NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

Prof. Sandip Chakraborty

Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Lecture 42: Algorand

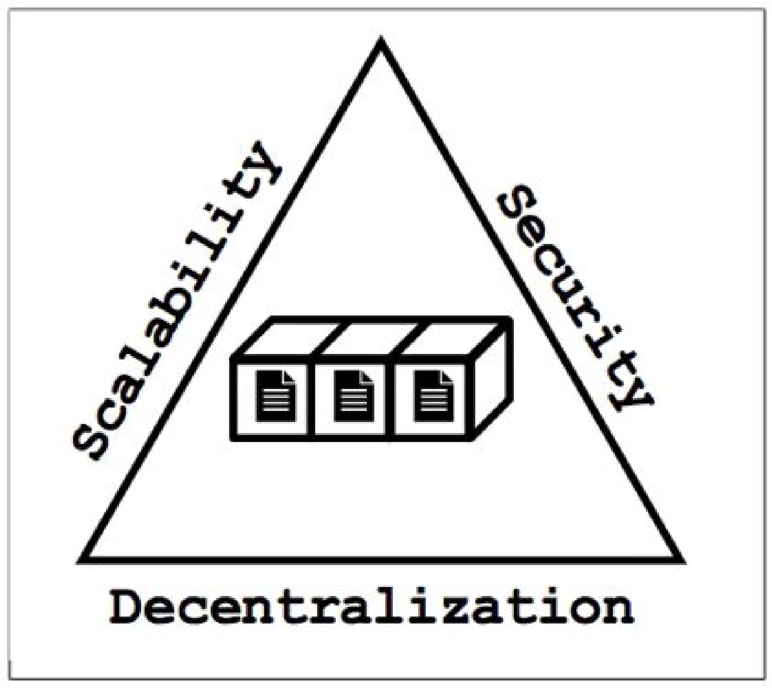
CONCEPTS COVERED

- **Algorand**

KEYWORDS

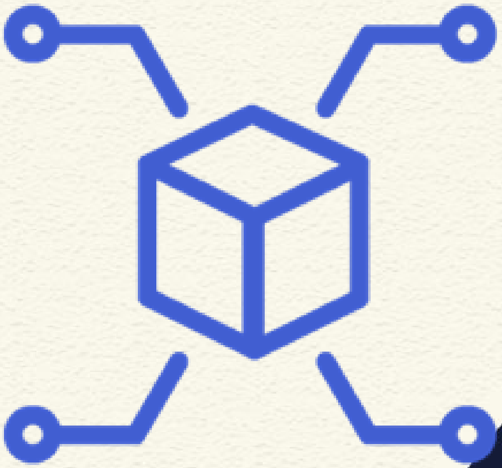
- **Cryptographic Sortition**
- **BA***

The Blockchain Performance Triangle



Is it ever possible to achieve all three simultaneously?

Algorand: Scaling Byzantine Agreements for Cryptocurrencies



Gilad, Y., Hemo, R., Micali, S., Vlachos, G., & Zeldovich, N. (2017, October). *Algorand: Scaling byzantine agreements for cryptocurrencies*. In *Proceedings of the 26th Symposium on Operating Systems Principles* (pp. 51-68). ACM.

Algorand: Overview

- **Key Idea**:
  - Consensus through Byzantine Agreement Protocol
- **Communication**:
  - Gossip protocol
- **Key Assumption**:
  - Honest majority of money

Algorand: Technical Advancement

- **Trivial computation**
  - simple operation like add, count
- **True decentralization**
  - no concentration of mining pool power, all equal miners and users
- **Finality of payment**
  - fork with very low probability, block appears, and the payment is fixed forever

Algorand: Technical Advancement

- **Scalability**
  - millions of users, only network latency (~1minute)
- **Security**
  - against bad adversary

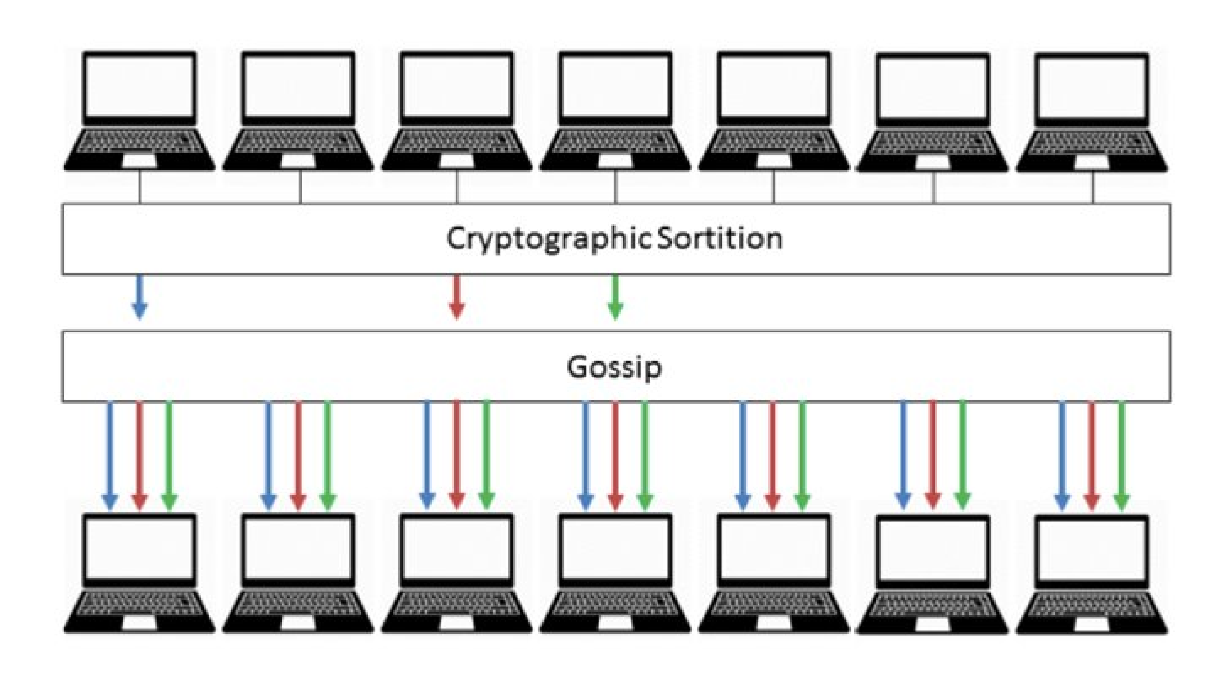
Architecture of Algorand

- Select a **random user**
  - prepare a block
  - propagate block through gossiping

- Select **random committee** with small number of users (~10k)
  - run Byzantine Agreement on the block
  - digitally sign the result
  - propagate digital signatures

- **Who select the committee?**

Cryptographic Sortition in Algorand



Cryptographic Sortition

- Each committee member selects himself according to per-user weights
    - Implemented using **Verifiable Random Functions** (**VRFs**)

- <hash,proof> ← $\mathrm{VRF}_{sk}(x)$
    - **x:** input string
    - **(pki,ski):** public/private key pair
    - **hash:** hashlenbit-long value that is uniquely determined by sk and x
    - **proof:** enables to check that the hash indeed corresponds to x

Committee Member Selection

<hash,proof,j> <--- Sortition(sk,seed,threshold,role,w,W)

- **seed:** publicly known random value
  - seed published at Algorand's round r using VRFs with the seed of the previous round r – 1
- **threshold:** determines the expected number of users selected for that role
- **role:** user for proposing a block/ committee member
- **w:** weight of a user
- **W:** weight of all users
- **j:** user gets to participate as j different "sub-users."

Byzantine Agreement in Algorand: BA*

- **Two phase**:
  - Two phase agreement –
    - *Final Consensus*
    - *Tentative Consensus*

Byzantine Agreement in Algorand: BA*

- **Strong Synchrony**: Most honest users (say, 95%) can send message that will be received by most other honest users within a known time bound
  - Adversary can not control the network for long
  - Ensures liveness of the protocol

Byzantine Agreement in Algorand: BA*

- **Weak Synchrony**: The network can be asynchronous for long (entirely controlled by adversary) but bounded period of time
  - **There must be a strong synchrony period after a weak synchrony period**
  - Algorand is **safe** under weak synchrony

Final Consensus

- One user reaches final consensus
    - Any other user that reaches final or tentative consensus in the same round must agree on the same block value (**ensures safety**)
    - Confirm a transaction when the block reaches to the final consensus

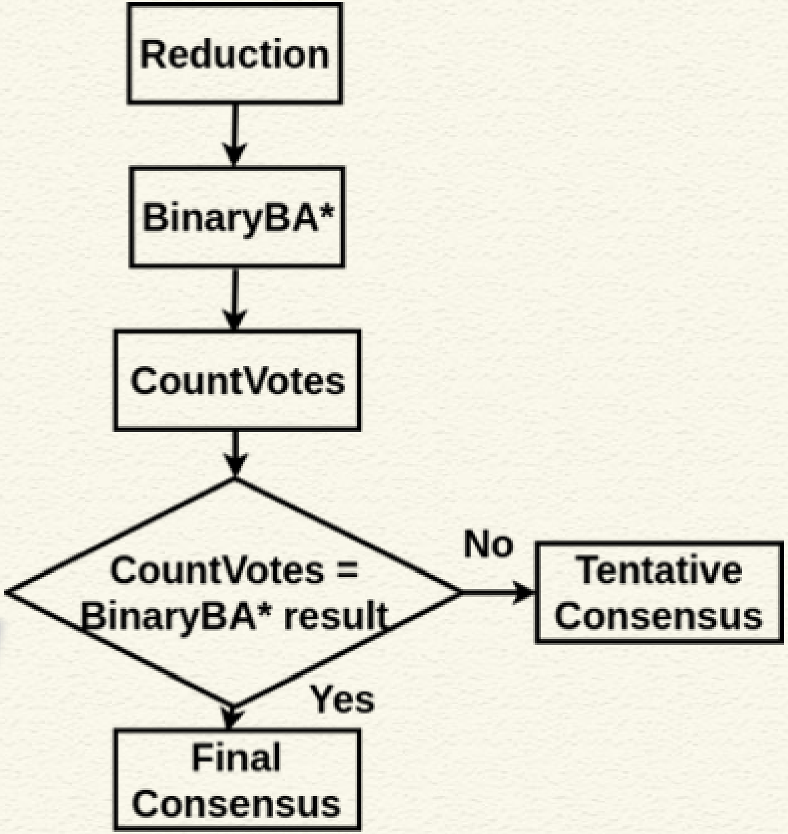
Tentative Consensus

- One user reaches tentative consensus
  - Other users may have reached consensus on a **<u>different (but correct)</u>** block
  - Can be in two cases
    - **<u>The network is strongly synchronous</u>** - adversary may be able to cause BA* to reach tentative consensus on a block - BA* is unable to confirm that the network was strongly synchronous
    - **<u>The network was weakly synchronous</u>** - BA* can form multiple forks and reach tentative consensus on two different blocks - users are split into groups

Coming out of Tentative Consensus

- Run BA* periodically to come out of tentative consensus - run the next round
  - Network cannot be under weak synchrony all the times
  - Cryptographic sortition ensures different committee members at different rounds of the BA*

BA* Overview


Reduction
BinaryBA*
CountVotes
CountVotes = BinaryBA* result
No
Tentative Consensus
Yes
Final Consensus

Conclusion

- Algorand has multiple advantages
  - Bitcoin like scalability
  - BFT like throughput
  - No fork

- **Caution: Needs a really large network**


Thank you