BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2023
http://excel.fit.vutbr.cz

Selfish mining simulation framework for multiple attackers on various blockchains

Ján Jakub Kubík*

Abstract
This work presents a discrete, static, stochastic simulation framework for analyzing selfish mining with multiple attackers across diverse blockchains. The framework captures key properties, assesses multi-attacker scenarios, and identifies risks, providing an accurate, adaptable, and scalable solution. This work aims to improve the understanding of the dynamics of attacks, expose vulnerabilities, and strengthen the security measures of the blockchain with respect to selfish mining.

*xkubik32@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

As the adoption of blockchain technology grows, addressing security threats such as selfish mining becomes crucial. There is a notable absence of comprehensive simulation frameworks to analyze selfish mining with multiple attackers on various blockchains. Our proposed framework seeks to bridge this gap by evaluating selfish mining that involves multiple attackers on various blockchains, improving the understanding of attack dynamics, and exposing vulnerabilities.

Considering the state-of-the-art in this area, our intention was to create a unified framework for a comparison of multiple consensus protocols in terms of selfish mining attacks. The core challenge involves abstracting all important blockchain properties relevant to selfish mining, analyzing attack scenarios, identifying risks, and evaluating security measures. The solution should be accurate, adaptable, and scalable.

The selfish mining problem was first identified in the Nakamoto consensus protocol [1]. Research on the Nakamoto consensus has investigated scenarios with one [1], two [2], and multiple attackers [3]. Other consensus protocols have mainly been explored for single attacker cases, typically using the Markov chain approach. For example, Subchain [4], Fruitchain [4], and Strongchain [5] have been analyzed with a single attacker but not with multiple attackers.

We present a discrete, static, stochastic simulation framework to model selfish mining involving multiple attackers in diverse blockchain networks. The framework consists of four entities: simulation manager, honest miner, selfish miners, and blockchains. It utilizes a uniformly weighted random function to select round leaders who mine new blocks and make decisions according to his type. The framework incorporates key features of individual blockchain networks, such as reward schemes, fork resolution rules, and the logic for gamma (γ), which replaces the propagation and validation times for individual blocks.

2. Selfish mining problem

Figure 1 is an example of a fork in the blockchain system. Forking occurs when multiple nodes simultaneously broadcast a block for validation on the blockchain network. It can be accidentally or with malicious intent.

One type of attack that can be carried out maliciously is a selfish mining attack. This involves a selfish miner building a private blockchain and selectively revealing his blocks to invalidate honest miners' work. The selfish miner only reveals this private branch to the public when it suits him. This is causing wasted effort for honest miners. By doing so, selfish miners can earn more revenue if more of his blocks are included in the blockchain. In carrying out such attacks, selfish miners use these actions:

- **Override** – an attacker publishes a private chain when the honest chain is one block shorter.

- **Adopt** – an attacker accepts an honest chain when it becomes longer than his private chain.
- **Match** – the attacker competes with the honest chain when the chains are equally long.
- **Wait** – an attacker continues to mine on the private chain when it is more than two blocks longer than the honest chain.

3. Framework for Nakamoto consensus

The simulation framework, illustrated in **Figure 2**, models selfish mining scenarios in the Nakamoto consensus, capable of handling one or multiple selfish mining attackers. The **black text** represents the base for a single attacker, while the **red text** indicates the updates necessary to manage multiple attackers. The simulation manager oversees actions starting with "M," while selfish miners' actions begin with "S," and honest miners' actions commence with "H."

The simulation starts by parsing the YAML configuration files and creating miners and blockchains (**M1, M2, M3**). In each round, a round leader is selected, who mines a new block (**M4**). When an honest miner leads the round, it checks for an ongoing fork and decides its action accordingly (**HM mine_new_block**). On the contrary, selfish miners add newly mined blocks to their private blockchains and choose actions based on their private and public chain lengths (**SM mine_new_block**). The simulation manager guarantees correctness and determines subsequent steps (**M6, ..., M11**), including notifying selfish miners about public blockchain updates (**M12**), allowing them to decide on their next action (**SM decide_next_action**), and using **MED resolve_overrides** or **MED resolve_matches** functions based on their chosen action.

In scenarios with multiple attackers, challenges such as simultaneous attacks, multiple ties, cascading releases, and simultaneous matches may occur. To address these, the framework iterates through all selfish miners (**SD1, ..., SDY**), storing information in an action object store (**M13**). Problematic scenarios are resolved through updates in the resolve_overrides function and loops, enabling multiple blockchain overrides in a single round. The resolve_matches function is also adapted to accommodate multiple attackers. To tackle simultaneous match and new mine issues, both the simulation manager and the selfish miner's functions are updated. Miner's actions now depend on his presence in the action object store and the lengths of their private and public blockchains.

4. Supported consensus protocols

The framework supports three consensus protocols: **Nakamoto** [6], **Subchain** [7], and **Strongchain** [5]. Nakamoto consensus involves mining blocks, while Subchain has two models based on it for selfishly mining weak and strong blocks. Both use the longest chain rule for fork resolution. Strongchain, built on Nakamoto's architecture, utilizes weak and strong headers and the strongest chain rule. These consensus protocols are customizable and require mining power, simulation rounds, and specific parameters such as γ and block ratios.

5. Simulation experiments

5.1 Thresholds for successful selfish mining

Various consensus protocols exhibit different selfish mining thresholds with multiple attackers. **Table 1** displays thresholds for Nakamoto consensus, which match the current research findings. Strongchain shows similar consistency in **Table 3**. For Subchain, only selfish mining on strong blocks is profitable, as seen in **Table 2**, while weak block mining remains unprofitable.

5.2 Graphs of selfish mining

We recreated Nakamoto consensus graphs with varying γ for one attacker **Figure 3** and two attackers **Figure 4** to validate blockchain behavior. Subchain graphs were generated for one attacker with different values of γ. **Figure 5** shows similar behavior to Nakamoto with minor differences for selfish strong block mining, while **Figure 6** confirms that weak block mining is unprofitable.

6. Conclusions

The framework is validated for multiple attackers in the Nakamoto consensus and a single attacker in the Strongchain consensus, but not for Subchain due to the lack of relevant results. The validation results match the previous findings of selfish mining. Subchain examines two strategies: unprofitable weak blocks and profitable, strong blocks. In particular, the successful selfish mining threshold is slightly higher in Subchain, but a more powerful miner can gain higher proportional rewards than Nakamoto.

In the future, we plan to expand the framework by incorporating additional consensus protocols, such as Fruitchain [8], and conducting thorough evaluations of their performance and security against selfish mining attacks.

References

[1] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In Nicolas Christin and Reihaneh Safavi-Naini, editors, *Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*, volume 8437 of *Lecture Notes in Computer Science*, pages 436–454. Springer, 2014.

[2] Shiquan Zhang, Kaiwen Zhang, and Bettina Kemme. A simulation-based analysis of multiplayer selfish mining. In *IEEE International Conference on Blockchain and Cryptocurrency, ICBC 2020, Toronto, ON, Canada, May 2-6, 2020*, pages 1–5. IEEE, 2020.

[3] Shiquan Zhang, Kaiwen Zhang, and Bettina Kemme. Analysing the benefit of selfish mining with multiple players. In *IEEE International Conference on Blockchain, Blockchain 2020, Rhodes, Greece, November 2-6, 2020*, pages 36–44. IEEE, 2020.

[4] Ren Zhang and Bart Preneel. Lay down the common metrics: Evaluating proof-of-work consensus protocols' security. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 175–192. IEEE, 2019.

[5] Pawel Szalachowski, Daniel Reijsbergen, Ivan Homoliak, and Siwei Sun. Strongchain: Transparent and collaborative proof-of-work consensus. In Nadia Heninger and Patrick Traynor, editors, *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*, pages 819–836. USENIX Association, 2019.

[6] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. online, 5 2009. Accessed: 2022-11-10.

[7] Peter R. Rizun. Subchains: A technique to scale bitcoin and improve the user experience. *Ledger*, 1(1):38–52, 2016.

[8] Rafael Pass and Elaine Shi. Fruitchains: A fair blockchain. In Elad Michael Schiller and Alexander A. Schwarzmann, editors, *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017*, pages 315–324. ACM, 2017.