Lecture 11: Proof of Stake

https://web3.princeton.edu/principles-of-blockchains/

Professor Pramod Viswanath
Princeton University

This lecture:
PoW is energy inefficient;
PoS: Energy efficient alternative;
PoS version of longest-chain protocol

This lecture

Proof of Stake (PoS)

energy efficient alternative
replacement to PoW

Simple way to implement PoS

within the longest chain protocol

Vulnerabilities

nothing at stake (NaS) attack
grinding attack

Proof of Work

Find nonce such that

H(hash(parent.header), Merkle root of tx, nonce) < threshold

Finding entails work
Nonce is the proof

Doing this work allows miners to participate meaningfully in the protocol

PoW is a Sybil and spam resistant leader election mechanism

Proof of Stake

Allow meaningful participation based on stake
    block proposers own coins

Level of participation proportional to stake
    higher probability of being a proposer

Doing work is replaced by owning coins
    energy efficient
    capital efficient -- no need for mining hardware

Idea 1

PoS attempt 1

H(hash(parent.header), Merkle root of tx, public key) < threshold x stake

Problem: Grinding

can try different set of tx such that Merkle root of tx works out "correctly"

so probability is not purely proportional to stake

Idea 2

PoS attempt 2

H(hash(parent.header), public key) < threshold x stake

Got rid of transaction hash

Problem: Liveness

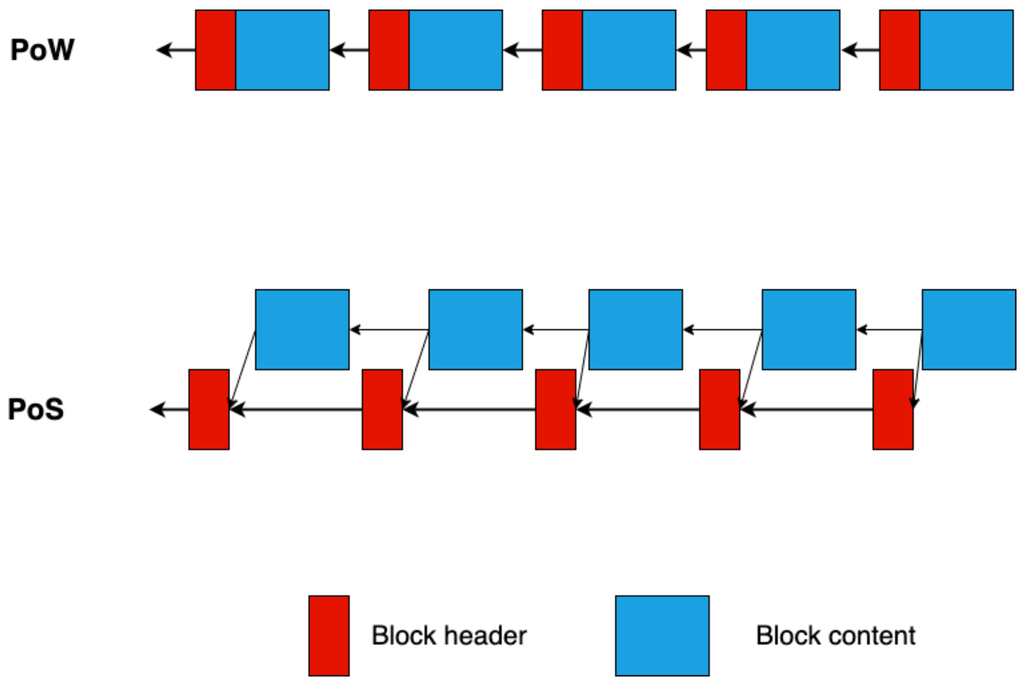
just one trial to form a block

unlike PoW where nonce can be tried at will

Idea 3

PoS attempt 3

H(hash(parent.header), timestamp, public key) < threshold x stake



Idea 3

PoS attempt 3

H(hash(parent.header), timestamp, public key) < threshold x stake

Problems:

1. Block content is not tamper-resistant against an adaptive adversary
2. Public election: vulnerable to bribery/corruption, not resistant to adaptive adversary

Crypto Primitive 1

Key-Evolving Signatures (KES) are signature schemes, where:

- ➢ pk remains the same
- ➢ sk updated in every step, old sk erased
- ➢ impossible to forge old signatures with new keys

- used for signing blocks
- helps achieve adaptive security

attacker corrupts old blocks sometime in the future

Crypto Primitive 2

Verifiable Random Function (VRF)

VRF(sk , x) $\rightarrow$ (y , π)

Verify(pk , x , y , π) $\rightarrow$ True/False

- used for signing blocks
- helps achieve adaptive security

attacker corrupts upcoming blocks well in advance

Idea 4

PoS attempt 4

VRF(hash(parent.header), timestamp, secret key) < threshold x stake

Attack: Nothing at Stake

VRF(hash(parent.header), timestamp, secret key) < threshold x stake

Longest chain rule

    parent is tip of longest chain

Adversary deviates

    can grow on all blocks (even Genesis)

No computation limit to deviation

    unlike PoW

    Nothing at Stake (NaS)

NaS Tree

Honest participants

grow chain as a Poisson process

growth rate linear in time

Adversary

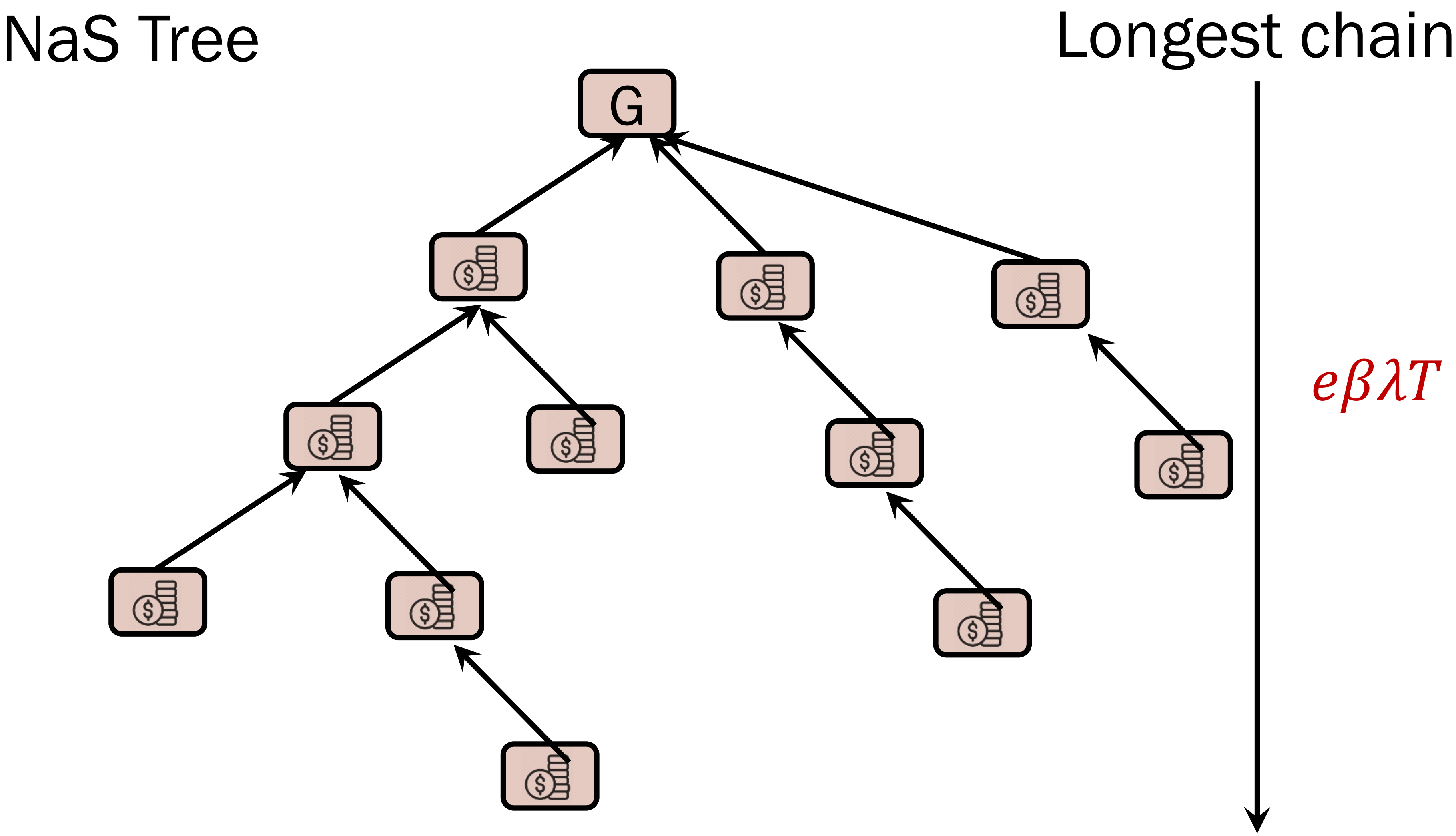
grows a tree of blocks

number of blocks grows exponentially in time

NaS

allows adversary to compete with honest participants unevenly

NaS Tree and Longest Chain



A scalable PoS blockchain in the open setting, Fan and Zhou, 2016

Security of PoS Longest Chain

Honest participants

grow chain as a Poisson process

growth rate linear in time $(1-\beta)\lambda T$

Adversary

grows a NaS tree in private

longest chain length $e\beta\lambda T$

Security against Private attack

$(1-\beta)\lambda T > e\beta\lambda T$ or $\beta < \frac{1}{1+e} \approx 27\%$

Security

Security against Private attack

$$\beta < \frac{1}{1+e} \approx 27\%$$

Secure against all attacks

Cornell University

the Sim

arXiv.org > cs > arXiv:1910.02218

Search...

Help | Advar

Computer Science > Cryptography and Security

Proof-of-Stake Longest Chain Protocols: Security vs Predictability

Vivek Bagaria, Amir Dembo, Sreeram Kannan, Sewoong Oh, David Tse, Pramod Viswanath, Xuechao Wang, Ofer Zeitouni

(Submitted on 5 Oct 2019 (v1), last revised 23 Feb 2020 (this version, v3))

The Nakamoto longest chain protocol is remarkably simple and has been proven to provide security against any adversary with less than 50% of the total hashing power. Proof-of-stake (PoS) protocols are an energy efficient alternative; however existing protocols adopting Nakamoto's longest chain design achieve provable security only by allowing long-term predictability (which have serious security implications). In this paper, we prove that a natural longest chain PoS protocol with similar predictability as Nakamoto's PoW protocol can achieve security against any adversary with less than 1/(1+e) fraction of the total stake. Moreover we propose a new family of longest chain PoS protocols that achieve security against a 50% adversary, while only requiring short-term predictability. Our proofs present a new approach to analyzing the formal security of blockchains, based on a notion of adversary-proof convergence.

Comments: 65 pages, 16 figures

Subjects: **Cryptography and Security (cs.CR)**

Cite as: arXiv:1910.02218 [cs.CR]

(or arXiv:1910.02218v3 [cs.CR] for this version)

Bibliographic data

[Enable Bibex (What is Bibex?)]

Submission history

From: Xuechao Wang [view email]

Boosting Security Threshold

Security against all attacks

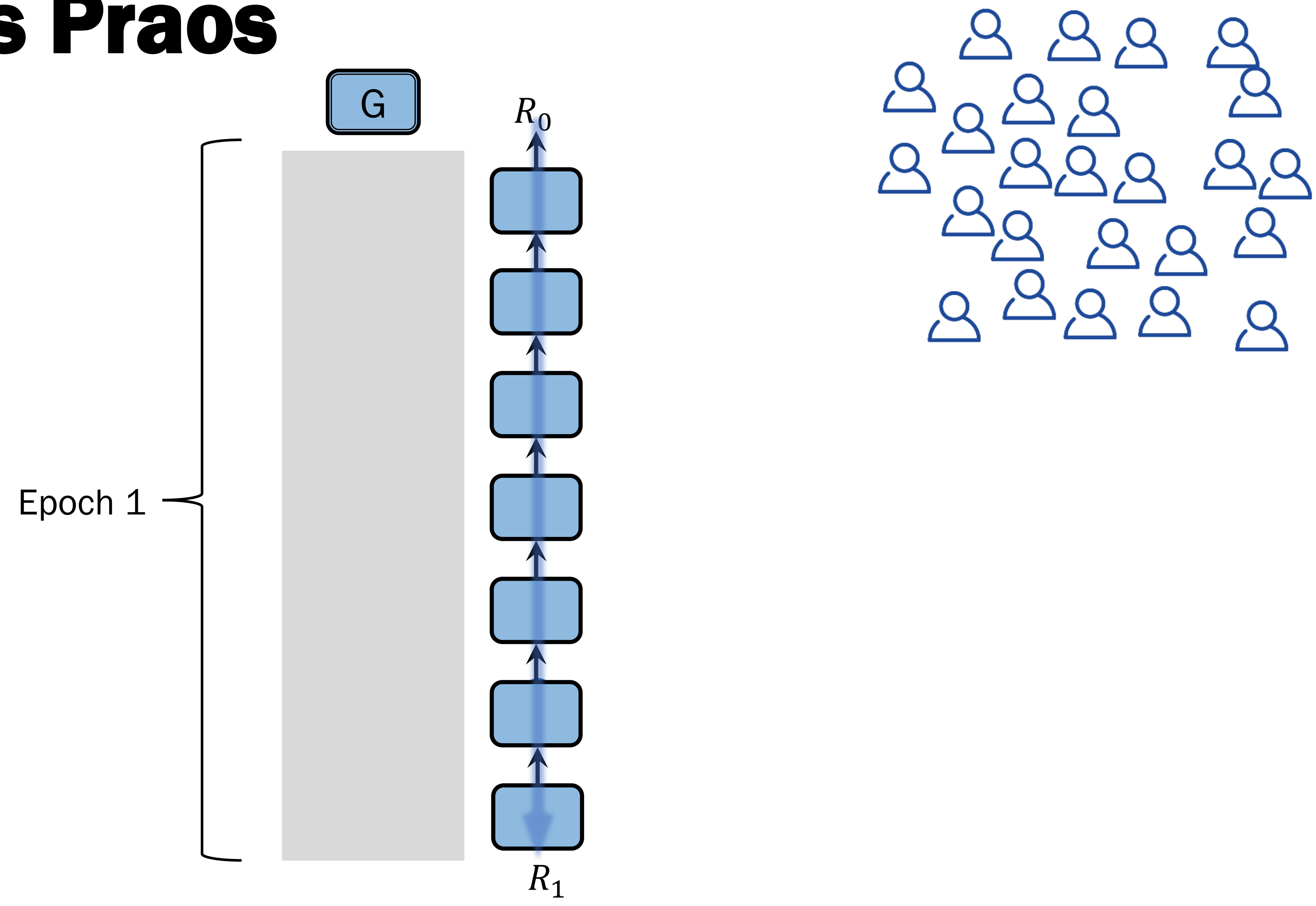
$$\beta < \frac{1}{1+e} \approx 27\%$$

Key idea: Reduce the number of randomness sources

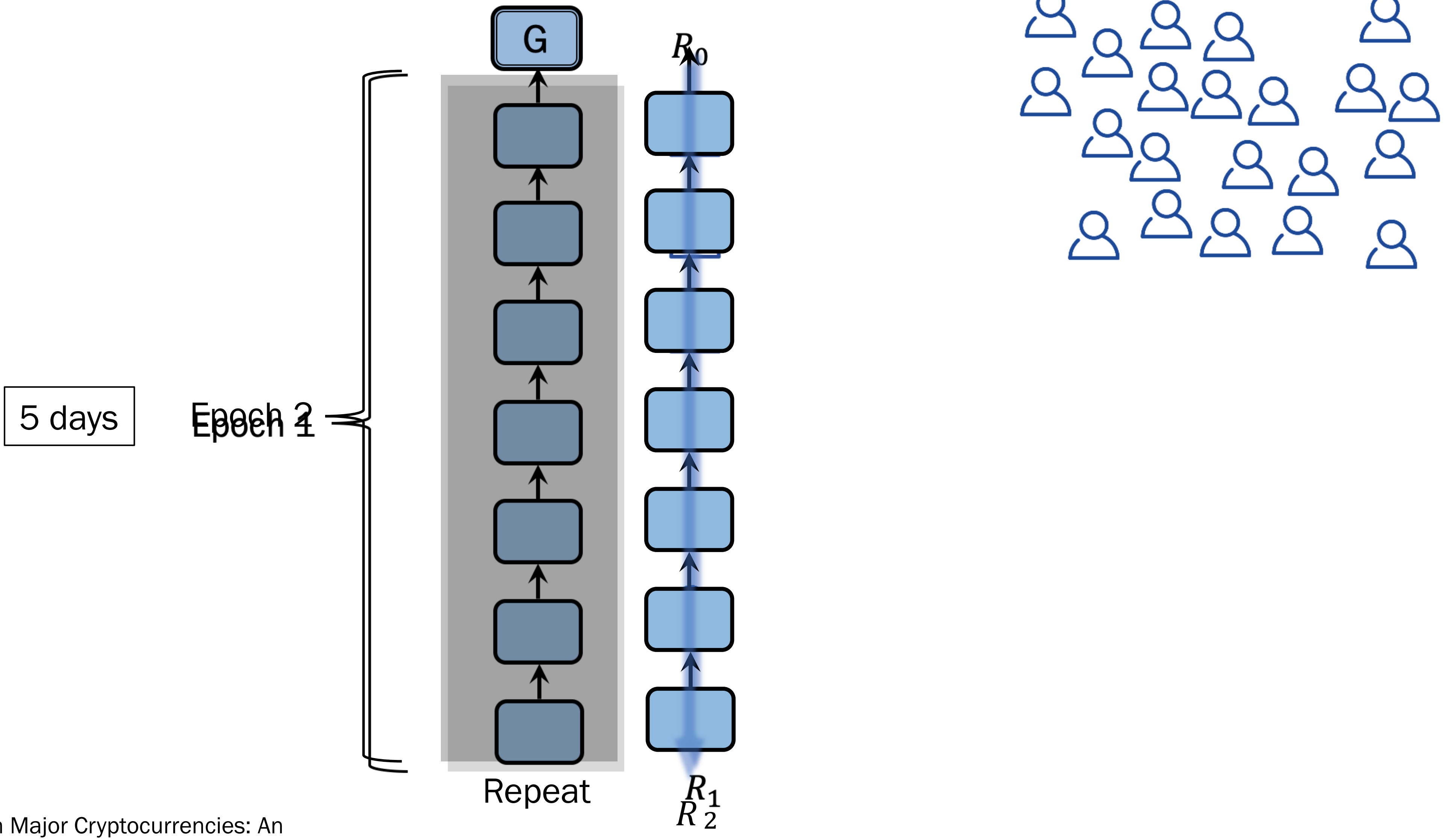
Idea 1: Only use randomness from genesis block

VRF(hash(Genesis), timestamp, secret key) < threshold x stake

Ouroboros Praos

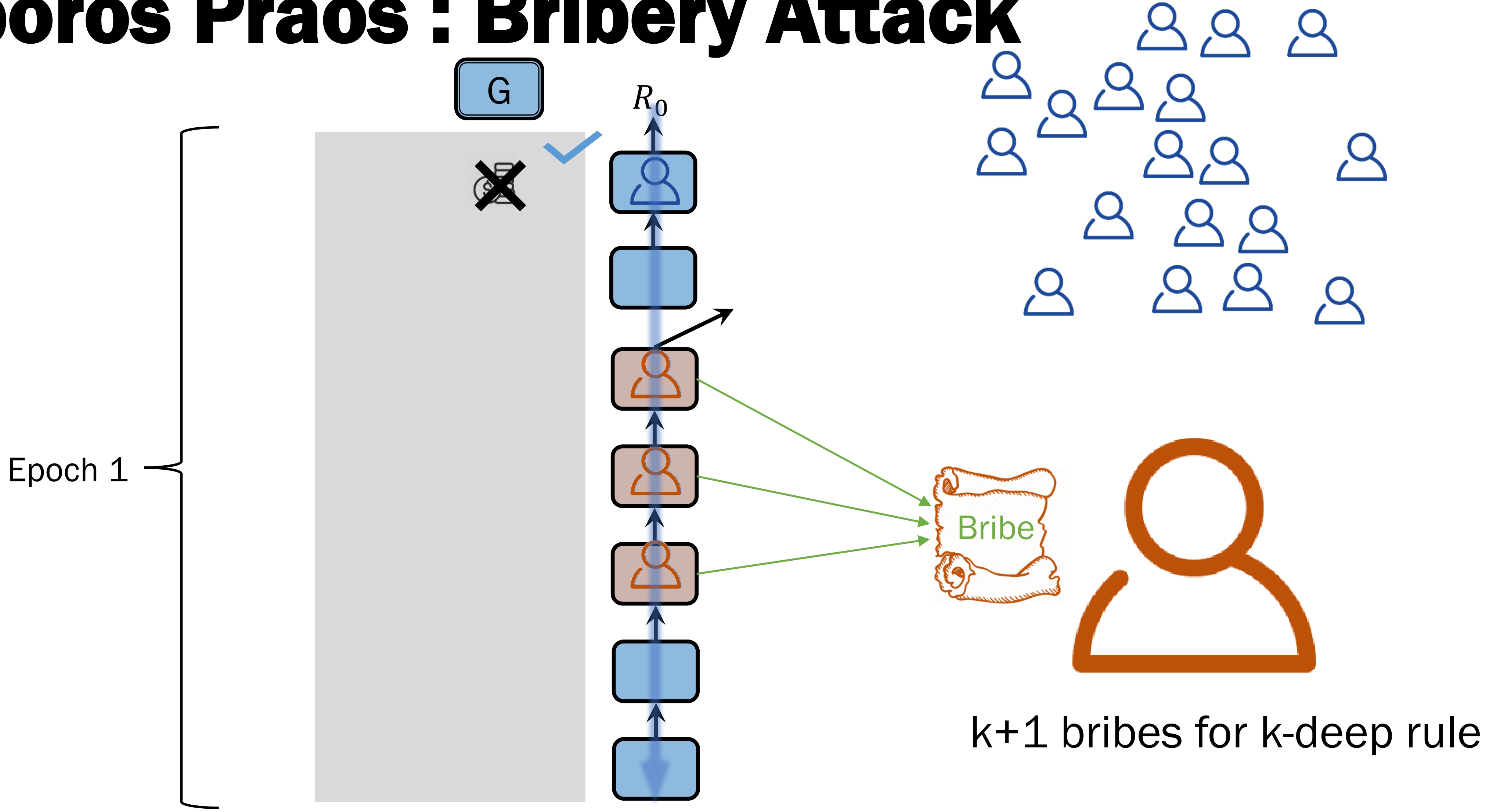


Ouroboros Praos



Stütz, Rainer, et al. "Stake Shift in Major Cryptocurrencies: An Empirical Study." *arXiv preprint arXiv:2001.04187* (2020).

Ouroboros Praos : Bribery Attack



Boosting Security Threshold

Security against all attacks

$$\beta < \frac{1}{1+e} \approx 27\%$$

Key idea: Reduce the number of randomness sources

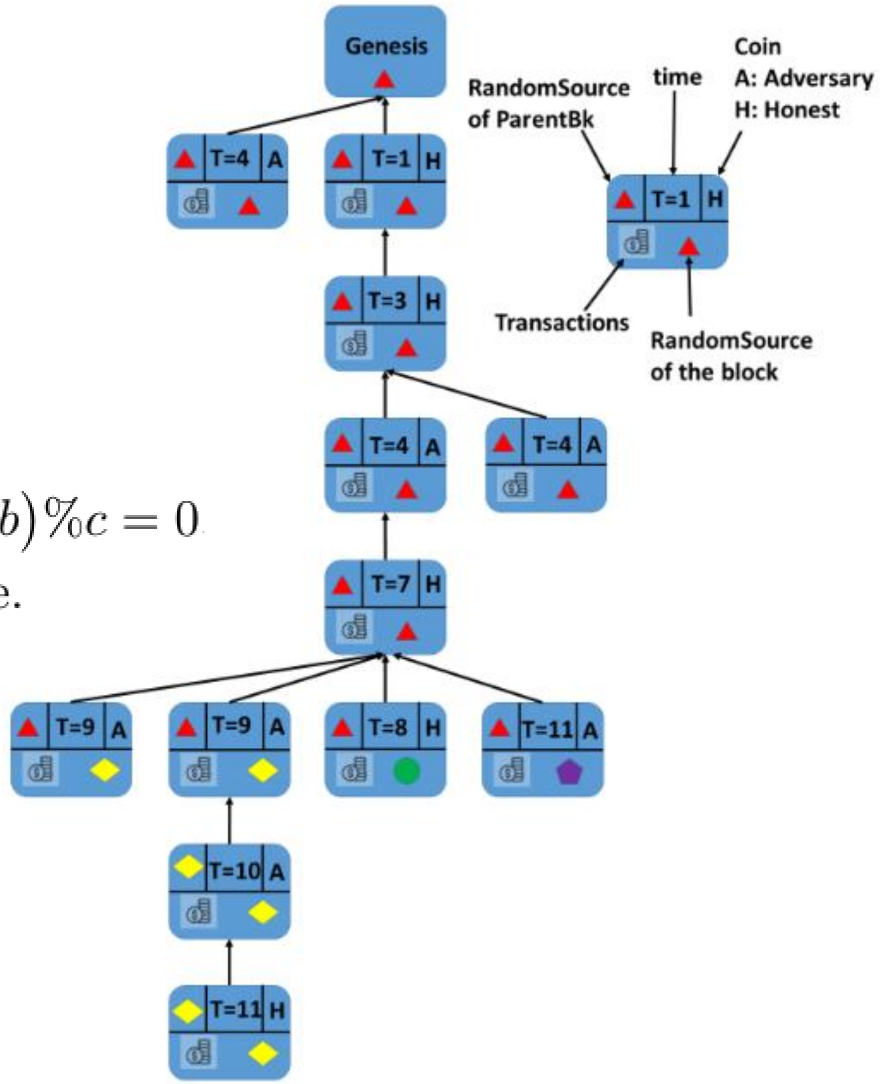
Idea 2: c-correlation

c-correlation

- Update the randomness of a block only at blocks with height multiples of c

$$\mathrm{RandSource}(b) := \begin{cases} \mathrm{VRF}(\mathrm{RandSource}(\mathrm{parent}(b)), \mathrm{ts}, \mathrm{sk}), & \text{if } \mathrm{depth}(b) \% c = 0 \\ \mathrm{RandSource}(\mathrm{parent}(b)), & \text{otherwise.} \end{cases}$$

$$\mathrm{VRF}(\mathrm{RandSource}(\mathrm{parent}), \mathrm{ts}, \mathrm{sk_n}) < T \cdot \mathrm{stake_n}$$

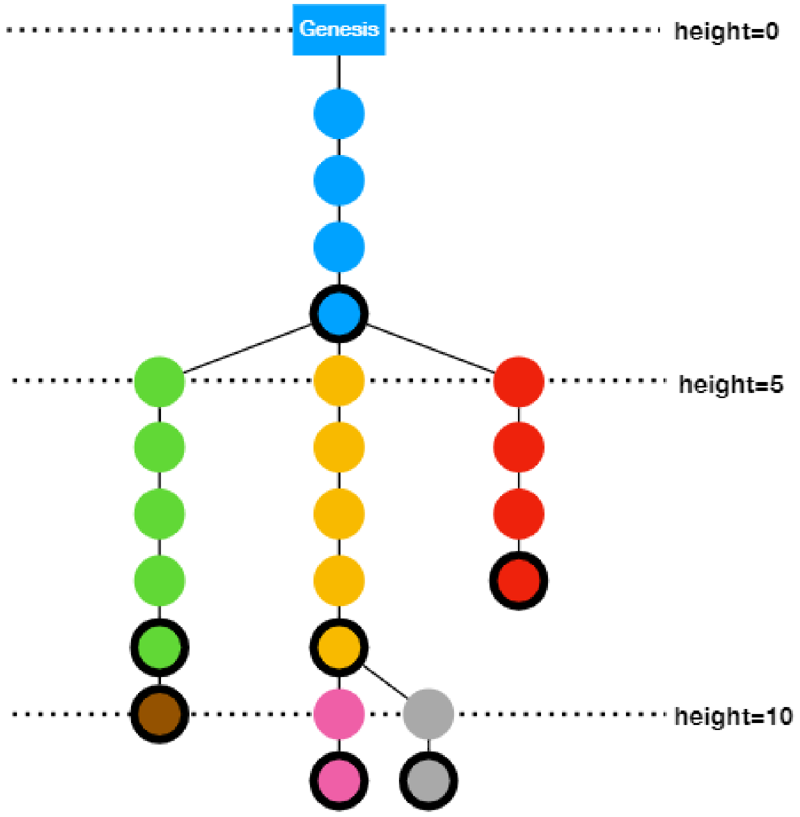


Analysis of private NaS

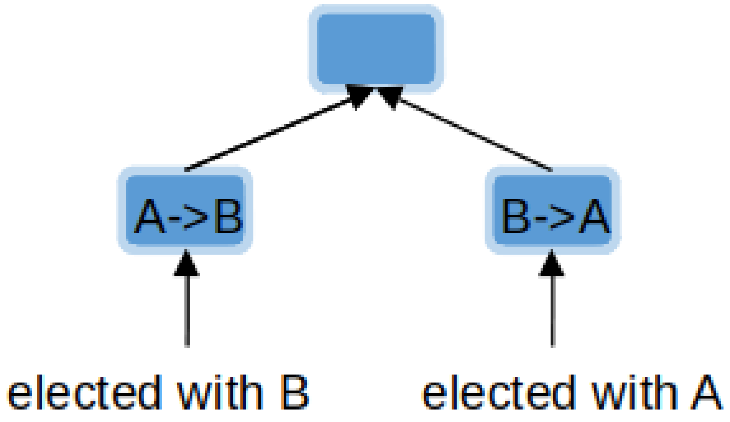
- Godfather-block: height(b)%c=0
- Only fork at the parents of godfather-blocks

c	1	2	3	4	5	6	7	8	9	10
ϕ_c	e	2.22547	2.01030	1.88255	1.79545	1.73110	1.68103	1.64060	1.60705	1.57860
β_c	$\frac{1}{1+e}$	0.31003	0.33219	0.34691	0.35772	0.36615	0.37299	0.37870	0.38358	0.38780

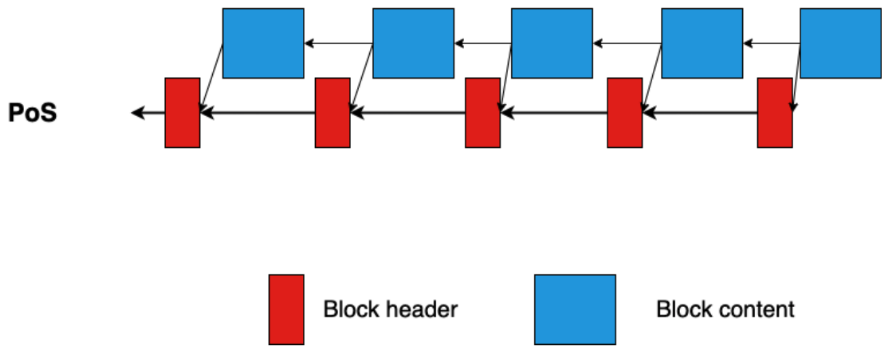
$\beta_c=1/(1+\phi_c)$: threshold for private NaS attack



Dynamic stake



- Flaw of static stake
  - Prevent nodes from leaving and joining
  - A coin with no actual stake can participate
- What if stake is updated immediately?
  - Grinding attack: once the adversary is elected as a leader at round i, it can include transactions at round i to transfer all stake to a coin that has a higher chance of winning the election at round i + 1
- What if stake is updated with a delay of s blocks?
  - Long range attack: have a private chain with s blocks

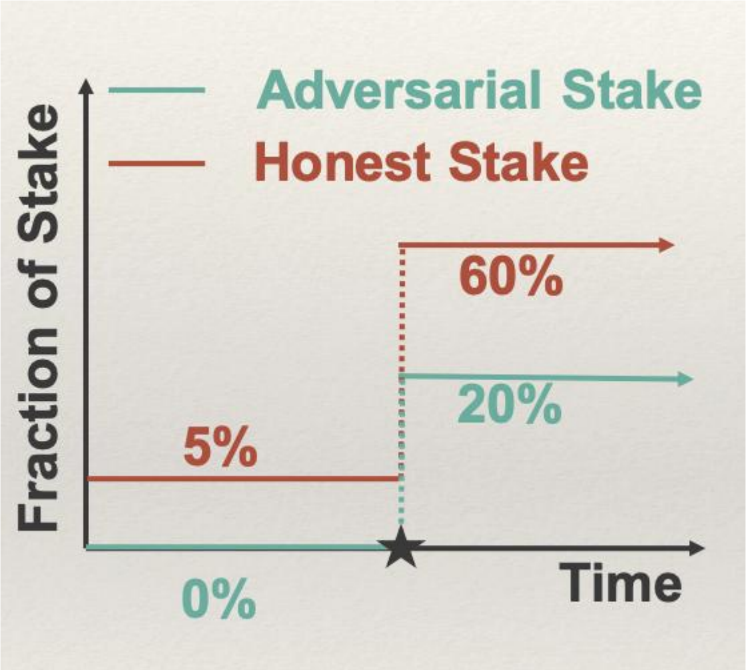


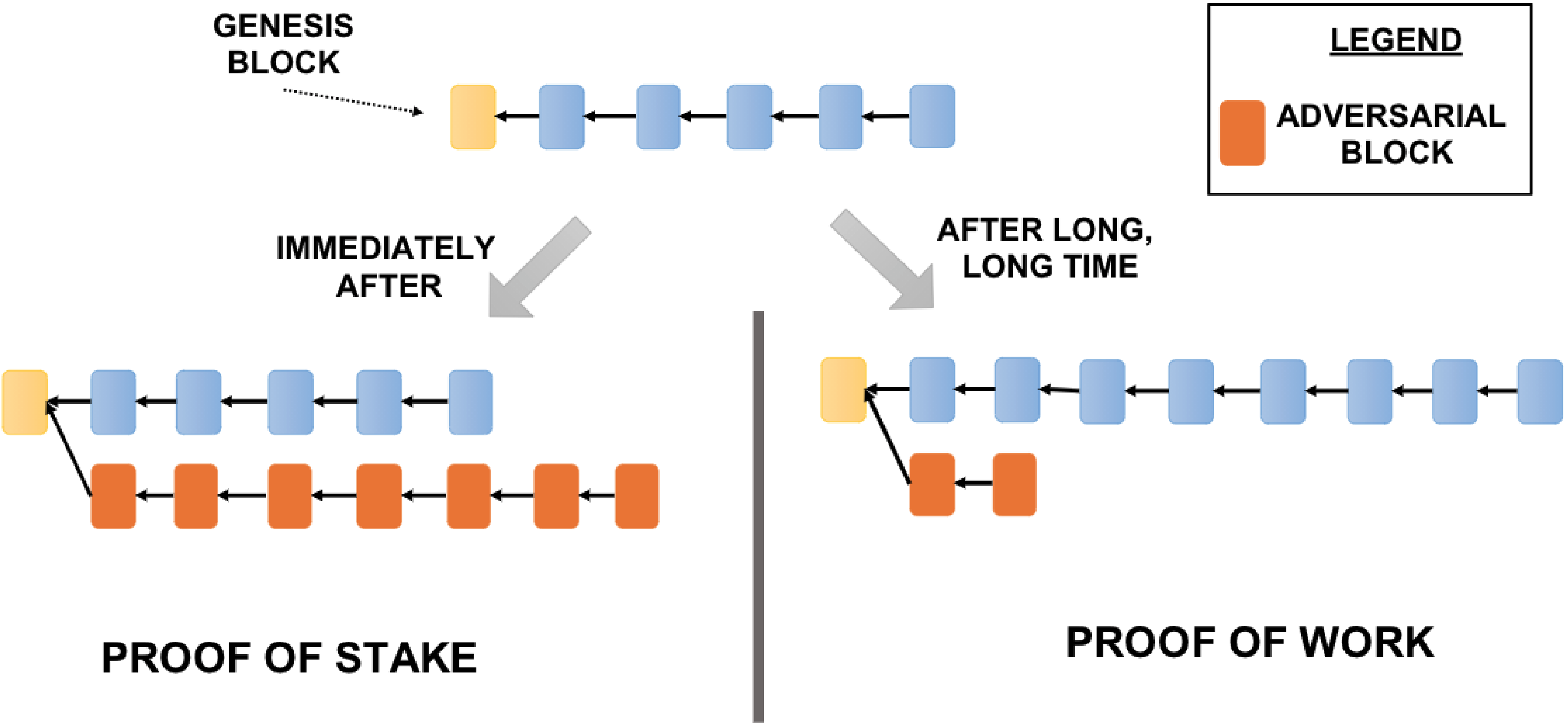
New Fork Choice Rule: s-truncation

- Stake is updated with a delay of s blocks

- Chain rule: When comparing two chains, both chains are truncated up to s blocks after the fork. Whichever truncated chain is mined in shorter time (and hence denser) is chosen to be mined on.

Dynamic availability





Crypto Primitive 3

Verifiable Delay Function (VDF)

VDF(sk , x) $\rightarrow$ (y , π) Takes L steps

Verify(pk , x , y , π) $\rightarrow$ True/False Takes much less than L steps

- Proof of sequential work

PoSAT: PoS with arrow-of-time

$$\mathrm{VRF}(\mathrm{RandSource}(\mathrm{parent}), \mathrm{ts}, \mathrm{sk_n}) < T \cdot \mathrm{stake_n}$$

$$\Downarrow$$

$$\mathrm{VDF}(\mathrm{RandSource}(\mathrm{parent}), \mathrm{ts}, \mathrm{sk_n}) < T \cdot \mathrm{stake_n}$$

Conclusion

- Several blockchain platforms are based on PoS
  - Ethereum 2.0, Solana, Cosmos, Near, Flow, Polka Dot
- Other features desired in PoS
  - Privacy, Finality,
  - Covered in the last third of this course