Analyzing and Improving Proof-of-Work Consensus Protocols

Public Defense
Ren Zhang
Advisor: Prof. Dr. Ir. Bart Preneel

1

History

June 2014,
I Was Lost

Possible next topics:

- Searchable symmetric encryption
- Bitcoin
- Secure logging
- Privacy-preserving surveillance

Claudia and/or Danny:

The best time to start has passed, but the topic may last for a few more years---enough to get a Ph. D.

Bitcoin's Price

2014 to 2019

10000 btc

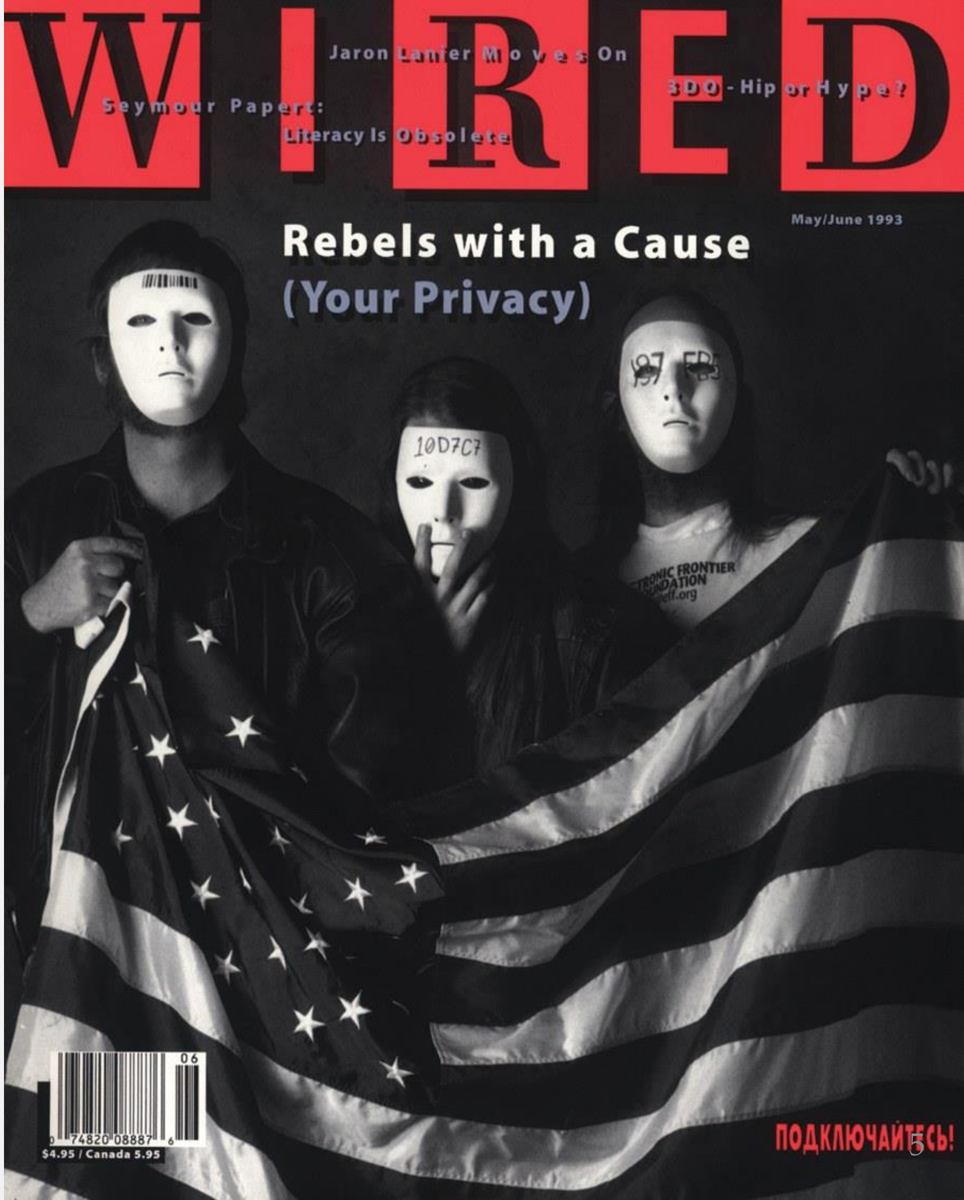
= 2 Papa John's pizzas (May 22, 2010)

= 200 million US dollars (Dec. 17, 2017)

Cypherpunk Movement

We the Cypherpunks are dedicated to building ***anonymous*** *systems. We are defending our privacy with cryptography, with anonymous mail forwarding systems, with digital signatures, and with* ***electronic money****.*

Eric Hughes,
A Cypherpunk's Manifesto
Mar. 9, 1993



Challenges for Satoshi

To run a digital currency…

1. How to remember who has how much money?
2. How to prevent Alice from spending Bob's money?
3. Who controls the money supply?

… for cypherpunks:

- **Open**: nodes dynamically join/leave
- **Decentralized**: no trusted party
- **Pseudonymous**, if not anonymous: no identity

Challenges for Satoshi

To run a digital currency…

1. How to remember who has how much money?
2. How to prevent Alice from spending Bob's money?
3. Who controls the money supply?

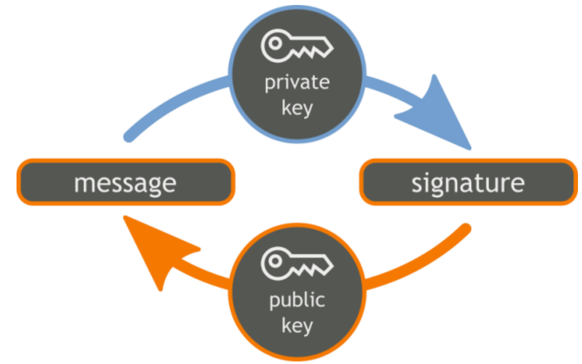
Satoshi's answers

1. Record all transactions in a public ledger
2. Transactions must be signed by the sender

Digital Signature (1975)

2. How to prevent Alice from spending Bob's money?

- Each **account** in the ledger is a public key
- Each **transaction** is signed by the private key of the sender



A digital signature verifies

- The signer's identity
- The signer's approval
- The integrity of the message

Challenges for Satoshi

To run a digital currency…

1. How to remember who has how much money?
2. How to prevent Alice from spending Bob's money?
3. Who controls the money supply?
4. **How to make sure the ledger is append-only?**

Satoshi's answers

1. Record all transactions in a public ledger
2. Transactions must be signed by the sender

4. The ledger is maintained via …

2

Nakamoto Consensus

PoW (1992)
Hashcash (1997)

Proof of work (PoW): a computational task (puzzle) that is

- Challenge-specific
- Easy to generate/verify
- Moderately hard to solve

Hashcash: a simple PoW puzzle:

Find x such that $\mathbf{H}(\text{challenge}|x)<d$

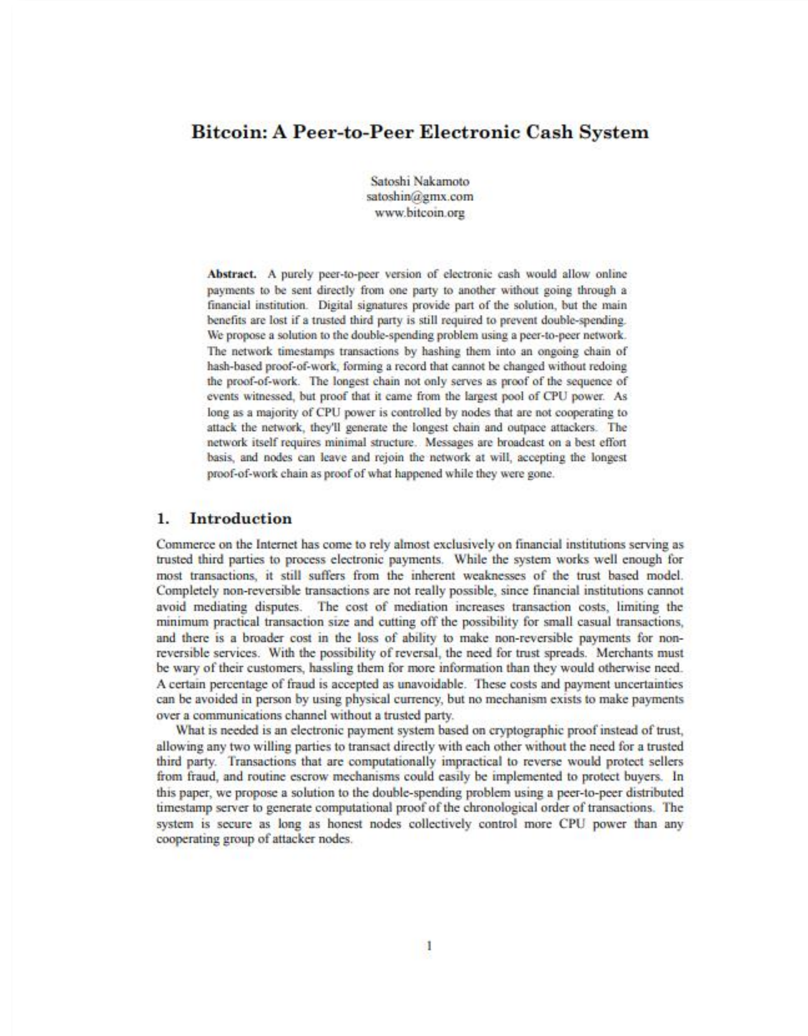
- The only way to solve the puzzle is to enumerate x values

Hash function $\mathbf{H}$:

- Easy to compute $\mathbf{H}(x)$ from x
- Infeasible to compute x from $\mathbf{H}(x)$

Bitcoin (2008)

- On Oct. 13, 2008, Satoshi Nakamoto sent a paper "Bitcoin: A peer-to-peer electronic cash system" to a cypherpunk mailing list
- Bitcoin was launched on Jan. 3, 2009
- Now Bitcoin confirms 300,000 transactions / day
- The paper is 8281 times



Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for non-reversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

1

Challenges for Satoshi

3. **Who controls the money supply?**

Convention enforced by the software

4. **How to make sure the ledger is append-only?**

Via Nakamoto Consensus

5. **How to store the ledger?**

Everyone who runs the software keeps a copy

BitcoinCore

MEET "EVERYONE"

Reachable nodes as of Wed Apr 03 2019 14:09:46 GMT+0200 (Central European Summer Time).

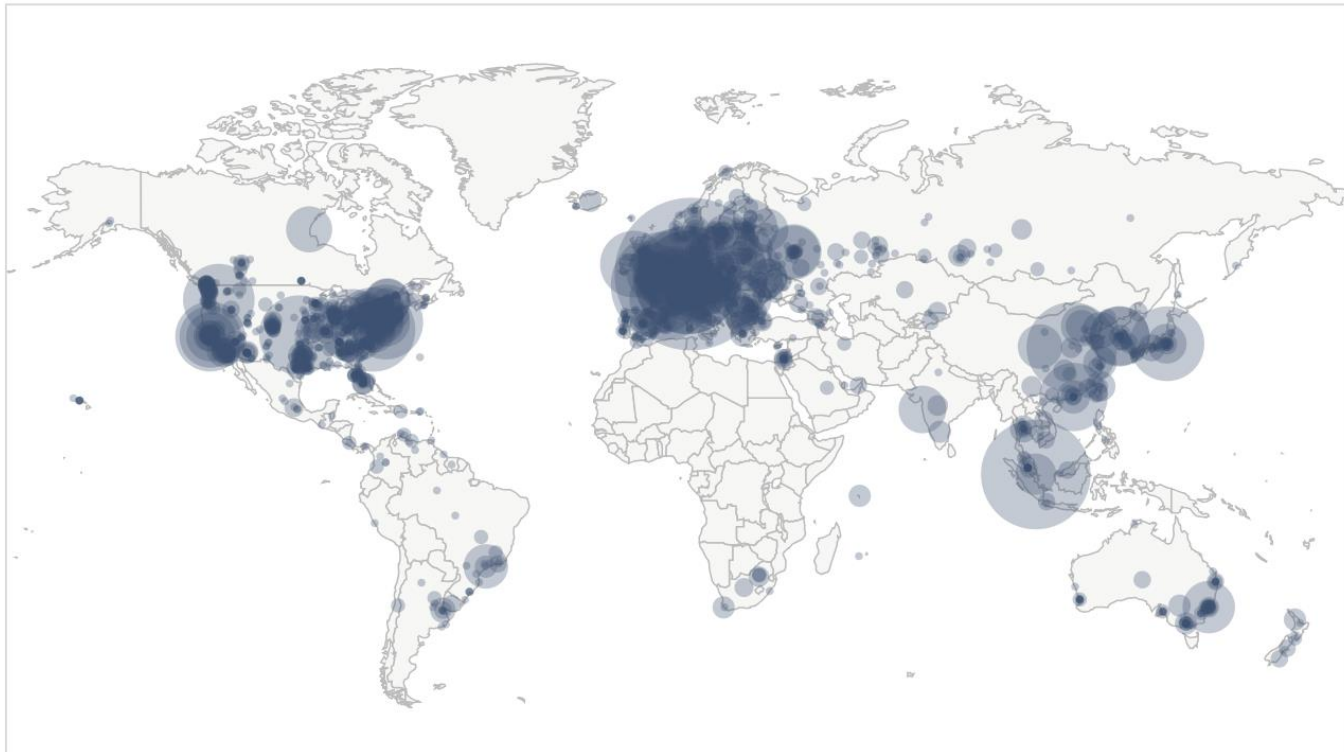
10136 NODES

24-hour charts »

Top 10 countries with their respective number of reachable nodes are as follow.

RANK	COUNTRY	NODES
1	United States	2532 (24.98%)
2	Germany	1925 (18.99%)
3	France	628 (6.20%)
4	Netherlands	526 (5.19%)
5	Canada	360 (3.55%)
6	China	354 (3.49%)
7	United Kingdom	352 (3.47%)
8	Singapore	336 (3.31%)
9	Russian Federation	275 (2.71%)
10	n/a	263 (2.59%)

More (99) »



From bitnodes.earn.com

Challenges for Satoshi

4. **How to make sure the ledger is append-only?**

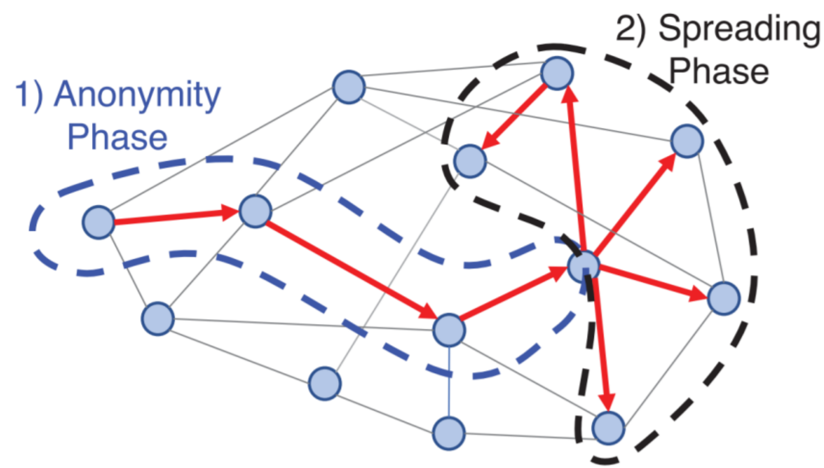
Via Nakamoto Consensus

5. **How to store the ledger?**

Everyone who runs the software

6. **How to notify the others about a transaction?**

Broadcast it to the "everyone" via gossip (5 sec to 50% of nodes, 15 sec to 90%)



Venkatakrishnan, Shaileshh Bojja, Giulia Fanti, and Pramod Viswanath. "[Dandelion: Redesigning the bitcoin network for anonymity.]"

NC's Goals and Challenges

Goals

- Everyone agrees on the same ledger
- The ledger is append-only

Challenges

- Open: nodes dynamically join/leave
- Decentralized: no trusted party
- Pseudonymous: no identity

A Simplified Protocol

1. Each **miner** collects new transactions into a block
2. In each round **a chosen miner** broadcasts its block
3. Other nodes accept the block only if all transactions in it are valid
4. Miners who accept the block will include its hash in the next block they create

Questions:

7. **How to choose that miner?**
8. **How to choose among conflicting histories?**

To Choose the Miner

- Every miner works on finding the solution "nonce" to the following puzzle:

H(transactions, prev_block, nonce)<target

- The target is dynamically adjusted so that on average a block is found every 10 min
- Whoever finds the solution first broadcasts the block

Property: the probability that a miner is selected is proportional to its computing power

Questions:

7. ~~**How to choose that miner?**~~
8. **How to choose among conflicting histories?**
9. **Why would people want to be that miner?**

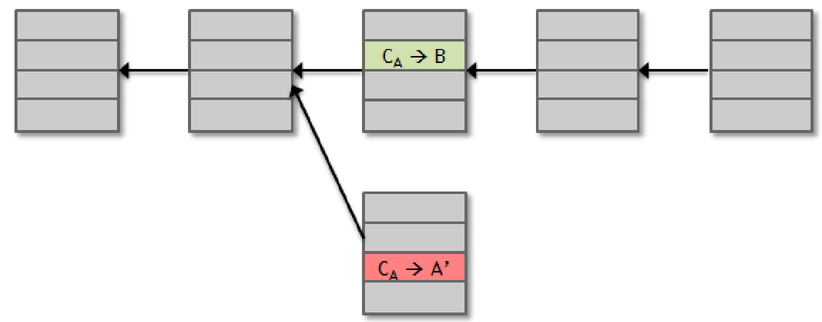
To Choose the History

When multiple chains are mined with the same "previous block":

- Choose the chain that is **most computationally challenging to produce** (usually the longest);

  two blocks w/ target 10
  = one block w/ target 5

- Or, in a tie: **the first received**



Questions:

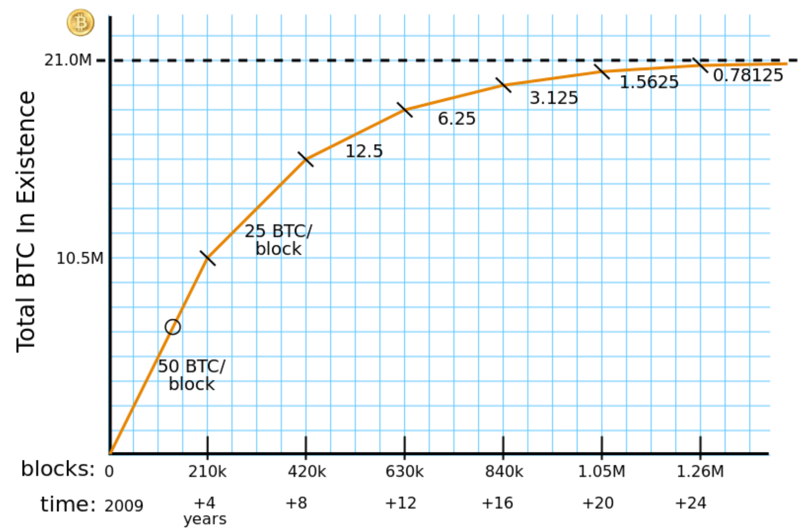
8. ~~**How to choose among conflicting histories?**~~

9. **Why would people want to be that miner?**

MINING: INCENTIVE

To encourage mining:

- A coinbase tx in a block has no input and issues a fixed amount of mining reward (new btc) to the miner
- Each transaction submits a small **transaction fee** to the miner (think of it as a tip)



The reward halves every four years
Now: 18.1M/21M mined

Questions:

9. ~~**Why would people want to be that miner?**~~

Data from BTC.com

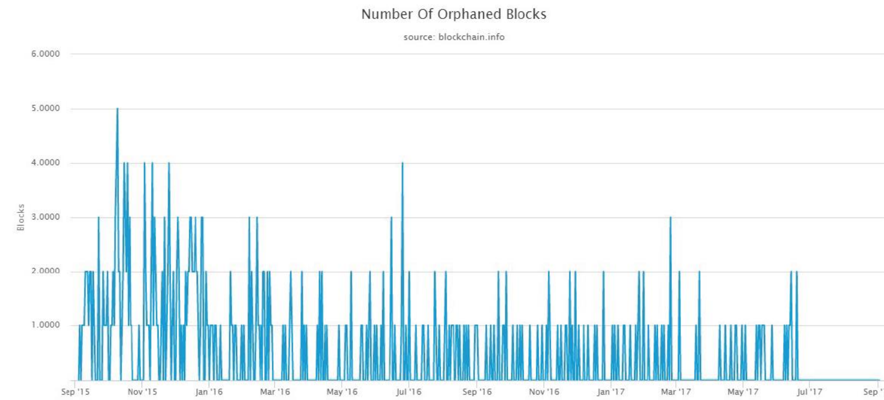
Naturally Orphaned Blocks

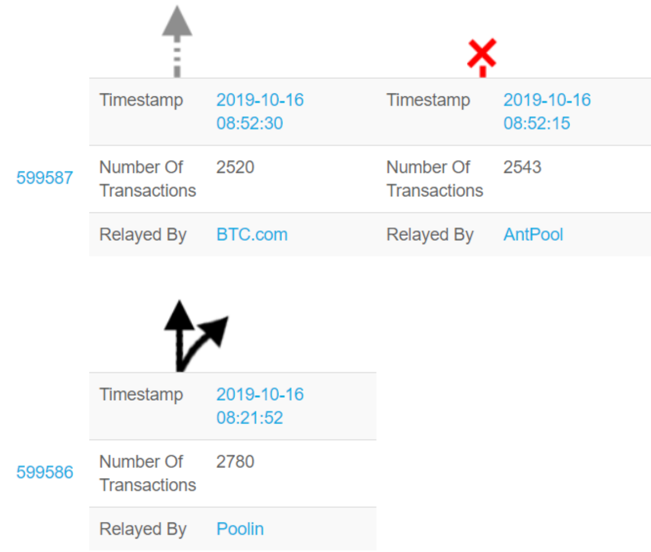
Orphaned blocks: blocks do not end up in the main chain, can be

- Natural: honest blocks mined during other blocks' propagation
- Attacker blocks

Naturally orphaned block frequency:

- Before 2017.7, roughly one per day
- 6 in 2018; 4 in 2019, thanks to the **compact block** mechanism

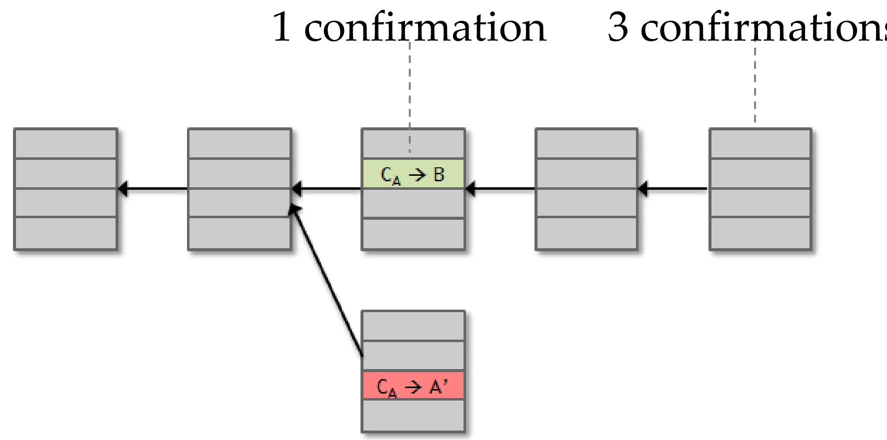




Data from blockchain.com

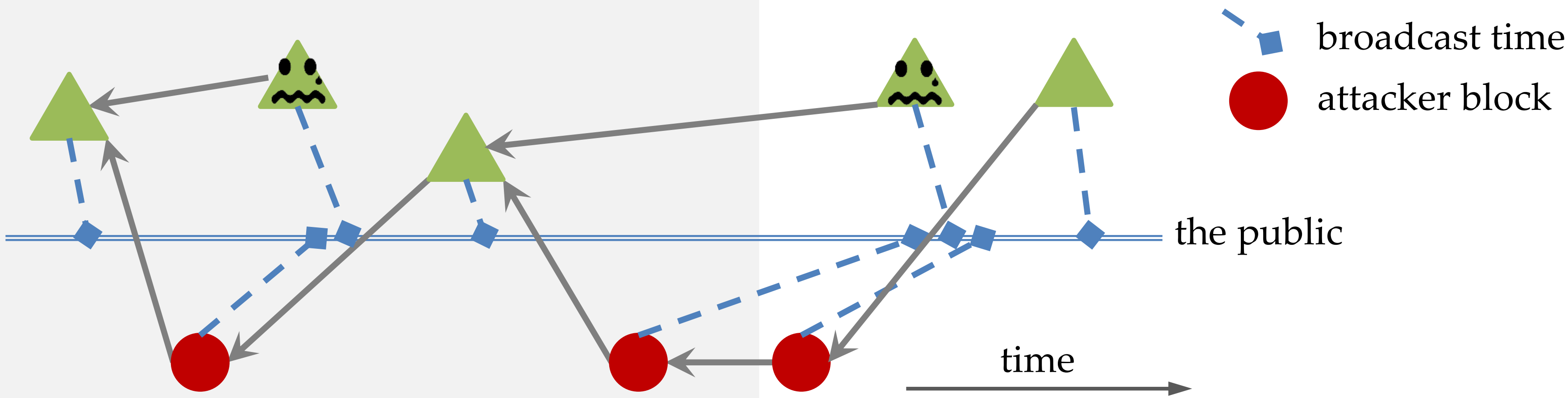
Security Properties

- Irreversibility is probabilistic: one can never be fully sure that a transaction is irreversible
- Double-spending probability **decreases exponentially** with # confirmations



- The attacker with 1/3 total mining power may find three blocks in a row and invalidate the green transaction with 1/27 probability
- A >50% attacker can arbitrarily reverse history

The Imperfect Chain Quality



Low Throughput

Alipay™
VISA
bitcoin
ethereum

Transactions per second

- 12,000 average
- 256,000 peak
- 2,000 average, 56,000 peak
- ≈ 5 (≈1 MB / 10 min)
- ≈ 15 (≈10^7 gas / 14 sec)

Alternative PoW Protocols

GOSHAWK

SUBCHAINS

BYZCOIN

PUBLISH OR PERISH

TORTOISE AND HARES

BAHACK'S IDEA

BITCOIN-NG (AETERNITY, WAVES)

BITCOIN'S NAKAMOTO CONSENSUS

ETHEREUM POW

DECOR+ (ROOTSTOCK)

GHOST-DAG

SPECTRE

CHAINWEB

FRUITCHAINS

PHANTOM

BOBTAIL

THE INCLUSIVE PROTOCOL

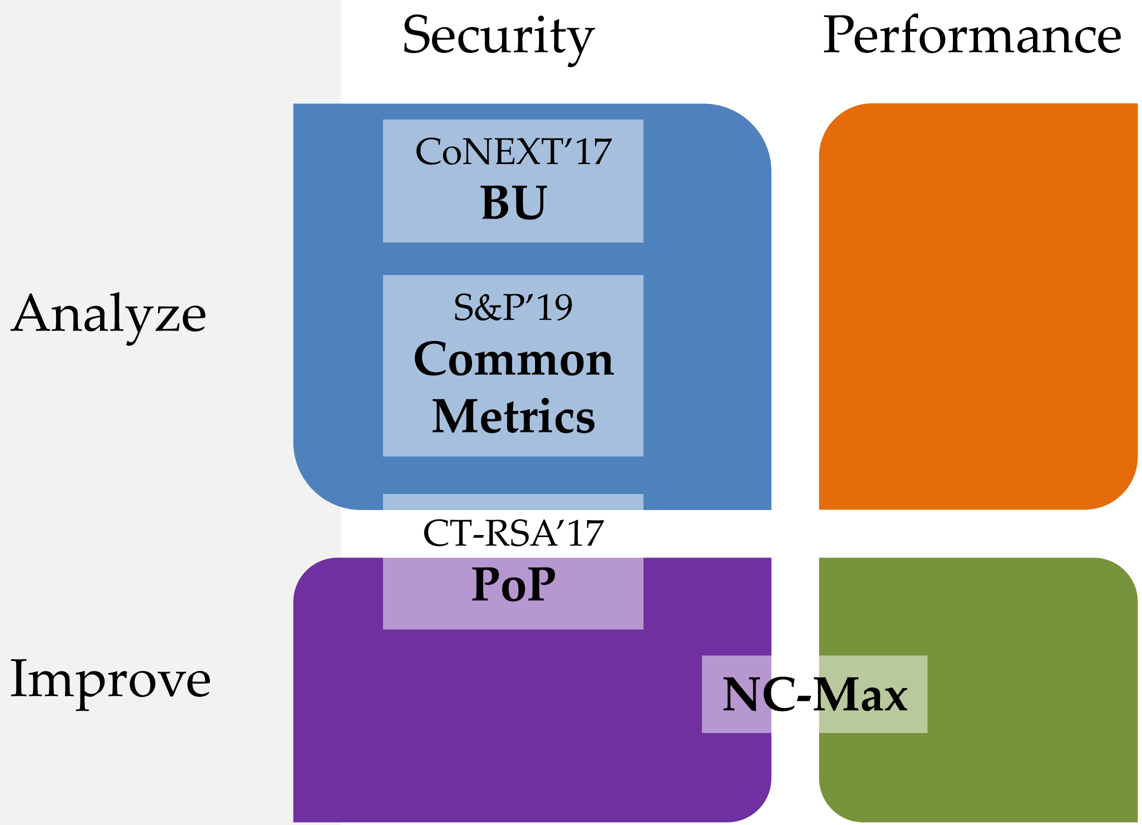
GHOST

CONFLUX

3

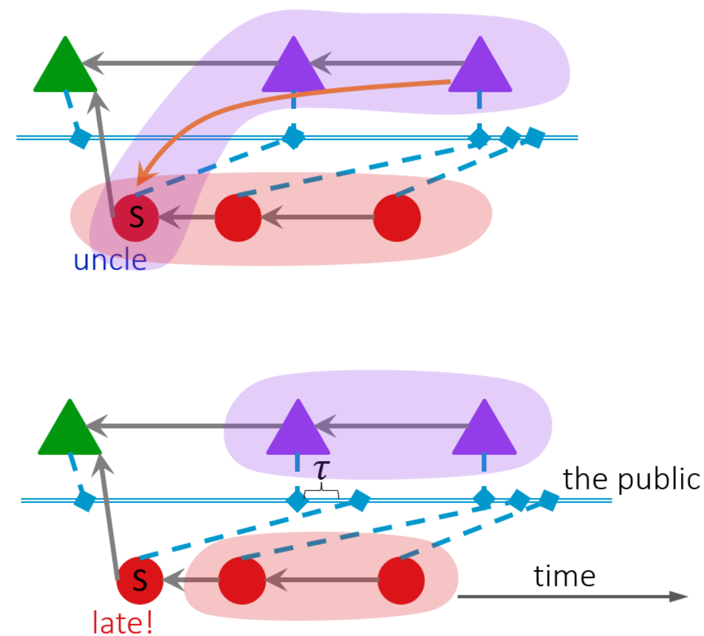
My Contributions

My Contributions



Publish or Perish: a Defense against the Chain Quality Attack

- Highlighted the origin of this attack: Bitcoin's high partition tolerance
- Proposed a defense that is
  - **Backward-compatible**: eventually converges to the longest chain; no need to change the reward scheme or the block data structure
  - **Effective**: outperforms existing backward-compatible defenses



Analyzing BU Mining Protocol

Bitcoin Unlimited:

- A Bitcoin scaling proposal that received the largest mining power support (40%) until late June, 2017

How to scale?

- Miners decide their own block size → No block validity consensus (BVC)

Secure?

- Attacks "cost the attacker far more than the victim"


BITCOIN MAGAZINE

News Guides Price Explorer Events Store Search

Aaron Van Wirdum

Aaron van Wirdum is interested in technology and how it affects social and political structures. He has been covering Bitcoin since 2013, focusing on privacy, scalability and more. Hodls BTC.

March 29, 2017

MINING

Bitcoin Unlimited Miners May Be Preparing a 51% Attack on Bitcoin

Bitcoin News › Articles › Bitcoin Unlimited Miners May Be Preparing a 51% Attack on Bitcoin

What We Did: Compare BU and Bitcoin

Incentive models \ Security claims	BU is secure when BVC is absent	BVC will emerge
Compliant & Profit-Driven	😭	😭
Non-Compliant & Profit-Driven	😭	Not meaningful
Non-Profit-Driven	😭	

Impact


Research Finds Design Flaws in Scaling Proposal Bitcoin Unlimited

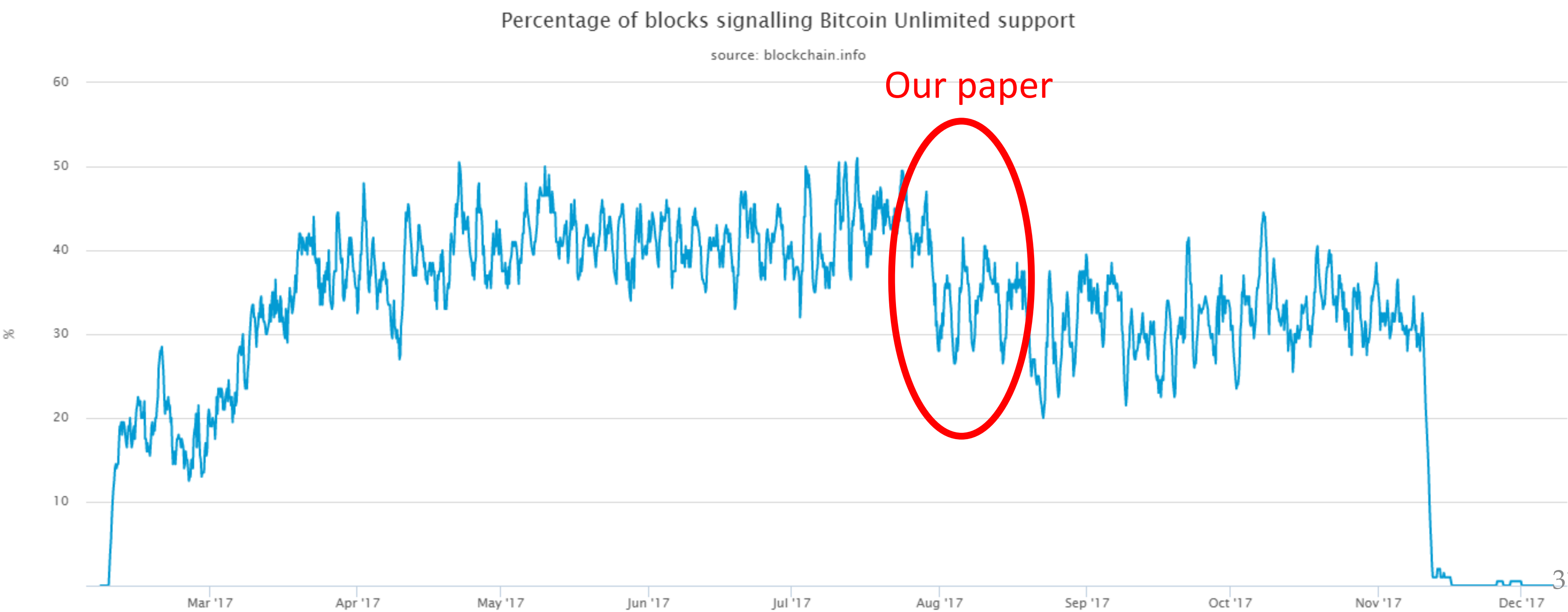
6 13

Pete Rizzo

Jul 19, 2017 at 14:45 UTC

NEWS

A new research paper from international analyst group IMEC has found that changes to bitcoin proposed by a software implementation called Bitcoin Unlimited would "magnify the effectiveness" of attacks on the network.



4

Common Metrics

The Story Behind

- PoP only **mitigates** the chain quality attack
- So I designed, modeled and evaluated dozens of ideas to improve NC, but none is perfect
- But these flawed ideas are keep being published with none or partial security evaluation
- I think people needs to be informed

Protocol	**Citations**
Fruitchains	131
Bitcoin-NG	631
Byzcoin	321
Subchains	19
DECOR+	3

Alternative PoW Protocols

GOSHAWK

SUBCHAINS

BYZCOIN PUBLISH OR PERISH

TORTOISE AND HARES

BAHACK'S IDEA BITCOIN-NG (AETERNITY, WAVES)

BITCOIN'S NAKAMOTO CONSENSUS

ETHEREUM POW DECOR+ (ROOTSTOCK)

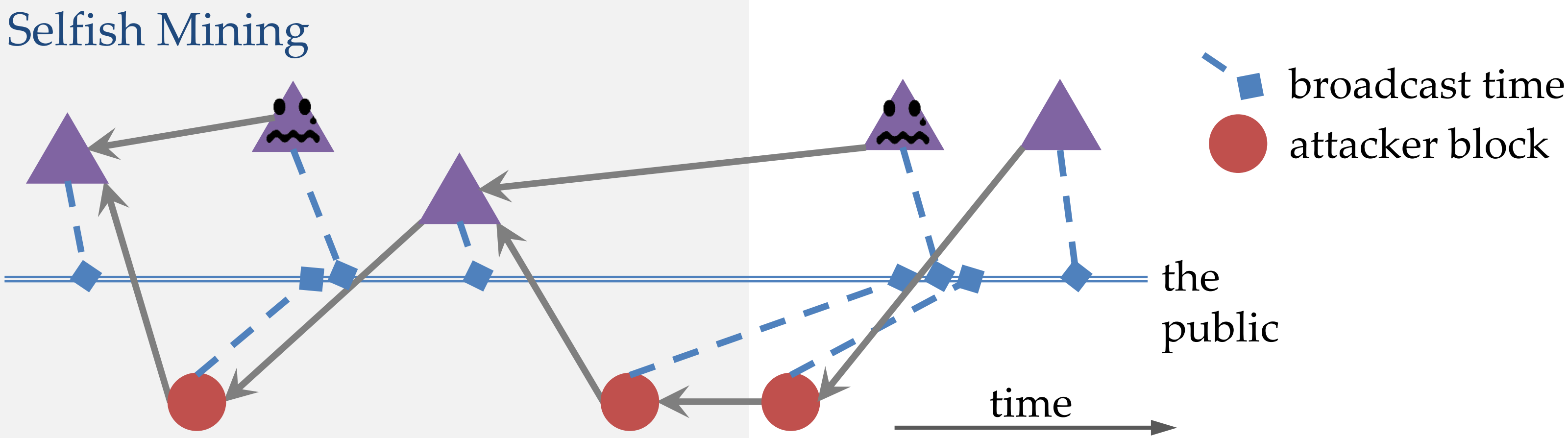
GHOST-DAG SPECTRE CHAINWEB

FRUITCHAINS PHANTOM BOBTAIL

THE INCLUSIVE PROTOCOL GHOST CONFLUX

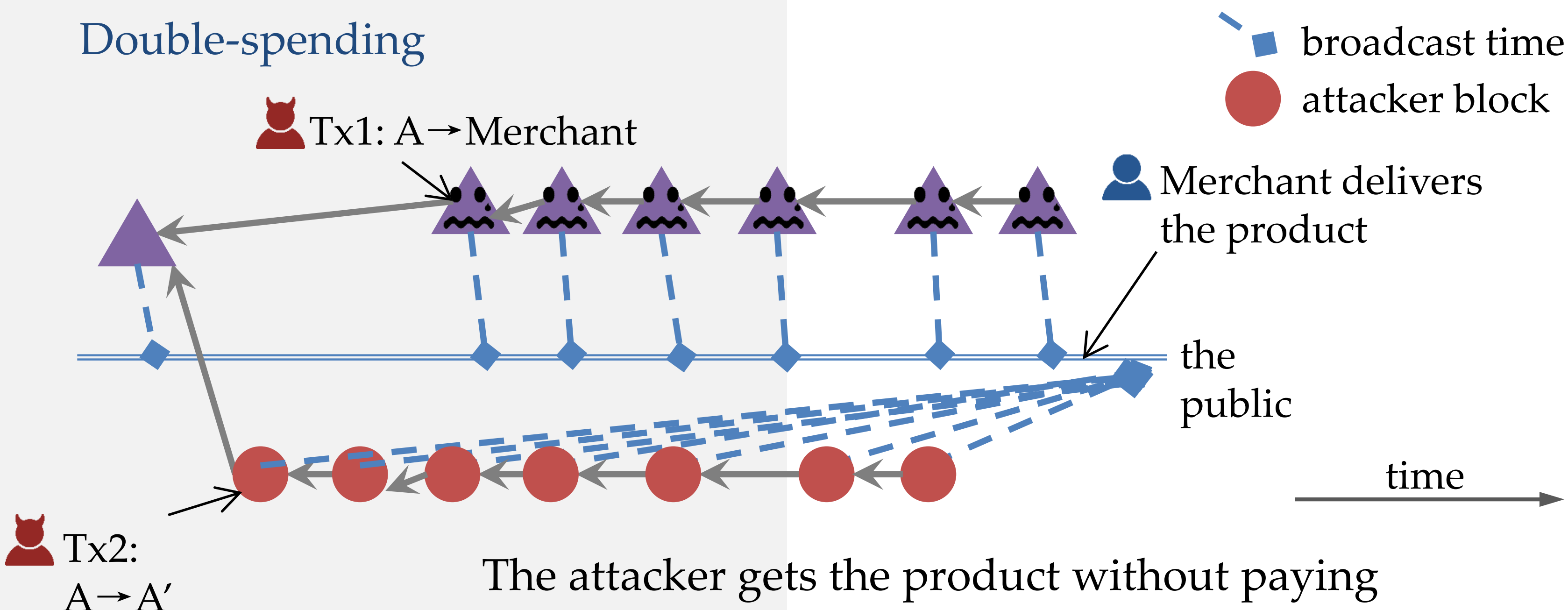

?
bitcoin
2
1

The Imperfect Chain Quality → Three Attacks



The attacker gains unfair block rewards; rational miners would join the attacker, which damages decentralization

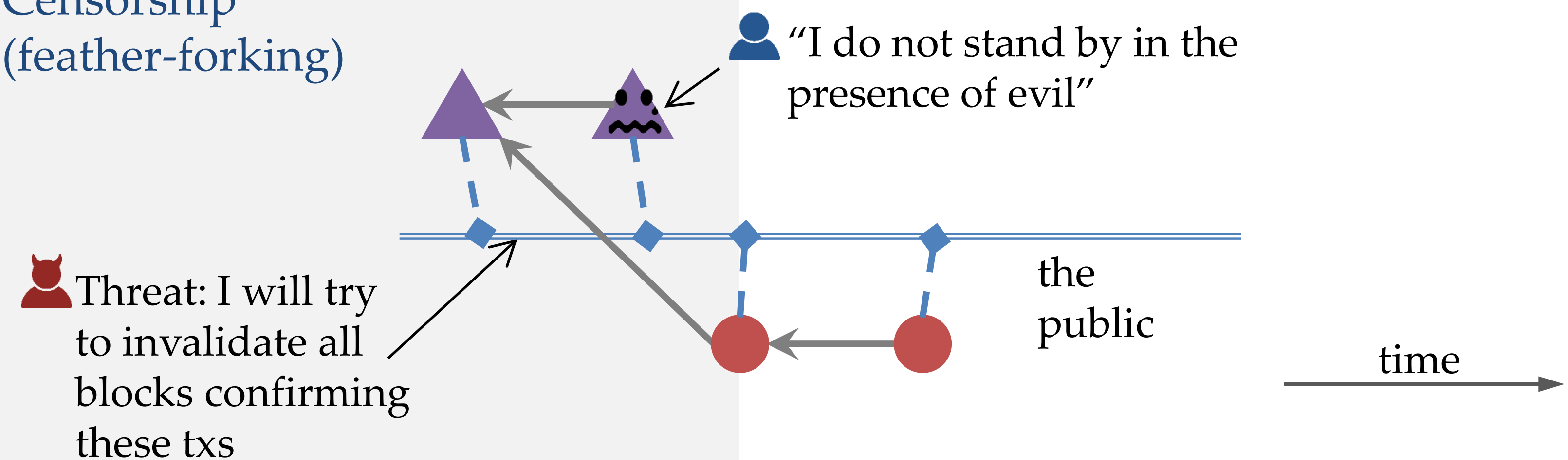
The Imperfect Chain Quality → Three Attacks



The attacker gets the product without paying for it

The Imperfect Chain Quality → Three Attacks

Censorship
(feather-forking)



Rational choice: join the attacker in censorship
The attacker becomes a *de facto* owner

Our Evaluation Framework: Four Metrics

A better-than-NC protocol needs to

- Achieve better chain quality ❶❷
- **Or** resist better against all three attacks:
  - Selfish mining ☞ incentive compatibility ❶
  - Double-spending ☞ subversion gain ❶
  - Censorship ☞ censorship susceptibility ❷

❶ profit-driven adversary

❷ byzantine adversary

BETTER-THAN-NC CANDIDATES

Better-chain-quality protocols:
"I can raise the chain quality"

- UTB: Ethereum PoW, Bitcoin-NG (Aeternity, Waves)
- SHTB: DECOR+ (Rootstock)
- UDTB: Byzcoin, Omniledger
- Publish or Perish

In the paper

Attack-resistant protocols:
"I don't need to raise the chain quality, I can defend against the attacks"

- Reward-all ("compensate the losers"): Fruitchains, Ethereum PoW, Inclusive, SPECTRE, PHANTOM, …
- Punishment ("fine all suspects"): DECOR+, Bahack's idea
- Reward-lucky (content-based reward): Subchains, Bobtail

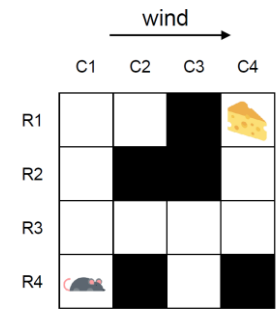
Method: Markov Decision Process

An MDP models

- A strategic player's behavior
- In a partly stochastic environment

It can solve

- The optimal strategy
- That maximizes the utility



A strategy player: the malicious miner; all other miners follow the protocol

Behavior: when to publish how many blocks, which chain to mine on

Partly stochastic: the next block may be mined by the attacker, by an honest miner on an attacker block, or by an honest miner on an honest block

The utility: more block rewards, more double-spending rewards, or more orphaned honest blocks


The
Evalu
Results

Simplified Results

😀 better
😕 it depends
😟 worse

"Better-chain-quality"	Chain Quality
Uniform tie-breaking	😟
Smallest-hash tie-breaking	😟
Unpredictable deterministic tie-breaking	😟
Publish or perish	😕

"Attack-resistant"	Incentive compatibility	Subversion gain	Censorship susceptibility
Reward-all ☞Fruitchains	😟	😟	😀
Punishment ☞Reward-splitting	😀	😀	😟
Reward-lucky ☞Subchains	😟	😟	😟

Simplified Results

😀 better
😕 it depends
😟 worse

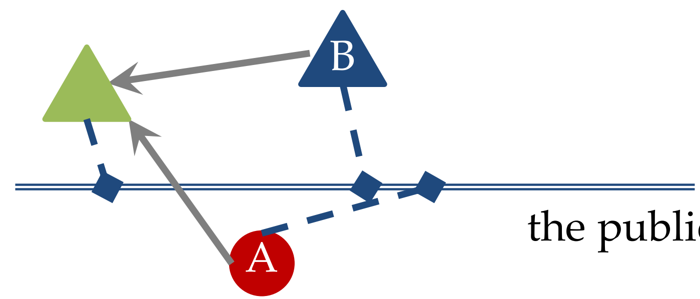
"Better-chain-quality"	**Chain Quality**
Uniform tie-breaking	😟
Smallest-hash tie-breaking	😟
Unpredictable deterministic tie-breaking	😟
Publish or perish	😕

"Attack-resistant"	**Incentive compa-tibility**	**Subver-sion gain**	**Censorship suscepti-bility**
Reward-all ☞Fruitchains	😟	😟	😀
Punishment ☞Reward-splitting	😀	😀	😟
Reward-lucky ☞Subchains	😟	😟	😟

Better-Chain-Quality: SHTB & UDTB

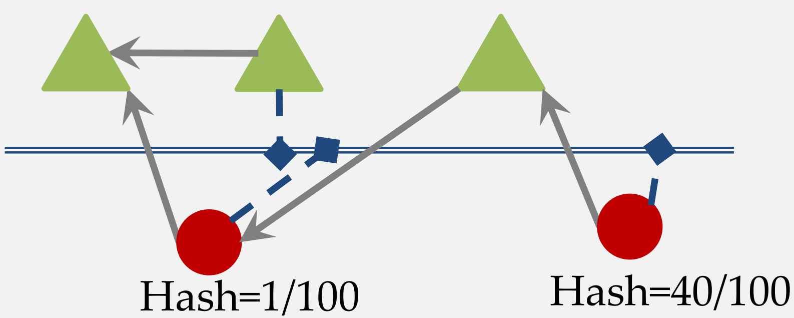
In a tie:

- **NC**: mine on the first-received block
- **Smallest-hash tie-breaking**: Compare H(A) and H(B): mine on the smallest hash
- **Unpredictable deterministic tie-breaking**: using a deterministic PRF; compare, e.g., H(A⊕B, A) and H(A⊕B, B), mine on the smaller

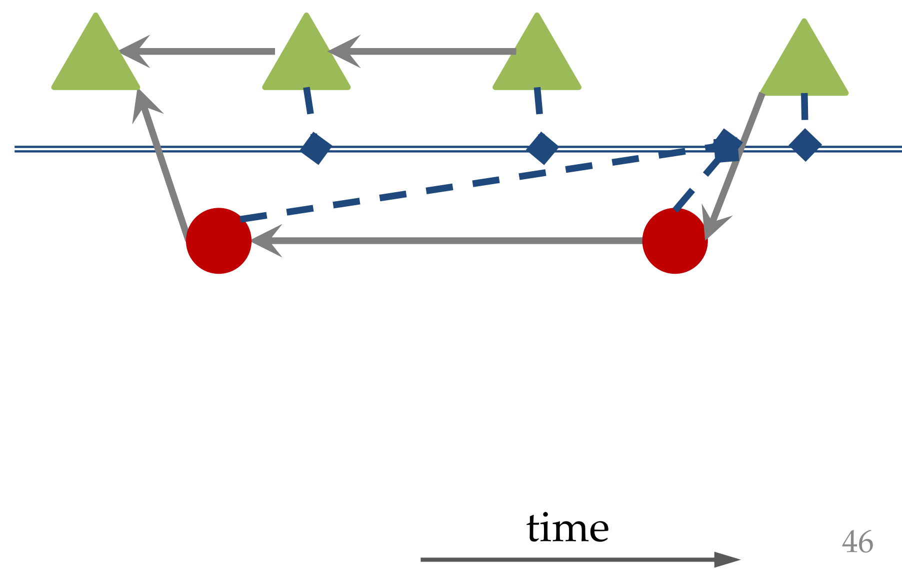


Chain Quality is Worse

SHTB: "selective block publishing"
+ "catch up from behind"



UDTB: "catch up from behind"



Insight: Information Asymmetry

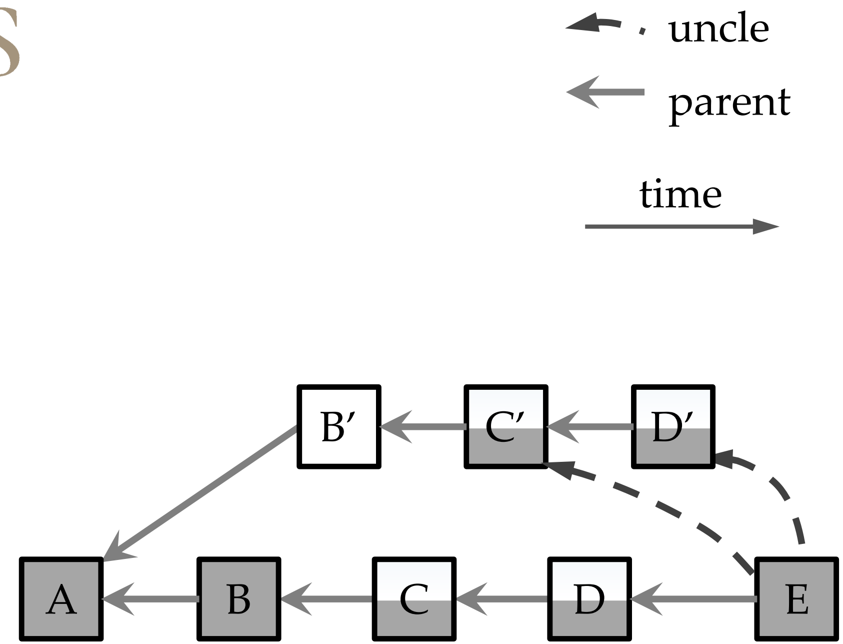
The attacker acts on all info:

1. Local: secret blocks, my system clock
2. Public and *a posteriori* verifiable: public block content
3. Public but not *a posteriori* verifiable: block publishing time, whether the network is partition
4. Network condition: latency, propagation advantage

Compliant miners only act on "2."

Attack-Resistant ☞ Punishment: RS

- Blocks refer to orphaned blocks as uncles
- An uncle is valid if height(host)-height(uncle) < TimeOut (B′ is hopeless if TimeOut = 3)
- Each block reward is **evenly split** among competing block & uncles of the same height



RS Results

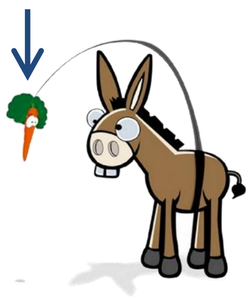
Incentive compatibility & Subversion Gain😃

- Punishment works for profit-seekers!

When attacker controls 10% mining power, 6-conf., subversion bounty =

- 102 block rewards in NC
- 346 in RS
- 0 in Fruitchains

Subversion bounty: Min double-spending reward to incentivize double-spending attack attempts



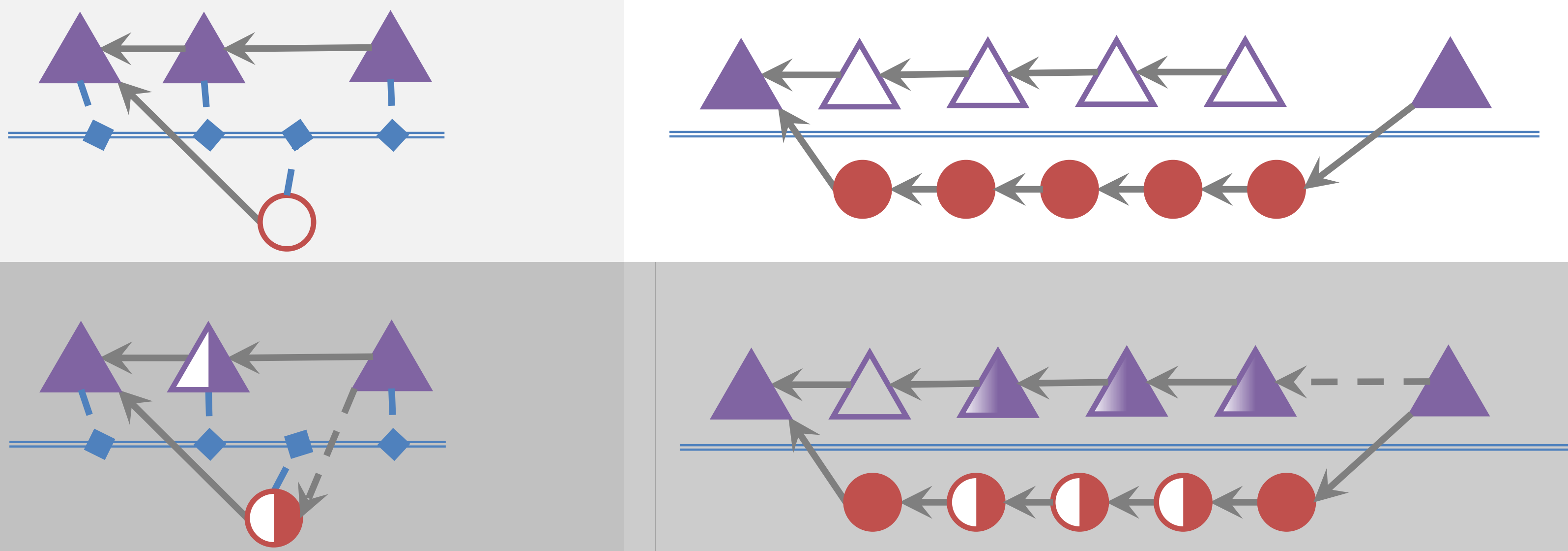
Censorship Susceptibility of RS

Weak attackers 😟

Strong attackers 😀

In NC

In RS



Insight: Rewards Don't Solve the Attacks

A dilemma: "Rewarding the bad vs. punishing the good"

- Reward all -> no risk to double-spend
- Punish -> aid censorship
- Reward lucky -> lucky≠good

A common mistake

- Attackers have different incentives; no reward scheme discourages all of them

Discussion

- No protocol comprehensively outperforms NC

What not to

- Designing protocols too complicated to analyze
- Security analysis
  - against one attack strategy
  - against one attacker incentive
  - with unrealistic parameters

Better chain quality via practical assumptions

- Awareness of network conditions
- Loosely synchronized clock
- Real-world commitments

Better attack resistance via outsourcing liability

- Additional punishment rules
- Solve at layer 2

5

NC-Max

NC-Max: Breaking the Throughput Limit of NC

- Confirmed and eliminated the bottleneck in NC's low throughput
- Dynamically adjusts the throughput base on the network condition
- Proved that selfish mining is not profitable within our new difficulty adjustment mechanism

NC's Throughput Limit

Throughput ↑ :
Block size ↑ , block interval ↓

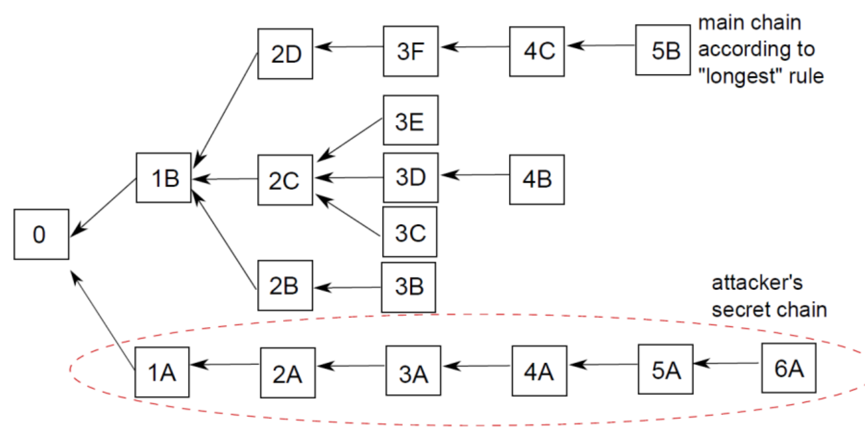
☟

Orphans ↑

☟

Security ↓ , Throughput ↓

Too many orphans are bad for security and performance



How to Break the Throughput Limit

Fresh transactions in a block $\downarrow$

☟

Block propagation delay $\downarrow$

☟

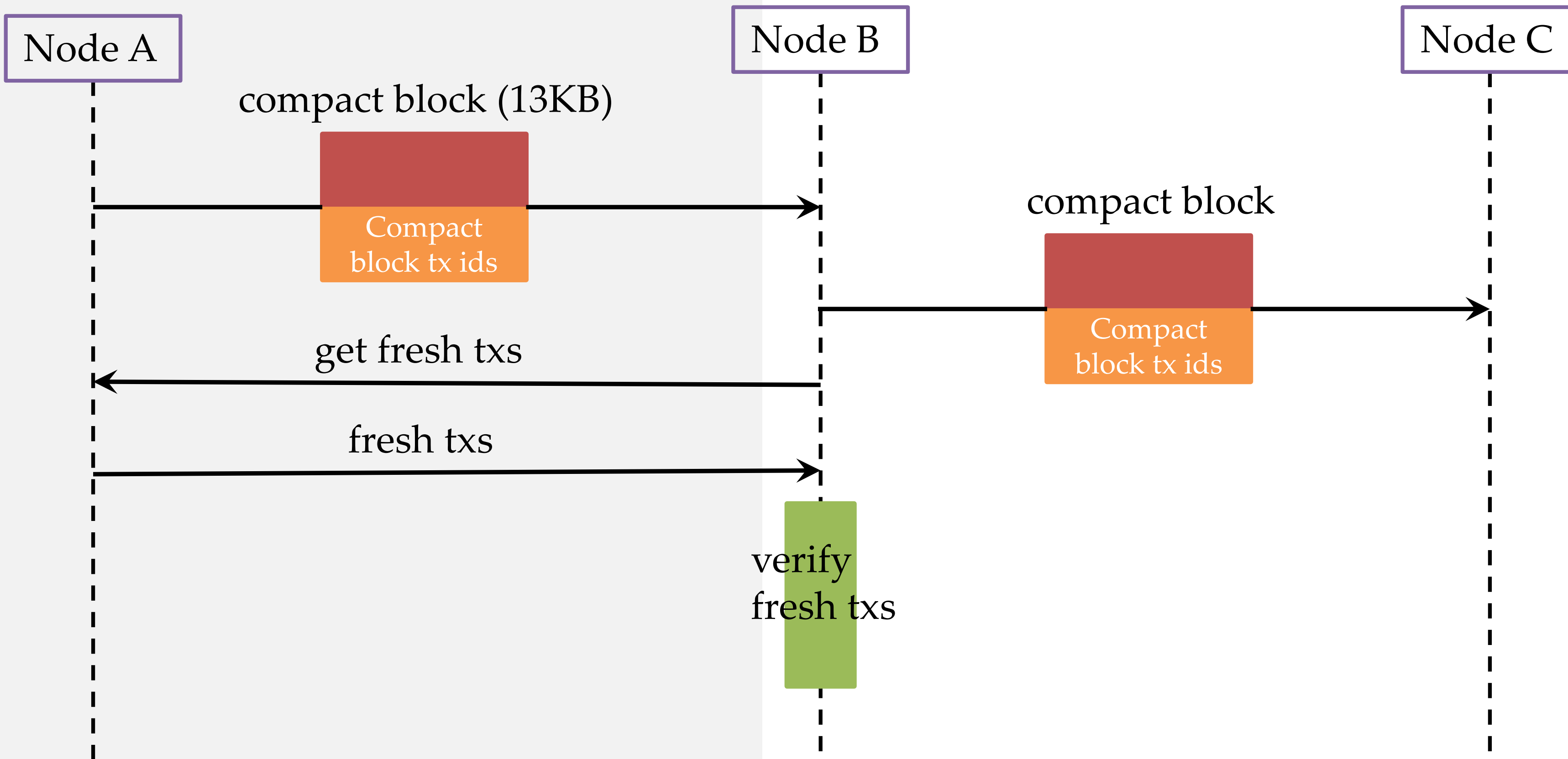
Orphans $\cancel{\uparrow}$ $\downarrow$

☟

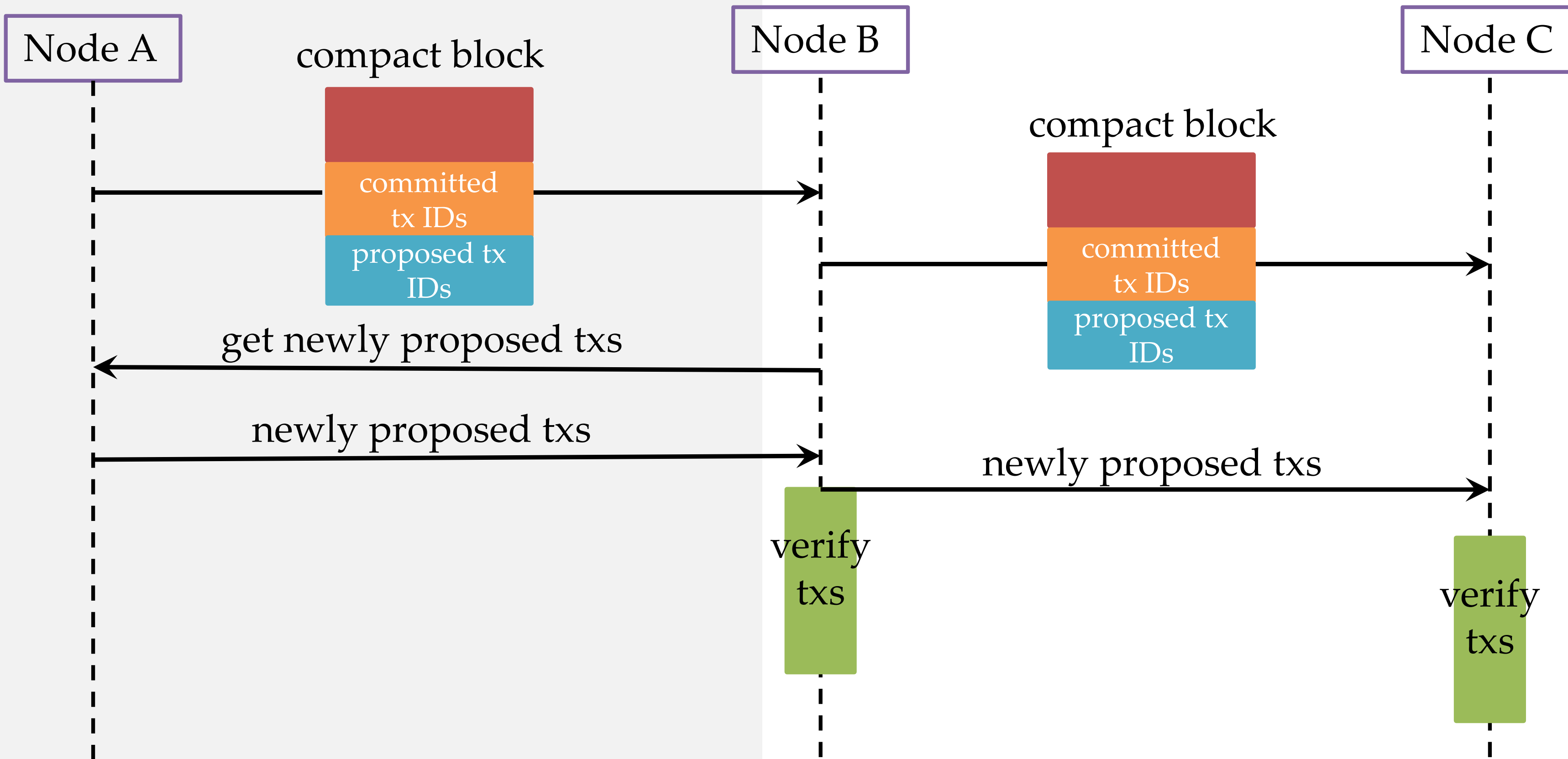
Security $\cancel{\downarrow}$ $\uparrow$, Throughput $\cancel{\downarrow}$ $\uparrow$

Fresh transactions: newly broadcast transactions that have not finished propagating to the network when they are embedded in blocks

Block Propagation (NC)

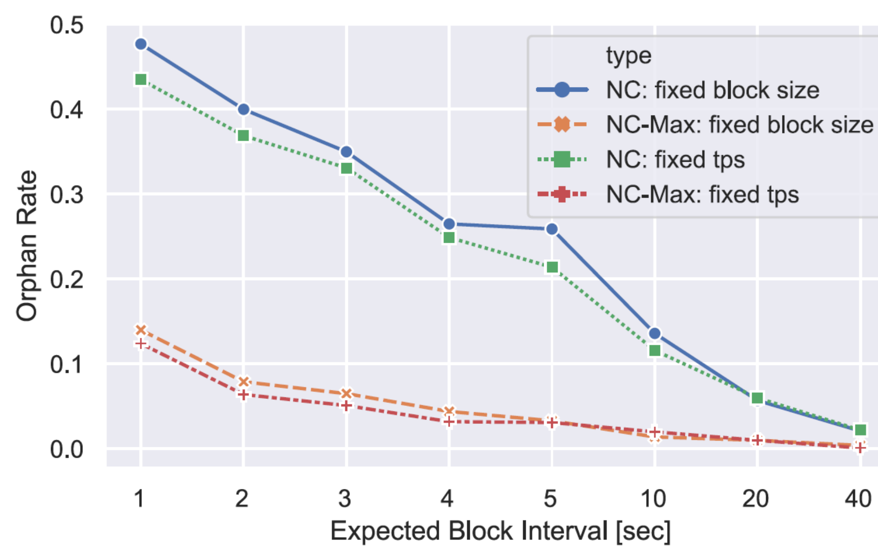


Block Propagation (NC-Max)

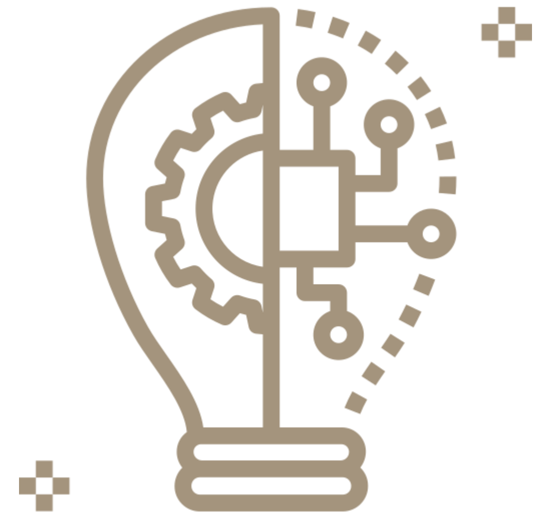


Results

- NC: when block interval = 20 sec, block size = 1 MB, 100 TPS orphan rate = 6%
- NC-Max, same orphan rate:
  - same transaction throughput, block interval = 2 to 3 sec
  - same block size, block interval = 3 to 4 sec (≥ 500 TPS)



6



Conclusion

Conclusion

What we did

- Comprehensively analyzed the security of a broad number of proposals and revealed their vulnerabilities
- Identified the root causes and proposed solutions
- Demonstrated how network-level optimization could improve both security and performance

General Insights

- Simulating one attack is not a security proof; resistant against one attack doesn't infer security
- Analyzing security with AI/game theory is a promising direction
- Negative results are publishable if you collect many
- Performance can only be improved by analysis on the actual bottleneck

Publications

Conferences

1. M. Herrmann, **R. Zhang**, K. Ning, C. Diaz, and B. Preneel. Censorship-resistant and privacy-preserving distributed web search. In *14th IEEE International Conference on Peer-to-Peer Computing (P2P)*. IEEE, Sep. 2014
2. **Ren Zhang** and Bart Preneel. Publish or Perish: A backward-compatible defense against selfish mining in Bitcoin. In *The Cryptographers' Track at the RSA Conference (CT-RSA)*, volume 10159 of *LNCS*, pages 277–292. Springer, February 2017
3. Emad Heydari Beni, Bert Lagaisse, **Ren Zhang**, Danny De Cock, Filipe Beato, and Wouter Joosen. A voucher-based security middleware for secure business process outsourcing. In *Engineering Secure Software and Systems (ESSoS)*, volume 10379 of *LNCS*, pages 19–35. Springer, 2017
4. **Ren Zhang** and Bart Preneel. On the necessity of a prescribed block validity consensus: Analyzing Bitcoin Unlimited mining protocol. In *13th International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, pages 108–119. ACM, December 2017
5. Madhusudan Akash, Iraklis Symeonidis, Mustafa A Mustafa, **Ren Zhang** and Bart Preneel. SC2Share: smart contract for secure car sharing. In *International Conference on Information Systems Security and Privacy (ICISSP)*, pages 163–171. SciTePress, February 2019

Publications

Conferences

6. **Ren Zhang** and Bart Preneel. Lay down the common metrics: Evaluating proof-of-work consensus protocols' security. In *40th IEEE Symposium on Security and Privacy (S&P)*, pages 1190–1207. IEEE, May 2019

7. Vincent Reniers, Yuan Gao, **Ren Zhang**, Paolo Viviani, Akash Madhusudan, Bert Lagaisse, Svetla Nikova, Dimitri Van Landuyt, Riccardo Lombardi, Bart Preneel and Wouter Joosen. Authenticated and Auditable Data Sharing via Smart Contract. To appear in *ACM/SIGAPP Symposium On Applied Computing*

Draft & Submitted

1. **Ren Zhang**, Dingwei Zhang, Quake Wang, Jan Xie, and Bart Preneel. NC-Max: Breaking the throughput limit of Nakamoto Consensus. September 2019

2. Sarah-Louise Justin, **Ren Zhang**, Gunes Acar and Bart Preneel. Short Paper: Monitoring the Bitcoin Network for Malicious Behavior.

Questions