Algorand Consensus

Giuseppe Persiano

Università di Salerno

Blockchain

Algorand Consensus

Decentralized Byzantine Agreement protocol that leverages pure proof of stake (Pure PoS)

- can tolerate malicious users
- no need for a central authority, as long as a supermajority of the stake is in non-malicious hands.
- extremely fast (currently, one new block every $\approx$ 4 sec)
- requires minimal computational power per node

▸ Source: https://developer.algorand.org/docs/get-details/algorand_consensus/

Verifiable Random Functions

[Micali, Rabin, and Vadhan, 1999]

- A Cryptographic Primitive:
  - a public key $\mathtt{pk}$
  - a secret key $\mathtt{sk}$
- Given value x for PRF F and $(\mathtt{pk}, \mathtt{sk})$
  - it is possible to $y = F(\mathtt{pk}, x)$ and a proof Π that y is the correct value
- Given value (x, y, Π) and a public key $\mathtt{pk}$
  - it is possible to verify that y is the correct value
- Used to conduct a "lottery"
  - each user can check using the PRF if it selected to take part to consensus

PRF used by Algorand described

Participation Keys

- To take part in consensus, a user must be online
- This means that some secret key of the user should be given to a node
- Users should not use the *spending key* for consensus.
  If the node is compromised, the user could lose the algo
- Instead...

Participation Keys

- a user generates and registers a *participation key* for a certain number of rounds
- it also generates a collection of *ephemeral* keys, one for each round, signs these keys with the participation key, and then deletes the participation key.
- each ephemeral key is used to sign consensu messages for the corresponding round, and is deleted after the round is over.
- Using participation keys ensures that a user's algos are secure even if their participating node is compromised.
  Deleting the participation and ephemeral keys after they are used ensures that the blockchain is forward-secure and cannot be compromised by attacks on old blocks using old keys.

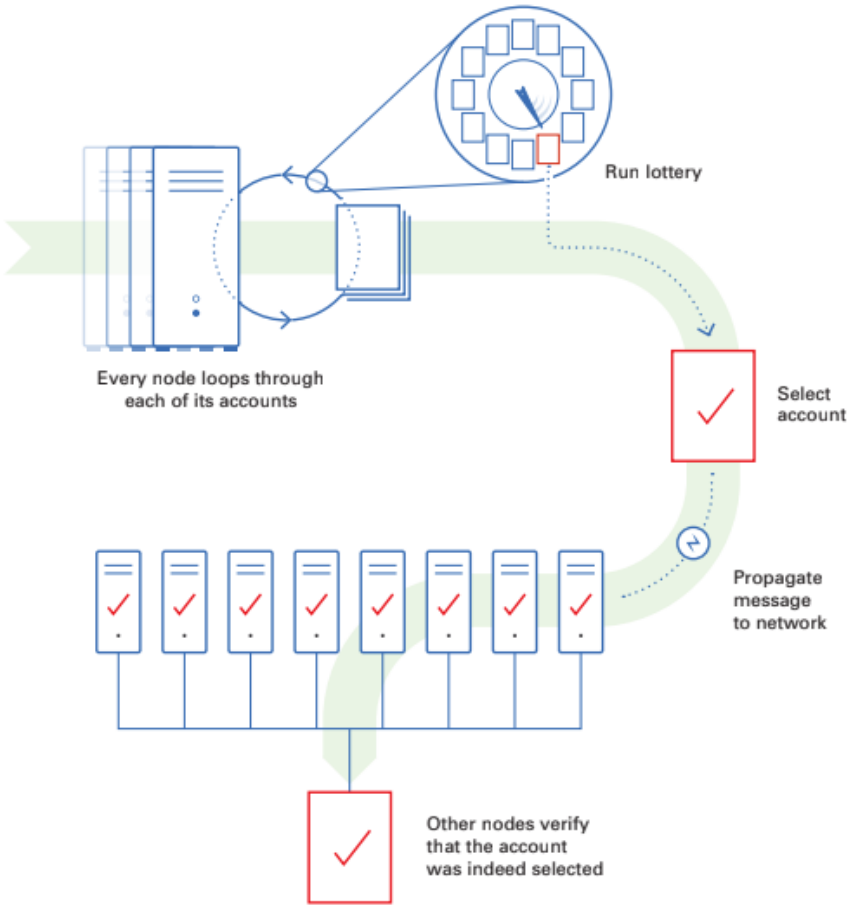
Consensus

1. Block Proposal
2. Soft Vote
3. Certify Vote.

All messages in each phase are signed with one of the ephemeral keys of the round.

I - Block proposal

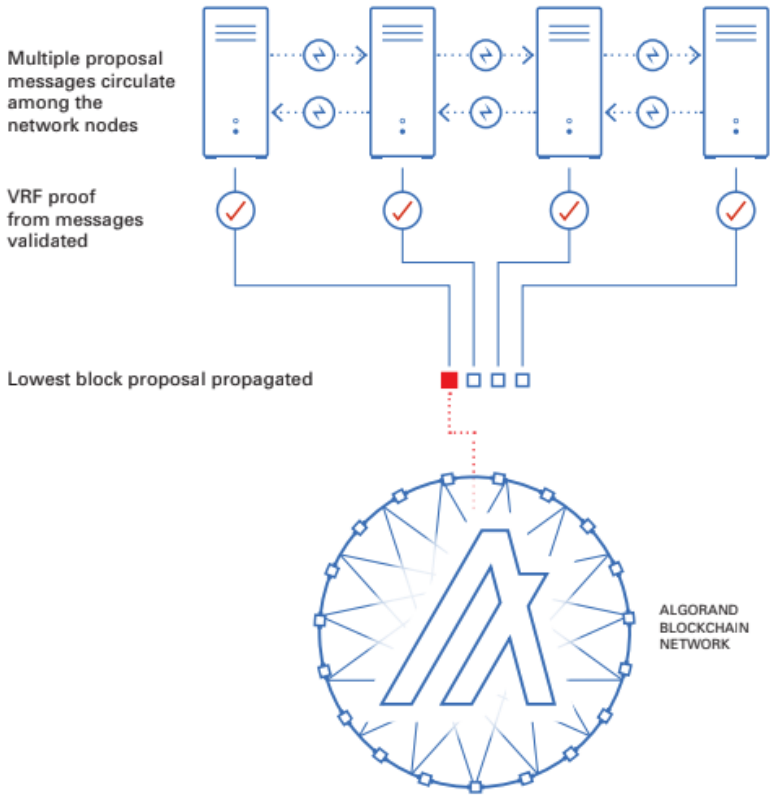
- A node has a certain number of accounts
- The node uses a PRF to run a lottery and select each account with probability proportional to number of tokens of the account
- Propagates selected accounts to other nodes and poposed block
- Nodes verify that account was selected by the PRF



II - Soft Vote

Filtering down the number of proposals to only block

- Each node get proposals from other nodes
- Verify the signature of the block and the selection through the VRF proof Π
- The block with the lowest PRF hash will be propagated
- This is done for a fixed amount of time



II - Soft Vote

- Each node runs VRF again to see if any of the accounts is chosen for the soft vote committee
- If any account is chosen it will have a weighted vote based on the number of algos the account has
- These votes will be for the lowest VRF block proposal calculated at the timeout and will be sent out to the other nodes along with the VRF Proof.
- A new committee is formed until a quorum is reached

III - Certify Vote

- A new committee checks if there is any overspending or other problem with the block proposed
- Committee is selected by using the PRF and each account has a weight equal to its number of algos