CS251 Fall 2022

(cs251.stanford.edu)

Consensus in the Internet Setting

Ertem Nusret Tas

Recap of the Last Lecture

- Byzantine Generals Problem
- Definition of Byzantine adversary
- Synchronous, asynchronous and partially synchronous networks
- State Machine Replication (SMR)
- Security properties for SMR protocols: Safety and Liveness
- A secure SMR protocol: Streamlet

Sybil Attack

How to select the nodes that participate in consensus?

Two variants:

- *Permissioned:* There is a *fixed* set of nodes (previous lecture).
- *Permissionless*: Anyone satisfying certain criteria can participate.

Can we accept any node that has a signing key to participate in consensus?

Sybil Attack!

Sybil Resistance

Consensus protocols with Sybil resistance are typically based on a bounded (scarce) resource:

	Resource dedicated to the protocol	Some Example Blockchains
Proof-of-Work	Total computational power	Bitcoin, PoW Ethereum…
Proof-of-Stake	Total number of coins	Algorand, Cardano, Cosmos, PoS Ethereum…
Proof-of-Space/Time	Total storage across time	Chia, Filecoin…

How does Proof-of-Work prevent Sybil attacks?

We assume that the adversary controls a small fraction of the scarce resource!

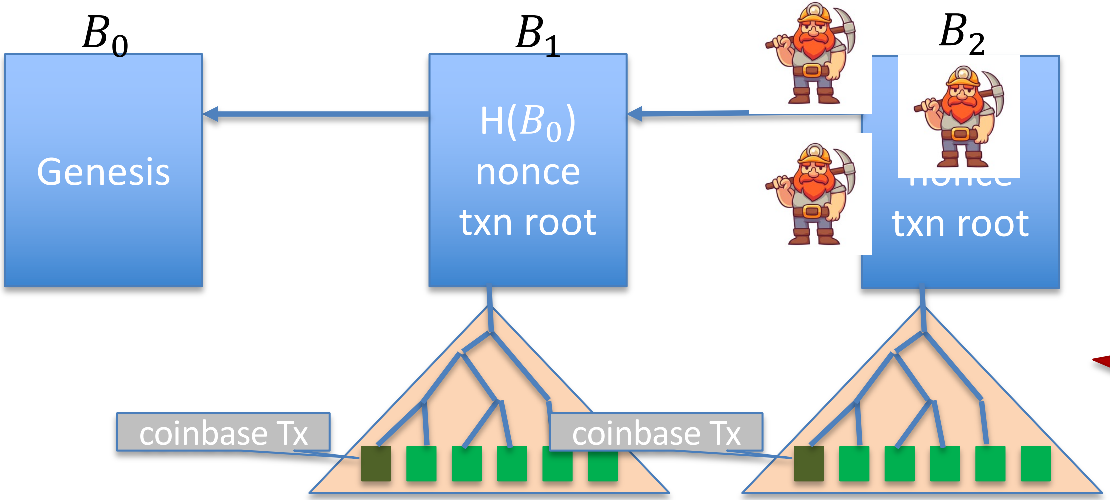
Bitcoin: Mining

Difficulty: How many nonces on average miners try until finding a block?

To mine a new block, a miner must find *nonce* such that

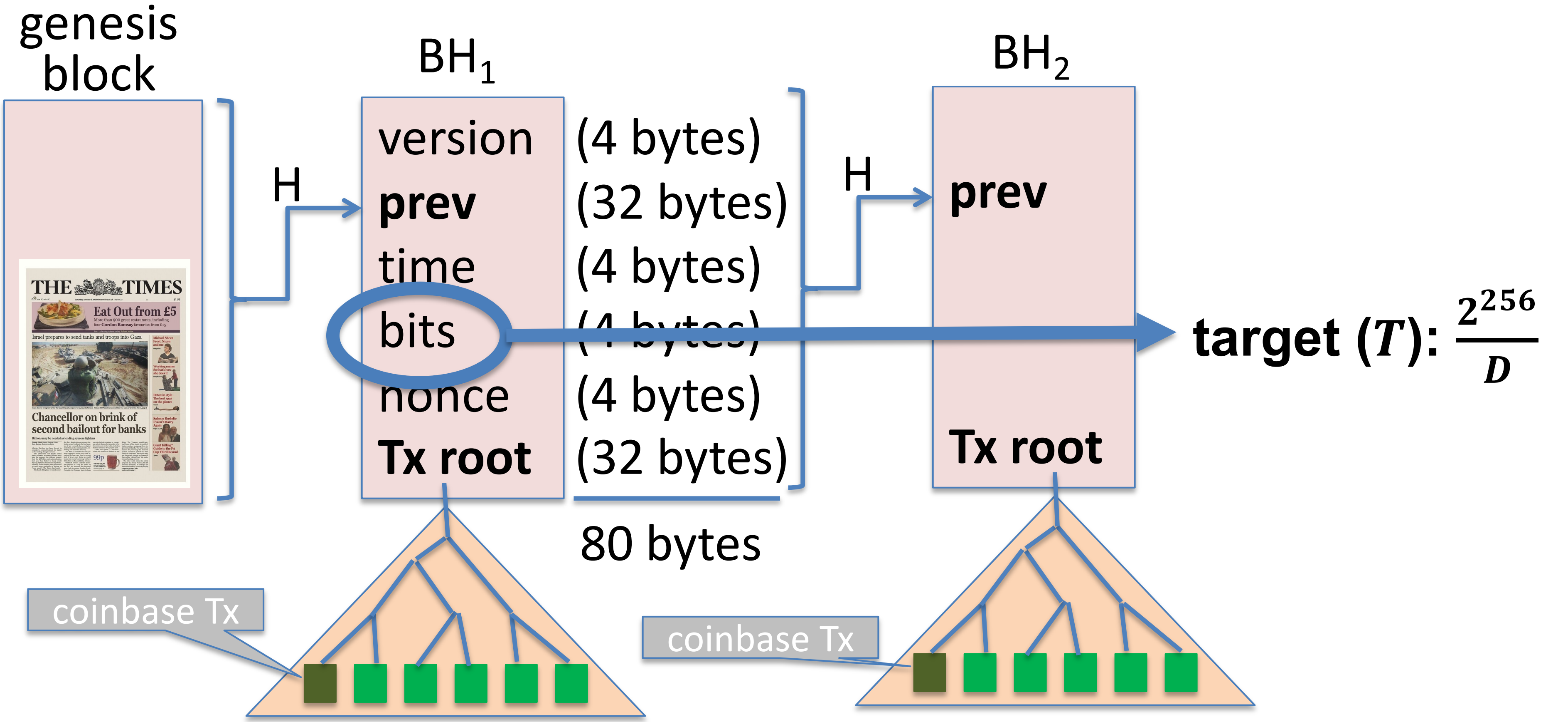
$$H(h_{prev}, txn\ root, nonce) < \text{Target} = \frac{2^{256}}{D}$$

Each miner tries different nonces until one of them finds a nonce that satisfies the above equation.



New block: random process but approximately once in every 10 minutes

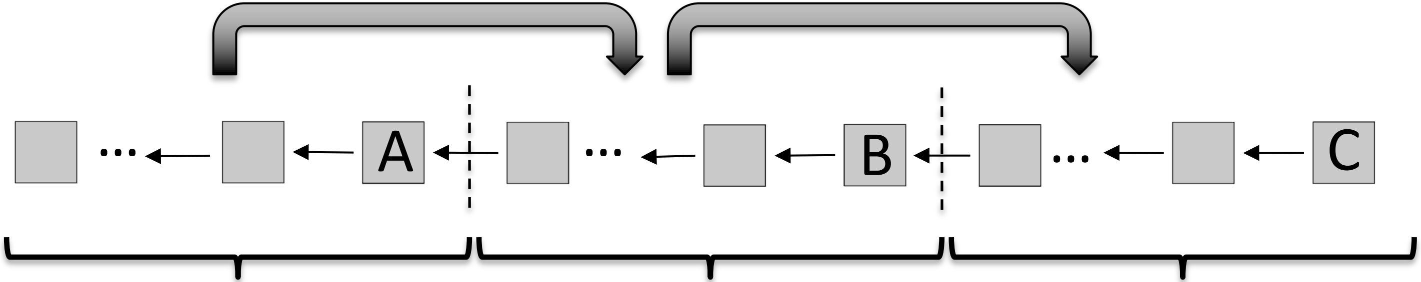
Bitcoin: Block Headers



Bitcoin: Difficulty Adjustment

New target: $T_2 = T_1 \frac{t_1}{2016 \times 10\ mins}$

New target: $T_3 = T_2 \frac{t_2}{2016 \times 10\ mins}$



2016 blocks
Time it took to mine: t_1(min)
Target: T_1

2016 blocks
Time it took to mine: t_2(min)
Target: T_2

2016 blocks
Time it took to mine: t_3(min)
Target: T_3

New target is not allowed to be more than 4x old target.

New target is not allowed to be less than ¼ x old target.

The Bitcoin Backbone Protocol with Chains of Variable Difficulty (2016)

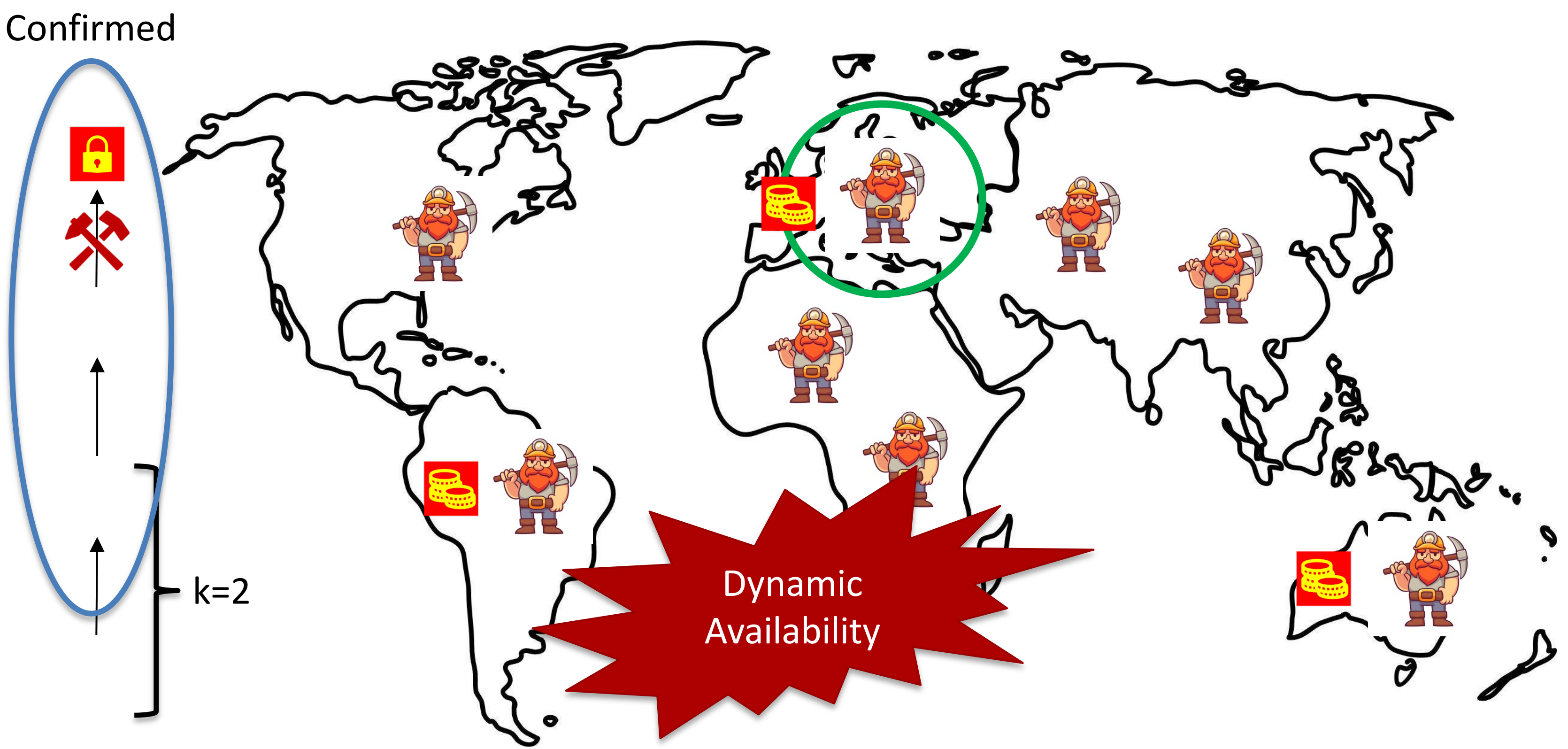
Nakamoto Consensus

Chain with the highest difficulty!

Bitcoin uses **Nakamoto consensus**:

- **Fork-choice / proposal rule:** At any given time, each honest miner attempts to extend (i.e., mines on the tip of) the heaviest (longest for us) chain in its view (Ties broken adversarially).
- **Confirmation rule:** Each miner confirms the block (along with its prefix) that is *k*-deep within the longest chain in its view.
  - In practice, $k = 6$.
  - Miners and clients accept the transactions in the latest confirmed block and its prefix as their log.
  - Note that *confirmation* is different from *finalization*.
- **Leader selection rule:** Proof-of-Work.

Nakamoto Consensus



Bitcoin vs. Streamlet

	Bitcoin	Streamlet
Fork-choice rule	Heaviest (Longest in our case) Chain	Longest Notarized Chain
Confirmation/finalization rule	*k*-deep prefix of the longest (heaviest) chain	Three adjacent blocks in a notarized chain from consecutive epochs
Leader selection rule	Determined by the difficulty *D*	With the help of a hash function

- Streamlet is not dynamically available: It loses liveness if n/3 or more nodes go offline!
- Bitcoin is dynamically available: It continues to confirm transactions even if the majority of the mining power goes offline.

Consensus in the Internet Setting

Characterized by *open participation*:

- Adversary can create many Sybil nodes to take over the protocol.
- Honest participants can come and go at will.

Goals:

- Limit adversary's participation.
  - **Sybil resistance (e.g., Proof-of-Work)!**
- Maintain availability (liveness) of the protocol against changing participation by the honest nodes.
  - **Dynamic availability!**

Security

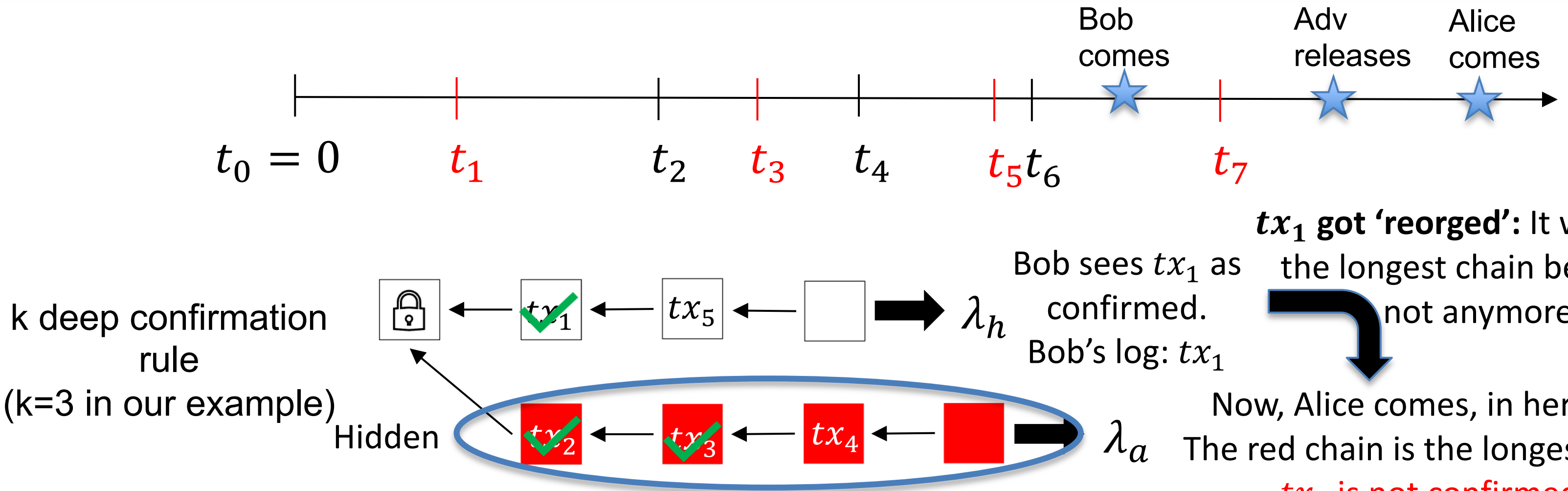
Can we show that Bitcoin is secure under synchrony against a Byzantine adversary?

What would be the best possible resilience?

$$\beta < 1/2?$$

Fraction of the mining power controlled by the adversary.

Nakamoto's Private Attack: $\beta \geq 1/2$



tx_1 got 'reorged': It was part of the longest chain before but not anymore!!

Now, Alice comes, in her view:
The red chain is the longest chain.
tx_1 is not confirmed!
Alice's log: $tx_2 tx_3$

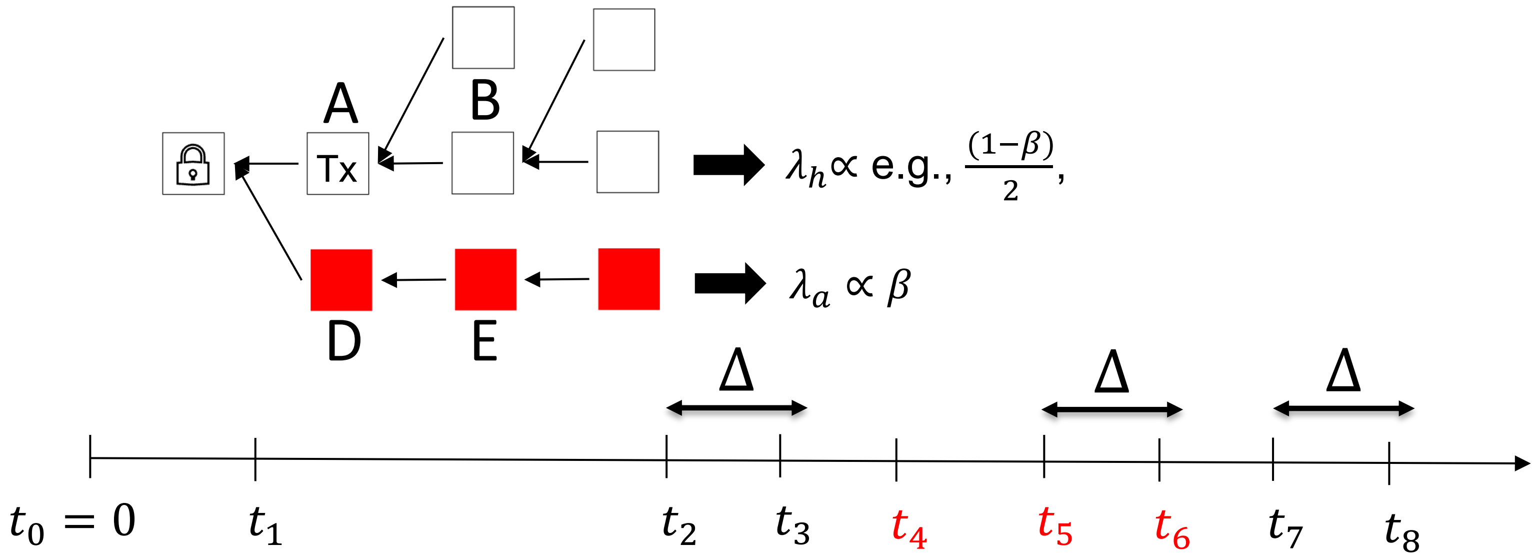
Private attack (mostly) succeeds if $\lambda_a \geq \lambda_h$, i.e., if $\beta \geq 1 - \beta$, i.e., if $\beta \geq \frac{1}{2}$.

Private attack (mostly) fails if $\lambda_a < \lambda_h$, i.e., if $\beta < 1 - \beta$, i.e., if $\beta < \frac{1}{2}$.

Can another attack succeed?

A Peer-to-Peer Electronic Cash System (2008)

Forking



Multiple honest blocks at the same height due to network delay.

Adversary's chain grows at rate proportional to (shown by $\propto$) β!

Honest miners' chain grows at rate less than $1 - \beta$ because of forking!

Now, adversary succeeds if $\beta \geq \frac{(1-\beta)}{2}$, which implies $\beta \geq \frac{1}{3}$!!

Security

Theorem: If $\beta < 1/2$, there exists a small enough mining rate $\lambda(\Delta, \beta) = \lambda_a + \lambda_h$ (by changing difficulty) such that Bitcoin satisfies security (safety and liveness) except with error probability $e^{-\Omega(k)}$ under synchronous network.

- This is the error probability for confirmation.
- We say 'confirmation' instead of finalization because when you *confirm* a block or transaction, you *confirm* it with an error probability…
- …unlike *finalizing* a block where there is no error probability*.

Now, we see why Bitcoin has 1 block every 10 minutes, instead of 1 block every second…

The Bitcoin Backbone Protocol: Analysis and Applications (2015)
Analysis of the Blockchain Protocol in Asynchronous Networks (2016)
Analysis of Nakamoto Consensus (2019)
Everything is a Race and Nakamoto Always Wins (2022)

Is Bitcoin the Endgame?

- Bitcoin provides Sybil resistance and dynamic availability.
- It can be made secure for any $\beta < \frac{1}{2}$.
- Is it the Endgame for consensus?

No!

- Bitcoin is secure only under synchrony unlike Streamlet that is secure under partial synchrony.
- It *confirms* blocks with an error probability as a function of *k*, unlike Streamlet that *finalizes* blocks.
- Energy?

Dark Side of Bitcoin: Energy

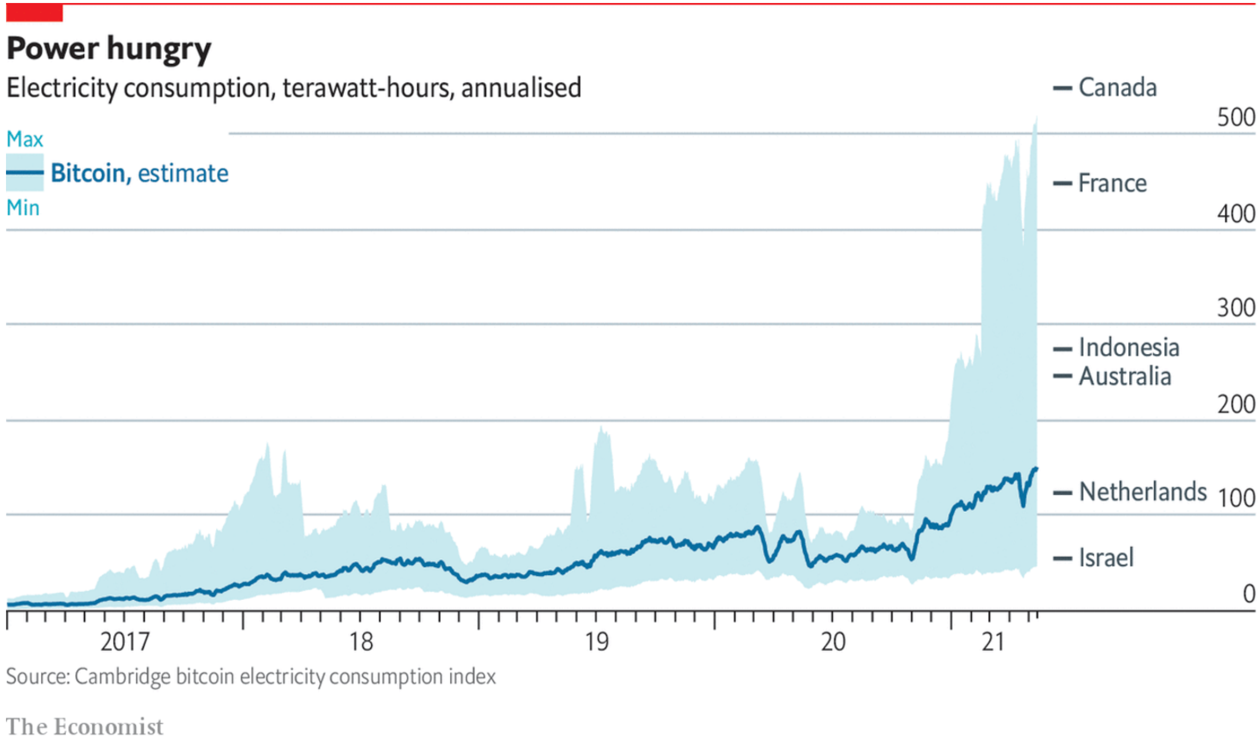
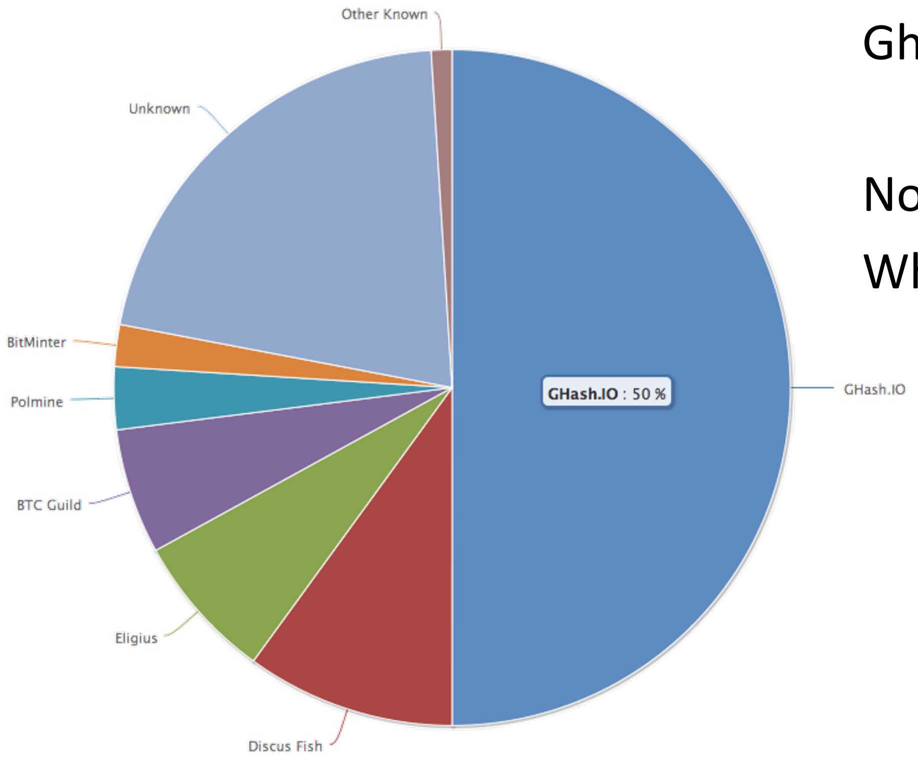


Photo taken from the article “As the price of bitcoin has climbed, so has its environmental cost” that appeared at The Economist on May 14th 2021.

No Attacks on Bitcoin?



Ghash.IO had >50% in 2014

- Gave up mining power

No Selfish mining attacks?

Why are visible attacks not more frequent?

- Miners care about the Bitcoin price.
- Might not be rational to attack.
- No guarantees for the future.

END OF LECTURE

Next lecture: Incentives and Accountability in Consensus

Optional Slides

Slides going forward is optional material and present a simplified security proof for Nakamoto consensus.

Reminder for Security (Optional)

Let's recall the definition of security for SMR protocols. Let ch_t^i denote the confirmed (i.e., k-deep) chain accepted by a client i at time t.

Safety (Consistency):

- For any two clients i and j, and times t and s: $\boldsymbol{ch_t^i} \preccurlyeq \boldsymbol{ch_s^j}$ (prefix of) or vice versa, i.e., chains are consistent.

No double spend

Liveness:

- If a transaction tx is input to an honest replica at some time t, then for all clients i, and times $s \geq t + T_{conf}$: $tx \in \boldsymbol{ch_s^i}$.

No censorship

Modelling Bitcoin (Optional)

Many different miners, each with *infinitesimal* power.

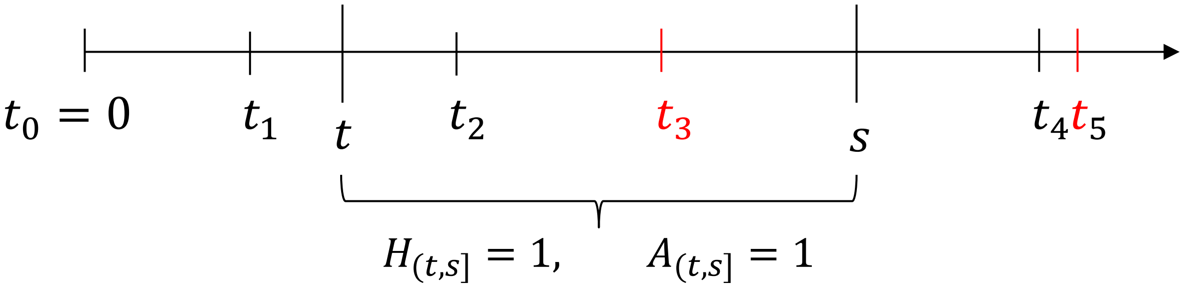
Total mining rate: λ (1/minutes). In Bitcoin, $\lambda = 1/10$.

Adversary is Byzantine and controls $\beta < \frac{1}{2}$ fraction of the mining power.

- Adversarial mining rate: $\lambda_a = \beta\lambda$
- Honest mining rate: $\lambda_h = (1 - \beta)\lambda$

Each mined block is adversarial with probability β *independent* of other blocks.

Network is synchronous with a known upper bound Δ on delay.



Security Proof: Safety (Optional)

Suppose there is at most one honest block at every height.

This is the case *if* the network delay $\Delta = 0$.

Private attack
is the best
attack!!

GR (2022): If **any** attack succeeds in violating a target transaction tx's safety, then the **private attack with premining** also succeeds in violating the target transaction's safety.

Bitcoin's Latency Security Analysis Made Simple (2022)

Security Proof: Safety (Optional)

We will show that if **any** attack succeeds in violating safety of a target transaction tx within the first honestly mined block, then the **private attack** also succeeds in violating the target transaction's safety.

For the full proof, see "Bitcoin's Latency Security Analysis Made Simple".

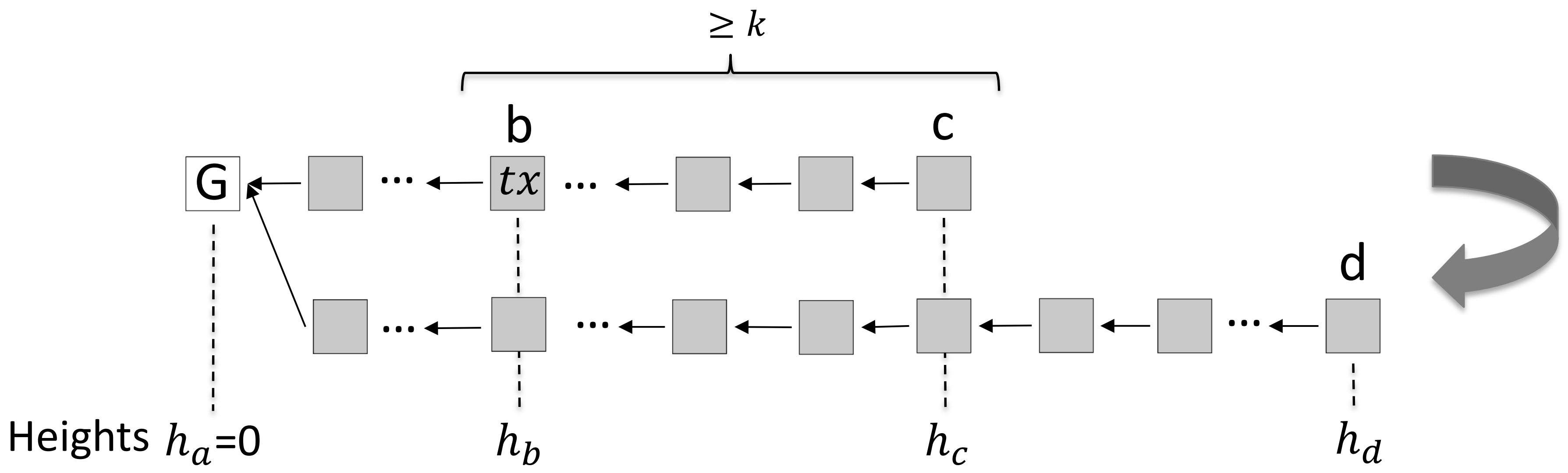
Security Proof: Safety (Optional)

Suppose a transaction tx is confirmed within the first block b mined by the honest miners in an honest view.

Let's observe a 'reorg' of block b by *some* attack.

We will show that the private attack will also succeed in 'reorging' b!

Security Proof: Safety (Optional)

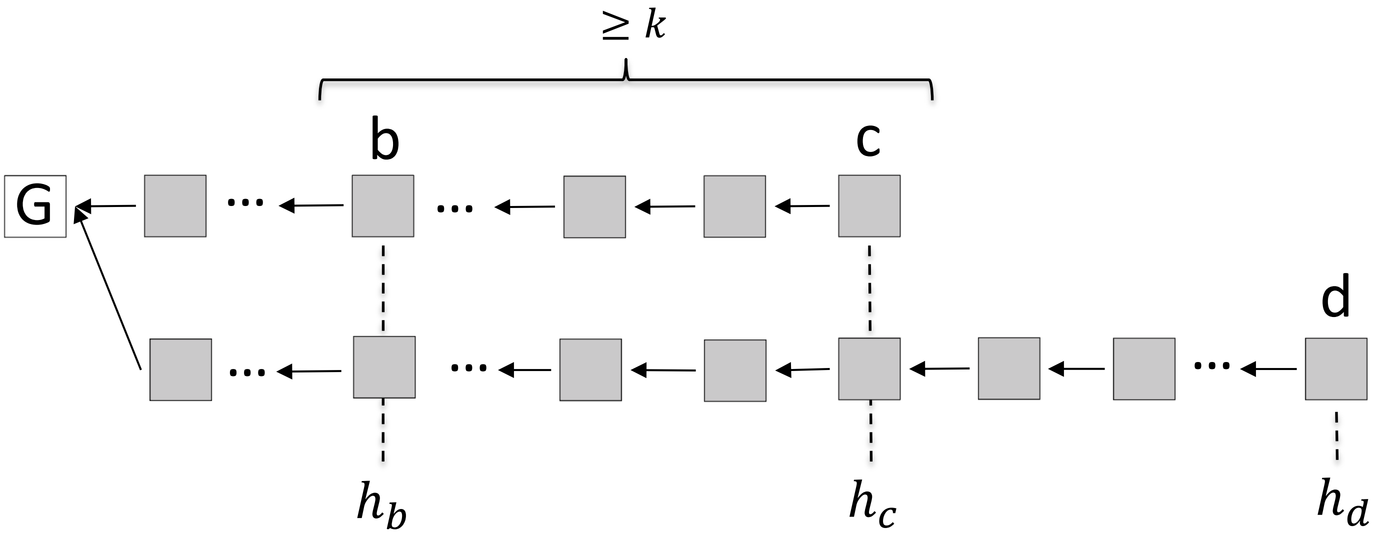


Block *b* contains the transaction tx that is 'reorged'.

Consider the *first time* that *t* block *b* is reorged.

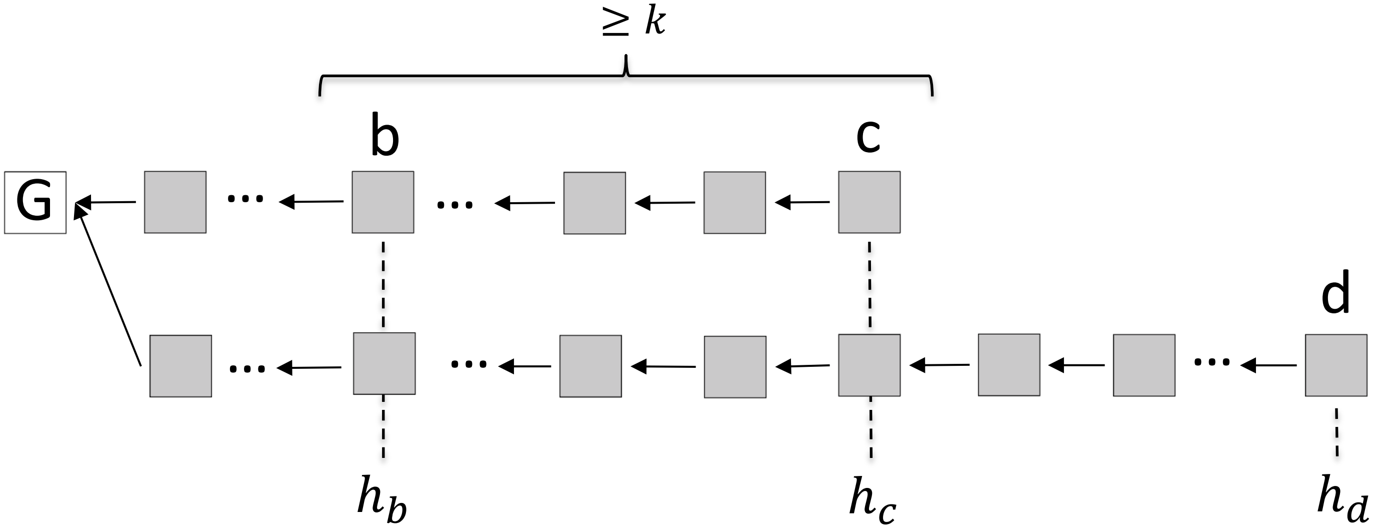
- Right before *t*, block *c* is seen at the tip of the longest chain by an honest node.
- Right after *t*, block *d* is seen at the tip of the longest chain by another (potentially the same) honest node.

Security Proof: Safety (Optional)



- **Fact 1:** At each height until h_c, there is at least one adversary block.
  - Why?
  - Because there can be at most one honest block at any height.

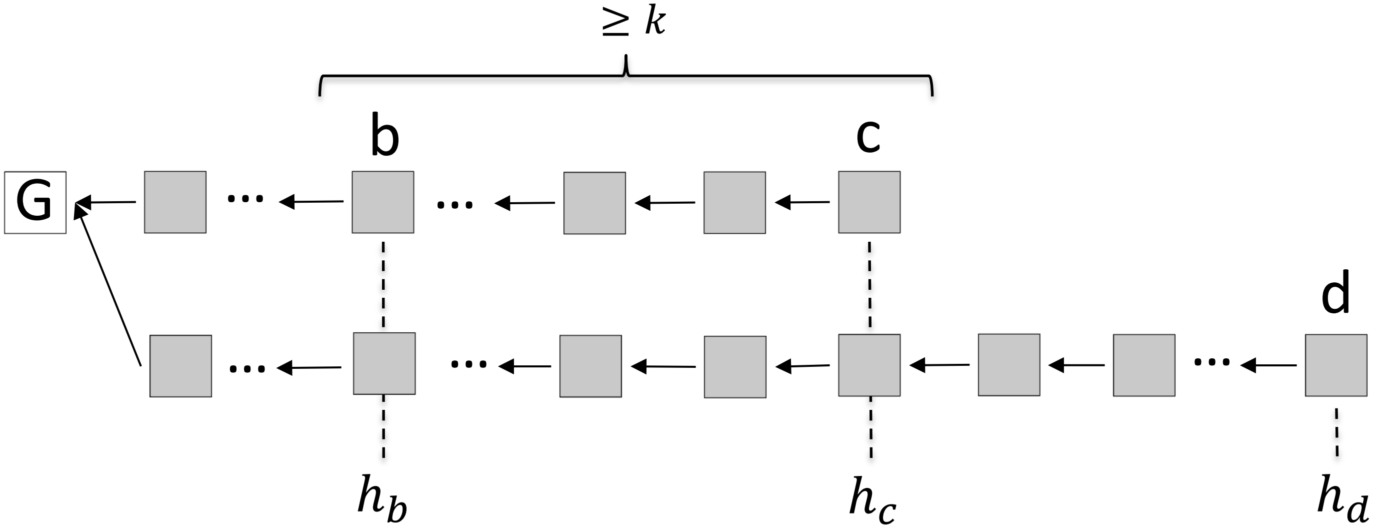
Security Proof: Safety (Optional)



- **Fact 2:** Every block after h_c thru h_d are adversarial (one block per height).
  - Why?
  - Otherwise, we contradict with the definition of blocks *c* and *d*.

$$A \geq h_d$$

Security Proof: Safety (Optional)



- **Fact 3:** There are at most h_c honest blocks.

$$H \leq h_c$$

Security Proof: Safety (Optional)

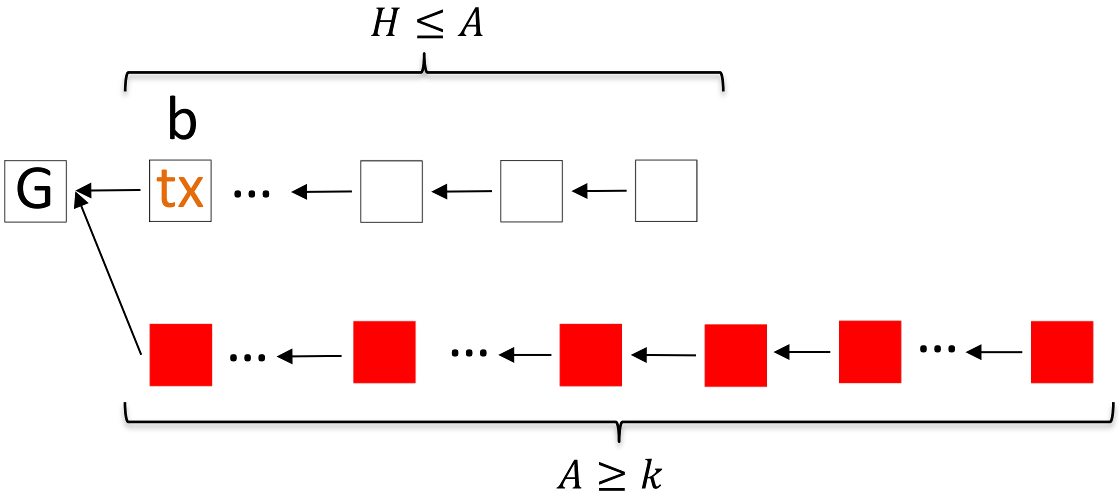
- Combining everything…
  - $H \leq h_c$
  - $A \geq h_d$
  - $h_d \geq h_c \geq k$
- This implies $A \geq H$ and $A \geq k$.

Private attack also succeeds!

Why?

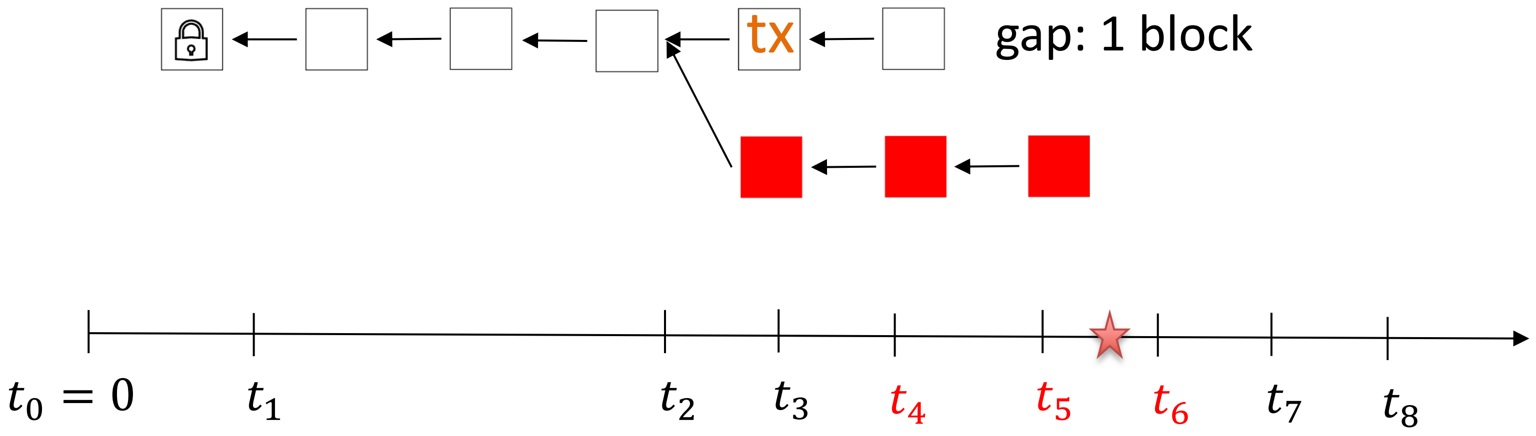
Security Proof: Safety (Optional)

$A \geq H$ and $A \geq k$:



Private attack also succeeds!

Security Proof: Safety (Optional)



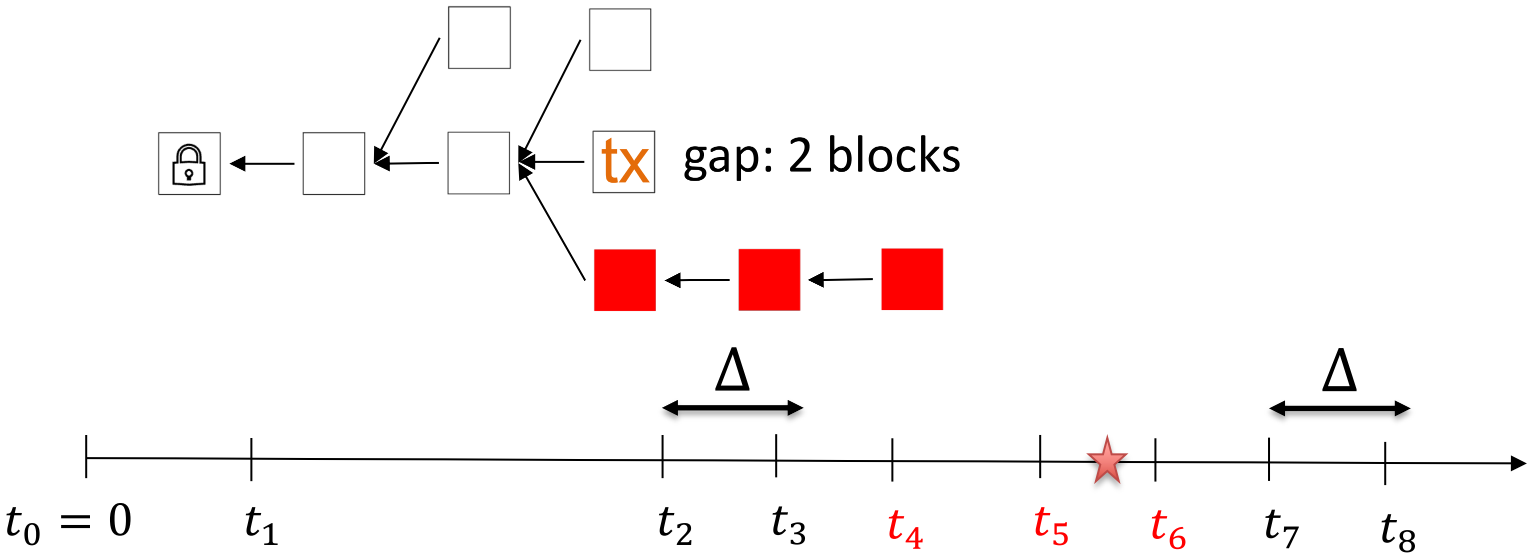
If every honest block is at a separate height…

Best attack to reorg a transaction is the **private attack with premining**!

Probability that a private attack with premining succeeds $\leq e^{-\Omega(k)}$; if $\lambda_a < \lambda_h$, i.e., $\beta < 1/2$!

Safety!

Security Proof: Safety (Optional)

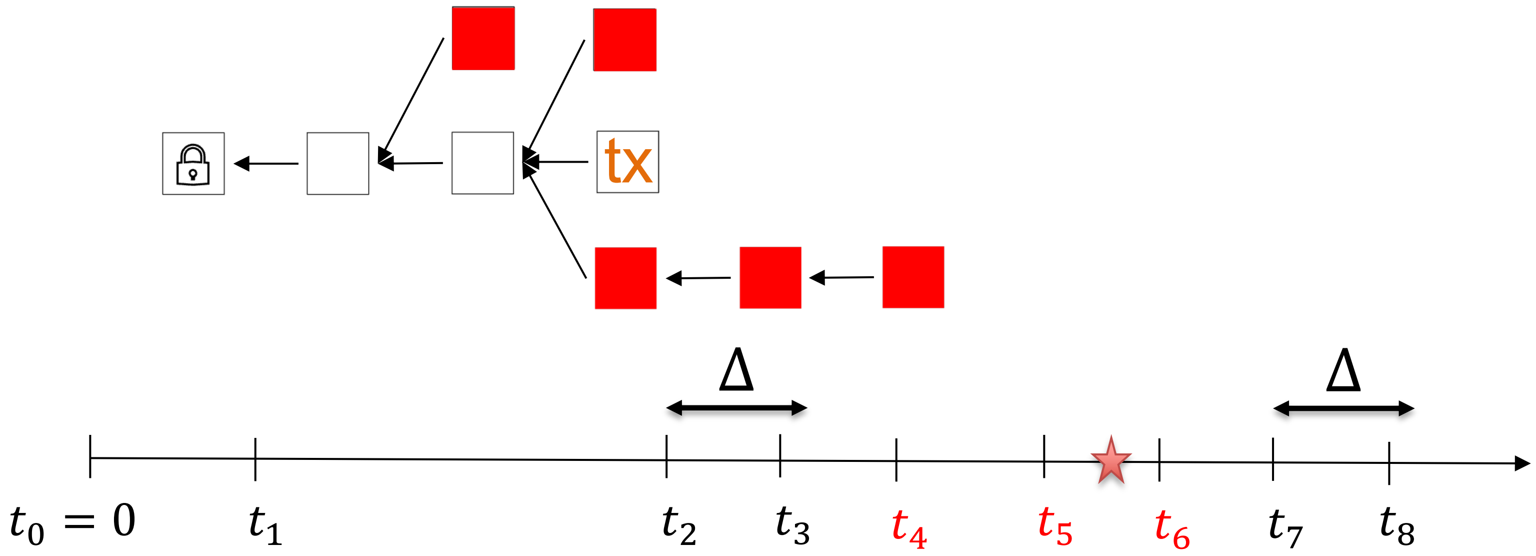


Multiple honest blocks at the same height due to network delay.

Forking!

Probability that a block is an honest block at a unique height: $e^{-\lambda\Delta}(1-\beta)$

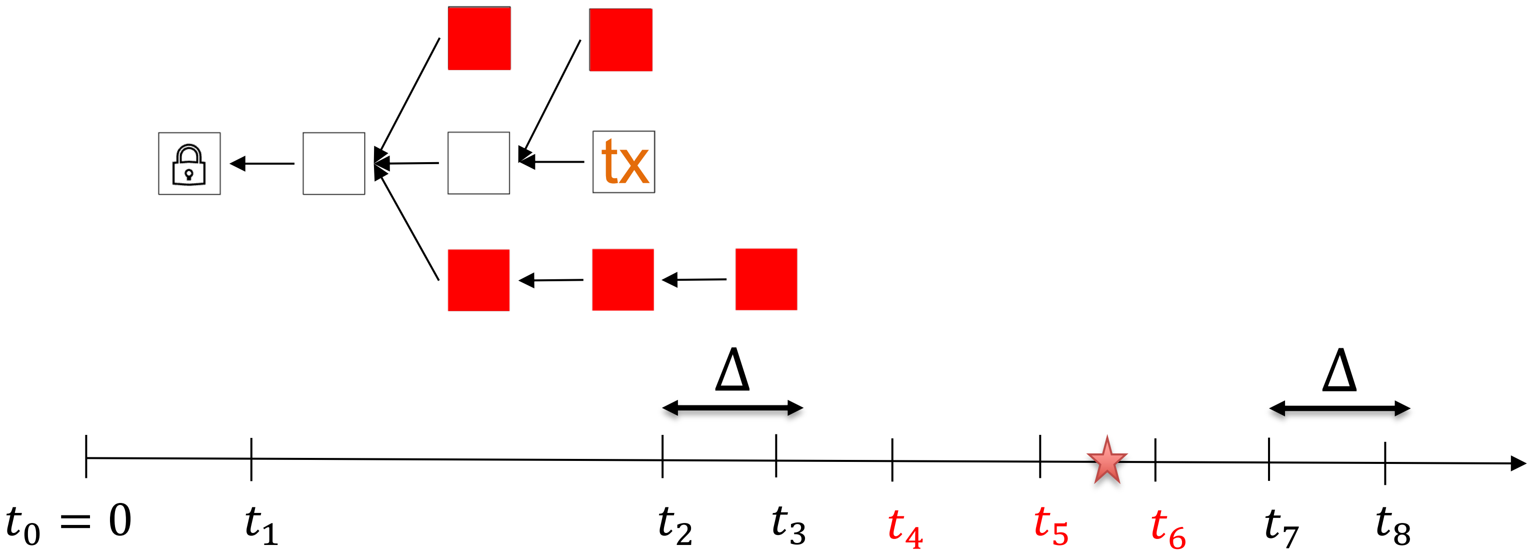
Security Proof: Safety (Optional)



Trick: We give honest blocks that fall into the same height as previous honest blocks to the adversary.

Mining rate of 'honest' blocks with new definition = $e^{-\lambda\Delta}(1-\beta)$

Security Proof: Safety (Optional)



Every honest block is *again* at a separate height!

Best attack to reorg a transaction is the private attack with premining.

Probability that a private attack with premining succeeds $\leq e^{-\Omega(k)}$; if $\frac{1}{2} < e^{-\lambda\Delta}(1-\beta)$.

Safety!

Security Proof: Liveness (Optional)

Growth rate of the blockchain $\geq e^{-\lambda\Delta}(1-\beta)\lambda$.

Arrival rate of adversary blocks: $\beta\lambda$

If $\frac{1}{2} < e^{-\lambda\Delta}(1-\beta)$, then $e^{-\lambda\Delta}(1-\beta)\lambda > \beta\lambda$.

Thus, over a sufficiently large time interval (call this u), the *k*-deep prefix of the longest chain in the view of each honest node must contain new honest blocks except with probability $e^{-\Omega(u)}$.

Liveness!