Rational Mining On Bitcoin

Soumen Pachal

Rational Mining On Bitcoin

Dissertation submitted in partial fulfillment of the requirements for the degree of

Master of Technology
in
Computer Science

by

Soumen Pachal
[Roll No: CS-1504]

under the guidance of

Dr. Sushmita Ruj
Assistant Professor
Cryptology and Security Research Unit

Indian Statistical Institute
Kolkata-700108, India

July 2017

To my family and my supervisor

CERTIFICATE

This is to certify that the dissertation entitled **"Rational Mining On Bitcoin"** submitted by **Soumen Pachal** to Indian Statistical Institute, Kolkata, in partial fulfillment for the award of the degree of **Master of Technology in Computer Science** is a bonafide record of work carried out by him under my 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.

Dr. Sushmita Ruj
Assistant Professor,
Cryptology and Security Research Unit,
Indian Statistical Institute,
Kolkata-700108, INDIA.

Acknowledgments

I would like to show my highest gratitude to my advisor, *Sushmita Ruj*, Cryptology and Security Research Unit, Indian Statistical Institute, Kolkata, for her guidance and continuous support and encouragement. She has literally taught me how to do good research, and motivated me with great 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.

Finally, I am very much thankful to my parents and family for their everlasting supports.

Last but not the least, I would 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.

Soumen Pachal
Indian Statistical Institute
Kolkata - 700108 , India.

Abstract

Bitcoin is a decentralized cryptocurrency. It is secure as long as majority of the computational resources are with honest miners who follow the Bitcoin protocol. There has been several attacks on Bitcoin mining process in recent years. Eyal showed a strategy called *selfish mining* by which miners can get more reward than their fair share. If 33% of the miners are follow the selfish mining strategy, then the Bitcoin system will no longer remain decentralized. We propose a new mining strategy called the *Rational Mining*, following which only 28% of miners are enough to make Bitcoin decentralized.

We analyze the different strategies and show how a miner can choose a strategy to maximize its gain under different parameter selection.

Keywords: *Bitcoin mining, Mining pool, Selfish miner, Stubborn miner, Rational miner, Transition Probability.*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Bitcoin is a decentralize cryptocurrency, first proposed by Satoshi Nakamoto [17] in 2008. The transactions are written on a distributed publicly verifiable ledger. This public ledger is called as a *blockchain* and is maintained by Bitcoin peers (nodes) . A blockchain consist of many blocks. Blockchain records all transaction between Bitcoin users. Multiple transaction are put in one block. The blocks are connected in such a way that tampering with any transactions affects all the subsequent blocks. This makes the blockchain immutable. The security of the blockchain is established by a chain of cryptographic puzzles, solved by a loosely-organized network of participants called *miners*. The miners solve a cryptographic puzzle as a proof of work in order to receive incentive (in the form of Bitcoin). The more mining power of a miner, the better are its chances to solve the puzzle first. The Bitcoin protocol requires a majority of mining resources are with honest miners. By construction, if a set of colluding miners comes to command a majority of the mining power in the network, the currency stops being decentralized and becomes controlled by the colluding group.

Bitcoin uses the concept of Proof-of-work(PoW). Proof-of-Work consists of cryptographic puzzle [6]. To form a block, miners collect the pending transactions instead of verifying the individual transactions. To validate a block, miners calculate a hash of this block and vary a nonce value until this hash value is less than a given target value. This target value is called the *difficulty level.* Solving the puzzle is computationally hard. Bitcoin uses the SHA-256 hash function. Miner choose nonce value in a brute force manner. For a nonce value, if the hash value is not lower than target value, then the only option is to try different nonces until a solution is obtained i.e, hash value is less than the target value. Thus the difficulty of the puzzle depends on target value. A miner immediately publish in the network a block when a valid hash (less than target value) value is found. When other miners receive that block, they can easily verify its correctness by comparing the hash value with the target value. A block is said to be a valid block if the majority of peers validate this block. The peers also update their local blockchain by adding this newly created block. After a successful block is added in the blockchain, the miner who first solved the Proof-of-Work will be rewarded with newly generated coins. Currently, the mining reward is 12.5 BTCs and this reward is reduced by half every four years. A small amount of transaction fees will also be rewarded. A target value is changed after every 2016 blocks approximately to fulfill the fairness and average waiting time for block validation. This adjustment of target value helps to keep per block verification time to approximately 10 minutes. In [13], Kraft propose an equation to change the target value for the Bitcoin system. The equation is

$$T_{new} = \left(\frac{G_{time}}{2016 * 10min} \right) * T_{old} \tag{1.1}$$

where T_{new} and T_{old} are the new target value and previous target value respectively, G_{time} is the time period to generate the last 2016 blocks in the Bitcoin.

Blockchain is a link-list based data structure. It store the transaction history in terms of block that combine the transaction in the merkle tree [16]. The blockchain increase in length as miners mine continuously in the network. A miner calculates a valid hash value for the block, adds the block in the local chain and broadcast it, the rest of the network check its validity. If it is correct, then miners update their local chain otherwise discard the block.

It may so happen that two valid solution are found in same time (approximately) or due to latency problem, distribution of a verified block is delayed. In these situations blockchain fork is created. When multiple branches appear, miners are free to choose a branch and mine at the top of that branch. The longer branch is accepted by the network. Bitcoin community recommended that after a block is mined it should receive enough block confirmations, currently 6 confirmations before the transactions in it are treated as valid transactions. Thus, a transaction is validated successfully on an average one hours later.

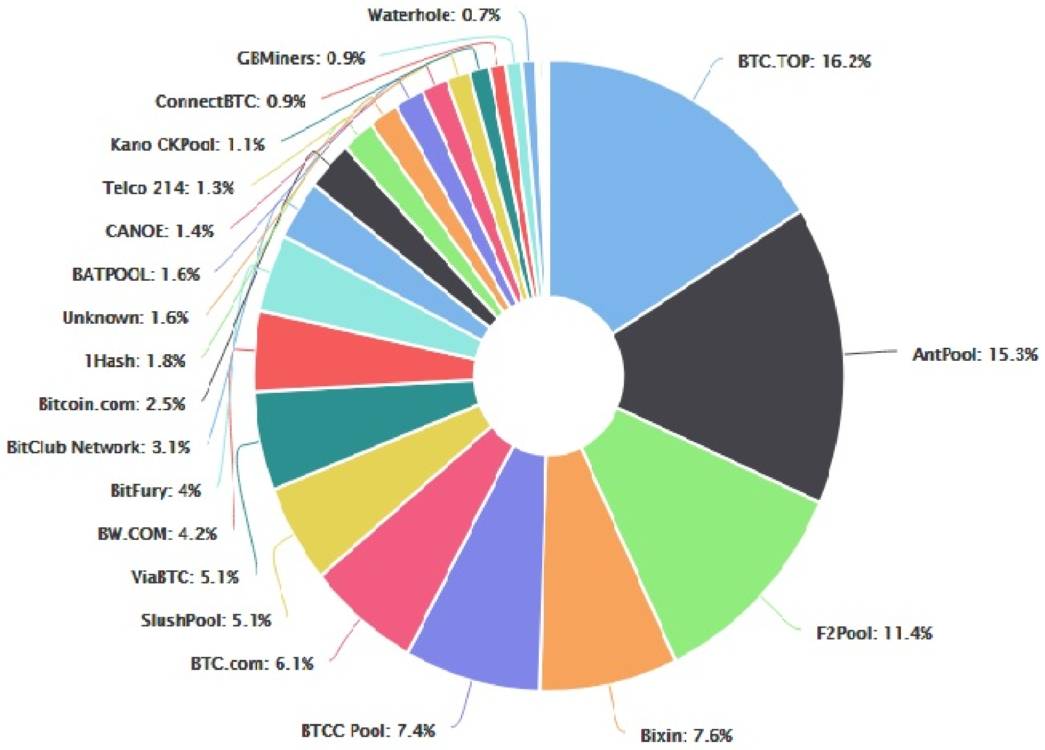

Figure 1.1: Mining Power Distribution in Present Market(June,2017)

In order to maximize their chance of solving the puzzle, multiple miners join hands in order to form a *mining pool*. This helps to sum up their computing power. In such a mining pool every miner needs to regularly submit a proof of work to the pool administrator to demonstrate their work towards solving the puzzle associated with a Bitcoin block. Every miner of pool attempts to find a PoW on a transaction set that contains a coinbase transaction which separates this transaction set from the transaction sets of other mining pools or solo miners. All members of a pool work together to mine blocks, and share their revenues when one of them successfully mines a block. The revenue is divided among its members according to their relative mining power. From figure 1.1, the three largest mining pool size is 42%. The largest

Table 1.1: Attack Scenarios form 1.1

Computational Power(α)	Scenarios
42%	The three largest mining pools today
31%	The two largest mining pools today
16%	Largest pool today

mining pool size is 16% today(June,2017). From [4], the hash power of a pool called GHash.IO reached to 54% for a day i.e, it exceeds the theoretical attack threshold of 51% in Bitcoin. Thus it is possible that size of a pool is greater than 50%.

Bitcoin is a decentralized cryptocurrency. Since it is decentralized, attackers find an easy way to fraud transactions. Double spending [3] is possible in Bitcoin. An user in the Bitcoin network makes a double spend if she spends the same set of coins in two different transactions simultaneously. Apart from double spending attacks, network level attacks, mining attacks are also found in Bitcoin network.

In [5], defined a class of mining attacks called *block withholding attacks*[14] where a miner gain more rewards by withholding a valid newly created block. Eyal and Sirer proposed *Selfish mining attack* [8] where a selfish miner, under certain conditions, can gain a disproportionate share of reward by deviating the honest miner, who follows the protocol. They show that a selfish miner controlling 33% of computing resources can completely disrupt Bitcoin system. Nayak et al.[19] proposed *stubborn mining strategy* and show that selfish mining [8] is not optimal for a large parameter space. A pool hopping attack is presented in [21]. Apart from this major attacks, we also saw some minor attack like Sybil Attack [7], eclipse attack [10]. Later we will discuss in details the selfish mining and stubborn mining strategies.

1.2 Our Contribution:

We formally define and analyze a new type of attack, in which a miner can get more revenue than both selfish mining and stubborn mining. We call this a *Rational Mining* strategy. A miner is called a *rational miner* if the miner mines honestly sometimes with probability p and mines selfishly/stubbornly with probability $(1 - p)$.

Let α be the computational power of a miner. We show that selfish mining/stubborn mining strategies are not always the best strategies for a miner who wants to maximize its gain. We show that by carefully choosing the parameters α and p, a rational miner can maximize its gain over selfish/stubborn miners.

The main contribution is to show that if a miner controls 28% of computational resources it can attack the Bitcoin system. This is a stronger result than [8], who showed that a miner needs to control 33% of resources to launch a full attack on Bitcoin.

In particular, we show that a miner with computational power more than 28% will get more revenue than honest miner if it chooses the honest mining strategy with probability 30%.

Our results show that it is possible for a miner with a given computational power to choose a strategy that maximize its gain. In our work we show two rational mining strategies. In the first case we show that a rational miner with computational power $\leq 44\%$ can get more revenue than selfish miner [8] if she (rational miner) chooses the honest mining strategy with probability $p = 14\% - 30\%$ approximately. In particular, a miner with computational power 40% can get more revenue than selfish miner if it chooses 18% honest mining strategy. Our results show that it is possible for a miner with a given computational power to choose a strategy that maximize its gain.

We also apply our strategy on [19] and call it our second rational mining strategy. In our second rational mining strategy, we show that a miner with computational power $19\% - 45\%$ can get more revenue than our first rational mining strategy as well as selfish mining strategy [8] if the second rational miner chooses the honest mining strategy with probability 12%.

1.3 Organization:

The rest of the thesis is organizes as follows: In Chapter 2, we discuss related work. In Chapter 3, we discuss our first rational mining strategy. In this chapter, we mathematically calculate the revenue of first rational miner and compare the results with other mining strategies. In Chapter 4,we present our second rational mining strategy. In this chapter also, we mathematically calculate the revenue of second rational miner and compare the result with other mining strategy. We conclude the thesis in Chapter 5.

Chapter 2

Related Work

2.1 Attack on Bitcoin Mining

In 2011, Rosenfled proposed Block with holding Attack. In Block with holding attack, pool member withholds an already mined block to waste resources of honest miners and decreases the pool revenue. In [21] authors discuss Block with holding attack and considers it as a non-incentivized sabotaging attacks, simply to sabotage the pool profits. In [21],two type of block withholding attack are presented called (1) Sabotage and (2) Lie in wait. In first case attacker does not gain any coins, but it just makes the other miners to loose. While in the second case, attacker performs a block concealing attack like selfish mining attack[8], stubborn mining attack [19].

In 2013, Eyal and Sirer [8] discovered selfish mining attack.In their paper in section 7 entitled Related Work, we read: “In a block with holding Attack, a pool member decreases the pool revenue by never publishing blocks it finds.

We now describe the selfish mining strategy. Selfish miner follow the following strategy:

(1) When lead = 2 and honest miner mines the next block, then rational miner reveal her entire chain.

(2) When lead $= 0'$ and rational miner mines next block, then rational miner reveal her private chain.

(3) When lead $= 0$ or lead > 0 and Rational miner mies the next block, then Rational miner do not reveal her private chain.

A selfish miner can gain a disproportionate share of reward under certain conditions than honest miner. In [8], authors show that the Bitcoin mining protocol is not incentive-compatible. They present an attack with which colluding miners obtain a revenue larger than their fair share. A selfish miner can gain an unfair share of the block reward by deviating from the honest miner. Specially a selfish miner with more than 33% computational power get disproportionate gain by maintaining private block chain and with holding blocks that have been mined. In this case honest miner forced to perform wasted computations on a stale public branch. Selfish Mining works because honest miners are forced to spend their computation cycles on blocks that are destined to not be on the public chain.

In 2016, Nayak et. al. [19] proposed Stubborn mining attack. First we describe the stubborn mining strategy. A stubborn miner follows the following strategy:

A selfish miner would immediately reveal her private chain when lead $= 2$ and honest miner finds next block and closes the gap by 1. Here instead of revealing her entire private chain, rational miner reveals the next block on her private chain only so that the length matched with the public chain and the state transitions to lead $1'$. In general, When lead was more than 2 and honest miner finds the next block, a selfish miner does not reveal her private chain and thus the state machine transitions to lead $= k - 1$, but here rational miner would reveal immediately one block so that the chain divided into two forks and thus the state transition goes to lead $(k - 1)'$. They show that Selfish Mining is not optimal for a large parameter space. They also show that, Stubborn miner can get 25% more revenue than selfish miner without any network level attack. A non-trivial combinations of stubborn mining and network-level attack will increase attacker's revenue. Stubborn mining strategies can perform up to 25% better than Selfish mining for many reasonable values of α and γ.

Heilman et al. demonstrated a network-level eclipse attack [10] where a single node monopolizes all possible connections to a victim and eclipses it from the network. Their paper describes elaborate techniques to achieve eclipse attack on the Bitcoin network.

We saw many attacks on bitcoin protocol. Now we discuss the possible countermeasures for the bitcoin attacks.

2.1.1 Countermeasures:

Countermeasures for Double Spending:

One of the main problems in Bitcoin technology is double spending. One default solution used against this problem is proof of work technique. Use of this technique limits the capabilities of an adversary in terms of its computational resources. There are two possible ways to deal with double spending. They are as follows:

- detect a double spending instance by monitoring the blockchain progress and once detected, identify the adversary and take some actions.
- use preventive measures.

The first method works well in centralize online banking system, but in Bitcoin its not suitable due to use of continuously varying the public keys as a wallet address, thus it provides anonymity to users, and the lack of transaction rollback scheme once it is successfully added in the blockchain. Therefore the second approach is more desirable in Bitcoin. In [12] authors describe three techniques that can be used to detect a possible double spending in fast payment systems:

- using a listening period.
- inserting observers.
- forwarding double spending attempts.

In this approach, a peer checks whether a transaction is an attempt to double spend whenever it receive a new transaction, if so, then peer forward the transaction to their neighbors.

In [12], to control the double spending, authors proposed another solution, where all the participating users deposit a safety amount similar to an agreement. If an attacker tries to double spend and it is detected, the deposit amount will be deducted.

In [2][4], authors described a countermeasure by prohibiting the merchant to accept incoming connections, thus an adversary cannot directly send a transaction to the merchant. This forces the adversary to broadcast the transaction over the Bitcoin network, and it ensures that the transaction will end-up in the local view of all the miners that forwards it. Later if the adversary tries to double spend the miners will know about it.

From [4], we have seen a method called "proof of reputation", where the honest miners will get a token based on the current market value. The number of tokens issued can vary with the market value. If the miner has the token, it will be reputed in the mining market pool. The token has a value, and according to which the coins are deposited from all the miners from time to time and is fixed by the network. More the reputation of the miners chain, more the other blocks merge with that chain.

Countermeasure for Private Forking:

A dishonest miner can privately mines set of blocks and by doing this the miner can intentionally forks the blockchain. Now this makes the Bitcoin network vulnerable to various attack such as block discarding attack, block withholding attack, selfish mining attack, bribery attacks to name a few. These attacks mainly aimed to cheat Bitcoin mining incentive system. Therefore at any point of time a major challenge for Bitcoin Protocol developer is detecting and mitigating the faulty forks from the set of available forks. If at any instance of time a miner encounter the presence of multiple forks of same length, all peers of it is notified with that information and it randomly chooses one fork to extend. By this approach, number of branches in the blockchain can be decreased which in turn decreases the ability of selfish mining.

In [9] authors introduced the concept of Freshness Preferred which places the unforgeable timestamps in blocks and prefer blocks with recent timestamps. This approach uses Random Beacons [20] in order to stop miners from using timestamps from the future. As the selfish mining uses strategic block withholding technique, the proposed strategy will decrease the incentives for selfish mining because withheld blocks will lose block races against newly minted or “fresh” blocks. A similar but most robust solution for selfish mining that requires no changes in exiting bitcoin protocol proposed in [22] [18]. The authors suggest a fork-resolving policy that selectively neglect blocks that are not published in time, and it appreciate blocks that includes pointer to competing blocks of their predecessors. Therefore, if the secretly mined block is not published in the network until a competing block is published, it will contributes to neither or both branches, thus it gets no benefits in wining the fork race.

Countermeasure for Block Withholding Attack:

In Block with holding attack [21][15], pool member withholds an already mined block to waste resources of honest miners and decreases the pool revenue. In [5], authors discussed some countermeasure for block withholding attack. In pool, include only known and trusted miner. If the revenue drops from the expectation, close the pool as soon as possible. In [1], authors discussed some cryptographic commitment schemes for countering the block withholding attack.

Countermeasure for Eclipse Attack:

In eclipse attack[10], dishonest miner manipulates the others miners. The IP addresses to which the eclipsed miner uses are diverted towards the attacker. An eclipse attacker can hold multiple IP address to disconnect the eclipsed miner from rest of the network. Two types of attack are found in eclipse attack [10]: (i) Infrastructure attacks and (ii) Botnet attacks. First attack is on internet service provider which holds the multiple addresses. It can detect multiple addresses which connects peer-to-peer in the network. In botnet attacks, an attacker can detect addresses in a particular range. In both cases eclipsed attacker manipulates the bitcoin network. In [10] authors discussed some possible countermeasure for eclipse attack. By using whitelists, disabling all incoming connections one can stop eclipse attack.

Chapter 3

Rational Mining Strategy

Let α be the fraction of the network's total hashpower maintained by the attacker and β be the fraction of the hashpower of the honest miner such that $\alpha + \beta = 1$. If a miner follows the protocol, then we called this miner as a honest miner. If a miner follows the selfish mining strategy [8], then we called this miner as a selfish miner. A miner is called a rational miner if the miner mines honestly sometimes with probability p and mines selfishly with probability $(1 - p)$.

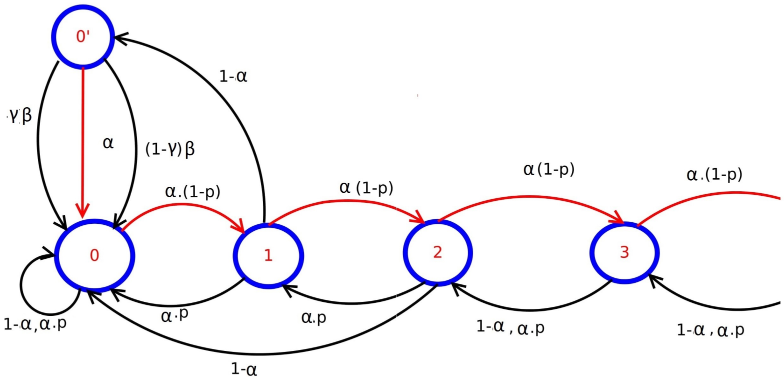

Figure 3.1: State Machine

Now we analyze the expected reward for our system. In this chapter, first we calculate the transition probabilities of each state by considering a state machine. Secondly, we calculate the reward of the rational miner. Lastly we compare our result with [8]. In our result a miner with computational power $\leq 44\%$ can get more reward than [8]. Also we represent the graphs with different choice of α, γ and p.

First we represent the blockchain at different states. (%Here upper one is public chain and lower one is private chain)

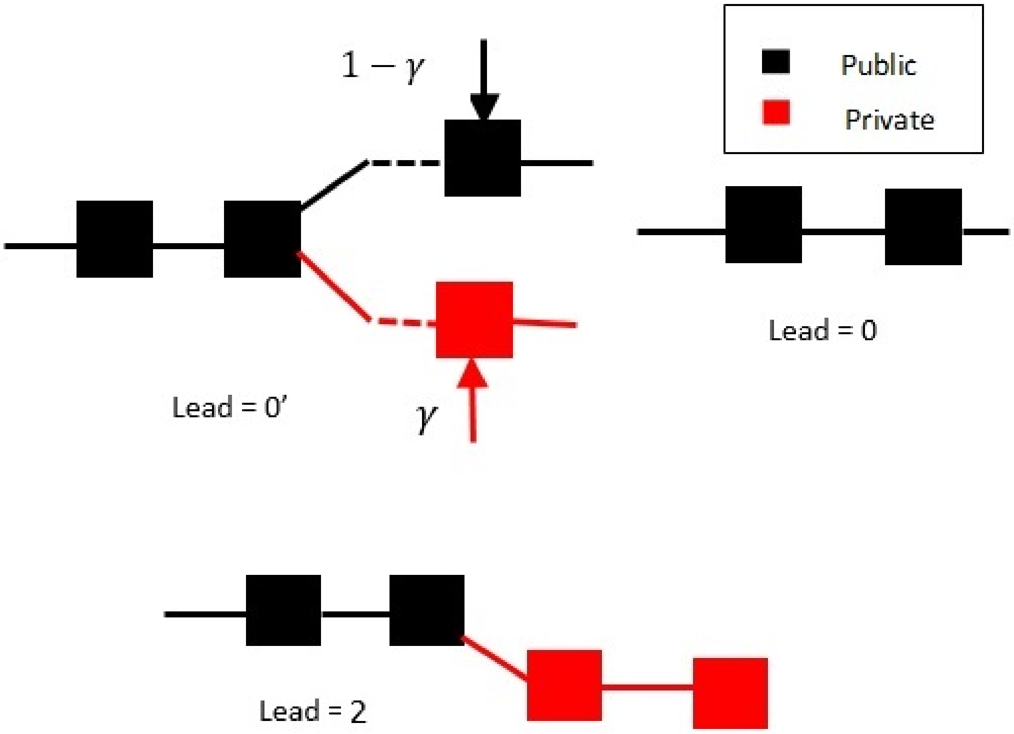

Figure 3.2: State Description

Figure 3.1 illustrate the progress of the system as a state machine. The states of the system represent the lead of the rational miner. Lead is defined as the difference between the length of private chain(i.e, hidden from the rest of the network), maintained by rational miner and length of public chain i.e, lead is the difference between the number of unpublished blocks of rational miner and the public chain. The State $0'$ denotes that there exist a fork, the revealed portion, the two forks are equal in length and honest miner divided between mining on these two forks, i.e, State 0 is the state where there is no fork. There is only one public chain. State $0'$ is the state where there are two public branches of length one: the main chain and branch that was maintained by rational miner and publish at that time to match the main chain.

Also we denote γ, the ratio of honest miners that choose to mine on the rational miner's private chain and the other (1-γ) of the honest miner mine on the main branch.

For state t = 0,1,2,... with computational power $\alpha.(1-p)$ rational miner mines a block and in that case lead will increased by 1 and with computational power $\alpha.p$, the rational miner mines honestly i.e, in that case rational miner publish a block and lead decreased by 1. In state t, with frequency (1-α), the honest miner mines a block and in that case also lead will be decreased by 1 i.e, lead will be $(t-1)$.

Table 3.1: Table of Notation

Name	Description
α	Computational power of Rational miner
β	Computational power of honest miner
γ	Fraction of honest miner's network that will mine rational miner block when honest and rational miner have released a block at the same time resulting in an equal length fork.
p	Probability that Attacker mines honestly.
$0, 1, 2, 3, \ldots$	Lead by rational miner
$0', 1', 2', \ldots$	There is a fork and revealed portion of forks are equal in length.
R_{hm}	Honest miner revenue
R_{rm}	Rational miner revenue
$Lead$	Difference between the length of rational miner's chain and honest miner's chain.

3.1 State Description

In honest mining strategy, honest miner reveals the block immediately after mining it. In selfish mining strategy, selfish miner follow some strategy: (1) When lead = 2 and honest miner mines the next block, then rational miner reveal her entire chain. (2) When lead = $0'$ and rational miner mines next block, then rational miner reveal her private chain. (3) When lead = 0 or lead > 0 and Rational miner mies the next block, then Rational miner do not reveal her private chain.

For example, in a particular state, if a rational miner mines a block, then she will take decision that she will publish the block or keep it in her own private chain. If she keep it privately, lead will increased by 1, otherwise decreased by 1.

3.2 State Probabilities

From the state machine 3.1, we calculate the probabilities distribution over the state space. Here P_i is the probability of being in the i-th state.

3.2.1 Transition Probability Calculation

Let S_{ij} represents the transition to move from i-th State at $(t-1)$th time to j-th State at t-th time. Now we calculate the transition probability P_i for the i-th State. Also $i^{(t)}$ denotes the i-th state at t-th time.

0-state:

$S_{00} : 0^{(t-1)} \to 0^{(t)}$, $S_{10} : 1^{(t-1)} \to 0^{(t)}$, $S_{20} : 2^{(t-1)} \to 0^{(t)}$, $S_{0'0} : 0'^{(t-1)} \to 0^{(t)}$

$$\begin{aligned} P_0 &= (1-\alpha)P_0 + \alpha p P_0 + \gamma\beta P_{0'} + (1-\gamma)\beta P_{0'} + \alpha P_{0'} + \alpha p P_1 + (1-\alpha)P_2 \\ \Rightarrow \alpha(1-p)P_0 &= (1-\alpha+\alpha.p)P_1 + (1-\alpha)P_2 \end{aligned}$$

$0'$-state:

$S_{10'} : 1^{(t-1)} \to 0'^{(t)}$

$$S_{0'} = (1-\alpha)S_1$$

1-state:

$S_{01} : 0^{(t-1)} \rightarrow 1^{(t)}$, $S_{21} : 2^{(t-1)} \rightarrow 1^{(t)}$

$$\begin{aligned} P_1 &= \alpha(1-p)P_0 + \alpha p P_2 \\ \Rightarrow \alpha(1-p)P_1 &= (1-\alpha+\alpha.p)P_2 \end{aligned}$$

2-state:

$S_{12} : 1^{(t-1)} \rightarrow 2^{(t)}$, $S_{32} : 3^{(t-1)} \rightarrow 2^{(t)}$

$$\begin{aligned} P_2 &= \alpha(1-p)P_1 + \alpha p P_3 + (1-\alpha)P_3 \\ \Rightarrow \alpha(1-p)P_2 &= (1-\alpha+\alpha p)P_3 \end{aligned}$$

k-state:

$S_{k-1,k} : (k-1)^{(t-1)} \rightarrow k^{(t)}$, $S_{k+1,k} : (k+1)^{(t-1)} \rightarrow k^{(t)}$

$$\begin{aligned} P_k &= \alpha(1-p)P_{(k-1)} + (1-\alpha)P_{(k+1)} + \alpha p P_{(k+1)} \\ \Rightarrow \alpha(1-p)P_k &= (1-\alpha+\alpha p)P_{(k+1)} \end{aligned}$$

Thus we obtained the following equations:

$$\alpha(1-p)P_0 = (1-\alpha+\alpha.p)P_1 + (1-\alpha)P_2 \tag{3.1}$$

$$\alpha(1-p)P_1 = (1-\alpha+\alpha.p)P_2 \tag{3.2}$$

$$\forall k \geqslant 2, \alpha(1-p)P_k = (1-\alpha+\alpha p)P_{(k+1)} \tag{3.3}$$

$$P_{0'} = (1-\alpha)P_1 \tag{3.4}$$

$$\sum_{k=0}^{\infty} P_k + P_{0'} = 1 \tag{3.5}$$

From (5.2) and (5.3) we get,

$$\forall k \geq 2, P_k = \left(\frac{\alpha(1-p)}{1-\alpha+\alpha p}\right)^{k-1} P_1 \tag{3.6}$$

From (5.1) we get,

$$\alpha(1-p)P_0 = (1-\alpha+\alpha p)P_1 + \left(\frac{(1-\alpha)\alpha(1-p)}{1-\alpha+\alpha p}\right) P_1$$

$$\Rightarrow P_0 = \left[\frac{1-\alpha+\alpha p}{\alpha(1-p)} + \frac{1-\alpha}{1-\alpha+\alpha p}\right] P_1 \tag{3.7}$$

From (5.5) we get,

$$\begin{aligned} \sum_{k=0}^{\infty} P_k + P_{0'} &= 1 \\ \Rightarrow 1 &= \left[\frac{1-\alpha+\alpha p}{\alpha(1-p)} + \frac{1-\alpha}{1-\alpha+\alpha p}\right] P_1 + \sum_{k=1}^{\infty} \left(\frac{\alpha(1-p)}{1-\alpha+\alpha p}\right)^{(k-1)} P_1 + (1-\alpha)P_1 \end{aligned}$$

$$P_1 = \frac{\alpha(1-p)(1-\alpha+\alpha p)(1-2\alpha+2\alpha p)}{(1-\alpha+\alpha p)^3 + (1-\alpha)\alpha(1-p)(1-2\alpha+2\alpha p)(2-\alpha+\alpha p)} \tag{3.8}$$

The above expression hold if $2\alpha - 1 \leq 2\alpha p$. Using the value of P_1, we can obtained all other values.

3.3 Reward Calculation

Now we analyze the Revenue of the rational miner and honest miner using the transition probability over the state space. If the block end up in the main chain, then only the miner will get the the block reward.

Case 1: Two branches of length 1 and honest miner mines on the rational miner chain. In that case rational miner and honest miner obtain a revenue of one each.

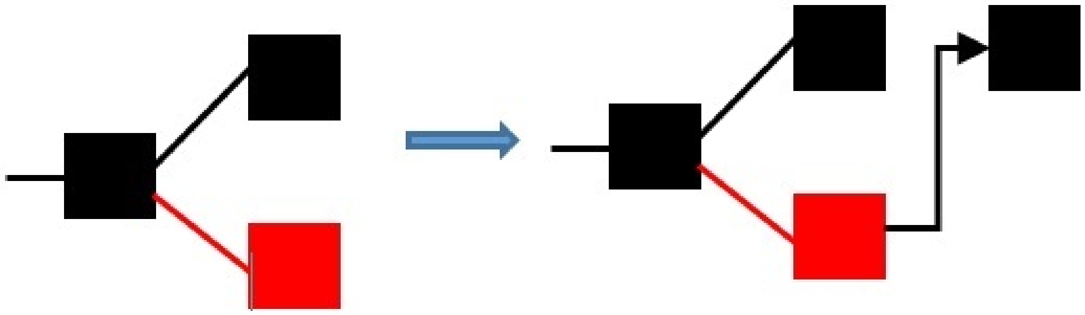

Figure 3.3: Honest miner mines on rational miner's chain in state $0'$

Case 2: Two branches of length 1 and the rational miner mines a block. In that case rational miner will get a revenue of 2.

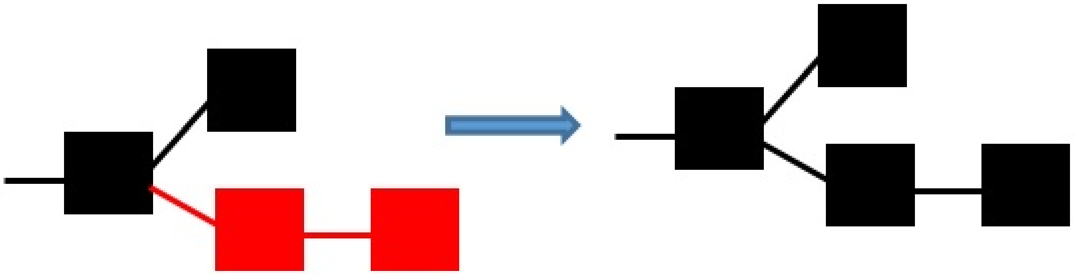

Figure 3.4: Rational miner mines a block in state $0'$

Case 3: Two branches of length 1. Honest miner mines a block on the main chain. In that case Honest miner will a revenue of 2.

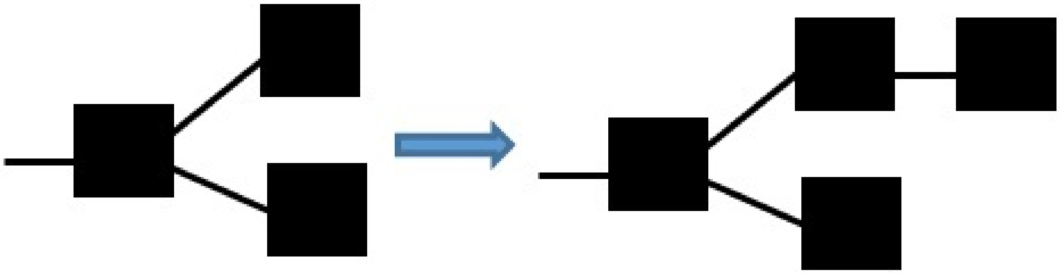

Figure 3.5: Honest miner mines a block on honest chain in state $0'$

Case 4: There is no secret block. Honest miner mines a block. In that case honest miner will get a revenue.

Case 5: Lead is 2 and honest miner mines a block, Then Rational miner will publish two blocks. In that case Rational miner will get a revenue of 2 as it is longer than previous chain.

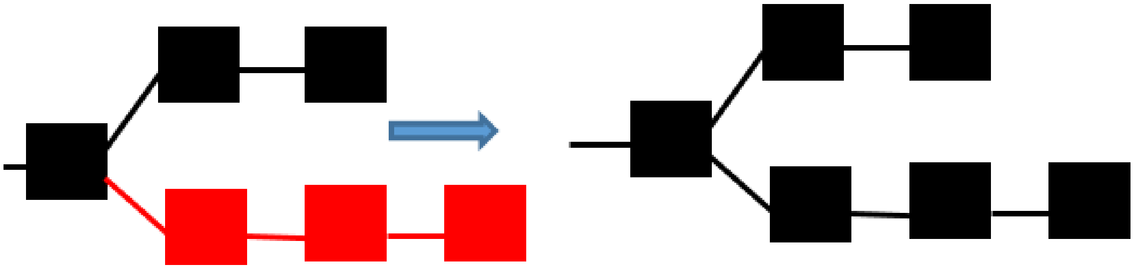

Figure 3.6: Honest miner finds a block in honest chain in state 2

Case 6: Lead is more than 2, honest miner mines a block. In that case Rational miner will get a revenue of 1.

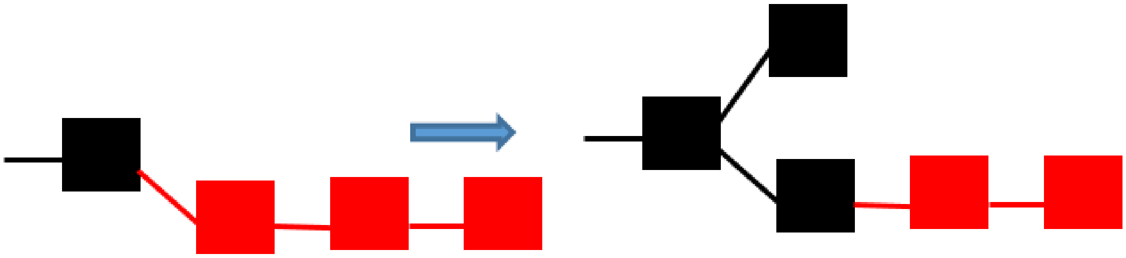

Figure 3.7: Honest miner mines on honest chain in state > 2

Now we calculate the revenue of the honest miner and Rational miner by using transition probabilities.

$$R_{hm} = P_{0'}.\gamma(1-\alpha).1 + P_{0'}.(1-\gamma).(1-\alpha).2 + P_0(1-\alpha).1 \tag{3.9}$$

$$\begin{aligned} R_{rm} = P_{0'}.\alpha.(1-p).2 + P_{0'}.\gamma.(1-\alpha+\alpha p) + P_2.(1-\alpha).2 + \alpha.p.Pr[i \geq 0] \\ +(1-\alpha+\alpha.p).Pr[i > 2].1 + P_0.\alpha.p \end{aligned} \tag{3.10}$$

Where R_{hm} and R_{rm} are the reward of the honest miner and rational miner respectively and $Pr[i \geq 0] = (1 - P_{0'})$ and $Pr[i > 2] = (1 - P_{0'} - P_0 - P_1 - P_2)$.

Now we calculate the revenue rate ratio by using the sate transition probability.

$$R_{rationalratio} = \frac{R_{rm}}{R_{rm} + R_{hm}} \tag{3.11}$$

$$R_{honestratio} = \frac{R_{hm}}{R_{rm} + R_{hm}} \tag{3.12}$$

Where $R_{rationalratio}$ and $R_{honestratio}$ are the revenue rate ratio of rational miner and honest miner respectively.

3.4 Comparative Study of Mining Strategies

Now we describe our theoretical result with graphs. we have use C language for implementing the theoretical results. We compare this result with selfish mining [8].

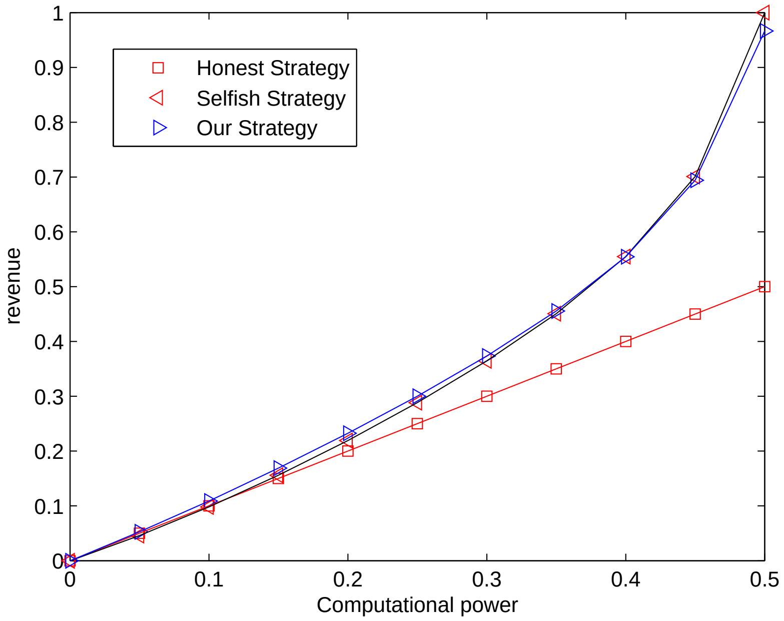

Figure 3.8: Comparison between selfish mining and our mining strategy with $p = 14\%$

In 3.8, we draw the revenue graph with respect to computational power α. In this graph we take $\gamma = 0.85$. From [19], we will take the range of γ as $0 \leq \gamma \leq 0.92$. In [8], we saw that a miner with more than 33% computational power can get more revenue than honest miner by maintaining her private chain and with holding blocks. In our result we see that from 3.8 any miner with $\alpha \leq 0.38$ can get more revenue than selfish miner[8] if she chooses honest mining strategy with probability 14%. Thus if a miner with computational power 38% and chooses honest mining strategy with probability 14%, she can beat selfish miner.

In 3.9, we also take the value of γ as $\gamma = 0.85$. In this case, a rational miner with computational power 42% will get more revenue than selfish miner [8] if the rational miner chooses honest mining strategy with probability 20%.

In 3.10, a rational miner with computational power 44% and $\gamma = 0.8$ will get more

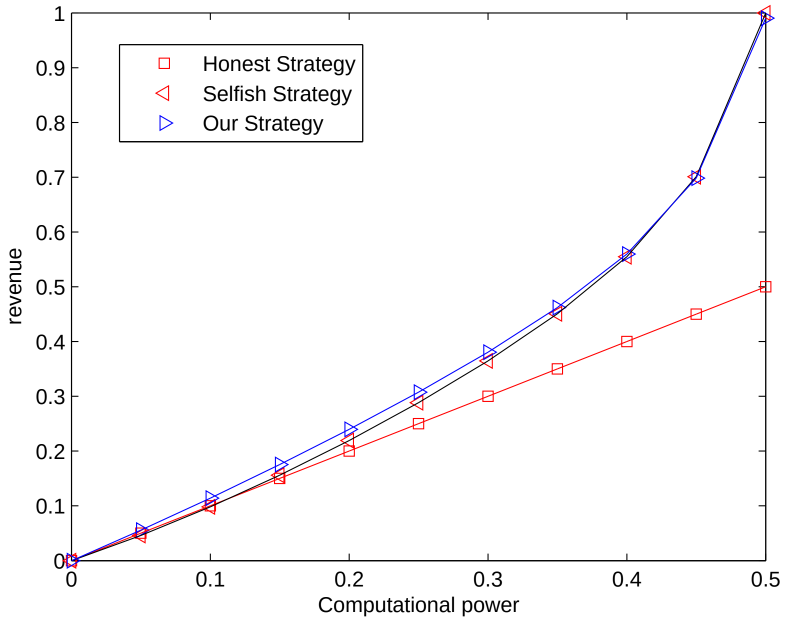

Figure 3.9: Comparison between selfish mining and our mining strategy with $p = 20\%$

revenue than selfish miner[8] if it chooses the honest mining strategy with probability 30%. From the above three graphs, we say that a rational miner with computational power less or equal to 44% can get more revenue than selfish miner [8] if it chooses the honest mining strategy with probability $14\% - 30\%$.

Now we draw a graph of revenue using rational mining strategy with different values of γ, compared to the honest bitcoin mining protocol.

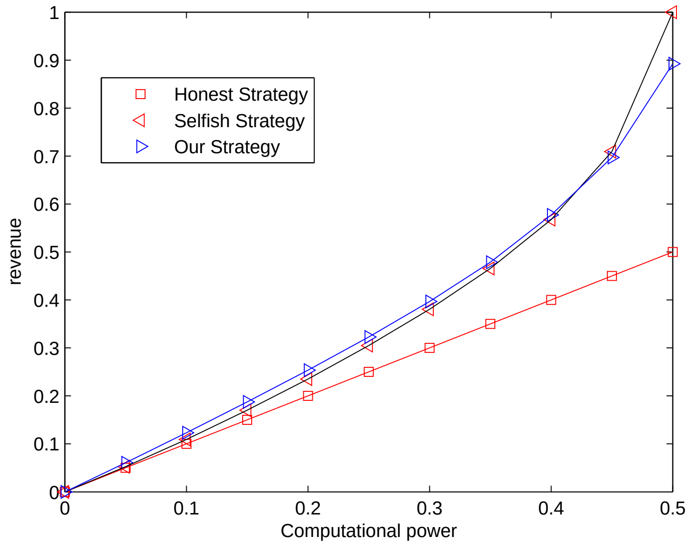

Figure 3.10: Comparison between selfish mining and our mining strategy with $p =$ 30%

In 3.11, we draw revenue graph using rational mining strategy and honest bitcoin mining protocol. In this graph rational miner chooses the honest mining strategy with probability 25%. From this graph we see that a rational miner with computational power more than 30% can get disproportionate gain than honest miner.

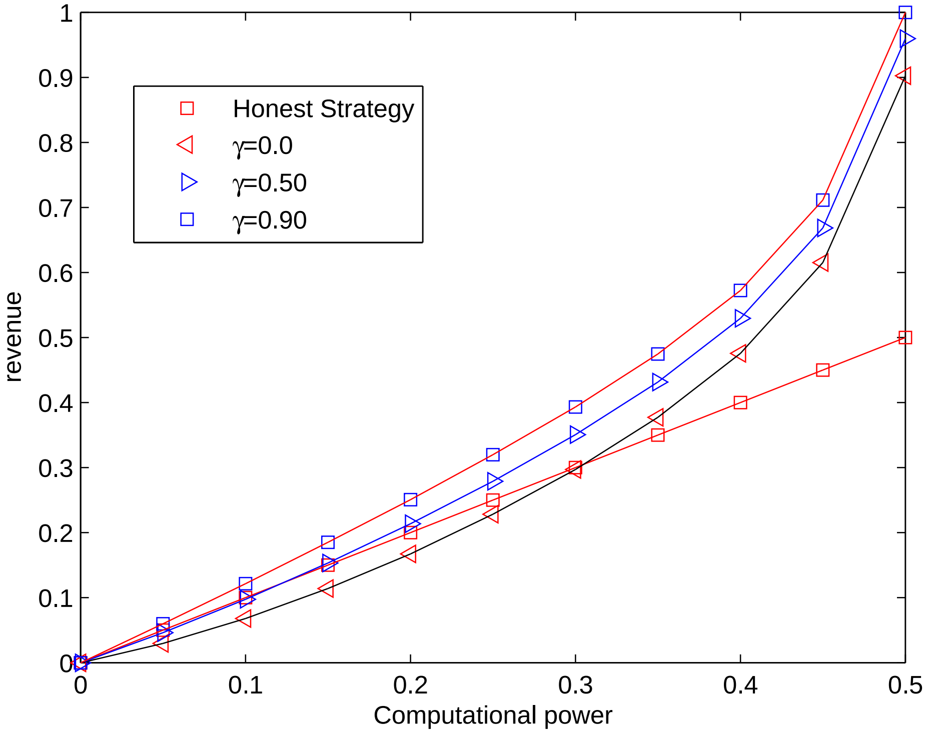

Figure 3.11: Revenue for different values of γ for $p = 25\%$

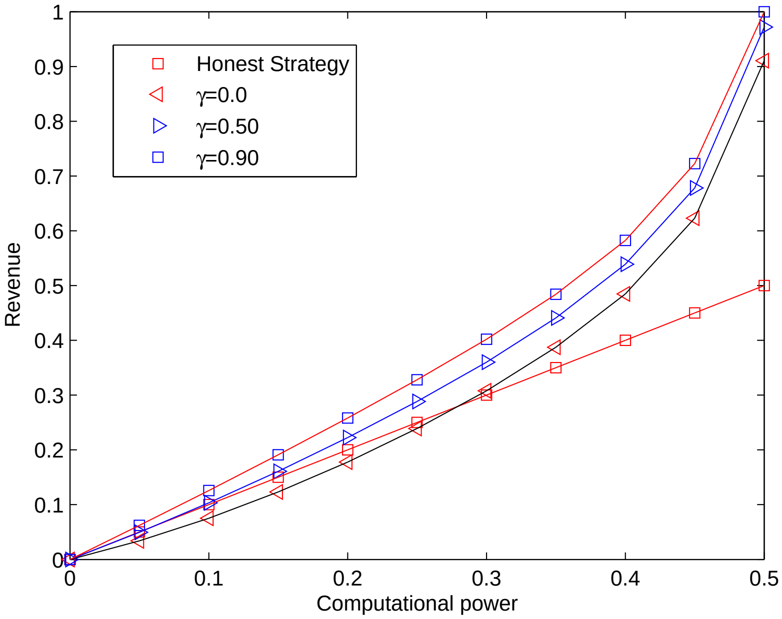

Figure 3.12: Revenue for different values of γ for $p = 30\%$

In 3.12, we draw revenue graph using rational mining strategy and honest bitcoin mining protocol. In this graph rational miner chooses the honest mining strategy with probability 30%. From this graph we can say that a rational miner with more than 28% computational power can get disproportionate gain than honest miner.

Chapter 4

Second Rational Mining Strategy

Let α be the fraction of the network's total hash-power maintained by the attacker and β be the fraction of the hashpower of the honest miner such that $\alpha + \beta = 1$. If a miner follows the protocol, then we called this miner as a honest miner. If a miner follows the stubborn mining strategy [19], then we called this miner as a stubborn miner. A miner is called a rational miner if the miner mines honestly sometimes with probability p and mines stubbornly with probability $(1 - p)$.

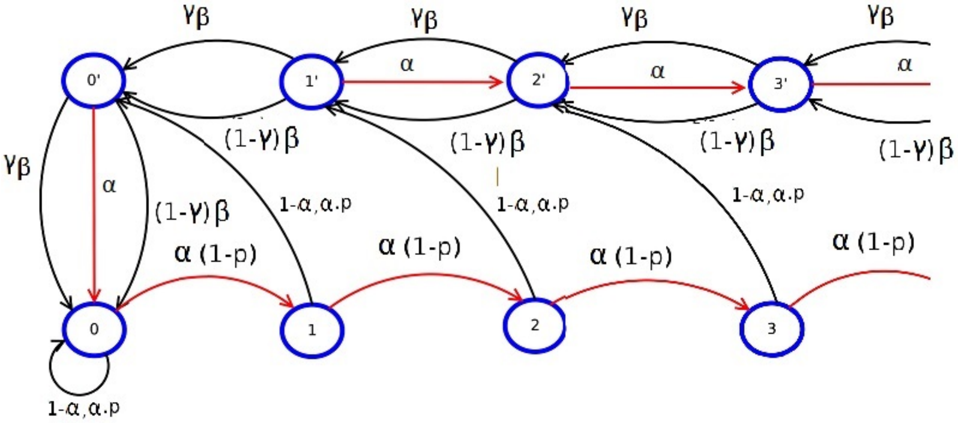

Figure 4.1: State Machine

In [8], a selfish miner would immediately reveal her private chain when lead $= 2$ and honest miner finds next block and closes the gap by 1. Here instead of revealing her entire private chain, rational miner reveals the next block on her private chain only so that the length matched with the public chain and the state transitions to lead $1'$. γ is the fraction of honest miner's network that will mine rational miner's chain when the revealed portions of two forks are equal in length.

If rational miner or γ fraction of honest miner advances rational miner's fork, then rational miner has successfully diverted a part of honest miner, $(1-\gamma)$ fraction, to do useless work. However if the $(1-\gamma)$ fraction of honest miner's succeeds in advancing rational miner's chain, rational miner may risk losing her private chain.

When lead was more than 2 and honest miner finds the next block, a selfish miner does not reveal her private chain and thus the state machine transitions to lead $=$ $k-1$, but here rational miner would reveal immediately one block so that the chain divided into two forks and thus the state transition goes to lead $(k-1)'$. From figure 4.1, in every state, with probability p rational miner mines honestly and mines stubbornly with probability $(1-p)$.

4.1 State Probabilities

From the state machine4.1, we calculate the probability distribution over the state space. Here P_i is the probability of being in the i-th state.

4.1.1 Transition Probability Calculation

Let S_{ij} represents the transition to move from i-th State to j-th State. Now we calculate the transition probability P_i for the i-th State. Also $i^{(t)}$ denotes the i-th state at t-th time.

0**-state:**

$S_{00} : 0^{(t-1)} \rightarrow 0^{(t)}, S_{00'} : 0'^{(t-1)} \rightarrow 0^{(t)}$

$$\begin{aligned} P_0 &= (1-\alpha)P_0 + \alpha p P_0 + \alpha P_{0'} + \gamma\beta P_{0'} + (1-\gamma)\beta P_{0'} \\ \Rightarrow P_{0'} &= \alpha(1-p)P_0 \end{aligned}$$

$0'$**-state:**

$S_{1'0'} : 1'^{(t-1)} \rightarrow 0'^{(t)}, S_{10'} : 1^{(t-1)} \rightarrow 0'(t)$

$$\begin{aligned} P_{0'} &= \gamma\beta P_{1'} + (1-\gamma)\beta P_{1'} + \beta P_1 + \alpha p P_1 \\ \Rightarrow P_{0'} &= (1-\alpha+\alpha p)P_1 + (1-\alpha)P_{1'} \end{aligned}$$

1**-state:**

$S_{01} : 0^{(t-1)} \rightarrow 1^{(t)}$

$$P_1 = \alpha(1-p)P_0$$

2**-state:**

$S_{12} : 1^{(t-1)} \rightarrow 2^{(t)}$

$$\begin{aligned} P_2 &= \alpha(1-p)P_1 \\ \Rightarrow P_2 &= \alpha^2(1-p)^2 P_0 \end{aligned}$$

$k \geq 1$,**k-state:**

$S_{(k-1)k} : (k-1)^{(t-1)} \rightarrow k^{(t)}$

$$\begin{aligned} P_k &= \alpha(1-p)P_{k-1} \\ \Rightarrow P_k &= \alpha^k(1-p)^k P_0 \end{aligned}$$

$1'$-state:

$S_{2'1'} : 2'^{(t-1)} \to 1'^{(t)}, S_{21'} : 2^{(t-1)} \to 1'^{(t)}$

$$\begin{aligned} P_{1'} &= \gamma\beta P_{2'} + (1-\gamma)\beta P_{2'} + (1-\alpha+\alpha p)P_2 \\ \Rightarrow (1-\alpha)P_{2'} &= P_{1'} - (1-\alpha+\alpha p)P_2 \end{aligned}$$

$2'$-state:

$S_{1'2'} : 1'^{(t-1)} \to 2'^{(t)}, S_{3'2'} : 3'^{(t-1)} \to 2'^{(t)}, S_{32'} : 3^{(t-1)} \to 2'^{(t)}$

$$\begin{aligned} P_{2'} &= \alpha P_{1'} + \gamma\beta P_{3'} + (1-\gamma)\beta P_{3'} + (1-\alpha+\alpha p)P_3 \\ \Rightarrow (1-\alpha)P_{3'} &= P_{2'} - \alpha P_{1'} - (1-\alpha+\alpha p)\alpha^3(1-p)^3 P_0 \end{aligned}$$

k'-state:

$S_{(k-1)'k'} : (k-1)'^{(t-1)} \to k'^{(t)}, S_{(k+1)'k'} : (k+1)'^{(t-1)} \to k'^{(t)}$,
$S_{(k+1)k'} : (k+1)^{(t-1)} \to k'^{(t)}$

$$P_{k'} = \alpha P_{(k-1)'} + \gamma\beta P_{(k+1)'} + (1-\gamma)\beta P_{(k+1)'} + (1-\alpha+\alpha p)P_{(k+1)}$$

Thus we obtained the following equations:

$$P_{0'} = \alpha(1-p)P_0 \tag{4.1}$$

$$P_{0'} = (1-\alpha+\alpha p)P_1 + (1-\alpha)P_{1'} \tag{4.2}$$

$$\forall k \geq 1, P_k = \alpha^k(1-p)^k P_0 \tag{4.3}$$

$$(1-\alpha)P_{2'} = P_{1'} - (1-\alpha+\alpha p)P_2 \tag{4.4}$$

From 4.2, we get

$$P_{0'} = (1-\alpha+\alpha p)P_1 + (1-\alpha)P_{1'}$$

$$\Rightarrow (1-\alpha)P_{1'} = \alpha^2(1-p)^2 P_0 \tag{4.5}$$

Now we have,

$$P_{2'} = \alpha P_{1'} + \gamma\beta P_{3'} + (1-\gamma)\beta P_{3'} + (1-\alpha+\alpha p)P_3$$

$$(1-\alpha)P_{3'} = P_{2'} - \alpha P_{1'} - (1-\alpha+\alpha p)\alpha^3(1-p)^3 P_0 \tag{4.6}$$

From 4.4 and 4.5, we get

$$(1-\alpha)P_{2'} = \frac{\alpha^2(1-p)^2}{(1-\alpha)}P_0 - (1-\alpha+\alpha p)\alpha^2(1-p)^2 P_0$$

$$\Rightarrow P_{2'} = \frac{\alpha^3(1-p)^3}{1-\alpha}(2-\alpha-p+\alpha p)P_0 \tag{4.7}$$

From 4.6 ,

$$(1-\alpha)P_{3'} = P_{2'} - \alpha P_{1'} - (1-\alpha+\alpha p)\alpha^3(1-p)^3 P_0$$

$$\Rightarrow (1-\alpha)P_{3'} = \left[\frac{\alpha^3(1-p)^2}{(1-\alpha)^2} - \frac{\alpha^3(1-p)^2}{1-\alpha} - (1-\alpha+\alpha p)\alpha^3(1-p)^3\right] P_0 \tag{4.8}$$

Now we get,

$$P_{3'} \;=\; \alpha P_{2'} + \gamma\beta P_{4'} + (1-\gamma)\beta P_{4'} + (1-\alpha+\alpha p)P_4$$

$$\Rightarrow (1-\alpha)P_{4'} = P_{3'} - \alpha P_{2'} - (1-\alpha+\alpha p)P_4 \tag{4.9}$$

Similarly we have,

$$(1-\alpha)P_{5'} = P_{4'} - \alpha P_{3'} - (1-\alpha+\alpha p)P_5 \tag{4.10}$$

$$(1-\alpha)P_{6'} = P_{5'} - \alpha P_{4'} - (1-\alpha+\alpha p)P_6 \tag{4.11}$$

$$\sum_{k=0}^{\infty} P_k + \sum_{k'=0}^{\infty} P_{k'} = 1 \tag{4.12}$$

In equation 4.12, put all the values of P_i from above equations, we obtain the value of P_0. Using this value of P_0, we will get all others values from the above equations.

4.2 Reward Calculation:

Here we calculate the reward for our second attack by using the transition probabilities. Let R_{hm2} and R_{rm2} be the revenue of the honest mining strategy and our second rational mining strategy respectively.

$$R_{hm2} = P_{0'}\gamma(1-\alpha) + P_{0'}(1-\gamma)(1-\alpha).2 + P_0(1-\alpha) \tag{4.13}$$

$$R_{rm2} = P_{0'}\alpha(1-p).2 + P_0\gamma(\beta+\alpha p) + P_2\beta.2 + \alpha.pPr[i>0] + Pr[i>2](1-\alpha+\alpha p) \\ +P_0\alpha p + Pr[i'>0]\gamma(1-\alpha) \tag{4.14}$$

4.3 Graph Representation

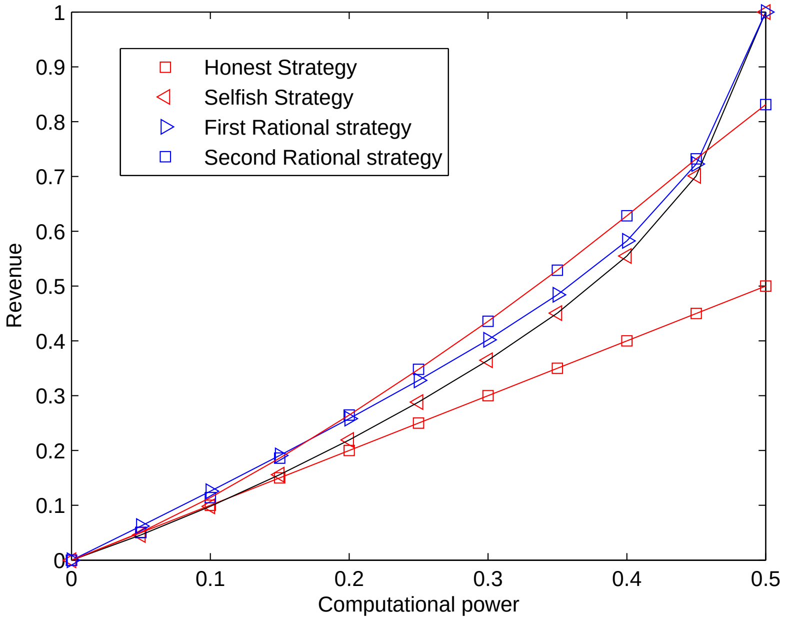

Figure 4.2: Comparison between second rational mining strategy and other mining strategies

In 4.2, we draw the the revenue graph with respect to computational power α. In this graph, we compare our second rational mining strategy with honest mining strategy, selfish mining strategy and our first rational mining strategy. For $\gamma = 0.85$ in our first rational mining strategy, we saw that a miner with computational power $\leq 44\%$ can get more revenue than selfish miner[8] if it chooses the honest mining strategy with probability 30%. In our second rational mining strategy, a miner with computational power $19\% - 45\%$ will get more revenue than our first rational miner if it chooses the honest mining strategy with probability 12%.

Chapter 5

Conclusion and Future Work

In this thesis we have shown selfish mining and stubborn mining strategies are not always the best strategies for a miner who wants to maximize its gain. We have also presented that by carefully choosing the parameters α and p, a rational miner can maximize its gain over selfish and stubborn miners. In our first rational mining strategy, we discussed that if a miner controls 28% of computational resources it can attack the Bitcoin system. This was a stronger result than selfish mining strategy which showed that a miner needs to control 33% of resources to launch a full attack on Bitcoin. In our second attack scheme, we proposed *second rational mining strategy* which is better than any other strategies for different choice of parameters. In "comparative study of Mining strategies"3.4 we have shown this result using graphs.

In our first (second) rational mining strategy, we collude honest mining strategy with selfish (stubborn) mining strategy, i.e, a rational miner choose honest mining strategy with probability p and choose selfish (stubborn) mining strategy with probability $(1 - p)$. In future we collude this three mining strategies (honest, selfish and stubborn), i.e, a miner choose honest mining strategy with probability p, selfish mining strategy with probability q and stubborn mining strategy $(1 - p - q)$.

Bibliography

[1] Samiran Bag, Sushmita Ruj, and Kouichi Sakurai. Bitcoin block withholding attack: Analysis and mitigation. *IEEE Trans. Information Forensics and Security*, 12(8):1967–1978, 2017.

[2] Lear Bahack. Theoretical bitcoin attacks with less than half of the computational power (draft). *CoRR*, abs/1312.7013, 2013.

[3] Tobias Bamert, Christian Decker, Lennart Elsen, Roger Wattenhofer, and Samuel Welten. Have a snack, pay with bitcoins. In *13th IEEE International Conference on Peer-to-Peer Computing, IEEE P2P 2013, Trento, Italy, September 9-11, 2013, Proceedings*, pages 1–5, 2013.

[4] Mauro Conti, Sandeep Kumar E, Chhagan Lal, and Sushmita Ruj. A survey on security and privacy issues of bitcoin. *CoRR*, abs/1706.00916, 2017.

[5] Nicolas T. Courtois and Lear Bahack. On subversive miner strategies and block withholding attack in bitcoin digital currency. *CoRR*, abs/1402.1718, 2014.

[6] Nicolas T. Courtois, Marek Grajek, and Rahul Naik. The unreasonable fundamental incertitudes behind bitcoin mining." arxiv preprint arxiv:1310.7935, 2013.

[7] John R. Douceur. The sybil attack. In *Peer-to-Peer Systems, First International Workshop, IPTPS 2002, Cambridge, MA, USA, March 7-8, 2002, Revised Papers*, pages 251–260, 2002.

[8] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography and Data Security - 18th International*

Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers, pages 436–454, 2014.

[9] Ethan Heilman. One weird trick to stop selfish miners: Fresh bitcoins, A solution for the honest miner. *IACR Cryptology ePrint Archive*, 2014:7, 2014.

[10] 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, Washington, D.C., USA, August 12-14, 2015.*, pages 129–144, 2015.

[11] Ghassan Karame, Elli Androulaki, and Srdjan Capkun. Double-spending fast payments in bitcoin. In *the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012*, pages 906–917, 2012.

[12] Ghassan Karame, Elli Androulaki, and Srdjan Capkun. Two bitcoins at the price of one? double-spending attacks on fast payments in bitcoin. *IACR Cryptology ePrint Archive*, 2012:248, 2012.

[13] Daniel Kraft. Difficulty control for blockchain-based consensus systems. *Peer-to-Peer Networking and Applications*, 9(2):397–413, 2016.

[14] Joshua A Kroll, Ian C Davey, and Edward W Felten. The economics of bitcoin mining, or bitcoin in the presence of adversaries.

[15] Loi Luu, Ratul Saha, Inian Parameshwaran, Prateek Saxena, and Aquinas Hobor. On power splitting games in distributed computation: The case of bitcoin pooled mining. In *IEEE 28th Computer Security Foundations Symposium, CSF 2015, Verona, Italy, 13-17 July, 2015*, pages 397–411, 2015.

[16] Ralph C. Merkle. A digital signature based on a conventional encryption function. In *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings*, pages 369–378, 1987.

[17] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, http://bitcoin.org/bitcoin.pdf.

[18] Arvind Narayanan, Joseph Bonneau, Edward W. Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and Cryptocurrency Technologies - A Comprehensive Introduction.* Princeton University Press, 2016.

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

[20] Michael O. Rabin. Transaction protection by beacons. *J. Comput. Syst. Sci.*, 27(2):256–267, 1983.

[21] Meni Rosenfeld. Analysis of bitcoin pooled mining reward systems. *CoRR*, abs/1112.4980, 2011.

[22] Ren Zhang and Bart Preneel. Publish or perish: A backward-compatible defense against selfish mining in bitcoin. In *Topics in Cryptology - CT-RSA 2017 - The Cryptographers' Track at the RSA Conference 2017, San Francisco, CA, USA, February 14-17, 2017, Proceedings*, pages 277–292, 2017.