Blockchain Principles and Applications

Amir Mahdi Sadeghzadeh, PhD

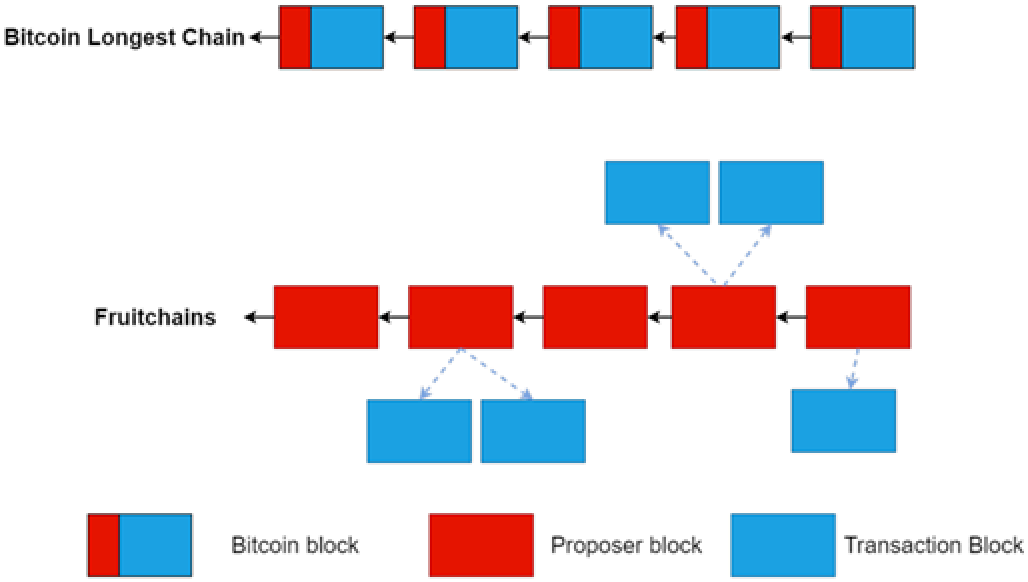
Data and Network Security Lab (DNSL)
Trustworthy and Secure AI Lab (TSAIL)

Most slides have been adapted from ECE/COS 470, Pramod Viswanath, Princeton 2024 and CS251, Dan Boneh, Stanford 2023.

Recap

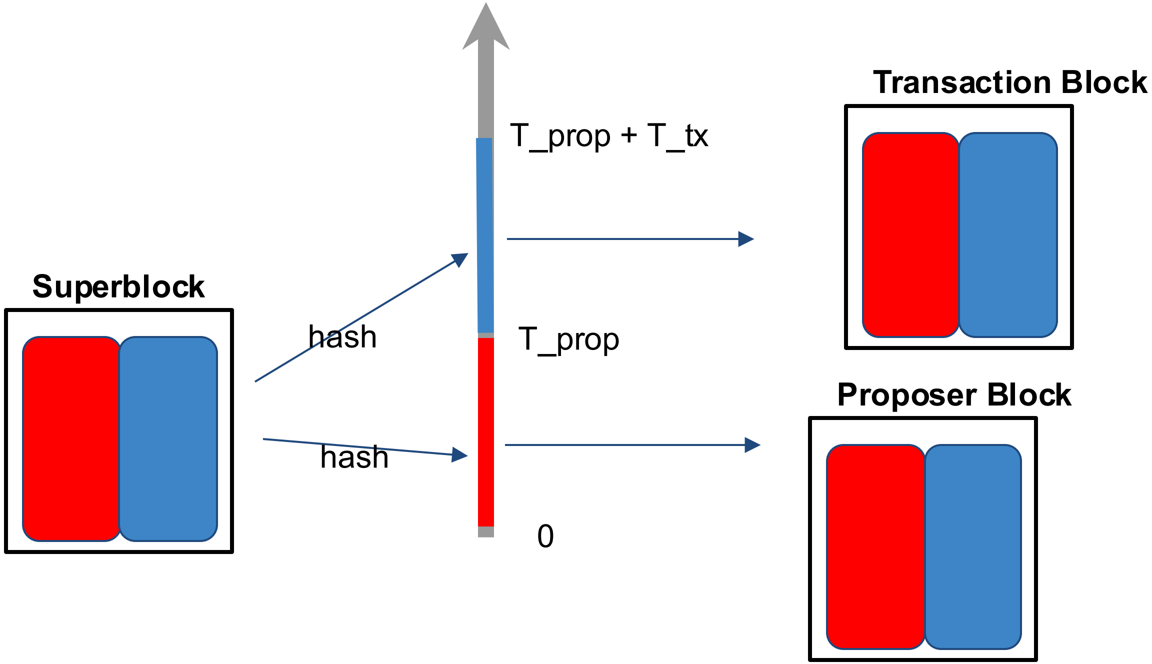
Fruitchains

Main idea: separate transactions (& their rewards) from blocks in the longest chain

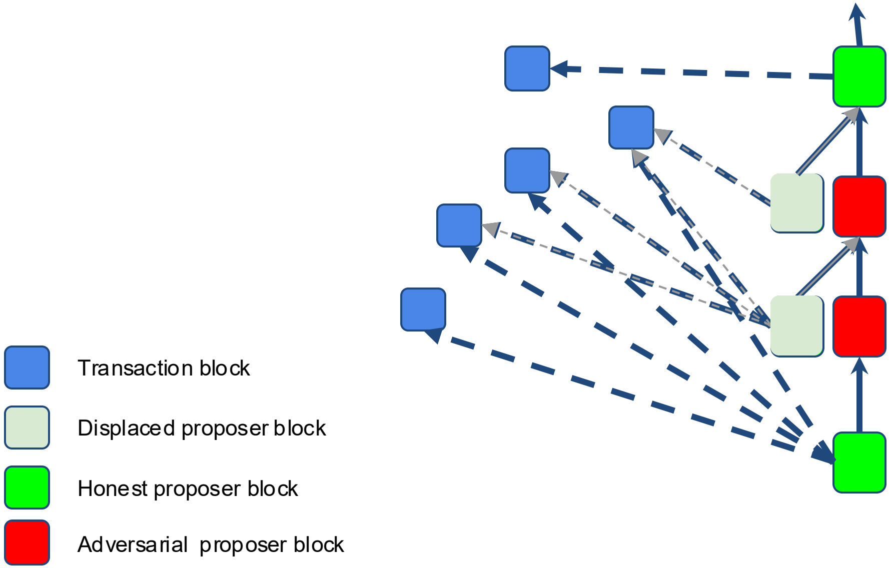


Cryptographic Sortition

How to do PoW for both types of blocks simultaneously?



Optimal chain quality

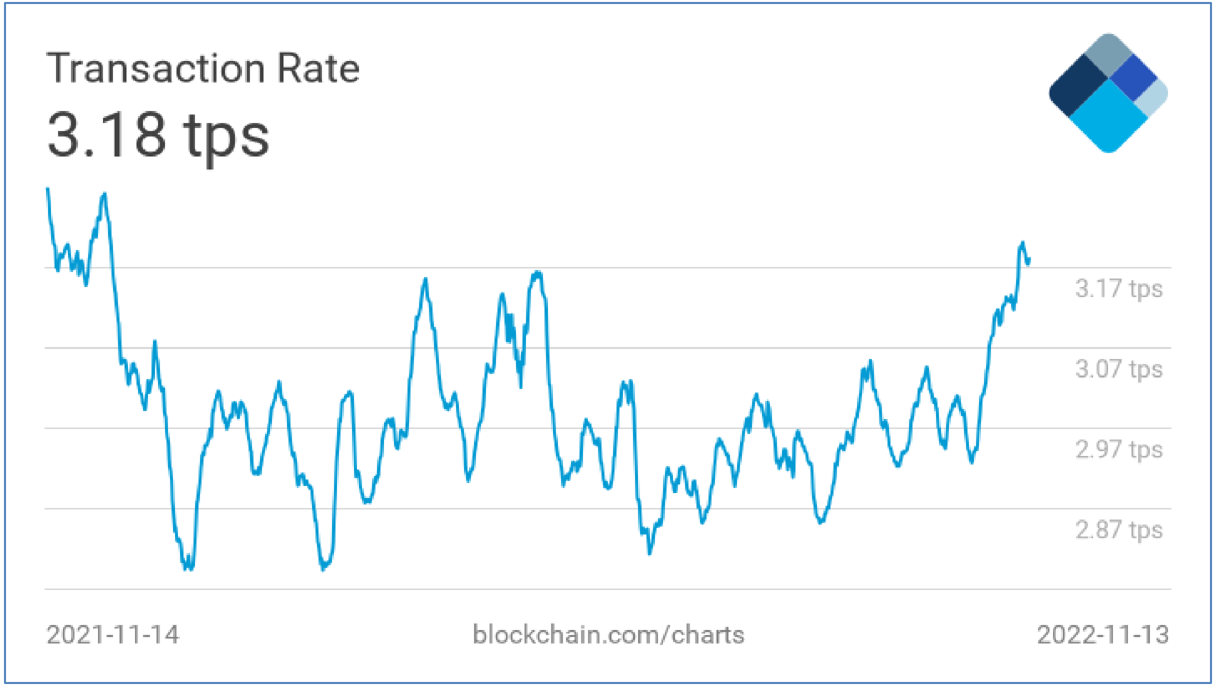


Fraction of honest transaction blocks **in the longest chain** → Fraction of honest transaction blocks **overall** →

$$\mathrm{CQ} \geq \frac{(1-\beta)\lambda_{\mathrm{tx}}}{(1-\beta)\lambda_{\mathrm{tx}} + \beta\lambda_{\mathrm{tx}}} = \boxed{1-\beta.}$$

Scaling Throughput

Bitcoin Tx per second



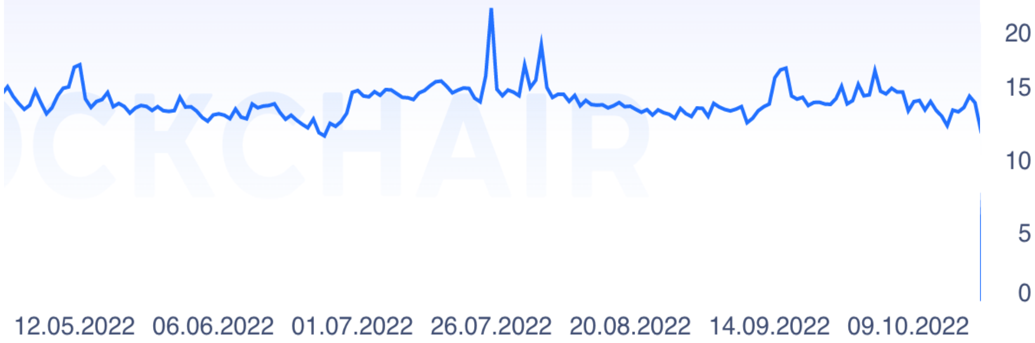
≈4200 Tx/block

1 block / 10 mins

⇒ max: 7 Tx/sec

Ethereum Tx per second

Ethereum avg Tx per second:



≈ 15 Tx/sec

Simple Tx: 21k Gas

max 30M Gas per block

⇒ max 1428 tx/block

1 Block/12s

⇒ max 119 tx/s

In comparison …

Visa: up to 24,000 Tx/sec (regularly 2,000 Tx/sec)

PayPal: 200 Tx/sec

Ethereum: 15 Tx/sec

Bitcoin: 7 Tx/sec

Goal: scale up blockchain Tx speed

Idea 3: Prism 1.0 or Fruitchains

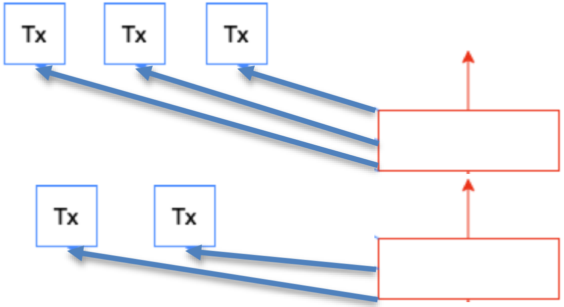
Bitcoin-NG is a good idea: it separated security from payload/data

Prism 1.0 is similar to Bitcoin-NG

- Consist of proposer blocks and transaction blocks

But

- Transaction blocks are not linked but referred by proposer blocks
- The PoW for transaction blocks is easy for throughput
- The PoW for proposer blocks is hard for security



Scaling Latency

Bitcoin latency

Time from when a transaction was broadcast until the transaction is confirmed in the ledger

- τ_1: Time from when a transaction was broadcast until the transaction is put into a mined block B
- τ_2: Time from when the transaction was put into a mined block B until block B is k-deep in the longest chain

$$\tau = \tau_1 + \tau_2$$

τ_2 is the real bottleneck, depends on how large k is.

Bitcoin latency

Assume low forking ($\lambda\Delta \ll 1$),

$$\tau = \frac{k}{(1-\beta)\lambda}$$

Depth of blocks

Block arrival rate

From Lecture 6, error probability

$$\epsilon = e^{-ck}$$

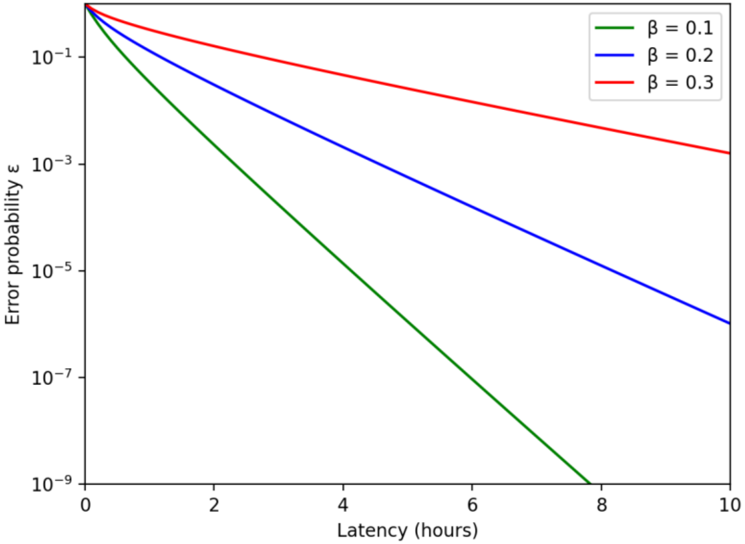
$$\tau = \frac{\frac{1}{c}\log(\frac{1}{\epsilon})}{(1-\beta)\lambda} = O(\frac{1}{\lambda}\log(\frac{1}{\epsilon}))$$

Latency and security are coupled

Bitcoin latency

$$\tau = O(\frac{1}{\lambda}\log(\frac{1}{\epsilon}))$$

Bitcoin: $\frac{1}{\lambda}$ = 10 minutes



Improve Bitcoin latency

Only way to improve latency is to

- reduce k; but this reduces security
- Increase λ; but this also reduces security

Ethereum: $\frac{1}{\lambda}$ = 15s; $k = 100$

- latency = 25 minutes
- Way better than Bitcoin performance; improvement simply by picking better parameters.

Improve Bitcoin latency

Question: can we make relatively small changes to the longest chain protocol and PoW mining while scaling latency?

Key Requirement:

- Do not want latency to depend on security level
- **Decouple security from latency**

Prism

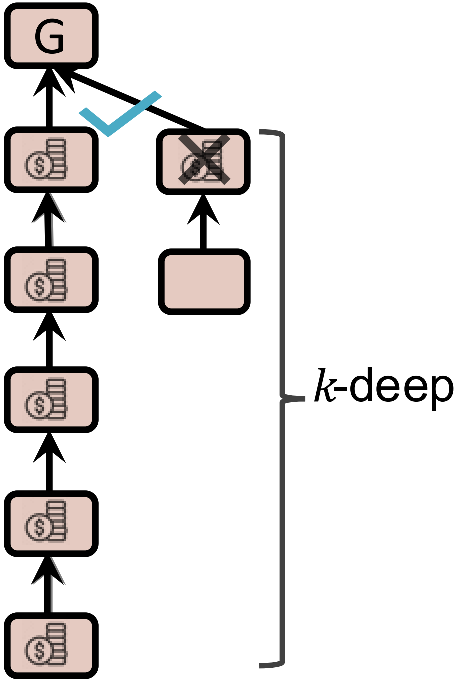
Prism achieves optimal latency

- **Decoupling principle**: separate performance from security

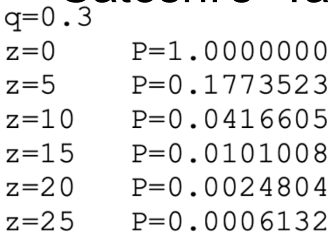
- Prism 1.0 achieves optimal throughput; last lecture

Decoupling voting

k-deep confirmation rule is a form of voting



Satoshi's Table



```
q=0.3
z=0      P=1.0000000
z=5      P=0.1773523
z=10     P=0.0416605
z=15     P=0.0101008
z=20     P=0.0024804
z=25     P=0.0006132
```

~~1 deep~~ => .45
25 deep => 0.0006

Can think of one block = one vote underneath B

k-deep = k votes in sequence

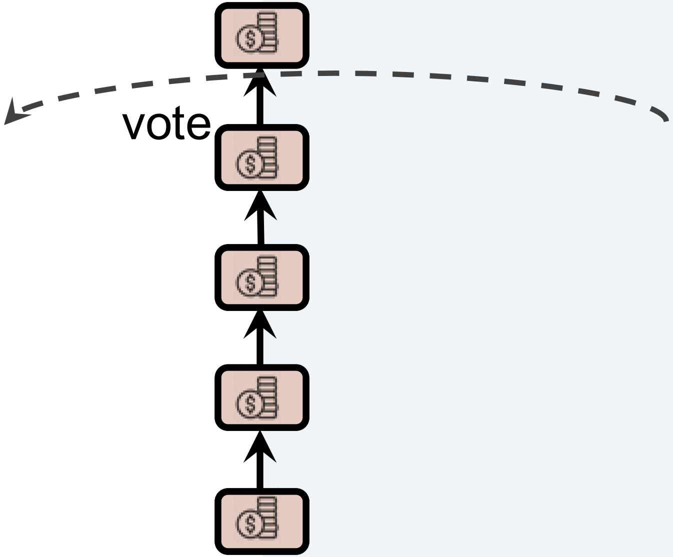
Really need k large to sample the miners

Bitcoin → Deconstruct

Proposing

Voting

Ledger construction

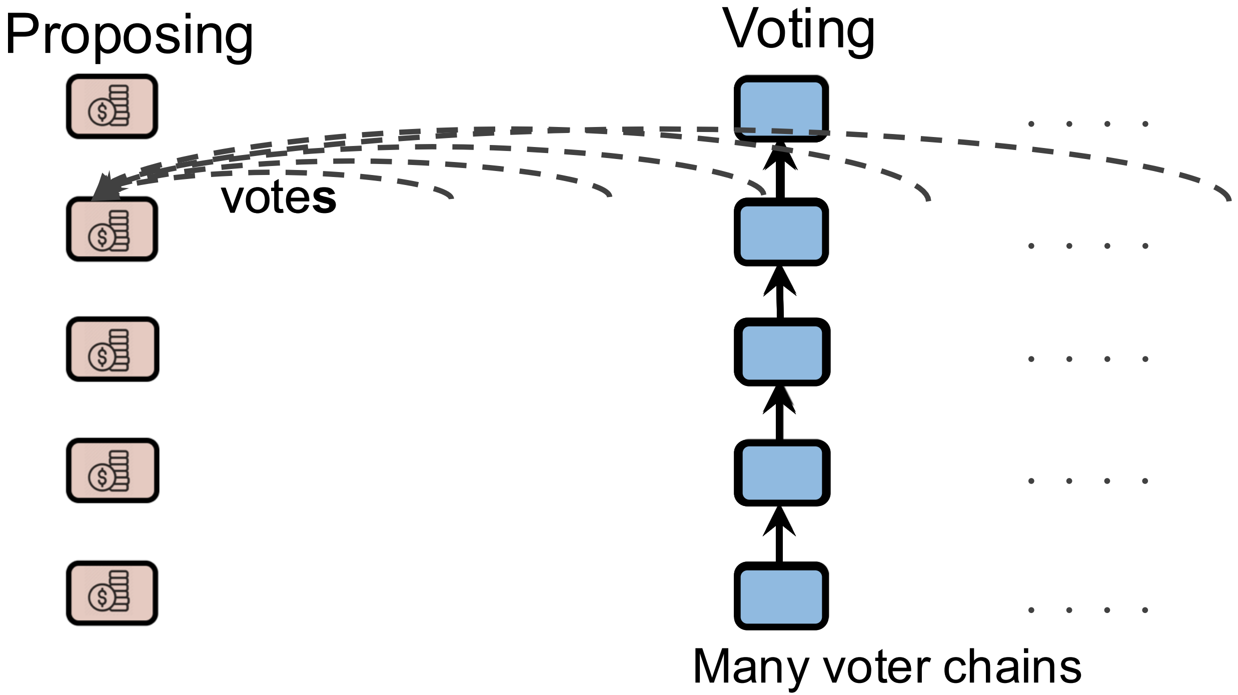


1. Select votes along longest voter chain

2. Order the proposer blocks by votes

Bitcoin → Deconstruct →Prism

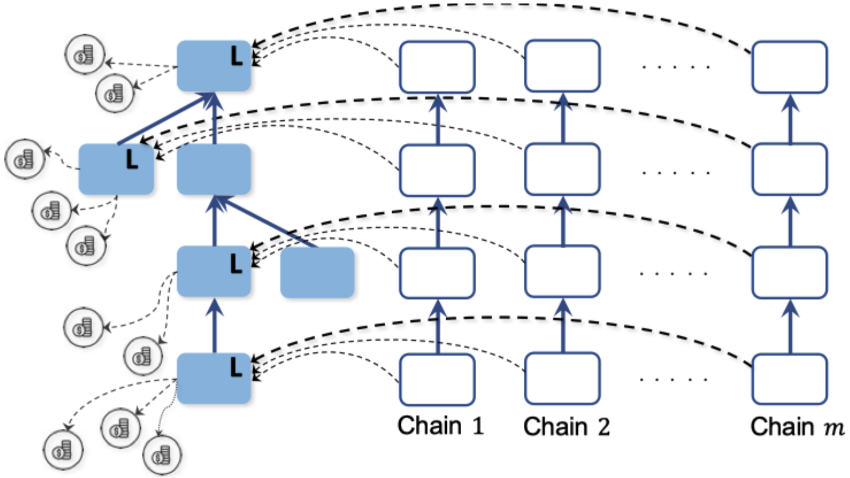
Fast Confirmation



Ledger Construction: For each level choose the proposer block with **maximum** votes

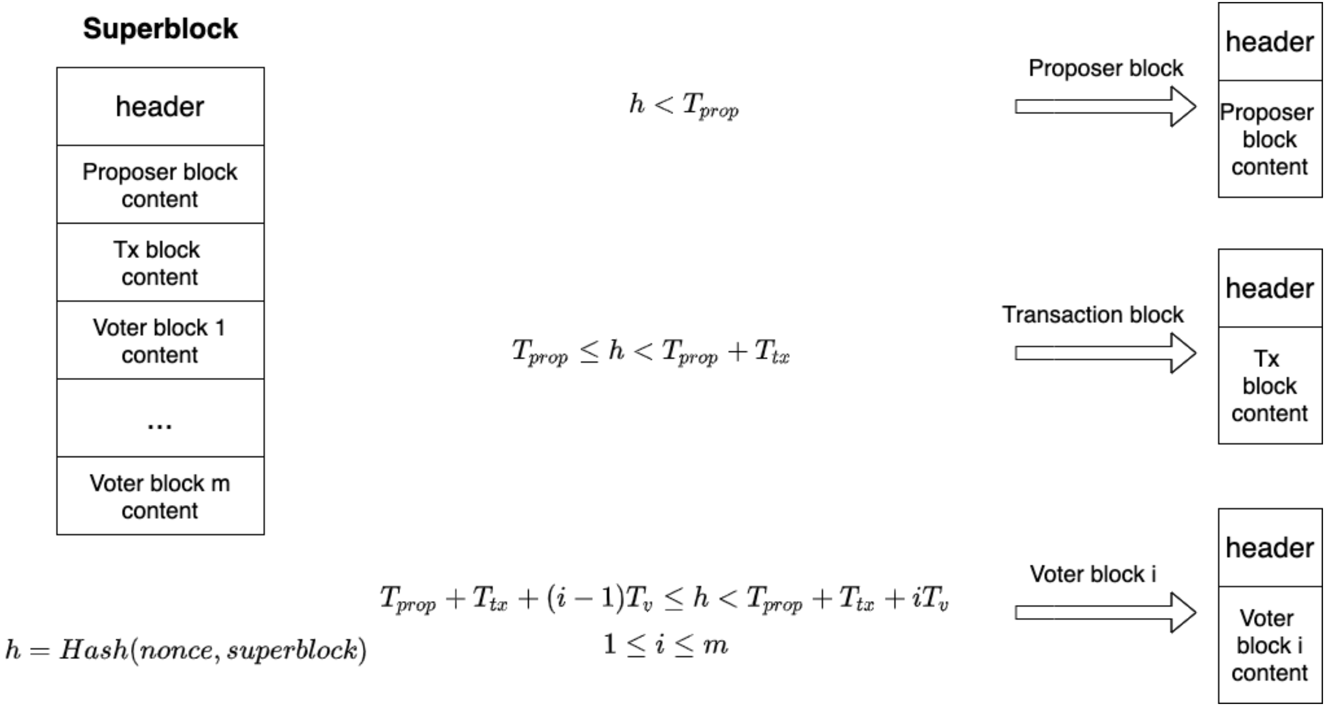
Prism

- Proposal rule: longest chain
- Voting rule:
  a) each voter chain votes for one and only one proposer block at each level
  b) each voter block votes for all the proposer levels that have not been voted by its parent.
- Mining rule: honest miner picks to be proposer/voter/transaction block at random



Cryptographic sortition

How do you prevent adversary from m focusing its mining power on a specific type of blocks or on a specific voter chain?



Superblock

header
Proposer block content
Tx block content
Voter block 1 content
…
Voter block m content

$h = Hash(nonce, superblock)$

$h < T_{prop}$ → Proposer block → header / Proposer block content

$T_{prop} \leq h < T_{prop} + T_{tx}$ → Transaction block → header / Tx block content

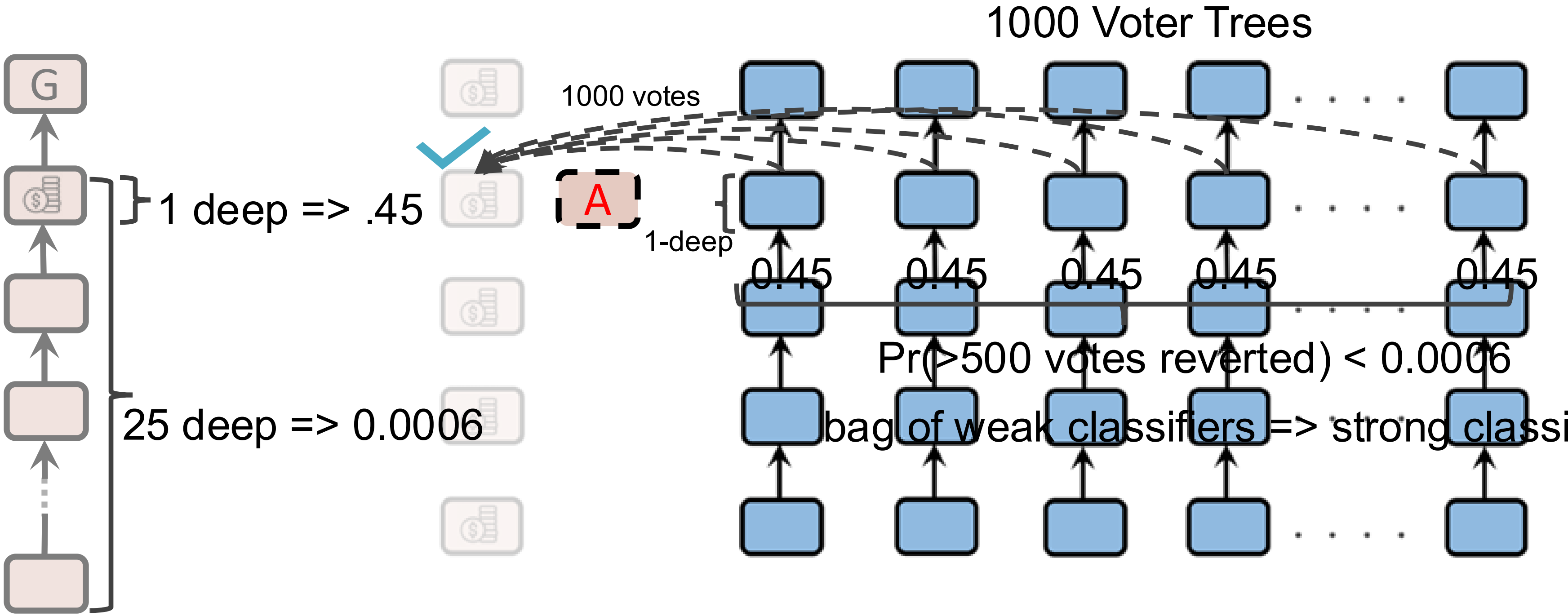
$T_{prop} + T_{tx} + (i-1)T_v \leq h < T_{prop} + T_{tx} + iT_v$

$1 \leq i \leq m$

→ Voter block i → header / Voter block i content

Fast confirmation

Bitcoin



Pr(>500 votes reverted) < 0.0006

bag of weak classifiers => strong classi

Ledger Construction: For each level choose the proposer block with **maximum** votes

Resources

- ECE/COS 470, Pramod Viswanath, Princeton 2024