

Consensus

Thierry Sans

# The consensus problem

In lecture 2, we said

*"one node is selected to create the next block"*

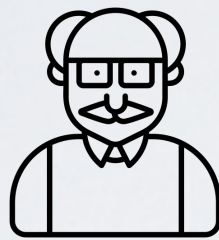
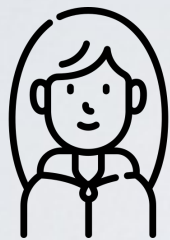
But how nodes agree on something in which some nodes in the P2P network might fail or not be honest?

# Outline

- Byzantine Agreement Problem and the Sybil attack
- Proof of Work (a.k.a Nakamoto Consensus)
- Proof of Stake
- Other consensus protocols

# The Byzantine Agreement Problem and The Sybil Attack

# The byzantine agreement problem

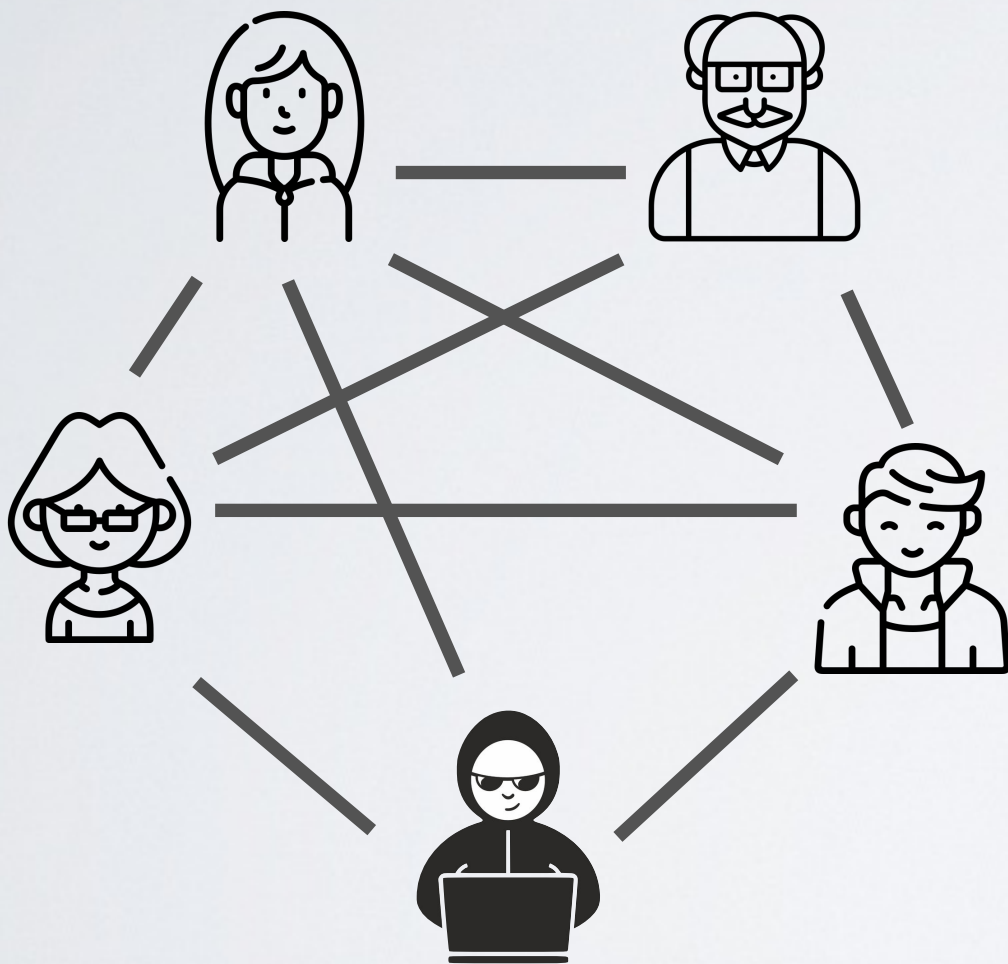


Generals attacking a fortress must decide whether to attack or retreat

- ✓ They must reach a common decision and make a coordinated action (consensus)
- Some participants might default or sabotage the consensus process (a.k.a Byzantine failure)



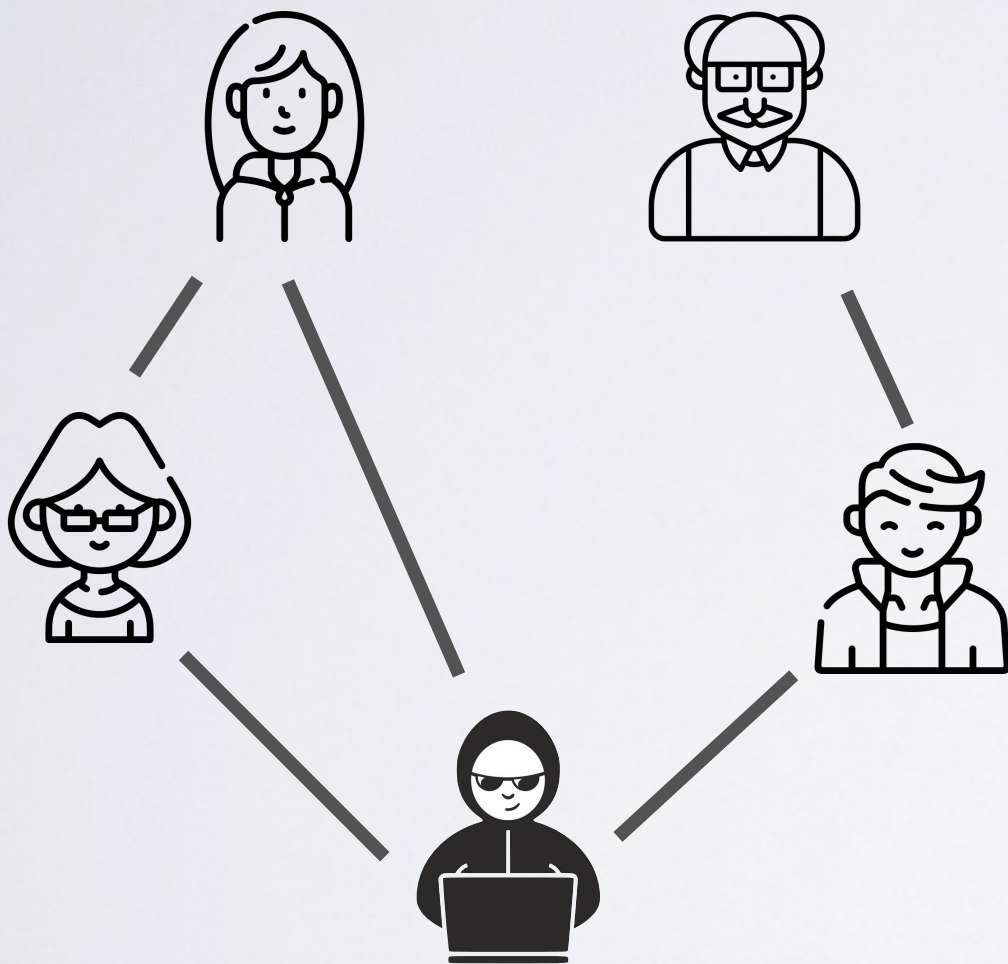
# A fully connected network



Each general sends their vote to all others individually

- (failure) some votes might not reach their destination
- (sabotage) a general might send different vote to different people

# A P2P network



Each generals sign and forward their vote to peers that we relay it (flooding algorithm)

- (failure) some votes might not reach their destination
- (sabotage) a general might send different vote to different people
- (sabotage) a general might not forward certain vote

# Requirements for Byzantine Fault Tolerance (BFT)

Given a system of  $n$  nodes,  $t$  of which are dishonest. When a node  $A$  broadcasts the value  $x$ , the other nodes are allowed to discuss with each other and verify the consistency of  $A$ 's broadcast, and eventually settle on a common value  $y$ .

The system is said to resist Byzantine faults if either

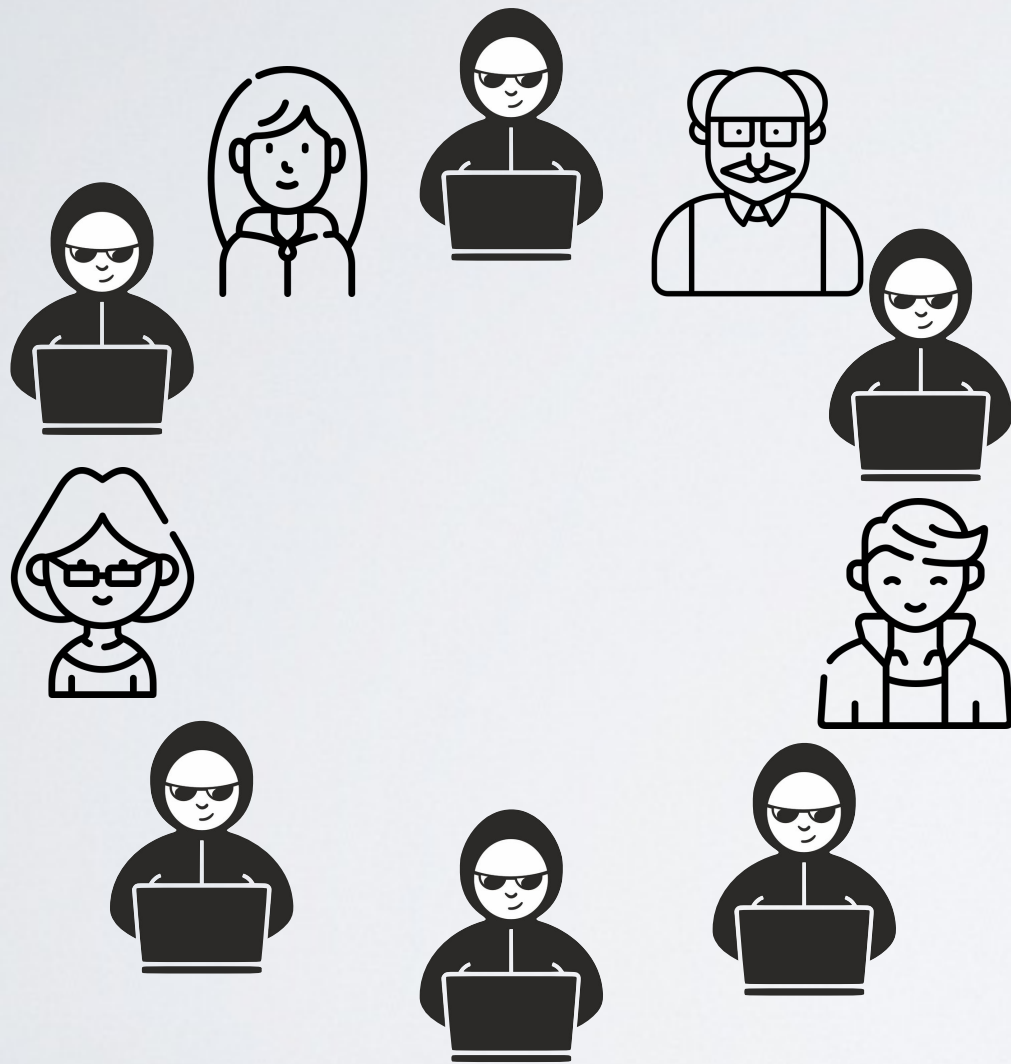
- $A$  is honest and all honest nodes agree that  $A$  says  $x$ , or
- $A$  fails or is dishonest and all honest node agree that  $A$  says  $Y$

Security properties:

- **Consistency** - Honest nodes do not contradict
- **Liveness** - Progress is made



# Another problem - Sybil Attack

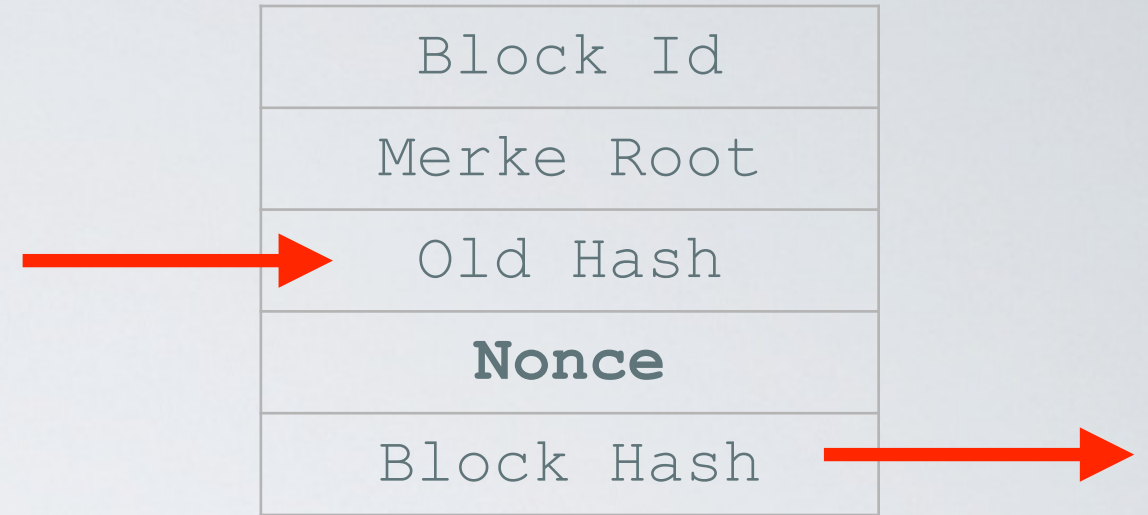


**If a reputation system,** an attacker can create and operate multiple nodes/identities to

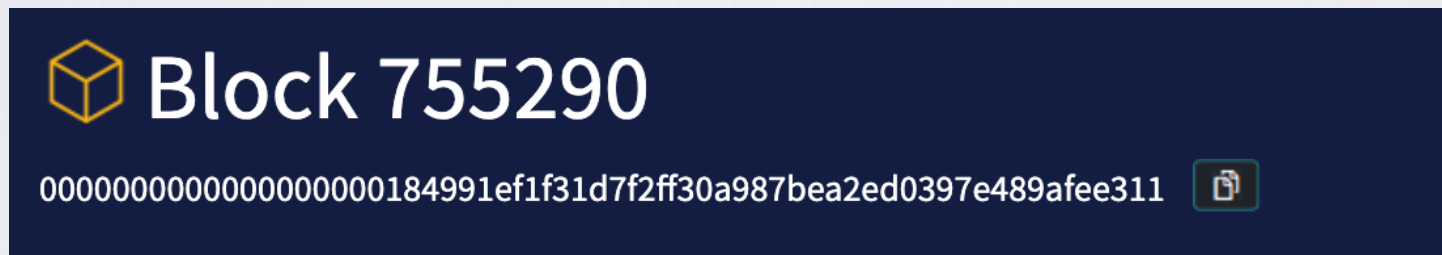
- Carry out a 51% attack to control the consensus outcome
- Block messages from honest nodes (P2P network)

# Proof of Work (a.k.a Nakamoto Consensus)

# The Mining Challenge



➔ Find a nonce such that  $H(\text{block})$  has a certain number of leading 0s



The number of zeros is determined from the target

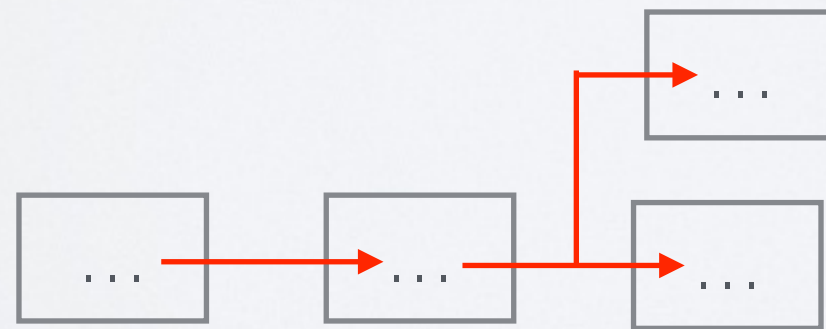
[https://en.bitcoinwiki.org/wiki/Difficulty\\_in\\_Mining#Bitcoin\\_mining\\_difficulty](https://en.bitcoinwiki.org/wiki/Difficulty_in_Mining#Bitcoin_mining_difficulty)

Target is adjusted every 2016 blocks (every 2 weeks) to mine a block every 10 minutes in average

- ✓ Finding the right nonce that outputs the right hash is hard (but not impossible) to compute
- ✓ Verifying the nonce and the block hash is easy to verify by other nodes

## Good but ...

- ✓ Prevents the sybil attack  
Probability of winning is relative to the computation power
- Does not fully solve the Byzantine Agreement Problem (yet)  
The mining challenge does not prevent **two valid** blocks be found and broadcasted at relatively same time

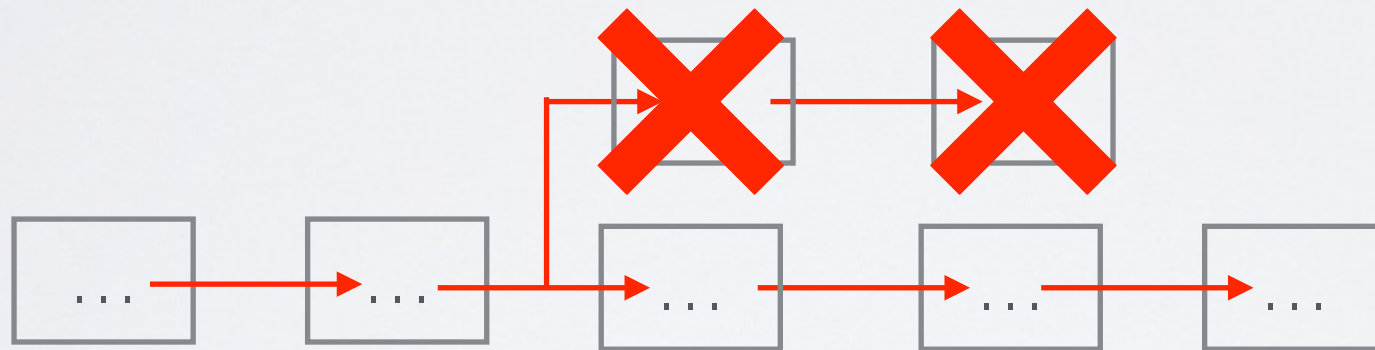




# The Longest-chain Wins Protocol

## Simple rule

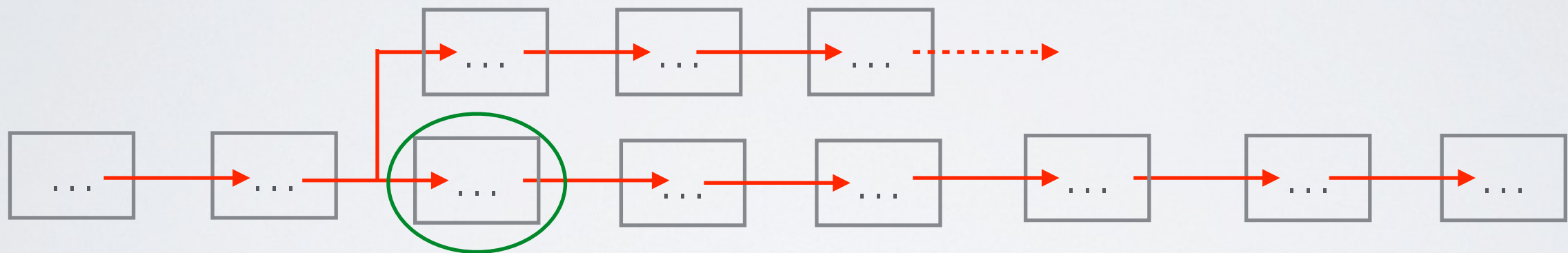
A miner must mine a block on the longest chain known



➔ **Consequence** : one should wait at least 6 confirmation blocks to consider a transaction as fully confirmed

# Double-spending attack

Can Mallory perform a double-spending attack by crafting a chain that exceeds 6 blocks previously confirmed?



- ✓ Binomial Random Walk (see Bitcoin paper)  
Impossible to catch up with the longest chain unless Mallory controls more than 51% of the blockchain computing power

# Criticism of Proof of Work

- All miners spend considerable amount of energy to eventually have only one of them being rewarded

# Proof of Stake

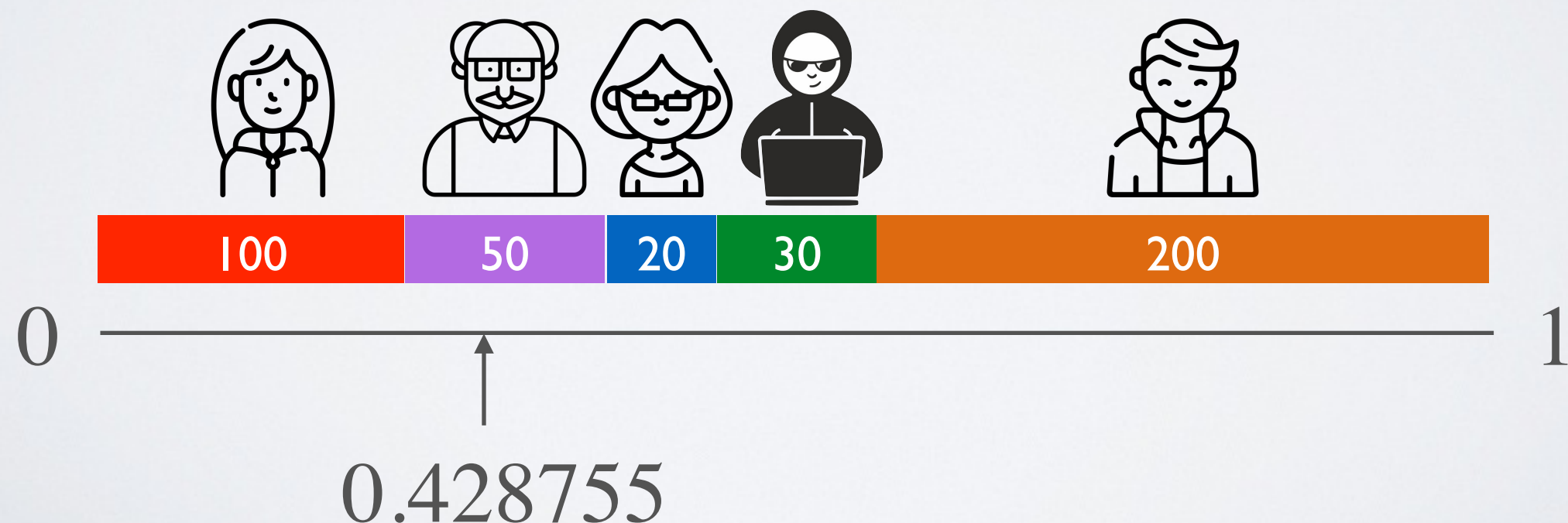


# A more energy efficient approach

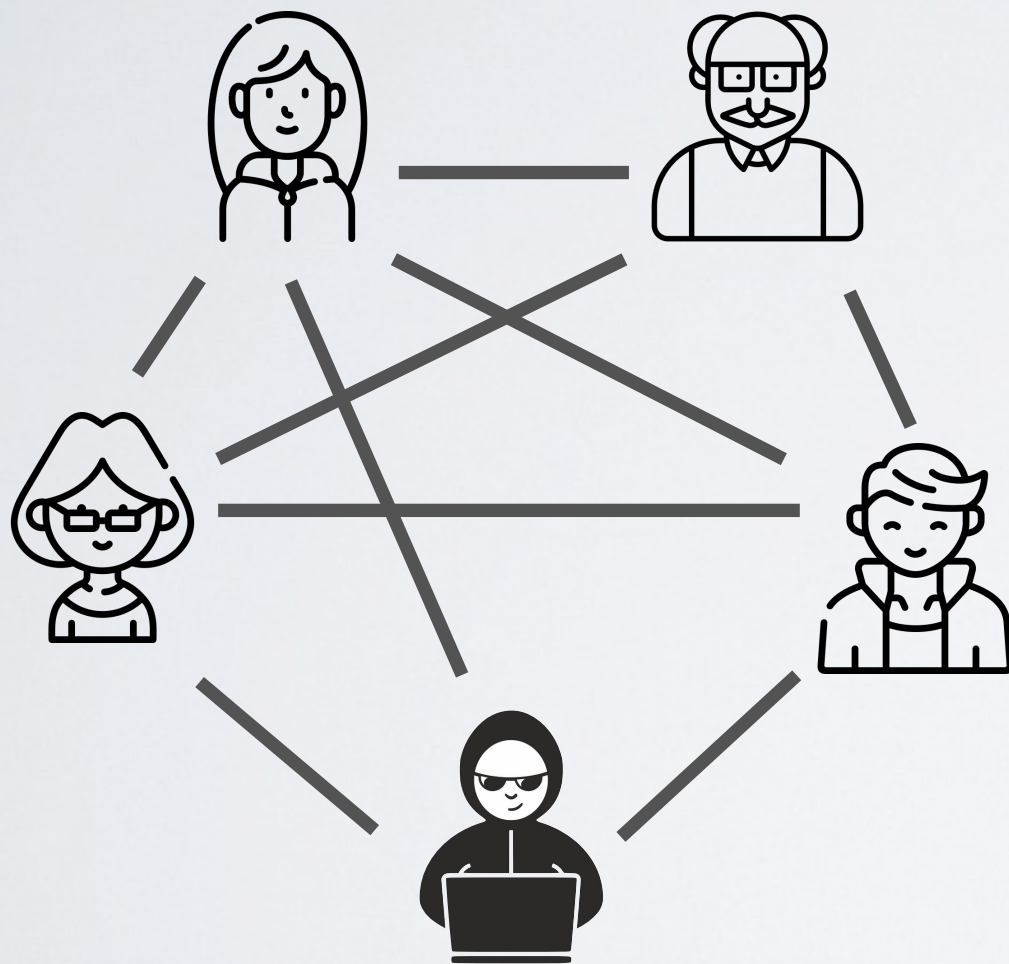
- ✓ Solving the Byzantine Agreement Problem by electing the node that will mine the next block
- ✓ Preventing a sybil attack by making the probability of being elected relative to the monetary power (staking)

# Election process

Electing one of the node to mine the next block requires **a beacon** : an ideal service that regularly publishes random value which no party can predict or manipulate



# Collect Random Approach



**Step 0** - Each node  $i$  generate a random number  $r_i$

**Step 1** - Each node commits its random number by broadcasting  $H(r_i)$

**Step 2** - Each node broadcasts its random  $r_i$  and verifies all committed hashes others

**Step 3** - Each node calculate the beacon value  
$$\text{beacon} = H(r_1 \parallel r_2 \dots \parallel r_n)$$

**Step 4** - The elected node mine the next block and broadcast it to the network

**Step 5** - The other nodes check the block  
If the block is invalid, the elected node lose its stake (slashing)

Beyond Pow and Pos



# Other consensus

- Proof of Stake variant based on pBFT - Practical Byzantine Fault Tolerant (Cosmos Tendermint)
- Proof of Authority ~ private network (Facebook Libra)
- Proof of History (Avalanche)
- DGA - Directed Acyclic Graph (IOTA)