On Bitcoin Cash's Target Recalculation Functions*

Juan Garay
Texas A&M University
garay@tamu.edu

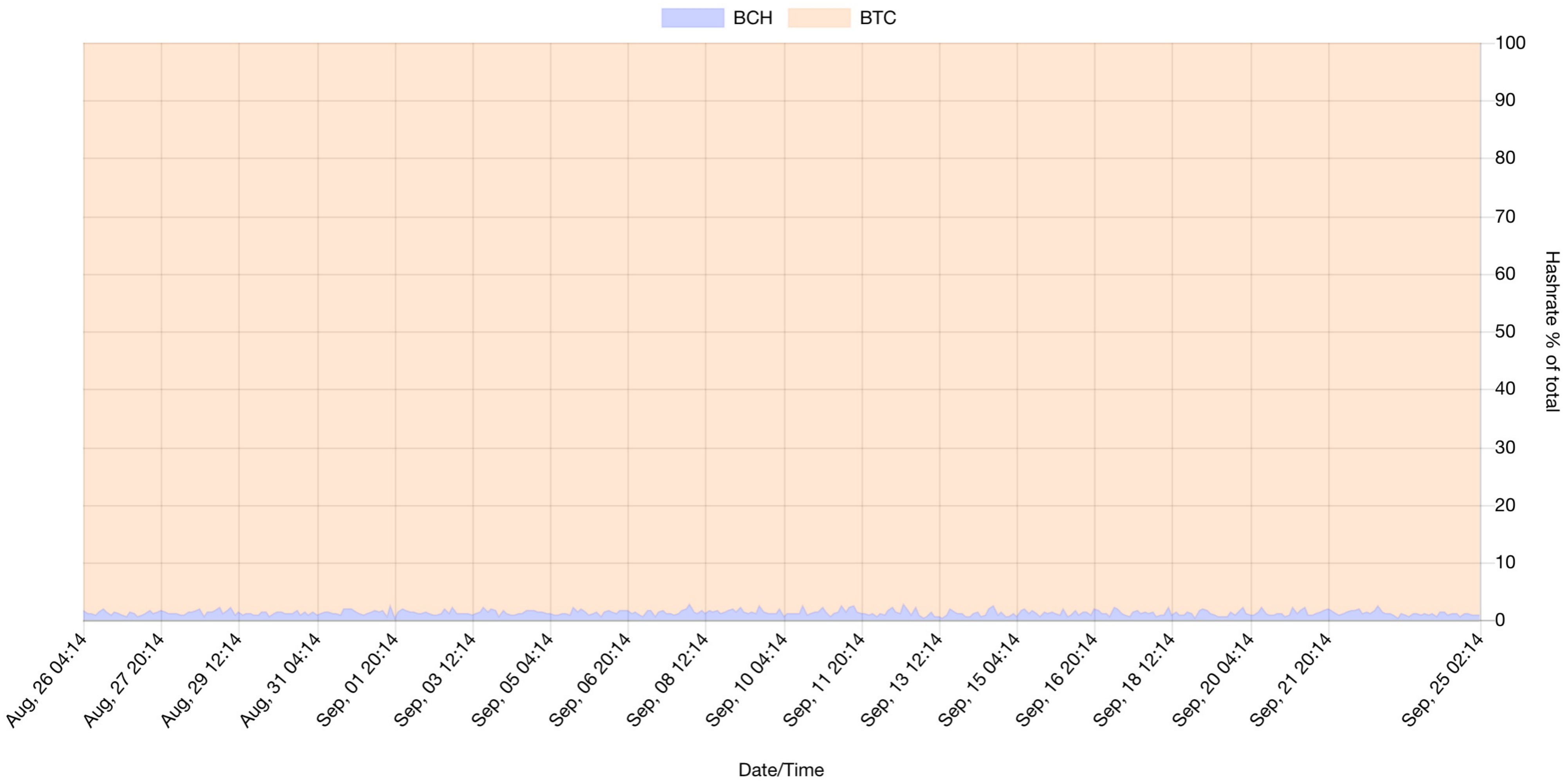
Yu Shen
Texas A&M University
shenyu.tcv@tamu.edu

Full version: https://eprint.iacr.org/2021/143.pdf

Bitcoin Cash

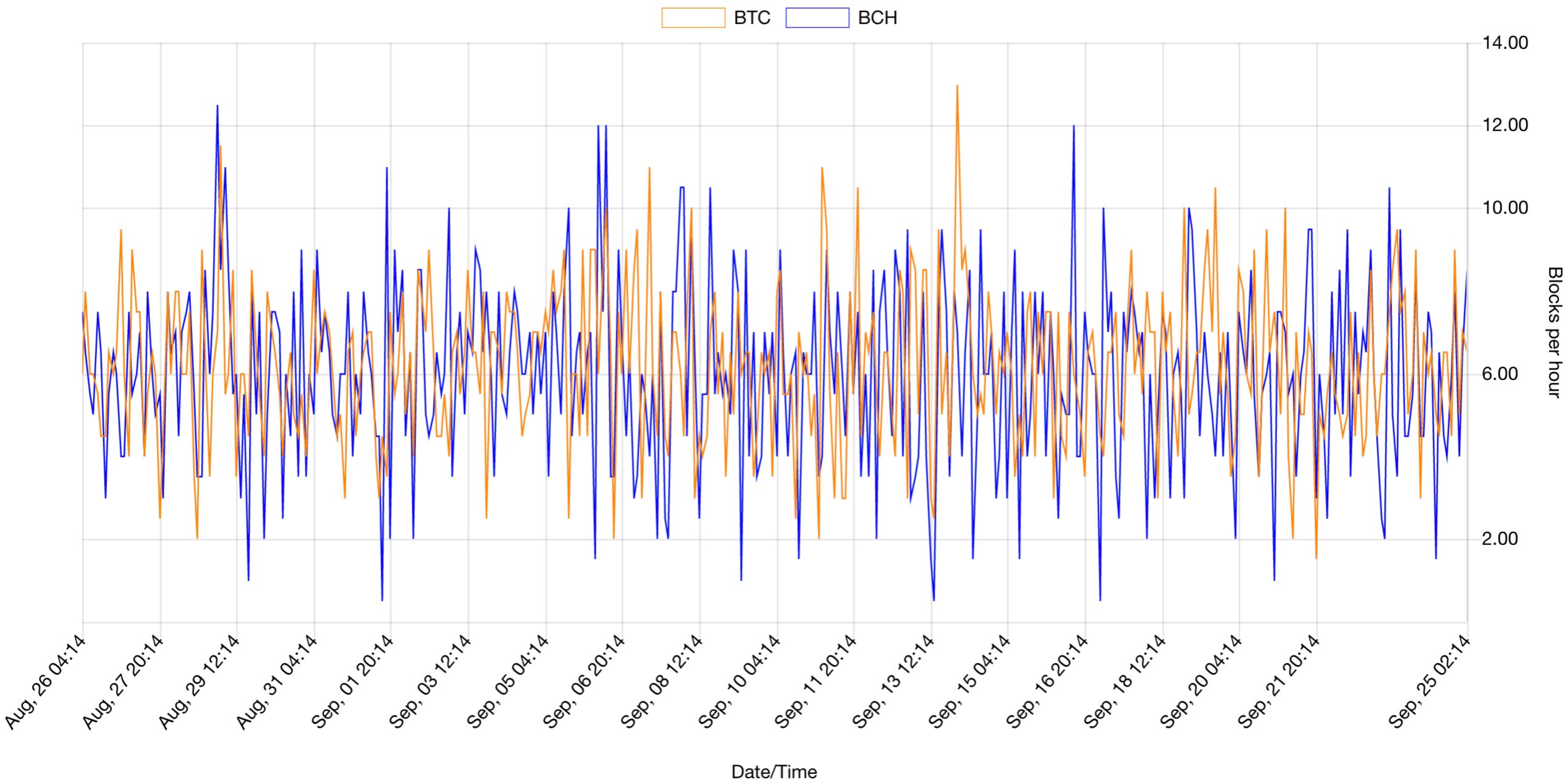
- A “hard fork” of Bitcoin.
- Created on Aug. 1 2017.
- Split ratio 1:1.
- Motivation: accommodate an increasing number of transactions.

Relative Hashrate in Percentage of Total



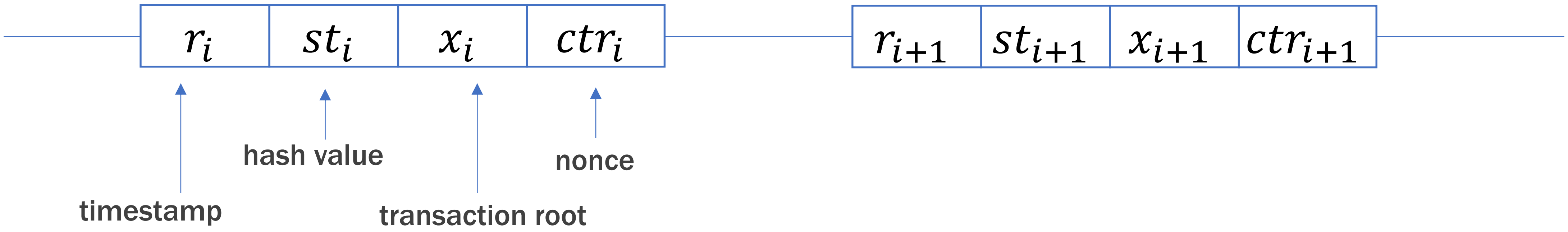
Source: https://fork.lol/pow/hashrate

Average Number of Blocks per Hour



Source: https://fork.lol/blocks/time

Blockchain Data Structure



- A block $\langle r, st, x, ctr \rangle$ is valid if it has a small hash value, providing a proof-of-work:

$$H(r, st, x, ctr) < T.$$

- A chain is valid if all its blocks provide a proof-of-work and each block extends the previous one:

$$\text{For each } i,\ st_{i+1} = H(r, st, x, ctr) \text{ and } r_{i+1} > r_i.$$

Bitcoin Cash’s Target Recalculation Function

EMA — Aug 1 2017 – Nov 13 2017

SMA — Nov 13 2017 – Nov 15 2020

ASERT — Nov 15 2020 – present

- *Emergency Difficulty Adjustment* (EDA):
  - Bitcoin’s Difficulty Adjustment Algorithm + decreasing the mining difficulty by 20%, if the time difference between 6 successive blocks was greater than 12 hours.
- *Simple Moving Average* (SMA):
  - Adjusts the mining difficulty after each block; a moving window of the last 144 blocks.
- *Absolutely Scheduled Exponentially Rising Targets* (ASERT):
  - Adjusts the mining difficulty after each block based on “anchor block”, block height and timestamp.

Bitcoin's Target Recalculation Function

- The target is recalculated every m blocks.
  - Bitcoin uses $m = 2016$ (approximately two weeks) and calls the period between two recalculation points an *epoch*.
  - If one want to extend the chain of length λm, first determines target T by the last m blocks.
- Informally, if the m blocks were calculated quickly, then increase difficulty (decrease T), otherwise decrease difficulty (increase T).
- Suppose the last m blocks were computed in Δ rounds for target T. If we want to have m blocks in every m/f rounds, set

$$T' = \frac{\Delta}{m/f} \cdot T \ (f = \text{block production rate}).$$

- Bahack's difficulty raising attack:
  - The adversary builds the next epoch all by himself with fake timestamps, resulting in huge difficulty for then next epoch.
  - Works with constant probability.

Bitcoin Cash's Target Recalculation Function

EMA → SMA → ASERT

Aug 1 2017 – Nov 13 2017 | Nov 13 2017 – Nov 15 2020 | Nov 15 2020 – present

- *Emergency Difficulty Adjustment* (EDA):
  - Bitcoin's Difficulty Adjustment Algorithm + decreasing the mining difficulty by 20%, if the time difference between 6 successive blocks was greater than 12 hours.
- *Simple Moving Average* (SMA):
  - Adjusts the mining difficulty after each block; a moving window of the last 144 blocks.
- *Absolutely Scheduled Exponentially Rising Targets* (ASERT):
  - Adjusts the mining difficulty after each block based on "anchor block", block height and timestamp.

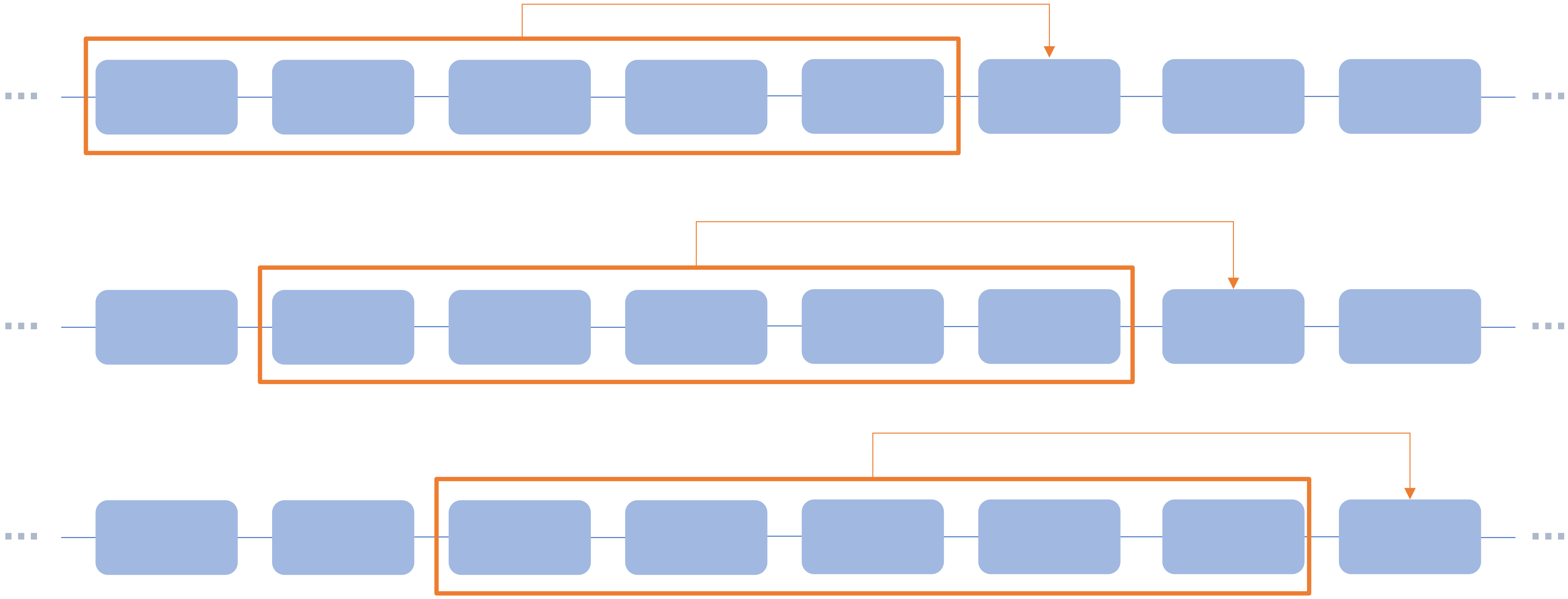
Bitcoin’s Target Recalculation Function

$$T' = \begin{cases} \frac{1}{\tau} \cdot T^{avg} & if \ \frac{\Delta}{m/f} \cdot T^{avg} < \frac{1}{\tau} \cdot T^{avg} \\ \tau \cdot T^{avg} & if \ \frac{\Delta}{m/f} \cdot T^{avg} > \tau \cdot T^{avg} \\ \frac{\Delta}{m/f} \cdot T^{avg} & otherwise \end{cases}$$

- *Simple Moving Average* (SMA):
  - Adjusts the mining difficulty after each block
  - A sliding window of last 144 blocks (approximately 1 day).
  - Based on the average target of the 144 blocks.
  - (Epoch-like) m: length of the sliding window.

Bitcoin's Target Recalculation Function (SMA)

$$T' = \frac{\Delta}{m/f} \cdot T^{avg}$$



Bitcoin’s Target Recalculation Function

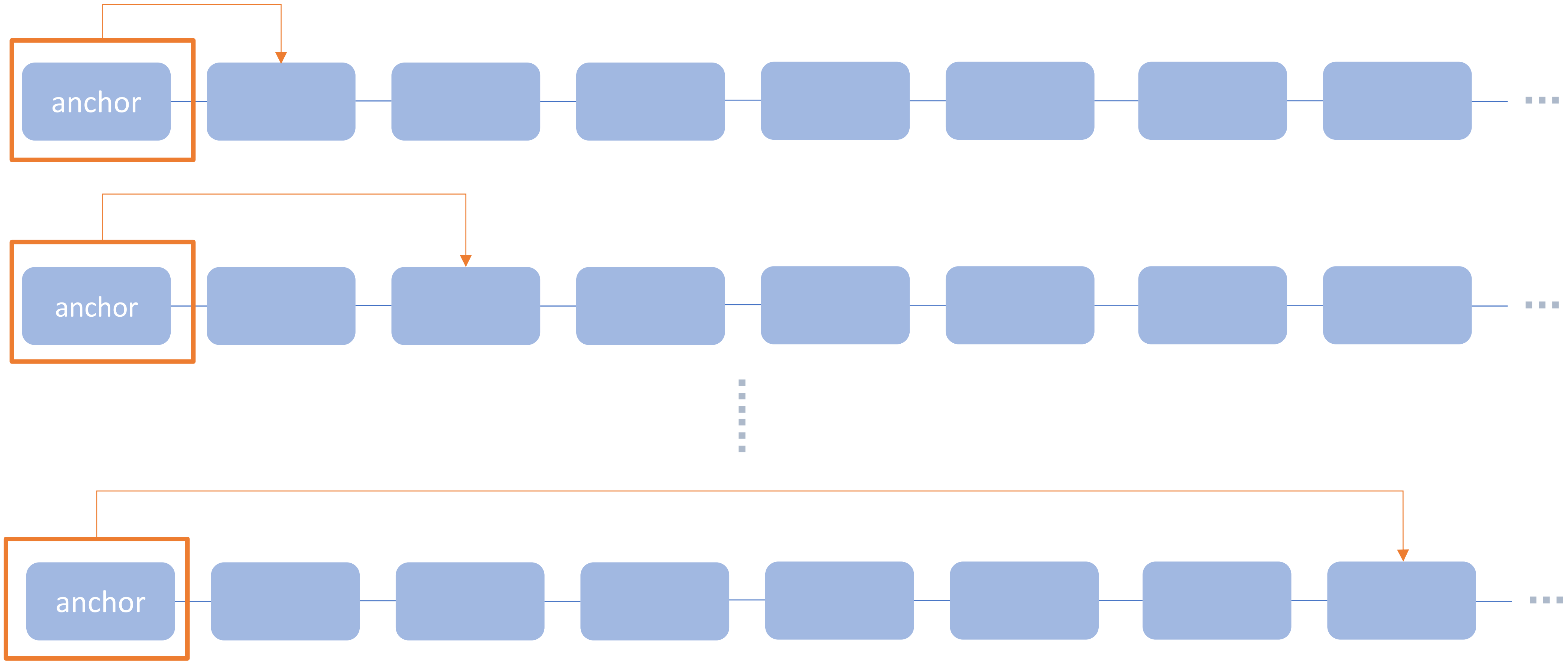
- *Absolutely Scheduled Exponentially Rising Targets* (ASERT):
  - Adjusts after each block.
  - Based on the comparison with the calibrated timestamp (the timestamp this block should have if it has the generating rate exactly f).
  - Intrinsically prevents the raising difficulty attack.
  - m: smoothing factor (288 in use, approximately 2 days).
- For v-th block with timestamp r_v, its target is calculated as

$$T' = T_0 \cdot 2^{\left(\frac{r_v - (v-1)/f}{m/f}\right)}$$

- Mathematical derivation: https://arxiv.org/abs/2006.03044

Bitcoin's Target Recalculation Function (ASERT)

$$T' = T_0 \cdot 2^{\left(\frac{r_v - (v-1)/f}{m/f}\right)}$$



This Work

- First formal analysis of Bitcoin Cash's target recalculation functions.
- New analysis methodology for target recalculation functions in the dynamic setting.
- Adopt the Bitcoin Backbone Protocol ([GKL15, GKL17]) as framework to analyze the security of Bitcoin Cash protocol.
- "Goodness" in Backbone Protocol: a property that shows the block generation rate is steady (close to f)
  - For SMA, it generally follows the approach in [GKL17], with improved proofs to overcome the adoption of average targets.
  - Previous analysis on goodness is *epoch-based*, which fails for the ASERT function.

Model

- Time is divided into *rounds*; network delay is Δ round bounded.
- Monotonically increasing timestamps.
- A total number of parties n and an adversary that controls t parties.
  - Honest parties act independently.
  - Parties controlled by the adversary collaborate.
- Parties communicate by *diffusing* a message.
  - The adversary can inject messages into a party's incoming message.
  - The adversary can reorder a party's incoming messages.
- Pseudonymous setting: parties cannot associate a message to a sender.
- Hash function is modeled as a *random oracle* (RO).

Respecting Environment^{++}

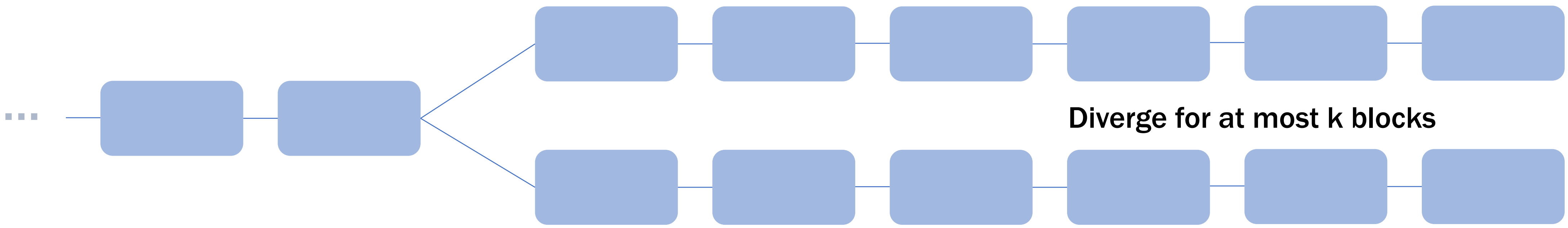
Static | **Permissionless** | **Dynamic**

- It is impossible to achieve desired properties in permissionless setting.
  - If the number of parties **increases** rapidly, it would generate too many forks (*Consistency* hurts).
  - If the number of parties **decreases** rapidly, transactions sent to the ledger cannot be confirmed (*Liveness* breaks).
- A dynamic respecting environment: the fluctuation of number of parties is bounded (cf. [GKL17]).

Definition 1. For $\gamma, \Gamma \in \mathbb{R}^+$, we call a sequence $(n_r)_{r\in\mathbb{N}}$ $(\langle\gamma,\sigma\rangle, \langle\Gamma,\Sigma\rangle)$-*respecting* if it holds that in a sequence of rounds S with $|S| \le \Sigma$ rounds, $\max_{r\in S} n_r \le \Gamma \cdot \min_{r\in S} n_r$ and for any consecutive sub-sequence rounds $S' \preccurlyeq S$ with $|S'| \le \sigma$ rounds, $\max_{r\in S'} n_r \le \gamma \cdot \min_{r\in S'} n_r$.

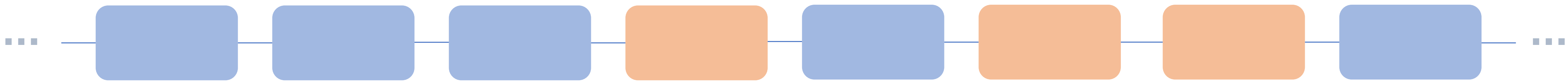
Blockchain Properties

Common Prefix:



Diverge for at most k blocks

Chain Quality:



The percentage of blocks mined by the adversary in the stable blockchain is bounded.

Ledger Property [GKL15]

A *robust transaction ledger* must satisfy:

Consistency

- For any two honest parties P_1, P_2, reporting $\mathcal{L}_1, \mathcal{L}_2$ at rounds $r_1 \leq r_2$, resp., it holds that the settled part of $\mathcal{L}_1$ is a prefix of $\mathcal{L}_2$.

Liveness

- If a transaction tx is provided to all honest parties for u consecutive rounds, then it holds that for any player P, tx will be in $\mathcal{L}$.

Summary of Parameters

- δ: Advantage of honest parties, $\forall r(t_r/h_r < 1-\delta)$.
- $\gamma, \sigma, \Gamma, \Sigma$: Determine how the number of parties fluctuates across rounds in a period (cf. Definition 1 and Fact 1).
- f: Probability that at least one honest party succeeds generating a PoW in a round assuming h_0 parties and target T_0 (the protocol's initialization parameters).
- m: Smoothing factor (cf. Definition 4).
- τ: Parameter that regulates the target that the adversary could query the PoW with.
- ϵ: Quality of concentration of random variables (cf. Definition 7).
- κ: The length of the hash function output.
- φ: Related to the properties of the protocol.
- L: The total number of rounds in the execution of the protocol.

$$\varphi = \Theta(m) = polylog(\kappa)$$

“Goodness” in ASERT function

$$T' = T_0 \cdot 2^{\left(\frac{r_v - (v-1)/f}{m/f}\right)}$$

- Observation: the next target in ASERT is w.r.t. timestamp and block height.
- Once we fix a sequence of number of parties:
  - For i-th block with timestamp r, and corresponding number of honest parties h_r, if $r = \frac{i-1}{f} + \frac{m}{f}\log\frac{h_0}{h_r}$ (the *calibrated timestamp*), the i-th block would have block generating rate exactly f.
  - r is a *good* target recalculation point if

$$\frac{i-1}{f} + \frac{m}{f}\log(2(2-\delta)\Gamma^3 \cdot \frac{h_0}{h_r}) \leq r \leq \frac{i-1}{f} + \frac{m}{f}\log(2\Gamma^3 \cdot \frac{h_0}{h_r})$$

“Goodness” in ASERT function

- A new variable X_i to describe the deviation of *calibrated timestamp*:

$$X_1 = 0 \text{ and } X_{i+1} = X_i + (r_{i+1} - r_i) - \frac{1}{f} - \frac{m}{f}\log(\frac{h_{i+1}}{h_i}) \text{ for } i \geq 0.$$

- Three parts:
  - $(r_{i+1} - r_i)$: the difference of their timestamps;
  - $1/f$: the ideal block interval;
  - (m/f)$\log(h_{i+1}/h_i)$: the influence of the party fluctuation.
- For good target recalculation points, X_i should satisfy

$$-\frac{m}{f}\log 2(2 - \delta)\Gamma^3 \leq X_i \leq \frac{m}{f}\log 2\Gamma^3.$$

“Goodness” in ASERT function

- **Problem**: we cannot bound the accumulation of the party fluctuation.

Definition 1. For $\gamma, \Gamma \in \mathbb{R}^+$, we call a sequence $(n_r)_{r\in\mathbb{N}}$ $(\langle\gamma,\sigma\rangle, \langle\Gamma,\Sigma\rangle)$-*respecting* if it holds that in a sequence of rounds S with $|S| \leq \Sigma$ rounds, $\max_{r\in S} n_r \leq \Gamma \cdot \min_{r\in S} n_r$ and for any consecutive sub-sequence rounds $S' \preccurlyeq S$ with $|S'| \leq \sigma$ rounds, $\max_{r\in S'} n_r \leq \gamma \cdot \min_{r\in S'} n_r$.

- The sequence allows for exponential growth.
  - The total run time is bounded by a polynomial (in κ), and thus the growth is also polynomially bounded.
- However, this is not enough for term $\frac{m}{f}\log(\frac{h_{i+1}}{h_i})$ (see above).

“Goodness” in ASERT function

- A new variable W_i to describe the deviation of a specific *calibrated timestamp* (i.e., *relatively calibrated timestamp*):

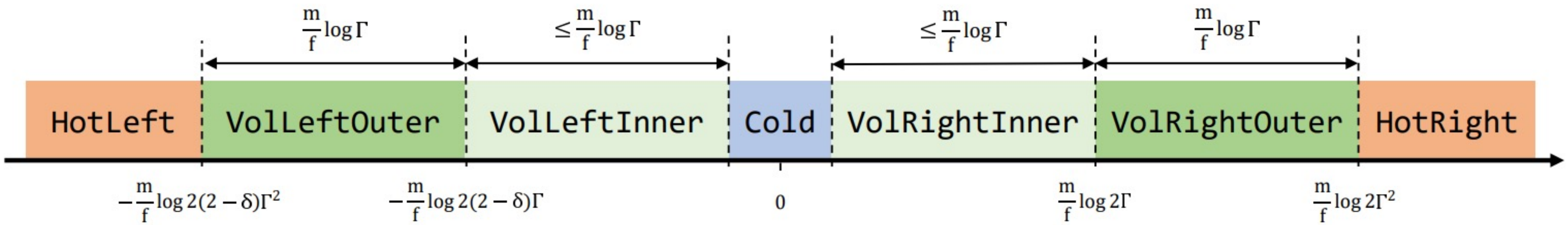
$$W_u = X_u \text{ and } W_{i+1} = W_i + (r_{i+1} - r_i) - \frac{1}{f} \text{ for } i \geq u.$$

- Two parts:
  - $(r_{i+1} - r_i)$: the difference of their timestamps;
  - $1/f$: the ideal block interval.
- For good target recalculation points, W_i should satisfy

$$-\frac{m}{f} \log 2(2 - \delta)\Gamma^2 \leq W_i \leq \frac{m}{f} \log 2\Gamma^2.$$

“Goodness” in ASERT function

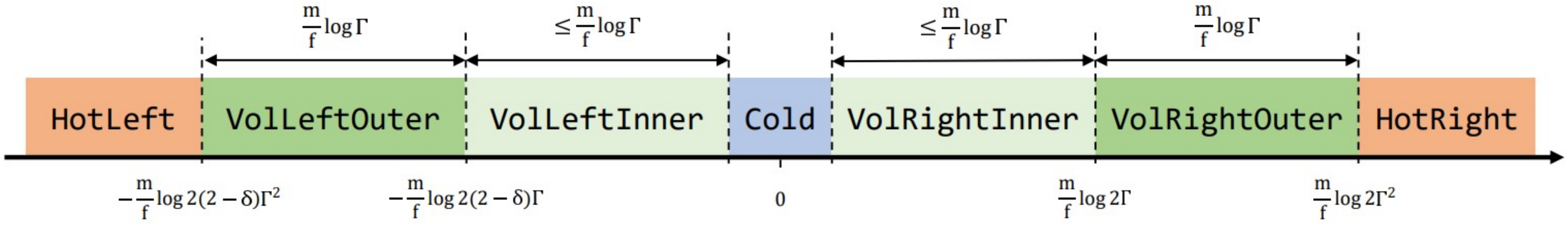
- The states based on W_i:



- For good target recalculation points, W_i should satisfy

$$-\frac{m}{f}\log 2(2-\delta)\Gamma^2 \leq W_i \leq \frac{m}{f}\log 2\Gamma^2.$$

“Goodness” in ASERT function



- For blocks $\{B_u, \ldots, B_v\}$ in in a sliding window, it holds that:
  1. If W_u is in state VolatileLeftInner, VolatileRightInner or Cold, the probability of W_i $(i > u)$ reaching HotLeft or HotRight is negligible.
     - Never escape to the Hot state (i.e., never break goodness).
  2. If W_u is in state VolatileLeftInner, VolatileRightInner or Cold, $W_i (i > u)$ will once return to Cold with overwhelming probability.
     - Always feasible to move the sliding window.
  3. For a block B_i $(i > u)$, with W_i (w.r.t. B_u) in state Cold, we can construct a new sliding window with W_i (w.r.t. B_i) in state VolatileLeftInner, VolatileRightInner or Cold.
     - Extend the analysis of a sliding window from the beginning to the whole execution.

Conditions to be satisfied

- In order to satisfy the analysis, two conditions on the parameters should be satisfied:
  - We will assume that ℓ is appropriately small compared to the length m of a sliding interval/window:

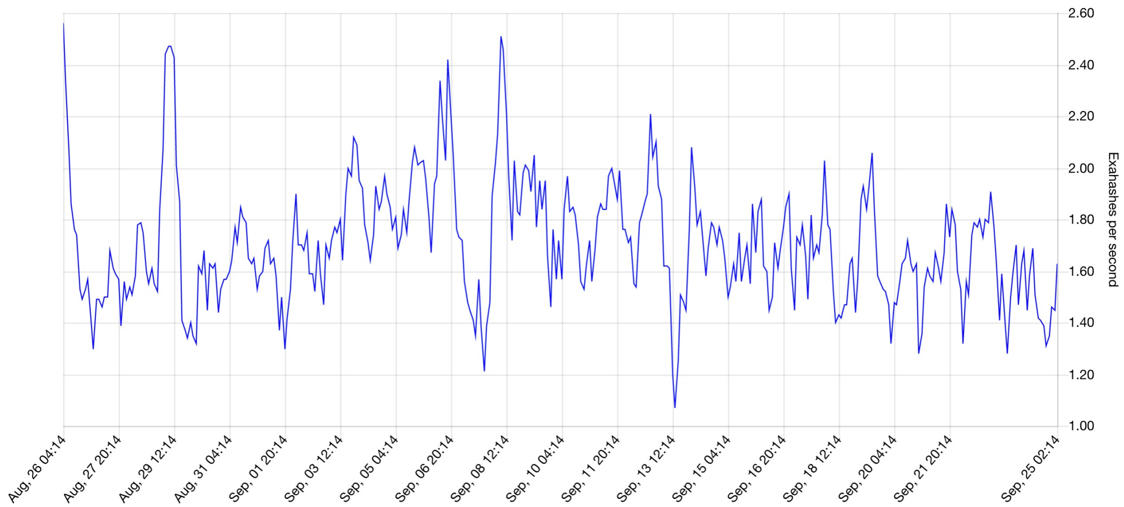
$$2\ell + 6\Delta \leq \frac{\epsilon m}{2\gamma\Gamma^3 f}.$$

  - The advantage δ of the honest parties over adversarial parties to be large enough to absorb error factors:

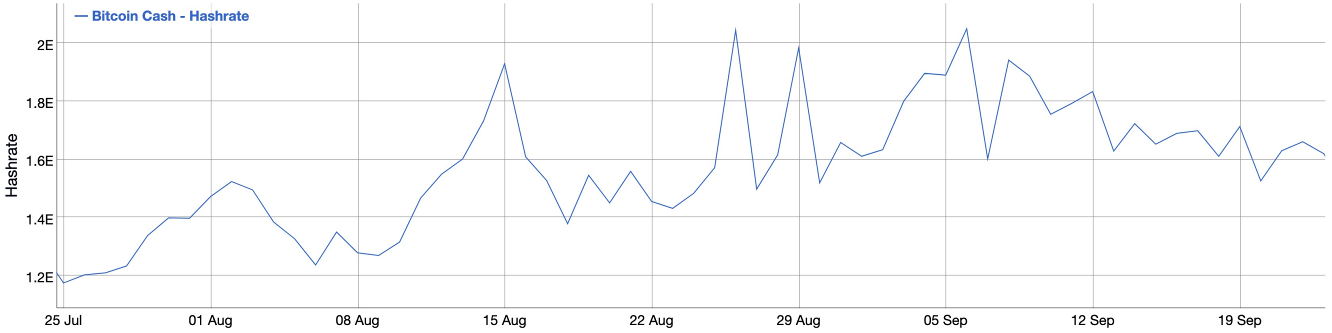
$$[1 - 2\gamma\Gamma^3 f]^{\Delta} \geq 1 - \epsilon \text{ and } \epsilon \leq \delta/8 \leq 1/8.$$

Bitcoin Cash's Party Fluctuation Ratio (Γ, γ)

- Extract from *hashrate.*
- Real-time hashrate: party fluctuation ratio > 8
- Adopt daily average hashrate.
- We consider two environments:
    1. quiet environment with $\Gamma = 1.398$ and $\gamma = 1.057$
    2. wild fluctuation with $\Gamma = 1.88$ and $\gamma = 1.099$



Real-time hashrate, source: https://fork.lol/pow/hashrate



Daily average hashrate, source: https://bitinfocharts.com/zh/comparison/bitcoin cash-hashrate.html

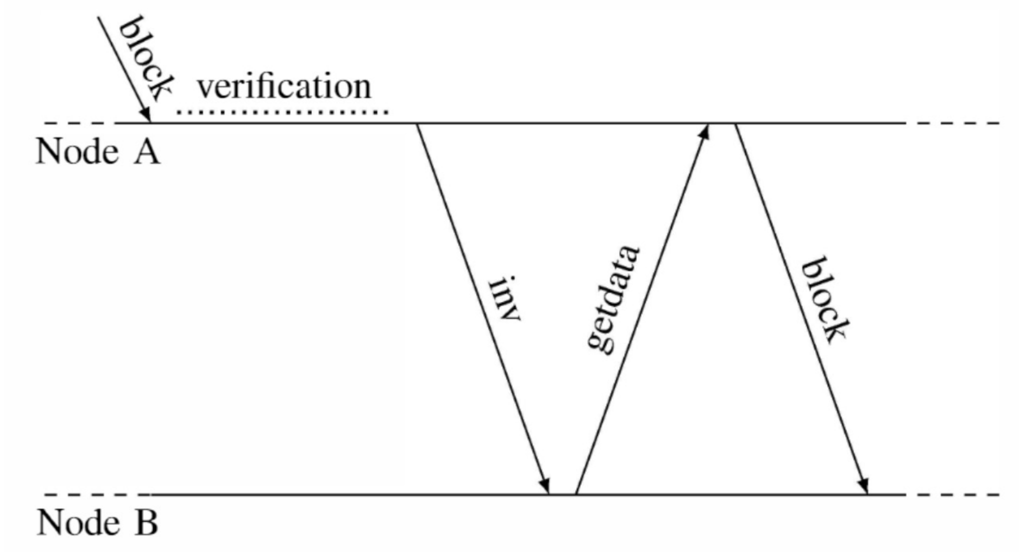
Bitcoin Security under Temporary Dishonest Majority

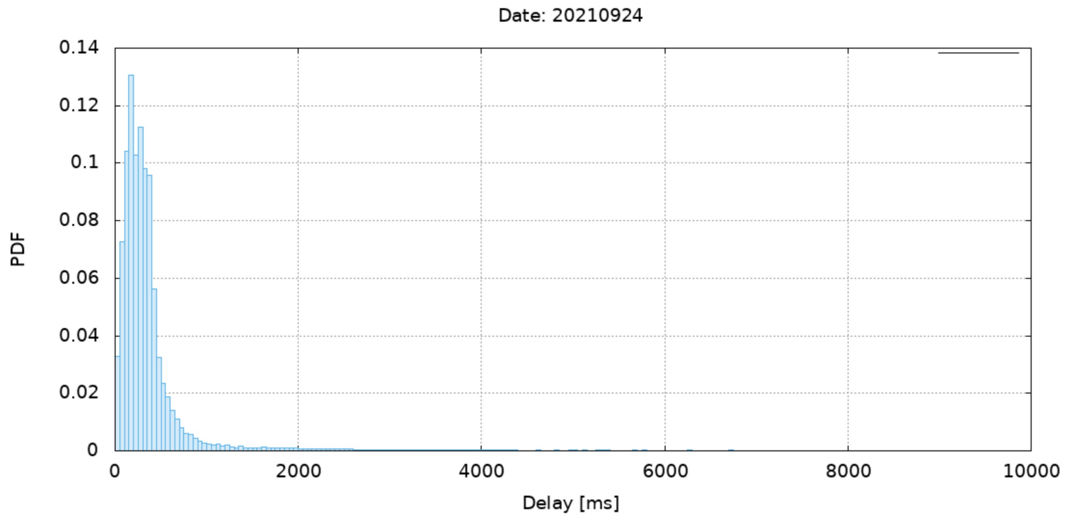
Georgia Avarikioti, Lukas Kaeppeli, Yuyi Wang, Roger Wattenhofer

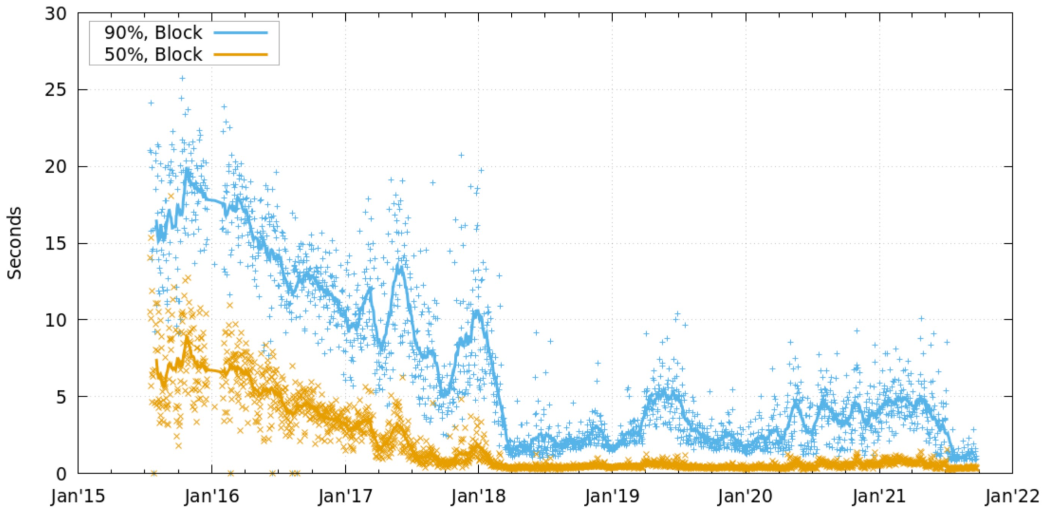
We prove Bitcoin is secure under temporary dishonest majority. We assume the adversary can corrupt a specific fraction of parties and also introduce crash failures, i.e., some honest participants are offline during the execution of the protocol. We demand a majority of honest online participants on expectation. We explore three different models and present the requirements for proving Bitcoin's security in all of them: we first examine a synchronous model, then extend to a bounded delay model and last we consider a synchronous model that allows message losses.

Bitcoin Cash's Block Propagation Time

- Network delay (Δ).
- Mainly stems from its multi-hop broadcast and block propagation mechanism.







Source: https://www.dsn.kastel.kit.edu/bitcoin/index.html

Real World Network & Parameters

Parameter	Value
Block generating rate f	0.01 (1 round = 6 seconds)
Network delay Δ	1 (=1 round=6 seconds)
Party fluctuation ratio Γ, γ	1.88, 1.099
Honest advantage δ	0.99
Quality of concentration ϵ	0.123

$$2\ell + 6\Delta \leq \frac{\epsilon m}{2\gamma\Gamma^3 f}.$$

$$[1 - 2\gamma\Gamma^3 f]^{\Delta} \geq 1 - \epsilon \text{ and } \epsilon \leq \delta/8 \leq 1/8.$$

Conclusions

- Under current parameters, the probability to escape to Hot state (break the goodness) is tiny ($< 10^{-9}$).
- Under current parameters, the probability of not returning to Cold state is also tiny ($< 10^{-12}$).
- ASERT is better than SMA, because wilder fluctuation can be inserted into ASERT function.
  - SMA fails when we use party fluctuation ratio $\Gamma = 1.88$.
- In order to achieve desired ledger properties, the smoothing factor m should be much larger (approximately several years) to get the ideal ledger properties.
- A target recalculation function framework?

Thank you!

Full version: https://eprint.iacr.org/2021/143.pdf