TEXAS A&M UNIVERSITY
Engineering

Thesis Defense:
A Formal Analysis of Bitcoin Cash

Yu Shen

Outline

- Background
- Bitcoin Cash Backbone Protocol
- Comparison with Real World Network

Cryptocurrencies

#	Name	Price	24h	7d	Market Cap	Volume	Circulating Supply	Last 7 Days
1	Bitcoin BTC	$49,795.58	▲ 8.43%	▼ 8.50%	$924,610,894,312	$53,548,614,604 1,079,658 BTC	18,642,200 BTC	
2	Ethereum ETH	$1,581.71	▲ 9.77%	▼ 12.07%	$179,839,542,025	$23,978,480,535 15,316,682 ETH	114,875,712 ETH	
3	Cardano ADA	$1.30	▼ 2.16%	▲ 19.21%	$41,459,440,348	$10,000,495,100 7,706,300,647 ADA	31,948,309,441 ADA	
11	Bitcoin Cash BCH	$506.79	▲ 8.67%	▼ 19.97%	$9,365,354,777	$3,635,754,277 7,247,188 BCH	18,668,063 BCH	

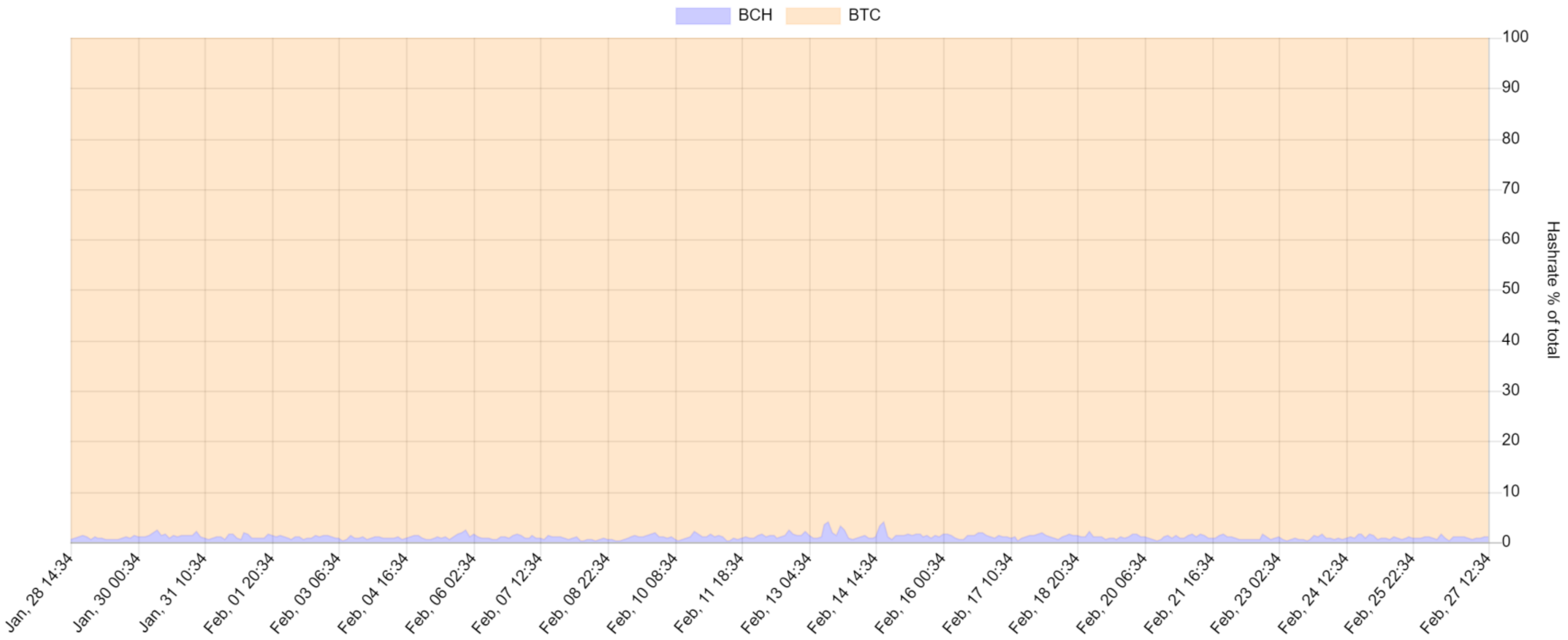
Basic Info about Bitcoin Cash

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

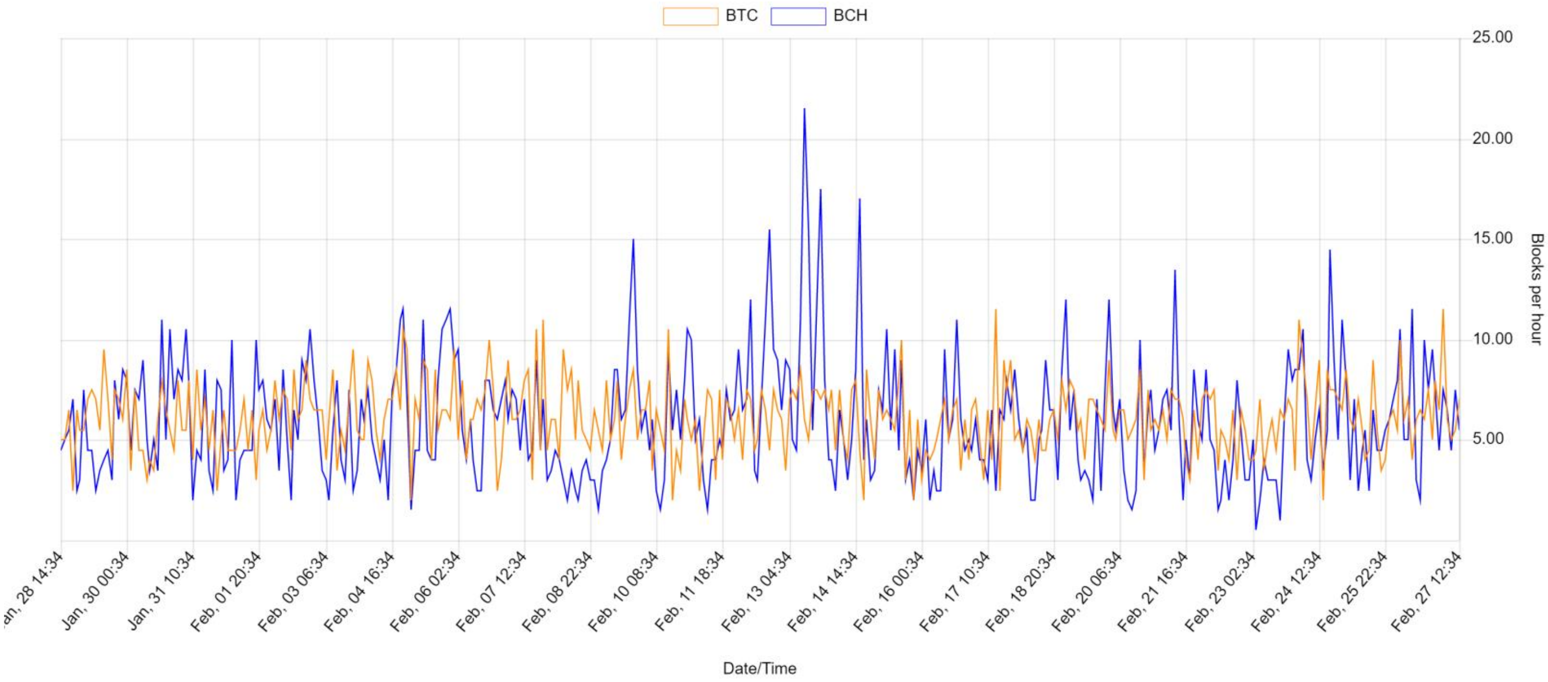
Bitcoin Cash vs. Bitcoin

	Bitcoin	Bitcoin Cash
Ledger Start	Jan 3 2009	Jan 3 2009, split at Aug 1 2017
Mining	Proof-of-Work(SHA-256)	
Block Size Limit	1MB -> 4MB	8MB -> 32MB
Issuance schedule	Initially 50 BTC(BCH) per block, halved every 210,000 blocks	
Block time	10 minutes	
Supply Limit	21,000,000 BTC(BCH)	
Target Recalculation	Every 2 weeks	Every block

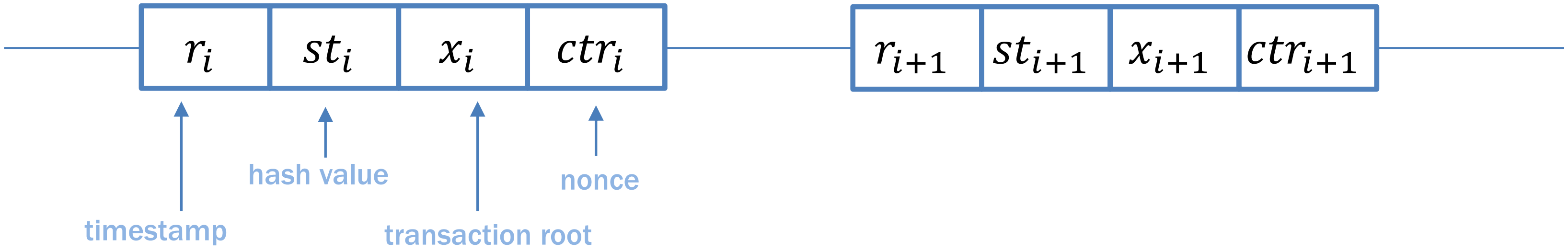
Relative Hashrate in Percentage of Total



Average Number of Blocks per Hour



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:

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

Bitcoin Cash's Target Recalculation Function

TEXAS A&M UNIVERSITY Engineering

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 DAA + decrease the mining difficulty of Bitcoin Cash 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 last 144 blocks.
- *Absolutely Scheduled Exponentially Rising Targets* (ASERT)

Bitcoin's Target Recalculation Function

TEXAS A&M UNIVERSITY Engineering

- 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

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

Bitcoin's Target Recalculation Function

TEXAS A&M UNIVERSITY Engineering

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

- 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 DAA + decrease the mining difficulty of Bitcoin Cash 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 last 144 blocks.
- *Absolutely Scheduled Exponentially Rising Targets* (ASERT)

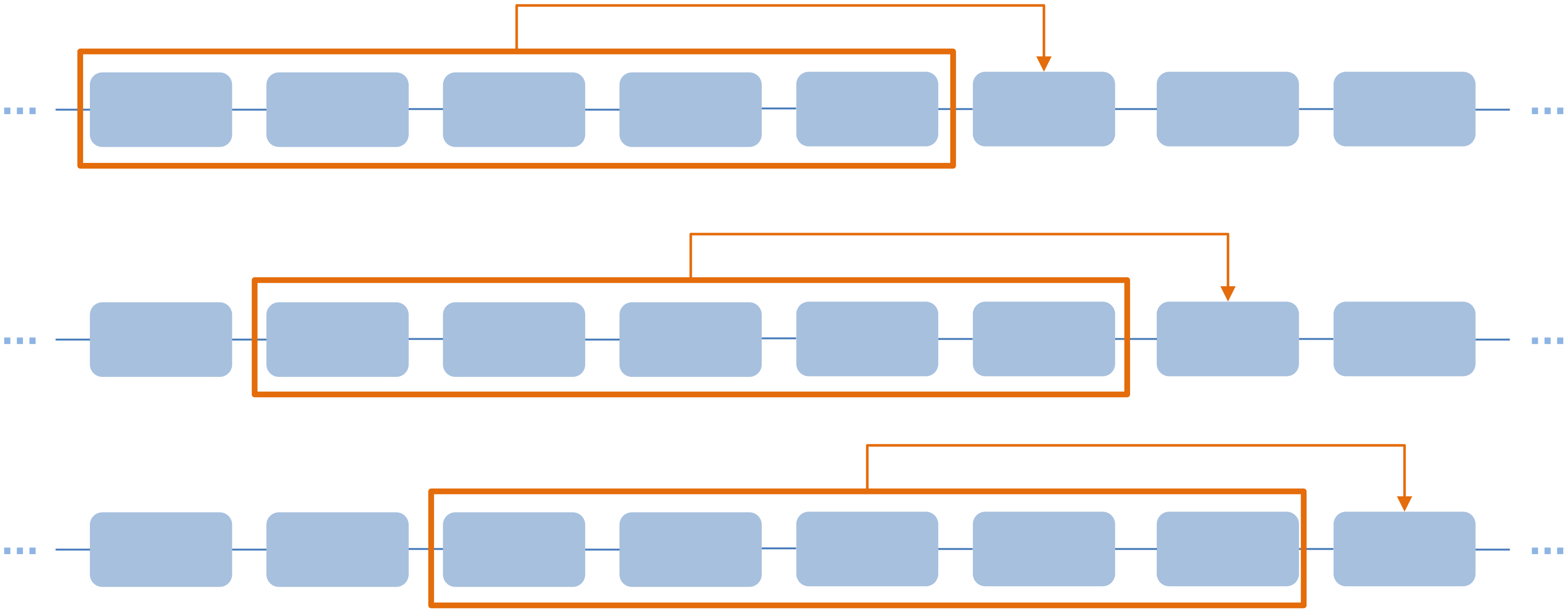
Bitcoin's Target Recalculation Function

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

- *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 Cash's Target Recalculation Function

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



Bitcoin Cash's Target Recalculation Function

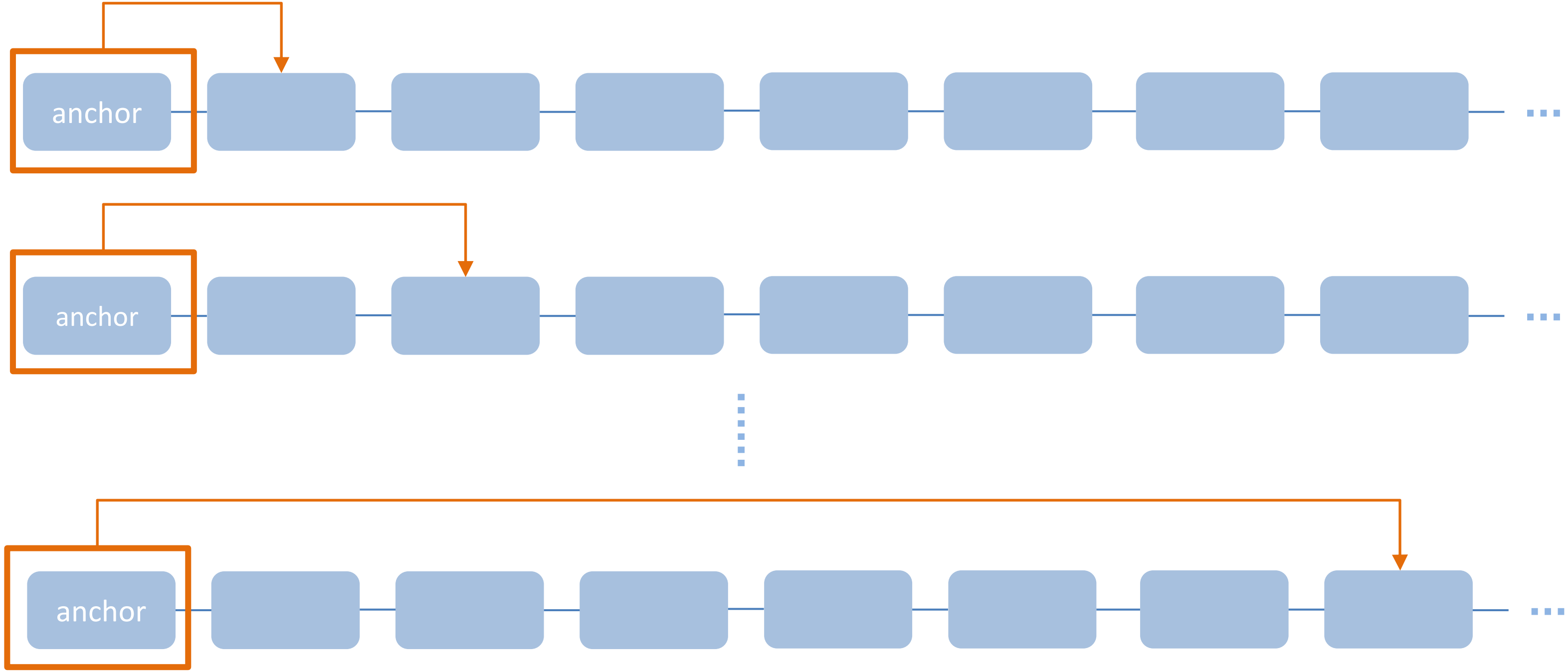
- *Absolutely Scheduled Exponentially Rising Targets* (ASERT):
  - Adjusts after each block.
  - Based on the comparison with the calibrated timestamp (the timestamp this block should have if it has the generating rate exactly f).
  - Intrinsically prevents the raising difficulty attack.
  - m: smoothing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

- For v-th block with timestamp r_v, its target is calculated by

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

Bitcoin Cash's Target Recalculation Function

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



Outline

- Background
- Bitcoin Cash Backbone Protocol
- Comparison with Real World Network

This Work

- A follow-up work of the Bitcoin Backbone Protocol ([GKL15, GKL17]).
- First formal analysis of Bitcoin Cash's target recalculation functions.
- New analysis methodology for target recalculation functions in the dynamic setting.

Model

- Time is divided into rounds.
- Bounded Delay Network: Δ round delay.
- 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.
- Anonymous setting: parties cannot associate a message to a sender.
- Hash function is modeled as a random oracle (RO).

Respecting Environment

TEXAS A&M UNIVERSITY Engineering

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

- It is impossible to achieve desired properties in permissinless 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 environment: the fluctuation of number of parties is bounded.

Respecting Environment

Static

Permissionless

Dynamic

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 environment Z can increase or decrease the total number of parties at the beginning of each round, but subject to a constraint.
  - Long term fluctuation: $\forall S, |S| = \Sigma, \max_{r \in S} n_r \leq \Gamma \cdot \min_{r \in S} n_r$.
  - Short term fluctuation: $\forall S', |S'| = \sigma, \max_{r \in S} n_r \leq \gamma \cdot \min_{r \in S} n_r$.
  - Consistent with the recalculation function that adjusts difficulty for each block.

Blockchain Properties

TEXAS A&M UNIVERSITY Engineering

Common Prefix

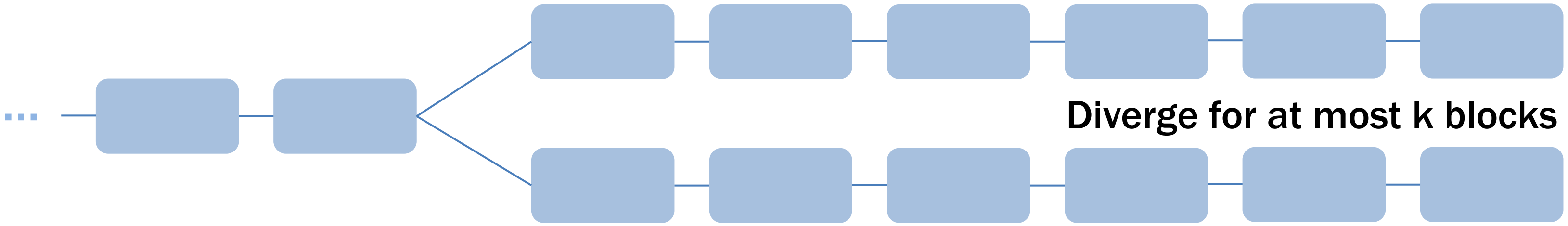
- With parameter $k \in N$, at any round of the execution, if a chain C belongs to an honest party, then for any valid chain C' in the same round such that either diff(C') $>$ diff(C), or diff(C') $=$ diff(C) and head(C')was computed no later than head(C), it holds that $C^{\lceil k} \preccurlyeq C'$ and $C'^{\lceil k} \preccurlyeq C$.

Chain Quality

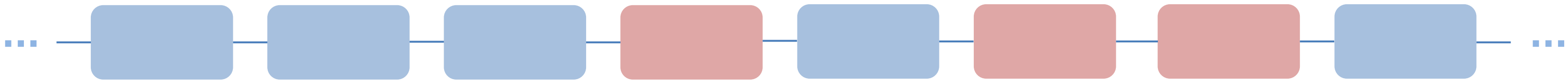
- With parameters $\mu \in R$ and $\ell \in N$, for any party P with chain C in $view_{\Pi,A,Z}$, and any segment of that chain of difficulty d such that the timestamp of the first block of the segment is at least ℓ smaller than the timestamp of the last block, the blocks the adversary has contributed in the segment have a total difficulty that is at most $\mu \cdot d$.

Blockchain Properties

Common Prefix:



Chain Quality:



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

Ledger Properties

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 $\mathcal{L}$.

Bitcoin Cash Backbone Protocol

TEXAS A&M UNIVERSITY Engineering

- In each round r, each party with a chain C_0 performs the following:
  - Receive from the network chains $C_1, C_2, \ldots$
  - Choose the first heaviest chain C among the valid ones in $\{C_0, C_1, C_2, \ldots\}$ (Heaviest means the largest accumulated difficulty).
  - Try to extend the heaviest chain C (Modeled as a Bernoulli trial with a probability of success that depends on the target T).
  - Suppose its last block is the i-th one and equal to (r_i, st_i, x_i, ctr_i) with $st = H(r_i, st_i, x_i, ctr_i)$. Find a ctr such that $H(r, st, x, ctr) < T$. If succeed, let $C \leftarrow C \parallel (r, st, x, ctr)$
  - If $C \neq C_0$(miner extends the chain or switch to another heavier chain), diffuse the new chain C.

Bitcoin Cash Backbone Protocol

Algorithm 4 The Bitcoin Cash backbone protocol in the dynamic setting at round "round" on local state $(st, \mathcal{C})$ parameterized by the *input contribution function* $I(\cdot)$ and the *chain reading function* $R(\cdot)$. The ready flag is **false** if and only if the party was inactive in the previous round.

if ready = **true then**
    $\textsc{Diffuse}(\text{'ready'})$
    $\tilde{\mathcal{C}} \leftarrow \mathsf{maxvalid}(\mathcal{C}$ all chains $\mathcal{C}'$ found in $\textsc{Receive}())$
    $\langle st, x\rangle \leftarrow I(st, \tilde{\mathcal{C}}, \mathtt{round}, \textsc{Input}(), \textsc{Receive}())$
    $\mathcal{C}_{\mathsf{new}} \leftarrow \mathsf{pow}(\mathtt{round}, x, \tilde{\mathcal{C}})$
    if $(\mathcal{C} \neq \mathcal{C}_{\mathsf{new}}) \vee (\text{'}\mathbf{Join}\text{'} \in \textsc{Receive}())$ **then**
        $\mathcal{C} \leftarrow \mathcal{C}_{\mathsf{new}}$
        $\textsc{Diffuse}(\mathcal{C})$ ▷ chain is diffused when it is updated or when someone wants to join.
    end if
    if $\textsc{Input}()$ contains $\textsc{Read}$ **then**
        write $R(\mathbf{x}_{\mathcal{C}})$ to $\textsc{Output}()$
        $\textsc{Diffuse}(\mathsf{RoundComplete})$
    end if
else
    ready $\leftarrow$ **true**
    $\textsc{Diffuse}(\mathsf{Join}, \mathsf{RoundComplete})$
end if

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.

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

Proof Roadmap

- Assuming the execution begins with good initial parameters (the initial block production rate is very close to f).
- Consider a sliding window of $\Theta(m)$ rounds.
- If a chain is C is adopted by an honest party, then C satisfies the following with overwhelming probability (in κ):
  - Is never abandoned by honest parties for $\Omega(m/f)$ rounds,
  - Is $O(m/f)$-accurate,
  - Has “good” recalculation points,
  - Has blocks with good targets.

Proof Roadmap

TEXAS A&M UNIVERSITY Engineering

- Accuracy: no adversarial blocks are present with a timestamp that deviates too much from its real creation time.

Definition 6 (Accuracy). A block created at round u is *accurate* if it has a timestamp v such that $|u - v| \leq \ell + 2\Delta$. A chain is *accurate* if all its blocks are accurate. A chain is *stale*, if for some $u \geq \ell + 2\Delta$, it does not contain an honest block with timestamp $v \geq u - \ell - 2\Delta$.

- Goodness: for a round r, the probability of block generation given current target T_r and number of miners n_r, is very close to the initial block generation rate f.

Definition 5 (Goodness). Round r is *good* if $f/2\gamma(2-\delta)\Gamma^3 \leq ph_r T_r^{\min}$ and $ph_r T_r^{\max} \leq 2\gamma\Gamma^3 f$. A target-recalculation point r is *good* if the target T for the next block satisfies $f/2(2-\delta)\Gamma^3 \leq ph_r T \leq 2\Gamma^3 f$. A chain is *good* if all its target-recalculation points are good.

Random Variables

- D_r: Honest party successfully extends a chain.
  - Sum of the difficulties of all blocks computed by honest parties.
- Y_r: Maximum difficulty among all blocks computed by honest parties.
- Q_r: Isolated successful (consider the Δ round delay).
  - Equal to Y_r when $D_u = 0$ for all $r < u < r + \Delta$ and 0 otherwise.

- Adversary: consider a set of consecutive adversarial queries J.
  - $A(J)$: sum of the difficulties of all adversarial blocks in J for target at least $T(J)/\tau$
  - $B(J)$: sum of the difficulties of all adversarial blocks in J for target at least $T(J)$

Typical executions

TEXAS A&M UNIVERSITY Engineering

- For the honest parties:

For any set S of at least ℓ consecutive good rounds,

$$(1-\epsilon)[1-2\gamma\Gamma^3 f]^{\Delta} ph(S) < Q(S) \leq D(S) < (1+\epsilon)ph(S).$$

- For the adversarial parties:

For any set J indexing a set of consecutive adversarial queries and $\alpha(J) = 2(\frac{1}{\epsilon} + \frac{1}{3})\varphi/T(J)$,

$$A(J) < p|J| + \max\{\epsilon p|J|, \tau\alpha(J)\} \text{ and } B(J) < p|J| + \max\{\epsilon p|J|, \alpha(J)\}.$$

- No insertions, copies, predictions.

Typical executions

- Get concentration of the random variables:

Definition 8. [MU05, Chapter 12] A sequence of random variables $X_0, X_1, \ldots$ is a martingale with respect to sequence $Y_0, Y_1, \ldots$, if, for all $n \geq 0$, (1)X_n is a function of $Y_0, \ldots, Y_n$, (2)$\mathbb{E}[|X_n|] < \infty$, and (3) $\mathbb{E}[X_{n+1}|Y_0, \ldots, Y_n] = X_n$.

Theorem 16. [McD98, Theorem 3.15] *Let $X_0, X_1, \ldots$ be a martingale with respect to the sequence $Y_0, Y_1, \ldots$. For $n \geq 0$, let $V = \sum_{i=1}^{n} \mathrm{var}(X_i - X_{i-1}|Y_0, \ldots, Y_{i-1})$ and $b = \max_{1 \leq i \leq n} \sup(X_i - X_{i-1} |Y_0, \ldots, Y_{i-1})$, where $\sup$ is taken over all possible assignments to $Y_0, \ldots, Y_{i-1}$. Then, for any $t, v \geq 0$,*

$$\Pr[(X_n \geq X_0 + t) \wedge (V \leq v)] \leq \exp\left\{ -\frac{t^2}{2v + 2bt/3} \right\}.$$

Theorem 3. *Assuming the Bitcoin Cash backbone protocol runs for L rounds, the event "E is not typical" is bounded by $\mathsf{poly}(L) \cdot e^{-\Omega(\mathsf{polylog}(\kappa))}$.*

Typical executions

TEXAS A&M UNIVERSITY Engineering

- Accuracy: can be proved by the properties of the typical execution, the honest party would accumulate more difficulties than the adversary party after some rounds.

- Goodness:
  - for SMA, it generally follows the approach in [GKL17], with modifications to overcome the adoption of average targets.
  - However, the previous analysis on goodness is *epoch-based*, which fails in the ASERT function.

“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 can capture 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 $\frac{m}{f} \log(\frac{h_{i+1}}{h_i})$ in the steps.

"Goodness" in ASERT function

TEXAS A&M UNIVERSITY Engineering

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

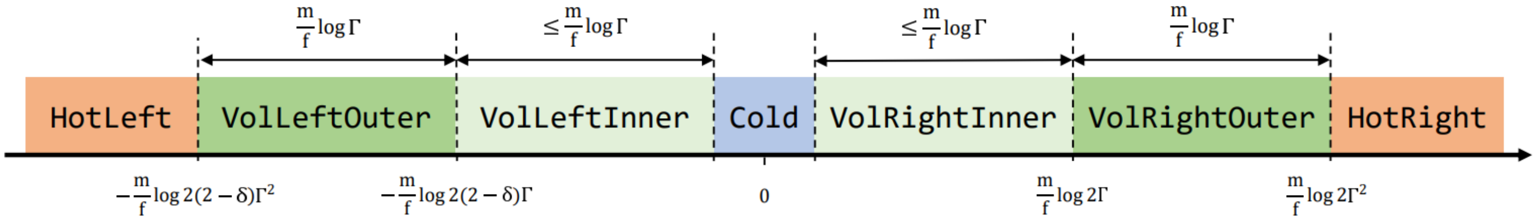
$$W_u = X_u \text{ and } W_{i+1} = W_i + (r_{i+1} - r_i) - \frac{1}{f} \text{ for } i \geq u.$$

- 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

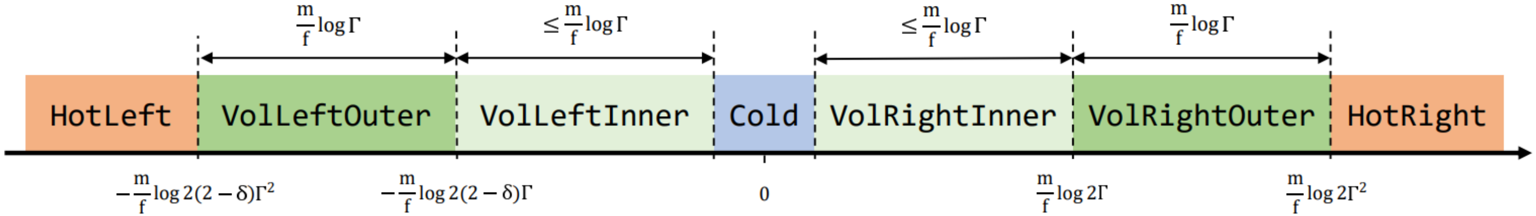
- The states based on W_i:



- For good target recalculation points, W_i should satisfy

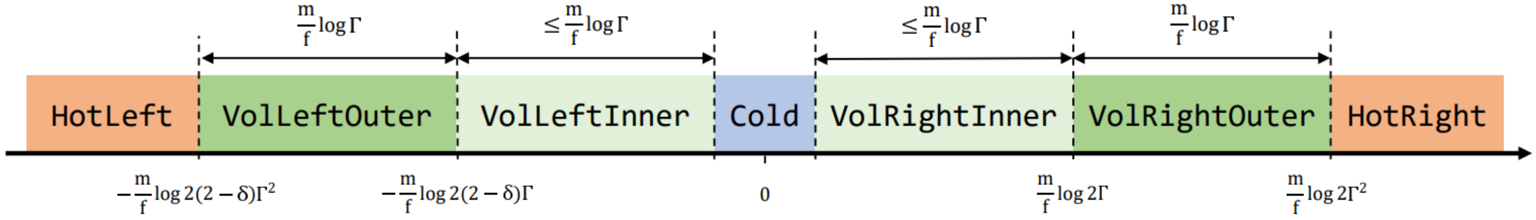
$$-\frac{m}{f}\log 2(2-\delta)\Gamma^2 \leq W_i \leq \frac{m}{f}\log 2\Gamma^2.$$

“Goodness” in ASERT function



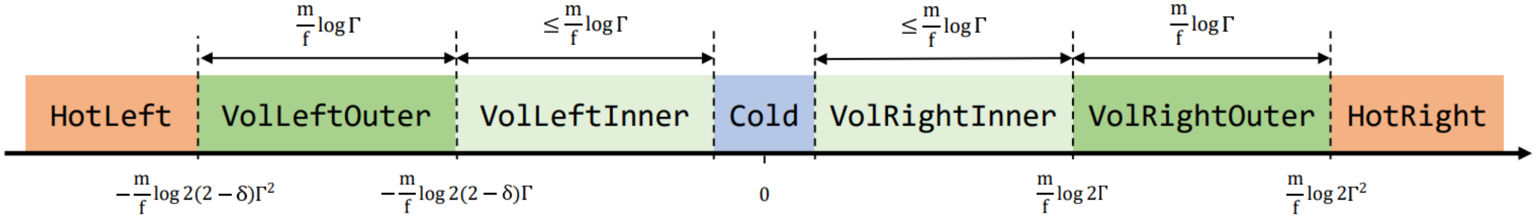
- For blocks $\{B_u, \ldots, B_v\}$ in in a sliding window, it holds that:
  - For a block $B_i, i > u$, with W_i (w.r.t. B_u) in state Cold, we can construct a new sliding window with W_i (w.r.t. B_i) in state VolatileLeftInner, VolatileRightInner or Cold.
  - Extend the analysis of a sliding window from the beginning to the whole execution.

“Goodness” in ASERT function



- For blocks $\{B_u, \ldots, B_v\}$ in in a sliding window, it holds that:
  - 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).

“Goodness” in ASERT function



- For blocks $\{B_u, \dots, B_v\}$ in in a sliding window, it holds that:
  - 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.

Conditions in the analysis

- 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:

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

Outline

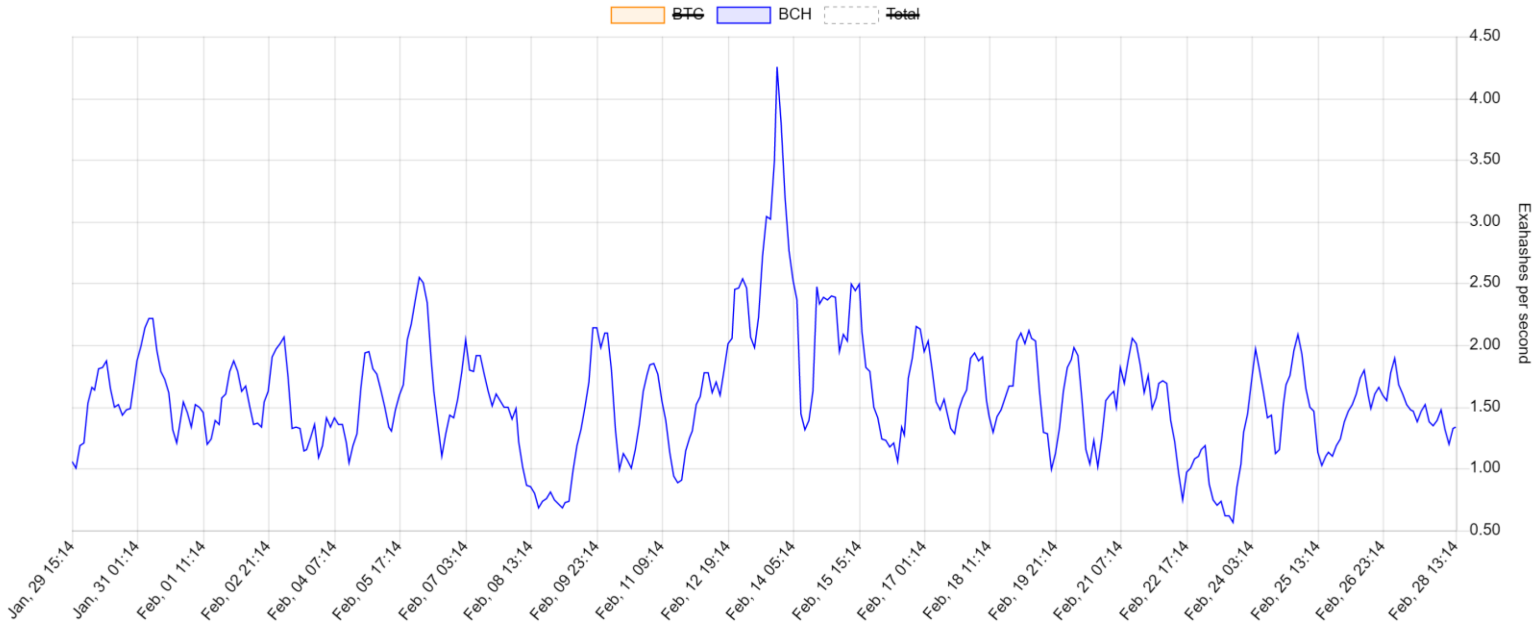
- Background
- Bitcoin Cash Backbone Protocol
- Comparison with Real World Network

Real World Network & Parameters

TEXAS A&M UNIVERSITY Engineering

- Party fluctuation (Γ, γ).
  - Extract it from *hashrate.*
- Network delay (Δ).
  - Mainly stems from its multi-hop broadcast and block propagation mechanism.
  - Block propagation time was $> 15s$ in 2014.
- Honest advantage (δ).
- Quality of concentration (ϵ).

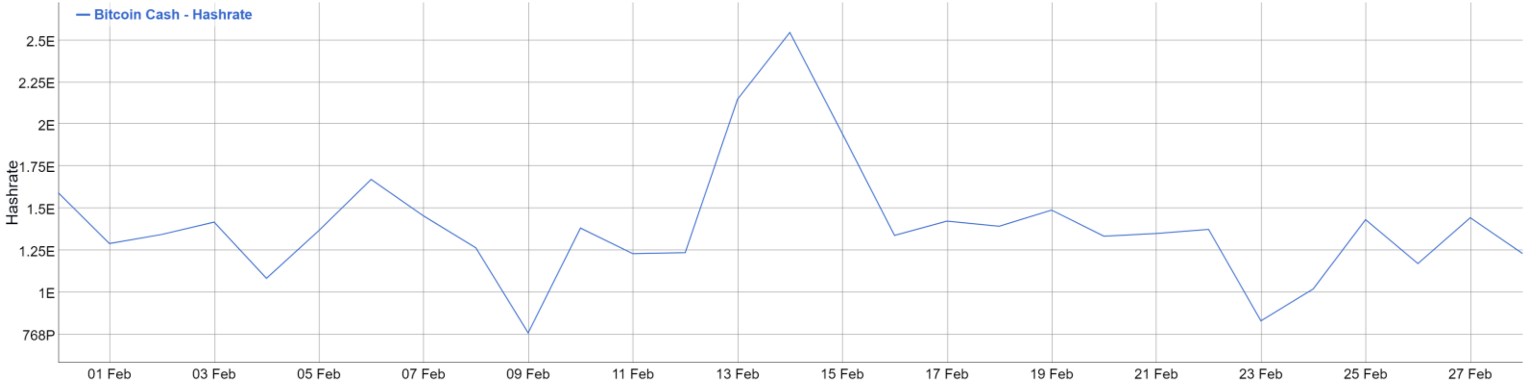
Bitcoin Cash’s Hashrate per Second



Party fluctuation ratio > 8.

Bitcoin Cash’s Daily Average Hashrate

TEXAS A&M UNIVERSITY
Engineering

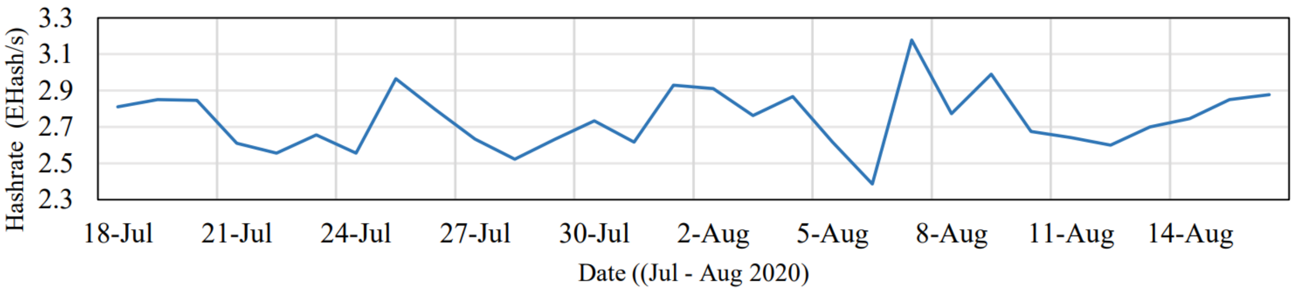


Bitcoin Security under Temporary Dishonest Majority

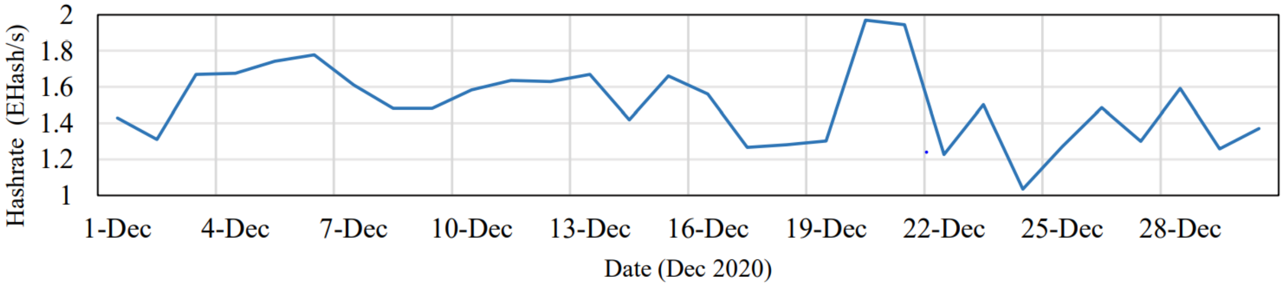
Georgia Avarikioti, Lukas Kaeppeli, Yuyi Wang, Roger Wattenhofer

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 Daily Average Hashrate



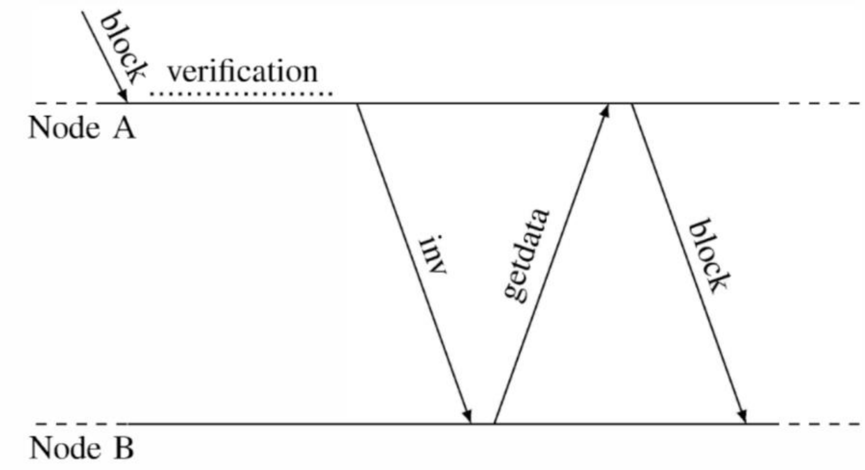
A quiet environment with $\Gamma = 1.398$ **and** $\gamma = 1.057$**.**



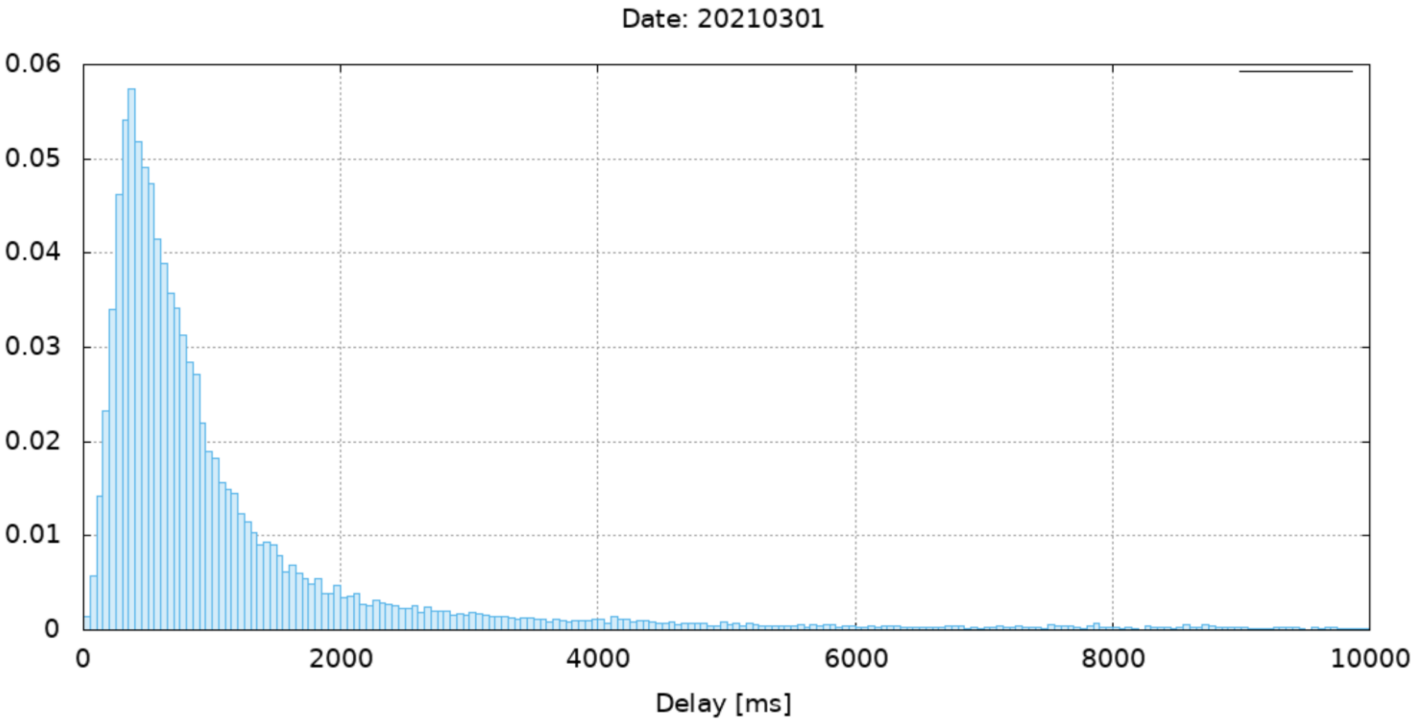
An environment of wild fluctuation with $\Gamma = 1.88$ **and** $\gamma = 1.099$**.**

Bitcoin Cash's Block Propagation Time

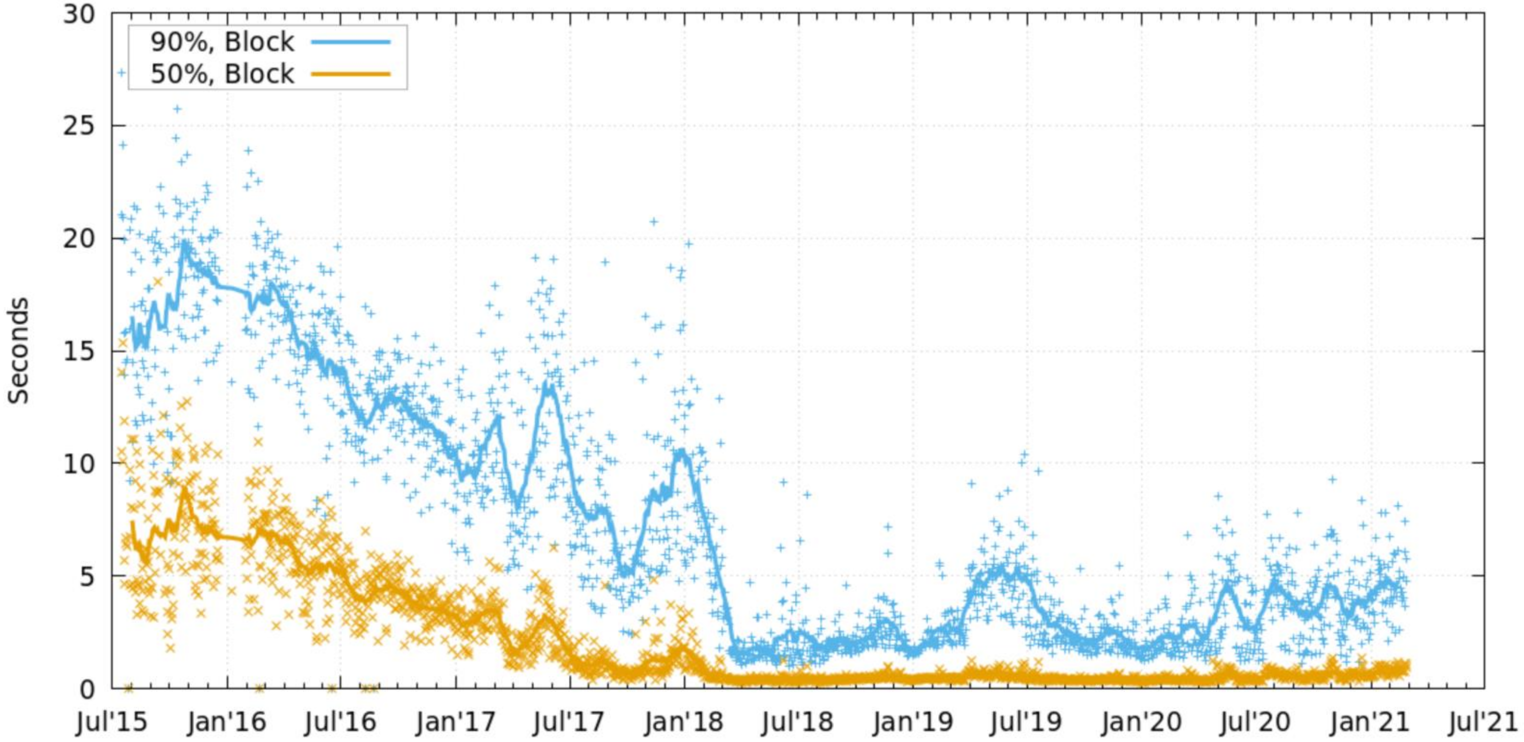
- Network delay (Δ).
  - Mainly stems from its multi-hop broadcast and block propagation mechanism.



Bitcoin Cash's Block Propagation Time



Bitcoin Cash's Block Propagation Time



Real World Specs

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 plugin $\Gamma = 1.88$.
- In order to achieve desired ledger properties, the smoothing factor m should be much larger (approximately several years) to get the ideal ledger properties.

Future/Ongoing Work

- Non-monotonically increasing timestamps
  - In Bitcoin/Bitcoin Cash, the timestamp of a block should be larger than the medium of the last 11 blocks.
  - This work assumes monotonically increasing timestamps.
- Adaptive adversaries

TEXAS A&M UNIVERSITY

Engineering