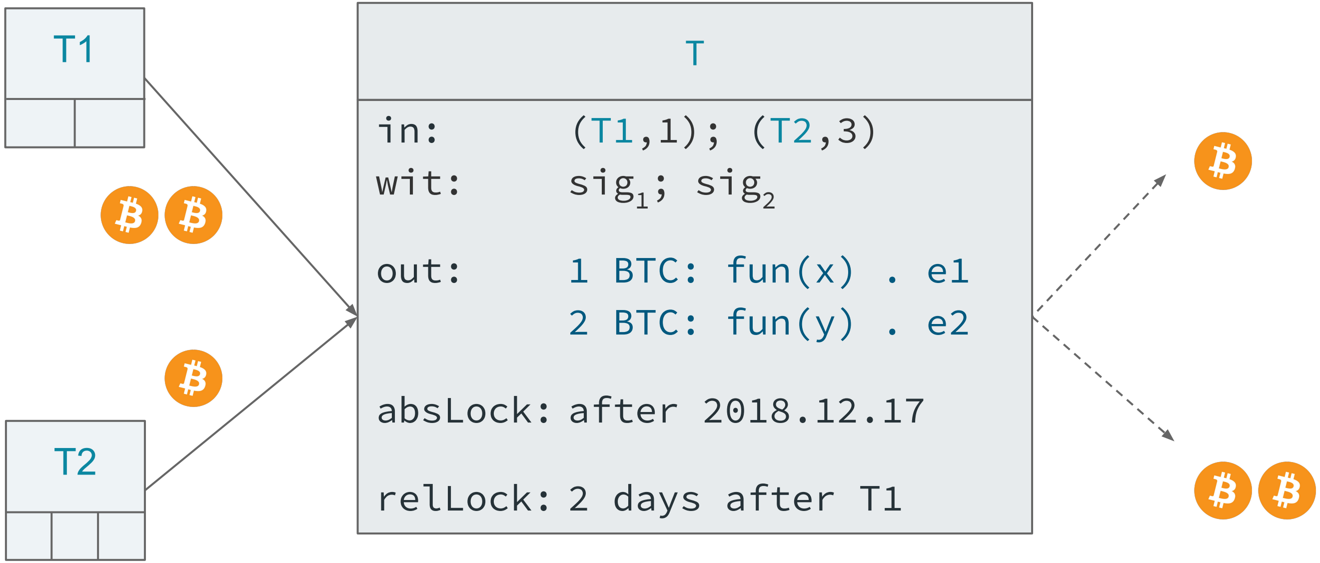
The security of Bitcoin

Stefano Lande
University of Cagliari
lande@unica.it

Bitcoin in depth

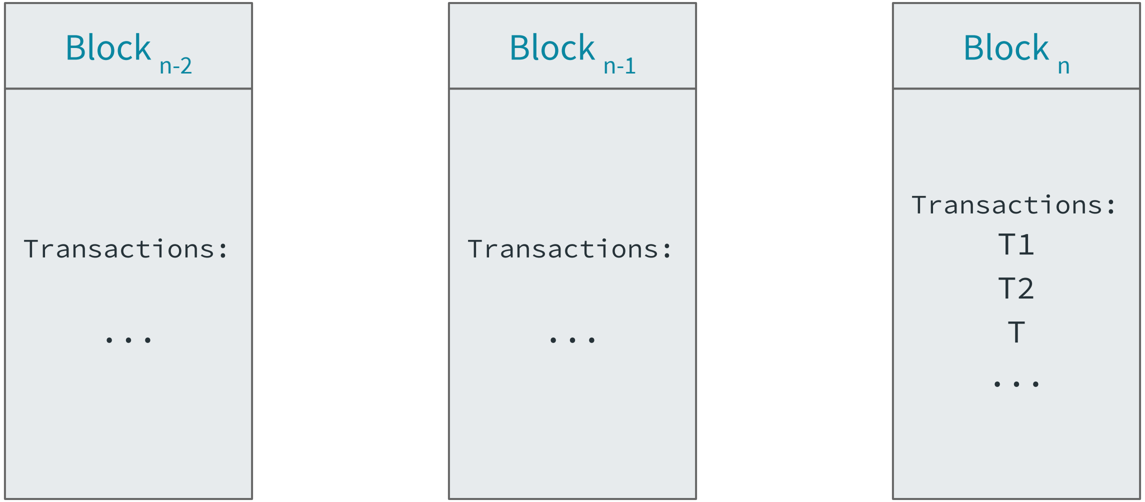
Bitcoin transactions

Clients submit *transactions* to the network



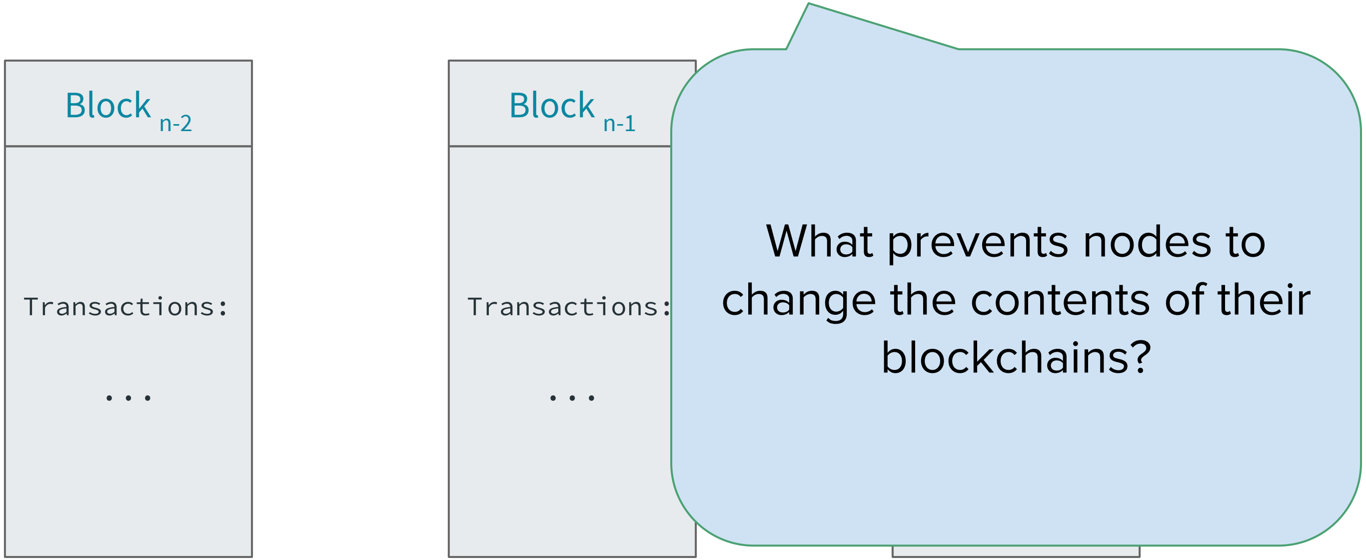
The blockchain

- A miner collects transactions into a *block*
- The block is propagated to the network
- Each miner add the new block to **his own blockchain**



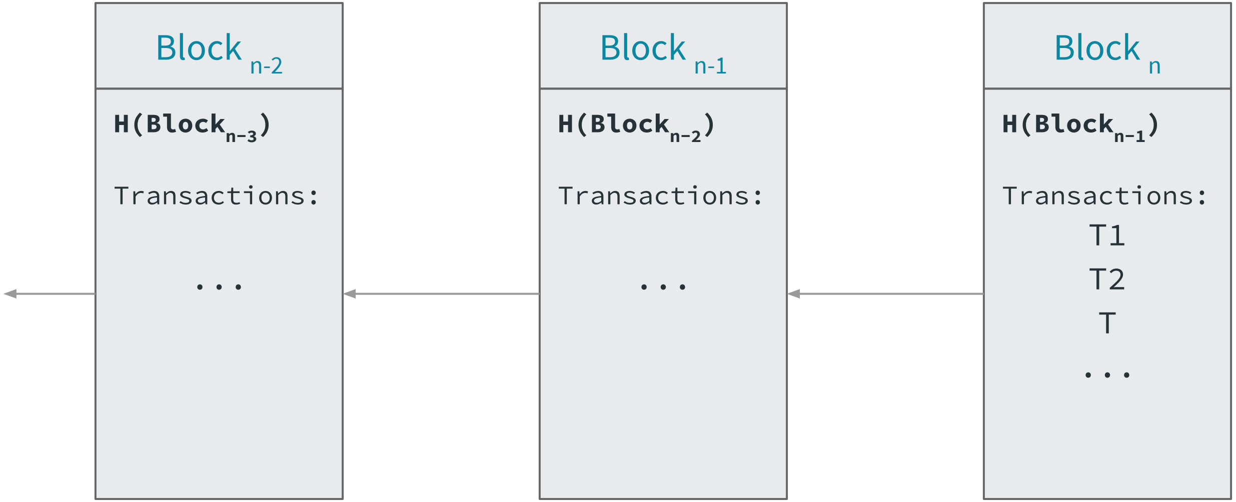
The blockchain

- A miner collects transactions into a *block*
- The block is propagated to the network
- Each miner add the new block to **his own blockchain**



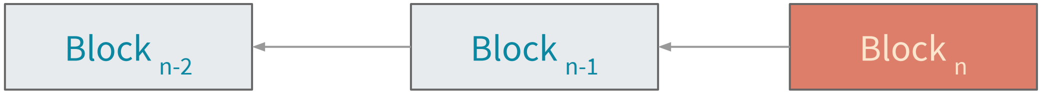
Immutability

- Each block is **hash-linked** to the previous one
- Tampering a block changes its hash
- Thus, the chain would be invalidated



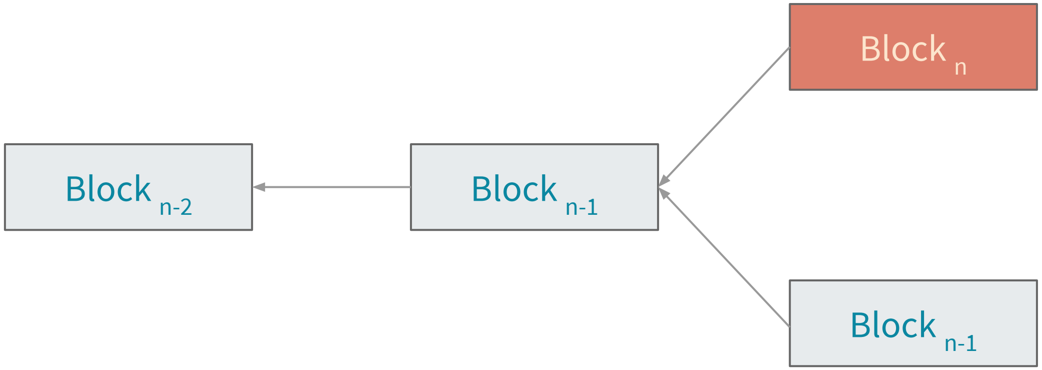
The consensus protocol

- Suppose an attacker broadcast a malicious block
- How the network reacts?



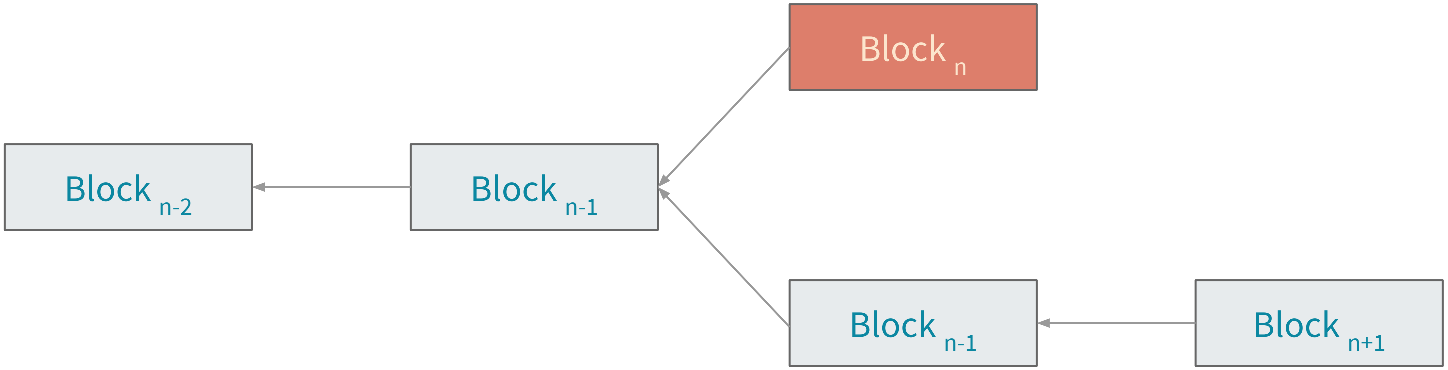
The consensus protocol

- Suppose an attacker broadcast a malicious block
- How the network reacts?
- Honest nodes ignore the malicious block (forking the blockchain)



The consensus protocol

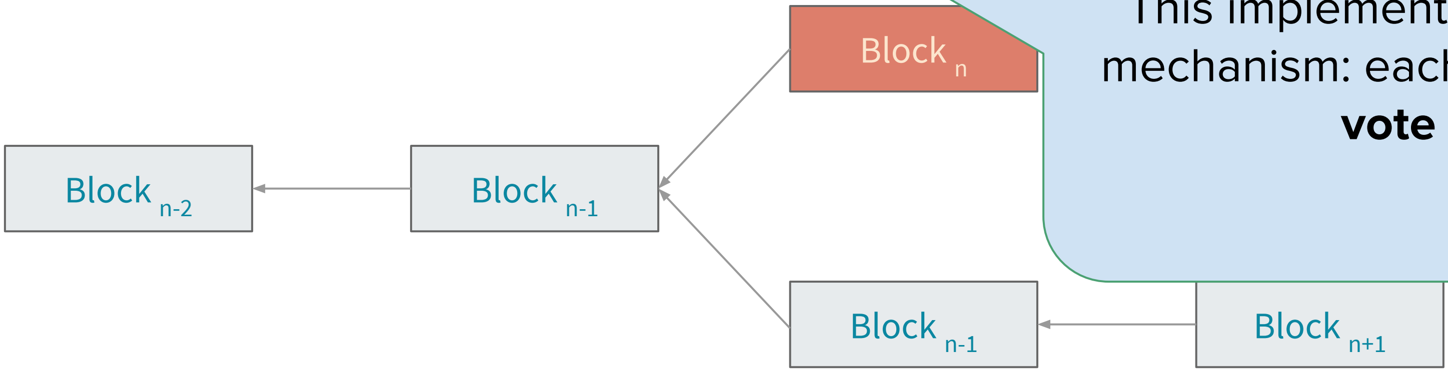
- Honest nodes continue building upon the honest brach
- After a period of time, the **longest branch** is considered the correct one



The consensus protocol

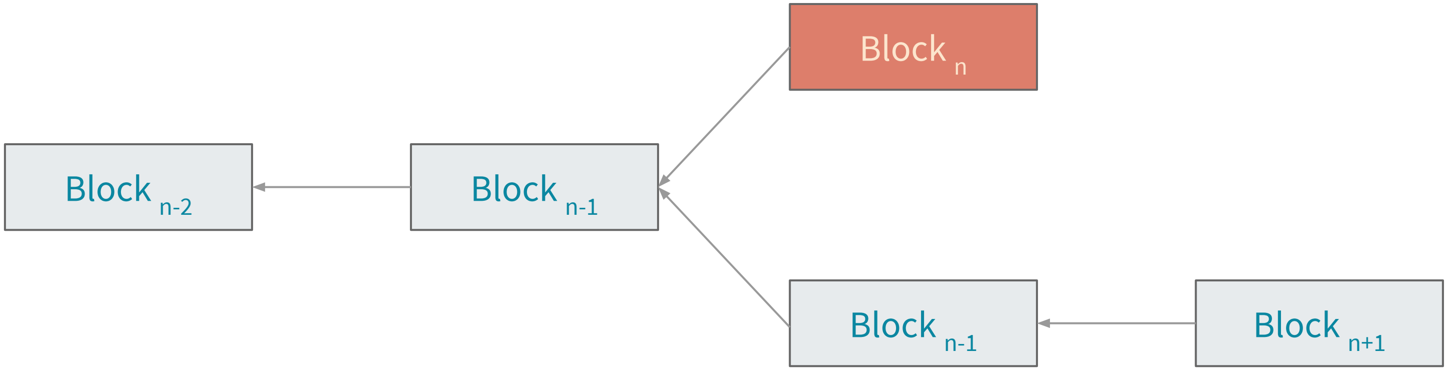
- Honest nodes continue building upon the honest brach
- After a period of time, the **longest branch** is considered the correct one

> This implements a voting mechanism: each block is a **vote**



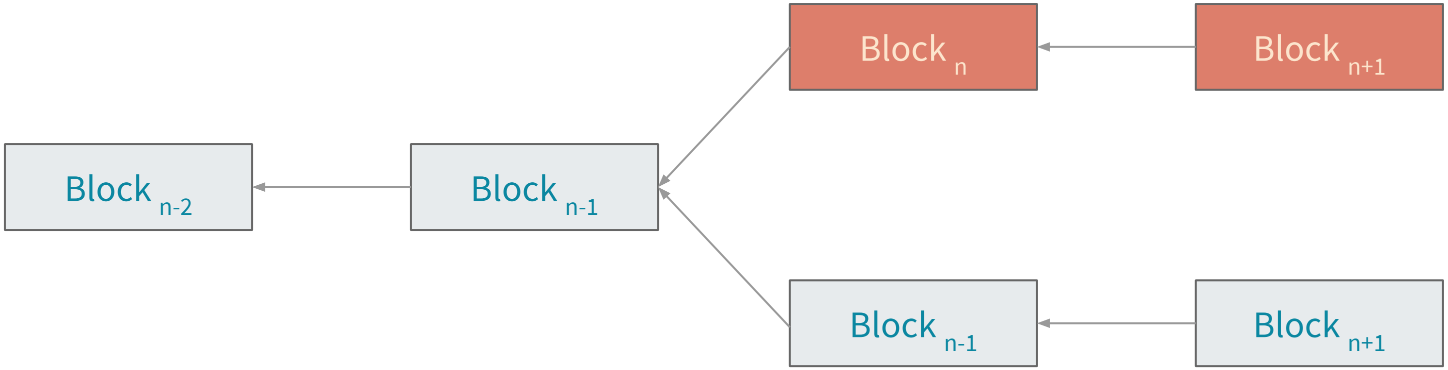
The consensus protocol

- But wait, creating nodes is free... (see Sybil attacks)
- So an attacker might control a large number of nodes to vote for the malicious branch



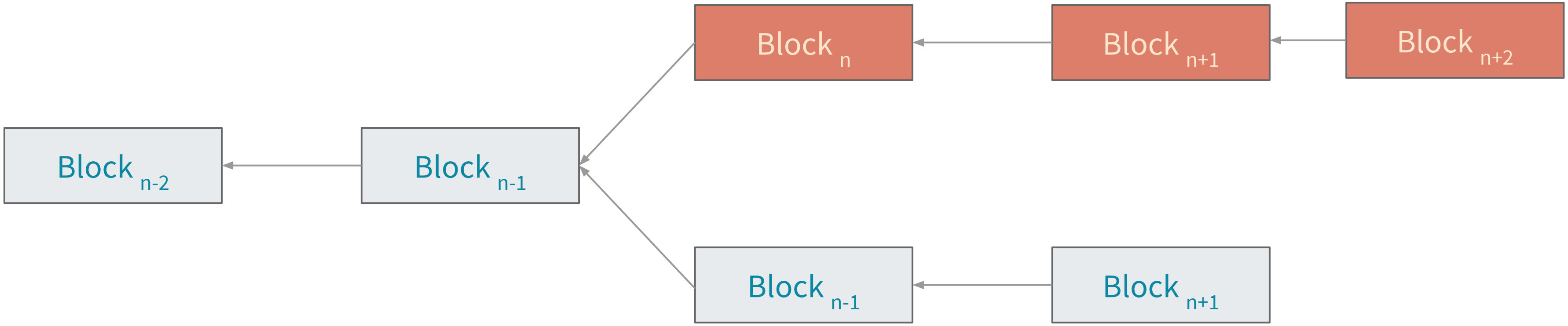
The consensus protocol

- But wait, creating nodes is free… (see Sybil attacks)
- So an attacker might control a large number of nodes to vote for the malicious branch



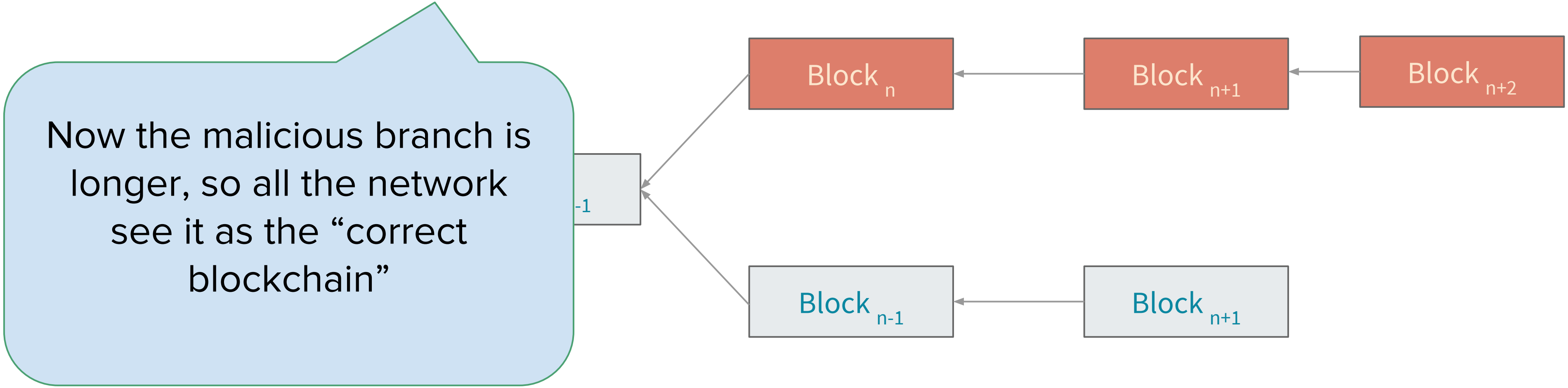
The consensus protocol

- But wait, creating nodes is free... (see Sybil attacks)
- So an attacker might control a large number of nodes to vote for the malicious branch



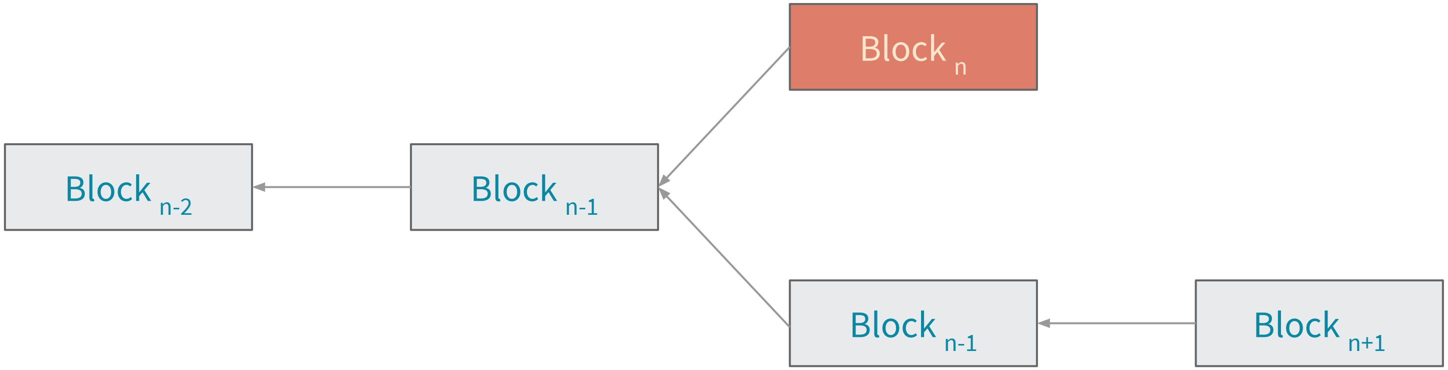
The consensus protocol

- But wait, creating nodes is free... (see Sybil attacks)
- So an attacker might control a large number of nodes to vote for the malicious branch



An anti-spam mechanism

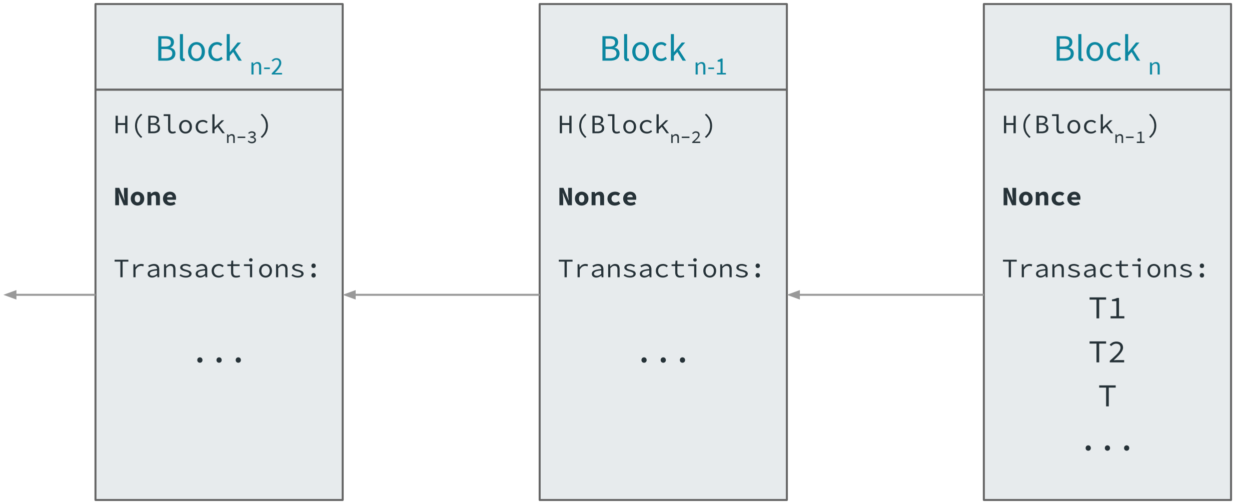
- Creating nodes is free but voting is not!
- Make block creation computationally expensive
  - ➔ “one CPU = one vote”



Proof of Work

- To be considered valid, a $block_n$ must contain a Nonce s.t

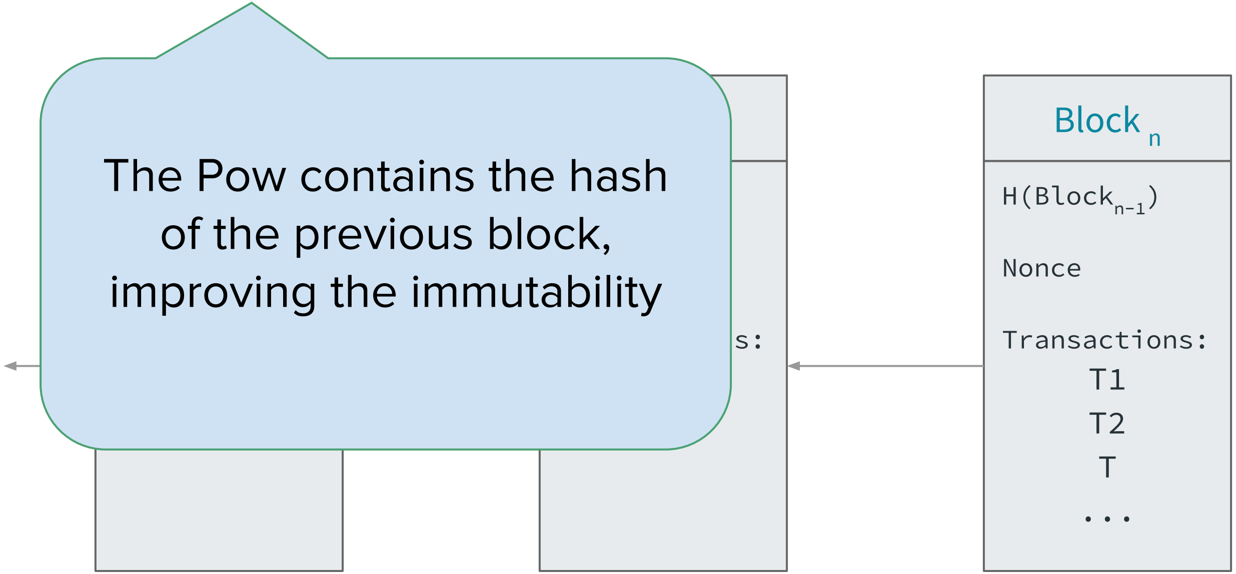
$$H(\ H(block_{n-1})\ ||\ \{T_i\}\ ||\ \mathbf{Nonce})\ <\ Target$$



Proof of Work

- To be considered valid, a $block_n$ must contain a Nonce s.t

$$\mathbf{H}(\ \mathbf{H}(block_{n-1})\ ||\ \{T_i\}\ ||\ \mathbf{Nonce})\ <\ Target$$



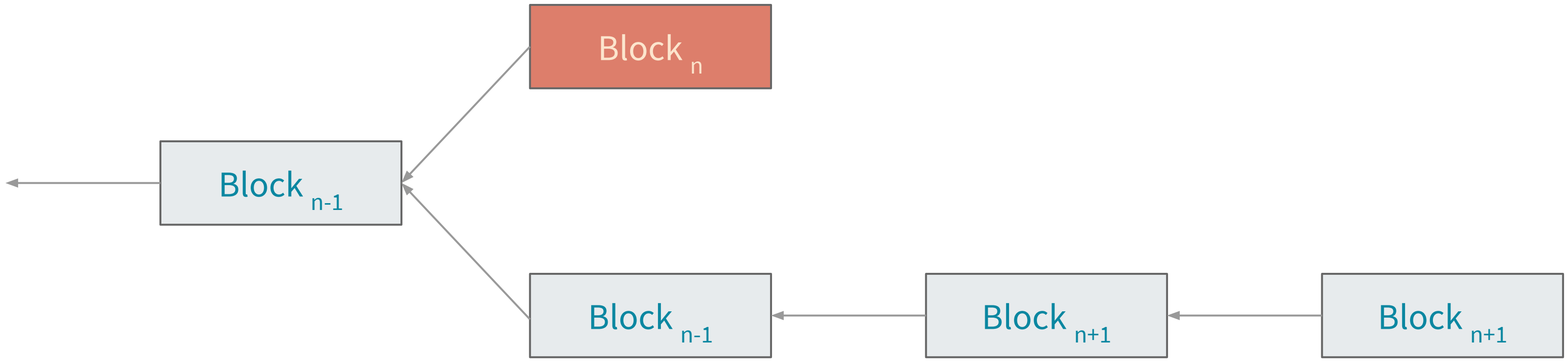
Proof of Work

- To be considered valid, a $block_n$ must contain a Nonce s.t

  $$\mathbf{H}(\ \mathbf{H}(block_{n-1})\ ||\ \{T_i\}\ ||\ Nonce) < Target$$

- If H is preimage resistant, finding the Nonce is possible only by **brute force (mining)**
- The difficulty is dynamically adjusted, so solving a PoW requires 10 minutes
  ➡ decrease Target as the total hashing power of the network increases

Proof of Work

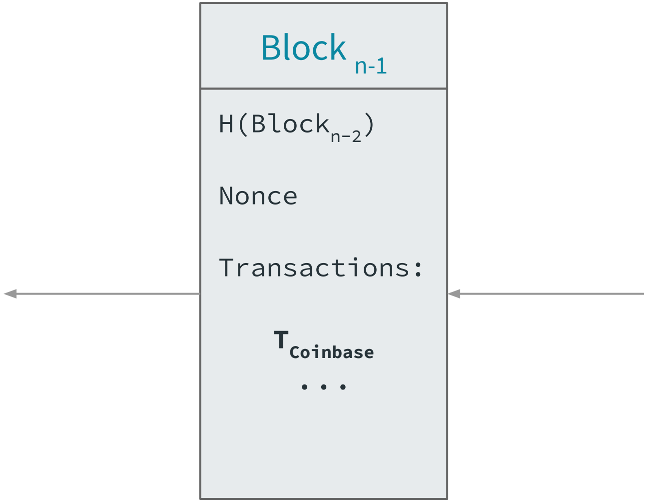


- More hashing power ➜ more voting power on the status of the blockchain
- Solving the PoW is computationally expensive

Why should nodes do that?

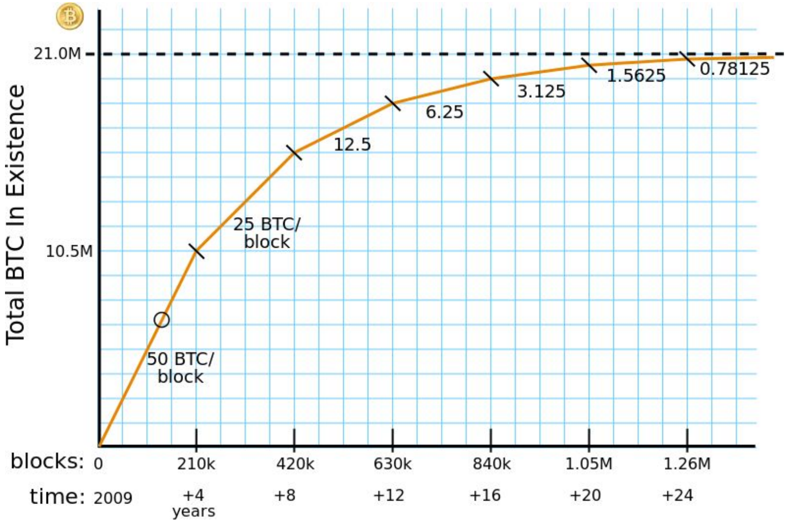
Incentive mechanism

- Each block creates a reward for the miner
- Explicit incentive: more blocks mined = more block rewards
- Implicit incentive: bitcoins would be worthless if the majority of miners is dishonest



Block reward

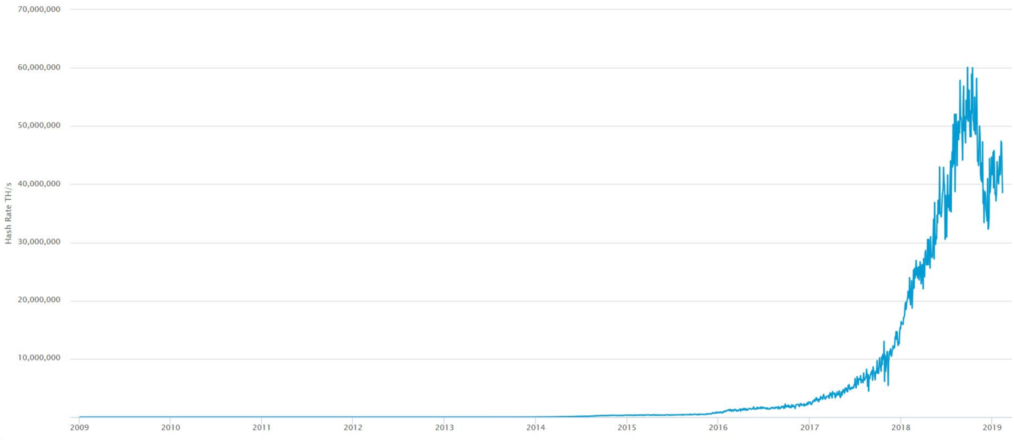
- The block reward was initially 50 btc
- It is set to halve every 4 years (now 12.5 btc)
  - The maximum supply of btc will converge to 21 millions btc
- Scarcity is necessary condition for a currency



Evolution of hashing

- At the moment, mining is profitable using only dedicated hardware
- The network power consumption is estimated to be 500 MW (Sardinia consumes ~ 1 GW)

Hardware	Introduction	Hash rate (h/s)
CPU	2009	10^5-10^8
GPU	late 2010	10^6-10^9
FPGA	Mid 2011	10^8-10^{10}
ASIC	Early 2013	10^{10}-10^{13}



Takeaways

- PoW is the **anti-spam mechanism**
- Longest chain is the **consensus mechanism**
- To determine the longest chain, nodes need to wait some time
  - Satoshi Nakamoto suggested 6 blocks

The Bitcoin Backbone protocol
(Garay et al.)

Theoretical analysis of the Bitcoin protocol

Motivation

- It is common to hear that Bitcoin is resistant if an attacker controls less than 50% of the total hashing power

- Is it really so simple?

Bitcoin as a turn-based game

- Time is divided in rounds
- In each round, each participant is allowed to query q times a random oracle
- Messages are sent through a “diffusion” mechanism
- The adversary can
    - spoof messages
    - inject messages
    - reorder messages

Modelling participants

- There are *n-t* honest participant
  - each one has q queries to the oracle per round
- The adversary controls t participant acting together maliciously
- Each participant has the same power ➔ flat interpretation

Desired property

k-common prefix:

$$\mathcal{C}_1^{\lceil k} \preceq \mathcal{C}_2 \text{ and } \mathcal{C}_2^{\lceil k} \preceq \mathcal{C}_1$$

If two players prune k blocks from their chains they obtain the same prefix

Preliminary definitions

- n ➔ number of participants
- t ➔ number of participants controlled by the attacker
- $p = D / 2^x$ ➔ probability to solve the PoW in a single query
- $\alpha = pq(n-t)$ ➔ expected solutions per round by honest participants
- $\beta = pqt$ ➔ expected solutions per round by the attacker

- $\gamma = \alpha - \alpha^2$ ➔ probability that at least one honest party computes a solution in a round
- $f = \alpha + \beta$ ➔ expected solutions per round by the whole network

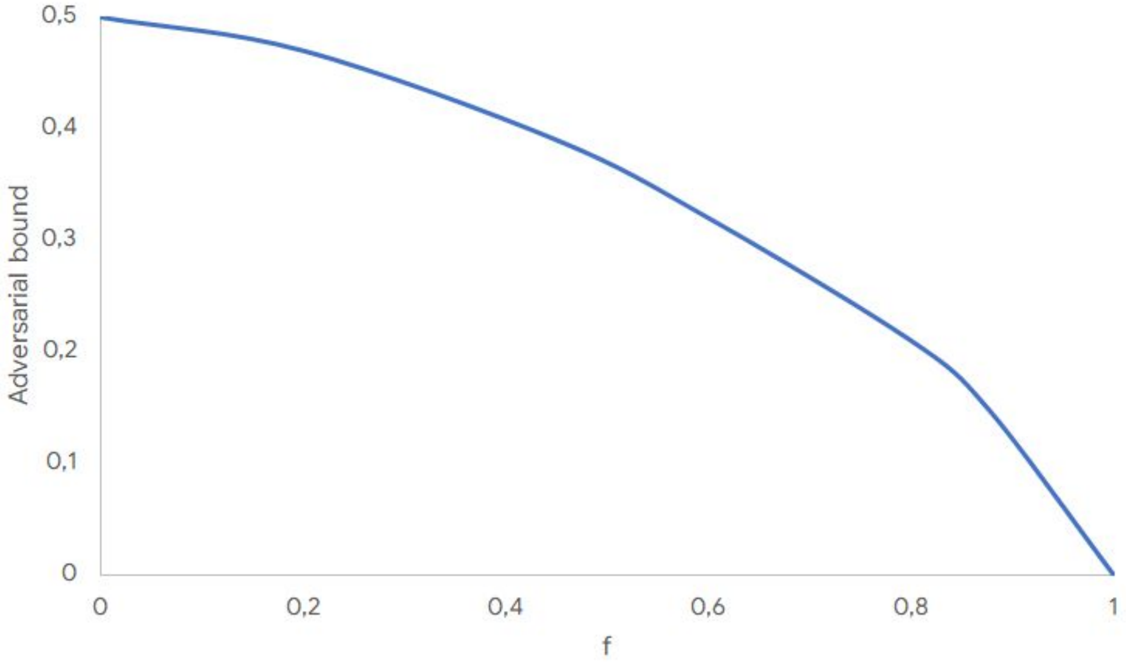
Theorem

Assume $f < 1$, if $\gamma > \lambda\beta$ for some $\lambda > 1$
that satisfies $\lambda^2 - f\lambda + 1 \geq 0$.

Let S be the set of chains of honest participants at a given round of the protocol.

The probability that S does not satisfy the k-common-prefix property is at most $e^{-\Omega(\lambda^3 k)}$

Graphical interpretation



Takeaways

- We saw that as $f \rightarrow 1$, the theorems provide no security guarantees
- f corresponds to the time required to solve the PoW compared to the network synchronization time
- Bitcoin is conservative by requiring 10 minutes, but
  - This harms scalability (high block time ➔ low transaction throughput)
  - An attacker can still “desynchronize” the network

Conclusion (not really)

The Bitcoin protocol satisfies *common prefix* and *chain quality* properties if the adversarial hashing power is less than ½

but only if the network is synchronous

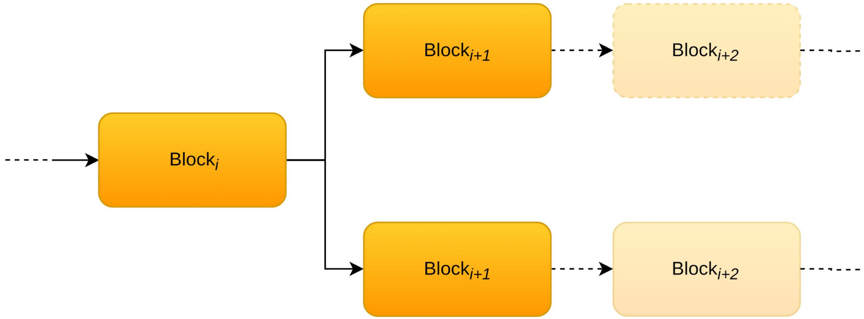
The selfish mining strategy

(Ittay and Gün Sirer)

A practical attack on Bitcoin

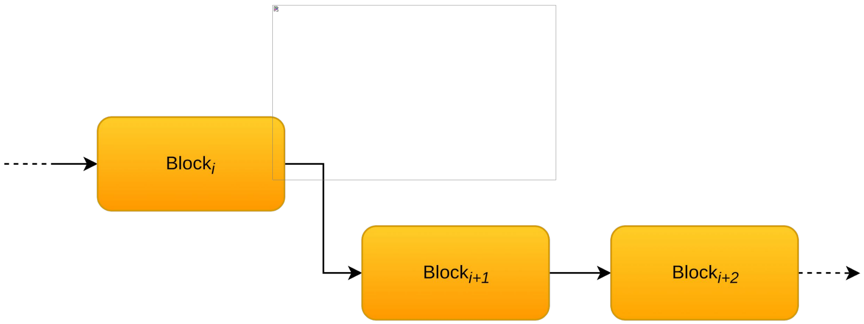
Non-malicious forks

- When two miners solve the Proof of Work at the same time, the blockchain is *forked* in two branches
- The other miners start to mine on the first block they receive from the network



Resolving forks

- One branch will eventually became longer than the other:
  - To resolve the fork, miners mine on the longest chain
- The shorter branch will be discarded
  - The work spent to mine its blocks is **wasted**
  - The block rewards are not collected

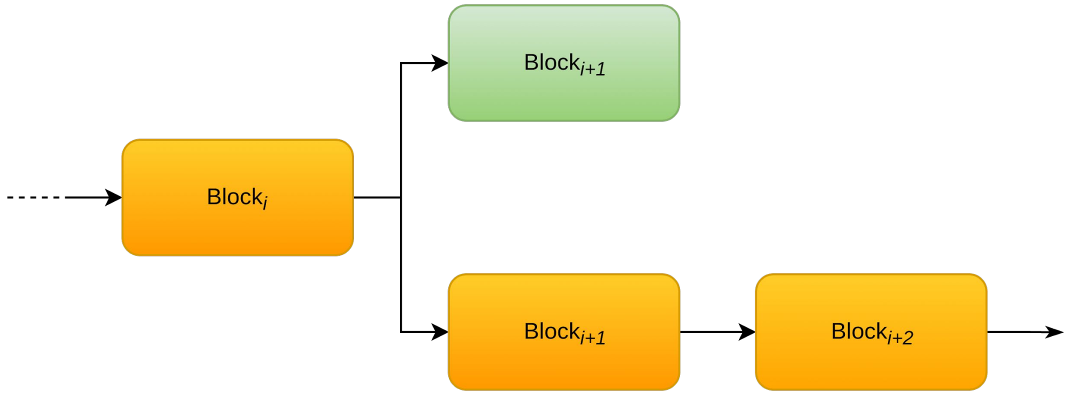


The Selfish-Mine strategy [3]

- The strategy allows a miner with sufficient power to obtain more revenue than its power ratio
- Force honest miners into performing computation on a branch that will be discarded
- How?
  - Keep newly discovered blocks private to create a private branch
  - Broadcast them strategically to invalidate honest miners work

Algorithm - 1

- When the private branch is shorter than the public branch, the attacker adopt the latter

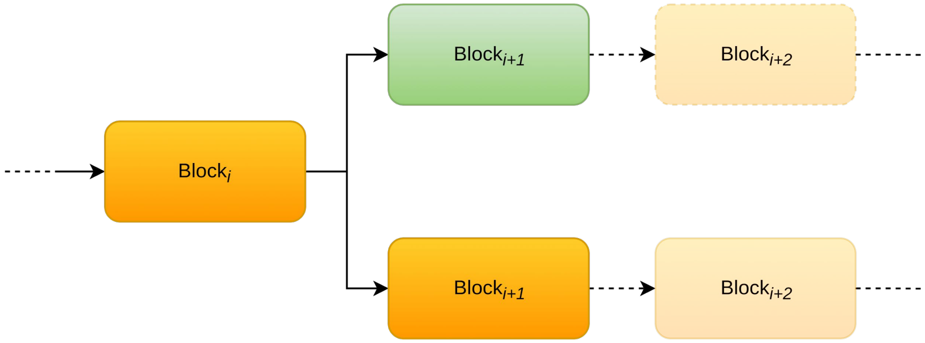


Algorithm - 2

- When the attacker finds a block, it keeps it private
- Outcomes:
    a. The honest miners find a block, nullifying the lead
    b. The attacker finds another block and extends the lead

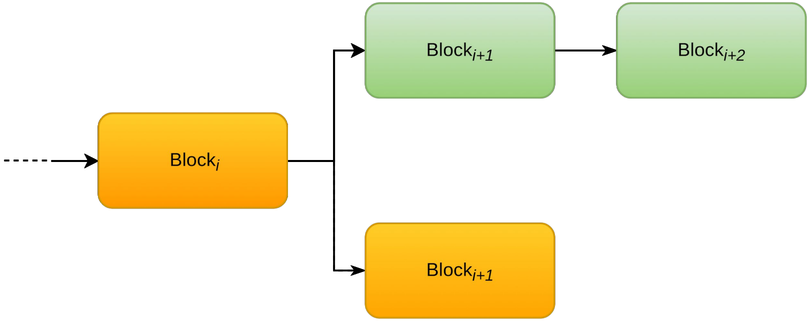
Algorithm - outcome a

- The honest miners find a block, nullifying the pool lead
- The attacker publishes immediately the private block:
  - The attacker continue to mine from the previously private block
  - The honest miners mine from either block, depending on which they receive first



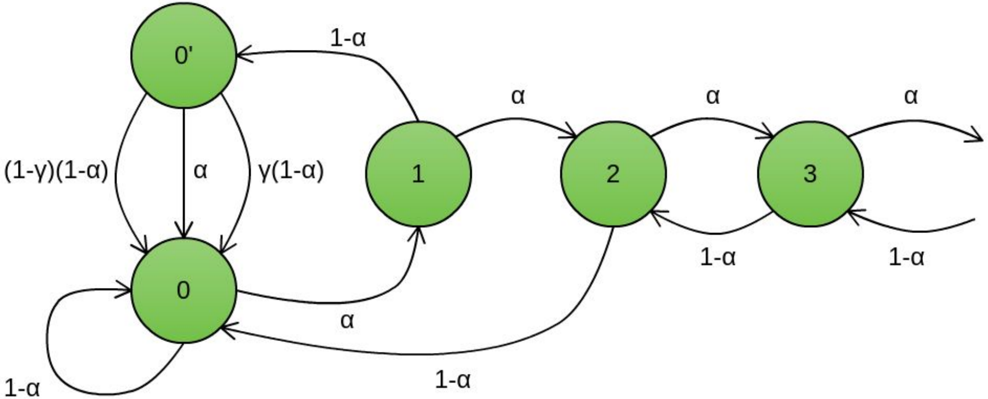
Algorithm - outcome b

- The attacker finds another block and extends the lead
- The attacker publishes a block for each block the honest miners find
- When the lead reduces to a single block, publish all the private branch
  - All the miners discard the shortest branch
  - If all the blocks in the private branch are published, the algorithm is back to the initial case

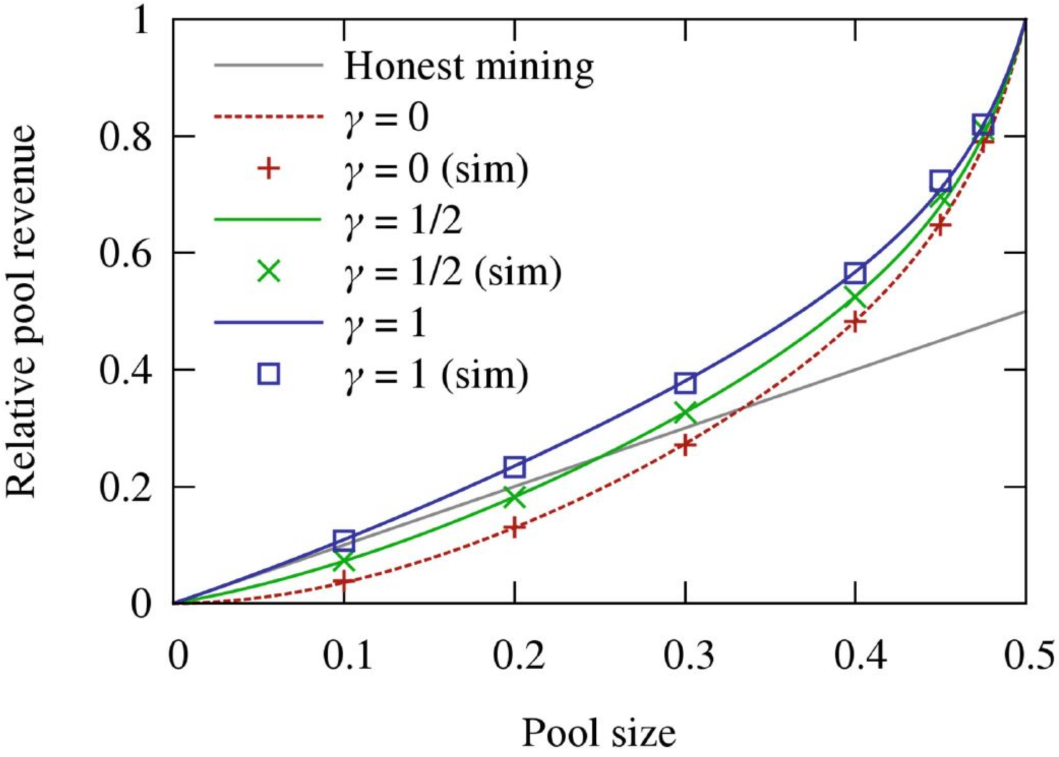


Analysis

- α: mining power of the attacker
- (1-α): mining power of the honest miners
- γ: ratio of honest miners that choose to mine on the attacker fork
- (1-γ): ratio of honest miners that choose to mine on the other fork



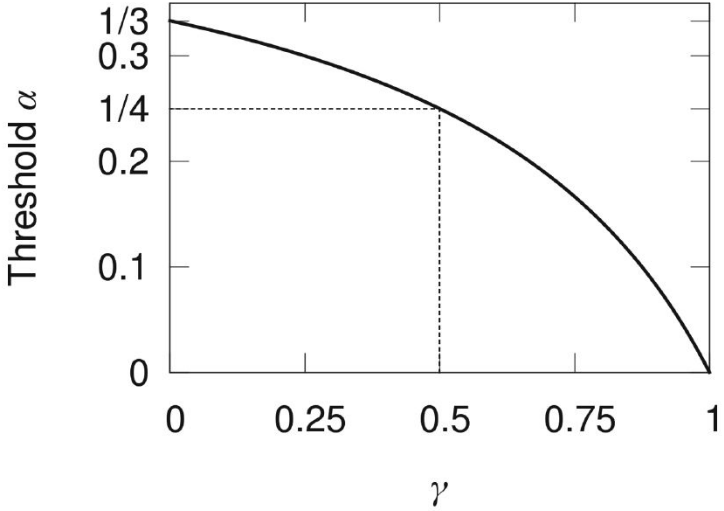
Results - 1



Honest mining
γ = 0
γ = 0 (sim)
γ = 1/2
γ = 1/2 (sim)
γ = 1
γ = 1 (sim)
Relative pool revenue
Pool size
0
0.2
0.4
0.6
0.8
1
0.1
0.3
0.5

Results - 2

- The graph shows the minimum power the attacker need to trump the protocol
- Even with γ = 0 (unrealistic) the threshold is ⅓
- γ can be easily increased with **zero-power nodes** (e.g., a botnet)



Consequences

- Once an attacker exceeds the threshold, it can increase its revenue by running the selfish mine algorithm
- Rational miner will join the attacker to increase their revenue
- The pool grows towards majority, gaining the control of the blockchain

Conclusion

- The theoretical analysis shows that the protocol withstands an attacker with up to 50% of the total hashing power only under a strong synchronicity assumption

- With the selfish-mining algorithm one can attack Bitcoin without controlling more than 50% of the total hashing power

Majority is not enough

References

- Satoshi Nakamoto. “Bitcoin: A Peer-to-Peer Electronic Cash System”
- Atzei, Bartoletti, Lande, Zunino. “A formal model of Bitcoin transactions”
- Garay, Kiayias, Leonardos. “The Bitcoin Backbone Protocol: Analysis and Applications”
- Eyal, Gün Sirer. "Majority is not enough: Bitcoin mining is vulnerable"