FINE 434: FinTech

Lecture 21

Professor Fahad Saleh

McGill University - Desautels

Founded by cryptography pioneer and Turing award winner, Silvio Micali, Algorand solves the "blockchain trilemma" with a platform that delivers decentralization, scalability and security. Algorand provides a foundation for existing businesses and new projects to operate globally in the emerging decentralized economy. Algorand's first-of-its-kind, permissionless, pure proof-of-stake protocol supports the scale, open participation, and transaction finality required to build systems for billions of users.

Decentralization arises by construction.

What about security and scalability?

Security

ALGORAND*

Jing Chen
Computer Science Department
Stony Brook University
Stony Brook, NY 11794, USA
jingchen@cs.stonybrook.edu

Silvio Micali
CSAIL
MIT
Cambridge, MA 02139, USA
silvio@csail.mit.edu

Validation under standard computer science assumptions...

Scalability

Scalable and Secure

Algorand's innovative protocol enables transaction throughput velocity on par with large payment and financial networks and maintains that performance while securely scaling to billions of users.

How do we overcome intrinsic issues of a distributed system?

Primary Concerns

- Forks
- Latency

Precluding Forks

What if we define block validity so that only valid blocks ever enter the blockchain?

The Byzantine Generals Problem

LESLIE LAMPORT, ROBERT SHOSTAK, and MARSHALL PEASE
SRI International

Algorand precludes forks (with high probability) by employing "byzantine agreement"

Byzantine Generals Problem

N Generals: 1 Commander and $N - 1$ Lieutenants

The Generals need to decide on whether to attack or not

Some generals may be malicious

Byzantine Generals Problem

N Generals: 1 Commander and $N - 1$ Lieutenants

The Generals need to decide on whether to attack or not

Some generals may be malicious

A Block Proposer is akin to the Commander

A Node is akin to a General

Attacking is akin to accepting the proposed block

How do we solve the problem?

We need sufficiently many honest generals

Why would we expect honest generals?

software. As mentioned earlier, Algorand assumes that the fraction of money held by honest users is above some threshold h (a constant greater than 2/3), but that an adversary can participate in Algorand and own some money. We believe that this assumption is reasonable, since it means that in order to successfully attack Algorand, the attacker must invest substantial financial resources in it. Algorand assumes

An Example

Try 7 generals with 2 malicious Lieutenants

`https://marknelson.us/posts/2007/07/23/byzantine.html`

Intuition
The majority of majorities gives honesty because the majority of a sufficiently large subset is honest

What about latency?

Communicating across many nodes poses difficulties

... but this is not a new problem

Akin to government, Algorand “elects” a committee thereby making latency manageable

“[Algorand] achieves scalability by choosing a committee — a small set of representatives randomly selected from the total set of users — to run each step of its protocol”

Committee Selection

"[Algorand] selects committee members in a private and non-interactive way"

"every user in the system can independently determine if they are chosen to be on the committee, by computing a function of their private key and public information from the blockchain"

Do you see any problems with this scheme?

Committee Selection

"[Algorand] selects committee members in a private and non-interactive way"

"every user in the system can independently determine if they are chosen to be on the committee, by computing a function of their private key and public information from the blockchain"

Do you see any problems with this scheme?
Learning by Doing...

Overview

- Gossip protocol
- Block proposal
- Agreement

Gossip Protocol

"Algorand implements a gossip network (similar to Bitcoin) where each user selects a small random set of peers to gossip messages to. To ensure messages cannot be forged, every message is signed by the private key of its original sender; other users check that the signature is valid before relaying it."

This "gossip protocol" determines network topology which in turn affects ability to scale. (Why?)

Block Proposal

Cryptographic Sortition

“a small fraction of users are selected at random, weighed by their account balance”

“each selected user [generates] a priority, which can be compared between users, and a proof of the chosen user’s priority”

“Since sortition is random, there may be multiple users selected to propose a block, and the priority determines which block everyone should adopt.”

Block Proposal: More Concretely

procedure Sortition($sk, seed, \tau, role, w, W$):

$\langle hash, \pi \rangle \leftarrow \text{VRF}_{sk}(seed||role)$

$p \leftarrow \frac{\tau}{W}$

$j \leftarrow 0$

while $\frac{hash}{2^{hashlen}} \notin \left[\sum_{k=0}^{j} B(k;w,p), \sum_{k=0}^{j+1} B(k;w,p)\right)$ **do**

  $j{+}{+}$

return $\langle hash, \pi, j \rangle$

Algorithm 1: The cryptographic sortition algorithm.

sk : secret/ private key

w : user's wealth/ weight

W: "total amount of currency units"

$$B(k;w,p) = \binom{w}{k} p^k (1-p)^{w-k}$$

"Sortition requires a role parameter that distinguishes the different roles [for which] a user may be selected"

Priority

Note that multiple users may be selected and that a user may be selected multiple times!

A “priority” is computed.

“Algorand users discard messages about blocks that do not have the highest priority”

Agreement

Algorand generates a committee using sortition.

This means that committee size is random.

Committee members “vote” on block proposer’s block

Voting occurs in 2 steps

Voting: Step 1

"As soon as one [block] has more than $T \times \tau$ votes, [the first round ends and this block moves to the second round.]"

"τ is the expected number of users that Sortition() selects for the committee"

"T is a fraction of that expected committee size $(T > \frac{2}{3})$ that defines [the] voting threshold"

There is a time-out in case no block clears the threshold...

Voting: Step 2

If Step 1 times out then Step 2 takes an empty block.

Committee votes on whether to accept proposed block

Multiple possible rounds... but no guarantee

Details, Details...

```
// No consensus after MaxSteps; assume network
// problem, and rely on §8.2 to recover liveness.
HangForever()
```

Algorithm 8: Binary*BA*⋆ executes until consensus is reached on either *block_hash* or *empty_hash*.

There exists a resolution process, but “when Algorand is recovering outside of a strongly synchronous period, we cannot ensure recovery within *s* time.”

Parameter	Meaning	Value
h	assumed fraction of honest weighted users	80%
R	seed refresh interval (# of rounds)	1,000 (§5.2)
τ_{PROPOSER}	expected # of block proposers	26 (§B.1)
τ_{STEP}	expected # of committee members	2,000 (§B.2)
T_{STEP}	threshold of τ_{STEP} for $BA\star$	68.5% (§B.2)
τ_{FINAL}	expected # of final committee members	10,000 (§C.1)
T_{FINAL}	threshold of τ_{FINAL} for $BA\star$	74% (§C.1)
MaxSteps	maximum number of steps in Binary$BA\star$	150 (§C.1)
$\lambda_{\text{PRIORITY}}$	time to gossip sortition proofs	5 seconds
λ_{BLOCK}	timeout for receiving a block	1 minute
λ_{STEP}	timeout for $BA\star$ step	20 seconds
λ_{STEPVAR}	estimate of $BA\star$ completion time variance	5 seconds

Decisions

What are the advantages of large committees?

What are the advantages of large thresholds?

What are the advantages to long time-outs?

Is something missing in the analysis?

How would you set tuning parameters?