Availability and Validity of Data in Sharded Blockchains

AnV is a blockchain sharding scheme through which a single beacon (relay chain) can validate numerous shard chains, with true shared security, minimal latency, good liveness, and relative efficiency in terms of computation, network, and storage.

Jeffrey Burdges, Handan Kilinc*, Alistair Stewart

Problem

Consider a blockchain consisting of multiple heterogeneous shards with full nodes responsible for each shard and light clients who do not have access to full states of these shards. How can light clients validate the transactions of shards in the relay chain with a limited knowledge?

Our Model

→ We call light clients validators. Validators are responsible to maintain a blockchain (relay chain) to validate the transactions of shards.

→ Relay chain has a secure proof-of-stake based blockchain with a BFT consensus protocol.

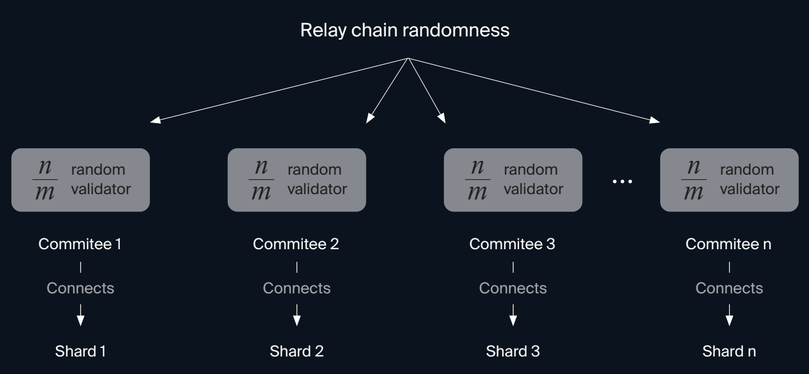
→ Relay chain generates a pseudo-random number at the end of every epoch.

→ A validator is able to verify transactions in the block of a shard with an algorithm Verify which receives a block and a proof as an input.

→ We have $n = 3f + 1$ validators where at most f validators are malicious and $m < n$ shards.

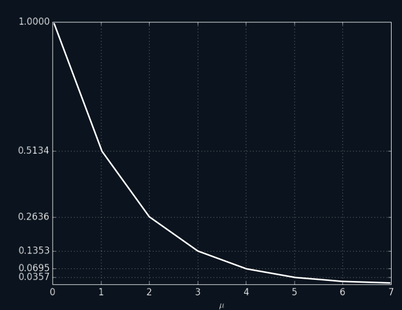
AnV Scheme

Pseudo-random number generated in the relay chain samples committees of n/m validators for each shard periodically. Each committee is responsible to check the validity of blocks of their shards.

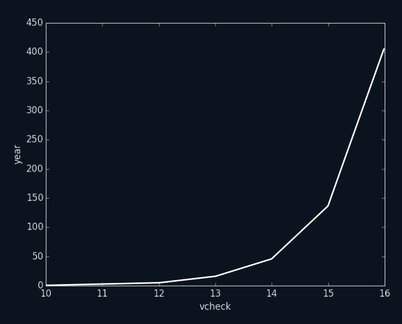


Security

The security is based on the fact that a block can be finalized with $2f + 1$ votes. The security of the validity based on the obscurity of extra validators that are assigned to check and having all responsible validators malicious.



Probability of attack when #Vcheck validators are malicious. $\#Vcheck = |C| + \mu$



The waiting time for an adversary for a successful attack given that commitee changes every 5 minutes.

Comitee Side

1 Block creation in the shard

$Shard_i$

b = b_{head}, b_{body}

$\texttt{Prove}(Shard_i, \texttt{b}) \rightarrow proof$

$D \leftarrow (b, proof)$

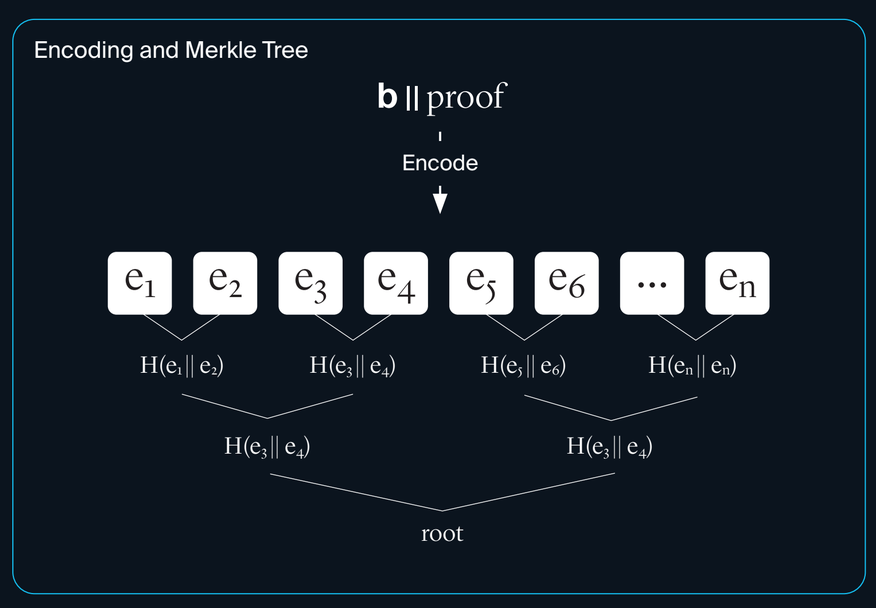
→ A full node of a shard generates a block, generates a proof of validity and forwards the block and the proof to its committee.

2 Verification and distribution of block header

→ Each validator in the committee checks the validity of the block.
- If the block is in valid, it discards.
- Otherwise, it generates erasure code pieces of the block and the proof for each validator in the relay chain. They can be reconstructed with at least $f + 1$ erasure code pieces. A validator distributes the pieces with signatures.

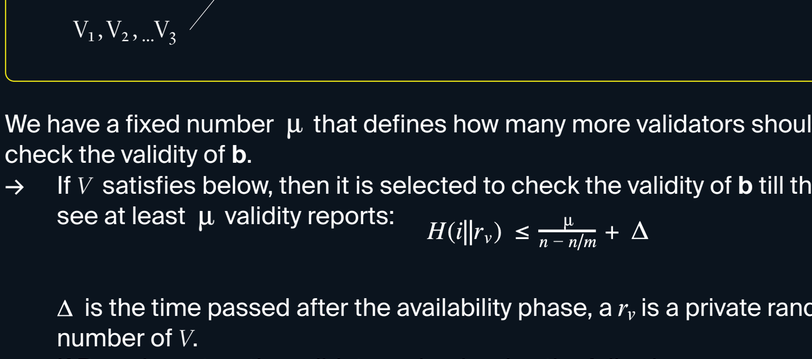
Commitee C_i

each V **in** C_i:
  if `Verify(D)`:
    `Encode(D,n,f+1)` → $e_1, e_2, ..., e_n$
    `MerkleTree(`$e_1, e_2, ..., e_n$`)` → $root, tree$
    `Sign(`$b_{head}, root$`)` → σ
    Send σ **to** $V_j \in C_i$
V **in** C_i:
  Send $\{\sigma\}, b_{head}, root, (e_j, MTproof)$ **to all** V_j



Relay Chain Side

When a block **B** including the block header **b** from a shard is generated, the validators vote for the block using the finality gadget. Before doing it, they need to make sure that the block header is valid.

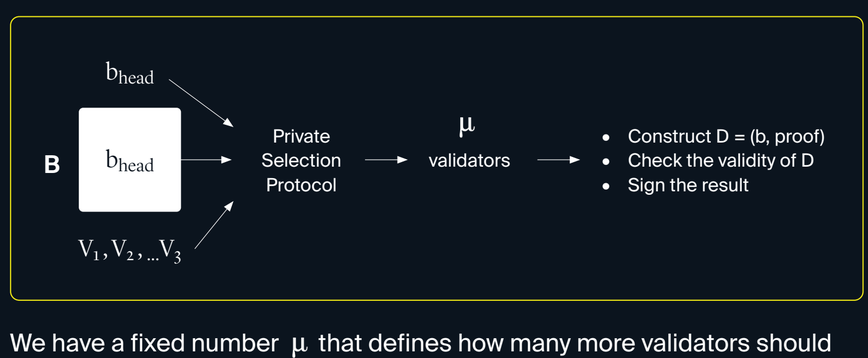


3 Availability phase

If V does not have the piece for **b** within time T, the validator announces the unavailability of its piece.

→ If V observes $f + 1$ unavailability reports, then it does not vote for **B**.
→ If there are $2f + 1$ unavailability announcements, then the validators who signed for the validity of **b** are slashed.
→ Otherwise, they run the next phase.

4 Private selection phase



We have a fixed number μ that defines how many more validators should check the validity of **b**.

→ If V satisfies below, then it is selected to check the validity of **b** till they see at least μ validity reports:

$$H(i \| r_v) \leq \frac{\mu}{n - n/m} + \Delta$$

Δ is the time passed after the availability phase, a r_v is a private random number of V.

→ If **B** equivocates, the validators checks also the following:

$$H(i \| H(B)) \leq \frac{\mu}{n - n/m}$$

5 Validity phase

→ If a validator is selected, it reconstructs **b** and **proof** and runs Verify.
- If **b** is valid, it signs for validity and publishes.
- Otherwise, it signs for invalidity and publishes.

→ If a validator sees an invalidity report, it does the same as the selected validator.

6 Finality phase

→ If there are $f + 1$ invalidity reports for **b**, then **B** is considered valid and cannot be finalized in the finality gadget.
→ All validity signers for an invalid block are slashed %100.

Advantages

→ Security of AnV does not require any honesty assumptions between validators of a committee. Therefore, we need much less validators in a committee than the committees in Ethereum 2.0.

→ Private secondary checks incentivize malicious validators to sign an invalid block because of the high risk of getting caught.

wiki.polkadot.network

research.web3.foundation

web3 foundation