GOAL
PoS blockchain consensus providing

- **Fast Consensus** (5 seconds)
- **Scalability** (1m participants)
- **Randomness**
- **Security** (provable)
PoS blockchain consensus providing

- **Fast Consensus** (5 seconds) → **Short block time, Instant Finality**
- **Scalability** (1m participants)
- **Randomness**
- **Security** (provable)
GOAL

PoS blockchain consensus providing

- **Fast Consensus** (5 seconds) → **Short block time, Instant Finality**
- **Scalability** (1m participants) → **Unpredictable, Unmanipulable**
- **Randomness**
- **Security** (provable)
POSS blockchain consensus providing

- **Fast Consensus** (5 seconds): Short block time, Instant Finality
- **Scalability** (1m participants): Unpredictable, Unmanipulable
- **Randomness**
- **Security** (provable): Selfish-mining, Nothing-at-stake
CHALLENGES
CHALLENGES:
Consensus & Scalability

Instant Finality

TRADE-OFF

Scalability
CHALLENGES:

Consensus & Scalability

Instant
Finality

PBFT chain

TRADE-OFF

Scalability

PoW chain
CHALLENGES: Consensus & Scalability

**Instant Finality**
- PBFT chain
  - Finality at every block
  - Few nodes

**Scalability**
- PoW chain
  - Probabilistic finality over time
  - Arbitrary many nodes
Derive randomness from a chain?

Chain is not random, manipulable

Assumes everybody agrees on the chain
Derive randomness from a chain?

**CHALLENGES:**

**Randomness**

Chain is not random, manipulable

Assumes everybody agrees on the chain

Must not depend on chain content (“grindable”)

Must not fork
CHALLENGES:
Randomness

“LAST ACTOR” BIAS

The “last actor” sees the randomness and aborts.

Any fallback mechanism introduces bias.
CHALLENGES: Randomness

“LAST ACTOR” BIAS

The “last actor” sees the randomness and aborts.

Any fallback mechanism introduces bias.

EXAMPLES:

- Miner discards block
- Commit-reveal schemes
- 2 or more parties involved
- ...
CHALLENGES:

Security

PoS ATTACKS

- Signatures are timeless
- Blocks can be withheld
- Selfish-mining
  Nothing-at-stake
- Signatures are cheap
- Blocks can be equivocated

Nothing-at-stake
FUNDAMENTAL OBSERVATIONS
Threshold groups do not have “last actors”
FUNDAMENTAL OBSERVATIONS:

No “last actors”

**k-of-n THRESHOLD GROUP**

k out of n members have to act

(necessary and sufficient)
Consensus without consensus
FUNDAMENTAL OBSERVATIONS:
Consensus without consensus
FUNDAMENTAL OBSERVATIONS:
Consensus without consensus

BLS Threshold signatures

Distributed key generation:
No trusted dealer required
FUNDAMENTAL OBSERVATIONS:

Consensus without consensus

BLS Threshold signatures

- Distributed key generation: No trusted dealer required
- Unique: For every message, only one signature verifies
FUNDAMENTAL OBSERVATIONS:

Consensus without consensus

BLS Threshold signatures

- Distributed key generation: No trusted dealer required
- Unique: For every message only one signature verifies
- Non-interactive: Signature shares created independently
FUNDAMENTAL OBSERVATIONS:

Consensus without consensus

- **BLS Threshold signatures**
  - Distributed key generation: No trusted dealer required
  - **Unique:** For every message only one signature verifies
  - **Non-interactive:** Signature shares created independently

“agree once”

“agree always”
Randomly sampled committees
FUNDAMENTAL OBSERVATIONS:
Randomly sampled committees

Universe → Random Sample
FUNDAMENTAL OBSERVATIONS:
Randomly sampled committees

Universe
30%

Random Sample
FUNDAMENTAL OBSERVATIONS:

Randomly sampled committees

Universe
30%

Random Sample
50% ?

Universe

Random Sample

BPASE 2018 • 28
FUNDAMENTAL OBSERVATIONS:

Randomly sampled committees

Universe
30%  

Random Sample
50%  ?
FUNDAMENTAL OBSERVATIONS:
Randomly sampled committees

Universe
30%

Random Sample
50% ?
**FUNDAMENTAL OBSERVATIONS:**

**Randomly sampled committees**

Group G *honest* if adversary controls <50% of G

<table>
<thead>
<tr>
<th></th>
<th>Prob(G honest) &gt; $1 - 2^{-x}$</th>
<th>Safe group size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>adversary controls</td>
</tr>
<tr>
<td></td>
<td></td>
<td>33.3%</td>
</tr>
<tr>
<td>$x$</td>
<td></td>
<td>of universe</td>
</tr>
<tr>
<td>40</td>
<td></td>
<td>423</td>
</tr>
<tr>
<td>64</td>
<td></td>
<td>701</td>
</tr>
<tr>
<td>80</td>
<td></td>
<td>887</td>
</tr>
<tr>
<td>128</td>
<td></td>
<td>1447</td>
</tr>
</tbody>
</table>

Based on Binomial distribution (arbitrarily large universe)
FUNDAMENTAL OBSERVATIONS

Enforcing publication
FUNDAMENTAL OBSERVATIONS:
Enforced publication

NOTARIZATION

Majority signature by a committee.
FUNDAMENTAL OBSERVATIONS:
Enforced publication

**NOTARIZATION**

Majority signature by a committee.

A committee is a random sample of the whole network.

A notarization is a proof of publication.
FUNDAMENTAL OBSERVATIONS:
Enforced publication

**NOTARIZATION**

Majority signature by a committee.

**VALID**

Only notarized blocks are valid.

A committee is a random sample of the whole network.

A notarization is a proof of publication.
FUNDAMENTAL OBSERVATIONS:

Enforced publication

1  2  3  4  5  Block height
FUNDAMENTAL OBSERVATIONS:

Enforced publication

1  2  3  4  5  Block height

G_1  G_2  G_3  G_4  G_5  Sequence of committees
FUNDAMENTAL OBSERVATIONS:
Enforced publication

<table>
<thead>
<tr>
<th>Block height</th>
<th>Time interval</th>
<th>Sequence of committees</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>G_1</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>G_2</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>G_3</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>G_4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>G_5</td>
</tr>
</tbody>
</table>
A committee is **active only** during the phase of the block height to which it was assigned.

A notarization is a **timestamp**.
FUNDAMENTAL OBSERVATIONS:
Enforced publication

Only notarized blocks are valid.

Blocks cannot be withheld.
Impossible to build on old blocks.
FUNDAMENTAL OBSERVATIONS:
Enforced publication

Can notarizations be withheld?
A committee is active only during the phase of the block height to which it was assigned.

A block can get “half-notarized”.
FUNDAMENTAL OBSERVATIONS:

Enforced publication
FUNDAMENTAL OBSERVATIONS:

Enforced publication

2b  3c

Every notarization contains honest signatures.
FUNDAMENTAL OBSERVATIONS: Enforced publication

Every notarization contains honest signatures.
FUNDAMENTAL OBSERVATIONS:
Enforced publication

Every notarization contains honest signatures.

Trick:
Block references the previous notarization
Notarizations cannot be withheld
FUNDAMENTAL OBSERVATIONS:

Enforced publication

Block height

Time interval

G_1  G_2  G_3  G_4  G_5  

Active group

Notarized Chains

1  2  3  4  5

2a  3a  4a  2b  3b  4b  5

1  3  2  3  5

FUNDAMENTAL OBSERVATIONS:

Enforced publication
FUNDAMENTAL OBSERVATIONS:

Enforced publication

1 2 3 4 5

Block height

Time interval

G_1  G_2  G_3  G_4  G_5

Active group

Notarized Chains

Enforced publication
FUNDAMENTAL OBSERVATIONS:

Enforced publication

Block height

Time interval

Active group

Notarized Chains

1 2 3 4 5

G_1  G_2  G_3  G_4  G_5

1 2a 3a 4a 5

2b 3b 4b 5

3c
THE SYSTEM
SOLUTION

- Unique threshold signatures → Randomness
- Committees → Scalability
- Notarization → Provable security
  Fast consensus
SOLUTION:
Protocol Layers

Consensus Protocol Layers

- Notary Layer
- Blockchain Layer
- Random Beacon Layer
- Identity Layer

External Actors

- Observers
- Users
- New Clients

Actions:
- Observe
- Transaction
- Register
SOLUTION: Rounds

1. Random value

2. Block proposals

3. Notarized Block

Random Beacon

Notary

Block Makers

SOLUTION:
SOLUTION:
Random Beacon
SOLUTION:
Random Beacon
SOLUTION:

Random Beacon
SOLUTION:
SOLUTION:
Blockchain
SOLUTION:

Blockchain
SOLUTION:

Blockchain
SOLUTION:
Blockchain
FUNDAMENTAL OBSERVATIONS:

Normal operation

First block maker **honest** + Network **synchronous** → Only one block gets notarized
SOLUTION:

Blockchain

```
0
FINAL

1
DEAD

2
DEAD
```

```
r - 3  r - 2  r - 1  r
```

```
FINAL  X  FINAL  X
```

D F I N I T Y
CONCLUSIONS
Consensus algorithm landscape

BYZANTINE AGREEMENT IN AUTHENTICATED SETTING

(n replicas, f adversary, asynchrony)

\[ f = 0 \]
Consensus algorithm landscape

BYZANTINE AGREEMENT IN AUTHENTICATED SETTING

(n replicas, f adversary, asynchrony)

\[ f = 0 \quad n > 3f \]
Consensus algorithm landscape

BYZANTINE AGREEMENT IN AUTHENTICATED SETTING

(n replicas, f adversary, asynchrony)

- $f = 0$
- $n > 3f$
- $n > 3f$
  + synchrony
  (for liveness)

FLP  Bracha/ Toueg  PBFT

probabilistic  practical
Consensus algorithm landscape

BYZANTINE AGREEMENT IN AUTHENTICATED SETTING

(n replicas, f adversary, asynchrony)

$f = 0$ $n > 3f$ $n > 3f$ $n > 2f$

+ synchrony (for liveness) + synchrony + probabilistic termination

probabilistic practical

FLP Bracha/ Toueg PBFT Blockchain
Consensus algorithm landscape

BYZANTINE AGREEMENT IN AUTHENTICATED SETTING

\( (n \text{ replicas, } f \text{ adversary, asynchrony}) \)

\[
\begin{align*}
  f &= 0 & n &> 3f \\
  n &> 3f & n &> 2f + \epsilon \\
  + \text{ synchrony} & \quad \text{for liveness) } & + \text{ synchrony} & \quad \text{for liveness & safety) } & + \text{ synchrony} & \quad + \text{ probabilistic termination} \\
\end{align*}
\]

- probabilistic
- practical
- scale

FLP \quad Bracha/Toueg \quad PBFT \quad Dfinity \quad Blockchain
Consensus algorithm landscape

BYZANTINE AGREEMENT IN AUTHENTICATED SETTING

\((n \text{ replicas, } f \text{ adversary, asynchrony})\)

\(f = 0\)

\(n > 3f\)

\(n > 3f\)

\(n > 2f + \epsilon\)

\(n > 2f\)

\(-\text{ probabilistic}\)

\(-\text{ practical}\)

\(-\text{ scale}\)

\(-\text{ fast termination}\)

- FLP
- Bracha/Toueg
- PBFT
- Dfinity
- Blockchain
Algorithm flexibility

PBFT
(synchrony only for liveness)
Algorithm flexibility

DFINITY (synchrony for liveness and safety)

PBFT (synchrony only for liveness)

\[ n/5 \quad n/3 \quad n/2 \]
Algorithm flexibility

DFINITY (synchrony only for liveness)

DFINITY (synchrony for liveness and safety)

PBFT (synchrony only for liveness)
Algorithm flexibility

- **DFINITY (synchrony only for liveness)**
- **DFINITY (synchrony for liveness and safety)**
- **PBFT (synchrony only for liveness)**
Open participation

Key Frame Block  
\[ e - 3 \]

Join tx: Group
0x2b197453

Join tx: Client
0x6e22e1ba

Key Frame Block  
\[ e - 2 \]

Key Frame Block  
\[ e - 1 \]

Key Frame Block  
\[ e \]

Activation Group
0x2b197453
Client
0x6e22e1ba

Chain Head
Randomness defines the new groups
The group members run distributed key generation
They join by providing their group public key
Challenge

How to agree on the group public key?
Solution

Use Dfinity chain at the current epoch to change the behavior of the system itself at future epochs.
Requires Agreement: Dfinity chain has already created an efficient medium for consensus

Using the same blockchain: the result will be used only on the side of a fork on which it is used
DKG Contract
Create a master public key and set of partial signing keys.
Previous versions of the DKG

- Studied extensively in the crypto literature
- Utilizes point-to-point communication
- Assumes/implements a reliable broadcast channel as a bullet-in-board
- We use Dfinity chain as the bullet-in-board
DKG phases

∞ It goes through the following phases
1. Dealing
2. Complaint
3. Justification
4. Registration

∞ Prefer contracts that are event driven

{-# LANGUAGE ScopedTypeVariables, FlexibleContexts #-}
module Main where

import Data.List (delete)
import Control.Monad
import Control.Monad.Task
import Contract
import System

type VerificationVector = String

data DkgEvent = Deal Int VerificationVector
              | Complain Int Int
              | Justify Int Int VerificationVector
              | GetGroupKey
              deriving (Eq, Show)
Dealing phase

Many dealers (contributors).

Each Dealer will

- Share a secret value using a polynomial
- Create commitment to the polynomial
- Encrypt the shares
- Send the vector to the contract

```
collectDeals deal@(i,_) deals =
case lookup i deals of
  Just _ -> logConsole "deal already exists", i >> return deals
  Nothing -> return (deal : deals)
```
Complain phase

- Receive your share and commitment vector
- Check the validity of your share
- Send your complain to the contract if:
  - No share received
  - The share was invalid
Justification phase

For each complaint:

- Send the share in clear text to the contract

For each justification, the contract:

- Check if the share is valid
- If it is valid, it removes the complain and accepts the sharing.
Registration phase

- The group public key is the sum of the verification vector of each successful dealer

- **Successful Dealer:** not have any unjustified complains remaining
System contracts can be used to manage the future behavior of the system.

Carefully design the contract that most of the code which do not need consensus is happening on the client side.

Only the data that need agreement over will pass to the contract.
THANK YOU

dfinity.org

twitter.com/dfinity

dfinity.org/tech

bit.ly/DFN_TestNet

medium.com/dfinity

reddit.com/r/dfinity