



-- Provide horizontal scalability and high transaction throughput.

Sunday, 09.21.2018

Contributions:

1. **Efficient Distributed Sharding (EDS)**
2. **TransEpoch**
3. **Atomix**
4. **Proof-of-Believability (PoB)**
5. **Micro State Block (MSB)**

## PoW

Pros: is its ability to defend against Sybil attack in a permissionless setting.

Cons:

- (1) unlike other modern cryptocurrencies, it takes more than an hour to confirm a transaction according its configuration;
- (2) it is difficult to develop various applications upon Bitcoin network;
- (3) the consensus mechanism wastes too much energy

=> Split data and state into shards

## Sharding challenge:

1. How to divide the system into shards.
2. How to reach consensus in each shard.
3. How to perform inter-shard transactions.

## Distributed Randomness Protocol

1. Main Method: we propose to use a client-server protocol, called [Distributed Randomness Protocol](#) (DRP), where a client communicates with a set of servers to generate an unforgeable, uniformly random value through [non-interactive zero-knowledge proof](#) (NIZK) and [publicly verifiable secret sharing](#) (PVSS).
2. In the reference paper: when sharding 512 nodes into groups of 32, our experiments show that RandHound can produce fresh random output after 240 seconds. RandHerd, after a setup phase of 260 seconds, is able to generate fresh random output in intervals of approximately 6 seconds. For this configuration, both protocols operate at a failure probability of at most 0.08% against a Byzantine adversary.
3. Two phases: Randomness generation and randomness verification.
  - a. Initially, the client starts a protocol run by broadcasting to all the servers a message including a randomly generated balanced grouping.
  - b. In the first phase, each server generates a random input value and creates shares only for other members of the same group using PVSS. Upon receiving encrypted shares with the NIZK proofs from all the servers (or timeout), the client chooses a subset of server inputs from each group. This allows the client fix each group's secret and the output of the protocol.
  - c. In the second phase, the servers decrypt and send their shares to the client as soon as the client receives a sign-off on input values in a global run of collective signature

(CoSi). Then the client combines the recovered group secrets to reveal the final random output.

## Efficient Distributed Sharding

The DRP protocol only works well without malicious or failure nodes, since it is performed by validators collectively.

=> Backup protocols: Omniledger solution.

## Operability During Epoch Transitions -- shard configuration scheme

- IOSChain uses a dynamical rolling scheme – it swaps out and in validators in batches for each epoch
- A key factor of the issue during the transition is the batch size, which is highly relevant to the safety of the system.
- Given our threat model that there are at most  $1/3$  malicious nodes, the maximum size of the swap batch should be less than  $\frac{1}{3}$  nodes.
- Use Omniledger solution: use the method of selecting a subset of the validators to be swapped out and replaced with new members

## Inter-Shard Transactions


- Introduce a Byzantine Shard Atomic Commit (Atomix) protocol, variant of the Omniledger algorithm
- In a nutshell, when a cross-shard transaction from node a at shard A to node b at shard B happens, the algorithm does the following:

- Create the TX within the shard **A** and let all nodes validate the transaction.
- If the TX is approved by all nodes in shard **A**, the transaction is logged in **A**'s blockchain. At the same time, the client will gossip a proof-of-acceptance to endorse the transaction, lock the fund of **a** into a UTXO, and send it to **B**. If the TX is rejected by nodes in **A**, the fund gets returned to **a**.
- **A**'s blockchain commits the TX to the **B**'s blockchain and have nodes in the receiver's shard validating the TX. If the TX is rejected by nodes in **B**, the fund gets returned to **a**.
- If the TX gets approved by all nodes in the **B**'s blockchain, the fund is released to **b**. If the TX is rejected by all nodes, the fund gets returned to **a**.

**Tokens and Motivations** -- IOS also plays a important role in calculating a user's believability score

**Proof-of-Believability:** The protocol divides all validators into two groups, **a believable league and a normal league**. Believable validators process transactions quickly in the first phase. Afterwards, normal validators sample and verify the transactions in the second phase to provide finality and ensure verifiability. The chance of a node being elected into the believable league is determined by believability score which is calculated by multiple factors (e.g., token balance, contributions to the community, reviews, etc). One with higher believability score is more likely to be elected into the believable league.

???: Believable validators also form smaller groups – one validator per group.



Corruption: we specify a sampling probability  $p$  that normal validators will sample transactions and detect inconsistencies. If a validator is detected as misbehaviour, it will lose all the tokens and reputation in the system

If a validator is detected as misbehaving, that validator will lose all tokens and reputation in the system and all its previously validated transactions will be re-checked.

When a shard doesn't have enough believable validators to form the league, due to either temporary downtime or being in the bootstrapping phase of the ecosystem, two-league committees would fall back to one-league.

### Storage:

Use Micro State Blocks (MSB), which is based on the State Block.  
Based on Omniledger.