BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2025
http://excel.fit.vutbr.cz

Safeguarding PoS Consensus: A Comparative Analysis of Proposer Protection Mechanisms Against Various Attack Vectors

Tereza Burianová*

Abstract

In Ethereum, the recent move to a Proof-of-Stake based consensus introduced a new vulnerability where the future block proposers' identity is not protected from malicious actors. The aim of this paper is to introduce the thesis, which provides an analysis of various attack vectors and a summary of proposed mechanisms' suitability, and to demonstrate the implemented simplified consensus framework that explores the performance of protection mechanisms against several configurable attack vectors. The framework includes two implemented mechanisms and also allows to incorporate additional mechanism implementations. The paper presents a preview of the resulting analysis, including two attack vectors under no protection mechanism and Whisk, including advanced attackers with an understanding of Whisk's weak points.

*xburia28@vut.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

[Motivation] Validator protection is a current point of research in the Ethereum community due to the network's move from Proof-of-Work to Proof-of-Stake (PoS) in 2022. Validators are the core of the network, updating and safeguarding the chain. It is therefore important to mitigate possible attacks that target the validators with the goal to harm them or the whole network.

[Vulnerability] In PoS, block proposers are selected from validators using the randomness source RANDAO. To allow the future proposers to prepare for their role, results are known in advance. This introduced a new vulnerability where this information is known to everyone, including possible malicious actors, enabling attacks like censorship or DoS.

[Protection mechanisms] Several methods have been proposed to protect proposers' identities. Based on existing analyses and discussions [1], the best suited mechanism category for Ethereum is thought to be the Secret Single Leader Election (SSLE). The most elaborate proposal for Ethereum is Whisk, based on shuffling, zero-knowledge proofs and elliptic-curve cryptography [2]. Another possibly suitable SSLE proposal is the homomorphic sortition, based on the Threshold Fully Homomorphic Encryption [3]. There are also other PoS blockchains with working SSLE mechanisms like Algorand [4] and Polkadot [5].

[Contributions] While Whisk authors have created an analysis of their mechanism including a short comparison to other mechanisms [2], there is no tool that would allow to test the mechanisms' effectiveness and suitability for the Ethereum consensus in a common framework. In this thesis, analyses of various mechanisms have been summarised first to provide an overview and comparison. Various attack vectors have also been investigated. Next, a simplified consensus framework and several attack scenarios have been designed. While the experiments focus on Whisk and homomorphic sortition, the framework allows to easily implement and incorporate additional mechanisms.

2. Secret Single Leader Election

Secret Single Leader Election (SSLE) is a mechanism that allows to select block proposers (leaders) in a way where only the proposer knows they have been selected in the given slot.

2.1 Whisk

Whisk is the most elaborate SSLE mechanism designed for Ethereum. Elliptic curve cryptography is used to conceal proposers' identity by using trackers instead of their index searchable in the state. As shown in Fig. 1, it consists of a pipeline of selecting a list of candidates, which are then shuffled by each proposer for a given time frame. Finally, proposers are randomly selected from the shuffled list of candidates. These phases happen simultaneously, ensuring a fresh set of secret proposers is always ready [2].

2.2 Homomorphic Sortition

Homomorphic sortition builds the mechanism on the Threshold Fully Homomorphic Encryption (ThFHE) that enables computation over encrypted data. Proposers' identity is hidden using encrypted random numbers. The whole algorithm is built on a set of independent FHE circuits [3].

3. Analysis

The poster presents an analysis of Whisk under two different types of attacks: malicious DoS attack and censorship. Both of these attacks also have an advanced version where the attacker is knowledgeable about Whisk and its weak points. The effect of the protection is shown in grids that use green squares to represent successfully proposed blocks and red squares to represent slots missed due to a successful attack.

3.1 Malicious DoS

In a malicious DoS attack, the attackers attempt to DoS all participants with the goal to negatively impact Ethereum. As shown in Fig. 2, attackers are not successful in all slots due to the percentage of successful validator-IP pairings (70 %) and individual DoS protection of some validators (20 %), but the amount of missed slots is high. In Fig. 3, Whisk is used as a protection mechanism and the improvement is significant. Since the attacker only has access to previous proposers' identity, the attack is only successful if the same proposer gets selected twice in a row, which is currently very rare due to the high amount of validators. Not even the advanced attack in Fig. 4, where the attacker randomly targets a high amount of proposer candidates, was successful. This attack additionally includes a waiting phase during shuffling, as shown by the arrows.

3.2 Censorship

Censorship also utilizes DoS as the core mechanism, but a smaller group of validators is targeted to negatively influence their participation in the consensus. This attack can even be performed by nation-state actors, who have a lot of resources, as a part of regulations or sanctions. Fig. 5 depicts the missed slots due to censorship of $\frac{1}{10}$ of validators. While the effect is not as visible, it still negatively affects the network and also targets the same proposers, discouraging them from further participation and harming the decentralization of the network. As shown in Fig. 6 and Fig. 7, the rarity of DoS success, shown in subsection 3.1, combined with a smaller group of targeted proposers, helped in all cases.

4. Results

The simulation has been launched 20 times with random seeds and the same configuration as in section 3 to collect more accurate data about the effectiveness.

Fig. 8 shows that while the loss of proposed blocks has been significant during a malicious DoS attack with no protection, the effect of DoS is negligible with Whisk. 55.43 % of slots have been missed with no protection compared to 1.36 % when using Whisk and only 1.36 % of validators have been affected compared to the previous 56.63 %.

As seen in Fig. 9, the effect of a censorship attack is insignificant from a network-wide perspective, but a significant amount of slots has been missed by victims. When using Whisk, only 0.21 % of slots have been missed compared to previous 6.0 %. With no protection, 61.82 % of selected victims had their proposal disrupted, compared to 2.19 % with Whisk.

There is a trade-off in using Whisk: in one simulation run, it has been measured that the consensus related computations have taken 14 times as long using Whisk compared to using no protection mechanism (5 160 ms compared to 74 961 ms with Whisk). This has to be taken into account during further analysis.

Acknowledgements

I would like to thank my supervisor Ing. Martin Perešíni for feedback and help with my thesis and Excel@FIT contribution.

References

[1] Vitalik Buterin. Secret non-single leader election.

[2] George Kadianakis, Justin Drake, Dankrad Feist, Gottfried Herold, Dmitry Khovratovich, Mary

Maller, and Mark Simkin. Whisk: A practical shuffle-based ssle protocol for ethereum.

[3] Luciano Freitas, Andrei Tonkikh, Adda-Akram Bendoukha, Sara Tucci-Piergiovanni, Renaud Sirdey, Oana Stan, and Petr Kuznetsov. Homomorphic sortition – secret leader election for pos blockchains, 2023.

[4] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. Cryptology ePrint Archive, Paper 2017/454, 2017. https://eprint.iacr.org/2017/454.

[5] Jeff Burdges, Fatemeh Shirazi, Alistair Stewart, and Sergey Vasilyev. Sassafras.