Poster: Bitcoin Meets Collective Signing

Eleftherios Kokoris-Kogias, Philipp Jovanovic, Nicolas Gailly,
Ismail Khoffi, Linus Gasser, and Bryan Ford
Swiss Federal Institute of Technology Lausanne (EPFL)
Email: {eleftherios.kokoriskogias, philipp.joavanovic, nicolas.gailly, ismail.khoffi, linus.gasser, bryan.ford}@epfl.ch

***Abstract*—While showing great promise, Bitcoin requires users to wait tens of minutes for transactions to commit – even then offering only probabilistic guarantees. This work introduces ByzCoin, a novel Byzantine consensus protocol that leverages scalable collective signing to commit Bitcoin transactions irreversibly within seconds. ByzCoin achieves Byzantine consensus while preserving Bitcoin's open membership by dynamically forming hash power-proportionate consensus groups representing recently-successful block miners. ByzCoin employs communication trees and achieves a throughput of up to 975 transactions per second (TPS), which is more than Paypal currently handles, with confirmation latencies of 15-20 seconds while mitigating double spending and selfish mining attacks.**

I. Introduction

The original Bitcoin paper [1] argues that transaction processing is secure and irreversible as long as the largest colluding group of miners represents less than 50% of the total computing capacity and at least about one hour has elapsed. This high transaction confirmation latency limits Bitcoin's suitability for real-time transactions. Later work revealed additional vulnerabilities to transaction reversibility, double-spending, and strategic mining attacks [2], [3].

The key problem is that Bitcoin's consensus algorithm provides weak, propabilistic consistency rather than strong. Strong consistency could offer cryptocurrencies three important benefits. First, all miners agree on the validity of blocks right away, without wasting computational power resolving inconsistencies (*forks*). Second, clients need not wait extended periods for certainty that a submitted transaction is committed. As soon as it appears in the blockchain, the transaction can be considered confirmed. Third, strong consistency provides *forward security*: as soon as a block has been appended to the blockchain, it stays there forever. While strong consistency for cryptocurrencies has been suggested before [4], existing proposals give up Bitcoin's decentralization, introduce new and non-intuitive security assumptions, and/or lack experimental evidence of performance and scalability.

This work introduces ByzCoin [5], a Bitcoin-like cryptocurrency enhanced with strong consistency based on the principles of the well-studied Practical Byzantine Fault Tolerance (PBFT) [6] algorithm. ByzCoin addresses three key challenges in bringing PBFT-based strong consistency to cryptocurrencies: (1) open membership, (2) scalability to hundreds of replicas, and (3) transaction commit rate.

II. ByzCoin Design

We start with a group of $n = 3f + 1$ PBFT replicas–the *trustees*–that has been fixed and globally agreed upon upfront and where f denotes the maximum number of faulty replicas. Prior work has suggested essentially such a design [4], though without addressing the scalability challenges it creates. Under these simplified assumptions PBFTCoin guarantees safety and liveness, since at most f nodes are faulty and thus the usual BFT security bounds apply. Subsequently we address these restrictions, transforming PBFTCoin into ByzCoin.

A. Opening the Consensus Group

Removing PBFTCoin's assumption of a closed consensus group of trustees presents two conflicting challenges. On the one hand, conventional BFT relies on a well-defined consensus group to guarantee safety and liveness. On the other hand, Sybil attacks [7] can break any open-membership protocol involving security thresholds, such as PBFT's assumption that at most f out of $3f + 1$ members are honest.

Bitcoin employs a mechanism already suited to this problem: proof-of-work mining. We adapt this technique into a *proof-of-membership* mechanism with the goal to maintain the "balance of power" within the BFT consensus group over a given fixed-size sliding *share window*. Each time a miner finds a new block, it receives a *share* which proves the miner's membership in the group of trustees and the share window is moved one step forward. Each miner wields "voting power" of a number of shares equal to the number of blocks the miner has successfully mined within the current window. Assuming collective hash power is relatively stable, this implies that within a window, each active miner wields a number of shares statistically proportionate to the amount of hash power that miner has contributed during this time period.

Since we can no longer assume voluntary participation, we need an incentive for nodes to obtain shares and remain active. For this purpose, we adopt Bitcoin's existing incentives of mining rewards and transaction fees – but instead of these rewards all going to the miner of the last block, we split a block's rewards and fees across all members of the current consensus group, in proportion to the number of shares.

B. Replacing MACs by Scalable Collective Signing

In our next refinement step towards ByzCoin, we tackle the scalability challenge caused by PBFT's non-transferrable MAC-based message authentication. By adopting digital signatures for authentication, we remove the necessity that all trustees communicate directly with each other and are able to use sparser and more scalable communication topologies enabling the current leader to collect and distribute third-party verifiable evidence that certain steps in PBFT have succeeded. By relying on tree-based communication structures [8], we can reduce communication complexity from $O(n^2)$ to $O(n)$.

Even with signatures providing transferable authentication, it is costly for the leader to distribute 1000 digital signatures

and wait for everyone to verify them. To tackle this challenge we build on the CoSi protocol [9] for collective signing (co-signing). CoSi does not implement consensus or BFT, but offers a primitive to implement *prepare* and *commit* messages during PBFT rounds. We implement a single ByzCoin round using two sequential CoSi rounds initiated by the current leader. The leader's announcement of the first CoSi round implements the *pre-prepare* and the generated co-signature implements the *prepare* phase of PBFT which ensures that a proposal *can be* committed consistently, but is insufficient to ensure that the proposal *will be* committed. The leader and/or some number of other members could fail before a supermajority of nodes learn about the successful prepare phase. The leader therefore initiates a second CoSi round to implement the PBFT protocol's *commit* phase, in which the leader obtains attestations that all the signing members witnessed the successful result of the prepare phase and make a positive commitment to remember the decision. This co-signature resulting from this second CoSi round effectively attests that a supermajority of members not only considers the leader's proposal "safe" but promises to remember it, and hence that the leader's proposal has committed. Finally, CoSi allows us to reduce the communication complexity even further from $O(n)$ to $O(\log n)$.

C. Decoupling Transaction Verification from Leader Election

While ByzCoin so far provides a scalable guarantee of strong consistency, ensuring that clients wait only for the next block, the time they still have to wait *between* blocks may nevertheless be significant. While ByzCoin's strong consistency might in principle make it "safe" from a consistency perspective to increase block mining rate, doing so could still exacerbate liveness and other performance issues, just as in Bitcoin. To enable lower client-perceived transaction latencies we build on the idea of Bicoin-NG [10] to decouple the functions of transaction verification from leader election and consensus group membership.

As in Bitcoin-NG, we use two different kinds of blocks. The first, *microblocks* represent transactions to be stored and committed. The current leader creates a new microblock every few seconds and uses the CoSi-based PBFT protocol above to commit and collectively sign it. The other type of block, *keyblocks*, are mined via proof-of-work and serve to elect leaders and create shares. This decoupling allows the current leader to propose and commit many microblocks, containing many smaller batches of transactions, within a keyblock mining period. Unlike Bitcoin-NG, in which a malicious leader could rewrite history or double-spend within this period until the next keyblock, ByzCoin ensures that each microblock is irreversibly committed regardless of the current leader's behavior. In Bitcoin-NG one blockchain includes both types of blocks, which introduces a race condition for miners, as microblocks are created, the miners have to change the header of their keyblocks to mine on top of the latest microblock. In ByzCoin, in contrast, the blockchain becomes two separate parallel blockchains. The main blockchain consists of all minded keyblocks. The microblocks form a secondary blockchain which depends on the primary to identify the era in which every microblock belongs. An overview on the final ByzCoin design is given in Figure 1.

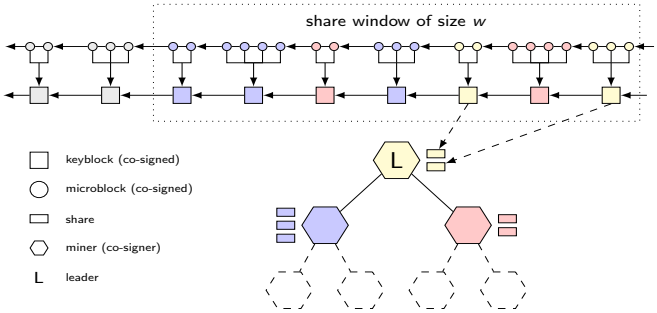

Fig. 1. Overview on the ByzCoin design

III. Evaluation

We have written a working prototype of ByzCoin[1] in the Go programming language [11] and evaluated it on DeterLab [12], using 36 physical machines that run up to 28 separate ByzCoin processes. To mimic a realistic wide-area network environment we imposed a round-trip latency of 200 ms between any two machines and a link bandwidth of 35 Mbps per simulated host. The results of our experiments show a maximum throughput of 975 TPS for a group of 148 miners. Furthermore, we measured signing latencies between 15 seconds, for 1 MB blocks, and 2 minutes, for 16 MB blocks.

Moreover, ByzCoin also mitigates double-spending and network partitioning attacks [2], [3], since an attacker would need to forge co-signatures to pre-compute or falsify blocks.

We refer to [5] for a more thorough discussion of our implementation results and a preliminary security analysis.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Oct. 2008.

[2] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Financial Cryptography and Data Security*, pp. 436–454, Springer, 2014.

[3] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on Bitcoin's peer-to-peer network," in *24th USENIX Security Symposium*, pp. 129–144, 2015.

[4] C. Decker, J. Seidel, and R. Wattenhofer, "Bitcoin Meets Strong Consistency," in *17th International Conference on Distributed Computing and Networking (ICDCN), Singapore*, January 2016.

[5] E. Kokoris-Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing," Feb. 2016.

[6] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *3rd USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, Feb. 1999.

[7] J. R. Douceur, "The Sybil attack," in *1st International Workshop on Peer-to-Peer Systems (IPTPS)*, Mar. 2002.

[8] V. Venkataraman, K. Yoshida, and P. Francis, "Chunkyspread: Heterogeneous unstructured tree-based peer-to-peer multicast," in *14th International Conference on Network Protocols (ICNP)*, Nov. 2006.

[9] E. Syta, I. Tamas, D. Visher, D. I. Wolinsky, P. Jovanovic, L. Gasser, N. Gailly, I. Khoffi, and B. Ford, "Keeping Authorities "Honest or Bust" with Decentralized Witness Cosigning," in *37th IEEE Symposium on Security and Privacy*, May 2016.

[10] I. Eyal, A. E. Gencer, E. G. Sirer, and R. van Renesse, "Bitcoin-NG: A scalable blockchain protocol," in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, (Santa Clara, CA), USENIX Association, Mar. 2016.

[11] "The Go programming language," Jan. 2015.

[12] "DeterLab network security testbed," September 2012.

[1] Available at https://github.com/dedis/cothority.