Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing

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

1. INTRODUCTION

While showing great promise, Bitcoin [13] faces some fundamental security and performance hurdles that prevent its large-scale adoption: In comparison to classic payment providers, such as VISA or PayPal, which are able to perform hundreds to thousands of transactions (tx) per second[1] with very low transaction confirmation latencies, Bitcoin's current transaction throughput is limited to about 7 tx/sec and its consensus mechanism requires users to wait tens of minutes for transactions commitment, even then, providing only probabilistic consistency gurantees. This means that inconsistencies (*forks*) might occur when different miners find new blocks independently and at about the same time which then splits the peers' views on the blockchain. Fork-resolution regularly destroys large numbers of transactions, sometimes even hours after their initial submission, thereby wasting all the computational power spent on the orphaned branch. As a consequence, Bitcoin's peer-to-peer network establishes a consistent view on the distributed ledger only eventually. The probabilistic consistency of Bitcoin's consensus mechanism is also one of the reasons that the cryptocurrency is susceptible to all kinds of attacks [1, 6–8, 10, 14]. One solution to mitigate many of those problems, is to eliminate Bitcoin's lazy fork-resolution mechanism and adopt *strong consistency* [3, 4], a more proactive approach that offers the following important benefits:

1. All miners agree on the validity of the blocks right away, without wasting computational power to resolve forks.
2. Clients need not wait extended periods to be certain that a submitted transaction is committed; as soon as it appears in the blockchain, the transaction can be considered confirmed.
3. Once a block has been appended to the blockchain, it stays there, forever (as long as there is an honest majority of miners). This property is also often referred to as *forward security*.

In this work we show how to implement strong consistency in Bitcoin by introducing ByzCoin [9, 11], a novel Byzantine consensus protocol. We moreover present results from our experimental evaluation which indicate that ByzCoin-improved Bitcoin can increase its throughput by two orders of magnitude, and finally also discuss some of the basic deployment challenges that need to be solved.

2. BYZCOIN

ByzCoin is a novel scalable Byzantine fault-tolerant (BFT) consensus protocol that provides strong consistency, while scaling to processing throughputs of hundreds of transactions per second, among hundreds to thousands of decentralized miners. ByzCoin utilizes an adaptation of the well-studied Practical Byzantine Fault Tolerance (PBFT) [2] and introduces four key improvements over Bitcoin:

1. ByzCoin's improved PBFT-like consensus mechanism commits Bitcoin transactions irreversibly within seconds.
2. ByzCoin preserves Bitcoin's open-membership property by dynamically forming hash power-proportionate consensus groups that represent recently-successful block miners.
3. ByzCoin uses communication trees to further optimize transaction commitments and verification under normal operation while guaranteeing safety and liveness under Byzantine faults.
4. ByzCoin decouples the election of a new leader from transaction verification, an approach inspired by Bitcoin-NG [5], that enables ByzCoin's transaction throughput to further increase.

Together, all these optimizations enable ByzCoin to achieve throughputs higher than PayPal currently handles, and to provide low confirmation latencies. Another benefit of ByzCoin's fast transaction commitment, ranging from a few seconds up to at most one or two minutes after submission, is the mitigation of double-spending and selfish mining attacks.

2.1 Design

An overview on ByzCoin's design is depicted in Fig. 1.

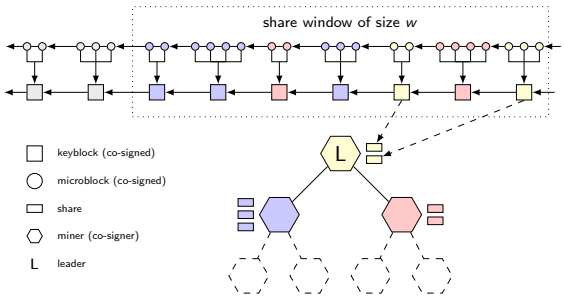

Figure 1: ByzCoin design

The lower part of Fig. 1 shows ByzCoin's consensus group that is comprised of recently-successful block miners and uses a PBFT-like mechanism to reach consensus. Instead of PBFT's MAC-based authentication, which has quadratic communication and computation complexity, ByzCoin uses *CoSi* [15], a distributed protocol that utilizes aggregated Schnorr signatures and tree-based communication to make large-scale, decentralized, collective signing practical. Overall, CoSi reduces communication complexity from quadratic to logarithmic, enables third-party verifiability, and signature verification in constant-time complexity. For more details on how exactly ByzCoin's consensus group mechanism works we refer to the research paper [11]. Another feature that ByzCoin adopts from PBFT is the role of the consensus group leader whose task is to bundle transactions into blocks and initiate new signing rounds. All actions, however, taken by the leader have to be approved by a two-thirds supermajority of the consensus group members, which effectively leads to strong consistency and all of its benefits discussed earlier. In case the leader misbehaves, the miners in ByzCoin's consensus group can start a voting round and dismiss the Byzantine node but only if, again, a two-thirds supermajority approves. The requirement of the two-thirds supermajority for decision making comes from Byzantine agreement theory [12] that permits at most f malicious/faulty nodes among a total of $3f + 1$ nodes. Another important part is the leader election mechanism, which brings us to the next component of ByzCoin's design.

[1] https://en.bitcoin.it/wiki/Scalability#Scalability_targets

The upper part of Fig. 1 shows ByzCoin's blockchain which is divided into two interdependent sub-chains: one for *keyblocks* and one for *microblocks*.

Keyblocks are used to manage ByzCoin's consensus group membership. These blocks are generated by the miners through proof-of-work roughly every 10 minutes, as in Bitcoin, and are collectively signed by ByzCoin's consensus group. Keyblocks form a regular blockchain. A miner who successfully mines a new keyblock is rewarded with a *consensus group share*, a so-called *proof-of-membership*, thereby gains entry into the consensus group, if he is not already a member, and becomes the next group leader. A fixed-size sliding window mechanism constitutes the total number of available shares: Any share beyond the current window expires and miners who no longer hold any valid shares drop out of the consensus group. The number of valid shares in the possession of a miner reflects his voting power within the consensus group, when committing transactions. Moreover, this number determines the portion of coins a miner receives as a reward, when a new keyblock is found. In other words, ByzCoin rewards not only the node that mines a new keyblock but instead splits, proportionate to the valid shares each miner holds, the produced coins among all miners of the consensus group. ByzCoin also uses this technique to split transaction costs as a reward, once no more coins can be mined. The proof-of-membership approach ensures liveness, as dormant miners are removed from the consensus group and the share-proportionate rewards further incite all miners to remain active and contribute to the progress of the system.

Microblocks, on the contrary, contain transactions, are proposed by the current leader, and, as they do not require proof-of-work, are committed much more frequently by the consensus group. Each microblock contains, in addition to the list of transactions, a hash of the last microblock to ensure total ordering, as well as a hash of the leader's keyblock to identify the era the microblock belongs to. Even though microblocks are created by the consensus group leader, ByzCoin's witness-mechanism deters leaders from misbehaving (such as mounting double-spend attacks), because any misconduct would be immediately detected by the other group members, which in turn can trigger a view change thereby removing the malicious node.

For more details on ByzCoin's design we refer to the research paper [11].

2.2 Experimental Results

To evaluate the design of ByzCoin, we wrote a prototype, available on GitHub as part of the cothority project[2], and conducted thorough experiments, measuring transaction confirmation-latency and throughput. We experimented with consensus group sizes between 144 and 1008 nodes, which corresponds to a window of successful keyblock miners ranging from the last day's up to the last week's. Fig. 2 shows ByzCoin's throughput in comparison to other systems. The data for our simulations is based on actual transactions from a portion of the Bitcoin blockchain.

The average latency we measured, for example for 32 MB blocks ($\approx$ 66000 tx) and a consensus group size of 144 members, was around 90 seconds. For this particular configuration, ByzCoin's throughput ($\approx$ 700 tx/sec) outperforms PayPal's as shown in Fig. 2. For a more elaborate discussion of ByzCoin's performance evaluation we refer again to the research paper [11].

2.3 Deployment Challenges

Developing a reasonable deployment strategy for ByzCoin on top of Bitcoin involves solving at least the following three challenges:

[2] https://github.com/dedis/cothority

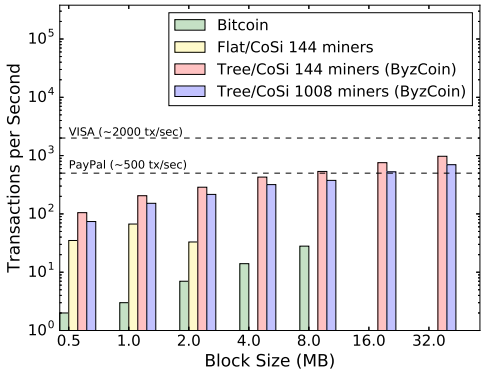

Figure 2: Throughput

1. Roll out code that is backward-compatible with the current Bitcoin system until a critical mass of miners supports the new ByzCoin consensus.
2. Build the initial consensus group which then switches to the new consensus mechanism once the above critical mass appears.
3. Handle (hopefully rare) PBFT deadlock events, e.g., because too many miners disappear in too short a time and no two-thirds supermajority is or will ever again be available in the current consensus group.

To address the first two challenges, we can utilize the already running Nakamoto consensus as a bootstrapping mechanism. From an outside point of view, Bitcoin would basically operate as usual, so long as the bootstrapping is not finished. A few things change from the perspective of the miners though: each miner puts his public key and contact information (IP address/port number), respectively, into every block he creates. Including the public key enables a miner to claim the block as a share once he finds the necessary proof-of-work; the contact information is required so that consensus group members are able to find each other and create the communication tree. As soon as the number of distributed shares hits the maximum share-window size, all miners in the consensus group switch to ByzCoin, the last miner to join the group becomes the leader, and the group co-signs the leader's keyblock. Afterwards, the leader creates the new microblock-subchain from his keyblock and starts creating and submitting microblocks to the co-signing consensus procedure. To handle the third challenge, we can use the Nakamoto consensus as a fall-back option: If miners notice a lack of progress from the PBFT consensus group for too long (perhaps after several view-changes), they return to committing transactions as part of their keyblocks, just as in vanilla Bitcoin, thus effectively reverting the system to its pre-ByzCoin agreement mechanism. As soon as a certain threshold of shares is again distributed, miners can re-start ByzCoin's consensus. Another option would be to use Bitcoin-NG as a fall-back mechanism, which has the advantage of providing similarly good performance as ByzCoin but guarantees not all of ByzCoin's security features.

3. CONCLUSION

ByzCoin is a scalable Byzantine fault tolerant consensus algorithm for open decentralized blockchain systems such as Bitcoin. ByzCoin's strong consistency increases Bitcoin's core security guarantees—shielding against attacks on the consensus and mining system such as N-confirmation double-spending, intentional blockchain forks, and selfish mining—and also enables high scalability and low transaction latency. We developed a wide-scale prototype implementation of ByzCoin, validated its efficiency with measurements and experiments, and have shown that Bitcoin can increase the capacity of transactions it handles by more than two orders of magnitude.

4. REFERENCES

[1] M. Apostolaki, et al. Hijacking Bitcoin: Large-scale Network Attacks on Cryptocurrencies. *arXiv preprint arXiv:1605.07524*, 2016.

[2] M. Castro et al. Practical Byzantine Fault Tolerance. In *3rd OSDI*, Feb. 1999.

[3] K. Croman, et al. On Scaling Decentralized Blockchains (A Position Paper). In *3rd Workshop on Bitcoin and Blockchain Research*, 2016.

[4] C. Decker, et al. Bitcoin Meets Strong Consistency. In *17th International Conference on Distributed Computing and Networking (ICDCN), Singapore*, January 2016.

[5] I. Eyal, et al. Bitcoin-NG: A Scalable Blockchain Protocol. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, Santa Clara, CA, Mar. 2016. USENIX Association.

[6] I. Eyal et al. Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography and Data Security*, pages 436–454. Springer, 2014.

[7] A. Gervais, et al. Tampering with the Delivery of Blocks and Transactions in Bitcoin. In *22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 692–705. ACM, 2015.

[8] E. Heilman, et al. Eclipse Attacks on Bitcoin's Peer-to-Peer Network. In *24th USENIX Security Symposium*, pages 129–144, 2015.

[9] P. Jovanovic. ByzCoin: Securely Scaling Blockchains. Hacking, Distributed, August 2016.

[10] G. O. Karame, et al. Double-spending fast payments in Bitcoin. In *19th ACM Conference on Computer and communications security*, pages 906–917. ACM, 2012.

[11] E. Kokoris-Kogias, et al. Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing. In *Proceedings of the 25th USENIX Conference on Security Symposium*, 2016.

[12] L. Lamport, et al. The Byzantine Generals Problem. *TOPLAS*, 4(3):382–401, 1982.

[13] S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System, 2008.

[14] K. Nayak, et al. Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack. In *1st IEEE European Symposium on Security and Privacy*, Mar. 2015.

[15] E. Syta, et al. Keeping Authorities "Honest or Bust" with Decentralized Witness Cosigning. In *37th IEEE Symposium on Security and Privacy*, May 2016.