Bitcoin:
Programming the Future of Money

Topics in Computer Science - ITCS 4010/5010, Spring 2025

Dr. Christian Kümmerle

Lecture 24

Proof-Of-Stake

- The content of this class are largely based on the lecture series of Tim Roughgarden on Proof-Of-Stake (Lecture 12) in series "Foundations of Blockchains".

UNIVERSITY OF NORTH CAROLINA
CHARLOTTE

Attacks

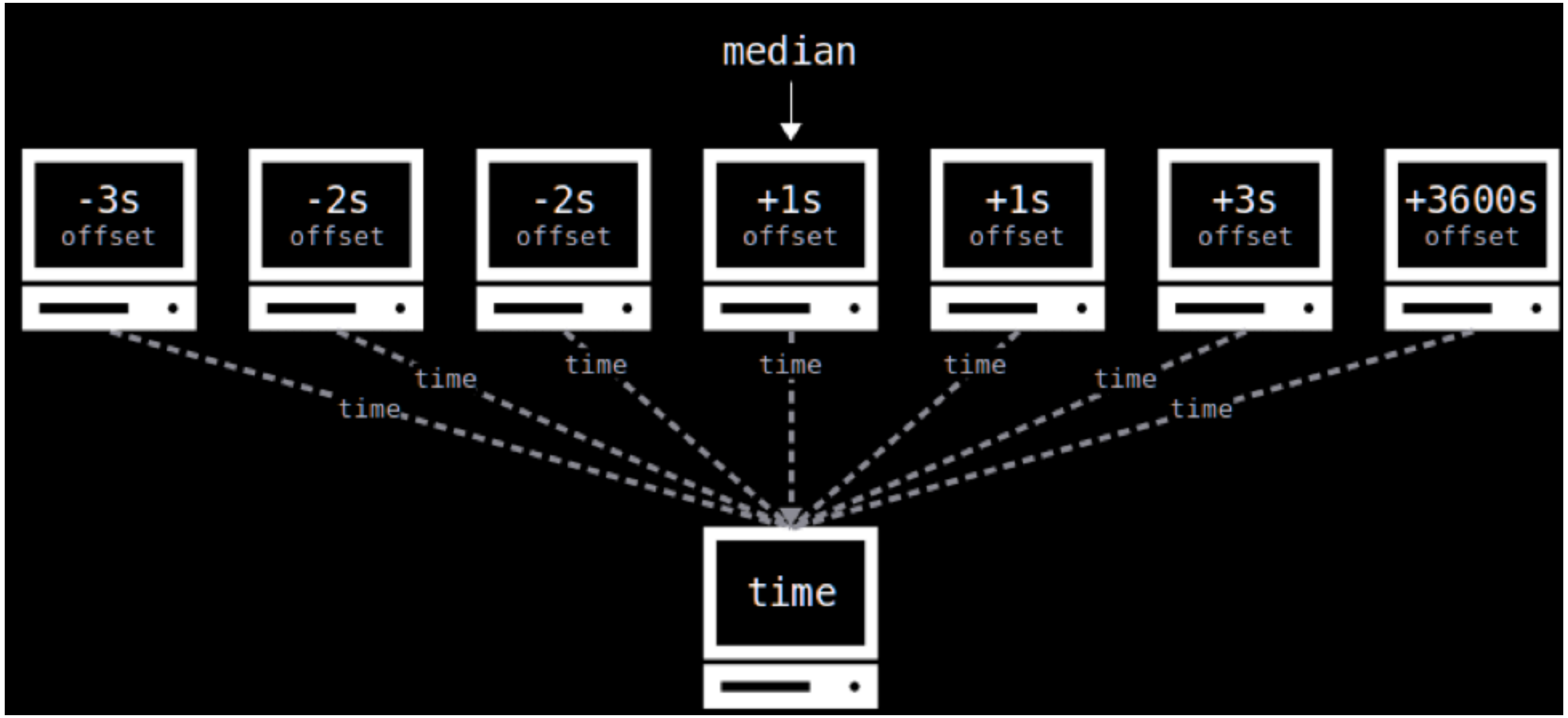
VALIDATION OF BLOCKS

For a block to be valid, the following rules need to be satisfied:

- Syntax of the block data structure needs to be correct (see also here).
- Block header hash is less than the target.
- Block time stamp is above the **Median Time Past** (See BIP113) (median time last 11 blocks in the chain).
- Block time stamp is below **Network Adjusted Time** plus two hours.
- Block size is below 1,000,000 vbytes.
- (Only) first transaction in transaction Merkle tree is the **coinbase transaction**.
- All transactions in block are valid.

NETWORK ADJUSTED TIME

Definition: Local time of node + median offset of all connected nodes



Rule: Block time stamp is below **Network Adjusted Time** plus two hours.

Q: How could this be manipulated?

THE DIFFICULTY ADJUSTMENT

Every 2016 blocks, mining difficulty is adjusted by updating value for "target" in the subsequent 2016 blocks based on:

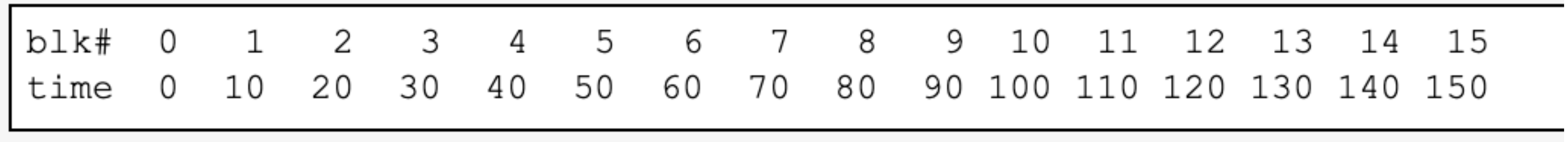
$$\text{new target} = \text{old target} \cdot \frac{(\text{time of current block}) - (\text{time of (current} - 2015\text{th) block})}{20160 \text{ minutes}}$$

- The target cannot increase by more than 400% in each adjustment period.
- The target cannot decrease by more than 75% in each adjustment period.

THE TIME WARP ATTACK

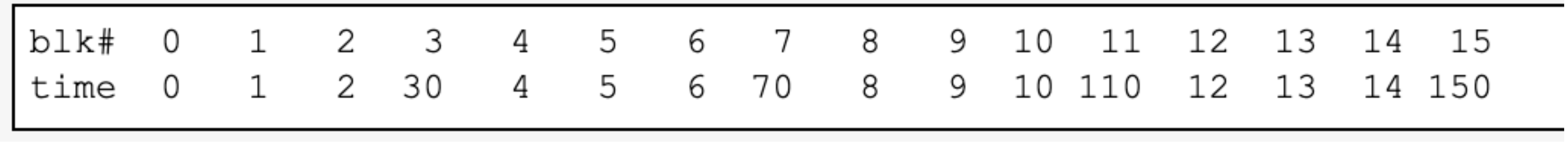
Assume here: Difficulty adjustment after 4 blocks.

Normal chain (example):



blk#	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
time	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150

Chain with manipulated time stamps:



blk#	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
time	0	1	2	30	4	5	6	70	8	9	10	110	12	13	14	150

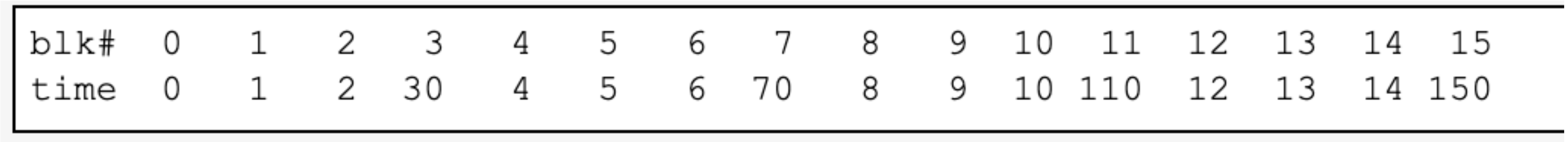
- Time passed between #3-#0: 30 min
- Time passed between #7-#4: 66 min
- Time passed between #11-#8: 104 min

THE TIME WARP ATTACK

Attack strategy:

- Miners set time stamps of blocks in alternating pattern:
  - First three blocks of adjustment period (blk# 0,1,2, 4,5,6, and 8,9,10 etc.) use time stamps **as small as possible** (while still choosing them above the Median Time Past, the median time stamp of last 11 blocks)
  - Last block of adjustment period (blk# 3,6,10) use time stamp that corresponds to actual time (**much larger,** but below Network Adjusted Time)

Chain with manipulated time stamps:



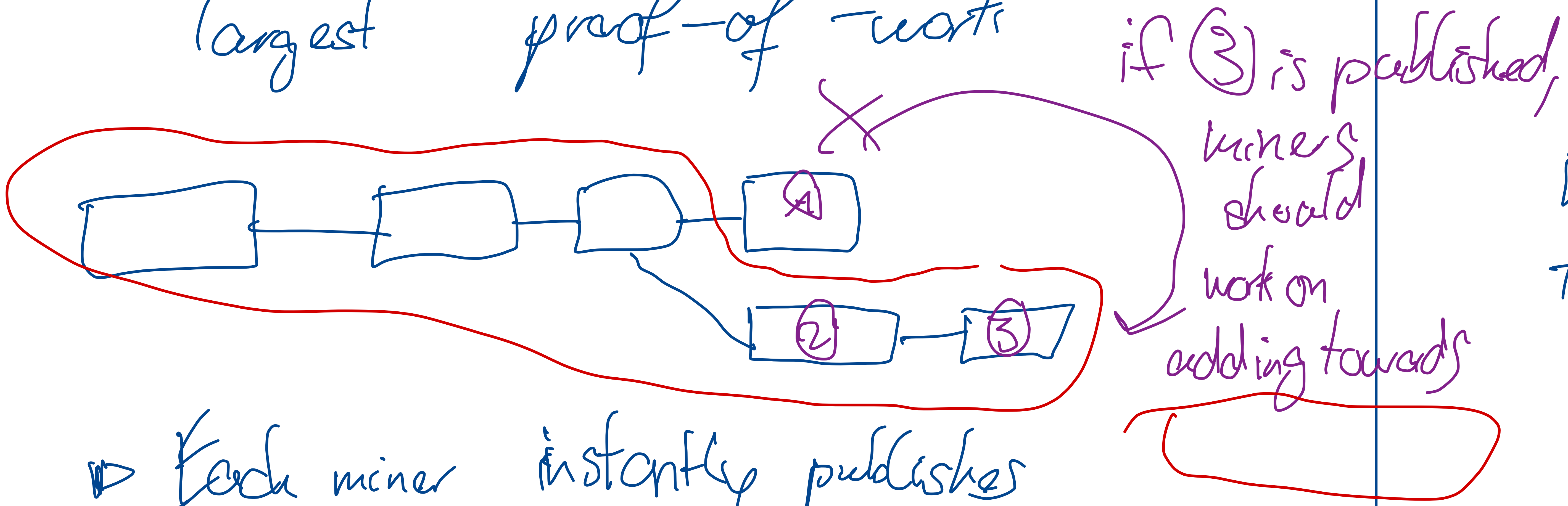
blk#	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
time	0	1	2	30	4	5	6	70	8	9	10	110	12	13	14	150

- Time passed between #3-#0: 30 min -> Difficulty in first period: 1 (relative measure)
- Time passed between #7-#4: 66 min -> Difficulty in second period: 1* (66 min / 30 min)$^{-1}$ = 0.4545
- Time passed between #11-#8: 104 min -> Difficulty in third period: 0.4545* (104 min / 30 min)$^{-1}$ = 0.1311

SELFISH MINING

Honest mining:

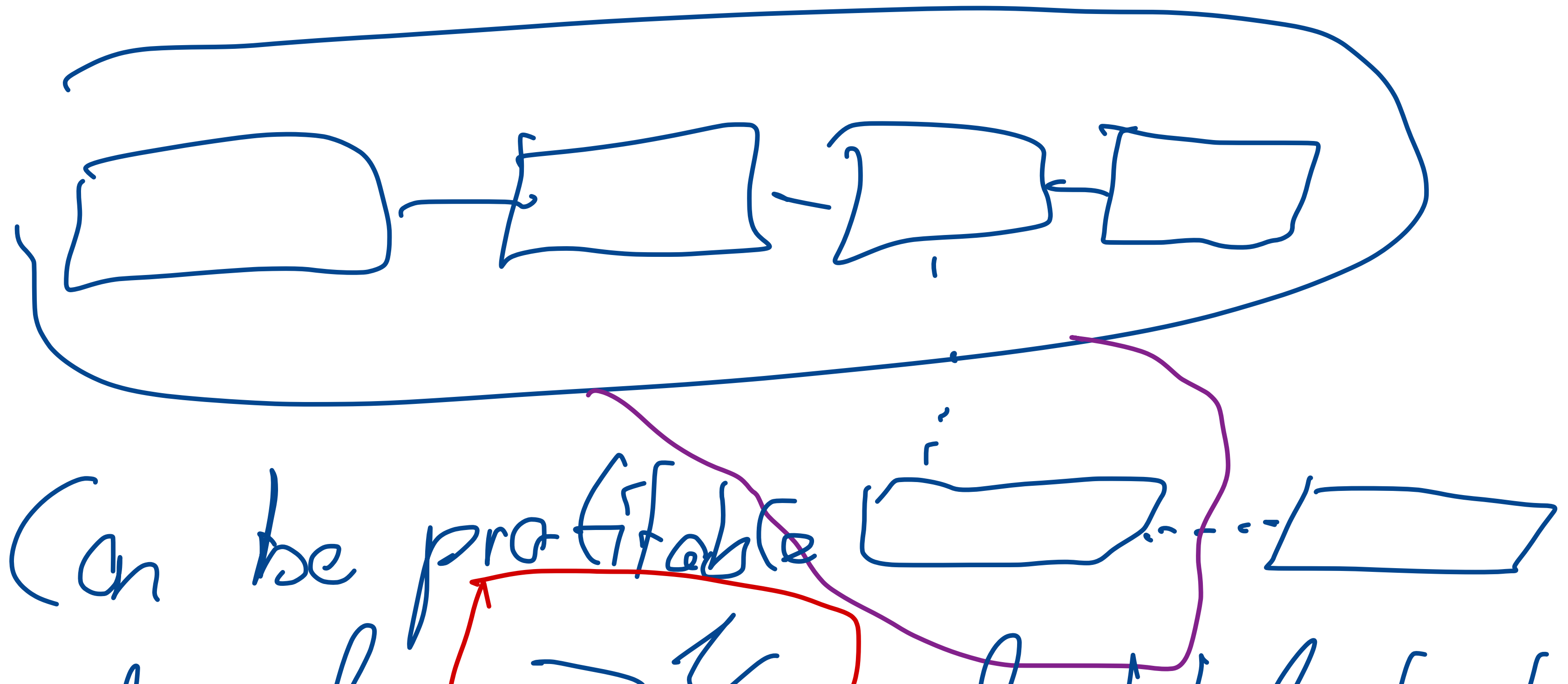
- Each miner keeps adding on "heaviest" chain i.e. chain with largest proof-of-work



- Each miner instantly publishes each block they mine
- Include those transactions that maximize total transaction fee revenue

Selfish mining

- A miner does not necessarily publish a block that they find
- They build "hidden" chain tips
- Under certain circumstances, the publish only part of hidden chain tip



⇒ Can be profitable already if $>1/3$ of total hashrate executes this attack

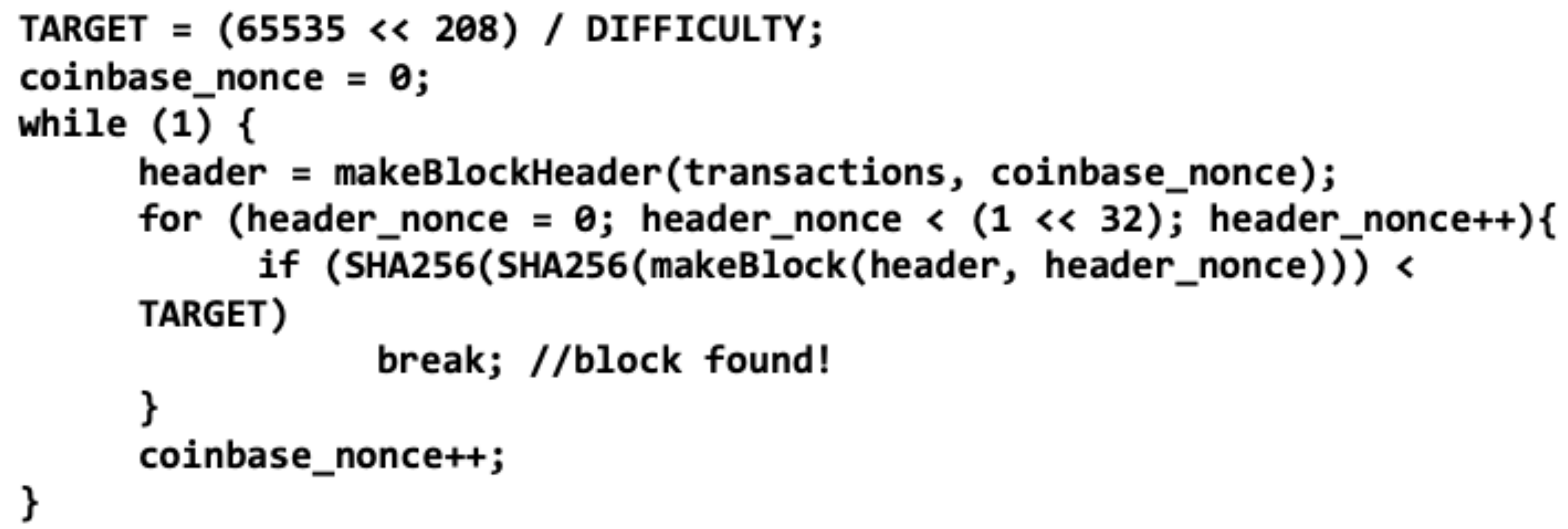
Proof-of-Stake

PROOF-OF-WORK

Which problem does Proof-of-Work (e.g., Bitcoin mining) solve?

Permissionless Consensus:

- A set of n nodes can communicate with each other, exchange messages with each other via broadcast (e.g., new transactions)
- Honest nodes communicate via broadcast
- Nodes intend to achieve agreement on current state (e.g., UTXO set in Bitcoin)
- An arbitrary number of nodes can enter protocol or leave protocol at any time (**permissionlessness**)
- A percentage α of dishonest nodes as large as possible should **not** lead to breakdown of agreement on current state



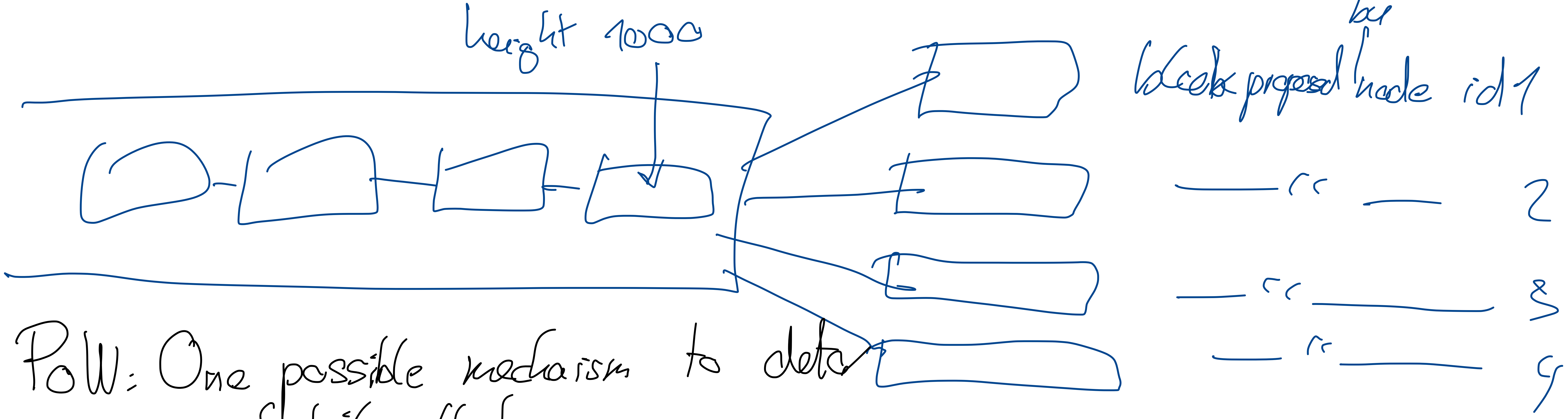
```
TARGET = (65535 << 208) / DIFFICULTY;
coinbase_nonce = 0;
while (1) {
      header = makeBlockHeader(transactions, coinbase_nonce);
      for (header_nonce = 0; header_nonce < (1 << 32); header_nonce++){
            if (SHA256(SHA256(makeBlock(header, header_nonce))) <
      TARGET)
                  break; //block found!
      }
      coinbase_nonce++;
}
```

Example Bitcoin mining pseudcocode

MAIN CHALLENGE IN PERMISSIONLESS CONSENSUS

Sybil Attack:

A single node creates large # of new “node IDs” and try to use this to attack the network (e.g., by proposing blocks)



PoW: One possible mechanism to deter Sybil attack

PoW selects block proposed by node id i to be added to chain with a probability proportional to hash rate fraction α_i of node i.

$\uparrow$ of total hash rate

WHAT IS PROOF-OF-STAKE?

An alternative sybil deterrence mechanism used by many alternative (i.e., non-Bitcoin) cryptocurrency/blockchain networks, e.g.:

- Ethereum (since 2022; before Fall 2022: Based on Proof-of-Work)
- BNB
- Cardano
- Avalanche
- Algorand
- …

WHAT IS PROOF-OF-STAKE?

Goal: Sybil-resistant random sampling mechanism

"validators"

Idea: Nodes that want to propose new blocks / vote on acceptance of blocks have to "lock up" a "stake".

lock up: deposit into some escrow contract

stake: usually: native currency unit

Desired property: $\mathbb{P}(\overset{\text{validator}}{\text{node } i} \text{ is selected to propose new block}) = \text{fraction of coins staked by node } i$

Q:

- ▷ How does it work in detail?
- ▷ Security guarantees?

WHY PROOF-OF-STAKE?

Observation: Most cryptocurrency networks use PoS!

Issues of PoW:

#1 Energy consumption of PoW

In PoS: No/little hash necessary, little energy consumption

#2 Latency of PoW → Faster blocks would lead to more chain re-orgs; internet communication time

#3: Recovery from 51%/ selfish attacks?

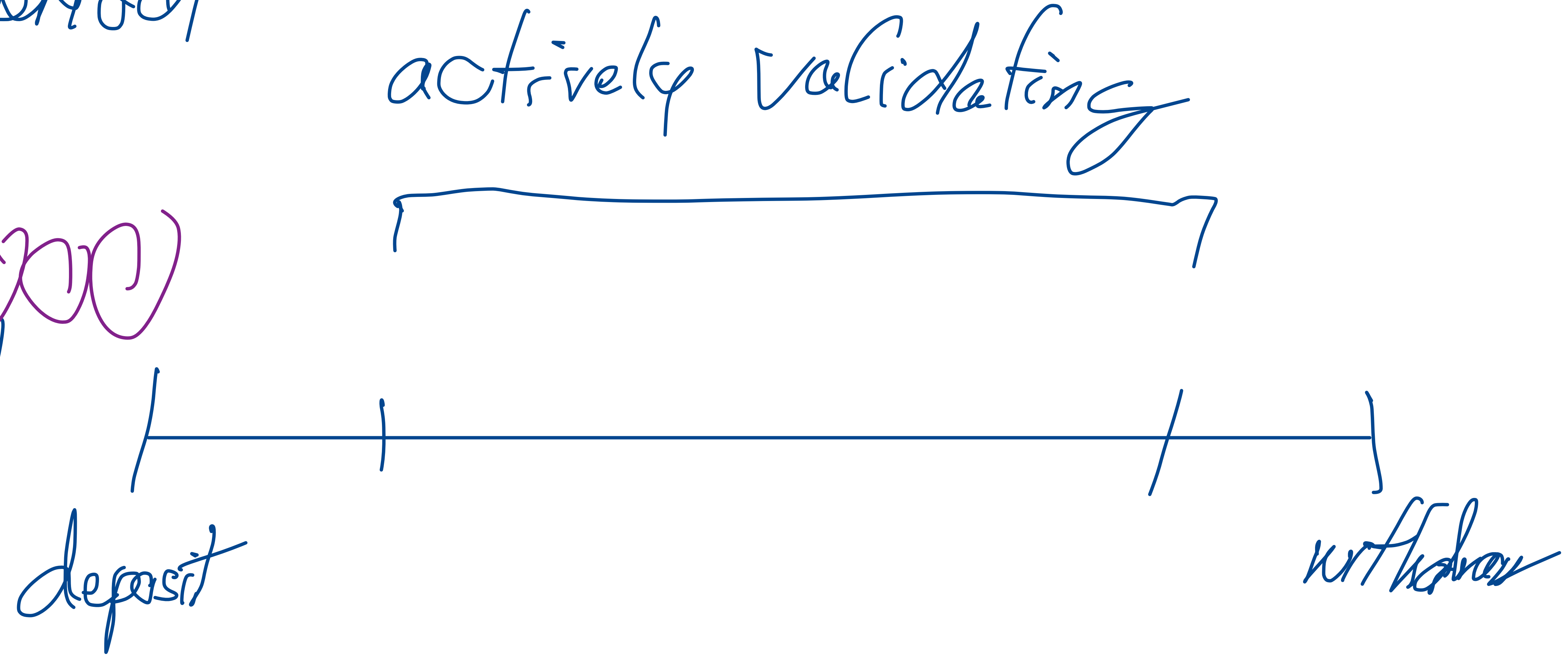
▷ PoS: Can recover from some attacks by "punishing" attackers

↳ "slashing"

MECHANICS OF STAKING

Design Decisions:

1. minimum/maximum lock-up period
2. min/max. staked coins
   for ETH: 32 ETH ~ $50,000
3. Warm-up / cool-down period in staking
4. distribution of stakings rewards
   typically 3-8% per annum
5. transaction fee → who earns?
   → how much?
6. "delegate" staking allowed



WHY PROOF-OF-STAKE IS HARD

Given: List $\{(\underbrace{pk_1}_{\text{public key of validator 1}}, \underbrace{q_1}_{\text{staked amount of val. 1}}), \ldots, (pk_n, q_n)\}$

Goal: Sample $pk \in \{pk_1, \ldots, pk_n\}$ w/ probability proportional to q_i

Challenge: Hard to implement w/o central authority!

↳ We need to define a fair, internal "pseudo" randomness

↳ Leads to attack vectors

PROOF-OF-WORK VS. PROOF-OF-STAKE

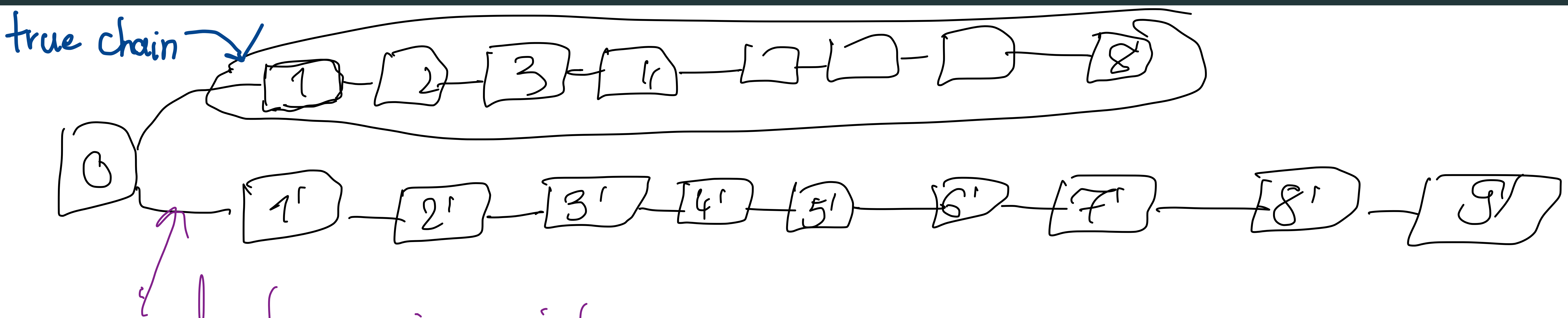
Advantages of PoS:

- Less energy consumption
- Lower latency possible / better finality guarantees
- Recovery from 51% attacks / punishment of bad actors within protocol possible

Disadvantages of PoS (vs. PoW):

- Significant additional complexity -> Possibility of bugs, lack of transparency
- Additional attack vectors (e.g., due to possibility of “costless simulation”, cf. Long-Range Attack)
- Less established proof record/ history (Bitcoin’s PoW works since 2009)
- Stronger trust assumptions
- (Possibly problematic) economic implications from how consensus works / protocol changes are implemented.

LONG-RANGE ATTACK



proposed chain using majority of validators at block height 0 (would work for any block height)

→ For details: Check Prof. Tim Roughg

Fundamental Problem: It is possible to costlessly recreate an alternative chain history using old validator keys!

wouldn't be a problem in PoW