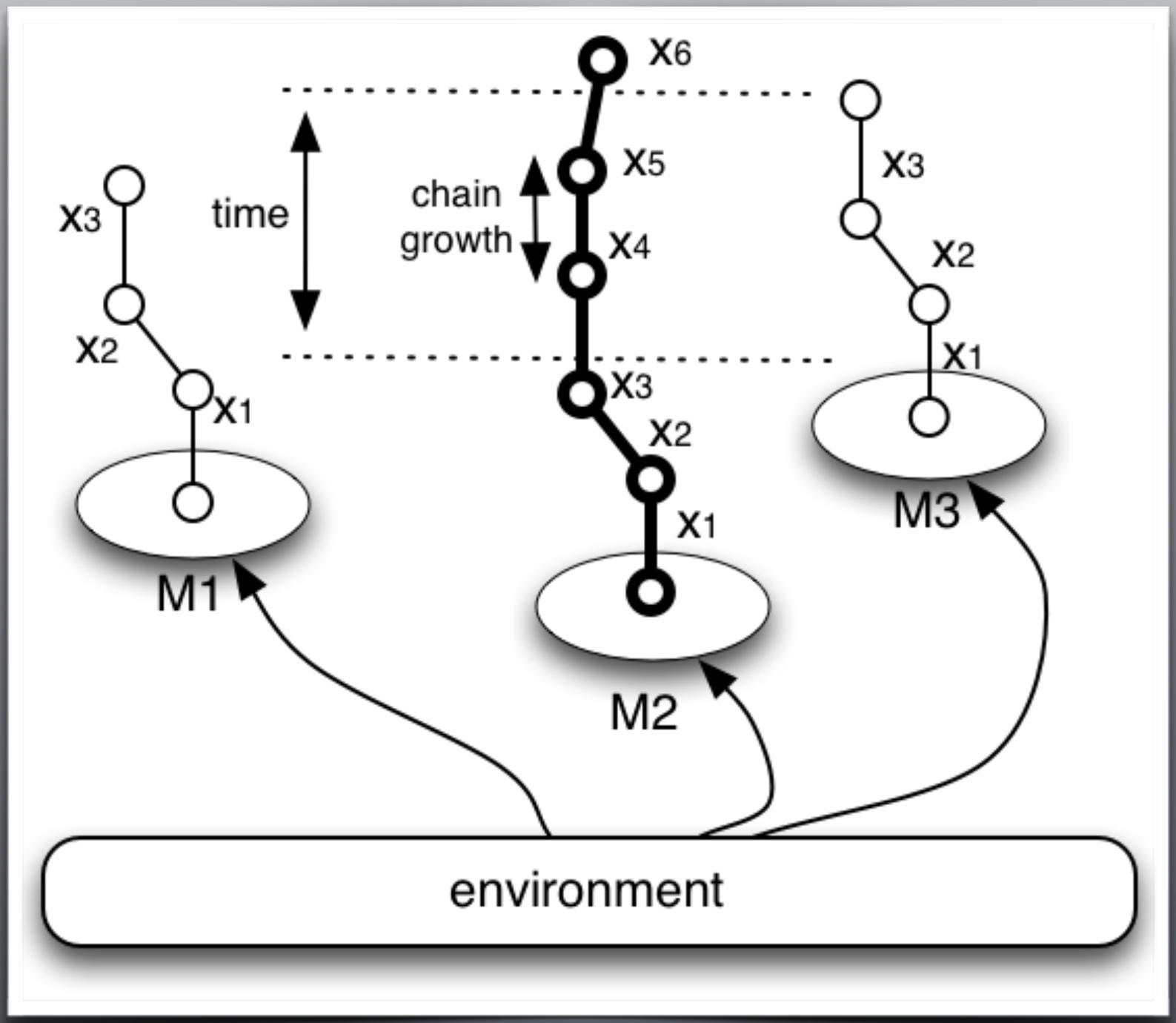
What is the potential to do forking in a covert / deniable way?

Covert Forkable Density

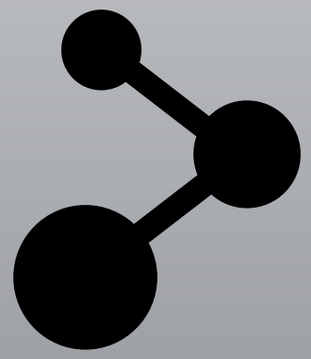
Theorem.
(1) The density of forkable strings drops exponentially in n, $2^{-\Theta(n)}$ assuming (1-ε)/2 Hamming Weight ratio.



Chain Growth:
does the chain grow?



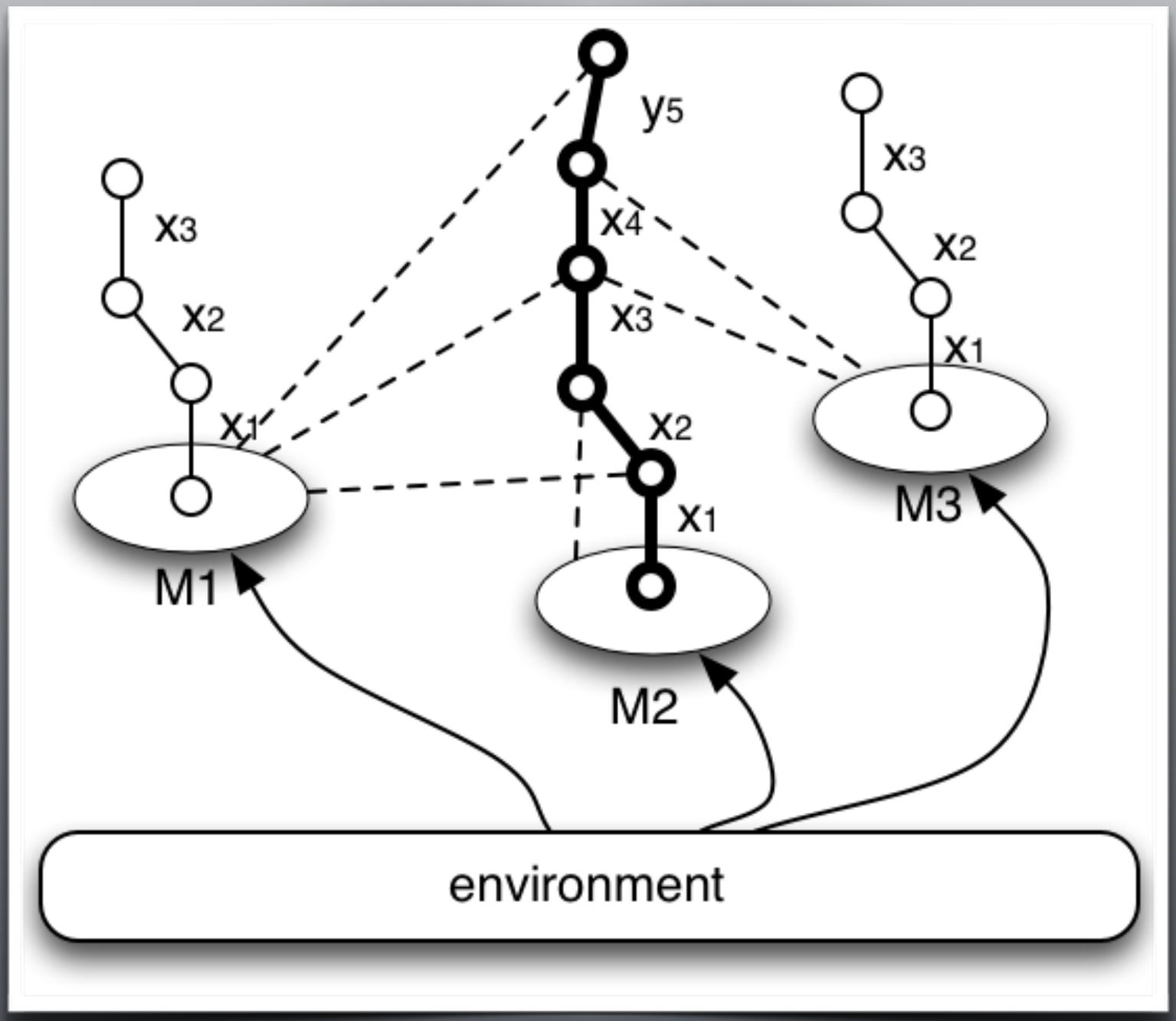
CG proof



As in Bitcoin, the “longest” chain wins rule, guarantees the honest parties’ chain cannot be hindered by adversarial actions.

it will grow with a speed proportional to at least the honest stakeholders ratio.

Chain Quality: are honest blocks going to be adopted by the parties?

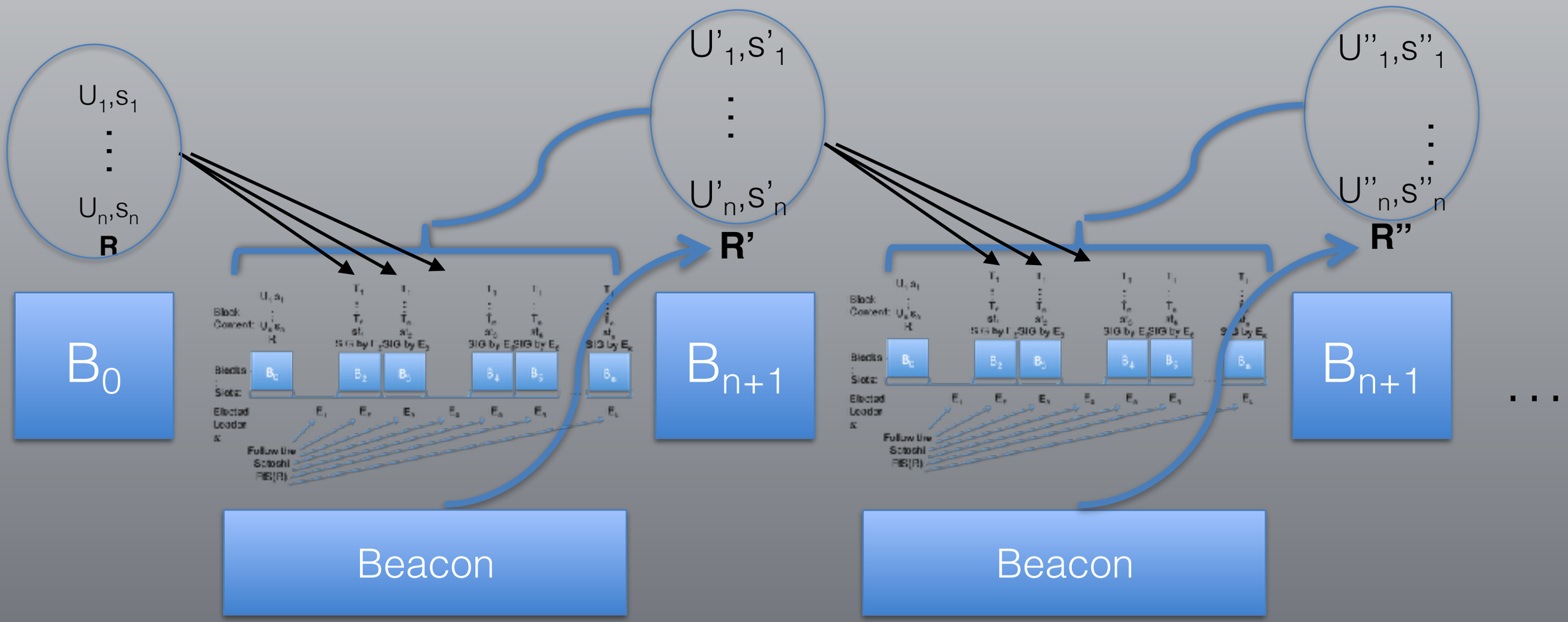


CQ Proof

By CG, observe that the rate of the honest parties chains will grow proportionally to at least the ratio of honest stakeholders.

In any sufficiently long sequence of slots, the number of blocks that legally can be contributed the adversary is below the bound.

Ouroboros: Dynamic Stake

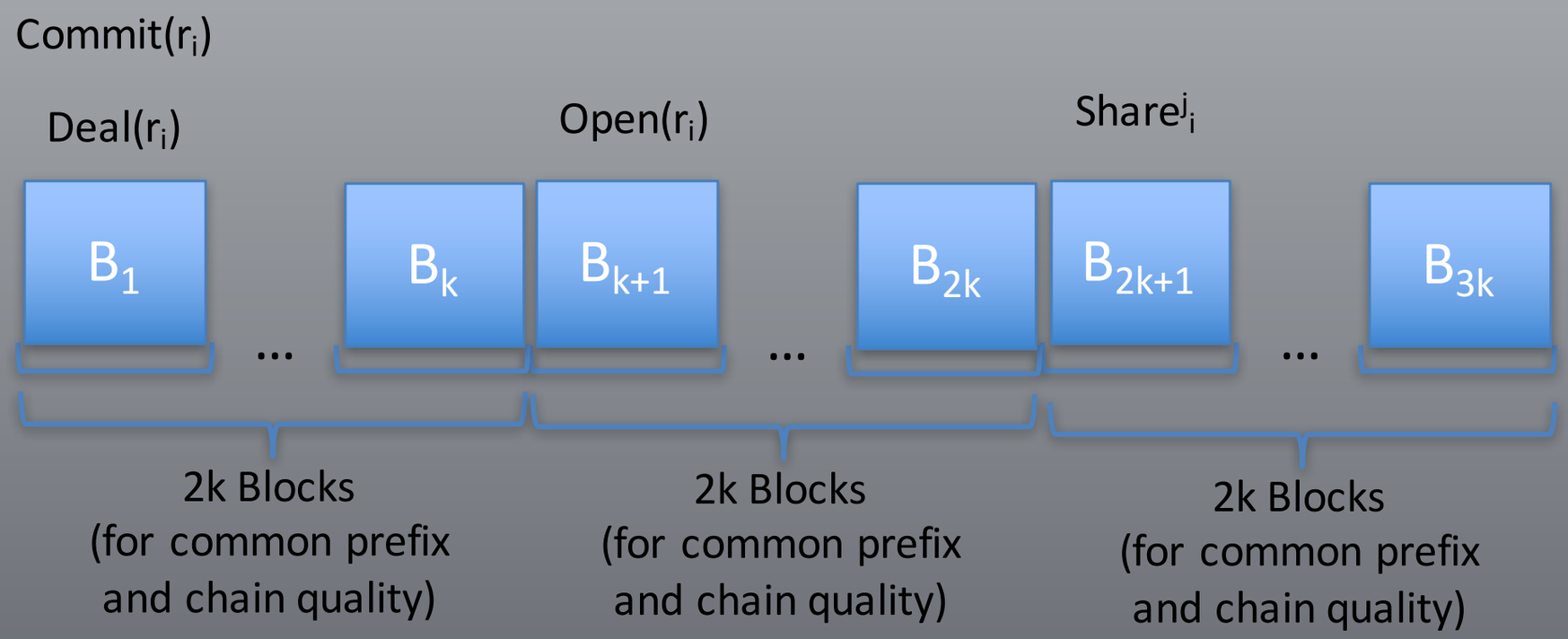


R **R'** **R''**

randomness beacon

Beacon via G.O.D. coin tossing

- For every stakeholder when each epoch starts:

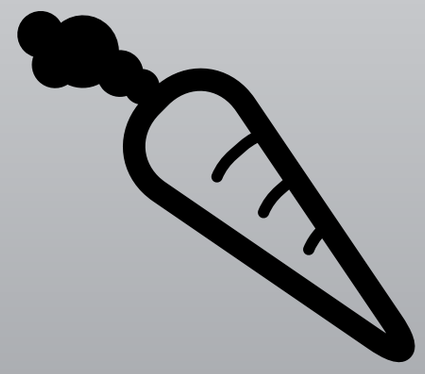


Use publicly verifiable secret-sharing (PVSS) for distributing commitment openings

Building Blocks

- Publicly Verifiable Secret Sharing:
  - [Schoenmakers99]; can be based on ECC.
- Commitments, many possibilities, e.g.,
  - DDH (Pedersen) Commitments: $g^m h^r$ where $h=g^t$ and both r and t are random.
- Classical coin tossing ideas (Blum) paired with VSS provide a simple secure multiparty computation protocol that emulates a randomness beacon.

Incentive Structure



How to incentivise parties to execute the protocol?

Introduce concept of “Input-Endorsers”

A sequence of transactions need to be endorsed in order to be included in a block.

Endorsed sequence can be included in any upcoming block up to *2k* slots in the future (inclusive).

Assumptions about protocol costs

- Our Assumptions :
  - Issuing blocks is easy (blocks contain only endorsed sequences of transactions, hence effort to verify transactions is passed to the endorsers).
- Expensive actions are:
  - Running the GOD protocol to simulate the randomness beacon. (need to issue commitments and open them)
  - Endorsing sets of transactions (need to verify them)

Reward Mechanism

- Epoch based.

    - After each epoch stabilizes, provide rewards for the following acts:
      1) being a committee member.
      2) endorsing a set of inputs.
      3) sending messages for the MPC protocol.

Approximate Nash Equilibrium Proof

- **Theorem.** Ouroboros is approximate Nash-equilibrium

- Proof: Consider a coalition of rational players that deviate from the protocol specification (while everyone else, follows the protocol).
  => no matter the strategy, chain quality ensures that endorsed inputs, and protocol messages always make it to the chain.

- Requirement: coalition should hold less than 1/2 of stake.

Dealing with online costs

- The protocol requires from a a set of stakeholders representing honest majority to be online frequently.

- We can relax this requirement, by using delegation. similar to delegative (or liquid) democracy, stakeholders can empower delegates to represent them in terms of protocol duties.

- Allows the natural formation of “stake pools” (akin to mining pools in bitcoin).

Delegation Mechanism

- Stakeholders can use the blockchain itself to assign/revoke delegation rights.
  - Simple approach: use proxy signatures.
- Committee selection works at the delegate level.
- A bound of, say, 1%, may be applied for committee participation. This ensures protocol costs can be kept low.

Prototype Implementation

- Prototype implementation in Haskell.
  - PVSS using elliptic curve crypto.
    - Digital signature is DSA.
    - Curve secp256r1 / NIST p-256 is used.
      - (the above choices are modularized and can be easily substituted).
- Geographically diverse deployment over Amazon cloud.

Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol

Aggelos Kiayias

Based joint work with Alexander Russell
Bernardo David Roman Oliynykov

for a pre-print check:
http://eprint.iacr.org/2016/889

INPUT I OUTPUT