A Report on Assignment 2
of
Simulating a double selfish mining attack using the P2P Cryptocurrency Network

Introduction of Blockchains, Cryptocurrencies, and Smart Contracts

CS 765

Created By

Santanu Sahoo (23M0777)
Arnab Bhakta (23M0835)

IIT Bombay
Mumbai

Instructor

Prof. Vinay Ribeiro

Contents

1 Introduction

The decentralized nature of peer-to-peer (P2P) cryptocurrency networks presents unique challenges and vulnerabilities that require careful analysis and mitigation strategies. As the backbone of digital currencies, these networks rely on strong cognitive mechanisms to ensure the authenticity and security of transactions. However, the emergence of adversaries has introduced new threats, prompting researchers and developers to explore different attack scenarios and their potential consequences.

In this report, we embark on a journey to investigate one such attack: **the selfish double mining attack.** First explained in **Eyal and Sirer's seminal paper, "Majority is not enough"** This sophisticated attack is typical of the adversarial scenario in decentralized networks The task at hand tasks us to execute this nefarious strategy simulator using the discrete event simulator developed in the previous lesson The understanding is in-depth.

The idea of selfishness is the basis of this effort. Originally introduced to exploit weaknesses in blockchain consensus protocols, in selfish mining miners hide legal blocks from the network to gain strategic advantage.

Another feature of a **selfish double-mining attack** lies in the coordination of two independent and self-interested miners, each acting alone and unaware of the other's actions Such ongoing communication this poses a formidable challenge to network security, as traditional mitigation techniques have proven inadequate against such stealth coordinated attacks Can be carried out.

Our simulation aims to replicate the complex dynamics of this attack in a P2P cryptocurrency network. By carefully manipulating the attack scenario and testing various parameters we seek to unlock insights into its effectiveness, implications for network security and blockchain integrity ramifications Through rigorous research and testing not if we understand technology it is a selfish double attack under only Possible-countermeasures. We are also trying to clarify things to be done.

2 Background

Cryptocurrencies function based on the principles of decentralization, utilizing blockchain technology to maintain a transparent and secure record of transactions. Within a blockchain network, miners play a crucial role in validating transactions and adding them to the blockchain through a process known as mining. However, the decentralized nature of these networks also exposes them to various attacks, including selfish mining.

Selfish mining takes advantage of vulnerabilities in the consensus mechanism of a blockchain network, with the aim of gaining an unfair advantage over honest participants. Instead of immediately broadcasting mined blocks to the network, selfish miners withhold them to create a separate chain alongside the public blockchain. By strategically releasing these withheld blocks, selfish miners can manipulate the structure of the blockchain, potentially leading to double-spending or compromising the security of the network.

The concept of **double selfish mining, proposed by Eyal and Sirer,** involves multiple selfish miners operating independently within the network. Each selfish miner acts in isolation, unaware of the presence or actions of others. This scenario complicates the detection and mitigation of selfish mining attacks, as traditional defense mechanisms may prove ineffective against coordinated malicious behavior.

In our simulation, we aim to replicate the dynamics of a double selfish mining attack within a peer-to-peer cryptocurrency network. By implementing the attack scenario and experimenting with different parameters, we seek to understand its impact on network security, block propagation, and the integrity of the blockchain. Through this study, we hope to gain insights into the effectiveness of selfish mining attacks and explore potential strategies for mitigating their impact on decentralized networks.

3 Proposing Double Selfish Mining Attack

The **Double Selfish Mining Attack** as conceptualized by Eyal and Sirer in their paper "Majority is not Enough" introduces a nefarious strategy aimed at undermining the integrity of decentralized cryptocurrency networks. Unlike traditional selfish mining, which involves a single miner withholding blocks to construct a secret chain, this attack scenario involves **two independent selfish miners** operating in isolation. Each selfish miner behaves as if they are the sole adversary in the system, unaware of the presence or actions of the other attacker.

In the context of this attack, the behavior of honest and selfish miners diverges significantly:

- **Honest Miner:** Honest nodes adhere to the protocol by mining on the longest chain visible to them. In the event of tie breaks, they follow Bitcoin rules, prioritizing the first longest chain they encounter.
- **Selfish Miner:** Selfish miners, on the other hand, adopt a deceptive approach. Each selfish miner considers the longest visible chain (LVC) to them, excluding their current private blocks, as the "honest chain." They attempt to selfishly mine blocks following the strategy outlined by Eyal and Sirer. Notably, the LVC may contain blocks released by the other selfish miner, but each selfish miner operates under the assumption that all other miners are honest.

The steps followed by a selfish miner in executing the double selfish mining attack are as follows:

1. **Secret Chain Generation:** The attack commences with the selfish miner attempting to generate a secret chain, starting from the genesis block. The rules for releasing secret blocks or initiating a new attack on another block are akin to those described in the original paper.
2. **Block Broadcasting and Attack Initiation:**
   - If the selfish miner's lead over the LVC is one block, and this lead diminishes to zero, the miner immediately broadcasts the secretly mined block. Subsequently, the miner continues to mine on top of their own block. If the LVC extends further in length, the selfish miner initiates a new attack from the last block of the LVC. Alternatively, if the selfish miner mines a block, they release it immediately and commence a new attack from that block.
   - When the selfish miner's lead over the LVC is two blocks and an additional block is added to the LVC, the selfish miner promptly broadcasts all secretly mined blocks and begins a new attack from the last block of the longest chain visible to them.
   - If the selfish miner's lead over the LVC exceeds two blocks, and the LVC increases in length by one block, the miner publicly releases one more block, creating a sub chain that competes with the new block at the end of the LVC. The selfish miner continues mining on top of their secret chain.
3. **Adaptation to LVC Length:** If, at any point, the length of the LVC surpasses the length of the selfish miner's private chain, the miner initiates a new attack from the last block of the longest chain visible to them.
4. **Block Propagation Strategy:** The selfish miner selectively withholds block propagation, prioritizing the dissemination of their own blocks over those generated by other nodes. This tactic aims to expedite the propagation of selfishly mined blocks within the network.

By meticulously executing these steps, the selfish miners aim to manipulate the blockchain's structure, potentially leading to double-spending and undermining the network's security. This intricate and coordinated attack underscores the challenges posed by adversarial behavior in decentralized networks and necessitates robust defenses to safeguard against such threats.

Additionally, to simulate realistic network conditions and message propagation delays, we consider latency's L_{ij} between pairs of peers i and j connected by a link. The latency is determined by the formula:

$$L_{ij} = \rho_{ij} + \frac{|m|}{c_{ij}} + d_{ij}$$

Where:

- ρ_{ij} is a positive minimum value corresponding to speed of light propagation delay, chosen from a uniform distribution between 10ms and 500ms at the start of the simulation.
- $|m|$ denotes the length of the message in bits.
- c_{ij} is the link speed between i and j in bits per second, set to 100 Mbps if both i and j are fast, and 5 Mbps if either of the nodes is slow. Note that 50% of the honest nodes in the network are slow while adversaries will always be fast.
- d_{ij} is the queuing delay at node i to forward the message to node j, randomly chosen from an exponential distribution with a mean of $96k$ bits/c_{ij}.

4 Metrics Definition

In this section, we define and explain the metrics used to evaluate the behavior and impact of adversaries in the cryptocurrency network.

4.1 Miner Proportion Utility for Adversary Node ($MPU_{\textbf{node adv}}$)

The metric represents the number of blocks mined by the adversary included in the final public main chain divided by sum total of all blocks mined by this attacker. Or in other words, it tells us how successful an attacker is at getting his blocks into the final blockchain against how many he has mined.

Mathematically, it can be expressed as:

$$MPU_{\text{node adv}} = \frac{\text{Number of blocks mined by an adversary in the final public main chain}}{\text{Total number of blocks mined by this adversary overall}}$$

4.2 Miner Proportion Utility for All Nodes ($MPU_{\textbf{node overall}}$)

The first metric is a ratio of the number of blocks that make up the final public main chain to the total number of blocks generated across all nodes in the network. In essence, this means that it measures the importance of the last public main chain relative to all blockchain and it takes into account inputs from every node in this system

Mathematically, it can be expressed as:

$$MPU_{\text{node overall}} = \frac{\text{Number of blocks in the final public main chain}}{\text{Total number of blocks generated across all the nodes}}$$

These definitions are essential for understanding and analyzing adversaries' behavior in the cryptocurrency network, which allows researchers and practitioners to determine the adequacy of defensive strategies and countermeasures.

5 Visualization and Analysis

There are n peers, each with a unique ID, where n is set at the time of initiation of the network. Some of these nodes (say z0 percent, where z_0 is a command line simulation parameter) are labeled "slow" and the others "fast".As we have only selfish miner, we categorise the hashing Powers of two selfish miner as ζ_1 and ζ_2 which will be given as command line input.

5.1 States

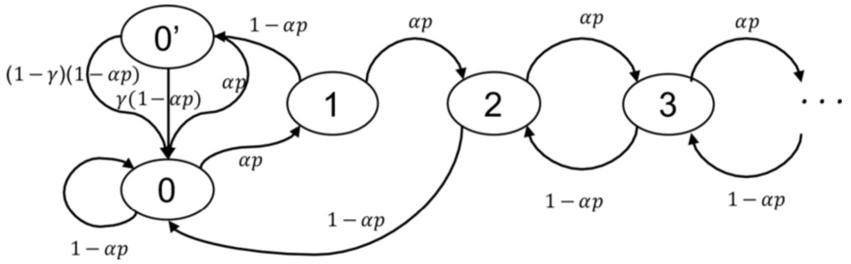

Figure 1: State Diagram

Where $\alpha * p$ is the hashing power of the selfish miner and $1 - \alpha * p$ is the hashing power of honest miner(p is smoothing factor) , γ is the fraction of honest mining power which receive on attacker chain.

5.2 Simulation parameter

Simulation Parameters are listed below which will be given as input at the time of simulation.

- N: Number of Honest Miners
- ζ_1: Hashing Power of Selfish Miner 1
- ζ_2: Hashing Power of Selfish Miner 2
- T_{tx}: Mean Inter arrival Time of Transactions
- T_{k}: Mean Mining Time of Blocks

5.3 First Simulation

We begin with the simulation with following parameter below and we observed the block chain tree of two selfish miner along with some honest miner. Also we calculated the the MPU ratio for two selfish miner and comparative analysis of the block chain tree of both honest and selfish miner.

```
NUMBER_OF_PEERS = 100  # n
Z1 = 0.3  # zeta1
Z2 = 0.001  # zeta2
AVG_TXN_INTERVAL_TIME = 100  # Ttx
## below parameters are unchanged for all the experiments
SAVE_RESULTS = True
Z0 = 0.5  # network z0 is slow
NUMBER_OF_TRANSACTIONS_PER_PEER = 200  # not used
# mean of exponential time interval bw transactions (ms)
INITIAL_COINS = 1000
EVENT_QUEUE_TIMEOUT = 5
BLOCK_TXNS_MAX_THRESHOLD = 1020  # 1020
BLOCK_TXNS_TARGET_THRESHOLD = 20
BLOCK_TXNS_MIN_THRESHOLD = 10
AVG_BLOCK_MINING_TIME = 10000  # avg block interval time (ms)

# sim stop conditions
MAX_NUM_BLOCKS = NUMBER_OF_PEERS * 3

NUMBER_OF_TRANSACTIONS = MAX_NUM_BLOCKS * BLOCK_TXNS_TARGET_THRESHOLD
```

5.3.1 Results:

```
"peer": "SelfishPeer(id=S01)",
"peer_id": "S01",
"type": "SelfishPeer",
"mpu_adv": 1.0,
"mpu_overall": 0.5165562913907285,
"num_blocks_public_chain_by_peer": 155,
"num_blocks_public_chain_by_all": 156,
"num_blocks_mined_by_peer": 155,
"num_blocks_mined_by_all": 302
```

```
"peer": "SelfishPeer(id=S02)",
"peer_id": "S02",
"type": "SelfishPeer",
"mpu_adv": 0,
"mpu_overall": 0.47635135135135137,
"num_blocks_public_chain_by_peer": 0,
"num_blocks_public_chain_by_all": 141,
"num_blocks_mined_by_peer": 0,
"num_blocks_mined_by_all": 296
```

5.3.2 Blockchain Tree

In this section, we visualize the blockchain tree involving a scenario with two selfish miners alongside one honest miner. To illustrate this, we present the prospective outcomes.

For Complete visualization of the blockchain tree for the simulation 01: Click Here.

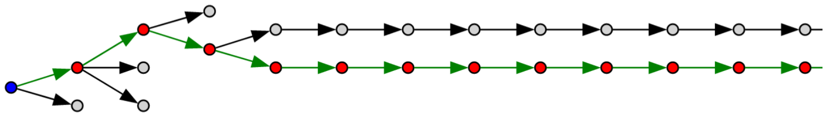

Figure 2: Blockchain Tree for Selfish miner 1

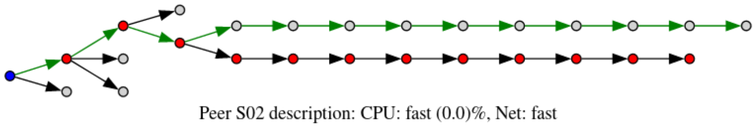

Figure 3: Blockchain Tree for Selfish miner 2

5.4 Second Simulation

As second simulation we begin the simulation by increasing the fraction of hashing power of first selfish miner from 0.3 to 0.4 but for the second selfish miner we remain its hashing power same as previous.

```
NUMBER_OF_PEERS = 100  # n
Z1 = 0.4  # zeta1
Z2 = 0.001  # zeta2
AVG_TXN_INTERVAL_TIME = 100  # Ttx
## below parameters are unchanged for all the experiments
SAVE_RESULTS = True
Z0 = 0.5  # network z0 is slow
NUMBER_OF_TRANSACTIONS_PER_PEER = 200  # not used
# mean of exponential time interval bw transactions (ms)
INITIAL_COINS = 1000
EVENT_QUEUE_TIMEOUT = 5
BLOCK_TXNS_MAX_THRESHOLD = 1020  # 1020
BLOCK_TXNS_TARGET_THRESHOLD = 20
BLOCK_TXNS_MIN_THRESHOLD = 10
AVG_BLOCK_MINING_TIME = 10000  # avg block interval time (ms)
# sim stop conditions
MAX_NUM_BLOCKS = NUMBER_OF_PEERS * 3
NUMBER_OF_TRANSACTIONS = MAX_NUM_BLOCKS * BLOCK_TXNS_TARGET_THRESHOLD
```

5.4.1 Results:

```
"peer": "SelfishPeer(id=S01)",
"peer_id": "S01",
"type": "SelfishPeer",
"mpu_adv": 1.0,
"mpu_overall": 0.5827814569536424,
"num_blocks_public_chain_by_peer": 175,
"num_blocks_public_chain_by_all": 176,
"num_blocks_mined_by_peer": 175,
"num_blocks_mined_by_all": 302
```

```
"peer": "SelfishPeer(id=S02)",
"peer_id": "S02",
"type": "SelfishPeer",
"mpu_adv": 0,
"mpu_overall": 0.49609375,
"num_blocks_public_chain_by_peer": 0,
"num_blocks_public_chain_by_all": 127,
"num_blocks_mined_by_peer": 0,
"num_blocks_mined_by_all": 256
```

5.4.2 Blockchain Tree

In this section, we visualize the blockchain tree involving a scenario with two selfish miners alongside one honest miner. To illustrate this, we present the prospective outcomes.

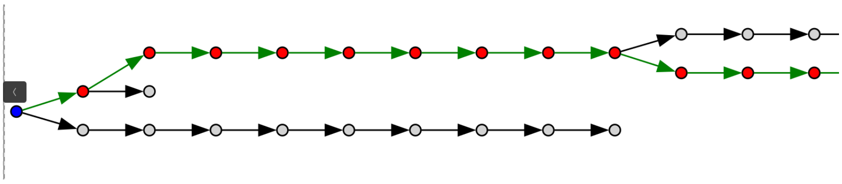

Figure 4: Blockchain Tree for Selfish miner 1

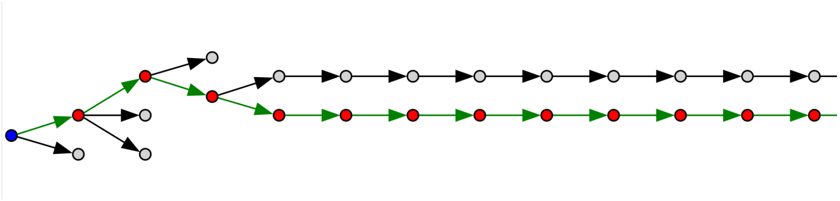

Figure 5: Blockchain Tree for Selfish miner 2

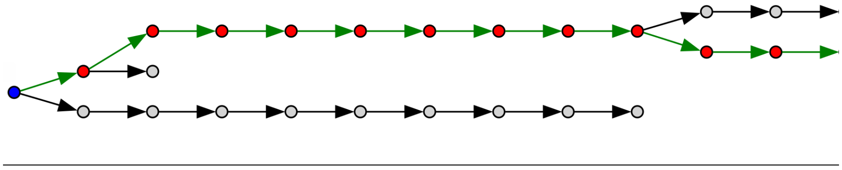

Figure 6: Blockchain Tree for Honest miner

For Complete visualization of the blockchain tree for the simulation 02: Click Here

5.5 Third Simulation

As third simulation we begin the simulation by increasing the fraction of hashing power of first selfish miner from 0.4 to 0.5 but for the second selfish miner we remain its hashing power same as previous.

```
NUMBER_OF_PEERS = 100  # n
Z1 = 0.5  # zeta1
Z2 = 0.001  # zeta2
AVG_TXN_INTERVAL_TIME = 100  # Ttx
## below parameters are unchanged for all the experiments
SAVE_RESULTS = True
Z0 = 0.5  # network z0 is slow
NUMBER_OF_TRANSACTIONS_PER_PEER = 200  # not used
# mean of exponential time interval bw transactions (ms)
INITIAL_COINS = 1000
EVENT_QUEUE_TIMEOUT = 5
BLOCK_TXNS_MAX_THRESHOLD = 1020  # 1020
BLOCK_TXNS_TARGET_THRESHOLD = 20
BLOCK_TXNS_MIN_THRESHOLD = 10
AVG_BLOCK_MINING_TIME = 10000  # avg block interval time (ms)

# sim stop conditions
MAX_NUM_BLOCKS = NUMBER_OF_PEERS * 3

NUMBER_OF_TRANSACTIONS = MAX_NUM_BLOCKS * BLOCK_TXNS_TARGET_THRESHOLD
```

5.5.1 Results:

```
"peer": "SelfishPeer(id=S01)",
"peer_id": "S01",
"type": "SelfishPeer",
"mpu_adv": 1.0,
"mpu_overall": 0.7417218543046358,
"num_blocks_public_chain_by_peer": 223,
"num_blocks_public_chain_by_all": 224,
"num_blocks_mined_by_peer": 223,
"num_blocks_mined_by_all": 302
```

```
"peer": "SelfishPeer(id=S02)",
"peer_id": "S02",
"type": "SelfishPeer",
"mpu_adv": 0.0,
"mpu_overall": 0.5176470588235295,
"num_blocks_public_chain_by_peer": 0,
"num_blocks_public_chain_by_all": 88,
"num_blocks_mined_by_peer": 1,
"num_blocks_mined_by_all": 170
```

5.5.2 Blockchain Tree

In this section, we visualize the blockchain tree involving a scenario with two selfish miners alongside one honest miner. To illustrate this, we present the prospective outcomes.

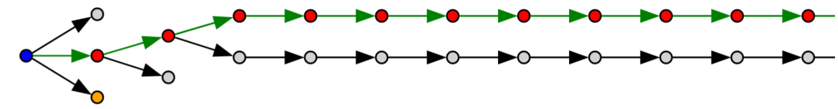

Figure 7: Blockchain Tree for Selfish miner 1

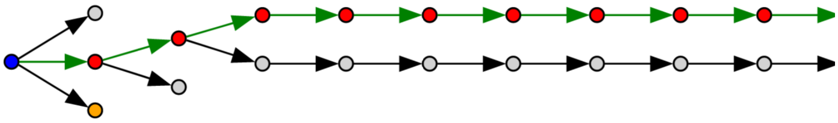

Figure 8: Blockchain Tree for Selfish miner 2

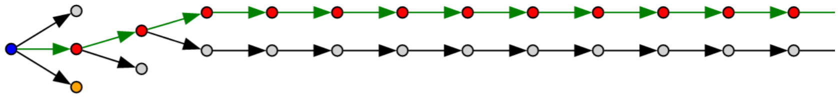

Figure 9: Blockchain Tree for Honest miner

For Complete visualization of the blockchain tree for the simulation 03: Click Here

5.6 Fourth Simulation

As fourth simulation we begin the simulation by increasing the fraction of hashing power of first selfish miner 0.3 but for the second selfish miner we change its hashing power 0.3.

```
NUMBER_OF_PEERS = 100  # n
Z1 = 0.3  # zeta1
Z2 = 0.3  # zeta2
AVG_TXN_INTERVAL_TIME = 100  # Ttx
## below parameters are unchanged for all the experiments
SAVE_RESULTS = True
Z0 = 0.5  # network z0 is slow
NUMBER_OF_TRANSACTIONS_PER_PEER = 200  # not used
# mean of exponential time interval bw transactions (ms)
INITIAL_COINS = 1000
EVENT_QUEUE_TIMEOUT = 5
BLOCK_TXNS_MAX_THRESHOLD = 1020  # 1020
BLOCK_TXNS_TARGET_THRESHOLD = 20
BLOCK_TXNS_MIN_THRESHOLD = 10
AVG_BLOCK_MINING_TIME = 10000  # avg block interval time (ms)

# sim stop conditions
MAX_NUM_BLOCKS = NUMBER_OF_PEERS * 3

NUMBER_OF_TRANSACTIONS = MAX_NUM_BLOCKS * BLOCK_TXNS_TARGET_THRESHOLD
```

5.6.1 Results:

```
"peer": "SelfishPeer(id=S01)",
"peer_id": "S01",
"type": "SelfishPeer",
"mpu_adv": 0.0,
"mpu_overall": 0.46179401993355484,
"num_blocks_public_chain_by_peer": 0,
"num_blocks_public_chain_by_all": 139,
"num_blocks_mined_by_peer": 135,
"num_blocks_mined_by_all": 301
```

```
"peer": "SelfishPeer(id=S02)",
"peer_id": "S02",
"type": "SelfishPeer",
"mpu_adv": 1.0,
"mpu_overall": 0.46357615894039733,
"num_blocks_public_chain_by_peer": 139,
"num_blocks_public_chain_by_all": 140,
"num_blocks_mined_by_peer": 139,
"num_blocks_mined_by_all": 302
```

5.6.2 Blockchain Tree

In this section, we visualize the blockchain tree involving a scenario with two selfish miners alongside one honest miner. To illustrate this, we present the prospective outcomes.

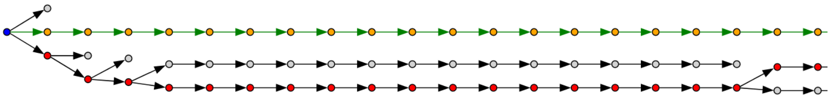

Figure 10: Blockchain Tree for Selfish miner 1

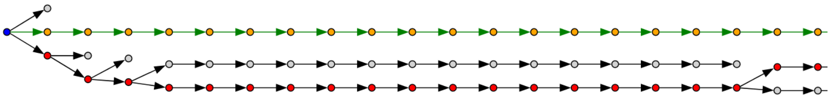

Figure 11: Blockchain Tree for Selfish miner 2

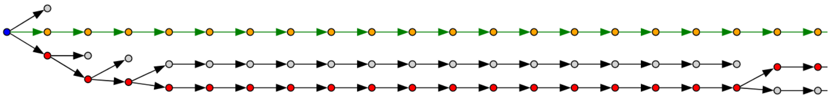

Figure 12: Blockchain Tree for Honest miner

For Complete visualization of the blockchain tree for the simulation 04: Click Here.

5.7 Fifth Simulation

As fifth simulation we begin the simulation by increasing the fraction of hashing power of first selfish miner 0.4 but for the second selfish miner we change its hashing power 0.3.

```
NUMBER_OF_PEERS = 100  # n
Z1 = 0.4  # zeta1
Z2 = 0.3  # zeta2
AVG_TXN_INTERVAL_TIME = 100  # Ttx
## below parameters are unchanged for all the experiments
SAVE_RESULTS = True
Z0 = 0.5  # network z0 is slow
NUMBER_OF_TRANSACTIONS_PER_PEER = 200  # not used
# mean of exponential time interval bw transactions (ms)
INITIAL_COINS = 1000
EVENT_QUEUE_TIMEOUT = 5
BLOCK_TXNS_MAX_THRESHOLD = 1020  # 1020
BLOCK_TXNS_TARGET_THRESHOLD = 20
BLOCK_TXNS_MIN_THRESHOLD = 10
AVG_BLOCK_MINING_TIME = 10000  # avg block interval time (ms)

# sim stop conditions
MAX_NUM_BLOCKS = NUMBER_OF_PEERS * 3

NUMBER_OF_TRANSACTIONS = MAX_NUM_BLOCKS * BLOCK_TXNS_TARGET_THRESHOLD
```

5.7.1 Results:

```
"peer": "SelfishPeer(id=S01)",
"peer_id": "S01",
"type": "SelfishPeer",
"mpu_adv": 1.0,
"mpu_overall": 0.6146179401993356,
"num_blocks_public_chain_by_peer": 184,
"num_blocks_public_chain_by_all": 185,
"num_blocks_mined_by_peer": 184,
"num_blocks_mined_by_all": 301
```

```
"peer": "SelfishPeer(id=S02)",
"peer_id": "S02",
"type": "SelfishPeer",
"mpu_adv": 0.0,
"mpu_overall": 0.6125827814569537,
"num_blocks_public_chain_by_peer": 0,
"num_blocks_public_chain_by_all": 185,
"num_blocks_mined_by_peer": 117,
"num_blocks_mined_by_all": 302
```

5.7.2 Blockchain Tree

In this section, we visualize the blockchain tree involving a scenario with two selfish miners alongside one honest miner. To illustrate this, we present the prospective outcomes.

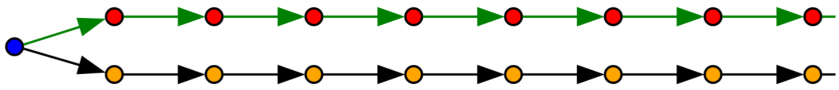

Figure 13: Blockchain Tree for Selfish miner 1

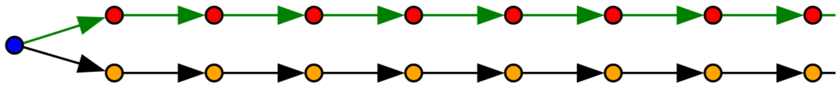

Figure 14: Blockchain Tree for Selfish miner 2

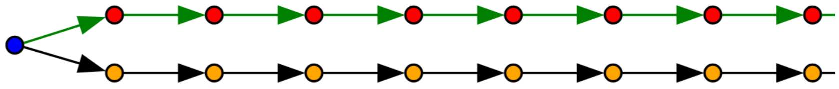

Figure 15: Blockchain Tree for Honest miner

For Complete visualization of the block tree for the simulation 05: Click Here.

5.8 Sixth Simulation

As Sixth simulation we begin the simulation by increasing the fraction of hashing power of first selfish miner 0.5 but for the second selfish miner we change its hashing power 0.3 .

```
SAVE_RESULTS = True

NUMBER_OF_PEERS = 100  # n
Z1 = 0.5  # zeta1
Z2 = 0.3  # zeta2
AVG_TXN_INTERVAL_TIME = 10  # Ttx

Z0 = 0.5  # network z0 is slow

NUMBER_OF_TRANSACTIONS_PER_PEER = 10
NUMBER_OF_TRANSACTIONS = NUMBER_OF_TRANSACTIONS_PER_PEER * NUMBER_OF_PEERS

# mean of exponential time interval bw transactions (ms)
INITIAL_COINS = 1000
EVENT_QUEUE_TIMEOUT = 5
BLOCK_TXNS_MAX_THRESHHOLD = 20  # 1020
BLOCK_TXNS_MIN_THRESHHOLD = 5
AVG_BLOCK_MINING_TIME = 100  # avg block interval time (ms)

# sim stop conditions
MAX_NUM_BLOCKS = (
    NUMBER_OF_TRANSACTIONS - NUMBER_OF_TRANSACTIONS_PER_PEER
) / BLOCK_TXNS_MAX_THRESHHOLD
```

5.8.1 Results:

```
"peer": "SelfishPeer(id=S01)",
"peer_id": "S01",
"type": "SelfishPeer",
"mpu_adv": 1.0,
"mpu_overall": 0.6046511627906976,
"num_blocks_public_chain_by_peer": 181,
"num_blocks_public_chain_by_all": 182,
"num_blocks_mined_by_peer": 181,
"num_blocks_mined_by_all": 301
```

```
"peer": "SelfishPeer(id=S02)",
"peer_id": "S02",
"type": "SelfishPeer",
"mpu_adv": 0.0,
"mpu_overall": 0.6026490066225165,
"num_blocks_public_chain_by_peer": 0,
"num_blocks_public_chain_by_all": 182,
"num_blocks_mined_by_peer": 120,
"num_blocks_mined_by_all": 302
```

5.8.2 Blockchain Tree

In this section, we visualize the blockchain tree involving a scenario with two selfish miners alongside one honest miner. To illustrate this, we present the prospective outcomes.

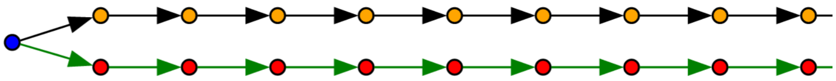

Figure 16: Blockchain Tree for Selfish miner 1

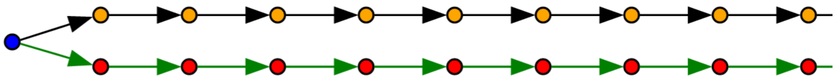

Figure 17: Blockchain Tree for Selfish miner 2

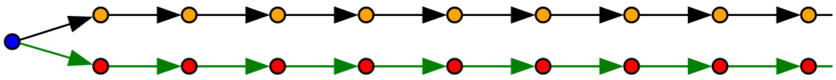

Figure 18: Blockchain Tree for Honest miner

For Complete visualization of the blockchain tree for the simulation 06: Click Here

6 Analysis of all the simulation Results

Double selfish mining attack						
First miner hashing power (ζ_1)	**0.3**	**0.4**	**0.5**	**0.3**	**0.4**	**0.5**
Second miner hashing power (ζ_2)	**0.001**	**0.001**	**0.001**	**0.3**	**0.3**	**0.3**
MPU Adversary 1	1	1	1	0	1	1
MPU Adversary 2	0	0	0	1	0	0
MPU Overall	0.49	0.53	0.62	0.46	0.61	0.60

Table 1: Double selfish mining attack

6.1 Graph For the Contribution in longest chain Vs (Zeta) ζ

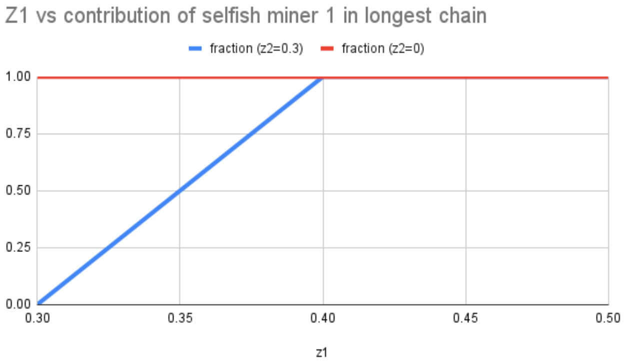

Figure 19: Plot of (Zeta) ζ_1

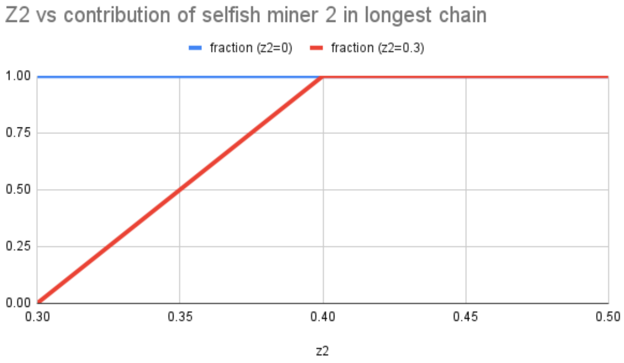

Figure 20: Plot of (Zeta) ζ_2

7 References

1. Title of Lecture 10. [Online] Available: https://people.orie.cornell.edu/mru8/orie3120/lec/lec10.pdf [Accessed on February 16, 2024].

2. Title of Chapter 3. [Online] Available: http://cs.baylor.edu/~maurer/aida/desauto/chapter3.pdf [Accessed on February 16, 2024].

3. Introduction to Discrete Event Systems. [Online] Available: https://www.cs.cmu.edu/~music/cmsip/readings/intro-discrete-even [Accessed on February 16, 2024].

4. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system", Decentralized Business Review, pp. 21260, 2008.

5. M. Jakobsson and A. Juels, "Proofs of work and bread pudding protocols", Secure information networks, pp. 258-272, 1999.

6. Eyal and Sirer's seminal paper, "Majority is not enough"