Hybrid Consensus: Scalable Permissionless Consensus

Rafael Pass (CornellTech, IC3) Elaine Shi (Cornell, IC3)

1 Introduction

The distributed systems and cryptography literature traditionally has focused on protocols whose participants are known *a priori*. Bitcoin's rapid rise to fame represents an exciting breakthrough: Bitcoin empirically demonstrated that by leveraging assumptions such as proofs-of-work, non-trivial secure applications can be built on top of a fully decentralized network where nodes join and leave freely and dynamically, and there is no pre-established trust between participants. In the remainder of the paper, we will refer to the two network settings as the *permissioned* setting and the *permissionless* setting respectively.

Informally speaking, Bitcoin's core consensus protocol, often referred to as Nakamoto consensus [14], realizes a "replicated state machine" abstraction, where nodes in a permissionless network reach agreement about a set of transactions committed as well as their ordering. Since the protocol relies on chaining of blocks of transactions, it is often referred to as the "blockchain".

Achieving consensus in the traditional permissioned model turns out to be a classical distributed systems problem, and there is a long line of research that seeks to design and optimize Byzantine consensus protocols [4, 7, 13]. The fact that we can obtain consensus in a permissionless model (relying on proofs-of-work) was the novel contribution of Bitcoin. In a sense, Bitcoin popularized a new model of distributed systems that was rarely considered in 30 years of classical distributed systems literature.

The price of decentralization? However, achieving consensus in the permissionless model comes at a cost. Since identities of nodes are not known *a priori*, it is imperative to defend against a Sybil attack where an attacker makes up arbitrarily many identities to outvote honest nodes. The Bitcoin protocol critically relies on proofs-of-work to roughly enforce the idea of "one vote per hashpower". The adoption of proofs-of-work, however, makes the Bitcoin protocol unscalable. As Croman et al. [5] point out, Bitcoin achieves terrible performance: the Bitcoin network can sustain at most 7 tx/sec, at a transaction confirmation time of $10+$ min (*c.f.* a main-stream payment processor such as Visa handles an average rate of $2,000$ tx/sec, and a peak rate of $59,000$ tx/sec). Further, each confirmed transaction costs roughly \$1 to \$6 if we were to amortize the network's total electrcity consumptions over all transactions being confirmed — today, this cost is in some sense being paid by the speculators of Bitcoin.

By contrast, traditional Byzantine consensus protocols (for the permissioned model) need not rely on proofs-of-work, and are capable of attaining high throughput and low response time, e.g., with about 100 PBFT [4] nodes deployed across multiple data centers on Amazon AWS, Kroman et al. [5] demonstrated a transaction throughput of $10,000+$ tx/sec, and transaction confirmation time on the order of seconds. This naturally raises an important question.

Is this price of decentralization inherent?

In this paper, we show that the "price of decentralization" need not be inherent if minor security relaxations are permitted. We propose a new consensus protocol that attains the best of both worlds: 1) formal security guarantees in the permissionless model; and 2) throughput and response times comparable to classical Byzantine consensus.

Scheme	TX conf. time	Processing/tx	% honest
Nakamoto [14], BitcoinNG [8]	$\Theta(\lambda\Delta)$	$O(n)$	$\sim \frac{1}{2}$
Fruitchain [16] (concurrent)	$\Theta(\lambda\Delta)$	$O(n)$	$\sim \frac{1}{2}$
Hybrid consensus over Nakamoto [14]	Opt: $O(\delta)$ Worst: $O(\lambda\delta)$	$O(\lambda)$	$\sim \frac{3}{4}$
Hybrid consensus over Fruitchain [16]	Opt: $O(\delta)$ Worst: $O(\lambda\delta)$	$O(\lambda)$	$\sim \frac{2}{3}$

Table 1: **Summary of our results.** n denotes the total number of nodes (assuming all nodes have equal hashpower); Δ denotes a pre-determined upper-bound on the network's transmission delay; δ denotes the actual delay of the network; λ is the security parameter for attaining $2^{-\lambda}$ security failure.

Our result is in the form of an "efficiency bootstrapping theorem" — much as the well-known hybrid encryption and OT-extension are "efficiency bootstrapping theorems", we use a slow blockchain protocol (called snailchain) such as Nakamoto consensus [10,14] to bootstrap fast permissioned byzantine consensus, the end result being a scalable consensus protocol in the permissionless model. For this reason, we call our protocol "hybrid consensus".

1.1 Our Results and Contributions

Throughout this paper, we consider a partially synchronous model where an adversarial network delivers messages, possibly out of order, in at most δ time steps.

Advantages of hybrid consensus. Hybrid consensus has the following compelling advantages in comparison with known blockchain protocols. We state these properties assuming typical parametrizations.

- **Latency.** Practically speaking, existing blockchain protocols such as Nakamoto incur a transaction confirmation time of 10 min and with a large (i.e., constant) probability of security failure. To attain $1 - 2^{-\lambda}$ security (e.g., important for high-valued transactions), one must wait for $O(\lambda)$ blocks thus incurring even a longer delay. From a theoretical perspective, Pass et al. [15] recently show that such a transaction confirmation time is in some sense inevitable for the Nakamoto consensus protocol. In particular, to achieve security against any $\alpha = O(1)$ fraction of corruption, one parametrize the puzzle difficulty such that the expected block interval is $c\Delta$ for some appropriate constant c, where Δ is a predetermined upper bound of the network's delay δ. Specifically, they show that $c = 60$ would resist a 49.57% attack, and $c \approx 3$ would resist a $1/3$ attack.

  Hybrid consensus fundamentally improves the transaction confirmation time in a manner that is of both practical and theoretical interest. As mentioned, existing blockchain protocols [8, 10, 14, 15] achieve a transaction confirmation time of $O(\lambda\Delta)$ to attain $1 - 2^{-\lambda}$ security where Δ is a possibly loose upper bound on the network's delay that must be determined a priori. Ideally, we would like the protocol's transaction confirmation time to depend on the actual network delay δ, not on the a-priori known upper bound Δ — we henceforth refer to this property as *responsiveness*. How to achieve responsiveness for permissionless consensus is an open theoretical challenge to which hybrid consensus is the first to provide a solution. Hybrid consensus achieves a transaction confirmation time of $O(\lambda\delta)$ in the worst case, where λ is the security parameter. Not only so, in the optimistic case it achieves a transaction confirmation time of $O(\delta)$, i.e., independent of the security parameter λ.

  Pragmatically speaking, the following factor also potentially makes Δ much bigger than δ which speaks in favor of hybrid consensus. In practice, block propagation in an open-enrollment, permissionless setting is achieved through an overlay network; and thus Δ must be parametrized to be an upper bound of the overlay network's delay. Therefore, Δ must not only account for multiple hops of direct IP links, but also account for potentially adversarial attacks targeted at the overlay routing protocol. By contrast, if hybrid consensus is deployed, once committee members are discovered, they can possibly communicate with each other as well as other nodes over direct IP links — in this case, the optimistic transaction confirmation time can depend on the δ which is the latency of these direct IP links.

- **Transaction processing.** Consider smart contract applications whose most popular embodiment is Ethereum. Today's blockchain protocols require all miners to execute the smart contract program for each transaction, thus incurring a linear in n processing cost — this makes existing blockchain protocols unscalable to large deployments. Hybrid consensus reduces the transaction processing cost to $O(\lambda)$, since regardless of how large n is, only a small committee of size $O(\lambda)$ need to execute the smart contracts and process the transanctions.

Security tradeoff. Hybrid consensus achieves the aforementioned advantages under a decentralized, permissionless security model — but with minor relaxations in comparison with Nakamoto consensus. As Pass et al. [15] show, Nakamoto consensus is secure against a *fully adaptive* adversary as long as at any point of time, the adversary controls no more than $1/2 - \epsilon$ fraction of nodes.

Hybrid consensus relaxes the security model to attain the aforementioned advantages. First, it is obvious that any scheme that downselects from n nodes to a λ-sized committee cannot secure against adaptive corruption, since the adversary can simply target the selected committee members. However, we can indeeed prove the security of hybrid consensus protocol assuming that it takes a while for an adversary to corrupt a node. This appears to be a realistic assumption since in practice, corruption is indeed not an instant operation, and it takes sometime to successfully infect a targeted, clean computer.

Second, depending on which underlying snailchain scheme is used to instantiate hybrid consensus, our scheme requires that $3/4+\epsilon$ or $2/3+\epsilon$ fraction nodes be honest. The most natural snailchain candidate is the original Nakamoto consensus [14] — but due to the selfish-mining attack and chain quality loss [9, 10, 15], we can only tolerate $1/4 - \epsilon$ corruption in this case. Alternatively, if we used Fruitchain [16] to instantiate the underlying snailchain in hybrid consensus, we can tolerate up to $1/3 - \epsilon$ corruption.

Comparison with closely related works. Although the idea of combining permissionless and permissioned consensus has been discussed in the community (e.g., the recent work by Decker et al. [6] and the concurrent and independent work ByzCoin [11]), to the best of our knowledge, no prior work has provided a formal treatment. As we show, combining permissioned and permissionless protocols is non-trivial both in terms of construction and in terms of proving security. Without a formal analysis, it is not clear what earlier/concurrent approaches achieve. For example, in the concurrent work Byzcoin [11], participants may not agree on the PBFT committee with constant probability thus breaking consensus. In both Decker et al. [6] and Byzcoin [11], consensus can be broken with probability 1 in the worst case if the adversary controls more than $\frac{1}{4}$ of the computation power, since Nakamoto's chain quality suffers from a loss resulting from a selfish mining attack (although Byzcoin actually claims resilience to a $\frac{1}{3}$ attack!). Further, Decker et al. [6] does not achieve responsiveness. Therefore, our work is distinct in the following senses: 1) we provide the first provably secure protocol that achieves *responsiveness* (i.e., transaction confirmation time depends on actual network performance, not an a-priori upper bound on the network's delay) in the permissionless model with proofs-of-work; 2) we are the first to formalize the precise model and security guarantees.

1.2 Technical Highlights

While the intuitive idea may seem simple, formally instantiating and proving hybrid consensus secure exposed numerous interesting technical subtleties and challenges.

Committee switchover and concurrent composition of consensus. Hybrid consensus involves using the snailchain to select rotating committees. Committee switchover is non-trivial, and would entail transiently overlapping BFT instances. Unfortunately, Lindell et al. [12] demonstrate the impossibility of concurrently composable Byzantine agreement! We circumvent this impossibility by observing that due to the way how hybrid consensus bootstraps from the snailchain, nodes have common knowledge of the set of (freshly chosen) public keys of committee members.

Adversarial selective opening of committees. Hybrid consensus forms PBFT committee by taking the miners of consecutive csize blocks. Here the choice of the committee is subject to influence by the adversary, since the adversary can launch block withholding or selfish-mining attacks to erase an honest node's block if it does not like the specific public key. We refer to such an attack as "selective opening of committees". To formally prove security under such selective opening attacks, instead of relying on a property-based security definition for the underlying BFT protocol, we need a strengthened security notion, which can be regarded as a UC notion of blackbox reduction to signature security.

Formal framework for composing consensus protocols. Our work demonstrates a framework for composing consensus protocols. Due to growing interests in cryptocurrencies and their applications, there is a clear and increasing appetite from the community for mixing and composing multiple consensus protocols — therefore our approach can be of independent interest. To achieve this, we rely on a Univeral Composition-like formal framework that enables composition of cryptographic protocols. We use the UC framework [1–3] in a way such that we define formal properties of protocols rather than relying on ideal behavioral specifications.

References

[1] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS*, 2001.

[2] R. Canetti, Y. Dodis, R. Pass, and S. Walfish. Universally composable security with global setup. In *Theory of Cryptography*, pages 61–85. Springer, 2007.

[3] R. Canetti and T. Rabin. Universal composition with joint state. In *CRYPTO*, 2003.

[4] M. Castro and B. Liskov. Practical byzantine fault tolerance. In *OSDI*, 1999.

[5] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, D. Song, and R. Wattenhofer. On scaling decentralized blockchains (a position paper). In *Bitcoin Workshop*, 2016.

[6] C. Decker, J. Seidel, and R. Wattenhofer. Bitcoin meets strong consistency. In *Proceedings of the 17th International Conference on Distributed Computing and Networking*, ICDCN '16, 2016.

[7] C. Dwork, N. Lynch, and L. Stockmeyer. Consensus in the presence of partial synchrony. *J. ACM*, 1988.

[8] I. Eyal, A. E. Gencer, E. G. Sirer, and R. van Renesse. Bitcoin-NG: A scalable blockchain protocol. In *NSDI*, 2016.

[9] I. Eyal and E. G. Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *FC*, 2014.

[10] J. A. Garay, A. Kiayias, and N. Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Eurocrypt*, 2015.

[11] 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. *CoRR*, abs/1602.06997, 2016.

[12] Y. Lindell, A. Lysyanskaya, and T. Rabin. On the composition of authenticated byzantine agreement. *J. ACM*, 53(6):881–917, Nov. 2006.

[13] J.-P. Martin and L. Alvisi. Fast byzantine consensus. *IEEE Trans. Dependable Secur. Comput.*, 3(3), 2006.

[14] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[15] R. Pass, L. Seeman, and abhi shelat. Analysis of blockchain protocol in asynchronous networks. `https://eprint.iacr.org/2016/454`.

[16] R. Pass and E. Shi. Fruitchains: an ϵ-incentive compatible blockchain. Manuscript, 2016.