Scaling Bitcoin 2016 "Retarget" • October 9th Milan, Italy

ENHANCING BITCOIN SECURITY AND PERFORMANCE WITH STRONG CONSISTENCY VIA COLLECTIVE SIGNING

Lefteris Kokoris-Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser and Bryan Ford
EPFL

@LefKok

Swiss Federal Institute of Technology Lausanne

Bitcoin Blockchain

- What we have now:
  - Real-time verification is not safe (1 hour of delay)
  - Throughput is low (4 tx/sec)

Byzcoin Blockchain

- What can Byzcoin do:
  - Irrevocable transaction commitment in 20-90 sec
  - Throughput up to 974 TPS
  - Robust against double-spending, eclipsing, selfish mining
  - Light-weight client verification (suitable for mobile phones)

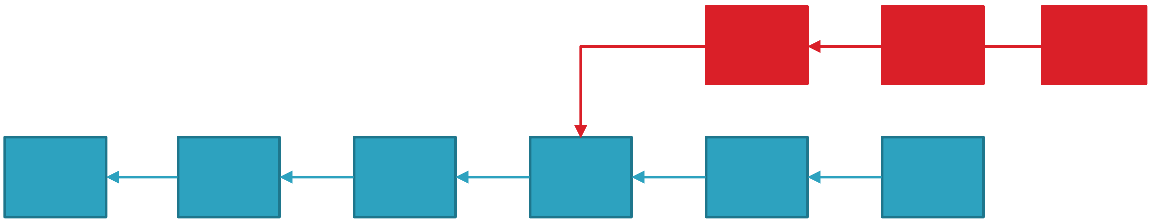
How?

- Use Practical Byzantine Fault Tolerance protocol to provide non-probabilsitic strong consistency
- Use Collective Signing to scale PBFT and decrease latency
- Use PoW to create hybrid permissionless BFT
- Use Bitcoin-NG to increase throughput

Talk Outline

- **Bitcoin and its limitations**
- Strawman design: PBFTCoin
- Opening the consensus group
- From MACs to Collective Signing
- Decoupling transaction verification from leader election
- Performance Evaluation
- Future work and conclusions

The Blockchain



Problem Statement

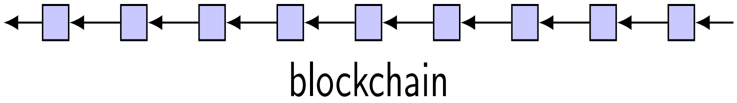
1. In Bitcoin there is no verifiable commitment of the system that a block will persist
   - Clients rely on probabilities to gain confidence.
   - Probability of successful fork-attack decreases exponentially

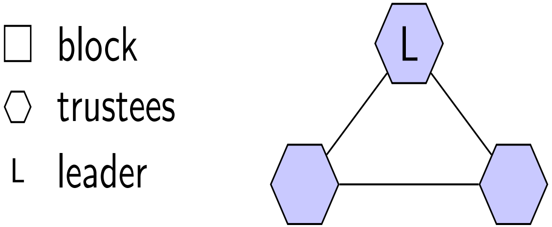
Talk Outline

- Bitcoin and its limitations
- **Strawman design: PBFTCoin**
- Opening the consensus group
- From MACs to Collective Signing
- Decoupling transaction verification from leader election
- Performance Evaluation
- Future work and conclusions

Strawman Design: PBFTCoin

- **3f+1** fixed "trustees" running **PBFT*** to withstand **f** failures
- Non-probabilistic strong consistency
  - Low latency
- No forks/inconsistencies
  - No double-spending

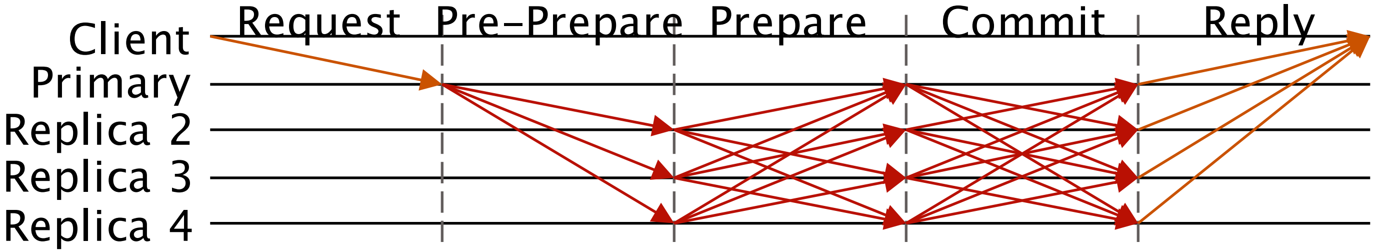




*Practical Byzantine Fault Tolerance [Castro/Liskov]

Strawman Design: PBFTCoin

- Problem: Needs a static consensus group
- Problem: Scalability
  - Dense communication pattern (limits consensus group size)
  - High client-side verification cost (excludes mobile phones/IoT clients)
  - Absence of third-party verifiable proofs (limits number of clients)

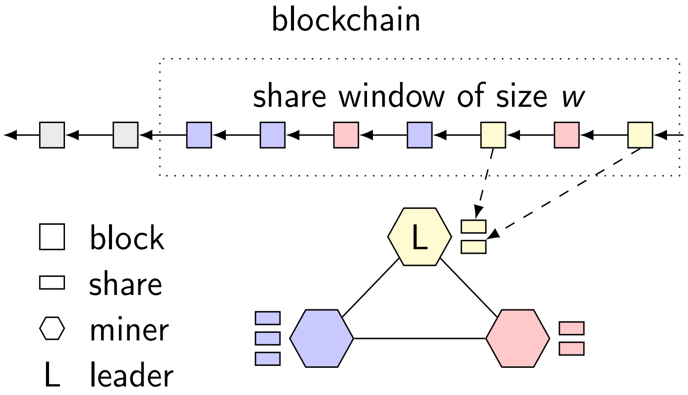


Talk Outline

- Bitcoin and its limitations
- Strawman design: PBFTCoin
- **Opening the consensus group**
- From MACs to Collective Signing
- Decoupling transaction verification from leader election
- Performance Evaluation
- Future work and conclusions

Opening the Consensus Group

- PoW against Sybil attacks
- One share per block
    - % of shares $\propto$ hash-power
- Window mechanism
    - Protect from inactive miners



Talk Outline

- Bitcoin and its limitations
- Strawman design: PBFTCoin
- Opening the consensus group
- **From MACs to Collective Signing**
- Decoupling transaction verification from leader election
- Performance Evaluation
- Future work and conclusions

From MACs to Signing

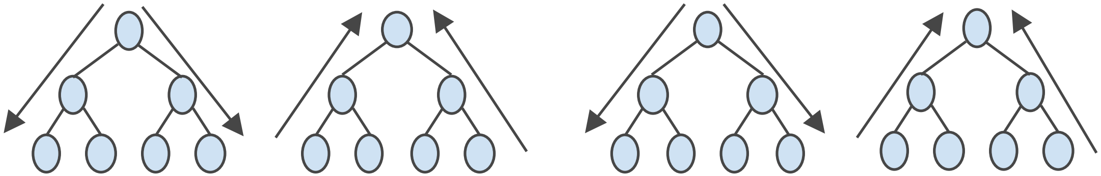
- Substitute MAC-based authentication (symmetric crypto) with public-key cryptography
  - ECDSA provides more efficiency
  - Third-party verifiable
  - PoW Blockchain as PKI
  - Enables sparser communication patterns (ring or star topologies)

From MACs to Collective Signing

- Can we get better communication patterns?
  - Multicast protocols transmit information in sub-linear steps
  - Use trees!!
- Can we allow for lightweight verification?
  - Schnorr multisignatures could be verified in constant time
  - Use signature aggregation!!
- Schnorr multisignatures + communication trees
  = Collective Signing [Syta et all, IEEE S&P '16]

CoSi

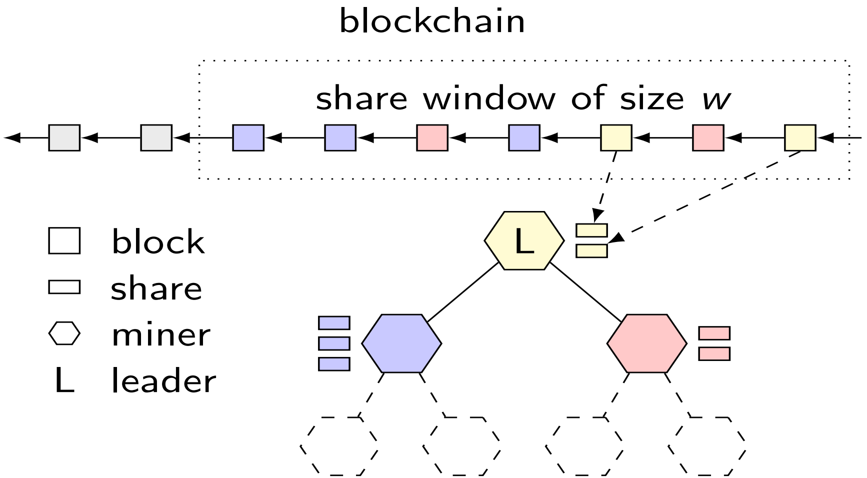
- Efficient collective signature, verifiable as a simple signature
- For the Ed25519 curve
  - 82 bytes instead of 9KB for 144* co-signers
  - 190 bytes instead of 63KB for 1008* co-signers



* Number of ~10-minute blocks in 1-day/week time window

Discussion

- CoSi is not a BFT protocol
- PBFT can be implemented over two subsequent CoSi rounds
  - Prepare round
  - Commit round



Problem Statement

1. In ~~Bitcoin~~ ByzCoin there is ~~no~~ a verifiable commitment of the system that a block will persist
2. Throughput is limited by forks
   - Increasing block size increases fork probability
   - Liveness exacerbation

Talk Outline

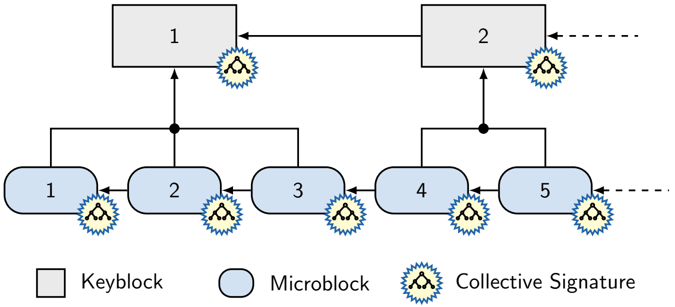
- Bitcoin and its limitations
- Strawman design: PBFTCoin
- Opening the consensus group
- From MACs to Collective Signing
- **Decoupling transaction verification from leader election**
- Performance Evaluation
- Future work and conclusions

Bitcoin-NG [Eyal et all, NSDI ’16]

- Makes the observation that block mining implement two distinct functionalities
  - Transaction verification
  - Leader election
- We enhance Bitcoin-NG with Byzantine consensus
  - No double-spending
  - Non-propabilstic security
  - Leader cannot misbehave

Decoupling Transaction Verification from Leader Election

- Key blocks:
  - PoW & share value
  - Leader election
- Microblocks:
  - Validating client transactions
  - Issued by the leader



Talk Outline

- Bitcoin and its limitations
- Strawman design: PBFTCoin
- Opening the consensus group
- From MACs to Collective Signing
- Decoupling transaction verification from leader election
- **Performance Evaluation**
- Future work and conclusions

Performance Evaluation

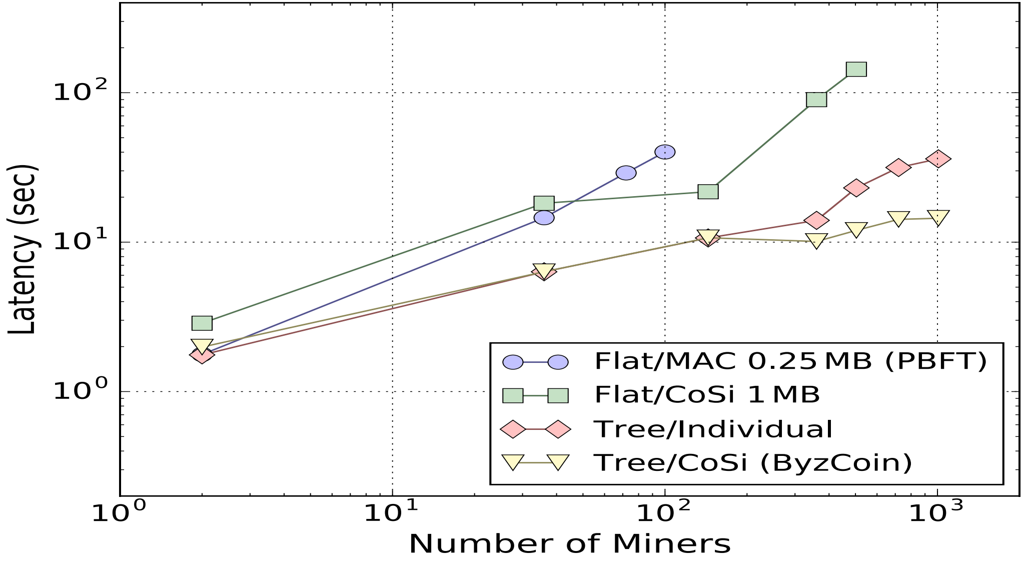
- Experiments run on DeterLab network testbed
  - Up to 1,008* miners multiplexed atop 36 machines
  - Impose 200 ms latencies between all servers
  - Impose 35 Mbps bandwidth per miner

* 1008 = # of ~10-minute key-blocks in 1-week time window

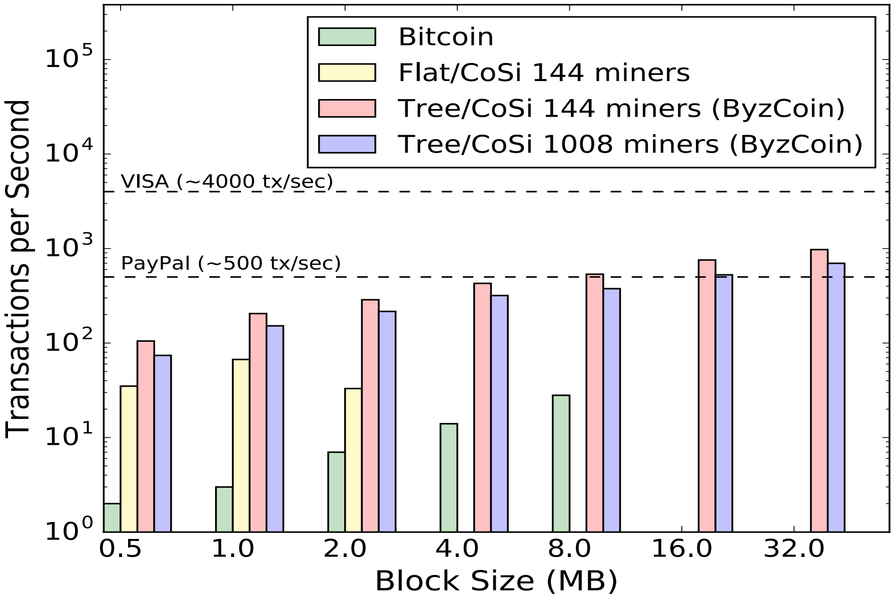
Performance Evaluation

- Key questions to evaluate:
  - What size consensus groups can ByzCoin scale to?
  - What transaction throughput can it handle?

Consensus Latency



Throughput


Bitcoin
Flat/CoSi 144 miners
Tree/CoSi 144 miners (ByzCoin)
Tree/CoSi 1008 miners (ByzCoin)
VISA (~4000 tx/sec)
PayPal (~500 tx/sec)
Transactions per Second
10^0
10^1
10^2
10^3
10^4
10^5
0.5
1.0
2.0
4.0
8.0
16.0
32.0
Block Size (MB)

Talk Outline

- Bitcoin and its limitations
- Strawman design: PBFTCoin
- Opening the consensus group
- From MACs to Collective Signing
- Decoupling transaction verification from leader election
- Performance Evaluation
- **Future work and conclusions**

Challenges for Ongoing Work

- Attacker with >= 1/3 of the shares
  - Switch to probabilistic consistency?
- Can currently only scale-up not scale-out
  - Split the state between different groups?
- Leader can exclude miners from the consensus
  - Instead of burning the bitcoins, donate them?

FAQ

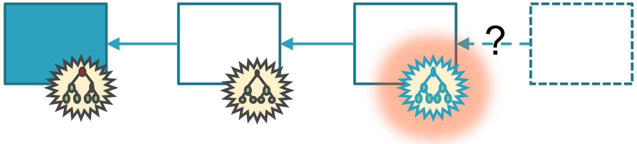
- What happens when an attacker gets more than 1/3?
- Does selfish mining occur in the key-block chain?
- How is the consensus group size selected?
- How do the miners make money?

Surviving 34% attacks

- Key-blocks keep being collectively signed with a needed margin of 51%
- Strong consistency is not immediate
  - Blocks will commit after 6 confirmations
  - Window starts from the last committed block
- Micro-blocks forfeit liveness, if 66% is not achieved

Defend Against Selfish Mining

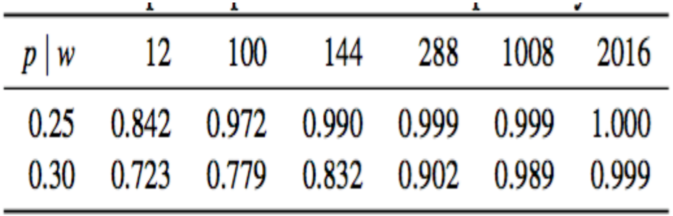
The PoW chain is (almost) fair even under 34% attacks.



Choosing Window Size

- Random sampling experiment
- Probability that the system picks less than $c = \lfloor w/3 \rfloor$
- P>0.99

$$P[X \leq c] = \sum_{k=0}^{c} \binom{w}{k} p^k (1-p)^{w-k}$$



$p \mid w$	12	100	144	288	1008	2016
0.25	0.842	0.972	0.990	0.999	0.999	1.000
0.30	0.723	0.779	0.832	0.902	0.989	0.999

How do the miners make money?

And why participate?

- Coinbase profit is distributed among the active signers
- Same for microblock fees
- Miner profits more when available the full window
- Miner keeps mining to get more shares that correspond to more revenue.

Future Work

- Alternatives to PoW
- Sharding to enable scaling-out
- Incremental deployment to existing cryptocurrencies
  - Model the system on Bitcoin's adversary*?
  - How do miners discover each other?
  - Robustness against 34% attacks?

*Analysis of the Blockchain Protocol in Asynchronous Networks [Pass, Seeman, Shelat]

Conclusion

- Use Collective Signing to scale BFT protocols
- Use PoW to create hybrid permissionless BFT
- Combine the above with Bitcoin-NG
- Demonstrate experimentally its practicality
  - 1MB blocks commit in ~24sec achieve ~150TPS
  - 32MB blocks commit in ~90sec achieve ~1000TPS
- ByzCoin increases the robustness of Bitcoin.

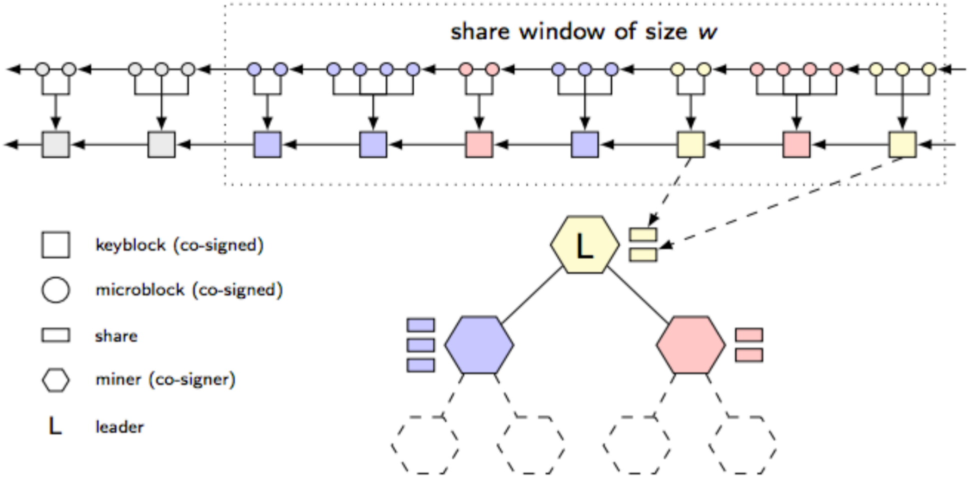
Thank you

eleftherios.kokoriskogias@epfl.ch

people.epfl.ch/eleftherios.kokoriskogias

@LefKok

Byzcoin: Bringing it all Together


share window of size *w*
L
keyblock (co-signed)
microblock (co-signed)
share
miner (co-signer)
L leader