PROOF OF STAKE

Kai Mast
CS639/839
Spring 2023

Announcements

- Midterm Thu, March 23rd 5:45pm in Bio chem 1120
- Review Session on Wed, March 22nd (usual class time)
- Project 2b will be released soon TM
- Please fill out the course evaluation
  - Any constructive feedback is welcome!
  - E.g., let me know if you find the pace and difficulty adequate

TODAY'S AGENDA

- Overview of Proof of Stake
  - Limitations of PoW
  - Challenges with PoS
- Discussion of two PoS protocols
  - Algorand
  - Ouroboros
- Final Project Topics

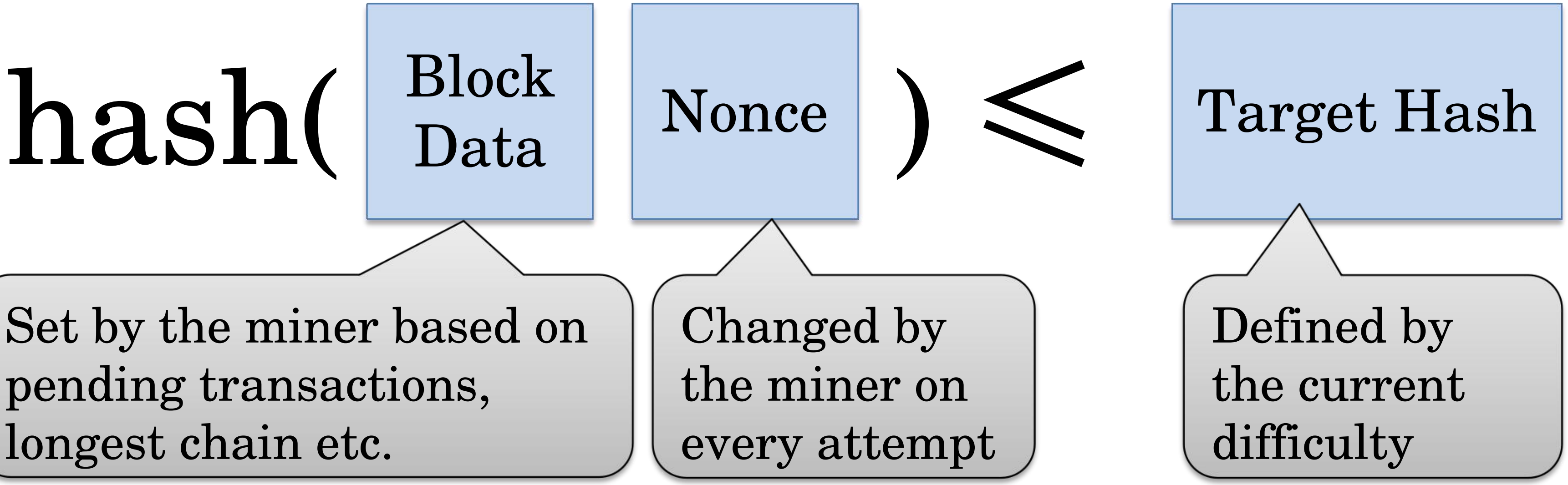
RECAP: PROOF OF WORK

Goal: Tie likelihood of generating a block to processing power

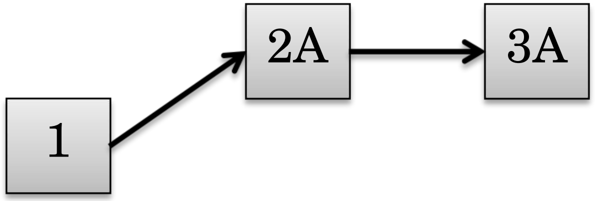
- Each node only has some finite amount of hardware

Approach: Create a very hard-to-solve task (the "crypto puzzle")

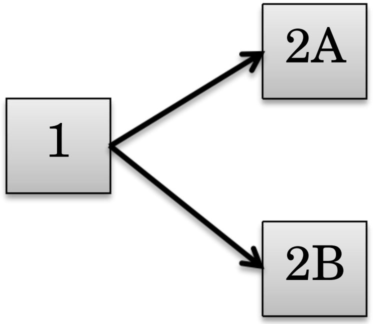
- Random tries are needed to find the solution
- We might need many attempts to solve it



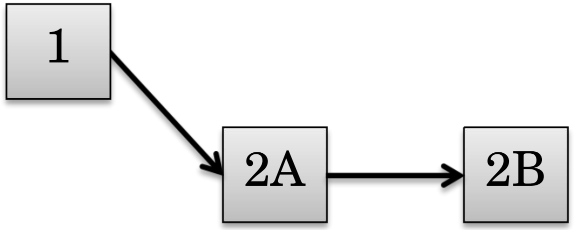
RECAP: INFORMATION ASYMMETRY



Node 1's View



Node 2's View



Node 3's View

- Each node sees some subset of all blocks
- In Bitcoin and Ethereum 1.0 there is no certain way of knowing which blocks have been seen by a majority of nodes

Why?

- Network failures and delays
- Attackers might not forward blocks

LEVERAGING INFORMATION ASYMMETRY

- Nodes that see blocks earlier have an advantage
  - Can start mining on the most recent block before others
- Nodes that do not see blocks in time have a disadvantage
  - Will mine on an outdated version of the chain
- Nodes can intentionally hide blocks
  - Selfish Mining
  - Eclipse Attacks

RECAP: ENVIRONMENTAL IMPACT OF POW

Annualized Total Bitcoin Footprints

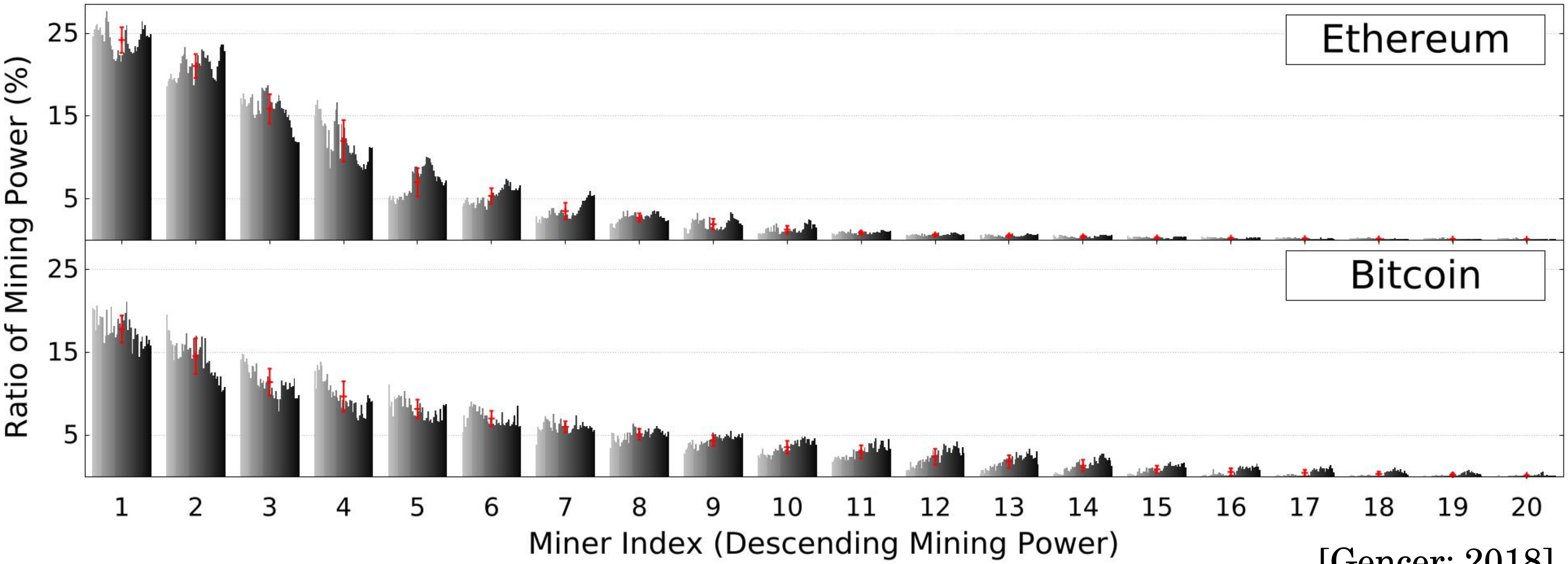
Carbon Footprint	Electrical Energy	Electronic Waste
50.83 Mt CO2	91.14 TWh	50.86 kt
Comparable to the carbon footprint of **Hungary**.	Comparable to the power consumption of **Philippines**.	Comparable to the small IT equipment waste of **the Netherlands**.

Single Bitcoin Transaction Footprints

Carbon Footprint	Electrical Energy	Electronic Waste
460.60 kgCO2	825.80 kWh	460.90 grams
Equivalent to the carbon footprint of **1,020,849** VISA transactions or **76,767** hours of watching Youtube.	Equivalent to the power consumption of an average U.S. household over **28.30** days.	Equivalent to the weight of **2.81** iPhones 12 or **0.94** iPads. (Find more info on e-waste here.)

Source: HTTP://economist.net/bitcoin-energy-consumption

CENTRALIZATION IN PROOF OF WORK



[Gencer; 2018]

In 2017

- Bitcoin: over 50% of mining power controlled **by four miners**
- Ethereum: over 50% of mining power controlled **by three miners**

CENTRALIZATION IN PROOF OF WORK

Reasons:

- More efficient to operate mining pools at large scale
  - Some fixed cost, e.g., cooling, easier to amortize

- Large mining pools have a more reliable revenue stream
  - Small miners may not find blocks for a long time

- Miners see their own blocks first
  - More likely that their next block will be part of the winning chain

PROOF OF STAKE

Idea: Assign voting power by stake, not mining power

- Stake is the amount of currency held by a particular entity

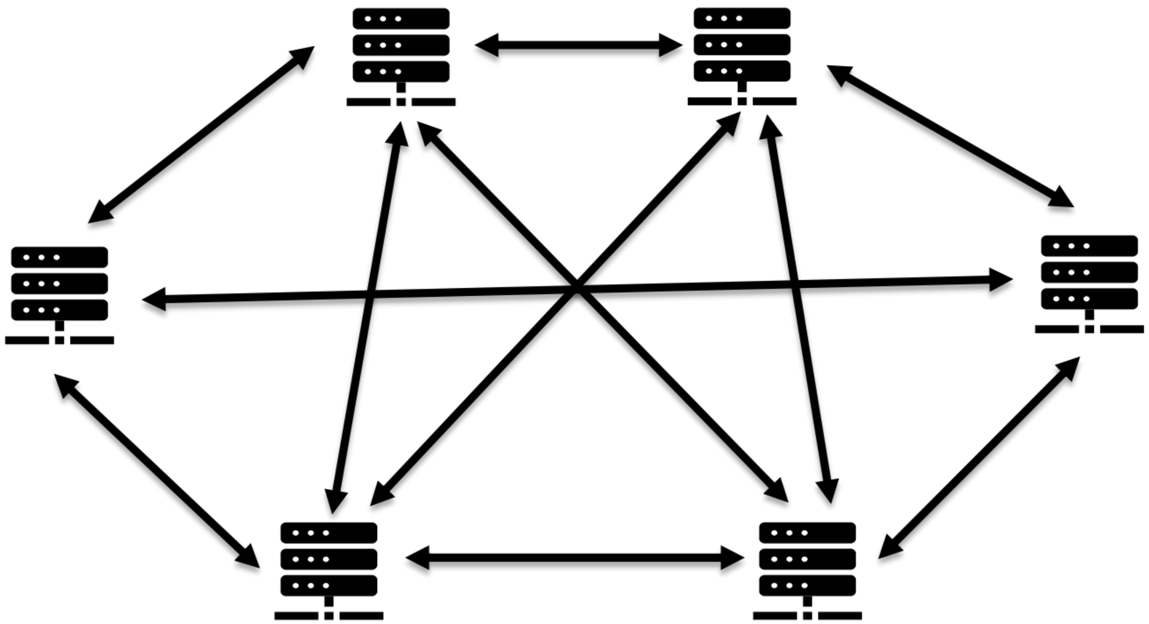
Challenge 1: How to pick block creators?

- We need some kind of randomness
- True randomness is hard to generate in the blockchain setting
- Attackers might try to influence the random number generation *(grinding attack)*

Challenge 2: *Nothing at Stake*

- Block creation is computationally cheap
- Easy for an attacker to try to create many blocks

PERMISSIONED CHAINS



Simplest version of Proof of Stake

- Fixed committee: Set of stakers always stays the same
- Each committee member has the same voting power

Generalizing Proof of Stake

Support **varying voting power**

- Either total balance of an entity or staked balance
- Staked balance: Need to lock up some money to be used for staking
  - Simpler to implement but less flexible

Support **delegation**

- Not everyone might have the resources to participate in consensus
- Allow for “stake pools”

POS-BASED APPROACHES

Randomize Schedule, e.g, Ouroboros
- Time is split into fixed-size slots
- Set a sequence of block creators in advance, each responsible for one slot

Random Selection, e.g., Algorand
- Time is split into fixed-size slots
- Every node has some chance to be part of the committee of a block

Random Sampling, e.g., Avalanche
- Ask other nodes about which transaction they have accepted
- Eventually converge to the same set of accepted transactions
- More about this in another lecture

Always: Voting power (or chance to be selected) is proportional to stake

Algorand

- Developed by Silvio Micali and others at MIT
- First published in 2017 at SOSP
- Main network launched in 2019

SYNCHRONICITY & FAILURE TYPES

Protocols are designed against a particular **synchronicity model**

For now, simplest case: *synchronous networks*

- Messages are never lost
- Messages are delivered within a known time bound

Protocol are designed against a particular **failure model**

For now, a fairly simple case: *crash failures*

- Nodes are bug free and honest
- Crashes can still happen

SIMPLIFIED ALOGRAND

- No Byzantine Failures
- Synchronous network
- Permissioned

A SYNCHRONOUS PERMISSIONED PROTOCOL

Time is split into fixed size slots (or rounds)

- Slots are larger than the maximum network delay
- All messages sent at the beginning of a slot, reach all nodes at the end of the slot

At the beginning of a slot, each node proposes at most one block per slot

- Each node has the same "voting power"

If we receive multiple blocks per slot, we have a *tiebreaker*

- Tiebreaker can be computed, e.g., by combining slot number and node id
  - H(slot_num | node_id)
- All nodes accept at most one block per slot

Simple one-round protocol: No forks possible

SIMPLIFIED ALOGRAND

- ~~**No Byzantine Failures**~~
- Synchronous network
- Permissioned

ADDING BYZANTINE FAILURES

Problem: Faulty nodes might propose conflicting blocks

- Attacker might not send block to all nodes
- Simple tiebreaker is not sufficient

Protocol now needs **three steps**

- Proposal: Each node can propose a block
  - Honest nodes will pick the block with the highest tiebreaker

- Reduction: Nodes broadcast which block they have accepted
  - Allows detecting if an attacker proposes multiple blocks at once
  - If a node receives the same block from a majority (2/3), start BA with that block
  - Otherwise, start BA with the *empty-block*

- Binary Agreement: Decide between a proposed block or *empty-block*
  - Need 2/3 majority to agree on a block

SIMPLIFIED ALOGRAND

- ~~No Byzantine Failures~~
- ~~**Synchronous network**~~
- Permissioned

LOOSENING NETWORK ASSUMPTIONS

The last few slides: Synchronous Network

- Known time bound for message delivery

Most realistic: Asynchronous

- No bounds on network delay
- Very hard, but not impossible to support

A compromise: Partial Synchrony

- Generally the network behaves synchronously
- Sometimes there might be a network partition
  - Can last any amount of time, but eventually the network will be synchronous again
  - Protocol will not make any progress during that time

A PARTIALLY SYNCHORNOUS PROTOCOL

- We might not reach final consensus on a block for every round
  - Some nodes might accept a block tentatively

- Tentative blocks are considered final if one of their ancestors are considered final
  - This means we can have forks

- Need to vote on competing forks using the same mechanism as voting on competing blocks
  - Network partition will eventually end and the network converges on a single chain

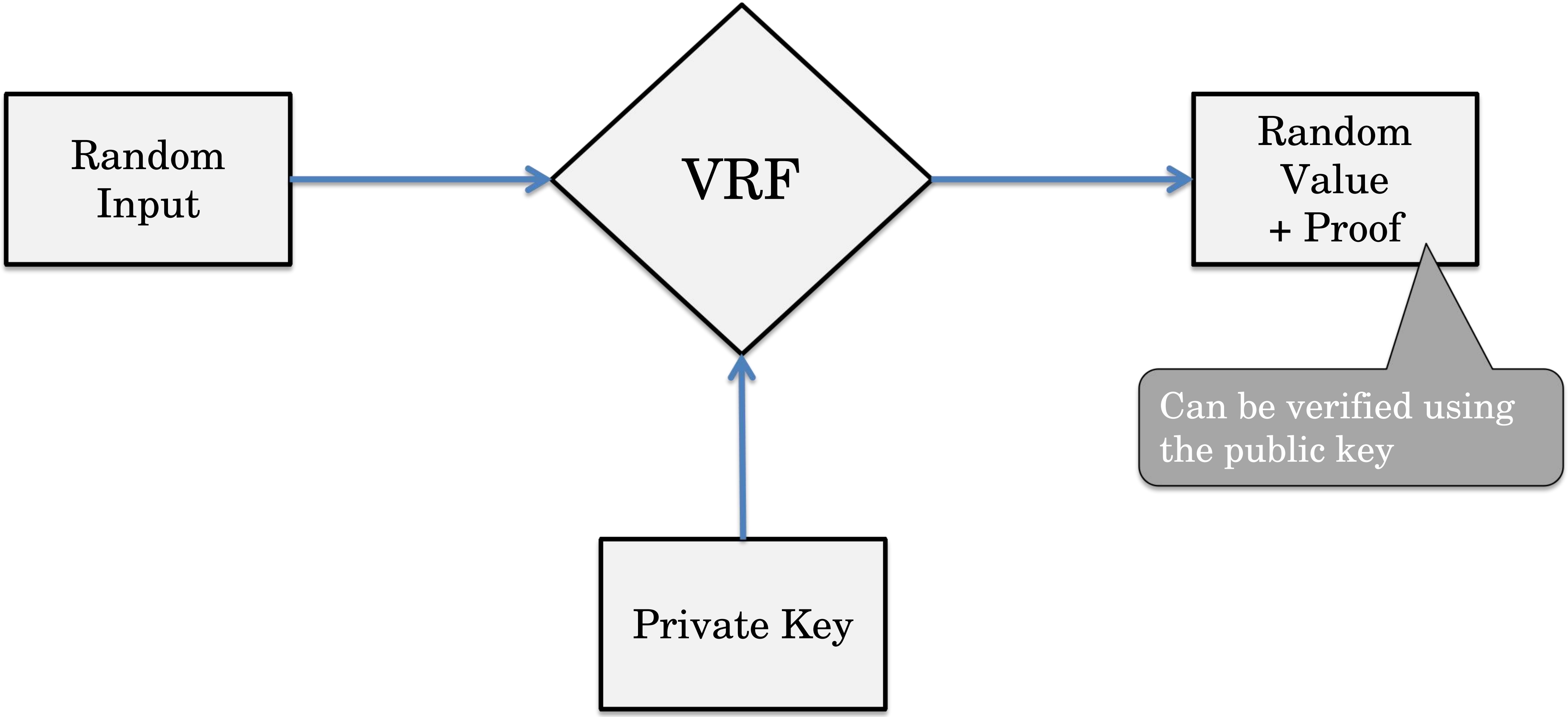
SIMPLIFIED ALOGRAND

- ~~No Byzantine Failures~~
- ~~Synchronous network~~
- ~~**Permissioned**~~

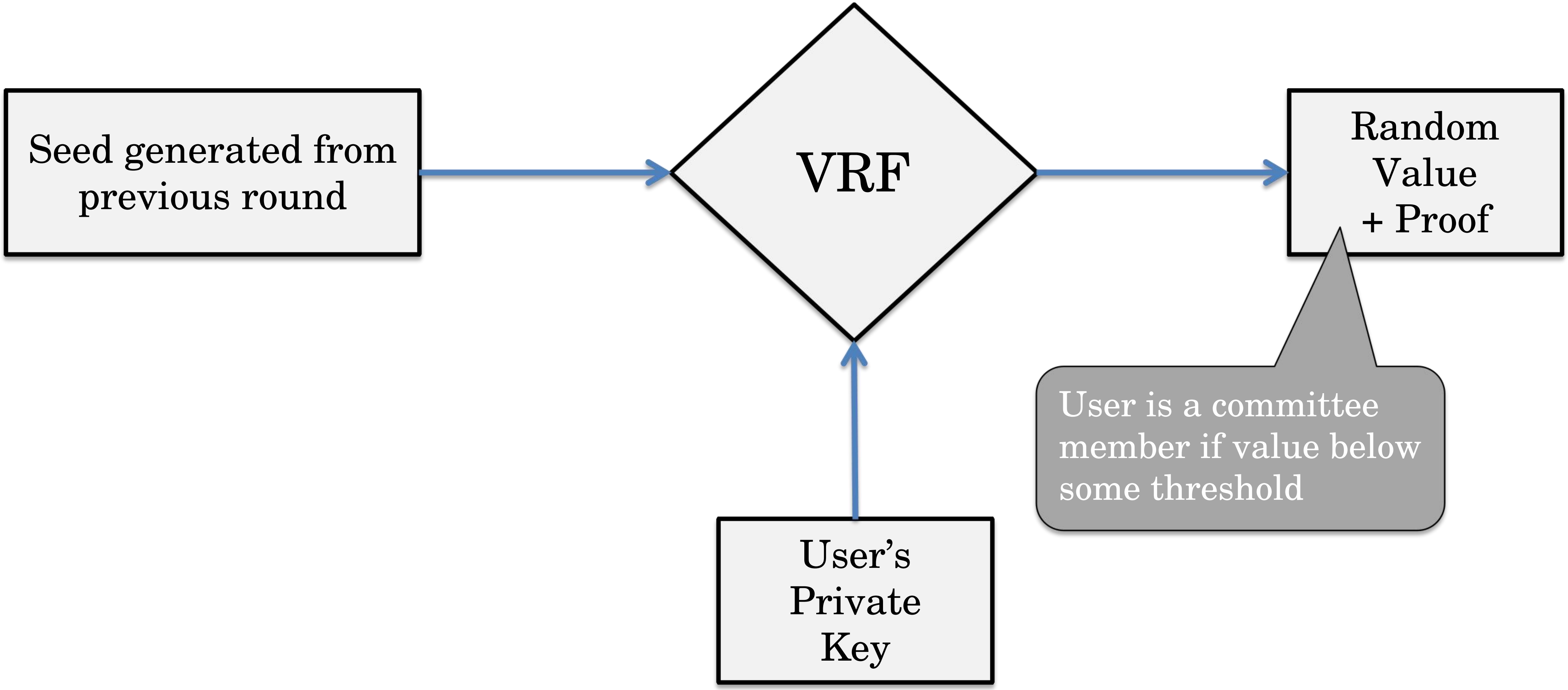
MAKING THE PROTOCOL PERMISSIONLESS

- So far, fixed set of validators
  - Not a public/permissionless system!
  - No stake, everyone has the same voting power
- We need to randomly pick membership
  - Committee should be a weighted random subset of all stakers
  - Weighted by stake
- Not all nodes should be able to create blocks
  - Creates a lot of unneeded network traffic
  - A smaller subset of the stakers are block proposers

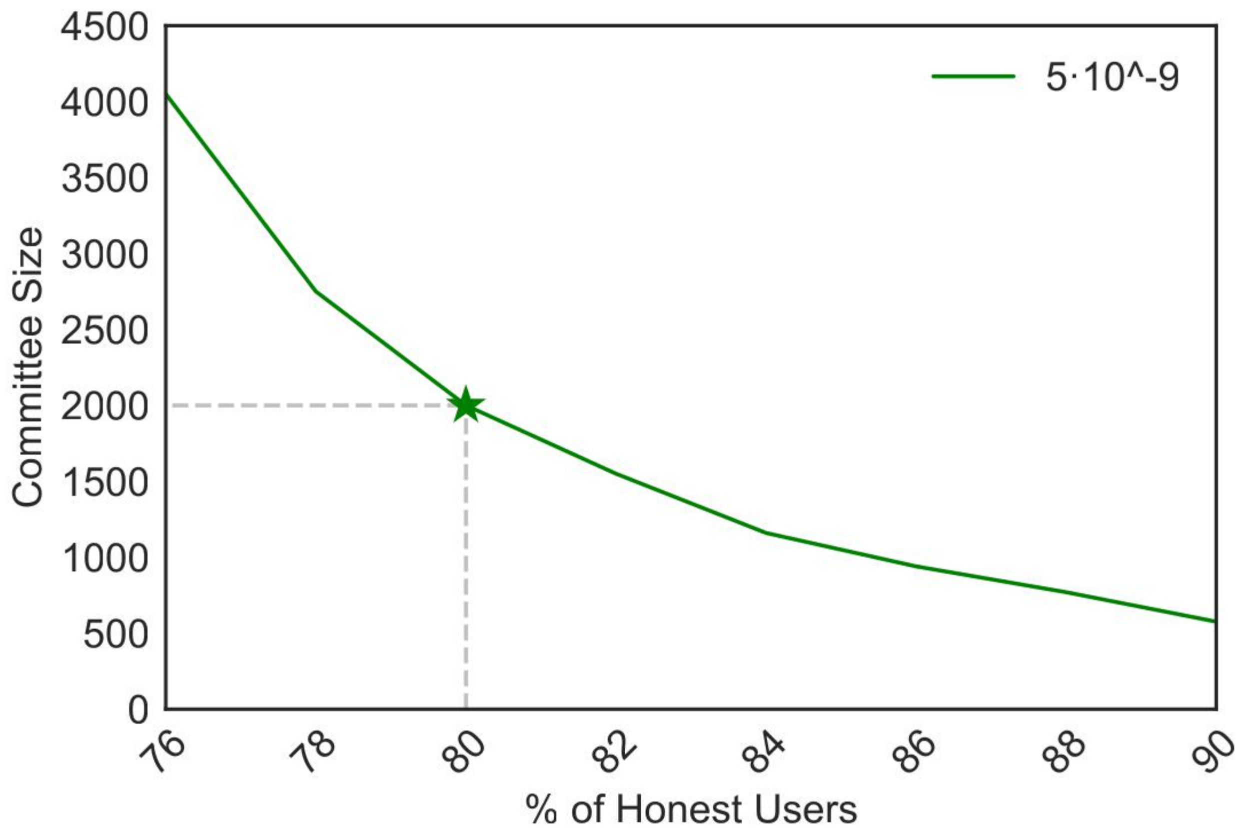
VERIFIABLE RANDOM FUNCTIONS



VRFS IN ALGORAND

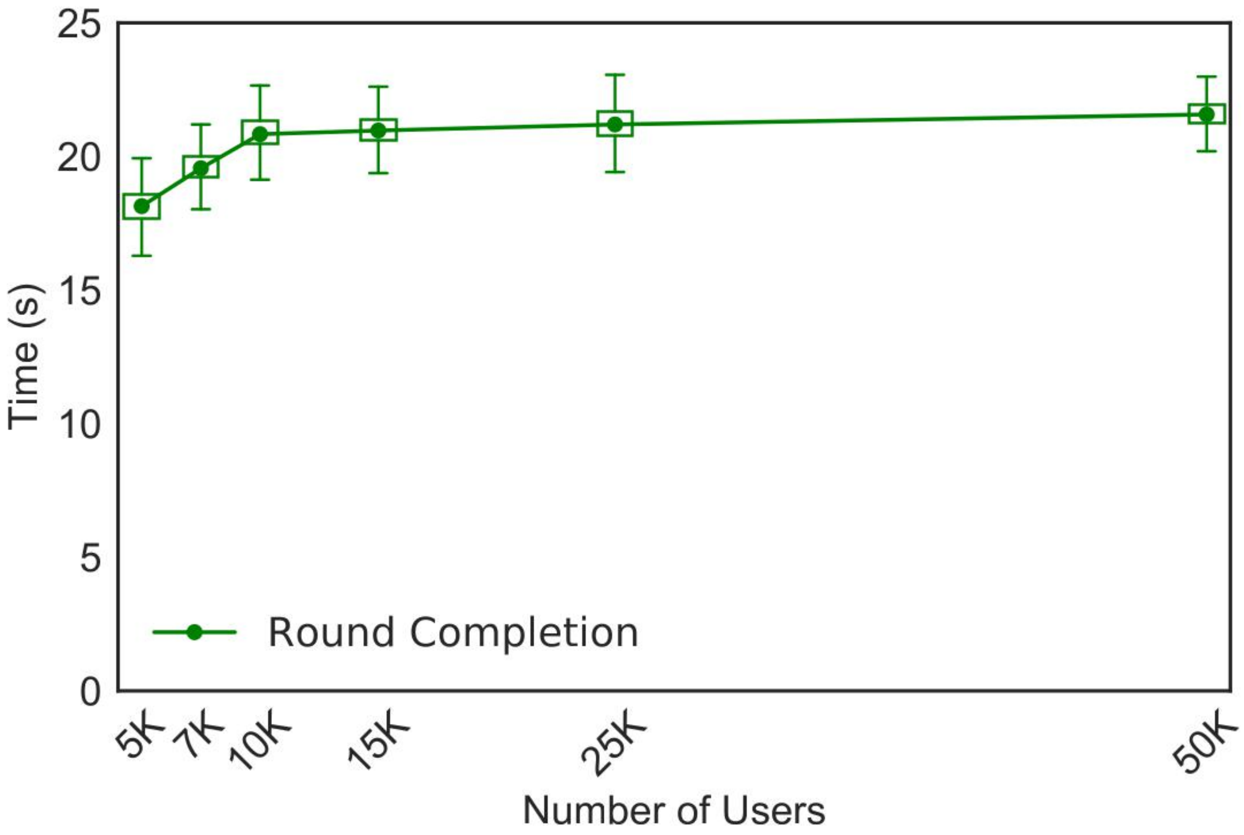


ALGORAND COMMITTEE SIZE



We need a large committee to ensure at least 2/3 are honest

ALOGRAND PERFORMANCE



- Measured on a geo-replicated network
- Algorand confirms blocks in less than 25s

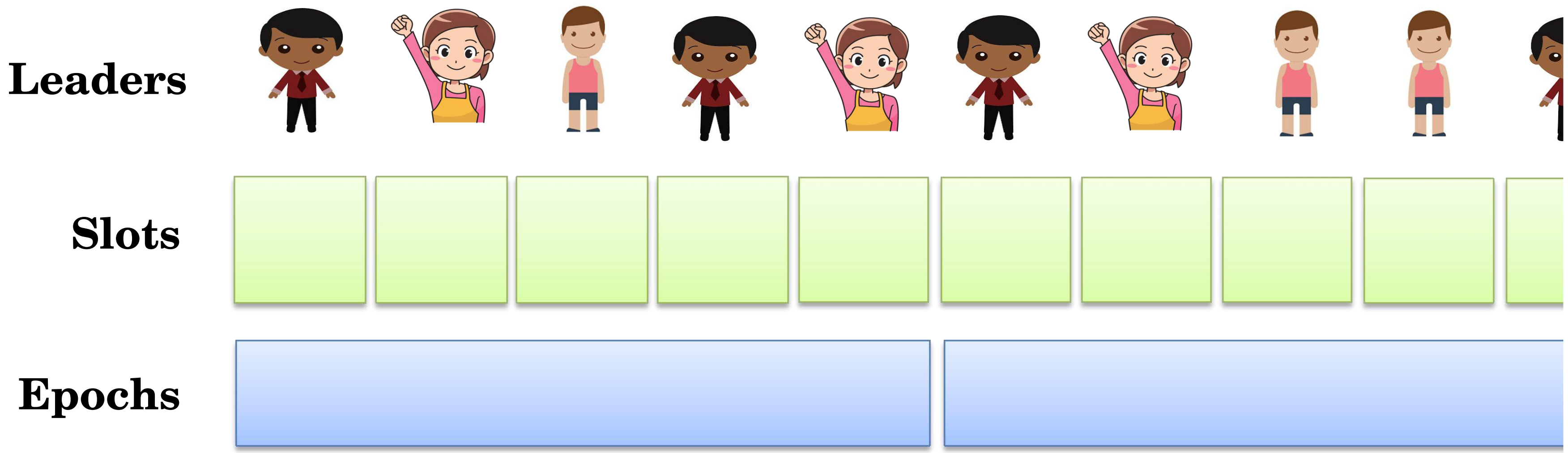
BREAK?

OUROBOROS

- The first PoS protocol that is provably correct
- First presented at CRYPTO 2017
- Basis for the Cardano blockchain
- Developed by folks at IOHK and University of Edinburgh
- We only discuss the most basic variant today

CARDANO

TIMING ASSUMPTIONS IN OUROBOROS



- Time is split into slots
- Slots are grouped into epochs

Network is synchronous
- Each block will be visible to all correct nodes at the end of a slot

(*not true for all versions of Ouroboros)

EPOCHS IN OUROBORS

An epoch consists of some fixed number of slots

At the beginning of an epoch

- Stake is updated depending on state changes in the previous epoch
- Randomness is generate through multi-party computation
  - out of the scope of this lecture
- Use randomness and state to **generate a leader schedule**
  - relies on VRFs, like Algorand

LEADERS IN OUROBOROS

- There is a pre-defined leader schedule for each epoch, but leaders can be faulty.
- There is exactly one leader (block creator) per slots

Honest Leaders:

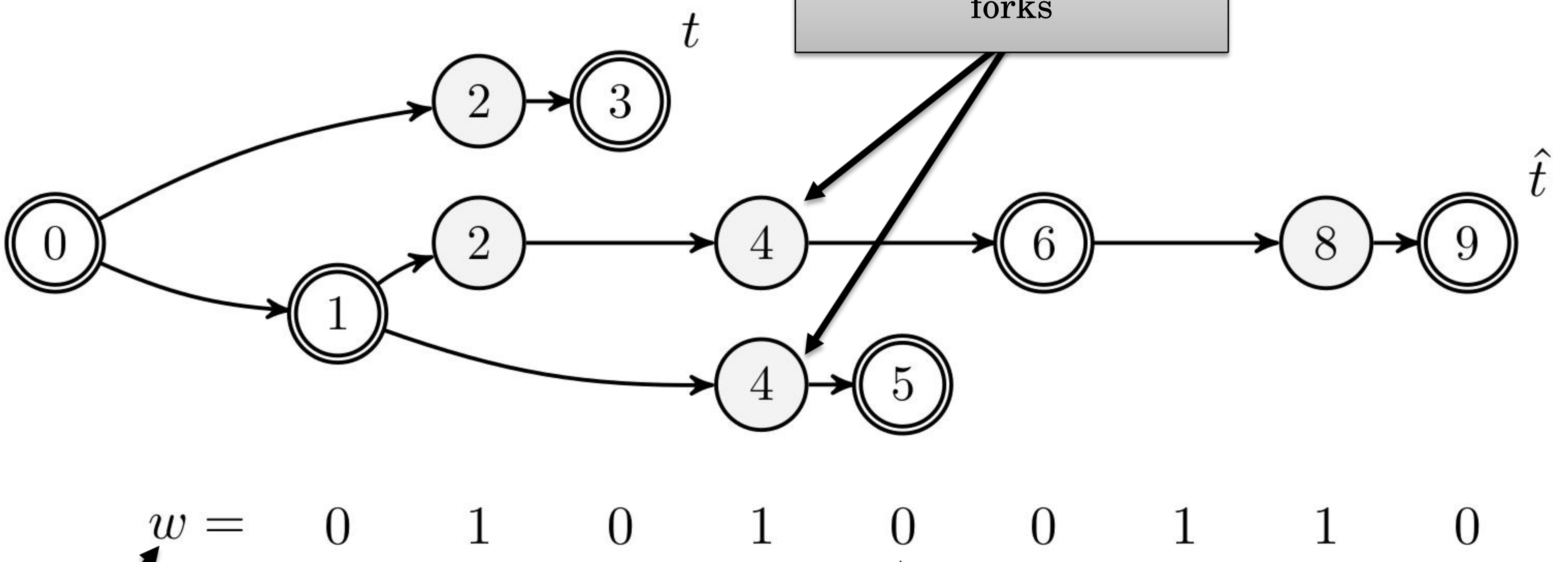
- Will always extend the longest chain
- Create at most one block

Faulty Leaders:

- May attempt to extend multiple forks in one slot
- May hide block its mines (*covert adversary*)

FORKS AND FORKABLE STRINGS

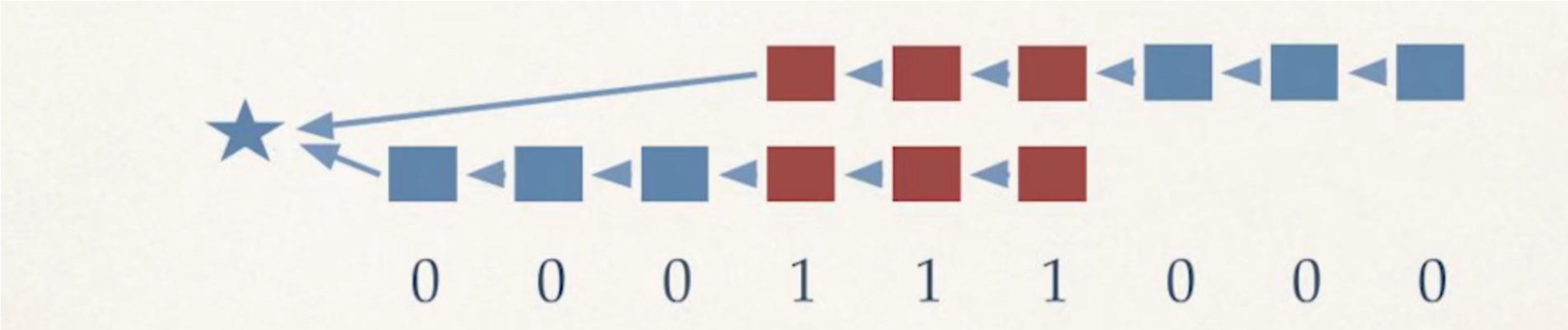
Malicious leader may append to multiple forks



String encodes leader schedule (whether leader is malicious)

Density of string is equal to the voting power of attackers (here 4/9)

FORKABLE STRINGS CONT.



A leader schedule (or "string") is **forkable** if the adversary can produce two disjoint paths with the same length.

- Forkable strings are impossible if density is <1/3
- In the paper they show prevention against adversaries as large as <1/2

(from Peter Gaži's talk at MIT)

OUROBOROS CONFIRMATION DELAY

Attack Power
(as a fraction of the total stake or mining power)

General adversary includes covert attacks

Adversary	BTC	OB Covert	OB General
0.10	50	3	5
0.15	80	5	8
0.20	110	7	12
0.25	150	11	18
0.30	240	18	31
0.35	410	34	60
0.40	890	78	148
0.45	3400	317	663

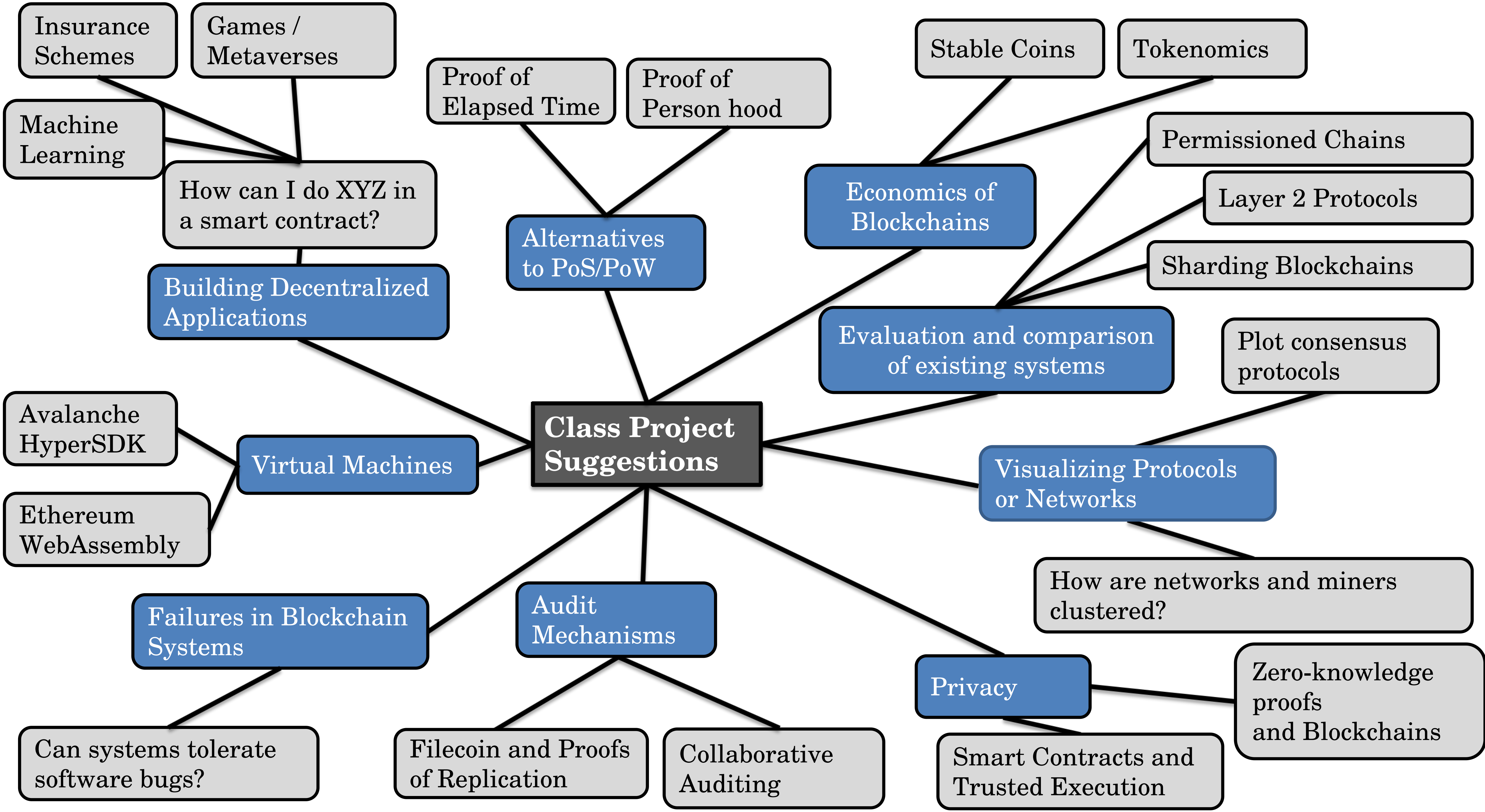
Time (in minutes) after which transaction is finalized with at least 99.9% certainty

POS: SUMMARY

Advantages
- Vastly less energy consumed
- Can be more decentralized

Disadvantages
- Not fully permissionless
- Protocols are generally more complicated
  - More potential for bugs and exploits

More on Proof of Stake in the next two lectures!


Class Project Suggestions
Building Decentralized Applications
How can I do XYZ in a smart contract?
Insurance Schemes
Games / Metaverses
Machine Learning
Alternatives to PoS/PoW
Proof of Elapsed Time
Proof of Person hood
Economics of Blockchains
Stable Coins
Tokenomics
Evaluation and comparison of existing systems
Permissioned Chains
Layer 2 Protocols
Sharding Blockchains
Visualizing Protocols or Networks
Plot consensus protocols
How are networks and miners clustered?
Virtual Machines
Avalanche HyperSDK
Ethereum WebAssembly
Failures in Blockchain Systems
Can systems tolerate software bugs?
Audit Mechanisms
Filecoin and Proofs of Replication
Collaborative Auditing
Privacy
Zero-knowledge proofs and Blockchains
Smart Contracts and Trusted Execution