An Overview of Algorand As A Blockchain Protocol Component

Ayse Karaman, Ph. D.
makalelr@gmail.com

Algorand is a Proof-of-Stake (PoS) Byzantium consensus protocol proposed by Silvio Micali – a highly distinguished authority in the field. The following give an overview of Algorand:

Consensus: reached thru a fast Byzantium agreement carried out by a set of "selected" users in each round. In a block-signing round, Algorand users take the roles of one round-leader to propose the next block in the chain, and verification committee (VC) members to approve the proposed/valid block. The "selection" here uses a cryptographic sortition that allows each user himself to find out whether he has a role in the round. For this calculation, the cryptographic sortition uses, along with the rest of the parameters, a random seed to disallow any manipulation of the outcome by the users and adversaries.

PoS: The amount of Algorand coins a user currently possesses is the only attribute of his that effects the possibility of his participation in a round of block signing. The rest of the parameters that go into this process are not specific to any users, and these parameters can not be manipulated by either a user himself or an adversary. The probability of a user's participation in a block-signing round is directly proportional (not exponential) to the ratio of his Algorand coins to the total amount of coins in the system.

Decentralized: The Algorand users are homogeneous in their functionality. Aside from the ledger look-up, Algorand does not rely on any central or intermediary nodes, any servers or a universal time. There are no super- or seed-nodes/users for P2P messaging or to aid any of the Algorand's functionality. The underlying message delivery is P2P gossiping. Each user listens to the network and acts on the protocol rules using its local functionality without relying on intermediary nodes or external processes. Algorand's only time-wise requirement is that the clocks are ticking at the same intervals on all Algorand nodes.

The ledger: Algorand assumes a ledger much like that of Bitcoin's – a basic, temporal sequence of blocks where each block is a set of approved transactions. The ledger is "posted-on-the-sky" – is public for everyone to see without any restrictions. The ledger is fully decoupled from the protocol. That is, the structure and architecture of the ledger does not effect Algorand, nor does Algorand impose a specific structure on the ledger beyond described above. However, a fast-access ledger is more effective on Algorand's performance than the performance of many well-known blockchain protocols due to the player replacability aspect of this protocol. In a block signing round, every step is carried out by a different VC, and each member of VC needs to look up the ledger as part of his work.

Permissionless: anyone with a public key (PK) can join the system at any time at his sole will. Algorand knows the users by their public keys and public keys only – to Algorand, a user is nothing but a PK, and each PK is exactly one user. Recall that, the chances a user will participate in block signing are proportional to the amount of Algorand coins he is holding. The user's coins are not locked during the assignment of block signing roles.

Block-signing outcome: Algorand assumes the honesty of 2/3+ of the stake in order to achieve 50%+ accuracy of the outcome. The ratio of 2/3 here is a theoretical bound of a Byzantium protocol which Algorand is. This property of Algorand is integral to its decentralized structure, and this ratio of 2/3 can not be reduced. Algorand imposes a time limit, in two parameters, on the duration of such rounds. In the suggested scheme [CM17, M17], a round of block-signing takes less than 5 minutes, and a new block is approved in less than 3 minutes in the expected case. The likeliest result of a round is either the approval of a valid block, or an empty block which effectively means that the VCs couldn't agree on a legitimate block of valid transactions. In Algorand, a third possibility – the possibility of approval of a fake block and thus forks and double-spending are negligible as long as 2/3+ of the stake is in the honest hands.

Algorand Security

Algorand's security structure is totally internal in its dynamics – its security precautions do not effect its functionality to the outsider. This section outlines Algorand's security scheme for awareness.

An underlying factor of Algorand's security scheme is the time element: the role players in a block-signing round are targets of adversaries for the obvious reasons, and such role players are open to attacks from the time they are known in the network. Algorand's security structure is centered around this fact.

The following lays out Algorand's security scheme:

• *Cryptographic Sortition*: is a secret, random selection process for round-leader and VC selection. Random in the sense that the outcome of this process can not be influenced by the adversaries or the user itself (outside the ratio of the stakes he holds). Secret in the sense that each user finds out whether he has a role in the current round/step in private, without intervention of an external process. By this, no other user knows that a user is a participant in that round until the user starts acting in his role.

• *Unique Keys*: In order to deliver a user-specific role assignment decision, cryptographic sortition takes as input a specific signature (specific to the round/step, randomized by a seed) that is issued by the user's static key. Algorand restricts these static keys with the uniqueness property – so that a given key can generate exactly one signature that can be verified for that key. This is to disallow repetitive signing till the desired result during cryptographic sortition and thus letting computational power into Algorand.

• *Player Replacability*: A block signing round of Algorand consists of multiple steps. Each of these steps is carried out by a separate set of players. Each such player "speaks" once to fulfill its role and does not carry on participating beyond one step in a round. By this, Algorand sets aside a time slice that would allow an adversary to corrupt an honest user and manipulate him at his work. The VC members and round leader do not retain any round state that is potentially valuable to an adversary.

• *Ephemeral Keys*: All the work that goes into a block-signing round is signed by ephemeral keys – each round participant "seals" its outgoing message with a single-use key and not with his own, static key. Once the user signs his message to be sent to other round/step participants, he destroys the secret key component of this key in order to avoid re-signing another, fake block in case an adversary has had time to corrupt him eventually.

• *Look-back Parameter*: Algorand security restricts possibilities of a powerful adversary's potential moves to negligible amounts. Still, to set aside the remote possibility of an adversary successfully injecting in a role player that it controls in an upcoming round, Algorand restricts the domain of potential participants of the current round to the users that existed in the system the last $k \geq 40$ rounds, k being the look-back parameter.

REFERENCES

[CM17] Chen, J., Micali, S., (2017), *ALGORAND*, https://arxiv.org/pdf/1607.01341.pdf.
[M17] Micali, S., (2017), *"Algorand: A Better Distributed Ledger," with Silvio Micali*, ACM Channel on Youtube, https://www.youtube.com/watch?v=_nQE_HAGlmM&t=213s.