CS251 Fall 2020

(cs251.stanford.edu)

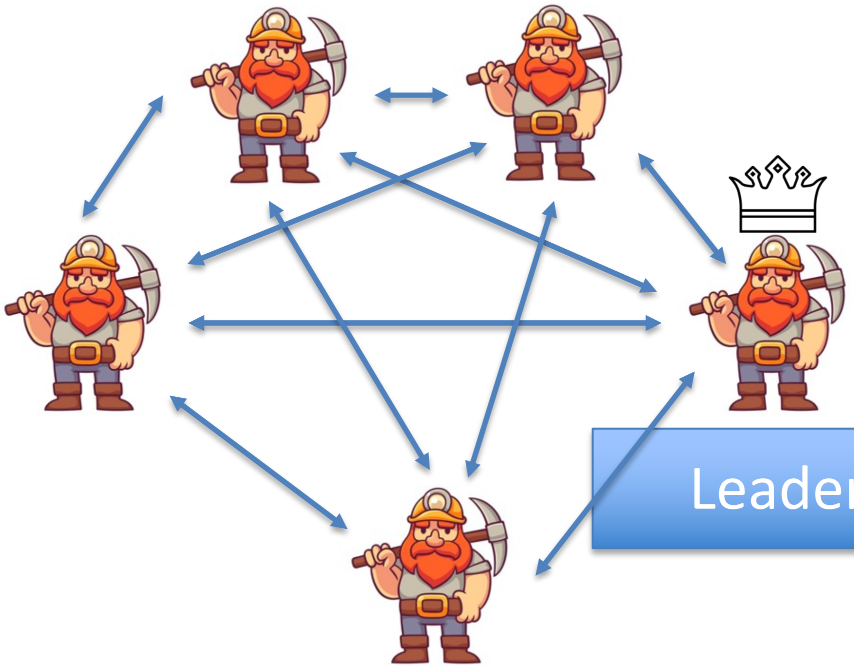
Nakamoto Consensus

Benedikt Bünz

Consensus

- Security Properties:
  - Consistency: Honest nodes do not contradict
  - Liveness: Progress is made
- Network Models
  - Synchronous: Messages get delivered immediately
  - Partially Synchronous: Messages are out of order

Consensus

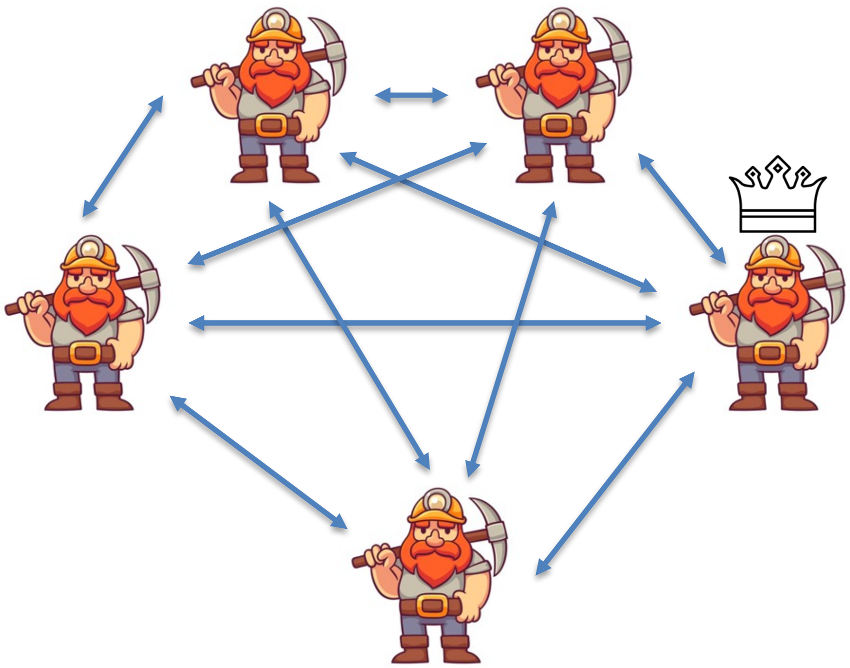


Accepts/Rejects TXs

Leader

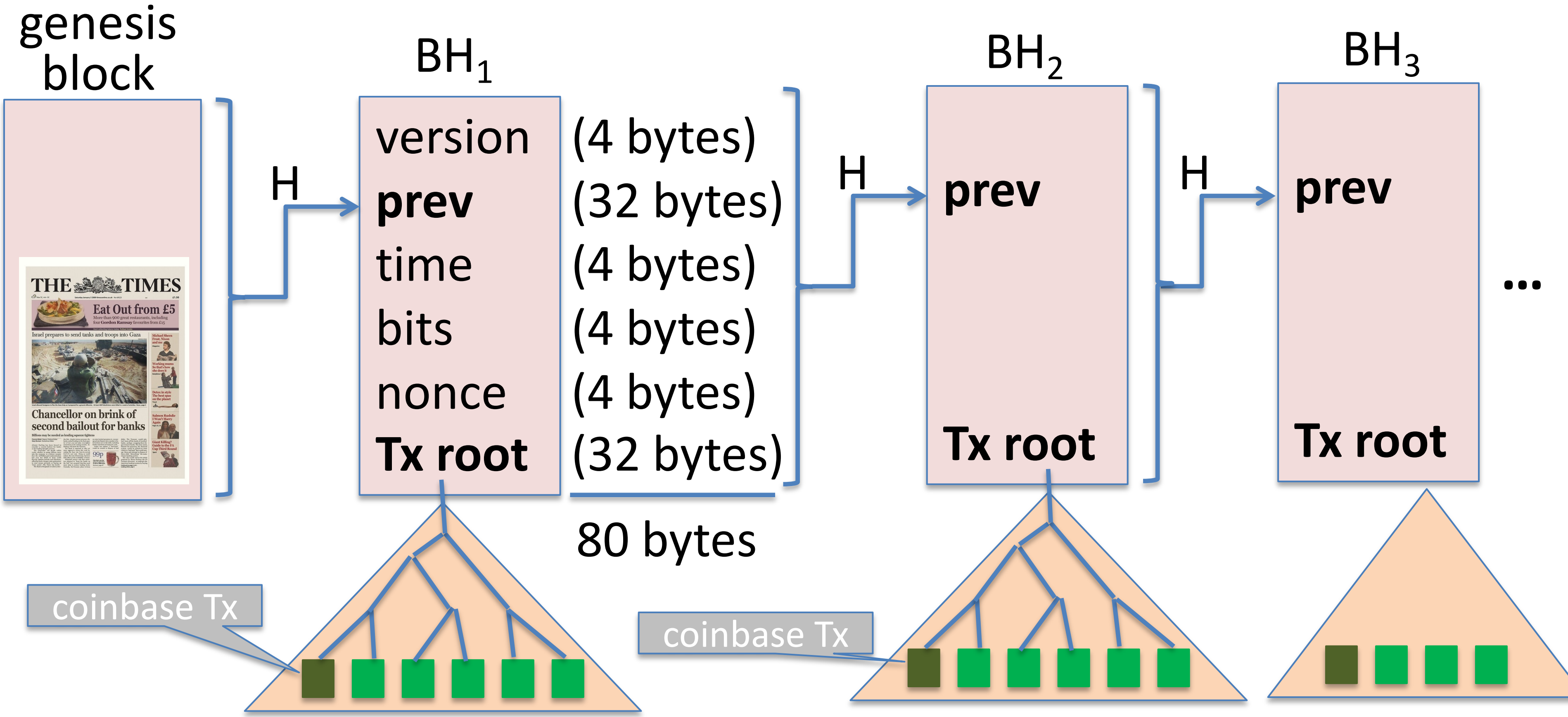
Committee

Problems with approach

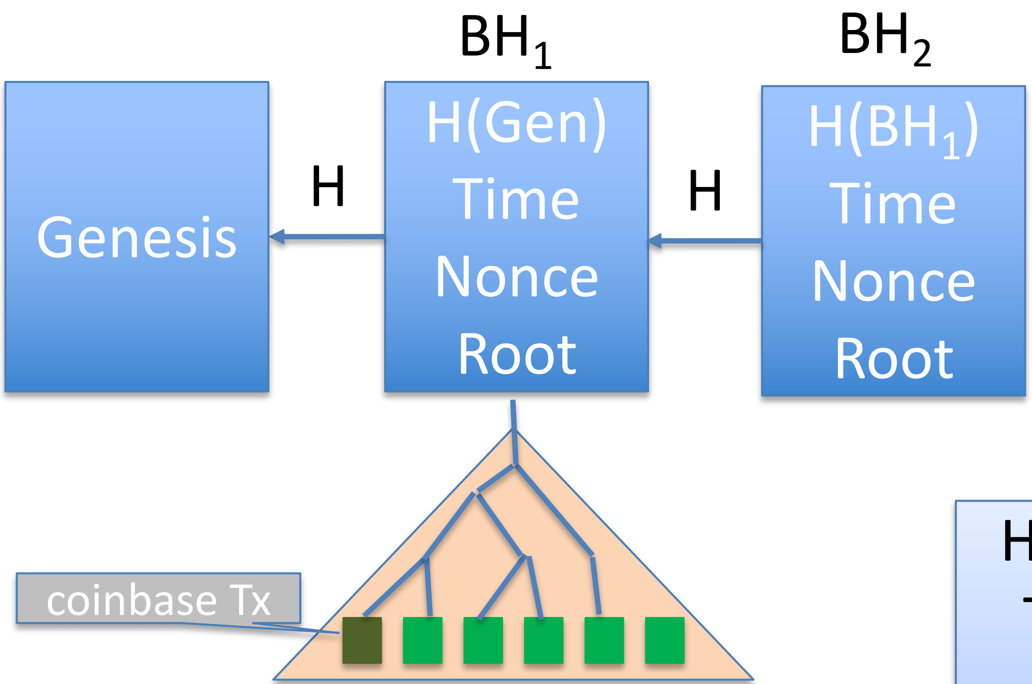


- Known committee
  - (must communicate)
- Large committee
  - Large communication
- Honest majority (incentives)
- Predictable Leader
  - Bribing 💸

Recap

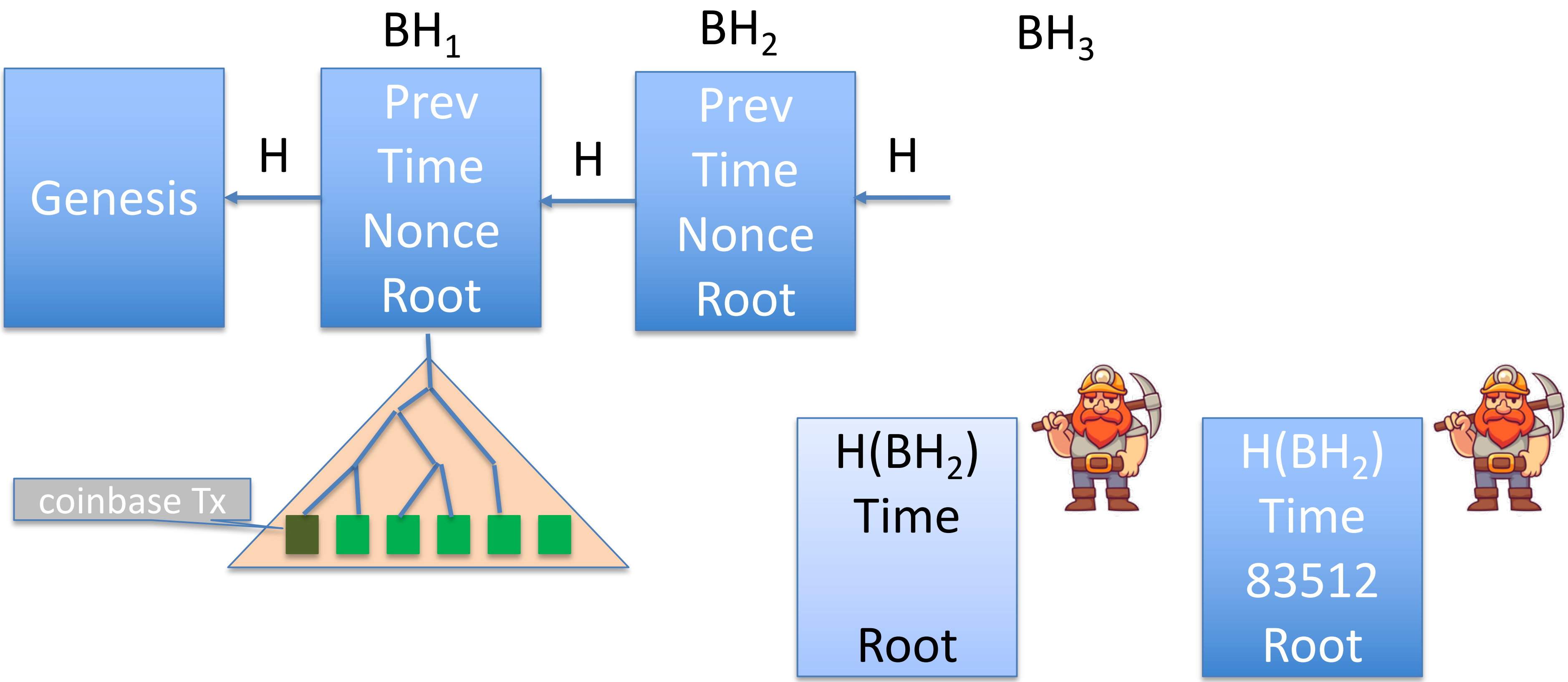

genesis block
THE TIMES
Eat Out from £5
Chancellor on brink of second bailout for banks
H
BH_1
version (4 bytes)
prev (32 bytes)
time (4 bytes)
bits (4 bytes)
nonce (4 bytes)
Tx root (32 bytes)
80 bytes
H
BH_2
prev
Tx root
H
BH_3
prev
Tx root
…
coinbase Tx
coinbase Tx

Nakamoto Consensus

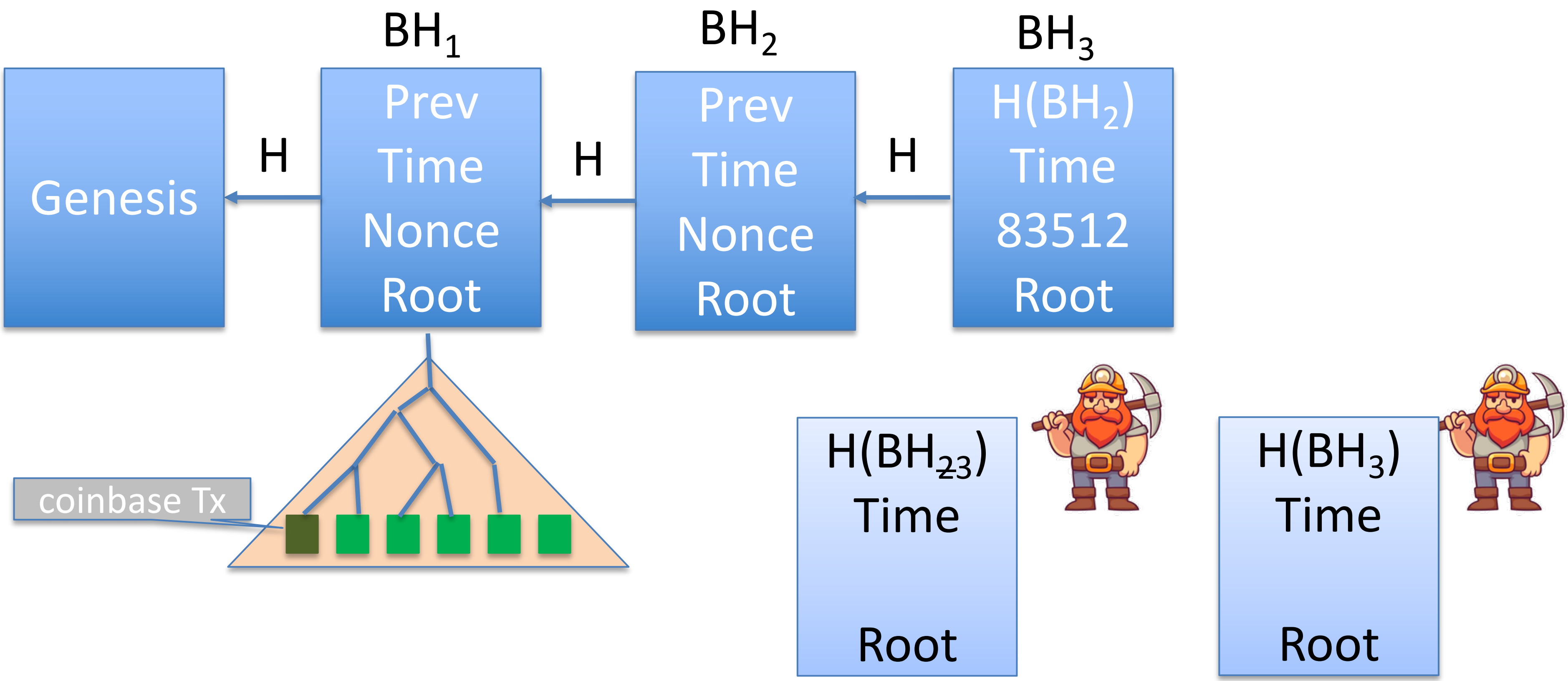


PoW:
Find nonce s.t.
H(Block)<Target
Target s.t. blocks
found every 10 min

Nakamoto Consensus



Nakamoto Consensus



Nakamoto Consensus

- Miners “race” to add blocks
  - Prepare Block Template
  - Find nonce (PoW solution)
  - One winner every ~10 min
  - Target adjusted every 2 weeks
  - Probability winning ~ Computation power

PoW:
Find nonce s.t.
H(Block)<Target

Prev Time Root	Prev Time Root

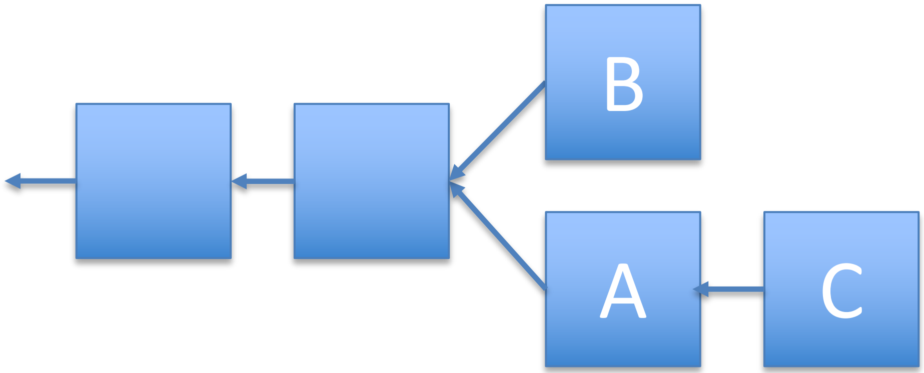
Nakamoto Consensus

- Miners “race” to add blocks
  - Need to find PoW solution
  - Probability winning ~ Computation power
  - One winner every ~10 min
  - Target adjusted every 2016 Blocks
  - On average 2016 blocks = 2 weeks
- (Honest) miners extend longest chain
- Timestamps must be roughly accurate
- *All transactions must be valid*
- Blocks/Transactions become final after *k* blocks
- Leader election/race combined with tx adding

PoW:
Find nonce s.t.
H(Block)<Target

Prev Time Root	Prev Time Root

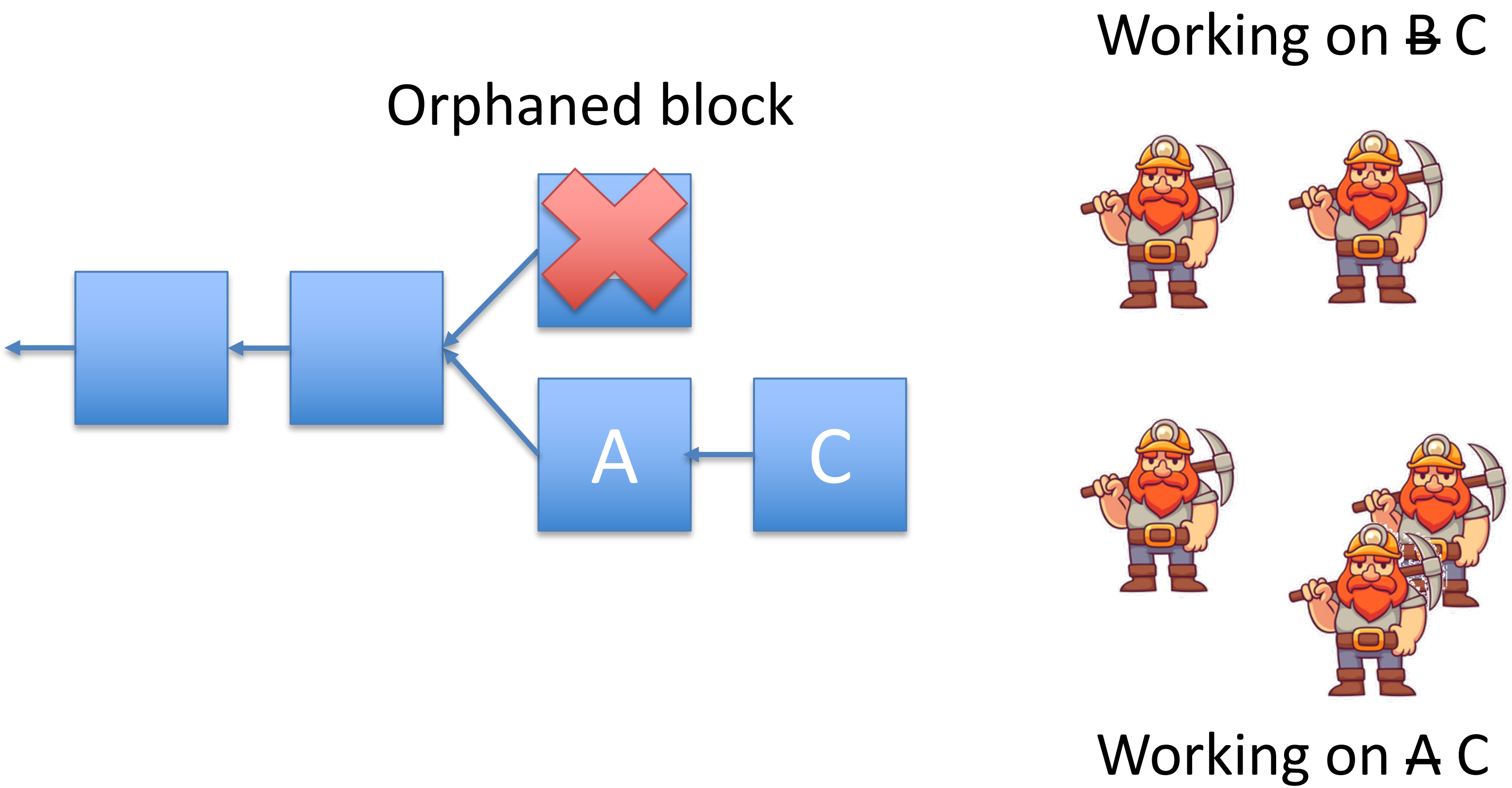
Forks and Orphans



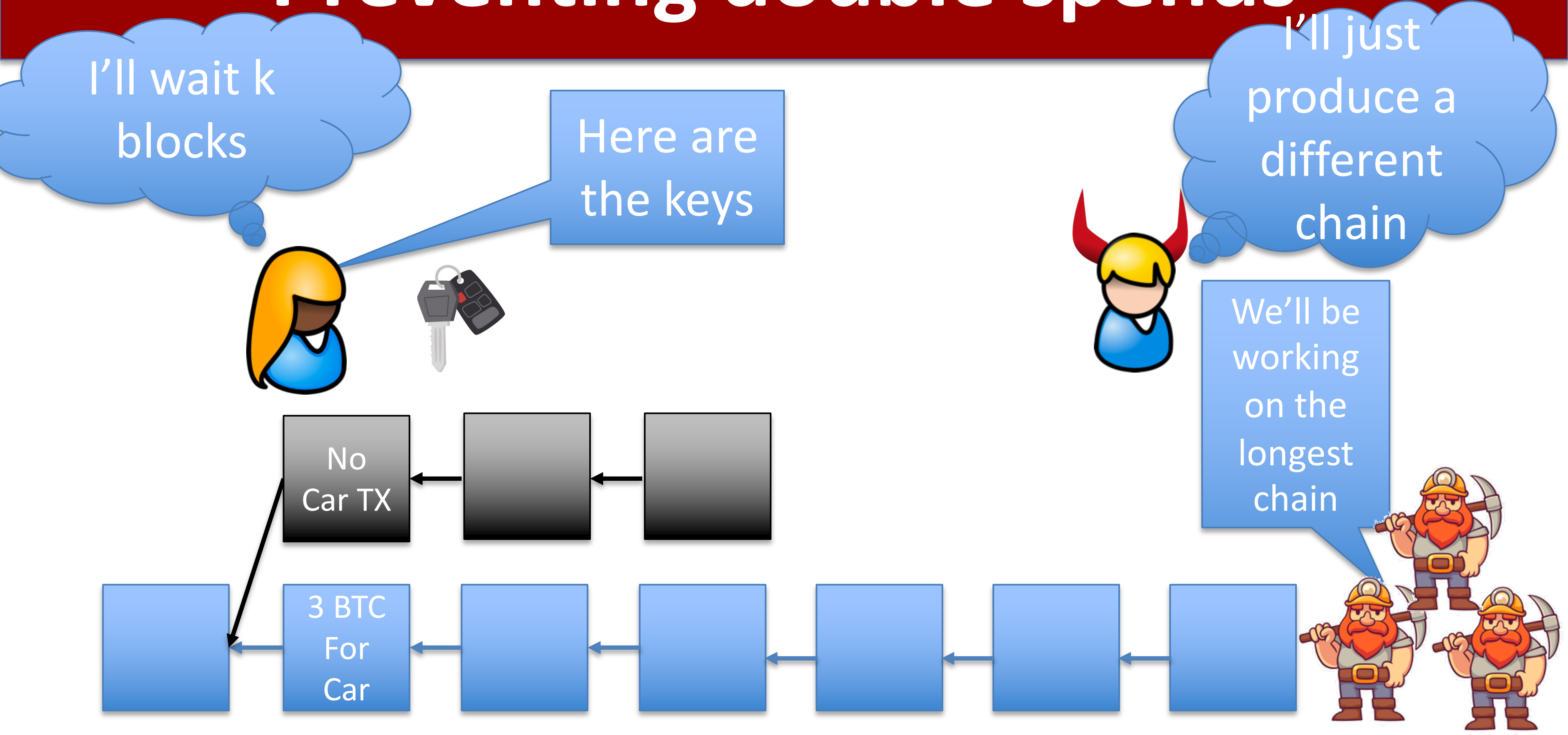
Working on B

Working on A

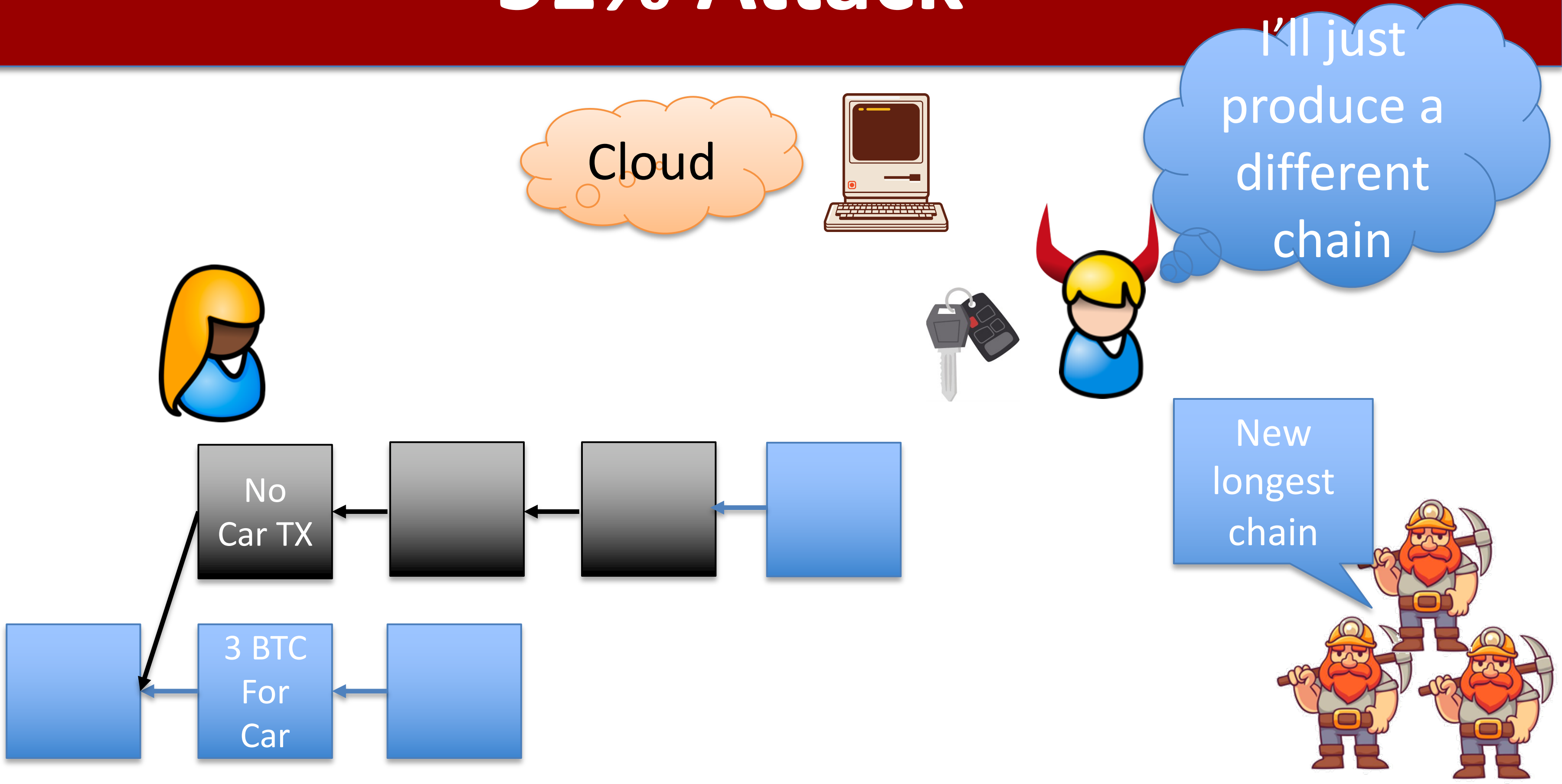
Forks and Orphans



Preventing double spends



51% Attack



I'll just produce a different chain
Cloud
New longest chain
No Car TX
3 BTC For Car

Nakamoto properties

1. **Consistency.** Honest nodes agree on all but last **k** blocks (except with prob. $O(2^{-k})$)
2. **Chain quality.** Any consecutive **k** blocks contain "sufficiently many" honest blocks (except with prob. $O(2^{-k})$). *Miners controlling p fraction of power should roughly mine p fraction of blocks.*
3. **Chain growth.** Chain grows at a steady rate.

    g-chain growth: Growth by k blocks every k/g "rounds"

Nakamoto properties => Blockchain

- Consistency implies Blockchain consistency

- Chain growth + chain quality implies Blockchain liveness
  - The chain grows by k blocks every k/g periods
  - By chain quality, a high fraction of blocks are contributed by honest miners, and therefore include all transactions they heard so far

Nakamoto consensus

<u>Consistency intuition:</u> Suppose adversary has 49% power

- Adversary can fork chain by 1 block faster than honest miners extend current chain w/ prob. close to ½, or by 2 with prob. ¼
  - No problem! If adversary broadcasts fork, everyone switches, this is now the longest chain
- What if miner forks chain 6 blocks deep and doesn't broadcast until it has a longer chain than honest?
  - Probability 1/64 it mines 6 blocks before honest mines 1
  - Probability $< 8 * 2^{-7}$ it mines 7 blocks before honest mines 2
  - What is probability adversary ever catches up?

Nakamoto consensus

<u>Consistency intuition:</u> (continued…)

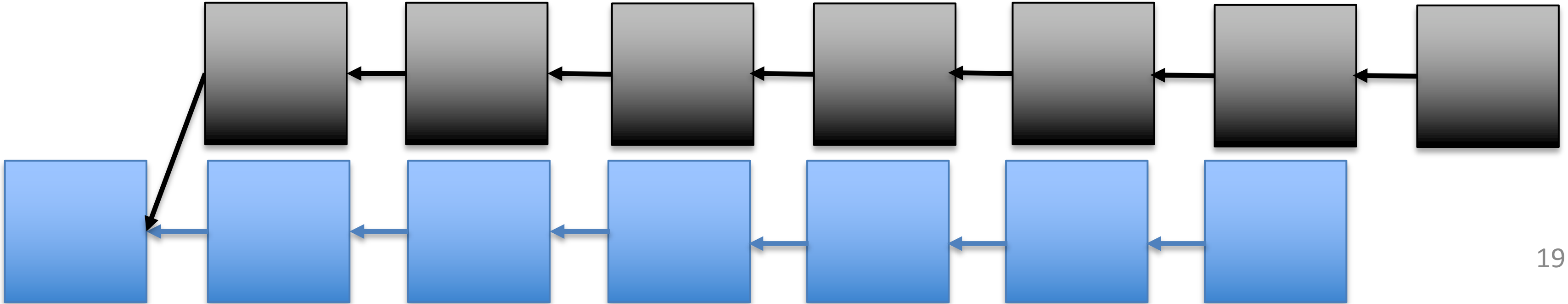
Suppose adversary has $p < 1/2$ fraction of power. What is the probability adversary catches up from 6 blocks behind?

- *Simplified model:* repeated rounds, in every round adversary catches up by 1 block with probability p, and falls behind by 1 block with probability $1 - \mathrm{p}$.
- Biased random walk on number line starting at 0, +1 with probability p and -1 with probability $1 - p$. Probability walk ever reaches 6?
- Probability P_z that walk ever reaches +z is $(\frac{p}{1-p})^z$ (e.g. p = 1/3, then P_6 < 0.0062)

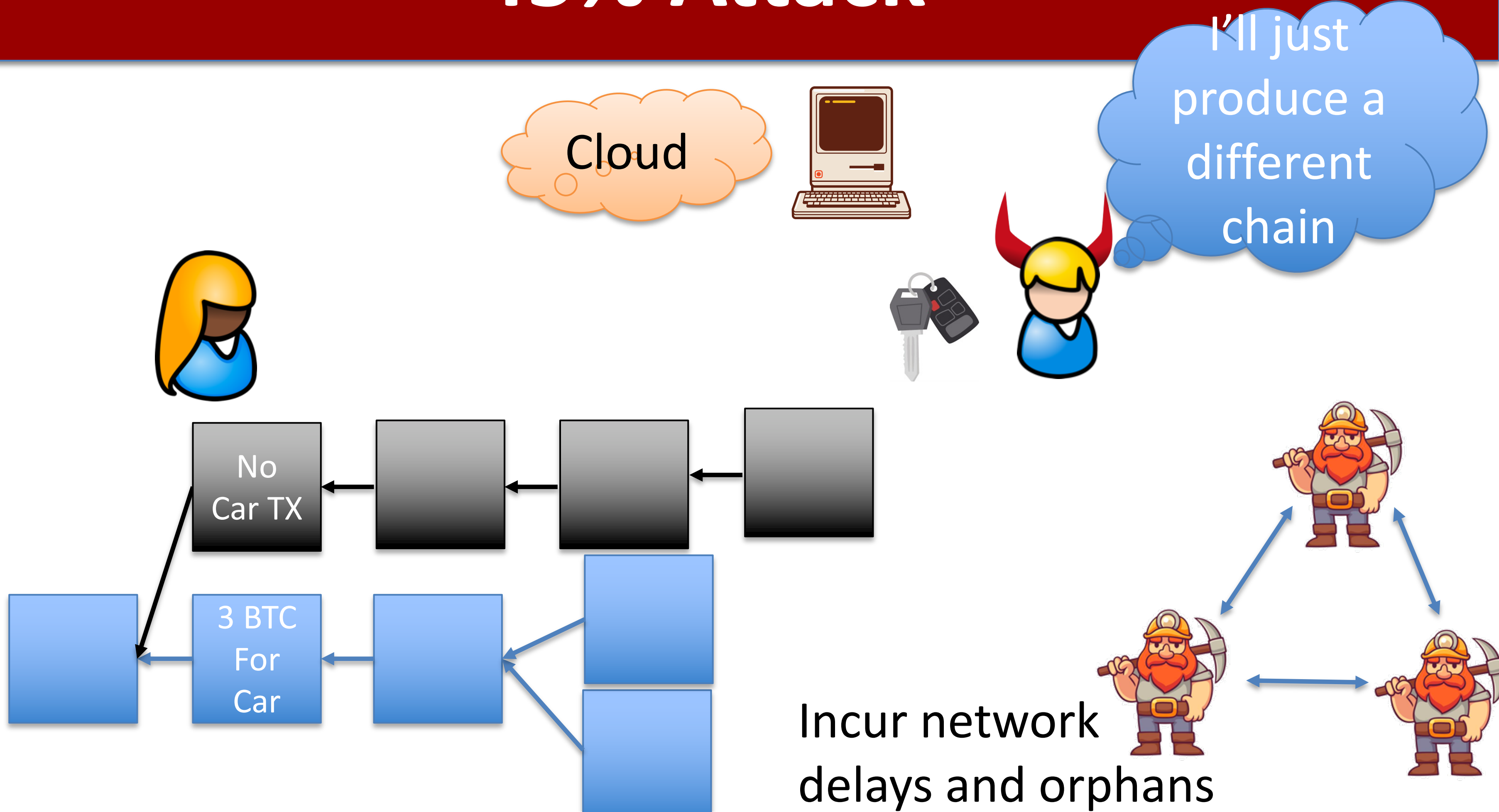
Nakamoto consensus

What goes wrong if adversary has $p > 1/2$ power?

- Adversary's private fork grows at faster rate than honest chain
- For any k, adversary starts k blocks behind, will eventually catch up to length of honest chain



45% Attack



Incur network delays and orphans

Nakamoto consensus

Network delay & work difficulty

- What happens if miners can solve puzzles faster than they can propagate solutions through network?
- Adversary might receive the next valid block Δ steps ahead of the other honest nodes ($\Delta = \text{delay}$)

$\Rightarrow$ *Adversary starts working on next puzzle with a Δ time head start over other honest nodes*

$O(\Delta)$ "free" hash trials

Nakamoto consensus

Adjusting difficulty for Δ

Formula from [SZ15]

Honest mining fraction (say 60%)

Network mining rate (1 Block/10min)

Adversary power (40%)

$$\frac{\alpha}{1+\alpha\lambda\Delta} > \beta$$

$\lambda\Delta$ is the mining rate * the delay. That is #blocks/delay (say 0.1)

Intuition:

On average, honest nodes waste a Δ steps of work every block they find, while adversary never wastes work. So "effective" reduced honest rate is $\frac{\alpha}{1+\alpha\lambda\Delta}$

DKT+ Theorem

Theorem: There exists a k such that Nakamoto Consensus has consistency and liveness if and only if:

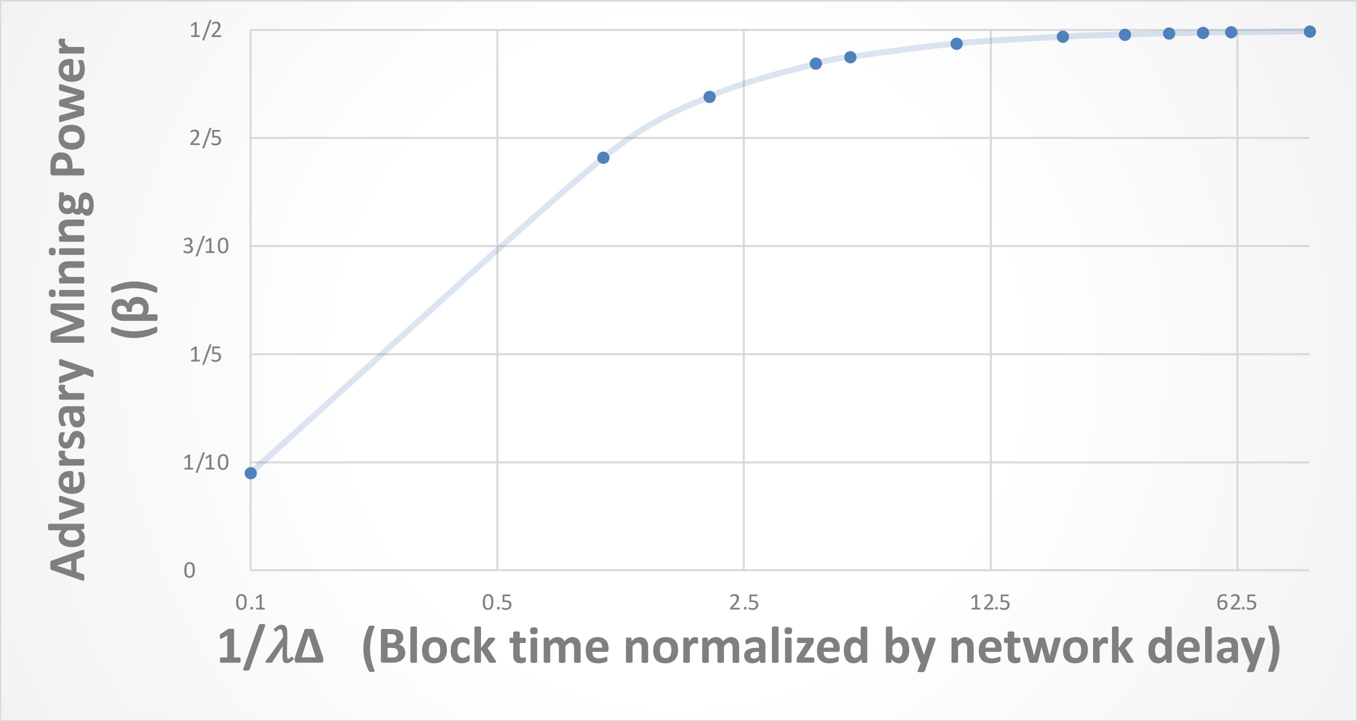
$$\frac{\alpha}{1 + \alpha\lambda\Delta} > \beta$$

Private chain attack = Actual security (was an open question)

<u>Interpretation:</u>

The less Δ relative to block time, the closer this gets to $\alpha > \beta$. For large Δ the adversary needs much less than 50% of the mining power to attack

DKT+ Theorem Graph



Nakamoto magically chose $\frac{1}{\lambda\Delta} = 60$ (10 min blocktime assuming 10s network delay)

Blue line = max value of β s.t. $\frac{\alpha}{1+\alpha\lambda\Delta} > \beta$

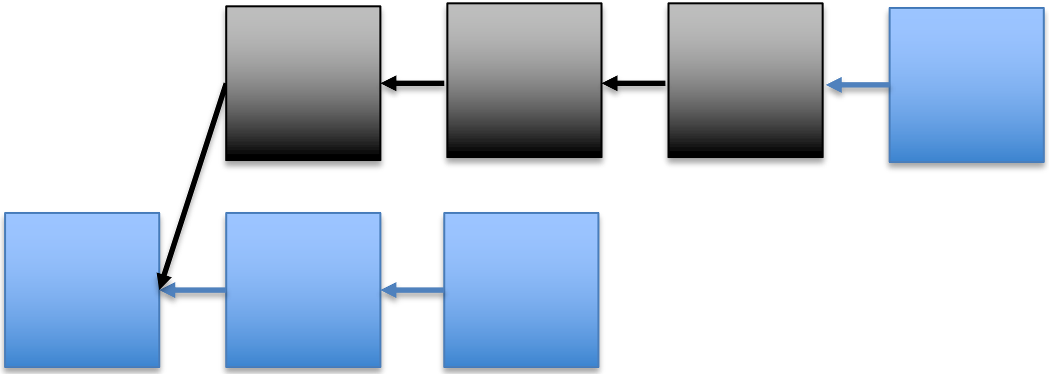
Short Forks and Liveness

Long forks are impossible but short forks may not be

This is a liveness issue

Need to ensure that some “honest” blocks are in the longest chain

Could be used to censor transactions



Nakamoto chain quality

- Chain Quality is percentage of honestly mined blocks
  - Honest mined blocks include all transactions!
  - Prevents censorship
- Say the adversary controls a p fraction of the mining power $p<$ ½
- Ideally honest parties mine a 1 – p fraction
- Can prove they mine at least $1 - \frac{p}{1-p}$ $\quad p = \frac{1}{3} \rightarrow Q = \frac{1}{2}$

If $\mathbf{p > \frac{1}{2}}$ then adversary could mine every block in worst case
$\Rightarrow$ chain quality is 0

Chain Quallity Theorem

- For every p < ½ , if mining difficulty is appropriately set as function of network delay Δ then Nakamoto consensus guarantees:
    1. Consistency (for α, β, Δ satisfying formula)
    2. Chain quality: $1 - \frac{p}{1-p}$ fraction blocks honest
    3. O(1/Δ)-Chain growth

Nakamoto Conensus and Partial Synchrony

- Nakamoto Consensus can be secure up to ½ corruptions
- Can tolerate network delays
- Contradicts partial synchrony lower bound?
  - No
  - Protocol needs a bound on delays (c)
  - Consistency broken even with honest nodes

Nakamoto Properties

- Anonymous participation
- Nodes can join/leave
  - Very scalable
  - Sleeping Beauty property
- Leader not known beforehand
  - Makes bribing harder
- Up to ½ corruptions

- Slow
  - Even when everyone is honest
- Resource intensive
  - PoS based possible
- No finality
- No guarantees under long delays

Incentives

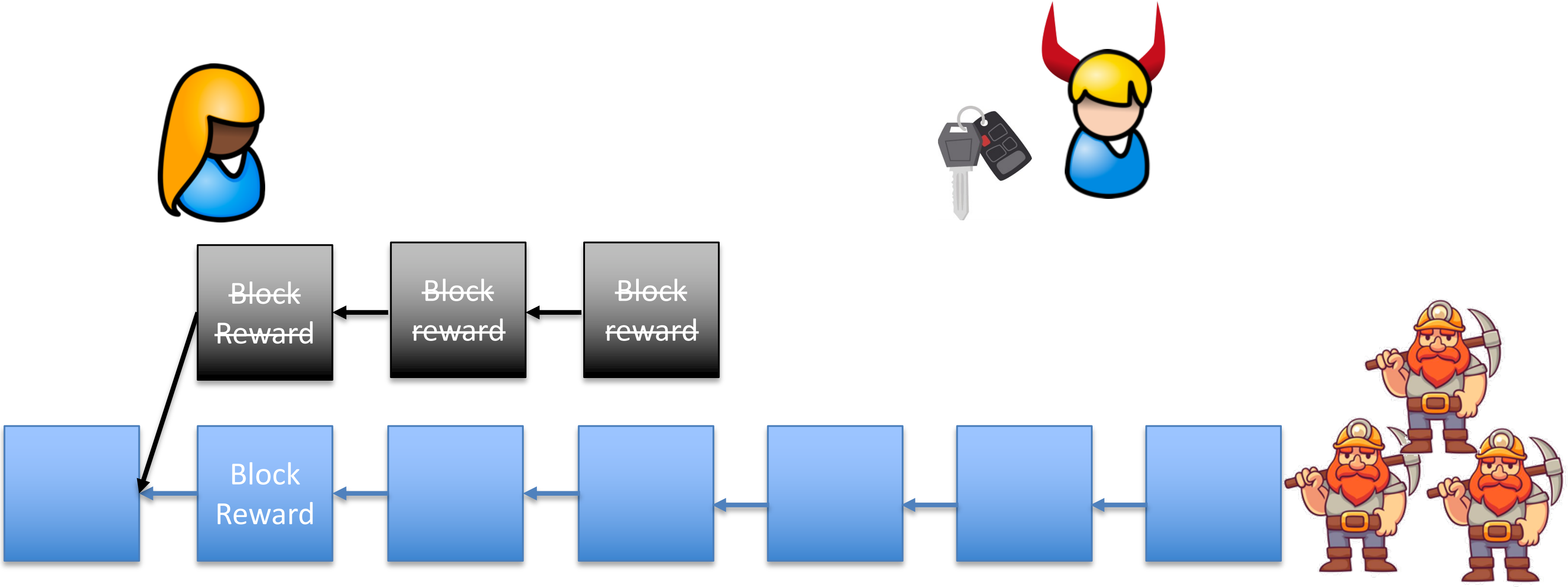
- Mining (solving PoW puzzles) is very expensive
- *Honest* majority does not seem realistic
- Satoshi’s genius idea: Combine issuance and rewards

- Block reward only paid if block part of longest chain
- High Variance -> Mining Pools

Block Reward 💸

Incentives

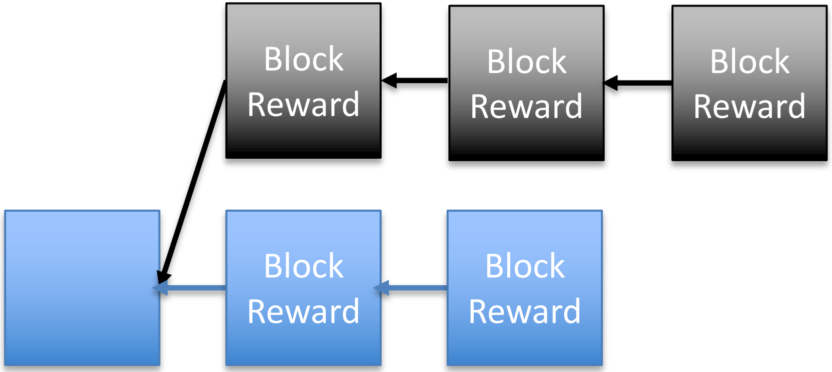
Large opportunity cost for unsuccessful attacks



Selfish mining attack

Attacker has 1/3 of mining power. Miner is rational (maximize rewards)

Keeps block private

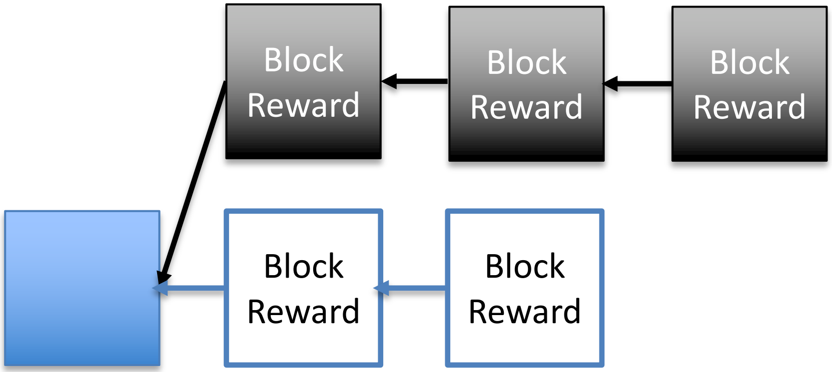


Once attacker has a two block lead he can mine until honest chains catch up

Selfish mining attack

Attacker has 1/3 of mining power. Miner is rational (maximize rewards)

Keeps block private



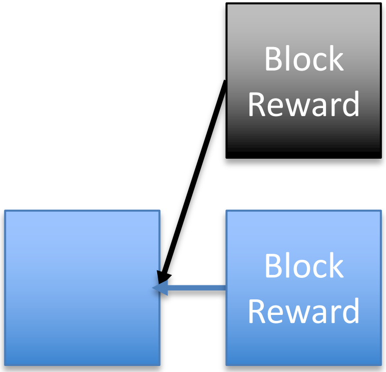
Once attacker has a two block lead he can mine until honest chains catch up

Attacker publishes chain and invalidates honest blocks

Selfish mining attack

Attacker has 1/3 of mining power. Miner is rational (maximize rewards)

Keeps block private



If honest miners finds block: Publish and it's a block race (Attacker has at least 1/3 p of winning)

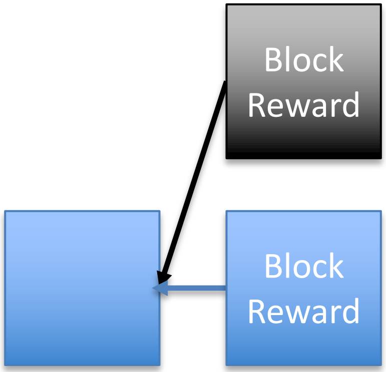
Selfish mining analysis

Honest reward=1/3

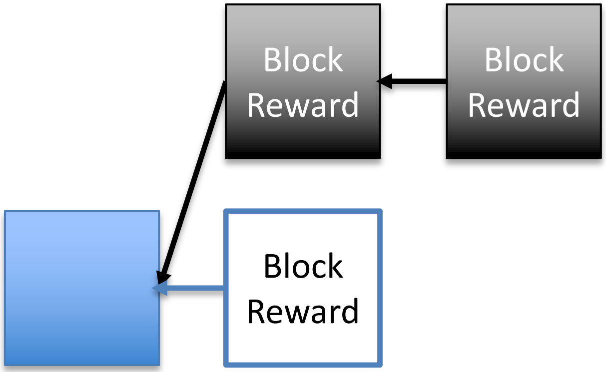
$$\frac{2}{3} * \frac{1}{3} * \frac{2}{3} + \frac{1}{3} * \frac{2}{3} = \frac{10}{27} > \frac{1}{3}$$

P Block Race:
2/3

P Run away: 1/3

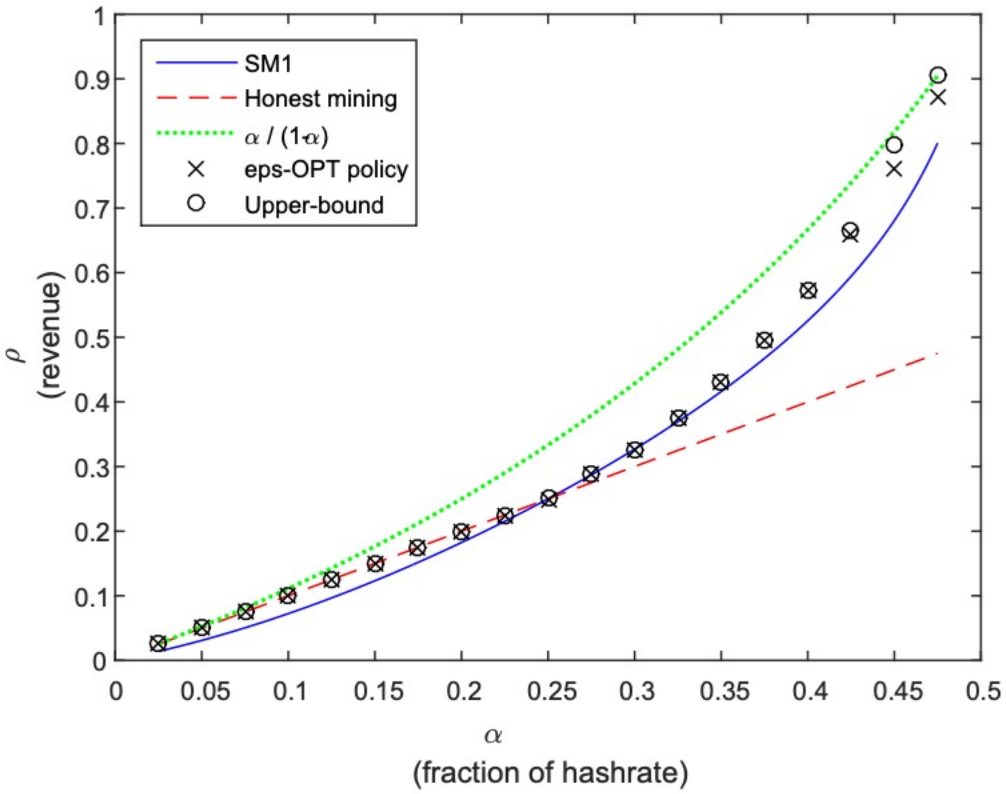


Win: 1/3 chance
2 of 3 blocks
Reward 2/3
Loose: 2/3 chance
Reward 0



Reward > 2/3

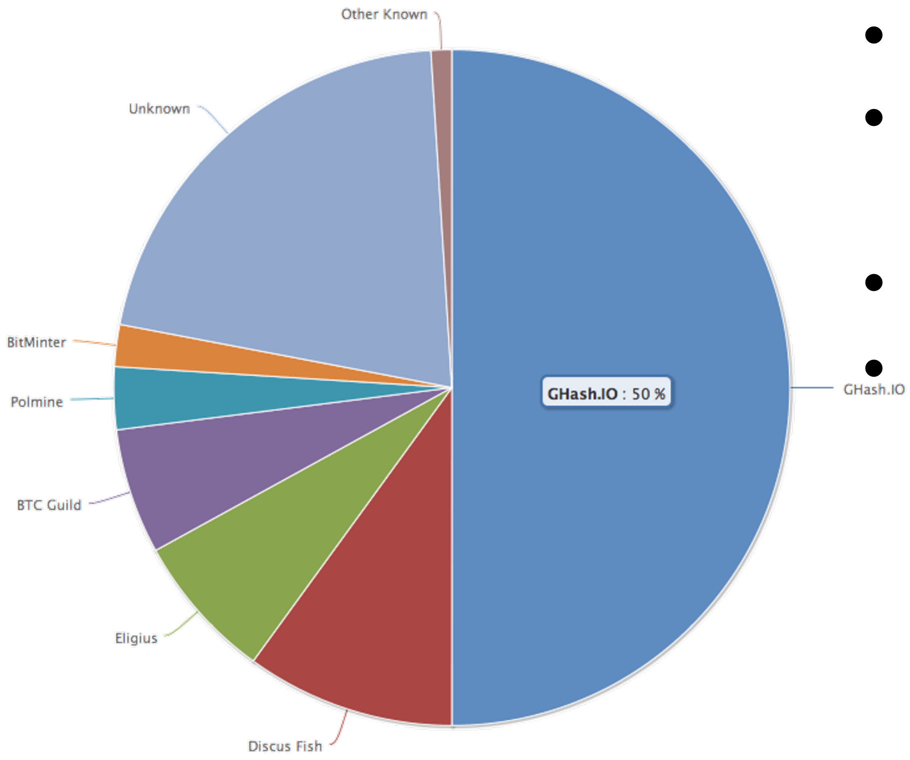
Selfish Mining



Optimal Selfish mining

Explains why chain quality <1-p

No Attacks in Practice?



- Attacks possible but not seen
- Ghash.IO had >50%
  - Gave up mining power
- No Selfish mining attacks
- Why?
  - Miners care about Bitcoin price
  - Not rational in $ terms to attack
  - Not guaranteed in the future

END OF LECTURE

Next lecture:

Randomness beacons, VDFs, large scale PoS