Algorand: from Theory to Practice

Jing Chen
Algorand Inc., Boston, MA 02116
jing@algorand.com

Abstract

A summary of my talk at the ApPLIED Workshop at DISC 2019.

Introduction

Blockchains stand to revolutionize the way a modern society operates. They can secure all kinds of traditional transactions, such as payments, in the exact order in which the transactions occur; and enable totally new transactions, such as cryptocurrencies and smart contracts. They can remove intermediaries and usher in a new paradigm for trust. As currently implemented, however, blockchains scale poorly and cannot achieve their enormous potential. Algorand is the first blockchain that is truly secure, scalable and decentralized. It is permissionless and works in a highly asynchronous environment. It dispenses with "proof of work" and "miners" and requires only a negligible amount of computation. Moreover, its transaction history does not "fork", guaranteeing immediate finality of a transaction the moment the transaction enters the blockchain. In this talk, I will briefly introduce Algorand's core technology, recent development and roadmap. The readers may refer to [7, 8, 4, 6] for more details.

Underlying the Algorand blockchain is a new Byzantine Agreement protocol that is highly efficient. It works under an adversarial model where the adversary can dynamically corrupt any user at any time, control the actions of a corrupted user, and perfectly coordinate the actions of all corrupted users. Even with such a strong adversary, the protocol achieves asynchronous safety and guarantees that the Algorand blockchain doesn't fork even when the underlying propagation network is partitioned. Accordingly, any transaction that appears on the blockchain is immediately final and can be relied upon, without the need of waiting for more blocks being added after it. On the other hand, the protocol achieves liveness as long as messages propagated by honest users are received by other honest users within a known time bound.

Another important idea that makes the Algorand blockchain scalable is *cryptographic self-selection.* Indeed, having millions of users participate in the Byzantine Agreement in order to select the next block is unrealistic. One possibility is to *publicly* select, at random, a subset of users to form a *committee* and participate on behalf of everybody. However, once the identities of the committee members become public, the adversary can corrupt them so that they behave maliciously. Instead, the Algorand blockchain has each user self-select herself into a committee. Thanks to cryptographic primitives such as unique signatures, cryptographic hash functions and verifiable random functions, a user can privately generate a unique "lottery ticket", which she can use to prove her membership in the committee if she is selected, but cannot cheat and convince others to accept her proof otherwise. When participating in the Byzantine agreement, a committee member propagates her winning ticket together with her proposal or voting message. No other communication is needed to find out who is selected. In this way, the adversary learns the fact that a particular user is selected only after the user has sent out her message in the protocol instead of before, and corrupting the user doesn't let the adversary control the message being sent out.

One more idea is needed here. If a selected committee participates in multiple steps of the Byzantine agreement, the adversary can learn the identities of its members and corrupt all of them after seeing their first messages, so that they behave maliciously in remaining steps. The Algorand blockchain is immune to this problem because its Byzantine agreement has an important property referred to as *user replaceability*: the protocol doesn't rely on users keeping private states, and the message that a user should send in a step can be determined solely based on messages that have been propagated to him/her in previous each steps. As such, the protocol has a committee randomly and independently selected for every step. Corrupting the committee members for a step does not give the adversary more power than random corruption in terms of controlling committee members for future steps.

In order to deal with Sybil attacks, the selection probability is the same for every token: that is, in effect, tokens are selected at random, and users that own the selected tokens participate in the Byzantine agreement. The users do not need to delegate their right of participation to a small group of super nodes, neither do they need to lock up their tokens for a long time in order to participate in the consensus protocol. Indeed, the approach here is a pure form of proof-of-stake.

Many other ideas have been introduced in the Algorand blockchain, but rather than covering them all today, I'd like to report on some recent developments on actually implementing the blockchain and putting it to work. The Algorand MainNet launched in mid-June 2019, and has been running smoothly since. The TestNet has been running since April, 2019; it runs the same version of the blockchain as the MainNet, so that developers can test their software (such as wallets) before running them on the MainNet. The code has been audited by third parties, and the entire code repository is open-sourced and available at [3]. Various tools for developers, such as SDKs for multiple programming languages and tutorials, can be found at [2]. We continue to enlarge the tool set, and have also launched a bug bounty program [1]. In addition to a pen-and-paper analysis, with collaborators at Runtime Verification, we have begun formal verification of the consensus protocol using the Coq proof assistant. Our model explicitly incorporates timing issues and adversarial actions, reflecting a more realistic environment that may be faced by a public blockchain. We have proved asynchronous safety of the protocol under this model, and a paper reporting on the progress is available at [5].

Since the MainNet launch, we continue to develop new technology to enable important applications on the blockchain. For example, in forthcoming versions, users will be able to issue their own fungible tokens directly in layer-1 of the blockchain. Moreover, users will be able to clear multiple transfers, among arbitrary sets of users and for arbitrary sets of layer-1 currencies, in a single transaction: that is, in a truly atomic way without relying on devices such as hashed time-locks. And there is still more to come. Interested readers can find the most up-to-date information at `https://www.algorand.com/`. Stay tuned!

References

[1] `https://bugcrowd.com/algorand`.

[2] `https://developer.algorand.org/`.

[3] `https://github.com/algorand/go-algorand`.

[4] Algorand blockchain features. `https://github.com/algorandfoundation/specs/blob/master/overview/Algorand_v1_spec-2.pdf`, 2019.

[5] M. A. Alturki, J. Chen, V. Luchangco, B. Moore, K. Palmskog, L. Peña, and G. Roşu. Towards a verified model of the Algorand consensus protocol in Coq. In *FMBC'19: Workshop on Formal Methods for Blockchains, 3rd Formal Methods World Congress*, 2019.

[6] J. Chen, S. Gorbunov, S. Micali, and G. Vlachos. Algorand Agreement: Super fast and partition resilient Byzantine agreement. Cryptology ePrint Archive, Report 2018/377, 2018. `https://eprint.iacr.org/2018/377`.

[7] J. Chen and S. Micali. Algorand: A secure and efficient distributed ledger. *Theoretical Computer Science*, 777:155–183, 2019.

[8] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich. Algorand: Scaling Byzantine agreements for cryptocurrencies. In *SOSP*, pages 51–68, 2017.