Generalized Framework For Deviant Mining Strategies on Bitcoin-like Blockchains

Andrew Magid†, Ketan Jog†

†Department of Mathematics, Columbia University

Background

Cryptocurrencies built on blockchain, like Bitcoin and Ethereum, have drawn a lot of academic and public interest in the past decade. A blockchain is a decentralized and distributed public ledger that records digital assets by storing historical transactions in cryptographically linked structures called blocks. Adding blocks to the blockchain verifies financial transactions. This process is dubbed "mining" since the "miner" is rewarded with newly minted cryptocurrency in exchange for their verification efforts. Recent work has shown that mining behaviours that deviate from the honest protocol are more profitable. This is an attack on peer-to-peer blockchain systems since deviant miners are able to reap greater profits than their honest counterparts. First, we present the taxonomy of existing deviant mining strategies. Then we model a generalization of Agent-Blockchain interactions using a unified framework by modeling agent dynamics independently from the blockchain to study deviant behaviours and perform a profitability and cost analysis on some of these strategies.

Taxonomy

Studies that develop deviant mining methods do not have a unified set of assumptions like mining costs, distribution of block generation times, difficulty adjustment, multiple agent simulation, or the latency of a forking event. We define an Agent A as follows: $A = \{\alpha, \gamma, P, A\}$ where α is the proportion of its computational power relative to the mining pool, γ is the proportion of miners the agent communicate its block to, P is the computational cost per unit time incurred by the agent, and action algorithm A is a function that takes in the state of the blockchain as input, and returns the number of blocks the agent publishes. This function A is the differentiating factor between different deviant mining agents.

Algorithm 2 SM1

```
Variables
publishedBlocks ← longest published chain
privateChain ← private chain
lead ← len(privateChain) - len(publishedBlocks)
procedure SM1(publishedBlocks)   ▷ returns number of blocks to publish
    if lead < 0 then                          ▷ the published chain is longer
        return 0
    else if lead ≤ 1 then                                  ▷ lead of 0 or 1
        return len(privateChain)
    else                                                ▷ the lead is greater equal 2
        return len(publishedBlocks)   ▷ return equal blocks and cause a fork
```

Algorithm 1 HM

```
Variables
publishedBlocks ← size of longest published chain
privateChain ← size of private chain
procedure HM(publishedBlocks)   ▷ returns number of blocks to publish
    if len(privatechain) != 0 then
        return len(privatechain)
```

Figure 1: Selfish and Honest Mining Strategies

An honest agent is one that follows the honest mining protocol as outlined by the original bitcoin paper[3]. Selfish agents withhold their minted blocks and reveal them later to gain more profits[1]. A Lead Stubborn miner waits until the honest miners catch up with him to broadcast all of his secret blocks[4]. Intermittent selfish miners start off as selfish miners, and switch to honest miners when the difficulty decreases[5]. Smart Mining is another strategy that switches between honest mining and staying idle in consecutive epochs[2].

Assumptions

i. The block interarrival times is an exponential process

ii. The communication delay is negligible compared to the block creation time

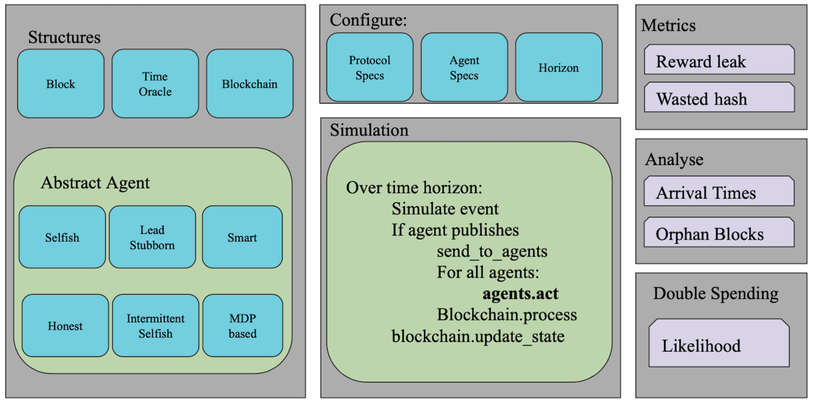
iii. Block reward and not transaction fees are considered in the profitability analysis

iv. The power distribution over the mining pool is homogeneous

v. The difficulty adjustment is accounted for maintains epoch period

vi. Concurrent block arrival during forking process

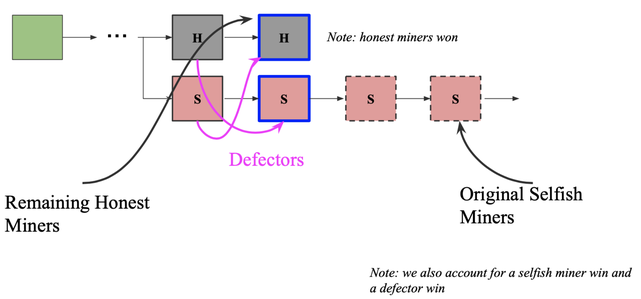
Simulator Design



The blockchain specification is defined by the cryptocurrency and its protocol. A configuration file allows parameterization of the blockchain. The user is able to define the size of each *window* which is the number of blocks per epoch. The initial difficulty and expected block arrival time (ex. 10 min for BTC) all contribute to define the difficulty adjustment algorithm.

The block arrival times are processing by a *BlocktimeOracle* which maintains a deque of cumulative times that automatically gets extended upon its previous state.
Multi-agent dynamics are achieved by continuously pinging agents for any blocks to be transmitted publicly and then retransmitting them to all agents in the system if any are received. The agents and simulation are decoupled thereby resembling the actual blockchain network.

Defector Flow

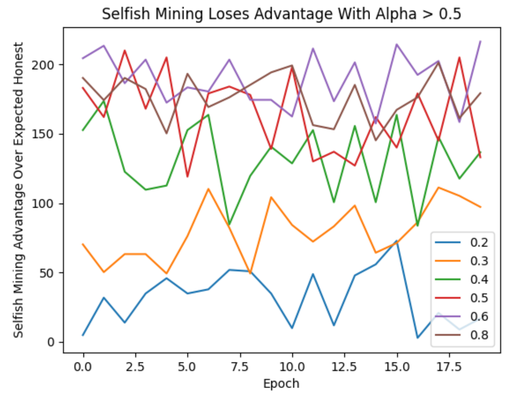


Note: we also account for a selfish miner win and a defector win

The states of all agents are used to determine who is mining at any point in time. This allows the simulation to account for mined blocks even during a fork when a selfish agent is in the lead or in a multi-agent setting. At the ith fork, the miners on the honest chain (H_i) and selfish chain (S_i) are calculated recurrently using H_{i-1} and S_{i-1}.

At any fork, there is a percentage of honest miners that choose to not mine on their own chain. Since no agent knows the identity of any other agents besides themselves, an honest miner chooses a forked block at random to mine on.
These agents are called *defectors*. We account for *defector flow* which is the complex movement of defecting agents that is created during a series of successive forks.

Cost Analysis



i. Mining cost is estimated using certain hardware assumptions

ii. $L = C_{\text{pool}} \times \alpha \times T_i$

iii. Profits are calculated based on real world market data

iv. $P = |M| \times R$

The hashrate is estimated based on difficulty and used to find wasted mining power. Deviant Mining behaviors are compared by counter-factually running them on the same protocol and environment configurations. The difference in profit is used as a metric of relative efficacy.

Remarks

We designed framework for testing deviant mining strategies on bitcoin-like blockchains. We compute the absolute and relative reward of these strategies and compare each of them in one setting.
This model is extendible to multi-agent simulation runs on any blockchain protocol. We would like to see how multiple agents and node topology affect the effectiveness of these strategies in the future.
Finally, we would like to thank Oliver Li for his help during the CSUREMM 2021 summer program.

References

[1] Ittay Eyal and Emin Gun Sirer. "Majority is not Enough: Bitcoin Mining is Vulnerable". en. In: (), p. 18.

[2] Guy Goren and Alexander Spiegelman. "Mind the Mining". en. In: Phoenix AZ USA, pp. 475–487.

[3] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System". en. In: (), p. 9.

[4] Kartik Nayak et al. "Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack". en. In: Saarbrucken, pp. 305–320. DOI: 10.1109/EuroSP.2016.32.

[5] Kevin Alarcón Negy, Peter R. Rizun, and Emin Gün Sirer. "Selfish Mining Re-Examined". en. In: Cham, 2020, pp. 61–78.