VIP-193: COMMITTEE-BASED POA

Ziheng (Peter) Zhou
VeChain Foundation
peter.zhou@vechain.com

Zhijie Ren
VeChain Foundation
zhijie.ren@vechain.com

February 15, 2021

1 Overview

The Proof-of-Authority consensus algorithm [1], or PoA in short, is efficient of using network bandwidth. It divides time into rounds with a fixed length and assumes that the majority of its (authorized) nodes perform consensus in the same round. In each round, nodes select a leader (the node responsible for generating a new block) based on the round number, block height and their local views of the active nodes. Therefore, the procedure can be considered instant, which allows more time for transmitting transaction (TX) data in each consensus round.

However, PoA cannot prevent a malicious leader from causing temporary inconsistency by producing multiple blocks. To improve the security of PoA, we propose to introduce a committee (a group of selected nodes) to endorse the new block generated in each consensus round. The verifiable random function (VRF) [2] is used for committee selection in each round. With the committee mechanism, a malicious leader would have to collude with committee members to cause inconsistency. However, the property of VRF guarantees that the committee is selected randomly. Therefore, it makes it much more difficult for adversaries to launch such attacks.

It is assumed that the network is synchronous and the messages are transmitted through gossip communication, i.e., if an honest node sends a message or an honest node receives a message, all other honest nodes will receive same message within at most a delay of τ, which is determined by the network configuration and is known in advance.

2 Specifications

2.1 Notations

Symbol	Description
N	Total number of nodes
$u = 1, 2, \ldots, N$	node index
A_u	account address of u
$H(\cdot)$	Cryptographic hash function
$\texttt{SIG}_u(\cdot)$	Signature of u
r	Consensus round number
$l_r = 1, 2, ..., N$	Leader in round r

s_r	Block proposal produced by l_r
$\beta_{u,r}, \pi_{u,r}$	VRF hash and proof produced by u in round r

2.2 VRF

The verifiable random function can be considered a public-key version of keyed cryptographic hash. The hash of a given message β can only be computed by the owner of the private key SK and can be verified by anyone given the public key PK and message α. The scheme defines the following functions:

- $\beta, \pi \leftarrow \texttt{PROVE}(\texttt{SK}, \alpha)$ - function that generates VRF proof;
- $\texttt{T/F}, \beta \leftarrow \texttt{VERIFY}(\texttt{PK}, \alpha, \pi)$ - function that verify VRF proof;

2.3 Overview of Producing a New Block

Algorithm 1 describes the high-level procedure for leader l_r to produce a new block in round r.

Algorithm 1 Procedure for l_r to publish a new block.

1: Prepare and broadcast proposal s_r and signature $\texttt{SIG}_{l_r}(H(s_r))$
2: Collect endorsement messages $e_{u,r}$ from committee members
3: Prepare and broadcast new block B_r

2.4 Block Proposal

At the beginning of round r, leader l_r computes and signs a proposal of the proposed new block and broadcasts it for the committee members to endorse. A block proposal should include:

- Parent header reference
- Merkle root of the transactions to be included in the new block
- Timestamp $t_r = r * \Delta$ where Δ is the fixed length of each round

2.5 Endorsement

Nodes u that are selected as the committee members need to prepare and broadcast endorsements $e_{u,r}$. An endorsement $e_{u,r}$ should include:

- Hash of the endorsed block proposal, $H(s_r)$
- VRF proof of the committee membership, $\pi_{u,r}$
- Signature, $\texttt{SIG}_u(H(s_r))$

Algorithm 2 shows the procedure of endorsing a block proposal.

Algorithm 2 Procedure for endorsing block proposal s_r in round r.

1: Verify $\texttt{SIG}_{l_r}(H(s_r))$
2: Discard s_r and exit if having already received another block proposal from l_r in this round
3: Verify that s_r refers to the head of the current canonical chain as the parent block
4: Verify t_r in s_r against local time
5: Verify that l_r is the authorized leader in round r
6: Verify that u qualifies as a committee member in this round
7: Compute and broadcast $(H(s_r), \pi_{u,r}, \texttt{SIG}_u(H(s_r)))$

2.6 Block

After receiving and verifying endorsements, leader l_r needs to include the relevant VRF proofs and message signatures $\mathcal{E}_r = \{e_{u,r}\}$ where $e_{u,r} = (\pi_{u,r}, \texttt{SIG}_u(H(s_r)))$ in the new block B_r when preparing the block. Moreover, elements of $\mathcal{E}_r$ are organized in a fixed order in the block. For each pair $(e_{u,r}, e_{v,r})$, $e_{u,r}$ is placed in front of $e_{v,r}$ if $\beta_{u,r} < \beta_{v,r}$. Note that β can be computed using function `VERIFY` defined in Section 2.2.

Besides the VRF proofs from the committee members, leader l_r also needs to compute his own VRF proof π_{l_r} and include it in the block. The corresponding β_{l_r} is used to compute the random beacon, which will be discussed in Section 2.7.

2.7 Committee Membership

To determine committee membership, each node u needs to compute a common message $\texttt{M}_r$ in each round r, use the private key $\texttt{SK}_u$ to compute

$$\beta_{u,r}, \pi_{u,r} \leftarrow \texttt{PROVE}\left(\texttt{SK}_u, \texttt{M}_r\right) \tag{1}$$

and compare $\beta_{u,r}$ against a predefined threshold ϵ. Node u is selected as a committee member in round r if $\beta_{u,r} \leq \epsilon$.

It is desirable to add randomness in the computation of $\texttt{M}_r$ such that adversaries only know their committee memberships in a limited period of time in future. To do that, we adopt the random beacon scheme described in [3].

In particular, we divide the whole consensus process into epochs each of which contains a fix number of Q blocks. At the end of each epoch, we compute a number, or a random beacon for generating messages for blocks of the next epoch. Let b_u^{m+1} be the beacon computed by node u at the end of the m'th epoch. To determine his committee membership for any block in epoch $m+1$ with block height h, u computes the common message $\texttt{M}$ as:

$$\texttt{M} \leftarrow H(b_u^{m+1} \,\|\, h). \tag{2}$$

According to [3], beacon b_u^{m+1} can be computed as:

$$b_u^{m+1} = H\left(\beta_1^m \,\|\, \beta_2^m \,\|\, \cdots \,\|\, \beta_Q^m\right) \tag{3}$$

where β_i^m is the VRF hash that can be computed from the VRF proof generated by the block producer in the i'th block of the m'th epoch.

2.8 Security Analysis

The main goal of introducing the committee mechanism is to make it more difficult for adversaries to cause inconsistency. Perhaps the most damaging of such attacks is the double-spending attacks (DSAs) [4] where adversaries are allowed to take control of a few consecutive consensus rounds such that they can produce two parallel branches to launch a DSA. Here we derive explicitly the formula for computing the probability of launching such an attack.

Let p_ϵ be the probability of a node being selected as a committee member. This probability is directly related to threshold ϵ and is equal to every node thanks to VRF. Let us assume that there are f malicious nodes that can behave arbitrarily. For any block with a committee of size d, adversaries have to control both the leader and d committee members to produce two valid yet different blocks to launch a potential attack. We can compute the probability as:

$$F(p_\epsilon, d, f) = \sum_{i=d}^{f} \binom{f}{i} {p_\epsilon}^i \left(1 - p_\epsilon\right)^{f-i} \tag{4}$$

We only consider blocks backed by a minimum number of committee members as substantially contributing to the system's overall security. Let $\hat{d}$ be the number. We name a block a **heavy** block if it is backed by at least $\hat{d}$ committee members. Now, based on Equation 4, we can compute the probability of adversaries launching a DSA after observing k heavy blocks as:

$$F_{\text{DSA}} = \left(\frac{f}{N}\right)^k \prod_{d_i} F\left(p_\epsilon, d_i, f\right). \tag{5}$$

References

[1] Vechain development plan and whitepaper. https://cdn.vechain.com/vechainthor_development_plan_and_whitepaper_en_v1.0.pdf, 2018.

[2] D. Papadopoulos, D. Wessels, S. Huque, M. Naor, J. Vcelak, L. Reyzin, and S. Goldberg. Making NSEC5 practical for DNSSEC. *IACR ePrint*, 1999. https://eprint.iacr.org/2017/099.pdf.

[3] B. David, P. Gaži, A. Kiayias, and A. Russel. Ouroboros Praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain. In *Proc. Annu. Int'l Conf. Theory Appl. Cryptographic Techn.*, 2018.

[4] N. Satoshi. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf, 2008.