Increasing Bitcoin Revenue by Leveraging Rational Miners

Sayanil Ghosh

Increasing Bitcoin Revenue by Leveraging Rational Miners

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

of
MASTER OF TECHNOLOGY
in

CRYPTOLOGY AND SECURITY

Submitted by:
SAYANIL GHOSH (CrS2213)

Advisors
Prof. Dr. Ir. Bart Preneel
Prof. Dr. Bimal Kumar Roy

Daily Supervisor
Roozbeh Sarenche

COSIC

INDIAN STATISTICAL INSTITUTE, KOLKATA

KATHOLIEKE UNIVERSITEIT LEUVEN

To my family and friends

CERTIFICATE

This is to certify that the dissertation entitled **"Increasing Bitcoin Revenue by Leveraging Rational Miners"** submitted by **Sayanil Ghosh** to Indian Statistical Institute, Kolkata, in partial fulfillment for the award of the degree of **Master of Technology in Cryptology and Security** is a bonafide record of work carried out by him under our supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of this institute and, in my opinion, has reached the standard needed for submission.

Bart Preneel
Professor,
COSIC
Katholieke Universiteit Leuven,
Leuven, BELGIUM.

Bimal Kumar Roy
Professor,
Cryptology and Security Research Unit,
Indian Statistical Institute,
Kolkata-700108, INDIA.

Acknowledgments

I would like to show my highest gratitude to my advisors, Prof. Dr. Ir. Bart Preneel and Prof. Dr. Bimal Kumar Roy for their guidance and continuous support and encouragement.

I would also like to thank my daily supervisor, Roozbeh Sarenche for his precious suggestions and discussions. He has taught me how to do research and motivated me with excellent insights and innovative ideas.

My deepest thanks to all the teachers of Indian Statistical Institute, for their valuable suggestions and discussions which added an important dimension to my research work.

I am very much thankful to my parents and family for their everlasting support.

Lastly, I would also like to thank all of my friends for their help and support. I thank all those, whom I have missed out from the above list.

Sayanil Ghosh

Sayanil Ghosh

Abstract

Blockchain technologies have received a lot of attention over the past years. The fundamental part of each blockchain is the underlying consensus mechanism that ensures the blockchain peers agree on the state of the blockchain ledger. One of the most popular blockchains, Bitcoin is a Proof of Work(PoW) based blockchain whose underlying fork choice rule is called the longest chain.
In longest chain-based blockchains a popular attack strategy is selfish mining. In selfish mining, an attacker does not immediately publish the block. It keeps the adversarial fork private and strategically decides on the publishing time of the blocks included in this private fork.
Selfish mining only applies when the attacker's mining share is greater than 25%. Miners can collude with each other to achieve the lower bound for revenue gain. But, in the first epoch of selfish mining the block generation ratio does not increase for the selfish miner. Therefore in the first epoch, the revenue of a selfish miner does not increase compared to the revenue when it is mining honestly. Hence, miners may not agree to collude to avoid their loss.
In this thesis, we propose the *Risk-Free Collusive Selfish Mining* strategy. In this new strategy, the attacker is incentivizing rational miners to follow its strategy. The strategy is designed in such a way that the collaborators do not lose anything, hence the rational miners are more likely to follow the attacker's strategy. We have shown that an attacker with 20% or more computational power can gain more revenue than their fair share when colluding with collaborators who have 10% mining power.
Though *Risk-Free Collusive Selfish Mining* is a viable strategy, there is one risk if the collaborators publish the shared block. Therefore, we propose *Transaction Exclusion Attack* strategy where the non-compliant miners cannot ruin the attacker's strategy. In this strategy, the attacker only publishes the header instead of a full block. If there is a header in the system a dilemma between the rational miners arises regarding the choice between two strategies : (1) Mine an empty block on top of a block for which only header is published, (2) Mine on top of the previous block, in this attack scenario. We have analyzed if the rational miners are incentivized if they choose to mine on top of a block for which only header is published. The plot of the utility function shows that mining on top of a header benefits rational miners as long as the ratio of transaction fees and total fees is less than a certain value.

Keywords: *Bitcoin, Selfish Mine, Eclipse Attack, Risk-Free Collusive Selfish Mining, Transaction Exclusion Attack.*

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

In this new era, the popularity of blockchain is increasing day by day. The idea of a system with no central authority is truly intriguing. Bitcoin is the most popular cryptocurrency and holds the highest market cap*. Satoshi Nakamoto claimed in the Bitcoin whitepaper [Nak08] that the miners are incentivized to follow honest behavior. But later it was shown that there are many attack strategies such as Selfish mining [ES13], eclipse attack [HKZG15] etc. where the attacker gains more revenues than honest miners. In the first chapter of the thesis, we will go through some of the existing attack strategies. Then we will propose a new attack strategy in the second chapter and show how it is better than the existing strategies. We should show that this attack can compromise Bitcoin security.

In the third chapter, we will explore some strategies to incentivize rational miners such that they tend to follow the attacker's strategy. Rational miners always seek strategies that generate more revenues, so, if some strategy gives more incentive then the rationals are more likely to follow that strategy. We will compare the two strategies using the utility function and show which strategy is more beneficial for some rational miners in the system.

1.2 Background and Preliminaries

1.2.1 What is Blockchain

Blockchain is a distributed immutable public ledger. It is decentralized so no central authority controls the whole system. Each block is an ordered list of transactions between clients in the network. Every block contains the previous block's hash, which cryptographically links the blocks together. Some of the most popular blockchains

*https://www.forbes.com/digital-assets/crypto-prices/?sh=3bf3bb032478

are Bitcoin, Ethereum, Solana, and Cardano. The main objective of a blockchain is to create a consensus between the nodes; for that, blockchains use different consensus mechanisms. In this thesis, we will be focusing on Bitcoin and its Proof-of-Work consensus.

1.2.2 Bitcoin

Bitcoin [Nak08] is the world's first decentralized cryptocurrency. It was first introduced by Satoshi Nakamoto in 2009. It is the most popular cryptocurrency with a market cap of around 1300 billion USD [†].

PoW and Forking Rule

Bitcoin uses Proof-of-work(PoW) to achieve consensus between nodes in the network. In this mechanism, miners have to solve a cryptographic puzzle to create the PoW. The header of a Bitcoin block contains the version number, Hash of the previous block, Merkle root of the transactions inside the block, timestamp, and nonce. To create a valid block a miner has to find the nonce value such that the hash of all the fields inside the header is less than some target value set by the difficulty adjustment mechanism of Bitcoin.

$$\text{Hash(Hash(Previous block), Merkle Root,Timestamp, Nonce)} < \text{Target}$$

The difficulty adjustment mechanism adjusts the target value after every 2016 block to ensure that on average a new block is found every 10 minutes.

In the network, there can be a scenario where two blocks are proposed simultaneously creating a fork in the chain. Bitcoin follows the longest chain mechanism to resolve this forking issue. In this protocol, the miners will follow the longest chain as a greater amount of work is done to create that chain and discard the shorter chain.

1.2.3 Attack Strategies on Bitcoin

In the whitepaper of Bitcoin, Satoshi Nakamoto claimed that honest mining is the best possible strategy to gain the highest possible revenue in Bitcoin architecture. The introduction of selfish mining [ES13] later disproved this claim.

Selfish Mining

According to the honest mining strategy, a miner must immediately publish a block after mining. However, the miner does not publish it immediately in this selfish mining attack. The selfish miner keeps a private chain and strategically publishes blocks

[†]https://www.binance.com/en/price/bitcoin

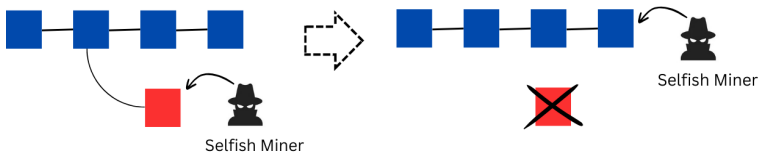

Figure 1.1: Selfish Mining Scenario 1

from it to gain more revenue.

The selfish miner decides the actions depending on the current state of network. It keeps track of the relative lengths of the private chain and the public chain and decides upon that to generate more revenue. Here we will explain the behaviour of the selfish miner in different scenarios and try to clarify its strategy.

- **Scenario 1:** The First scenario, as described in Fig 1.1, arises when the length of the public chain is greater than the private chain, i.e. the selfish miner is behind. As compared to the honest miners selfish miner has much less computation power. Therefore it is not very likely that its private chain will overtake the public chain at a point in the future. Hence, the selfish miner discards it private chain and adopts the main or the public chain and starts mining on top of the last block mined in the public chain.

- **Scenario 2:** In the second scenario, depicted in Fig 1.2 and 1.3, where the selfish miner finds a block before the honest nodes on the public branch. This event gives the selfish miner a one-block lead over the public chain. The selfish miner will not publish this block immediately, it will keep this block private and try to gain more revenue with its advantage over the public chain. Here, two cases can arise which are discussed below in detail.

  - **Case 1(Fig 1.2):** In this first case, the honest nodes on the public chain successfully mine a block which nullifies the lead of the selfish pool. In response to this event, the selfish pool immediately publishes its private block, which initiates a block race between the two chains. In this scenario, the selfish miner will mine on top of its private chain(which is now published) and honest miners will mine on top of either branch that they heard about first. The ratio of the honest miners that mine on top of the adversarial chain in the case of a same-height fork race depends on the propagation factor of the network. After this three outcomes are possible.

    1. The selfish pool mines a block before the honest miners on the other chain. It publishes these two blocks immediately and gets revenue from two successfully mined blocks.

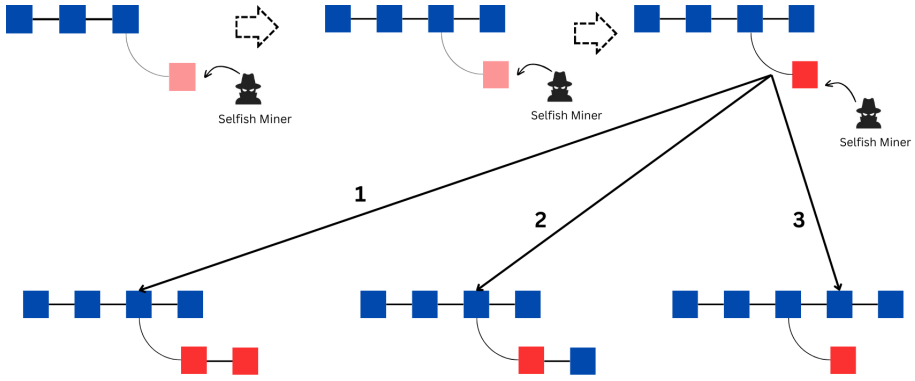

Figure 1.2: Selfish Mining Scenario 2 Case 1

2. The honest miners mine after the revealed block of the selfish pool. Then the selfish miner gets revenue from their own block.
3. The honest miners on the public chain mine a block. Then the selfish pool gets nothing.

– **Case 2 (Fig 1.3):** In the second case, the selfish miner again succeeds in mining a block on top of the previous block. Now the selfish miner has a two-block lead compared to the public chain. It will continue to mine on top of the private chain and once honest miners mine a new block on top of the public chain and reduce the attacker's lead to n¿1 block, the selfish miner publishes one of the blocks of his private chain. As the mining power of honest miners is greater than that of the selfish miner, the public chain would eventually catch up to the private chain. So, at some time the public chain will reduce the lead. As soon as the lead reduces to one block the selfish miner publishes all its private blocks and as the attacker's chain is one block longer, the honest miners will leave the public chain and start mining on top of the attacker's chain.

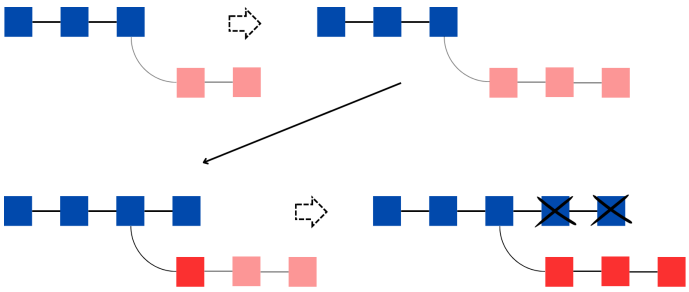

Figure 1.3: Selfish Mining Scenario 2 Case 2

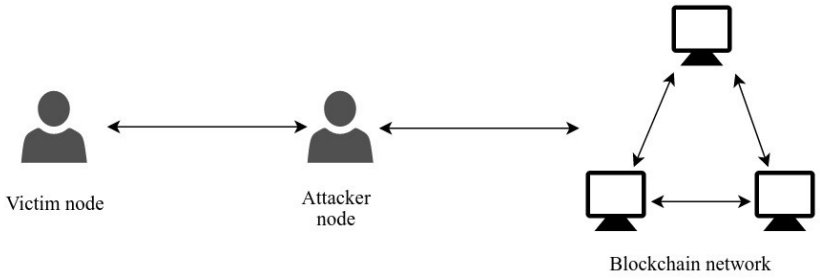

Figure 1.4: Eclipse Attack

Eclipse Attack

Eclipse Attack [HKZG15] is a network-based attack that aims to disconnect a mining node from the remaining nodes in the network. As in Fig 1.4 the attacker sits between the victim node and the blockchain network. The attacker controls all the incoming and outgoing connections of the eclipsed node.

Selfish Mining with Eclipsing

We will see a scenario where both the selfish mining and the eclipse attack are combined [NKMS16]. After a successful eclipse attack, the attacker can feed any view of the blockchain in favor of itself.

As in Fig 1.5, if there is a block race, a selfish miner can only feed the view of the blockchain with the private chain and thus compel the eclipsed node to mine on top of the last block of the private chain.

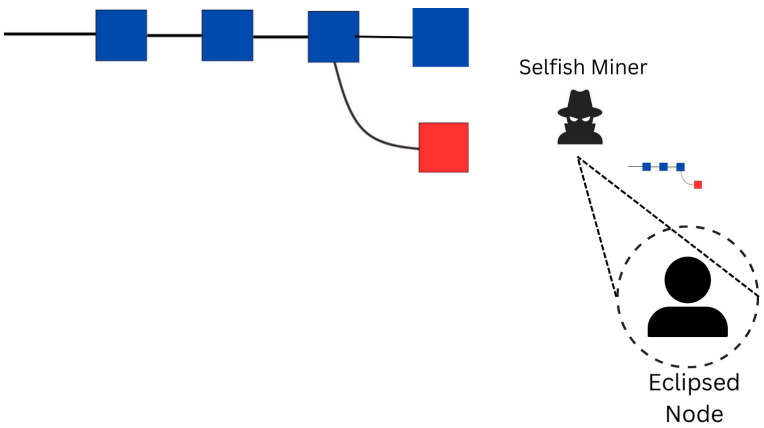

Figure 1.5: Selfish Miner with Eclipsing

1.3 Motivation

In section 1.2.3 we have seen some attack strategies like selfish mining [ES13] and Eclipse attack [HKZG15]. Despite having several attack strategies why should we look for another new strategy? The answer lies in the following part.

We have seen that by using a selfish mining strategy a miner can gain more revenues. But a mining pool needs at least 25% mining power to generate more revenues than honest miners. This indicates that most of the mining pools are not incentivized to follow selfish mining strategies.

One possible solution to reduce the minimum power share required for selfish mining is to combine this attack with an eclipse attack. By conducting an eclipse attack, the mining pools A selfish miner with a mining power share of less than 25% can gain profit with the help of an eclipse attack by following these techniques:

- Destroy the computation power of the victim by completely isolating it from the network and therefore increasing the attacker's mining share.
- As in Fig 1.5, a selfish miner can only feed the view of the blockchain with the private chain and thus compel the eclipsed node to mine on top of the last block of the private chain. With this strategy, the mining share of the selfish miner increases.

However, launching a successful eclipse attack is difficult. The attacker must control all the incoming and outgoing connections of the victim node for a successful attack. This attack becomes expensive to be a feasible attack strategy.

Another solution can be miners colluding to achieve the lower bound (25%) for generating more revenues than honest miners. One thing to note here is that selfish mining is not a riskless attack. Regardless of the several honest blocks that the selfish miner manages to orphan in the first epoch, it can not increase its block generation rate. However, after the DAM (Difficulty Adjustment Mechanism), the difficulty of mining will be reduced, indicating that the attacker's mining rate will increase. However, many miners may not be ready to take this loss of resources and launch the attack. Hence, the colluding strategy is not feasible.

We will propose a strategy where the collusion of miners with the attacker becomes riskless. In Chapter 2, we propose *Risk-Free Collusive Selfish Mining* that can reduce the minimum power share required for selfish mining without the need to perform difficult attacks such as an eclipse attack. This attack exploits the rationality of miners to incentivize them to follow the attacker's strategy.

Chapter 2

Risk-Free Collusive Selfish Mining

2.1 An Overview of The New Approach

Though the Bitcoin whitepaper [Nak08] claimed that honest behavior is always incentivized inside the network, the introduction of selfish mining proved this claim wrong. If there are strategies that result in greater revenue, rational miners are incentivized to divert from honest protocol to obtain a higher profit. We have to consider the fact that, selfish mining is not a riskless attack. During the first epoch of selfish mining (before a difficulty adjustment mechanism), selfish mining is not more profitable than honest mining. Regardless of the several honest blocks that the attacker manages to orphan in the first epoch, it can not increase its own block generation rate. However, after the DAM, the difficulty of mining will be reduced, indicating that the attacker's mining rate will increase. However, many rational miners may not be ready to take this loss of resources and launch the attack.
As we have seen selfish mining has its own risks. Moreover, the miner needs at least 25% of the total computational power to become profitable, which is relatively high. One solution we can say is that the attacker asks the rational pools to collude with each other. However, rational pools may not accept this proposal since selfish mining in the first epoch is risky. So, as an attacker how can we convince the rational mining pools to collude? The answer is by making the attack risk-free for the rationals. We will explain a strategy that makes the collusion risk-free for rational miners.
In this new strategy (Fig 2.1), we are incentivizing miners to follow the attacker's strategy under the assumption that the miners are rational. If the attacker mines one block, it will reveal the block only to the rational miners.

We will go through several scenarios in this strategy and understand the attacker's behavior:

- If the collaborator mines a block, he should publish the block to all the miners.
- If the attacker mines a block, it only reveals the block to the collaborator. Both the attacker and the collaborator continue selfish mining on top of this secret

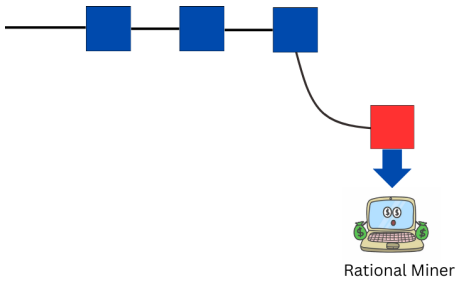

Figure 2.1: Risk-Free Collusive Selfish Mining

block to mine a private chain that is hidden from the other miners. If any one of these 2 mines a new block in the private chain, they share it with each other.

- If the length difference between the public chain and the private chain is reduced to 1, the collaborator will publish the private chain to all the miners.

Here, the rational miners are incentivized to follow the attacker's chain. Consider the scenario where rational miners are mining on top of the attacker's chain. After some time the public chain or the main chain becomes longer than the attacker's chain, hence the blocks are discarded. In this case, the discarded blocks were mined by the attacker and hence the rational miners do not have to face the loss of their resources. In other words, we can say that if the lead of private chain is only by one block, then the private chain can get orphaned. However, if the lead is greater than 1, the private chain will always win the fork race. Since the first block of the private chain is not from the collaborator, it will never lose the race. Therefore, we can say that the rational miners or the collaborators does not lose blocks if it follows the attacker's chain.

2.2 Mathematical Model and Notations

We will analyze this strategy using a state transition machine. The system is a set of miners $1, \cdots, n$ where each miner i has mining power m_i, such that $\sum_{i=1}^{n} m_i = 1$. Each miner mines on top of the chain head and finds the next block for that head after a time interval which is exponentially distributed with mean proportional to m_i^{-1}.
A group of miners can also form a pool and can behave as a single agent. The reward generated by the pool is distributed as per the relative mining powers of the miners. In our strategy, there are three types of miners: honest, attacker, and rational. The attacker is using the power of the rationals to mine on its private chain and discard more blocks created by the honest miners.

- α - Fraction of computational power possessed by the attacker

- β - Fraction of computational power possessed by collaborators
- γ - Attacker's block propagation rate inside the network

2.3 Analysis

As depicted in (Fig 2.2) we express the strategy as a state transition machine. As we can see in Fig 2.2 each state $s = 0, 1, 2, 3, ...$ represents the difference of the length between the public chain and the private chain where the attacker mines on top of the private chain with some collaborators under certain conditions and others mine on top of the public chain. For example, s = 0 denotes there is no difference between the private chain and the public chain all the miners are mining on top of the same block. The state changes when a block is created by the miners.
As we can see in Fig 2.2 if the attacker finds a block, the state changes from 0 to 1 as the height difference becomes 1. The probability of transition from state s=0 to state s=1 is equal to α since the computation power of the attacker is proportional to α. In this state s=1, the attacker will release this block only to the collaborators and they will start mining on top of the adversarial block. Hence, from now on the the probability that a new block gets mined in the private chain i.e. state transition probability from state s = i to s = i+1 for all i¿0 becomes $\alpha + \beta$. The state f_0 denotes the same-height fork race where both the public and private chains have one block. If the honest miners create a block at s=1, then the attacker publishes its block immediately creating a fork and the state changes to f_0. Since this transition depends on the honest miners finding a block the transition probability is $1 - \alpha - \beta$. In state s=f_0, according to the block propagation rate in the network, γ fraction of the honest miner will mine on top of the attacker's chain. From the state f_0 there can be three possible cases.

- The attacker and the rationals find a new block with probability $\alpha + \beta$.
- One of the honest miners from the γ fraction creates a block on attacker's chain with probability $\gamma(1 - \alpha - \beta)$.
- The honest miners find a block on the honest chain.

We will continue by calculating the state probabilities from this state transition machine.

From the State Transition machine in Fig 2.2, to calculate the probabilities over the state space we get the following equations:

$$\alpha p_0 = p_{f_0} + (1 - \alpha - \beta)p_2 \tag{2.1}$$

$$(1 - \alpha - \beta)p_1 = p_{f_0} \tag{2.2}$$

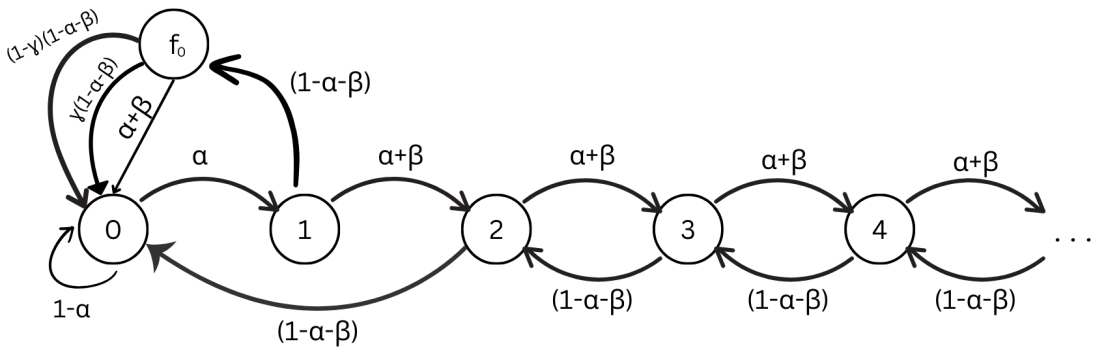

Figure 2.2: State Transition

$$\alpha p_0 = p_1 \tag{2.3}$$

$$(\alpha + \beta)p_k = (1 - \alpha - \beta)p_{k+1} \text{ for } k \geq 1 \tag{2.4}$$

$$\sum_{k=0}^{\infty} p_k + p_{f_0} = 1 \tag{2.5}$$

Solving the above equations we get the state probabilities which are the following :

$$p_0 = \frac{1 - 2\alpha - 2\beta}{(1 - 2\alpha - 2\beta) + 2\alpha(1 - \alpha - \beta)^2}$$

$$p_1 = \frac{\alpha(1 - 2\alpha - 2\beta)}{(1 - 2\alpha - 2\beta) + 2\alpha(1 - \alpha - \beta)^2}$$

$$p_k = \left(\frac{\alpha + \beta}{1 - \alpha - \beta}\right)^{k-1} p_1 \text{ for } k \geq 2$$

$$p_{f_0} = \frac{\alpha(1 - \alpha - \beta)(1 - 2\alpha - 2\beta)}{(1 - 2\alpha - 2\beta) + 2\alpha(1 - \alpha - \beta)^2}$$

2.4 Relative Revenue

The probability distribution over the state space gives a foundation for analyzing the revenue obtained by the attacker and by other miners. The revenue for finding a block goes to a miner only if the block is included in the main chain. We will calculate how much revenue is created in each state.

- **State 0 :** There are two transitions (Fig 2.3) from this state. In one of them, the attacker creates a block but doesn't publish. Hence, it will be decided later if that block will be included in the chain. In the other case, the honest miners

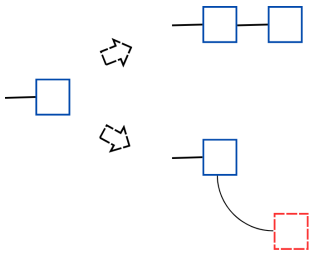

Figure 2.3: Possible transitions from State 0

mine a block which is added to the chain, and revenue of one block is gained by the honest miners.
In this case, the only revenue will be earned by the honest miners. The revenue is

$$r_{hon} = p_0(1 - \alpha)$$

- **State 1 :** In this state (Fig 2.4), if the attacker and the rational miners create a block on the private chain, then the reward for this block will be considered later. Under this scenario, the state transitions to s=2, and the attacker and the rational miners continue mining on top of the private chain. On the other hand, if the honest miners mine on the public chain then the attacker will publish the block creating a fork in the system. In this case, the state transitions to s= f_0.

- **State 2 :** Similar to the case s=0 here (Fig 2.5) if the attacker and the rational miners create a block they keep it private, the state changes to s=3 and they keep mining on top of the private chain. If the honest miners mine on the public chain then the difference in length reduces to 1, therefore the attacker and the rational miners publish the two blocks that were kept private till now. Afterward, the honest miners leave their chain and accept the other chain as it is longer. In the latter case, the attacker and the rational miners gain revenues from the two blocks. According to our strategy, the first block will always be

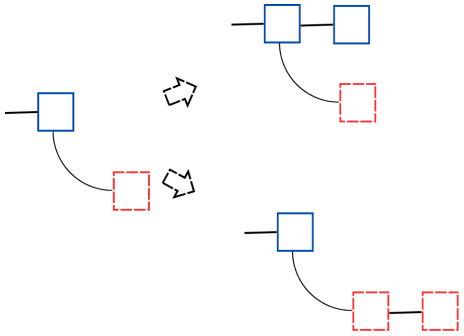

Figure 2.4: Possible transitions from State 1

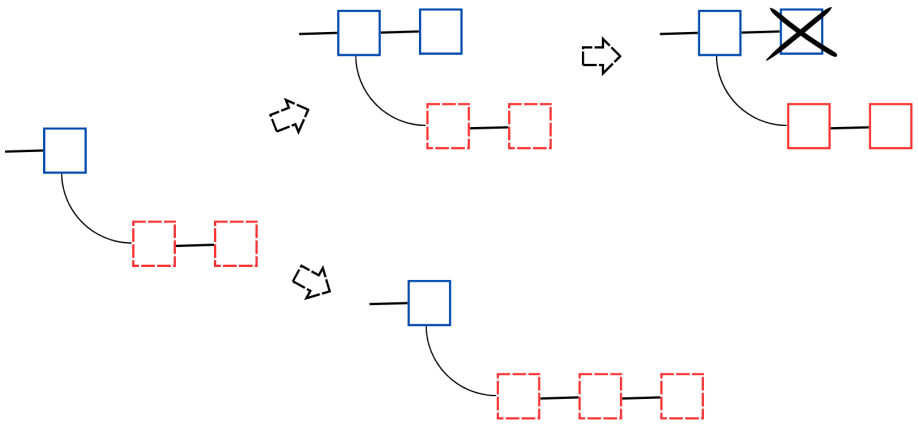

Figure 2.5: Possible transitions from State 2

mined by the attacker and the second block will be either the attacker's or the rational's with probability $\frac{\alpha}{\alpha+\beta}$ and $\frac{\beta}{\alpha+\beta}$, respectively. Therefore the revenues from this case will be as follows:

$$r_{att} = p_2(1-\alpha-\beta)(1+\frac{\alpha}{\alpha+\beta})$$

$$r_{rat} = p_2(1-\alpha-\beta)\frac{\beta}{\alpha+\beta}$$

- **State 3 and above :** These states are similar. For all the states above $s = 3$, two scenarios are possible. In the first case, the attacker and the collaborators mine a block extending the private chain. In this case, the state increases by one. Another possible case is when the honest miners mine on the public chain. When this happens, the selfish and rational pool publishes one block at that height. This block will eventually be added to the main chain since the private chain is longer. So, the revenue in this case will be:

$$r_{att} = P[i>2](1-\alpha-\beta)\frac{\alpha}{\alpha+\beta}$$

$$r_{rat} = P[i>2](1-\alpha-\beta)\frac{\beta}{\alpha+\beta}, \text{ where } P[i>2] = \sum_{i>2}^{\infty} p_i$$

$$, \text{ where } P[i>2] = \sum_{i>2}^{\infty} p_i$$

- **State f_0 :** In this state (Fig 2.6), there is a fork race in the system. A block race between the attacker and the honest miners is in place. From this state three outcomes are possible which are listed as follows::

  1. The attacker and the rational miners mine a block and receive the revenue of two blocks.
  2. The γ fraction of the honest miners mine a block on top of the attacker's chain. In this case, the attacker gets revenue for one block and the other goes to honest miners.
  3. The honest miners extend the public chain, therefore honest miners get revenue for two blocks.

  Now the revenue in this case will be:

$$r_{att} = p_{f_0}\gamma(1-\alpha-\beta) + 2p_{f_0}\alpha + p_{f_0}\beta$$

$$r_{rat} = p_{f_0}\beta$$

$$r_{hon} = 2p_{f_0}(1-\gamma)(1-\alpha-\beta) + p_{f_0}\gamma(1-\alpha-\beta)$$

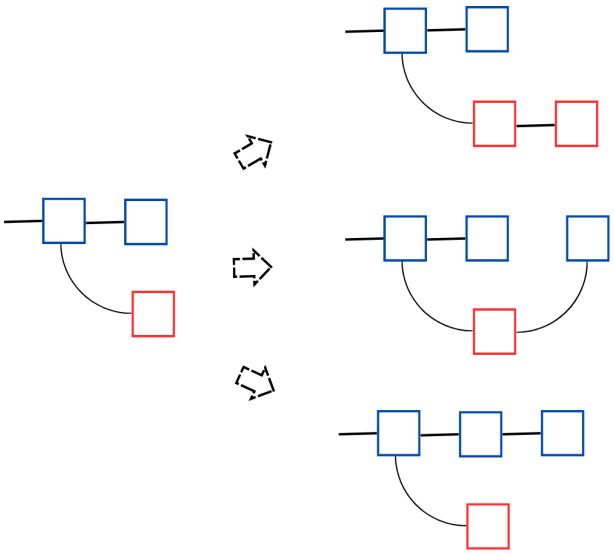

Figure 2.6: Possible transitions from State f_0

We have seen all possible cases inside our strategy. Now, we evaluate the revenues gained by the attacker, rational and honest miners respectively.

$$r_{att} = p_2(1-\alpha-\beta)(1+\frac{\alpha}{\alpha+\beta})+P[i>2](1-\alpha-\beta)\frac{\alpha}{\alpha+\beta}+p_{f_0}\gamma(1-\alpha-\beta)+2p_{f_0}\alpha+p_{f_0}\beta$$

$$r_{rat} = p_2(1-\alpha-\beta)\frac{\beta}{\alpha+\beta} + P[i>2](1-\alpha-\beta)\frac{\beta}{\alpha+\beta} + p_{f_0}\beta$$

$$r_{hon} = p_0(1-\alpha) + 2p_{f_0}(1-\gamma)(1-\alpha-\beta) + p_{f_0}\gamma(1-\alpha-\beta)$$

The relative revenue (R) of the attacker is defined as the following:

$$R = \frac{r_{att}}{r_{att} + r_{rat} + r_{hon}}$$

The final expression of the relative revenue of the attacker is the following:

$$R = \frac{\gamma\alpha(1-\alpha-\beta)^2(1-2\alpha-2\beta) + \alpha(2\alpha+\beta)(2-\alpha-\beta)(1-2\alpha-2\beta) + \alpha^2(\alpha+\beta)}{\alpha(\alpha+\beta)^2 + (1+\alpha)(1-2\alpha-2\beta)}$$

2.5 Simulation

Thus far, have described a mathematical model to represent the mining process and evaluate the revenue generated by different kinds of miners inside the system. In this section, we will simulate our strategy and see if it matches our calculation in section 2.4.
We have simulated for 10^6 steps, where in each step a new block is mined by one of the players. according to the state transition machine described in the previous section. In Fig 2.7, the relative revenue of an attacker is depicted as a function of its mining share under different strategies.

- The blue curve represents the relative revenue of the attacker when it performs normal selfish mining.
- The green dashed curve is the relative revenue of the attacker if it follows our strategy (The colluding rational miner's mining share is assumed to be 10%.).
- The black line is the revenue for honest behavior.

The red dots represent the attacker's revenue under our strategy that is obtained from simulation. As can be seen in Fig. 2.7, the relative revenue from the simulation matches our theoretical calculation.

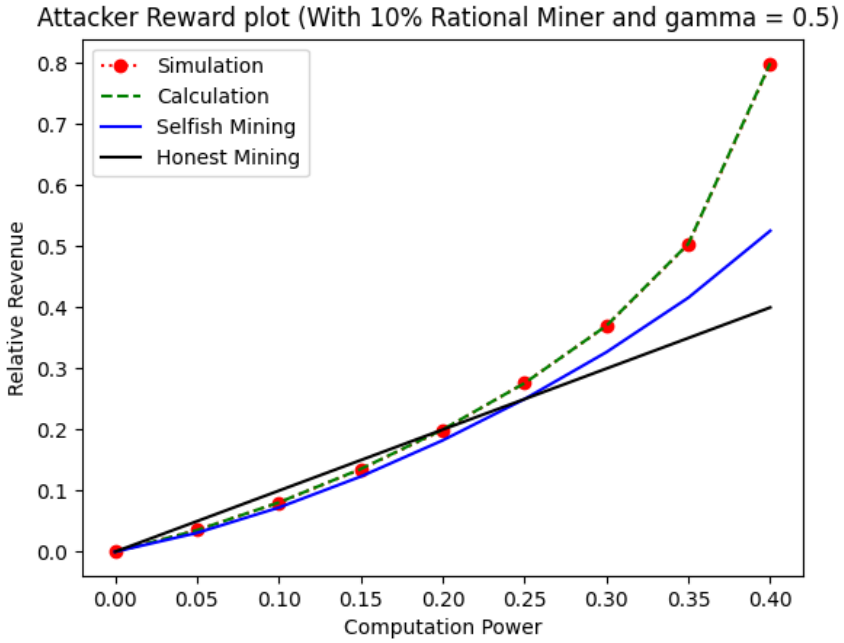

Figure 2.7: Plot of Relative Revenue with 10% collaborators and $\gamma = 0.5$

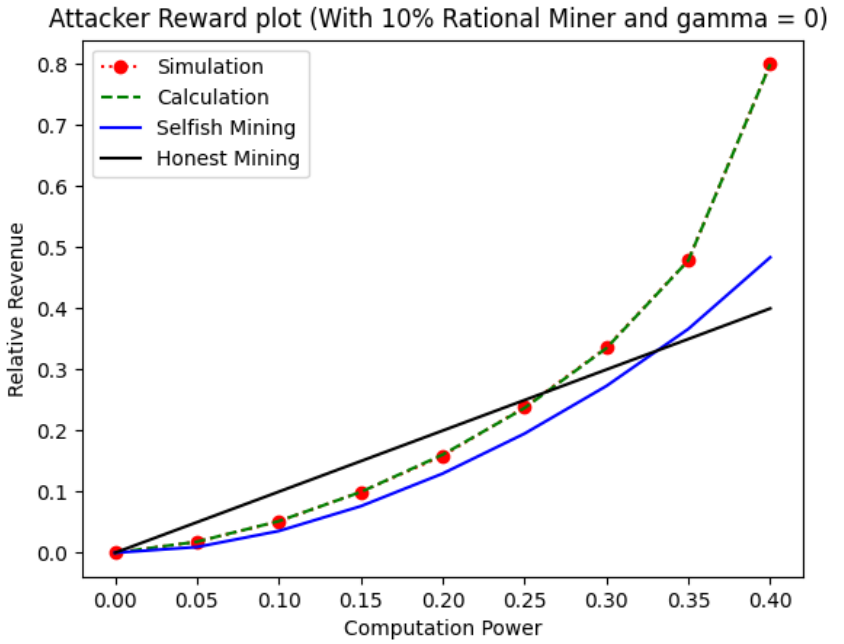

Figure 2.8: Plot of Relative Revenue with 10% collaborators and $\gamma = 0$

In Fig 2.7 we have assumed that the computational power of the rational miners is 10%. In the plot above we can see that our strategy is dominating normal selfish mining strategy. Besides, our strategy dominates honest mining when the attacker's mining power is greater than 20%. As it is shown in paper [ES13] for selfish mining the miner needs at least 25% computational power to generate more revenues than honest miners. But following our strategy that bound reduces to 20%. Fig 2.8 depicts the relative revenue of the attacker while $\gamma = 0$.

In Fig 2.9 and 2.10, we have assumed that the computational power of the rational collaborators is 15%. In the plot, we can see that our strategy is dominating the normal selfish mining strategy.

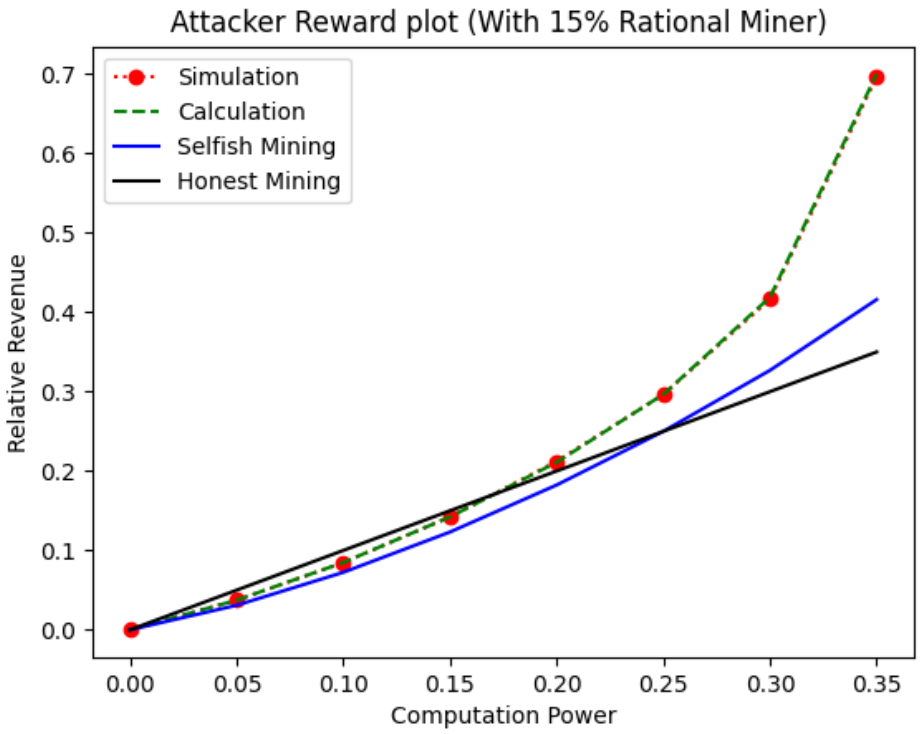

Figure 2.9: Plot of Relative Revenue with 15% collaborators and $\gamma = 0.5$

Besides, our strategy dominates honest mining when the attacker's mining power is greater than 17.5% when γ is 0.5. Fig 2.10 depicts the relative revenue of the attacker while $\gamma = 0$.

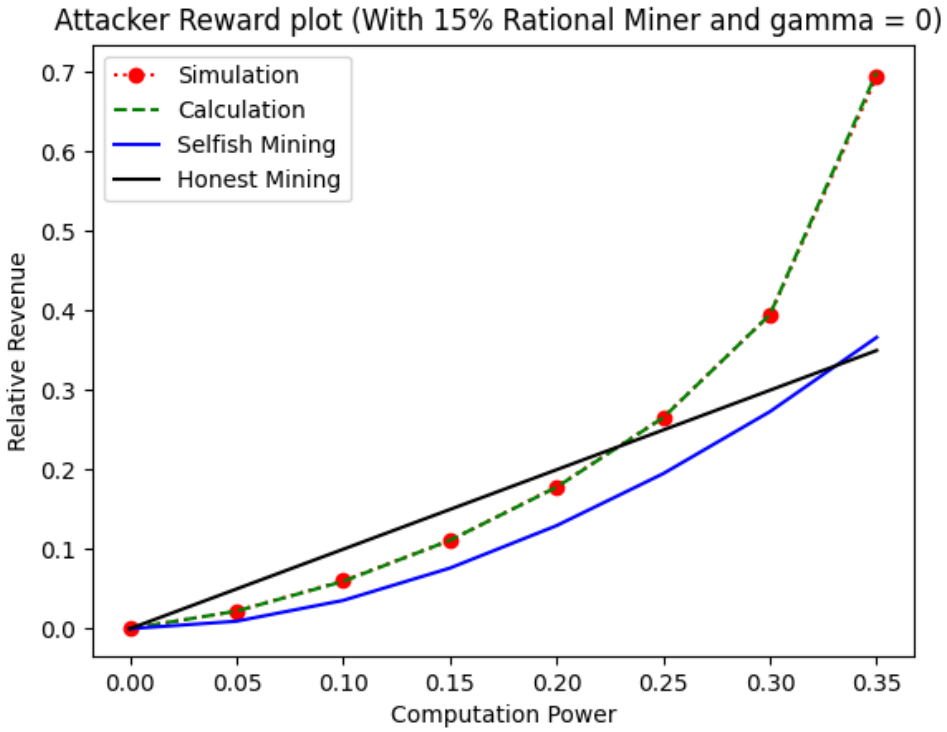

Figure 2.10: Plot of Relative Revenue with 15% collaborators and $\gamma = 0$

Chapter 3

Transaction Exclusion Attack

3.1 Overview

In Chap 2 we have seen a strategy where the attacker exploits rational miners' computational power to extend the private chain. The rationals join the pool only when the attacker mines the first block and then only they are incentivized in a way that they do not have to lose resources on the first block if it is not added to the main chain. At the end of the previous section, we have seen that the attacker only needs 20% of the total computation power and the help of the rational miners with only 10% of the total hash power to gain more revenue compared while mining honestly.
One problem with this approach is that once the collaborators hear about the block, a non-compliant miner may publish it to all nodes rather than following the adversarial strategy. In this section, we introduce an attack in which even a non-compliant miner cannot ruin the attacker's strategy.
There is a Denial of Service Attack in blockchain also known as BDoS [MJP+19]. In this paper, we can see the attacker only publishes the header instead of the whole block. If miners mine on top of the header then the attacker discards the block and creates a scenario to stop the system.

Let us see the scenario after the attacker publishes the header part instead of a full block. There can be two strategies that the other miners can follow.

- **Strategy 1:** Mine an empty (transaction-less) block on top of the block for which only the header part is released.
- **Strategy 2:** Mine on top of the previous block.

If the miners follow the first strategy it will get advantage over other miners but it has to mine an empty block losing the transaction fees. However, if the miner follows the second strategy, it loses the opportunity to gain an advantage over other miners. Hence the rational miners face a dilemma when only header is published. In the following sections, we will analyze and show which strategy is more feasible for the

rational miners. We will see that the rational miners gain more if they mine on top of the header instead of the previous block.

3.2 Mathematical Model and Notations

We will analyze this method of header publication from the perspective of a rational miner. We examine this as an infinite-horizon game since the average utility over infinite time is similar to the average utility of finite games. We will define shortly what the utility of a certain strategy is before the analysis begins. Using the Markov chain, we can compute the utility function of each strategy as a function of other player's strategy.
The mathematical model is a game played among rational miners when they are aware of the attacker's behavior. In this game players are the miners where each rational miner $\mathcal{P}_i$ has a mining power α_i. We also assume that the attacker has computation power α_A
For each miner $\mathcal{P}_i$ we define the following:

- $\prod_i(t)$ - Expected profit until time t
- $R_i(t)$ - Revenue until time t
- $C_i(t)$ - Cost until time t

We have that $\prod_i(t) = R_i(t) - C_i(t)$. Average revenue and cost per time unit for $\mathcal{P}_i$ is denoted by $\hat{R}_i \triangleq \lim_{t\to\infty} \frac{R_i(t)}{t}$ and $\hat{C}_i \triangleq \lim_{t\to\infty} \frac{C_i(t)}{t}$ respectively. Therefore, the average profit per time unit, for $\mathcal{P}_i$, will be $\hat{\prod}_i \triangleq \hat{R}_i - \hat{C}_i$.

This system progresses in rounds. Each round has a duration. We assume λ is the block mining rate. For example in Bitcoin $\lambda = \frac{1}{10.60}s^{-1}$, hence a block created every 10 minutes.

Each block generates two types of revenues (1) Block reward denoted by f_b and (2) transaction fees denoted by f_t. If a block contains transactions the total revenue of the block will be $f_b + f_t$. However, if the block is transaction-less the the block revenue is f_b.

To define the utility function, we normalize the expected profit by the miner's mining power. The utility function U_i of P_i is thus: $U_i \triangleq \frac{\hat{\prod}_i}{\alpha_i}$.

In this setup, the cost of block generation is constant since no miners stop the mining

process. Therefore we can exclude the cost and evaluate the time-averaged revenue for both cases when the P_i mines on top of the header and P_i mines on top of the previous block. From the revenue, we will calculate the utility of the two strategies as per the definition and compare them to conclude which strategy is better for the rational miners to follow.

3.3 Analysis of Utility Between Strategies

We will consider a general scenario for our analysis. Here $\mathcal{P}_i$ is a rational miner with computation power α_i. It can decide between any two of the strategies (1) mine on top of the header or (2) mine on top of the previous block. We will evaluate the utility for each case when $\mathcal{P}_i$ follows one of the strategies. In each case, the other rational miners excluding $\mathcal{P}_i$ can be divided into two groups, one following the first strategy and the other following the second strategy. We assume that R_1 number of rational miners follow the first strategy with computation power α_{R_1} and R_2 number of rational miners follow the second strategy with computation power α_{R_2}.

3.3.1 Mine on Top of The Header

Fig 3.1 denotes different states in the state transition machine.

- **State 0:** This state denotes the initial system of the blockchain when all miners are mining on top of one block.
- **State 1:** The attacker mines a block and only publishes the header of the block without revealing the whole block.
- **State 0':** Some of the rational miners mine on top of the header and extend it with an empty block.
- **State 2:** Some rational miners create a block on top of the previous block creating a fork inside the system.

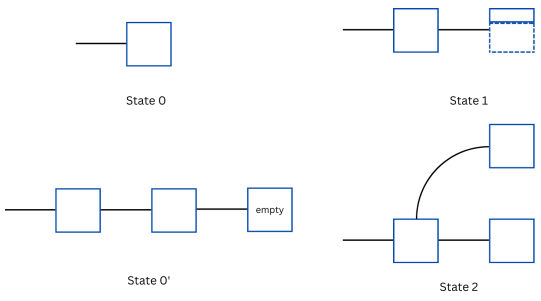

Figure 3.1: Different State of the STM for strategy 1

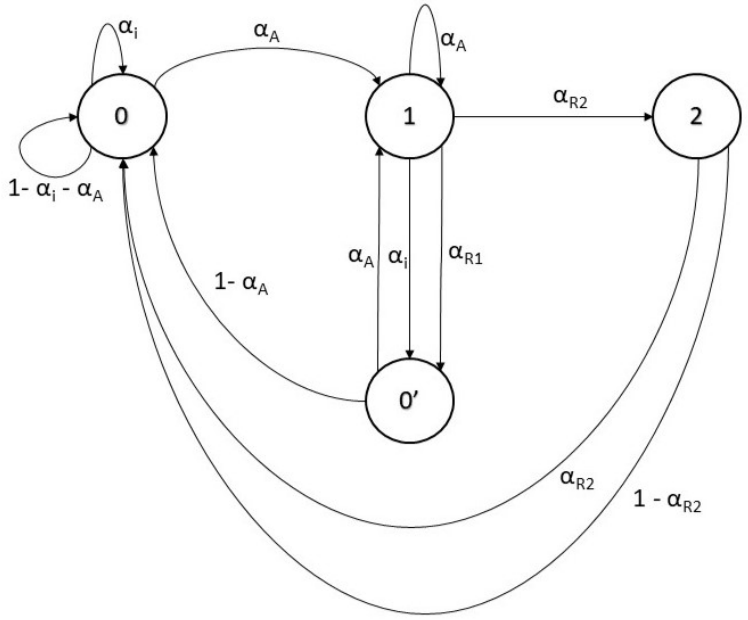

Figure 3.2: STM if mined on top of Header (Strategy 1)

In Fig 3.2 we can see the state changes from 0 to 1 with probability α_A since the computation power of the attacker is α_A. The attacker will publish only the header part of the block and thus the system will be in state 1. From state 1, the miner set R_1 will mine on top of the header and the miner set R_2 will mine on top of the previous block. In this scenario, $\mathcal{P}_i$ will be mining on top of the block whose header is released in the system. One thing to notice is that from state 2 to state 0 there is a fork between two blocks, in which one is created by the attacker (B_A) and the other is created on top of the previous block (B_P). Now, only the rational miners who created B_P will mine on top of it, and others will choose B_A. The reason for this is that B_A was created before B_P. Therefore, B_A will contain transactions with less amount of transaction fees compared to B_P, so, it is more beneficial to the other rationals if they extend the chain including B_A.

We assume the probability distribution over state space is $p_0^{S_H}, p_{0'}^{S_H}, p_1^{S_H}$ and $p_2^{S_H}$. From Fig 3.2 we get the following equations.

$$(1 - \alpha_A)p_{0'}^{S_H} + p_2^{S_H} = \alpha_A p_0^{S_H} \tag{3.1}$$

$$(\alpha_i + \alpha_{R_1})p_1^{S_H} = p_{0'}^{S_H} \tag{3.2}$$

$$\alpha_A p_0^{S_H} + \alpha_A p_{0'}^{S_H} = (1 - \alpha_A)p_1^{S_H} \tag{3.3}$$

$$\alpha_{R_2} p_1^{S_H} = p_2^{S_H} \tag{3.4}$$

$$p_0^{S_H} + p_{0'}^{S_H} + p_1^{S_H} + p_2^{S_H} = 1 \tag{3.5}$$

After solving the above equations we get the probabilities as follows:

$$p_0^{S_H} = \frac{(1-\alpha_A)(\alpha_i + \alpha_{R_1})}{(1-\alpha_A)(\alpha_i + \alpha_{R_1}) + \alpha_A(\alpha_i + \alpha_{R_1}) + \alpha_A + \alpha_A\alpha_{R_2}}$$

$$p_{0'}^{S_H} = \frac{\alpha_A(\alpha_i + \alpha_{R_1})}{(1-\alpha_A)(\alpha_i + \alpha_{R_1}) + \alpha_A(\alpha_i + \alpha_{R_1}) + \alpha_A + \alpha_A\alpha_{R_2}}$$

$$p_1^{S_H} = \frac{\alpha_A}{(1-\alpha_A)(\alpha_i + \alpha_{R_1}) + \alpha_A(\alpha_i + \alpha_{R_1}) + \alpha_A + \alpha_A\alpha_{R_2}}$$

$$p_2^{S_H} = \frac{\alpha_A\alpha_{R_2}}{(1-\alpha_A)(\alpha_i + \alpha_{R_1}) + \alpha_A(\alpha_i + \alpha_{R_1}) + \alpha_A + \alpha_A\alpha_{R_2}}$$

Revenue For Strategy 1

Here we will calculate the time-averaged revenue of the rational miner $\mathcal{P}_i$ if it follows the first strategy. For each state, we will see what will be the average revenue for $\mathcal{P}_i$.

- **State 0:** When the system is in state 0, the rational miner can mine a full block. The revenue for this event will be:

$$p_0^{S_H}(f_b + f_t)\alpha_i\lambda$$

- **State 1:** In this case the rational miner will mine on top of the header. Therefore it will create an empty block with no transactions. Hence the revenue will be:

$$p_1^{S_H} f_b \alpha_i \lambda$$

- **State 0':** Here the miner $\mathcal{P}_i$ will mine a full block and gain the usual revenue.

$$p_{0'}^{S_H}(f_b + f_t)\alpha_i\lambda$$

- **State 2:** The miner will be working on the attacker's branch to resolve the fork and if it succeeds then it will gain the revenue of a full block. Hence the revenue can be obtained as follows:

$$p_2^{S_H}(f_b + f_t)\alpha_i\lambda$$

Utility for Strategy 1:

As defined in sec 3.2 we will evaluate the utility for the miner $\mathcal{P}_i$ if it follows the first strategy. We already have all the possible revenues, hence the utility is as follows:

$$\begin{aligned} U_i^{S_H} &= \frac{1}{\alpha_i}[p_0^{S_H}(f_b+f_t)\alpha_i\lambda + p_1^{S_H} f_b\alpha_i\lambda + p_{0'}^{S_H}(f_b+f_t)\alpha_i\lambda + p_2^{S_H}(f_b+f_t)\alpha_i\lambda] \\ &= [(1-p_1^{S_H})f_t + f_b]\lambda \end{aligned}$$

3.3.2 Mine on Top of Previous Block

Fig 3.3 expresses the state transition if $\mathcal{P}_i$ does not accept the header and mines on top of the previous block. The states are the same as those depicted in Fig 3.1 except for state 3. In this state, the miner $\mathcal{P}_i$ mines on top of the previous block. If P_i manages to mine a block, a fork race will occur. In this fork race, all rational miners will mine on top of the other chain since this block will contain transactions with more fees.
In Fig 3.3 the transitions are almost similar to the previous strategy. The only part where it differs is $\mathcal{P}_i$ does not mine on top of the header so we can see an arrow from state 1 to state 3. $\mathcal{P}_i$ mines on top of the previous block and creates a fork in the system.

We assume the probability distribution over state space is $p_0^{S_{pb}}, p_{0'}^{S_{pb}}, p_1^{S_{pb}}, p_2^{S_{pb}}$ and $p_3^{S_{pb}}$. We get the following equations from Fig 3.3.

$$(1-\alpha_A)p_{0'}^{S_{pb}} + p_2^{S_{pb}} + p_3^{S_{pb}} = \alpha_A p_0^{S_{pb}} \tag{3.6}$$

$$\alpha_{R_1} p_1^{S_{pb}} = p_{0'}^{S_{pb}} \tag{3.7}$$

$$\alpha_A p_0^{S_{pb}} + \alpha_A p_{0'}^{S_{pb}} = (1-\alpha_A)p_1^{S_{pb}} \tag{3.8}$$

$$\alpha_{R_2} p_1^{S_{pb}} = p_2^{S_{pb}} \tag{3.9}$$

$$\alpha_i p_1^{S_{pb}} = p_3^{S_{pb}} \tag{3.10}$$

$$p_0^{S_{pb}} + p_{0'}^{S_{pb}} + p_1^{S_{pb}} + p_2^{S_{pb}} + p_3^{S_{pb}} = 1 \tag{3.11}$$

After solving the above equations we get the probabilities as follows:

$$p_0^{S_{pb}} = \frac{(1-\alpha_A)\alpha_{R_1} + \alpha_{R_2} + \alpha_i}{(1-\alpha_A)\alpha_{R_1} + \alpha_{R_2} + \alpha_i + \alpha_A + \alpha_A\alpha_{R_1} + \alpha_A\alpha_{R_2} + \alpha_A\alpha_i}$$

$$p_{0'}^{S_{pb}} = \frac{\alpha_A\alpha_{R_1}}{(1-\alpha_A)\alpha_{R_1} + \alpha_{R_2} + \alpha_i + \alpha_A + \alpha_A\alpha_{R_1} + \alpha_A\alpha_{R_2} + \alpha_A\alpha_i}$$

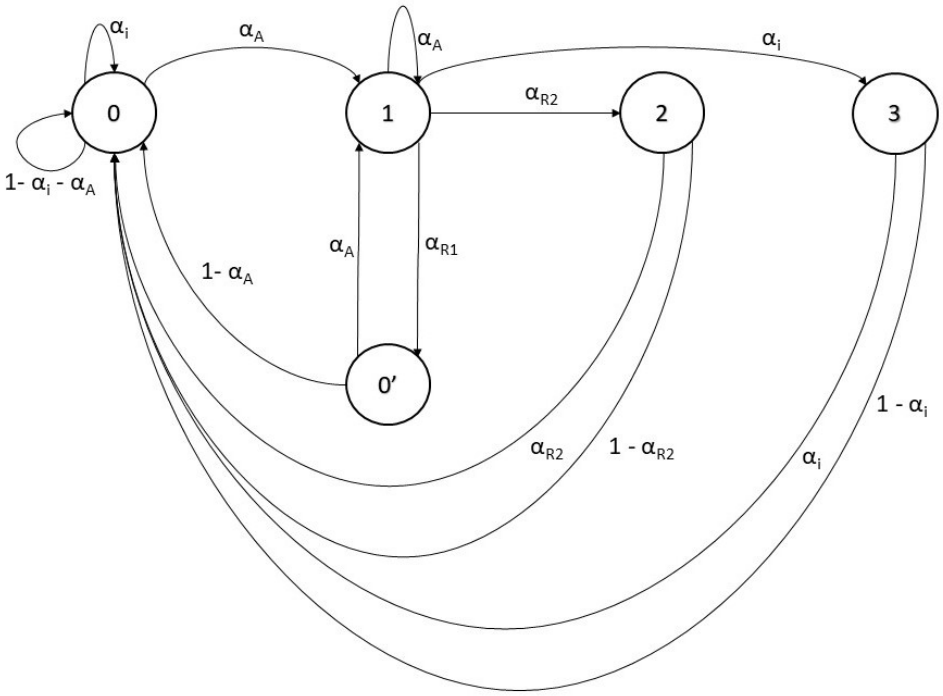

Figure 3.3: STM if mined on top of previous Block

$$p_1^{S_{pb}} = \frac{\alpha_A}{(1-\alpha_A)\alpha_{R_1} + \alpha_{R_2} + \alpha_i + \alpha_A + \alpha_A\alpha_{R_1} + \alpha_A\alpha_{R_2} + \alpha_A\alpha_i}$$

$$p_2^{S_{pb}} = \frac{\alpha_A\alpha_{R_2}}{(1-\alpha_A)\alpha_{R_1} + \alpha_{R_2} + \alpha_i + \alpha_A + \alpha_A\alpha_{R_1} + \alpha_A\alpha_{R_2} + \alpha_A\alpha_i}$$

$$p_3^{S_{pb}} = \frac{\alpha_A\alpha_i}{(1-\alpha_A)\alpha_{R_1} + \alpha_{R_2} + \alpha_i + \alpha_A + \alpha_A\alpha_{R_1} + \alpha_A\alpha_{R_2} + \alpha_A\alpha_i}$$

Revenue For Strategy 2

- **State 0:** In this state all miners are mining on the same chain. $\mathcal{P}_i$ can create a full block and generate revenue as follows:

$$p_0^{S_{pb}}(f_b + f_t)\alpha_i\lambda$$

- **State 1:** $\mathcal{P}_i$ can generate revenue from this state only if it creates a block on top of the previous chain and successfully extends it. From Fig 3.3 we can see that

it means a state change from 1 to 3 and then 0. Hence, time average revenue from state 1 is:

$$\alpha_i p_1^{S_{pb}}(f_b + f_t)\alpha_i\lambda$$

- **State 0′:** In this case attacker's chain is extended and $\mathcal{P}_i$ is working on this chain since it is longer. If $\mathcal{P}_i$ creates a block it will get the following revenue:

$$p_{0'}^{S_{pb}}(f_b + f_t)\alpha_i\lambda$$

- **State 2:** The miner will be working on the attacker's branch to resolve the fork and if it succeeds then it will gain the revenue of a full block. Hence the revenue is as follows:

$$p_2^{S_pb}(f_b + f_t)\alpha_i\lambda$$

- **State 3:** $\mathcal{P}_i$ will work on its own block and upon extending gets the revenue as follows:

$$p_3^{S_pb}(f_b + f_t)\alpha_i\lambda$$

Utility for Strategy 2:

As defined in sec 3.2 we will evaluate the utility for the miner $\mathcal{P}_i$ if it follows the second strategy. We already have revenues of P_i in strategy II, hence the utility is as follows:

$$\begin{aligned} U_i^{S_{pb}} &= \frac{1}{\alpha_i}[(p_0^{S_{pb}} + \alpha_i p_1^{S_{pb}} + p_{0'}^{S_{pb}} + p_2^{S_pb} + p_3^{S_pb})(f_b + f_t)\alpha_i\lambda] \\ &= (1 - p_1^{S_{pb}} + \alpha_i p_1^{S_{pb}})(f_b + f_t)\lambda \end{aligned}$$

3.3.3 Utility Between Strategies:

In the previous sections (3.3.1 and 3.3.2), we calculated the utility of $\mathcal{P}_i$ for following both strategies 1 and 2. If we take the difference then we get the following expression:

$$U_i^{S_H} - U_i^{S_{pb}} = [(1 - p_1^{S_H})f_t + f_b]\lambda - (1 - p_1^{S_{pb}} + \alpha_i p_1^{S_{pb}})(f_b + f_t)\lambda$$

In the expression above, we assume that $r = \frac{f_t}{f_b+f_t}$. Replacing the values of $p_1^{S_H}, p_1^{S_{pb}}$ and r from our previous calculation in section 3.3.1 and 3.3.2 we get the following expression:

$$U_i^{S_H} - U_i^{S_{pb}} = \frac{\alpha_A(1 - \alpha_i)}{r[1 + \alpha_A(\alpha_{R_2} + \alpha_i)]} - \frac{\alpha_A}{1 - \alpha_{R_2} + \alpha_A\alpha_{R_2}}$$

3.4 Plot and Conclusion:

Thus far we have seen the mathematical calculation for comparing the two strategies. In section 3.3.3 we get the expression for the difference between the two utility functions for two strategies.
Here in Fig 3.4 we depict the plot of that expression for different parameters and show that the difference is above zero for each case. In Fig 3.4 the x-axis denotes the computational power of P_i and the y-axis is denoted by $U_i^{S_H} - U_i^{S_{pb}}$. We have assumed $r = 0.2$ and the attacker's computational power is 20%. The assumptions for each curve are as follows:

- For the red curve $R_2 = 70\%$.
- For the green curve $R_2 = 60\%$.
- For the blue curve $R_2 = 50\%$.
- For the black curve $R_2 = 40\%$.

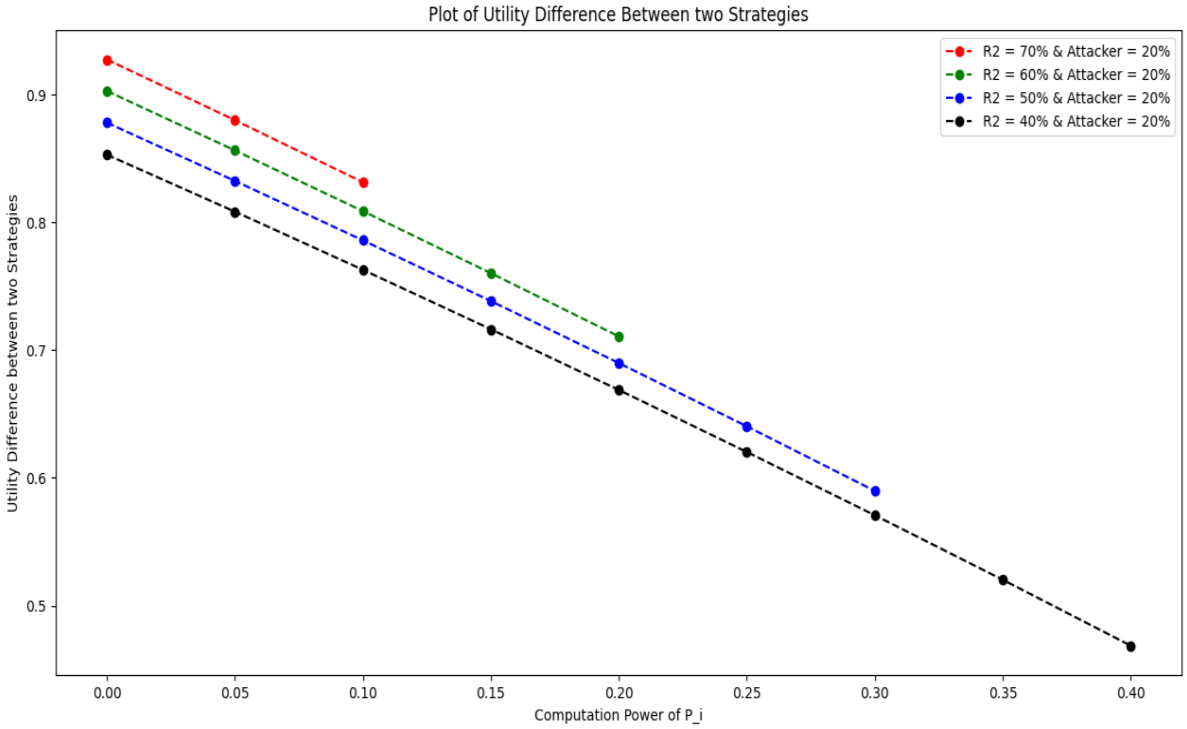

Figure 3.4: Utility Difference between two Strategies

As depicted in Fig 3.4, in each case the utility difference is positive which implies that for the rational miner P_i mining on top of the header is feasible for each of these scenarios. Here we have assumed that r=0.2 which means the transaction fees must be equal to 2.5 BTC since the current block reward is 3.125 BTC. Though the transaction fees varies but average transaction fee is almost equal to 0.000041 BTC. Since, r is in the denominator of the first team in the expression of $U_i^{S_H} - U_i^{S_{pb}}$, the expression will be still positive if $r < 0.2$ in these scenarios.
We can conclude from the Fig 3.4 that for the rational miner P_i mining on top of the block for which only header is published is more beneficial than mining on top of the previous block when the ratio of transaction fees and total fees is less than 0.2 even if most of the other miners(R_2) are mining on top of the previous block.

3.5 Attacker's Revenue:

Thus far we have seen the analysis of the attack with respect to a rational miner. Let us delve into the part of how the attacker is creating better revenues. As we know, if the block generation time increases then the number of transactions increases in the mempool, and the transaction fees increase as well.

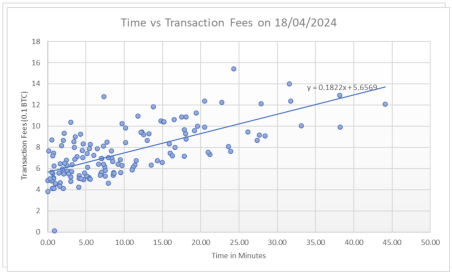

Figure 3.5: 18/04/2024

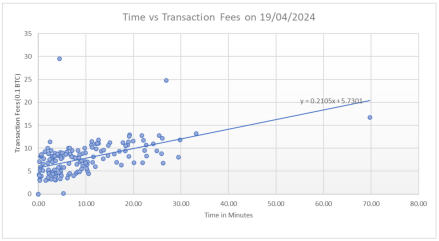

Figure 3.6: 19/04/2024

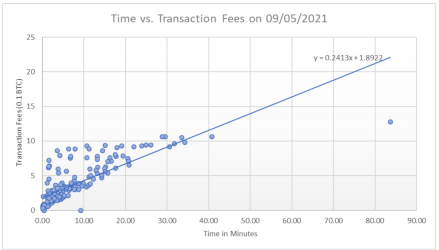

Figure 3.7: 09/05/2021

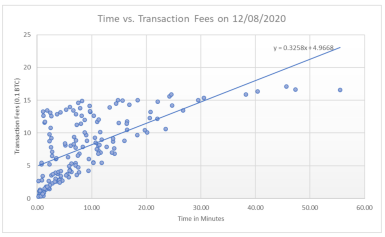

Figure 3.8: 12/08/2020

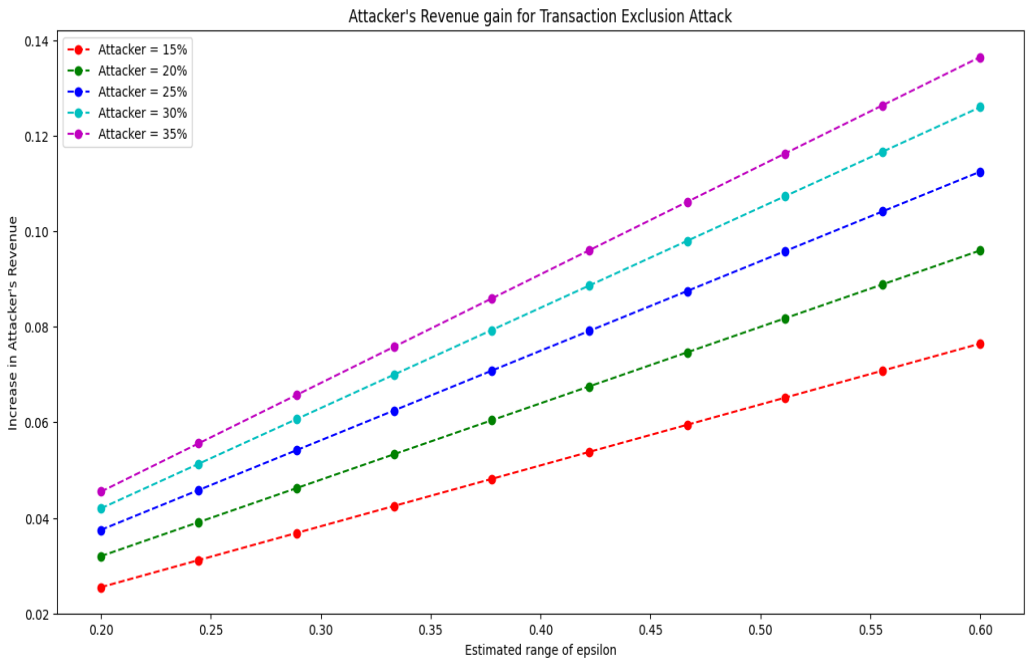

Figure 3.9: Plot of Increase in Attacker's Revenue in Transaction Exclusion Attack

So, when an empty block is produced then there can be an increase of transition fees. We assume that the fees increase by ϵf_t. We get the utility of the attacker as the following:

$$U_A = (f_b + f_t)\lambda + \epsilon f_t \alpha_A (1 - \alpha_A)\lambda$$

The ϵ in the above equation is the rate of change of transaction fees over time. To determine this we have gathered data on blocks. In Fig. 3.5, 3.6, 3.7, and 3.8* we have plotted the transaction fees with respect to block generation time. The slope of the straight line is our desired ϵ. For most of the cases, ϵ is almost equal to 0.2. We have created the plot and from the slopes we estimated the range of ϵ to be [0.2,0.6]. Now we will show the increment of the attacker's revenue when it follows the transaction exclusion attack. In Fig 3.9, the x-axis is the estimated range of ϵ and the y-axis represents the increment in the attacker's revenue. Each curve with a different color is for attackers with different computational power.

- **Red line:** For attacker with 15% computation power.
- **Green line:** For attacker with 20% computation power

*Plotted using data from https://gz.blockchair.com/bitcoin/blocks/

- **Blue line:** For attacker with 25% computation power
- **Cyan line:** For attacker with 30% computation power
- **Magenta line:** For attacker with 35% computation power

From figure 3.9, we can clearly see that an attacker with 15% computational power can increase its revenue by almost 4% if ϵ is around 0.35. Hence, we can conclude using this strategy that if ϵ is bigger then miners can increase their revenues.

Chapter 4

Conclusion and Future Work

4.1 Conclusion:

In conclusion, this study has demonstrated a new strategy *Risk-Free Colluding Selfish Mining* for generating more revenues with less computation power. We have gone through a detailed analysis of *Risk-Free Colluding Selfish Mining* and proved that our theoretical analysis is consistent with the simulation of the attack. In section 2.5 we can clearly see the lower bound for generating more revenue reduces to 20% and 17.5% in our strategy with only 10% and 15% collaborators respectively.

In Chap 3 we have analyzed the scenario where an attacker only publishes the header instead of the full block. In this case, we have a model with n rational miners and the attacker. We have calculated the utility of one rational miner P_i for two cases:

1. Mine on top of the block for which only the header is published.
2. Mine on top of the previous block.

As in Fig 3.4 it is clear the difference is positive for all the cases we have considered. Hence, we can conclude that for the rational miner P_i mining on top of the block for which only the header is published is more beneficial than mining on top of the previous block when the ratio of transaction fees and total fees is less than 0.2 even if most of the other miners(R_2) are mining on top of the previous block. We have also shown that the attacker can increase its revenue using this strategy depending upon the value of ϵ, which is the rate of increase in transaction fees as the time for block generation increases.

4.2 Future Work:

In the end, we will explore some future aspects of the results we have gotten through this thesis. We can use these results to design attack strategies that can compromise Bitcoin security. There are many other popular attack strategies one of which is the *Block Withholding Attack* (BWH) [Ros11]. In this attack, the attacker exploits the reward distribution system in a mining pool. The pool manager determines the hash power of other miners using a partial PoW and distributes the reward of a block accordingly. In this case, the attacker produces the partial proof of work but discards the full PoW. In this way, the attacker gets revenue from the pool without ever contributing to the pool.
In the paper *Miner's Dilemma* [Eya15] we see the analysis of pools launching *Block Withholding Attack* using infiltrating miners inside other pools.
In *FAW* attack [KKS+17], we see the attacker divides its computation power into two parts : (1) innocent mining and (2) infiltration mining. The infiltrators do the BWH attack but publish the FPoW when other honest miners outside the pool create a block and hence create a fork.

In chap 3 we have seen that the attacker can only publish the header of a full block and miners mining on top of that benefit. In future, we plan to combine the results discussed in this thesis with some attacks (BWH attack, FAW attack) and propose another attack strategy that can compromise Bitcoin security.

Bibliography

[ES13] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. *CoRR*, abs/1311.0243, 2013.

[Eya15] Ittay Eyal. The miner's dilemma. In *2015 IEEE symposium on security and privacy*, pages 89–103. IEEE, 2015.

[HKZG15] Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. Eclipse attacks on {Bitcoin's}{peer-to-peer} network. In *24th USENIX security symposium (USENIX security 15)*, pages 129–144, 2015.

[KKS+17] Yujin Kwon, Dohyun Kim, Yunmok Son, Eugene Vasserman, and Yongdae Kim. Be selfish and avoid dilemmas: Fork after withholding (faw) attacks on bitcoin. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 195–209, 2017.

[MJP+19] Michael Mirkin, Yan Ji, Jonathan Pang, Ariah Klages-Mundt, Ittay Eyal, and Ari Juels. Bdos: Blockchain denial of service. *CoRR*, abs/1912.07497, 2019.

[Nak08] Satoshi Nakamoto. Bitcoin whitepaper. *URL: https://bitcoin. org/bitcoin. pdf-(: 17.07. 2019)*, 9:15, 2008.

[NKMS16] Kartik Nayak, Srijan Kumar, Andrew Miller, and Elaine Shi. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In *2016 IEEE European Symposium on Security and Privacy (EuroSP)*, pages 305–320, 2016.

[Ros11] Meni Rosenfeld. Analysis of bitcoin pooled mining reward systems. *arXiv preprint arXiv:1112.4980*, 2011.