University of Birmingham

Gurpreet Singh Mannan
(642512)

Security of Blockchain: An investigation and analysis of mining attacks on Bitcoin

SECURITY OF BLOCKCHAIN: AN INVESTIGATION AND ANALYSIS OF MINING ATTACKS ON BITCOIN

By

GURPREET SINGH MANNAN
(642512)

A masters thesis submitted to
The University of Birmingham
for the degree of
MSc in Computer Science

Supervisor: Dr. David Galindo

School of Computer Science
College of Engineering and Physical Sciences
The University of Birmingham
September 2017

Abstract

"Bitcoin was created to serve a highly political intent, a free and uncensored network where all can participate with **equal** *access.*[1]*"*

Amir Taaki (British-Iranian revolutionary, hacker, and programmer)

Bitcoin protocol dictates that the "*system is secure as long as honest nodes collectively control more CPU power than any… attacker nodes*". However, Carlesten et al [3] show that miners turn from honest to rational behaviour, when the transaction fee is the majority avenue of profit. In addition to this, mining attacks or strategies against Bitcoin such as selfish mining by Eyal and Sirer [1] and stubborn mining by Nayak et al [2] show that miners can earn disproportionate rewards by deviating from the default mining strategy employed by Bitcoin. If an opportunity is available to gain greater profit than the computational power a miner controls, more and more *honest* miners will join the *dishonest* pool and this can grow to become the majority, which then has implications for the Bitcoin network. In this thesis, the author focuses on two key areas. The first area of focus is investigating the behaviour of miners using [3] and their simulator to show that the shift from honest to rational behaviour, where a miner decides on an action based on whether it will maximise their profit, is also prevalent when the block reward is the majority avenue of profit. The author shows that regardless of the block reward, a miner will always cheat if they are given the chance and their method of attack is to undercut a block to gain more revenue. The second area of focus is to investigate a hybrid strategy in comparison to other strategies as suggested in [2]. The author shows that a "*hybrid stubborn strategy*" outperforms honest, selfish and stubborn mining strategies given certain parameterisations in terms of disproportionate revenue gained from blocks mined.

[1] K. Spotz. *Bitcoin's "Political Neutrality is a Myth" - Amir Taaki Interview.* [online] Cointelegraph. Available at: https://cointelegraph.com/news/bitcoins-political-neutrality-is-a-myth-amir-taaki-interview. 2014. [Accessed 8 Jul. 2017].

Acknowledgements

I would firstly like to acknowledge my supervisor Dr. David Galindo. His passion and dedication to being a supervisor and the project has not wavered. He is a source of inspiration and has been instrumental in ensuring this project reaches fruition. I wholeheartedly thank him for everything he has done for me during the last few months and beyond. I would also like to thank Dr. Harry Kalodner from Princeton University for his invaluable feedback regarding the simulator, which came as a relief towards the start of the project. I would like to thank Dr. Kartik Nayak for supplying me with his simulator to confirm my own results and being a solid foundation of this thesis at a time of need. I would like to thank my friends in Computer Science for keeping me sane through the good times and the bad and my family for putting up with me for the last year.

CONTENTS

1 INTRODUCTION 7

2 TECHNICAL DETAILS 8

2.1 Peer to Peer 8

2.2 Hash Functions 9

2.3 Hash Pointers 10

2.4 Digital Signatures 11

2.5 Transactions 11

2.6 Blocks 13

2.7 Mining 14

3 LITERATURE REVIEW 16

3.1 Incentives and Rewards 16

3.2 Honesty 18

3.3 Attacks 18

3.4 Mining Attacks 20

4 MODEL AND SIMULATIONS 23

5 STRATEGIES 28

6 DISCUSSION AND ANALYSIS 34

6.1 Finding 1: Default miners will always cheat, given the chance 34

6.2 Finding 2: Undercutting is preferred 35

6.3 Finding 3: Selfish Mining has the capability to cause damage 37

6.4 Finding 4: Honest mining still has a place for low computational power and γ values 38

6.5 Finding 5: Lead stubborn strategies dominate the middle ground in terms of Ω and γ 39

6.6 Finding 6: Hybrid Stubborn has a place in the strategy space 41

7 IMPLICATIONS FOR BITCOIN 41

8 CONCLUSION 44

9 REFERENCES 46

10 CODE 48

11 BIBLIOGRAPHY 49

12 APPENDICES 50

12.1 Appendix 1: 50

12.2 Appendix 2: 52

12.3 Appendix 3 53

LIST OF FIGURES

Figure 1 : A round of SHA256 using 8 blocks 9
Figure 2 : How a blockchain using hash pointers is created [5] 10
Figure 3 : Bitcoin transactions stored in a Merkle Tree [5] 10
Figure 4 : Transactions in Bitcoin [6] 11
Figure 5 : A transaction script in Bitcoin [8] 13
Figure 6 : Structure of a Bitcoin Block [9] 14
Figure 7 : Taxonomy of Attacks on Bitcoin from Conti et al [16] 19
Figure 8 : File output and text output of the simulator 25
Figure 9 : Hybrid Stubborn Code from the simulator as shown in Appendix 2 26
Figure 10 : how the output of the simulator is constructed 27
Figure 11 : how the output of the simulator is shown 27
Figure 12 : Key terms for formalising each strategy 28
Figure 13 : How mining behaviour changes given the change in block reward 35
Figure 14 : Average adoption of strategies across all block rewards 36
Figure 15 : Distribution of computational power and mining pools [28] 37
Figure 16 : Selfish Mining profits based on computational power 38
Figure 17 : Average relative gains (in terms of percentage) of honest strategies over other strategies 39
Figure 18 : Percentage gains in terms of relative gains and profitability when employing a lead stubborn strategy 40
Figure 19 : Average relative gains (in terms of percentage) of hybrid stubborn strategies over other strategies 41
Figure 20 : Percentage gains in terms of relative gains and profitability when employing a hybrid stubborn strategy 41
Figure 21 : Percentage gains in terms of relative gains and profitability when employing a selfish strategy 42
Figure 22 : Percentage gains in terms of relative gains and profitability when employing a trail stubborn strategy 42

1 Introduction

Bitcoin is a cryptocurrency which has fast become one of the most popular cryptocurrencies with a rich and diverse set of users and backers. Bitcoin has had a turbulent history, being synonymous with political revolt, cybercrime and the Dark Web. Bitcoin's niche is the decentralized nature of it with the control and enforcement belonging to the users of the currency and not a central entity. Allowing its users to spend anonymously, the cryptocurrency is created by mining which involves huge amounts of computational power to compute and check transactions within the network. At the heart of Bitcoin is the blockchain which is a data structure unlike no other which is adopted as a public ledger which records all Bitcoin transactions attempted and completed. If a miner is successful in solving this puzzle, they can record their set of transactions on the blockchain. This set of transactions is recorded in a block and upon confirming that the block is valid, the miner collects a block reward as well as the transaction fees present within the block.

Based on the review of literature and previous work in the area, the author has therefore decided to focus on two areas within the research. Carlesten et al [3] and Carlesten [4] have shown that miners begin to act rationally when transaction fees become the main source of income when the block reward runs out. The author has decided to investigate whether this rational behaviour is also present if miners were given the opportunity to act dishonestly where the block reward accounts for the most reward. The author believes that these areas are important to explore as it seeks to undermine the core fabric of the Bitcoin network and incentivising honest behaviour. Similarly, Nayak et al [2] introduce a new set of strategies to the strategy space which are called stubborn strategies and they show that a stubborn strategy can outperform selfish and honest mining using certain parameterisations. The author has decided to investigate the hybrid strategies which [2] comment on as further work by implementing a hybrid strategy using strategies from [2]. The author believes this area of exploration is important as it will show whether another strategy which seeks to gain disproportionate revenue in comparison to the computational power the miner has. The exploration is firstly, is such a strategy possible and secondly, can it gain as much disproportionality as its predecessors in [2] and [3]. The focus in this thesis will therefore be Bitcoin mining as a source of attacks against Bitcoin. The author is aware of other substantial theoretical attacks against the currency but will focus on those resulting from mining.

2 Technical Details

It is important to, firstly, explore the technical details concerning the cryptocurrency before exploring potential attacks on it. In terms of technicality, it is important to consider the cryptographic properties of Bitcoin along with the technicality of the cryptocurrency. In terms of the thesis, the author would like to state a pre-requisite. In terms of the research conducted in this thesis, a singular miner or pool of miners is differentiated where possible and if not differentiated, are interchangeable.

2.1 Peer to Peer

Bitcoin works on a peer to peer network model which provides a low barrier for entry to buy Bitcoins or mine them. Using this peer to peer network, one user sends another some funds in the form of Bitcoins which the user signs using a digital signature. This is bundled into a transaction which declares that the person to whom the payment is being made is now the owner of the coins and this is then broadcast on the Bitcoin network. The nodes which are listening on the peer to peer network then confirm the authenticity of the message by propagating the message throughout the network. A node can be a full node which has the entire copy of the blockchain or a simplified payment verification node (SPV) who has a filtered access to transactions and stores only part of the blockchain. Once a sufficient amount of confirmations have been made, which currently stand at a heuristic of 6, this transaction is confirmed by the network and added to the block of transactions that have already been verified and confirmed, called the blockchain. An honest node, upon reviewing the block, will extend the longest chain, so the more confirmations of this acceptance the block gets, the more likely it is that the block will be present on the Bitcoin blockchain Transactions carried out within Bitcoin are known to all nodes and users in the network which are solely responsible for ensuring the reliability and truthfulness of the transactions. This agreement of the status of the blockchain is generally referred to as the consensus protocol.

The consensus protocol is difficult as the authenticity of nodes can come into question and as a peer to peer network is used, it suffers from the same issues that distributed systems have in the form of faults or latency. Bitcoin negates this issue by providing incentives to nodes to act in an honest manner. To gain consensus, new transactions are broadcast to all nodes within the Bitcoin network. Transactions which are new to a block are assembled in a pool and at a given point, a node is selected based on its computational power which broadcasts its block to the other nodes. The validity of the block will be ascertained, as to, whether the coins which are being spent are available to the sender and that the signature which verifies the transaction is valid. Selecting nodes based on computational power allows the nodes to compete for creating a block and provides a robust method with which to select a node. The computational power is determined by a *proof of work* in the form of solving a puzzle to produce a block. The proof of work

helps achieve stability and reliability within the blockchain. After this, a miner will compete with other to add the block containing the valid transactions which have not yet been added to the blockchain. The miner does this by solving computationally difficult hash puzzles and upon their success, the miner broadcasts the new block they have mined to the network where the nodes accept the block by including the hash of the block in the next block which is created in the blockchain, hence adding the new block to all the nodes copy of the blockchain and the process repeats.

2.2 Hash Functions

Using the peer to peer network, Bitcoins are created where 'miners' provide proof of work through the completion of hash puzzles. Mining is utilised for confirming transactions and adding transactions to the blockchain once enough of the nodes have confirmed that the transaction is true. Mining is often difficult and defined by protocols set out by the Bitcoin framework. Bitcoins are finite and often, a lot is needed for a node to solve this hash puzzle to produce a block or effectively mine a Bitcoin. Such proof of work uses a hash function as part of its algorithm. A hash function is where we take a string of inputs and create a fixed output. In terms of Bitcoin, the fixed output is 256 bits. Hash functions have certain properties which allow them to be at the heart of key cryptographic functions for encryption. Firstly, the hash function must be collision free where you cannot find two different values which hash to the same output. It must be easy to compute the hash but infeasible to invert the hash to recover the original input unless some key information is available and it must be puzzle friendly. This means that given any output, there is no way to express the hash without carrying out computations to find the hash and it could be any value. For its hash function, Bitcoin uses SHA256, as shown in Figure 1 which takes an input and creates blocks which are 512 bits in size. SHA256 works by starting with an initialisation vector which is 256 bits and this gets compressed by the compression function along with a 512 bit block and creates a fixed output of 256 bits which is then fed into the next compression function and so on. So long as the compression function which underlies SHA256 is collision free, so will SHA256. In fact, in Bitcoin, the hash function is applied twice.

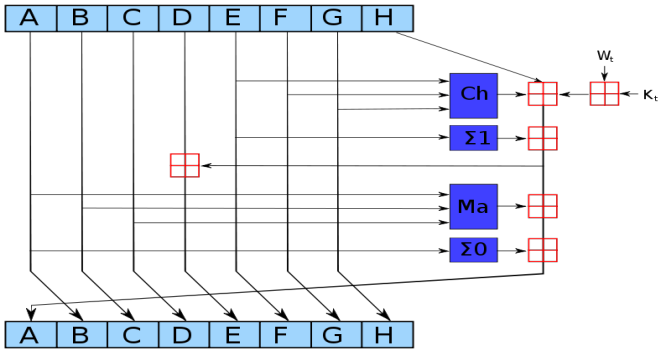

Figure 1 : A round of SHA256 using 8 blocks

2.3 Hash Pointers

Hash pointers are key to the success of the blockchain and the security of it. A hash pointer is used as a pointer in the data structure to where previous data or in the case of Bitcoin, where transactions are stored and provide a cryptographic hash of the data up until that point as shown in Figure 2. A hash pointer can perform actions like a pointer in a data structure with the added bonus of being able to identify if a change has been made to the data.

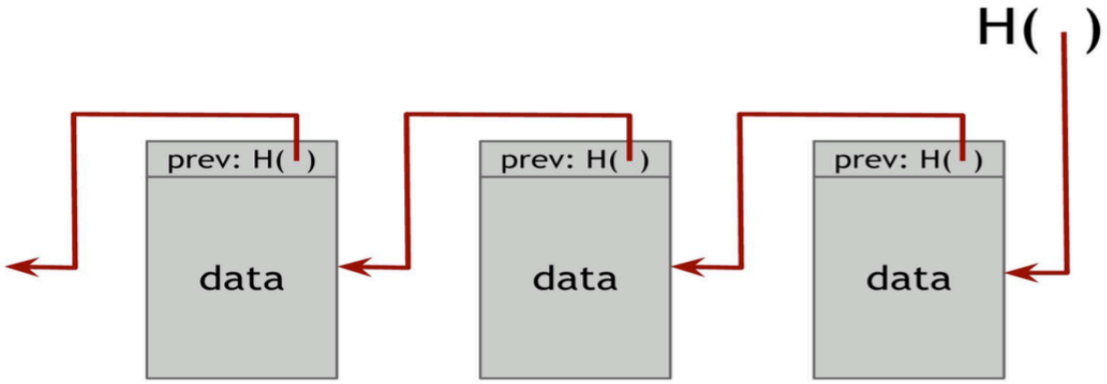

Figure 2 : How a blockchain using hash pointers is created [5]

The advantage of having such a structure is that if an adversary were to interfere with data in any of the blocks, they would also need to tamper with the hash pointers of that block which, in turn, would mean that the hash pointers for the previous block would need to be changed all the way through to the head of the structure, called the genesis block, which cannot be changed. Each block in this blockchain will have a header which contains metadata including a hash pointer of the previous block header which enables the linking element of the blockchain as well as the Merkle root node which summarises the new transactions which are included in the block which then can be utilised to track all Bitcoin transactions. The Merkle root node provides a mechanism to record all transactions in the form of a Merkle Tree as shown in Figure 3. Following the header, the block will contain a list of transactions which have been logged by the block.

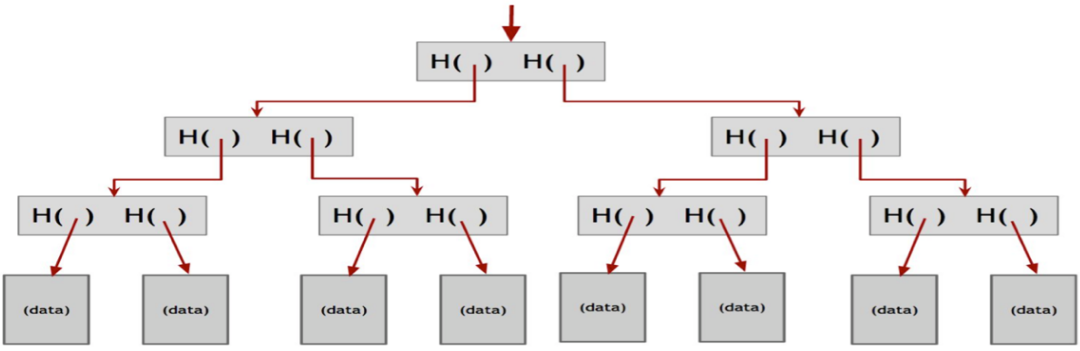

Figure 3 : Bitcoin transactions stored in a Merkle Tree [5]

The Merkle construction allows users to traverse the tree identifying as interference with the blocks as well as, in Bitcoin's case, to ascertain membership as to whether a transaction was indeed made.

2.4 Digital Signatures

Bitcoin utilises digital signatures to verify transactions and the creation of new coins within the network. Using the digital signature, only the sender of the transaction can sign the transaction but anybody viewing the transaction in the blockchain can verify that the transaction was indeed signed by the sender. One of the key security tests around signatures is whether signatures are forgeable or not. The unforgeable nature of the signature means that whatever strategy the adversary applies, there is only a negligible chance that they can forge a signature. In Bitcoin, the public key of a sender is equated to a digital signature and this identifies the individual who is making the transaction. All the sender must do is create a random key pairing where the public key is the identification of the sender which would normally be hashed and a secret key which can be used to confirm that the sender is who they say they are. The sender will then sign the transaction by using their private key to encrypt it. Others on the network can verify that the transaction did indeed come from the sender by using the sender's public key to decrypt the transaction. Bitcoin utilises Elliptical Curve Digital Signature Algorithm (ECDSA) for its standard of signatures which provides a source of good randomness and provides security.

2.5 Transactions

Each block within the blockchain contains several transactions. These are the methods in which payments are made within the Bitcoin network. The transaction transfers a certain number of Bitcoins from one account holder to another as demonstrated in Figure 4 below.

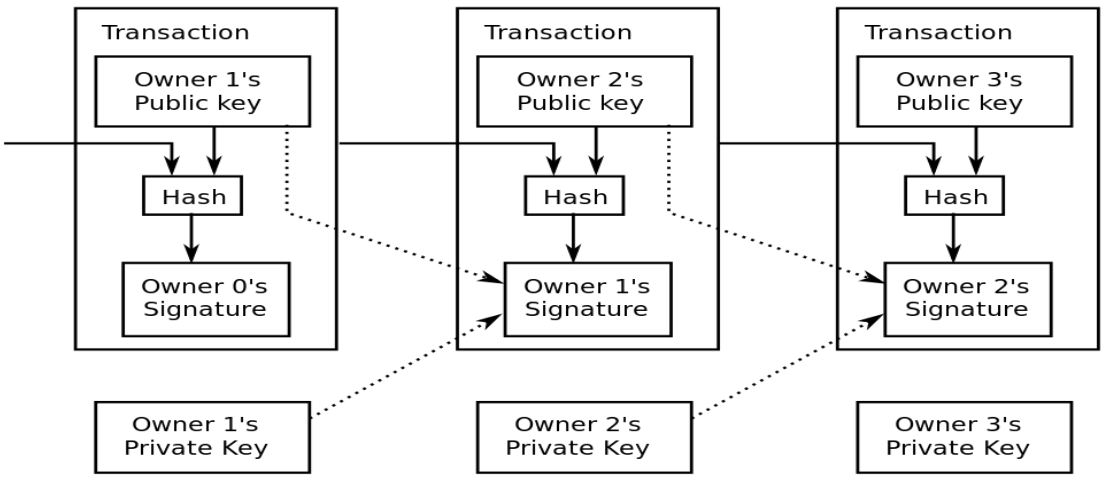

Figure 4 : Transactions in Bitcoin [6]

Each account holder is referred to by their Bitcoin address which is a 27-34 character alphanumeric value. An account holder is not limited to the number of accounts they can hold and normally, these accounts are managed through wallet software, providing the client side of the Bitcoin network. Transactions are composed of inputs and outputs. The inputs refer to the Bitcoins that are being consumed by the transaction in order to be spent. The Bitcoin used as input is consumed fully in that the whole Bitcoin amount is the input of the transaction. The input total must be greater than the outputs with the difference being processed as fees unless specified otherwise. The output will be the coins which are created by the transaction. If new Bitcoins are minted, no such input is needed as a special coinbase transaction is used. The transaction itself is formed with the hash of the previous transaction where the Bitcoin was last used along with the public key of the recipient of the Bitcoin [7]. Using this model, joint payments are also allowed where the two different senders of the payment are two inputs but both need to be independently signed for the transaction to be valid [6]. Once this transaction is made, it is digitally signed to authenticate the Bitcoin. This signature is checked using the same process of looking at past transactions. Now, if the total amount of the Bitcoin being referred to is not being sent to the recipient, a second output is the leftover amount of Bitcoin where the recipient is the sender. This provides a past transaction which can be used in subsequent transaction to prove ownership of the Bitcoin. Alternatively, some of the Bitcoin may be left as a transaction fee for the miner as an incentive to include the transaction in their block.

Once everything has been verified, the transaction is propagated throughout the network and each node adds the transaction to its memory as an unconfirmed transaction to the pool of transactions waiting to be verified. This allows the network to achieve consensus on the transaction and the order in which were made, so that the correct action can be taken. The transaction waits for a miner to include it within a block (after confirming that the transaction is legitimate). Each miner has a block of transactions and the miner gets to work to solve the computational puzzle to get the block included on the public blockchain. In certain circumstances, transactions may be left out because the transaction may be minimal (other sender's might have higher fees hence more revenue for the miner) or the block size for mining is limited, which we will explore later in the thesis. Once the block of transactions makes it to the public blockchain, any subsequent blocks added adds more confirmations that the transaction cannot be overturned.
The test of ownership of transactions is done through the examination of the script which shows where the coins being spent originated from. This helps prevent double spending attacks within Bitcoin. This script, as shown in Figure 5 contains metadata, inputs and outputs. The metadata contains data around the size of the transaction and the number of inputs and outputs. The hash of the transaction provides a method of uniquely identifying the transaction which then can be grouped together using hash pointers. The inputs section of the script is an array where each input is an element of the array. The

input specifies the previous transaction through a hash value as well as the index value of the previous transaction's outputs along with a signature to prove ownership. Finally, the output section of the script is another array where each output has a value and public key to whom the transaction is being paid too.

Figure 5 : A transaction script in Bitcoin [8]

2.6 Blocks

Each block contains a number of predefined points of data as shown in Figure 6. The block will first contain a block header which contains metadata about the block. This is followed by the hash pointer highlighting the previous block in the blockchain which provides proof of ownership. Next, the block contains a Merkle Tree which contains a tree of all the transactions within the block. This provides a method in which a check can be performed to see if the transaction is part of the block. The tree itself has a logarithmic length in the number of transactions in the block. Following that, the block provides data to help with the mining process such as the timestamp of the block and the nonce value which is incremented each time to allow for a proof of work solution to be achieved. Finally, the transactions and the number of them are recorded along with any coinbase transactions which are used to create new Bitcoins. This is attributed to the block reward along with transaction fees included as an incentive for the miner. Blocks were 1MB in size which means that the block could contain transactions to the limit but any blocks over the size would be discarded. This has been an area of controversy in terms of scalability for the currency as increasing the block size allows for more transactions to be validated, which due to the popularity and adoption of Bitcoin, would increase its capacity to handle transactions from approximately seven transactions per second to much higher levels to compete with mainstream payment services such as Visa. Also, more transaction fees could be made available for miners as more transactions within the block could be included. Conversely, the increase could lead to centralisation of the cryptocurrency owing to the increased hardware requirements that would be

needed to cope with the increase as well as putting undue pressure on nodes which could lead to a reduction in the number of nodes who opt out. However, the decision falls on the nodes of the network to decide what to do as they control the network. However, at the time of writing this thesis, a hard fork (a permanent divergence from default protocols) has occurred where two competing factions have developed separate methods to increase the block size. Segwit2x's method includes moving transactions outside each block where they are parallel to the block which allows a doubling of blocks and more transactions possible. Bitcoin Cash's method is to increase the size of each block to 8MB. At the time of writing, the implication of this hard fork is unknown but this highlights the importance of block size and it will also show how readily users of the cryptocurrency are to adopt one fork over the other which will also indicate the future of the cryptocurrency.

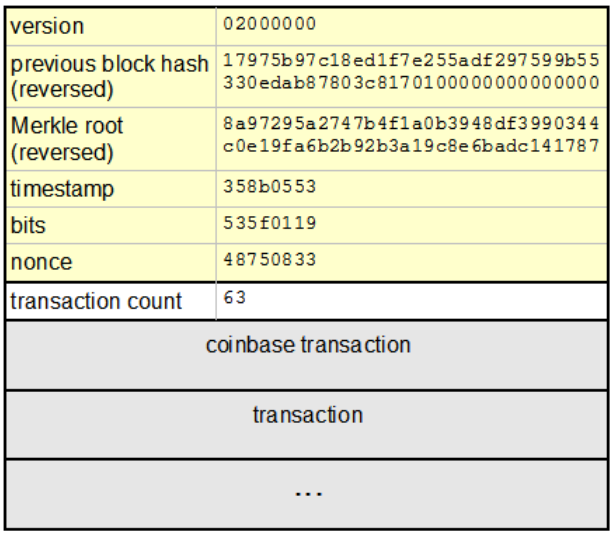

Figure 6 : Structure of a Bitcoin Block [9]

2.7 Mining

Miners in Bitcoin are utilised to store the blockchain, broadcast new blocks of transactions to the network, act as validators for transactions and provide consensus for the blockchain. Mining allows the network to produce or mint new Bitcoins to the system as a miner will receive a block reward when they discover a block which is then added to the public blockchain. A secondary reward is in the form of transaction fees which is normally the difference between the inputs and outputs of the transactions contained in a block but can also be an additional fee for the miner to include the transaction within their block. The predominate reason for these rewards is to motivate individual miners to act in an honest way and provide security. As the block reward continues to half every four years, as well as more miners trying their luck, transaction fees are set to become a more important mechanism of claiming profit in the Bitcoin currency. The block reward is, at the time of writing, currently 12.5 Bitcoins (BTC). The reward is only payable if the block makes it to the long term blockchain for Bitcoin. This allows Bitcoin to operate

in a way where behaving fair and honestly, where other nodes agree with your action is fruitful. The Bitcoin supply is finite and is expected to run out in 2040. Hence, transaction fees will become more prevalent. Transaction fees work like a tip for service received and are given to a block creator and these are most likely to drive the currency forward once the block reward runs out. The fee is added by a sender to their payment as an incentive for the miner who will mine the block in which the transaction appears.

Miners compete to be the first to mine a block containing the transactions that have yet to be added to the blockchain. To carry this out, the "*miner node*" will assemble its transaction block and propagate a Merkle Tree using different values to get the correct hash value to solve the hash puzzle. This is classed as a "*proof of work*". Mining is computationally difficult in that it requires a lot of computations so that the blocks which are discovered by miners continue to be found at a consistent speed. Mining such a block is difficult as the header of the block is processed using SHA256 along with a number used once, which must fall within a target area for the block to be accepted by the Bitcoin network. This target area is such that the hash of the block must contain a certain number of trailing zeros for which the probability is low so many attempts may be made. To achieve these values, the number used once, or "nonce" value is used which is incremented and the hash calculated each time. For a block to be considered, it must include this proof that it belongs to the blockchain and hence is valid. This proof is verified by other nodes within the network when they receive a block to determine authenticity of the block. To provide further security, the difficulty is adjusted every 2016 blocks to allow generation time of blocks to be on average a block every ten minutes. As the number of miners increases, the difficulty in finding a block that provides this proof increases as well. The only way to improve this is to check more and more values as quickly as possible before another miner does. The Bitcoin protocol dictates that blocks generated by malicious miners or pools of miners that do not fall within the target area are rejected by the network and become worthless. Miners and their work, allow the nodes within the network to reach a consensus which is tamper-resistant and secure [10]. Mining requires a significant amount of processing power as the computations to carry out to create a block are quite complex. To put this into perspective, a normal desktop computer would take 2^{24} years to find a block or in other words, to solve the hash puzzle. Other customisable methods of mining hardware to allow individuals to mine have been created from utilising graphical processing unit (GPU) to creating specific Bitcoin application-specific integrated circuit (ASIC) rigs but the computational power this hardware provides is still some way off what can be considered efficiently computable. To solve this, mining pools have been created where groups of miners attempt to mine a block and the reward is shared among them based on how much of the computation leading up to the finding of the block each miner did. However, as this paper will analyse, mining is susceptible to some attacks which can destabilise the Bitcoin network.

3 Literature Review

Before exploring the security of Bitcoin, it is necessary to review the literature around Bitcoin, focusing on the development of research on mining attacks or strategies. In terms of current literature, Bitcoin is a current research focus. Although cryptocurrencies such as DigiCash and b-money have been developed over the years, they have not really seen the success of Bitcoin or alternative currencies such as Etheruem or Litecoin. Bitcoin is different from other competitor cryptocurrencies due to its peer to peer design that allows for the decentralized nature of the Bitcoin network. The author will predominately focus on the security of the Bitcoin cryptocurrency but a comparison of the currencies in terms of mining strategies is outside the scope of this thesis.

3.1 Incentives and Rewards

The Bitcoin currency was founded within [7], where the author by the pseudonym of Satoshi Nakamoto suggests that the decentralised nature of the currency is only possible by miners remaining honest and the cryptocurrency promotes this ideal by incentivising honest behaviour of nodes within the network. Bonneau et al [11] comment that, in comparison to other cryptocurrencies, Bitcoin fills an important gap in the market in that it provides a network without trusted parties and without identities of the participants of the network, something which, at the time, was lacking in competitor cryptocurrencies. The incentives discussed by Nakamoto are in two forms; one in the form of a fixed block reward and one in the form of transaction fees. Once a block has been mined, it is announced to the network and the account holders use a heuristic of six confirmations from nodes before the block is accepted on the public blockchain. Bonneau et al [11] demonstrate that this consensus protocol used within the Bitcoin network does require some form of recompense in the form of a block reward to provide an incentive to find blocks and solve the cryptographic puzzles. Without such a reward system, miners would not follow the rules and their work would not reap any rewards. Interestingly, as highlighted in [12] there is an unspoken rule for Bitcoin miners, that they approve transactions honestly and then have a higher chance of reaping rewards.

The default protocol, as dictated in [1] and [7], states that a miner's chance of winning a block are approximately equal to the proportion of the total computational power they control. To this end, a miner with 0.3 computational power has roughly 30% chance of winning the block. If the miner has a substantial amount of computational power, they have a higher likelihood of winning the block. Nakamoto recognises that the security of the Bitcoin system and network is guaranteed so long as a single adversary, miner or pool of miners are not in possession of 50% or more of the total computational power (hashpower) used within the network. A miner can mine, independently or a group called a mining pool. Mining pools work to allow miners a regular income and to reduce the uncertainty for lone miners to gain rewards. Such pools are governed by a pool manager

who will assess and give out the rewards based on the contributions and more importantly, proof of work, from the participants of the pool. The more computational power a miner has, the better their chances of finding a block to solve the cryptographic puzzle. To this end, Vasek [13] proposes that miners and mining pools are even willing to attack one another in order to maximise their own profit.

It has long been considered, as in [7] and [14], that Bitcoin is only as strong as the security that undermines it and that for computational power of 50% and less, the fundamental security of the cryptocurrency holds if miners act honestly. However, Bahack [15] states that there is a possibility of attacking the network by accumulating computational power to outrun the honest mining majority of the network but this would become increasingly difficult the longer the blockchain become and the network grows. This magic figure of 50% is needed as any miner who has above this threshold could control the cryptocurrency. [15] highlights that this has serious implications for the security of Bitcoin where 40% of the network could collude and seek to block transactions and cause Bitcoin to be bought into disrepute. Eyal and Sirer [1] provide evidence that employing specific mining strategies brings disproportionate rewards to the miners. In fact, as soon as the alpha fraction of computational power for a miner is closer to 40%, we start seeing disproportionate rewards for miners employing specific mining strategies. What this means is that if a miner manages to control at least 40% of the hash power, they can obtain revenue for half of the blocks mined which is disproportionate and contrary to [7]. This means, following disproportionate revenue for selfish miners, this encourages new miners to join the selfish miner pool and the pool growing to a majority which can breach the 50% threshold proposed by [7]. This means, as [1] suggests that the Bitcoin protocol is not incentive compatible as using the selfish mining strategy allows a miner or pool of miners can obtain disproportionate revenue than their computational power. Eyal and Sirer [1] suggests that if a set of colluding miners were to command the majority computational power of the network, not only could they bring about disproportionate rewards for the majority but then the Bitcoin network would cease to be decentralised and be under the command of the colluding miners.

Contrary to this, Courtois and Bahack [12] state that just because some subversive strategy which seeks to divert honest miners to carry out dishonest actions is possible, it does not mean that Bitcoin is weak in terms of its security. The original white paper of Bitcoin, [7], suggests that playing by the rules of the protocol garners more profits than attacking the network. However, [12], then goes onto to explain that sometimes when miners deviate from the default protocol, they do not suffer a drop in rewards and more importantly, such actions can remain invisible to the network and not detected, which again, has implications for Bitcoin.

3.2 Honesty

As discussed above, the default protocol for Bitcoin is miners acting honestly and the cryptocurrency promoting this ideal by incentivising honest behaviour of nodes within the network. Nakamoto [7] explains that incentives help keep miners honest. Nakamoto goes on to state that it would be more profitable to stick to the rules as it would be beneficial for the adversary in the long term. Attacks such as majority mining attacks would damage the currency and the exchange rate of Bitcoin and that "*playing by the rules*" is the long term profitable goal. Nakamoto also goes on to explain that as long as honest nodes have the majority of the computational power in the network, an attack is not possible. In [1], talking about deviant mining strategies, Eyal and Sirer state that 2/3 of the mining power of the Bitcoin network needs to be honest to prevent such an attack from being carried out. However, honest miners, as explained in [1], can be forced to act rationally rather than honestly in order to maximise their rewards. A rational miner refers to a miner that only deviates from the protocol if such behaviour increases their advantage in the system [32]. [32] comments that a miner can increase their advantage by receiving blocks as quickly as possible and broadcasting them as fast as possible to the network. In [11], Bonneau et al comment that one of Bitcoin's premise is that rational behaviour can be recognised and incentives aligned and changed to ensure the security of the network through the correct utilisation of consensus. This suggests that if miners began to act rationally instead of honestly, rules and protocols could be changed in order to address this issue. However, this alignment may not be so clear cut when the block reward incentive is the minority award and the transaction fees play a much more prominent role. Carlesten [4] suggests that once incentives shift to mostly transaction fees, a realignment will be needed to maintain security. He further goes on to suggest that when this shift occurs, the number of uneven blocks in terms of transaction fees will allow rational miners a bigger incentive to fork blockchains in the hope of maximise their return. Carlesten finds that transaction fees being the dominant form of income bring with it different incentives as the value of mining blocks is small due to unclaimed transaction fees remaining in the blockchain. He suggests this will lead to deviant behaviour where miners try to fork the current head of the blockchain and encourage miners to side with them by leaving some of the transaction fees for the next miner. When the block reward diminishes, Carlesten et al [3] suggest that there will be a mining gap present when transaction fees take over as there will be an instance when immediately after a block is discovered, there will be zero expected reward from it but costs such as mining and electricity will make it unprofitable.

3.3 Attacks

The literature has become more prominent with potential attacks which can be launched against Bitcoin as researchers begin to understand the design and premise of the cryptocurrency. Figure 7 from [16], demonstrates, a taxonomy of attacks that are present or

have been identified by researchers. One such attack which has validity in the thesis and its coverage is the mining cartel attack [12] where most miners in a pool decide to ignore transactions originating from a certain group of miners who are not in the group which allows the pool to achieve higher gains. This is a possible result of employing specific strategies within Bitcoin acting dishonestly. As such, the prominent focus of the thesis is to look at mining attacks and strategies which could be employed but the author does note that along with adopting these strategies, the fact that these exist could lead to further aggressive attacks such as the ones listed in Figure 7.

Attack	Description	Primary targets	Adverse effects	Possible countermeasures
Double spending or Race attack [27]	spent the same coins in multiple transactions, send two conflicting transactions in rapid succession	sellers or merchants	sellers lose their products, drive away the honest users from network, creates blockchain forks	inserting observers in the network [27], communicating double spending alerts among peers [27], nearby peers should notify the merchant about an ongoing double spend as soon as possible [28], merchants should disable the incoming connections [29] [30]
Finney attack [31]	dishonest miner broadcasts a pre-mined block for the purpose of double spending as soon as it receives product from a merchant	sellers or merchants	facilitates double spending	merchants should wait for multi-confirmation messages for a transaction
Brute force attack [32]	privately mining a long blockchain fork to perform double spending	sellers or merchants	facilitates double spending and creates large size blockchain forks	inserting observers in the network [27], notify the merchant about an ongoing double spend as soon as possible [29]
Vector 76 or one-confirmation attack [33]	combination of the double spending and the finney attack	Bitcoin exchange services	facilitates double spending with larger number of coins	merchants should wait for multi-confirmation messages for a transaction
$> 50\%$ *hashpower or Goldfinger* [34]	adversary controls more than $> 50\%$ of computational power in the Bitcoin network	Bitcoin network, miners, Bitcoin exchange centers, and users	drive away the miners working alone or within small mining pools, weakens the effectiveness of consensus protocol, DoS	inserting observers in the network [27], communicating double spending alerts among peers [27], disincentivize large mining pools [35] [36], TwinsCoin [37], PieceWork [38]
Block discarding [39] [30] or Selfish mining [40]	miner (or mining pool) withhold the processed block(s) in order to earn inappropriate incentives	honest miners (or mining pools)	introduce race conditions by forking, waste the computational power of honest miners, with $> 50\%$ it leads to Goldfinger attack	ZeroBlock technique [41] [42], timestamp based techniques such as freshness preferred [43], DECOR+ protocol [44]
Block withholding [21] [45]	pool member withholds an already mined block	honest miners (or mining pools)	waste resources of fellow miners and decreases the pool revenue	include only known and trusted miners in pool, dissolve and close a pool as soon as the revenue drops from expected [39], cryptographic commitment schemes [45]
Bribery attacks [46]	adversary pay money to miners to mine on her behalf	miners and merchants	increases the success probability of carrying out a double spending or block withholding attack	increase the rewards for honest miners, communicate the miners that bribery might cause the long-term losses to the miners (including the dishonest miner) [46]
Refund attacks [47]	adversary exploits the refund policies of existing payment processors	sellers or merchants, users	merchant losses money while honest users might lose their reputation	use publicly verifiable evidence [47]
Punitive and Feather forking [48] [49]	dishonest miners want to blacklist transactions from a specific address	users	freeze the money held by Bitcoin users for forever	remains an open challenge
Transaction malleability [50] [51]	adversary can change the TXID without invalidating the transaction	Bitcoin exchange centers	Bitcoin exchange losses fund due to increase in double deposit or double withdrawal instances	use multiple metrics for transaction verification along with TXID [52], malleability-resilient "refund" transaction [50]
Wallet theft [13]	adversary stole or destroy private key of users	individual users or businesses	all the money in the wallet is lost	use of threshold signatures to achieve two-factor security [53] [54], use of hardware wallets [55], TrustZone-backed Bitcoin wallet [56]
Time jacking [57]	adversary speed-up the majority of miner's clock	miners	isolate a miner and waste its computational resources, influence the mining difficulty calculation process	use constraints tolerance ranges [57], network time protocol (NTP) or time sampling on the values received from trusted peers [58]
Sybil [15]	adversary creates multiple virtual identities in the network	Bitcoin network, miners, users	facilitates time jacking, DoS, and double spending attacks, threatens user privacy	Xim (a two-party mixing protocol) [59]
DDoS [60] [61]	adversary exhaust the network resources by launching a collaborative attack	Bitcoin network, businesses, miners, and users	deny services to honest users/miners, isolate or drive away the miners	Proof-of-Activity (PoA) protocol [62], stronger authentication with fast verification signatures
Eclipse or netsplit [63]	adversary monopolizes all of the victim's incoming and outgoing connections	miners and users	inconsistent view of the network/block chain at the attacked node, enable double spends with more than one confirmation	use whitelists, disabling incoming connections [63]
Tampering [64]	delay the propagation of transactions and blocks to specific nodes	miners, users	mount DoS attacks, considerably increase its mining advantage in the network, double spend transactions	modification of the block request management system [64]
Deanonymization [65] [66]	linking IP addresses with a Bitcoin wallet or public key address	users	user privacy violation/leakage	mixing services [67], CoinJoin [68], CoinShuffle [69]

Figure 7 : Taxonomy of Attacks on Bitcoin from Conti et al [16]

3.4 Mining Attacks

Mining attacks, as reported in [1] and [17], are strategies which seek to gain disproportionate rewards compared to a miner's computational power. The reward is the revenue gained from the blocks mined. As discussed in [1], Eyal and Sirer's selfish mining is the most popular strategy. Selfish mining is where a pool of miners who are dishonest can earn a proportion of revenue greater than their computational power. Selfish mining works by withholding publishing blocks to the network but instead publishing them to the pool only. The computational power that the honest miners have is greater than that of the colluding miners so with certain probability, in the long term, the public blockchain will eventually catch up and surpass the private blockchain. In the short term, however, the private blockchain may overtake the public blockchain. The selfish mining strategy is all about making other miners mine on blocks that will be lost and in the process, in the short term, minimise their own losses and maximise gains for the colluding miners [12]. Eyal and Sirer in [18] state that when selfish mining attacks are used, a selfish mining pool will reap more revenue than honest miners. This could lead to rational miners joining the selfish mining pool to negate costly investments in equipment and maximise their revenue. Decker and Wattenhofer [19] state that currently, some pools have enough computational power to launch an attack at this moment in time and that there are currently no mechanisms in the network to break up such an attack if it were to begin. Sapirshtein [20] refers to selfish mining as a workable strategy which works by manipulating the conflict resolution as part of the default protocol which states that one blockchain is valid and only blocks on this blockchain are rewarded. The selfish miner creates a fork to try and become that longest chain as well as getting honest miners to desert their blocks and join the race for the selfish miners' block. This can lead to a growth in the miner's power which would then eventually lead to other attacks such as double spending attacks and preventing transactions from certain parties to ever grace the blockchain.

Nayak et al [2] builds on this and provides evidence that although "*selfish mining*" strategy is powerful, it is not the one stop strategy. Nayak et al [2] suggest that selfish mining is not optimal for a large strategy space. They introduce stubborn mining strategies which increase adversary revenue by up to 25% in some parameterisation. Within their paper, Nayak et al have highlighted that hybrid strategies, where a combination of strategies are used are possible and may seek to provide more disproportionate rewards than stubborn strategies on their own. [2] go on to explain that accompanied by a network level attack, revenue for the adversary can be increased further with a successful eclipse attack where nodes are isolated from their peers on a network level can result in up to 30% gains as communication is controlled in the form of incoming and outgoing communications. Of the three stubborn strategies proposed by Nayak et al [2], it is the lead stubborn strategy and the trail stubborn strategy which show dominance. Nayak et al

suggest that when a miner adopts a trail stubborn strategy, by violating the longest chain rule, we can waste even more of the public mining power and in some instance 13% more compared to other stubborn strategies. The author will be focusing on a combination of stubborn strategies to develop their "*hybrid stubborn*" strategy within the thesis.

Recently, Pachal [21] has developed a hybrid strategy which they call the "*Rational Mining*" strategy where a rational miner mines honestly sometimes and selfishly or stubborn in other times with a probability of $1 - p$ where p is the probability that the miner mines honestly. He demonstrates that for careful consideration of the alpha value and the probability, a miner can maximise their gains to that over selfish or even stubborn strategies. Pachal shows that a miner with 0.28 alpha value or greater can get more revenue which is smaller than [1]'s 0.33 assessment. He demonstrates that when given a choice to choose a strategy, the miner will choose the one, with strong probability, to maximise their gain. Finally, he shows that if a miner chooses an honest mining strategy with 30% probability, they will get more revenue. Also, when the alpha value is 44% or more, the revenue gained by rational mining is higher when the probability of p is between 14% and 30%. With these strategies, [3] suggests that such strategic attacks against Bitcoin are possible because of the "*time varying*" nature of transaction fees which allow for a "*richer set of strategic deviations*".

This is a risky affair. As Göbel [22] highlights, the miners are giving up on a high probability of certain gains if it publishes the block in favour of betting that their private branch will take the lead in the short term. If this lead is two or more blocks, the colluding pool can reveal a block in response to the honest miners mining a block and publish two blocks when the deficit is cut down to one. To this end, the colluding miners are mining on their private blockchain and the honest miners are wasting their efforts on searching for blocks that will never form part of the public blockchain. Similarly [1] indicates that although both honest and selfish mining parties waste resources, the honest contingency loses more giving the selfish mining party an advantage.
Although the research in [1] and [2] demonstrates disproportionate rewards, Göbel [22] suggests that although dishonest mining (regarding selfish and stubborn mining strategies) can outperform honest mining, it does not imply that the colluding pool integrated more of their blocks into the public blockchain than if a default mining protocol was adopted. It further goes onto say that if a colluding pool use selfish mining, the colluding pool and the honest miners are worse off in comparison to if no colluding pool was present. Less blocks from both sides made it onto the public blockchain chain when miners deviated from default protocol. Similarly, critiquing selfish mining strategies, Courtois and Bahack [12] also state that in terms of there being two groups within the selfish mining strategy, "*there is no reason to assume that if there is a subversive strategy, there will be only one group following it*". This means that there may be more than one group acting selfishly in which case, selfish mining and the findings in [1] are invalid. [12] goes

further to say that the only strategy space where selfish mining takes something from honest mining is when honest miners waste their efforts on a block which is a child of a predecessor block which, as [12] suggests is currently 0.01% of miners wasted efforts. [11] also highlight that as no evidence of a selfish attack occurring has been recorded, whether selfish mining or other mining strategies constitute equilibria is unknown. Kroll et al [23] further argue against selfish mining stating that an honest strategy within Bitcoin is a Nash equilibrium stating that in terms of long terms goals, the majority are looking to adopt an honest strategy to maximise their gains. However, the outcome of the equilibrium depends on the players of the strategies and where the players are happy with the choice they have made given the choice of the competitor. [1] suggest that the default Bitcoin protocol, because it allows block withholding and mining strategies to take place, cannot constitute an equilibrium, further saying that Bitcoin mining is in fact not incentive compatible. Whereas, [24] state that if the miners are sufficient small in terms of their computational power, the default mining behaviour is an equilibrium.

In summary, according to [25] the Bitcoin protocol is incentive-compatible where the best strategy of a miner acting rationally is to be honest and where a set of colluding miners cannot earn incentives disproportionate to their efforts by deviating from the Bitcoin protocol. The default protocol of Bitcoin rewards the miner in proportion to their mining power so collusion of said miners does not pose a threat to Bitcoin. However, [1] suggests that this is not the case and there exists a strategy or strategies which can be used by a minority of miners to obtain more revenue than their computational power. The author therefore decided to explore mining attacks by looking at the concept of mining in two ways. Firstly, the author will explore if the honest miners could deviate from honest protocol to maximise profits, would they and if they would, which strategy would they adopt. The author will also explore the strategy explained in [1] as well as exploring alternative strategies which seek to allow miners to gain disproportionate rewards to their computational power. Secondly, the author will explore the concept of selfish mining in more detail and look at selfish mining against stubborn mining and the utilisation of the strategies to reap disproportionate rewards.

4 Model and Simulations

It is important we model the simulation so that at a later stage, this model can be adapted towards analysis and comparison with other cryptocurrencies. The model employed is where the block reward is currently at 12.5, 6.25 and 3.125 Bitcoins (BTC) respectively. When exploring stubborn strategies, the block reward is kept at 12.5 Bitcoins. Like [3], transaction fees are a supplement incentive for the miner which are modelled at a constant rate. Again, similar to [3], in a typical mining occurrence, the miner claims all unclaimed transaction fees from the block they have just mined. The model used has no network latency show so propagation is immediate and, as in [7], miners are continually mining. As mining strategies are being employed within the simulator, it is important to understand what constitutes a mining strategy. The strategy is based on three core principles. The first principle is the location the miner is mining at in terms of the blockchain. In terms of the Bitcoin network, this is currently on the longest blockchain they have knowledge about. If there is a tie between the longest blockchain, the oldest blockchain is chosen. The miner can choose to follow protocol or follow action against the "*normal*" protocol which can create a fork in the blockchain. The second principle involves the percentage of the transaction fees the miner will claim upon successfully mining a block. The third and final principle is broadcasting to others in the network on the claim of successfully finding a block.

In terms of risk, if by employing the selfish strategy the colluding miners find a block only for the honest miners to find the subsequent block, there is a risk that the selfish miner will not be rewarded for their efforts so to lessen the risk, the strategy states that upon hearing the honest miners have found a block when the selfish miners lead by one, the selfish miners publish their hidden block straight away and hopes that as information is being propagated across the network, some of the honest miners will hear of the selfish block before the honest block and mine on this block. A risk or a decision to take a risk can lead to the effect of the actions being unclear and every miner or node has a varying understanding or approach in terms of behaviour to risk. The miner may be risk averse who opts for the action with no risk, risk neutral where they are indifferent and risk preferred where they will opt for riskier action. Normally, in deciding a risk strategy, different methods could be used. One method is the Bernoulli principle where we have a utility function for an output and then the decision is made based on the highest expected utility. For selfish mining or any mining strategy, the miner needs to decide based on the expected utility of their action and the gain they get from that. This is synonymous with the risky action or decision being a lottery in that the miner will opt for the action which will reap the highest reward. As demonstrated in Kaiser [26], considering risk is significant for Bitcoin and that miners are trying to avoid uncertainty. Kaiser also shows that only if selfish mining is more profitable, any risk averse miner will consider turning to it.

To this end, a Bitcoin mining simulator will be used to demonstrate some of these mining strategies/attacks. In [3], the authors focused on the attacks when the block reward has reduced to zero and the incentive to mine is purely down to transaction fees. The simulations used to demonstrate the attacks in this piece will start on the current block rewards of 12.5 BTC to 0 BTC and analysing the results of various simulations and mining strategies in which the transaction fees are important but with an accompanying block reward. The focus will be whether in attacks involving mining, the network can encourage nodes to act dishonestly to maximise their incentive. To this end, the Bitcoin simulator provides two ways in which to do this.

Firstly, the "*strategic*" simulator runs a series of simulations where miner nodes have a choice of strategy to employ and they are always looking to use the most effective one. The output from the simulator is the percentage of miners who used a certain strategy in each round. The simulator is created in Xcode using C++. For all simulations excluding the selfish simulation running the first simulator, 1000 games were run with 200 miners in each game. The rate at which transaction come in is 50 * 100000000 (Satoshi per Bitcoin) / 600 (mean time in seconds to find a block). Each simulation is carried out with atomic miners of which there are small but infinite amount. Latency is not present. The "*selfish*" simulator predominately focuses on the selfish mining attack where it assesses how effective this strategy is. The output of the simulator focuses on the percentage of hash power controlled by the miner as a comparison to the percentage profit it has earned by mining a block. For the selfish simulation running the first simulator, the following were kept consistent. 50 games are run with 20000 being the maximum size of the blocksize. The rate at which transaction come in is 25* 100000000 (Satoshi per Bitcoin). The mean time taken to find a block is 600 seconds. Each simulation is carried out with atomic miners of which there are small but infinite amount. Again, latency is not present. Using the simulators, a comparison will be made in terms of the attacks against the block reward which halves every four year and whether if given the opportunity, an honest miner would become dishonest.

In terms of the simulator used from [3] and [4] the author has made some changes to the code to allow the simulator to be used for simulating the adoption of strategies with the block reward still present. Appendix 1 shows the code relating to the simulation which provides the simulation results which has been edited, helping improve the simulator and instructions to be more accessible. The simulator has the ability to run a strategy that can be formalised using the rules provided in [3] and above. The simulator, as discussed in [3] is time driven in that strategies gain weightings over time which make them more appealing for miners to explore. Simulations take place through rounds and during each round, the miner uses the blockchain and the transactions they are aware of and the miner then decides as to which block to extend and the set of transactions to include. After this, the simulator determines if the miner found a block and then the miner makes

the decision to disclose the block or not. Each round is part of a game in which, as in [3], you can parametrise the game using several miners, strategies and the computational power of the miner. Each game leads to a set of rewards which are tracked for each miner. For this thesis, the author has decided to implement several games as part of the simulation where the decision to adopt strategies. The decision to adopt strategies is based on how well the strategy has performed using a no regret learning notion of learning as in [3]. The weightings of each strategy are used in terms of this learning notion to allow for exploration of other strategies. This simulator, as in [3] and [4] uses a single atomic miner who has a finite number of computational power and uses the network data available to the whole network only, so has no special data to give them an advantage. The calculation for the weighting is provided in [3] and is based on the profits earned by the miners of that strategy and averaging them for each strategy used. The important point to make is that new strategy exploration is encouraged within the calculation of the weight as otherwise, miners would stick to a single strategy and not explore other strategies. After the simulation is run, output is provided as shown in Figure 8 outputs each of the strategies as part of the simulation as a text file and the percentage of miners who used the strategy during each game. The full results are provided in Appendix 3.

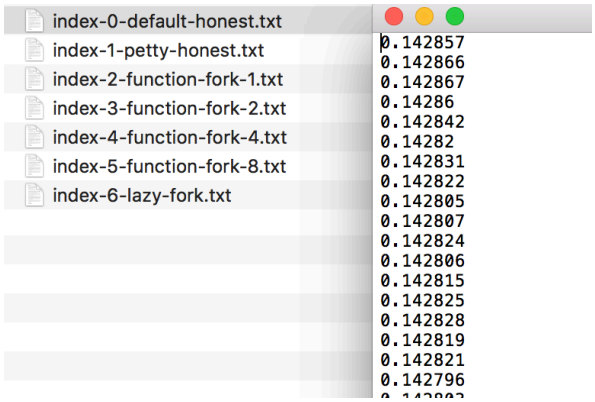

Figure 8 : File output and text output of the simulator

In terms of the second simulator, initially, the author developed their own simulator which provided data for the use of selfish mining in comparison to stubborn mining strategies. This simulator was developed in Python and this provided output in the form of profit probability resultant from varying values of alpha (Ω) and beta (β) as well as gamma (γ). Ω is used in the simulator to model the mining power of the "dishonest" miner. β is used in the simulator to model the mining power of the "honest" miner. γ is used in the simulator to provide the fraction of the honest miners who will support and mine on the adversary's chain in the event of a certain trigger event called a race condition. Similarly, to the simulator above, certain constants are maintained in the form of the number of simulations. In terms to check the findings, the author contacted [2] to confirm findings. [2] subsequently provided the author with their own simulator which they had used for their paper. As such, to provide as accurate findings, the author decided

to use the simulator provided by [2] and the hybrid stubborn strategy with the code for the simulator as shown in Figure 9.

```
public static Gain newLT(double a, double g){
        AbstractZeroHandler ltZeroHandler = new AbstractZeroHandler() {

                @Override
                public void zeroHandler(State init) {
                        double rand = StubbornMining.getRand();
                        if (rand < init.a) {
                                init.state = 1;
                        } else if (rand < init.a + init.g * (1 - init.a)) {
                                init.blocksGained += init.pending;
                                init.publicGained += 1;
                                init.pending = 0;
                                init.state = 0;
                                init.primeState = false;
                        } else {
                                init.state = -1;
                                init.primeState = false;
                        }
                }
        };

        Gain newLt = baseCode(a, g, ltZeroHandler, false);
        return newLt;
}
```

Figure 9 : Hybrid Stubborn Code from the simulator as shown in Appendix 2

The simulator provided by [2] is written in Java. Figure 9 provides the code for the hybrid stubborn strategy. Here, the code shows how the strategy progresses through the states explained in [2]. Here, blocks are either pending in that they are hidden and not discovered by the public miners and the states where a race condition is instigated (primeState) is also flagged if such an occurrence occurs. The code uses the Gain class provided as part of the simulator to record the gains made used Ω and γ values. For this thesis, 10000000 simulations were run to evaluate the profitability of the stubborn strategies. The simulation simulates the gains of the stubborn miner employing the different strategies while the honest miners mine honestly. Varying values of (Ω, β and γ are used where 100 sample paths are simulated containing 10^5 iterations, as shown in [2]. From the sample paths, the simulator estimates the mean of both stubborn and honest gains as well as the confidence interval for the estimate. In terms of the simulator, as suggested in [1], when strategies are compared for dominance, there are certain values where there is no dominant strategy which can be said at 95% confidence.

alpha gamma best_strategy | honest SM L F LF T LT FT LFT	
alpha	The computational power of the miner
gamma	The fraction of the honest mining power of the network that would back the adversary in the event of a race condition
best_strategy	The best strategy used for the combination of Ω and γ
honest	Honest Mining strategy
SM	Selfish Mining strategy
L	Lead Stubborn Mining Strategy
F	Equal Fork Stubborn Mining Strategy
LF	Lead/Equal Fork Stubborn Mining Strategy
T	Trail Stubborn Strategy

LT	Hybrid Stubborn Strategy
FT	Equal Trail Stubborn Strategy
LFT	Lead, Equal and Trail Stubborn Strategy

Figure 10 : how the output of the simulator is constructed

The output of the simulator has been modified by the author to accurately show increments of Ω by .5 each time. As shown in Figure 10, the output is in the form of a text file which provides firstly the Ω value being incremented along with the fixed γ value which is provided to the program. For these simulations, 10000000 simulations were run with the average taken at the end of the strategy completion. Different values of Ω, γ were attempted. Values for Ω up to 0.5 are considered as over this, the fundamental security properties of the currency are broken as the adversary can guarantee she earns all the mining reward. The output is the proportion of profits earned by the mining adopting the mining strategy using the values of Ω and γ. The values that are outputted in the order as shown in Figure 11.

```
0.1000 0.4000 0 | 0.1000 0.0651 0.0585 0.0573 0.0508 0.0634 0.0458 0.0550 0.0458
0.1500 0.4000 0 | 0.1500 0.1138 0.1013 0.0999 0.0867 0.1112 0.0758 0.0960 0.0757
0.2000 0.4000 0 | 0.2000 0.1719 0.1542 0.1537 0.1337 0.1686 0.1130 0.1480 0.1132
0.2500 0.4000 0 | 0.2500 0.2390 0.2194 0.2193 0.1947 0.2364 0.1623 0.2131 0.1621
0.3000 0.4000 1 | 0.3000 0.3161 0.2987 0.2985 0.2756 0.3148 0.2284 0.2939 0.2281
0.3500 0.4000 1 | 0.3500 0.4061 0.3969 0.3955 0.3839 0.4057 0.3208 0.3928 0.3216
0.4000 0.4000 4 | 0.4000 0.5172 0.5256 0.5178 0.5295 0.5182 0.4565 0.5187 0.4571
0.4500 0.4000 4 | 0.4500 0.6748 0.7059 0.6873 0.7283 0.6750 0.6628 0.6909 0.6620
0.5000 0.4000 1 | 0.5000 1.0000 0.9539 0.9502 0.9310 0.9428 0.9438 0.9475 0.9297
0.1000 0.4500 0 | 0.1000 0.0687 0.0629 0.0615 0.0563 0.0672 0.0514 0.0595 0.0512
0.1500 0.4500 0 | 0.1500 0.1185 0.1077 0.1061 0.0955 0.1157 0.0845 0.1025 0.0844
0.2000 0.4500 0 | 0.2000 0.1771 0.1630 0.1610 0.1459 0.1738 0.1252 0.1556 0.1254
0.2500 0.4500 0 | 0.2500 0.2445 0.2293 0.2276 0.2107 0.2412 0.1780 0.2215 0.1775
0.3000 0.4500 1 | 0.3000 0.3215 0.3097 0.3063 0.2941 0.3195 0.2472 0.3020 0.2476
0.3500 0.4500 1 | 0.3500 0.4111 0.4092 0.4031 0.4045 0.4103 0.3429 0.4009 0.3433
0.4000 0.4500 4 | 0.4000 0.5214 0.5379 0.5245 0.5495 0.5220 0.4791 0.5255 0.4801
0.4500 0.4500 4 | 0.4500 0.6777 0.7158 0.6923 0.7431 0.6765 0.6795 0.6957 0.6794
0.5000 0.4500 1 | 0.5000 1.0000 0.9602 0.9410 0.9460 0.9390 0.9543 0.9615 0.9619
```

Figure 11 : how the output of the simulator is shown

5 Strategies

Before reporting the findings of each of the strategies, it is important to discuss the strategies in the form of a comparison of each of the strategies against the default strategy which is utilised by Bitcoin. A simplified informal version is provided in Figure 13. This will provide a framework for the reader in the form of understanding of the core principles of the strategy which will help put some of the findings into context. The strategies are employed when miners deviate from the default protocol in terms of how to mine for the next block. As the thesis deals with mining strategies, as in [3], it is important to formalise these strategies to apply the same structure. As the majority of the strategies are adopted from [3], an attempt has been made to incorporate the method of formalisation shown within [3] with some slight changes in syntax as shown in Figure 12. For each strategy, the author will firstly, informally introduce the strategy before formalising the strategy. The core principles of each strategy are the same as in [3] which are to assess:

1. Which block to extend
2. How much of the available transaction fees to include in the block
3. Whether to disclose blocks to the nodes of the network so the data can be propagated.

Syntax	Explanation
m	Miner
B	Each block
T	Collection of Transactions
Mining(m)	Miner deciding the block to extend
TxFees	Total Transaction Fee
T(B)	The set of transactions included in B
TxFees(B)	The total transaction fees included in B
Rem(B)	The remaining transactions after the block has been mined which are not included in the block or the blockchain
Height(B)	Height of the blockchain which ends with B
H	Height of the current blockchain known to the network
Owner(B)	The miner who produced a block
Oldest $_i^m$	The block that the miner first became aware of it there is a race condition or their own block if it exists at height (i)
Most_i	Decide which block at height (i) contains the maximum remaining transaction fees
Most_i^m	The block at height (i) which the miner (m) mined or the one with the maximum remaining transaction fees.
Private^m	The height of the private chain which is not known to the public miners
Race $_i^m$	If there is a race condition where two competing blocks of height i exist
Private_i^m	The block at height i which the private miner (m) extends

Figure 12 : Key terms for formalising each strategy

It is important therefore, to start with the default protocol which is used by Bitcoin as a way of comparing other, "dishonest" strategies with.

Default Strategy [3]

As according to the rules, mine on the tallest block. If there is a tie, mine on the block which you heard off first. The miner claims 100% of the transaction fees and the miner publishes a block as soon as they discover one	
Which block to extend	Mining(M) = Oldest $_i^m$
How much of the available transaction fees to include in the block	Rem (Mining(M))
Whether to disclose blocks to the nodes of the network so the data can be propagated.	True

The default strategy is where the honest miner mines based on their computational power and attempts to solving the computationally difficult cryptographic puzzle to record their set of transactions (in the form of a block) on the public blockchain. The default miner mines on the longest blockchain and in the event of a tie where they receive information about two different blocks simultaneously, they mine on the block which they heard from first according to the block's timestamp. This forms the basis on how a mining strategy will determine which side of a fork it wants to support or to create a new fork. In terms of claiming rewards, the honest miner always take the whole reward, be that in the form of a block reward, transaction fees or both. Finally, as a default miner, as soon as the miner discovers a block, they notify the network and publish the block immediately and wait for the block to be confirmed as part of the public blockchain.

The strategies themselves can be split into two families. The first family of strategies focuses on instances where transaction fees are more prominent then the actual block reward and the avenues which miners can take to maximise their rewards in terms of transaction fees. The second family of strategies are more concerned with disproportionate rewards against on the actual computational power of the miner by withholding blocks in the hope of getting ahead.

Petty Complaint Strategy [3]

Mine on the tallest block and if there is a tie, mine off your block otherwise mine off the block with the least number of unclaimed transaction fees. The miner claims 100% of the transaction fees and the miner publishes a block as soon as they discover one	
Which block to extend	Mining(M) = Most$_h^m$
How much of the available transaction fees to include in the block	Rem (Mining(M))
Whether to disclose blocks to the nodes of the network so the data can be propagated.	True

Firstly, we have the petty compliant strategy. This strategy from [3] is used in the instance when the block reward is no longer the main source of income for a miner. This strategy deviates from the default protocol in the way that it deals with tie break in blocks. The petty complaint miner does not mine on the block they heard of first in the event of a tie. Instead, it favours the blocks that leaves the most remaining fees for the miner to collect. It does so in the hope of achieving greater revenue. In other words, the petty compliant miner always makes the decision on how to maximise their rewards. Although not much different other than the method resolution of a tie, as [3] suggests, this can lead to more aggressive mining strategies being employed.

Lazy Fork Strategy [3]

Fork the blockchain if the head of the blockchain has more unclaimed transaction fees. If it does not, mine on the tallest block. The miner claims 50% of the transaction fees and leaves 50% for another miner to prevent the block the miner mined from being undercut. The miner publishes a block as soon as they discover one	
Which block to extend	If Owner (Most_h^m) = M or Rem(Most_h^m) >= (Rem(Most_{i-1}) - Rem(Most_i)): Mining(M) = Most_h^m Else: Mining(M) = Most_{h-1}^m
How much of the available transaction fees to include in the block	Rem (Mining(M)) / 2
Whether to disclose blocks to the nodes of the network so the data can be propagated.	True

Next, we have the lazy fork strategy, again from [3]. This strategy is the first of the aggressive strategies which works on the premise of incentivising the petty compliant miners by extending the predecessor of their block and by ensuring that considerably less transaction fees are included. This method of replacing the block at the head with a new block in the effort of getting miners to defect is known as undercutting. The lazy fork strategy compares its rewards in terms of profit by undercutting or continuing to mine on a block and opts for the one with the greatest rewards. The strategy also anticipates falling foul of its own strategy but taking 50% of the transaction fees to prevent other miners adopting this strategy to do the same to them. This strategy would lead to many orphan blocks (blocks which are not extended) and as commented in [27], make Bitcoin more prone to double spend attacks.

Function Fork Strategy [3]

Mine on the location which allows you to mine the most valuable block which may involve choosing the block before the head of the blockchain as it offers the least number of unclaimed transaction fees. The miner claims the value of the function being used and publishes a block as soon as they discover one	
Which block to extend	If Owner(Most_h^m) = M or f(Rem(Most_h^m)) > min(f(Rem((Most_{h-1}^m)), (Rem(Most_{i-1})-Rem(Most_i))): Mining(M) = Most_h^m

	Else: Mining(M) = Most_{h-1}^{m}
How much of the available transaction fees to include in the block	If (Mining(M) = Most_{h}^{m}: include $\text{Val}_{\text{CONTINUE}}$(f) Else: include $\text{Val}_{\text{UNDERCUT}}$(f)
Whether to disclose blocks to the nodes of the network so the data can be propagated.	True

The next strategy is the function fork strategy. The function fork strategy explores the blocks at the head of the blockchain for expansion and only considers a block within this set which would leave the most transaction fees. The function fork miner can decide to either mine on top of the head of the blockchain, known as continuing or undercut the head of the blockchain. To decide regarding continuing or undercutting, the function fork miner compares continuing to the minimum revenue for undercutting. The method of progression is then decided to depend on the greater value. The function fork miner takes a function of the possible transaction fees it could claim and makes its decision based on the ability to maximise their profit with regard to the size of the block.

Selfish Mining Strategy [1]

Mine on your private chain if it is the last block you mined. If it is not, mine on the tallest block and if there is a tie, activate a race condition. The miner claims 100% of the transaction fees. Judiciously reveal blocks from the private branch to the public so that the honest miners will switch to the recently revealed blocks, abandoning the shorter public branch	
Which block to extend	Oldest $\overset{m}{\text{PRIVATE}^{M}}$
How much of the available transaction fees to include in the block	Rem (Mining(M))
Whether to disclose blocks to the nodes of the network so the data can be propagated.	If Height(B)=H: Yes Else if Race $\overset{m}{i}$ and Private^{m} = H+1: Yes Else: No

The selfish mining strategy, developed by [1] tries to get miners to mine on a block that will not make it to the public blockchain and become orphaned. The core principle behind selfish mining is to get miners to waste their computational power allowing the selfish miner to amplify their own computational power by maintaining a private chain which the public network knows nothing about. In stark contrast to the honest miner, when a

selfish miner discovers a block, they do not announce this to the network but instead continue to mine on their private chain in the hopes of discovering another block before the honest public has found a block. Should this be successful, this puts the selfish miner in pole position as they have 2 blocks ahead of the public blockchain. In the case of the honest miners discovering a block after this, the selfish miner need only reveal their private block to become the head of the blockchain, hence leading the honest mined block to become orphaned. With the selfish miner, with a lead of 2 block or more, [1] suggests that the selfish miner can guarantee that the rest of the network is wasting its computational power. There may be an instance where the selfish miners find a block and subsequently so do the honest miners. In this instance, the selfish miners would reveal their block and then it would come down to whichever block the nodes heard about first. Herein lies the tradeoff where, as [1] suggests, withholding blocks and creating a private chain of length 2 or more benefits the selfish miner but the selfish miner may also lose which results in their block becoming orphaned.

The next strategies, discovered by [2] belong to a family of strategies called stubborn mining strategies. In stubborn mining strategies, yet again, blocks are withheld leading to the rest of the Bitcoin network wasting its computational power on blocks that will not make it onto the public blockchain. In comparison to selfish mining, stubborn mining strategies look to increasing revenue by mining on the stubborn miner's private chain more often. In contrast to selfish mining, the stubborn miner can decide, if it is leading in terms of the blocks on the private blockchain, to reveal these blocks as well as how many of these blocks to reveal. As well as this, the stubborn miner, if a certain amount behind the honest miners can opt to support their blockchain and discard their attempts at a private blockchain.

Lead Stubborn Strategy [2]

Mine on your private chain if it is the last block you mined. If it is not, mine on the tallest block and if there is a tie, activate a race condition. The miner claims 100% of the transaction fees. In terms of publishing a block, reveal the next block on their private chain only, to match the length of the public chain.	
Which block to extend	Oldest $^{m}_{\text{PRIVATE}^M}$
How much of the available transaction fees to include in the block	Rem (Mining(M))
Whether to disclose blocks to the nodes of the network so the data can be propagated.	If(Height(B) = H): Yes Else if Race $^{m}_{i}$ and Private^{m} = H+1: Mining(B) = Private$^{m}_{Height(B)}$ Yes Else: No

The first such stubborn strategy is the lead stubborn strategy. In the lead stubborn strategy, if the stubborn miner is in lead with 2 blocks and the honest miners find the next subsequent block, the stubborn miner reveals a single block from their private chain to equate to the length of the public blockchain. This contrasts with the selfish miner who would instead reveal the entire contents of their private chain which then be favoured as opposed to the honest miners blockchain. For the lead stubborn strategy to work, a γ value is needed. Here, γ represents the fraction of the honest miners who will support the stubborn miners chain, which is known as a race condition. As with selfish mining, there are risks involved which are namely the stubborn miner losing their private chain but also benefits in the form of getting the honest miners who are supporting the public blockchain to waste their computational power. In the lead stubborn strategy, as opposed to selfish mining, if the lead of the private blockchain is two blocks or more, the lead stubborn miner reveals one of the private blocks to encourage a race condition. In contrast to the default mining strategy, lead stubborn miners opt to withhold blocks instead of immediately publishing them to the network in the hope of gaining greater revenue.

Trail Stubborn Strategy [2]

Mine on your private chain if it is the last block you mined. If it is not, mine on the tallest block. If you are m blocks behind, adopt the public blockchain. The miner claims 100% of the transaction fees. In terms of publishing a block, do not reveal the new block and continue to mine on it privately in the hope of catching up	
Which block to extend	Oldest $^{m}_{\text{PRIVATE}^M}$
How much of the available transaction fees to include in the block	Rem (Mining(M))
Whether to disclose blocks to the nodes of the network so the data can be propagated.	If(Height(B) = H): Yes Else if Race $^{m}_{i}$ and Private^{m} = H+1: Yes Else if H>2: Mining(B) = Private$^{m}_{Height(B)}$ Yes Else: No

The next member of the stubborn mining strategies is the trail stubborn strategy. The trail stubborn miner has a threshold value such that if the private chain of trail stubborn miners were to fall behind by the threshold value plus one, the trail stubborn miner would adopt the public blockchain and discard their efforts on their private blockchain. There is a risk within this strategy that if the trail stubborn miner would catch up and the forks be of equal length, all the honest miners would be mining on the public blockchain and hence no amount of γ would support the adversary in this race condition.

Hybrid Stubborn Strategy [2]

Mine on your private chain if it is the last block you mined. If it is not, mine on the tallest block and if there is a tie, activate a race condition. If you are m blocks behind, adopt the public blockchain. The miner claims 100% of the transaction fees. In terms of publishing a block, reveal the next block on their private chain only, to match the length of the public chain.	
Which block to extend	Oldest $^{m}_{\text{PRIVATE}^M}$
How much of the available transaction fees to include in the block	Rem (Mining(M))
Whether to disclose blocks to the nodes of the network so the data can be propagated.	If(Height(B) = H): Yes Else if Race $^{m}_{i}$ and Private^{m} = H+1: Mining(B) = Private$^{m}_{Height(B)}$ Yes Else if H<3: Mining(B) = Private$^{m}_{Height(B)}$ Yes Else: No

Using the family of stubborn mining strategies, the author has created a hybrid strategy which is discussed in [2] as an avenue of future work. [2] explains that there is an infinite number of strategies which can be developed by combining different strategies together. The author, as such, has developed a hybrid strategy from the lead stubborn and trail stubborn strategies. The strategy works as follows. The miner adapts its "stubbornness" based on factors during the mining process. As suggested in [2], the author has adopted the factors being the length of the private blockchain. The hybrid strategy dictates that the hybrid miners will have a lead stubborn attitude when it comes to race conditions but also adopt a trail stubborn attitude for the lead against the public blockchain or being behind the public blockchain. The author has chosen a threshold value of three for the strategy where a trail stubborn approach is adopted.

6 Discussion and Analysis

6.1 Finding 1: Default miners will always cheat, given the chance

Irrespective of the block reward, given the choice, the default compliant miner (honest miner) will always change their strategy to maximise their profits. The simulation demonstrates that as the block reward decreases the function fork strategy where the miner undercuts and leaves 50% of the transaction fees behind to stop others from undercutting them is consistently the most popular strategy irrespective of block reward. Interestingly, adopting a petty honest strategy where the miner focuses on transaction fee wealth (the amount of transaction fees that are unclaimed) as seen in [3], is a strategy which is not popular. This demonstrates that the default miner will act rationally instead of honestly

if they are given the opportunity as the miners in the simulation choose their path according to the weighting of exploring different strategies. Another interesting point is that as the simulation progresses although firstly, many drop the petty honest strategy, less and less miners decide to drop it as the block reward reaches zero. This shows that the findings in [3] are correct and that as the block reward reduces, the transaction fees will become more and more important, possibly leading miners to act dishonestly. [4] shows that petty compliant mining outperforms default complaint in terms of gaining revenue.

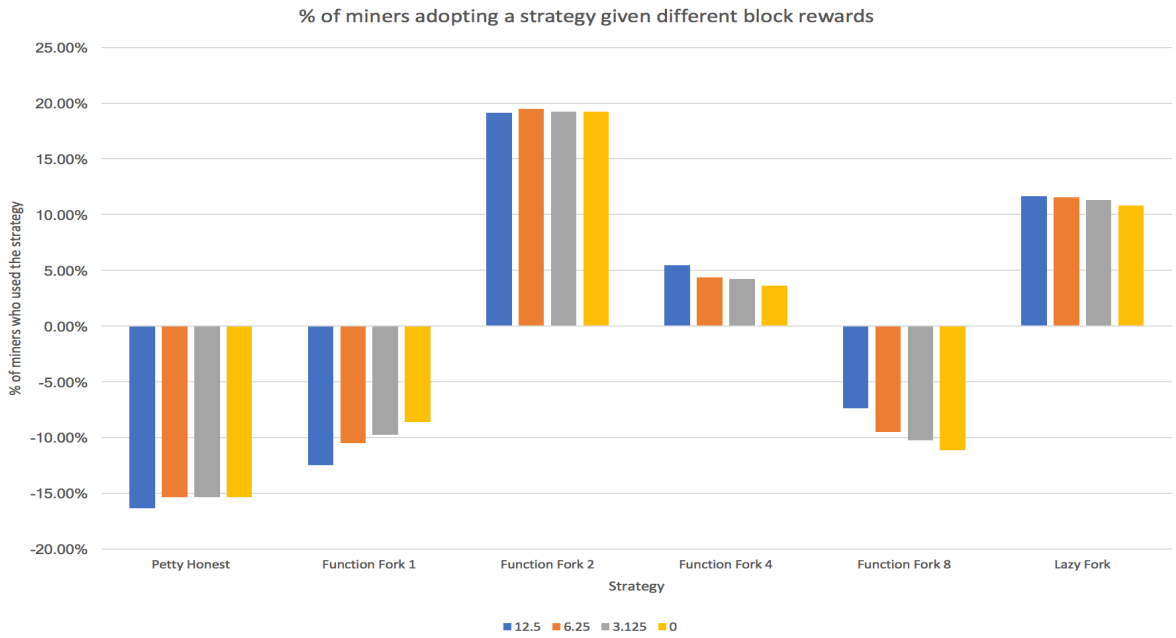

Figure 13 : How mining behaviour changes given the change in block reward

As Figure 13 demonstrates, most of the default compliant miners adopt a function fork or lazy fork strategy. This questions the design of the system, as the network is built on the assumption that some part of the network is honest. As miners are incentivised to act honestly, the findings show that this assumption is not realistic as one might believe. The direct implication is if dishonest behaviour is more rewarding then you assume the miner adopts the most rewarding strategies. The assumption that most of the network will work honestly is therefore called into question.

6.2 Finding 2: Undercutting is preferred

As Figure 13 demonstrates, the function fork strategy is the most popular mining strategy. Interestingly, the function fork strategy where the function gains all the transaction fees (Function Fork 1) is not popular at all. This is synonymous to acting selfish in terms of revenue not the mining behaviour. Therefore, the fact that these strategies where all the transaction fees are taken are not popular demonstrate that within the Bitcoin network, there needs to be some give and take in terms of incentivising others to collude with you

or to protect yourself from others colluding against you. As Figure 14 shows, a function fork strategy was adopted 24% of the time. The most popular mining strategy is Function Fork 2 where the miner takes half of the rewards and leaves half as an incentive for someone to build on their block after undercutting. This agrees with [4] who state that forking the blockchain by undercutting is preferred as this maximises the wealth of the blocks the miners mine on but also reduces the likelihood that other miners will undercut the colluding miners as there is a reward not to do so. 19% chose to abandon mining honestly where the deal breaker for a tie is to mine on the block where the most unclaimed transaction fees reside. Interestingly, when the amount taken is reduced, the adoption of this policy is reduced too which seems to indicate that miners that are "too selfish" are not preferred. This is demonstrated in the function fork 1 strategy which is not preferred where you take all of the rewards. Lazy forking is also popular among the honest miners that want to abandon their honesty and choose a different path. On average, 11% adopted this strategy after ditching honest mining. The lazy forking strategy looks to undercut the block that is sitting on the highest blockchain and if it does this, it takes 50% of the transaction fees and block reward to incentivise other miners not to fork its block.

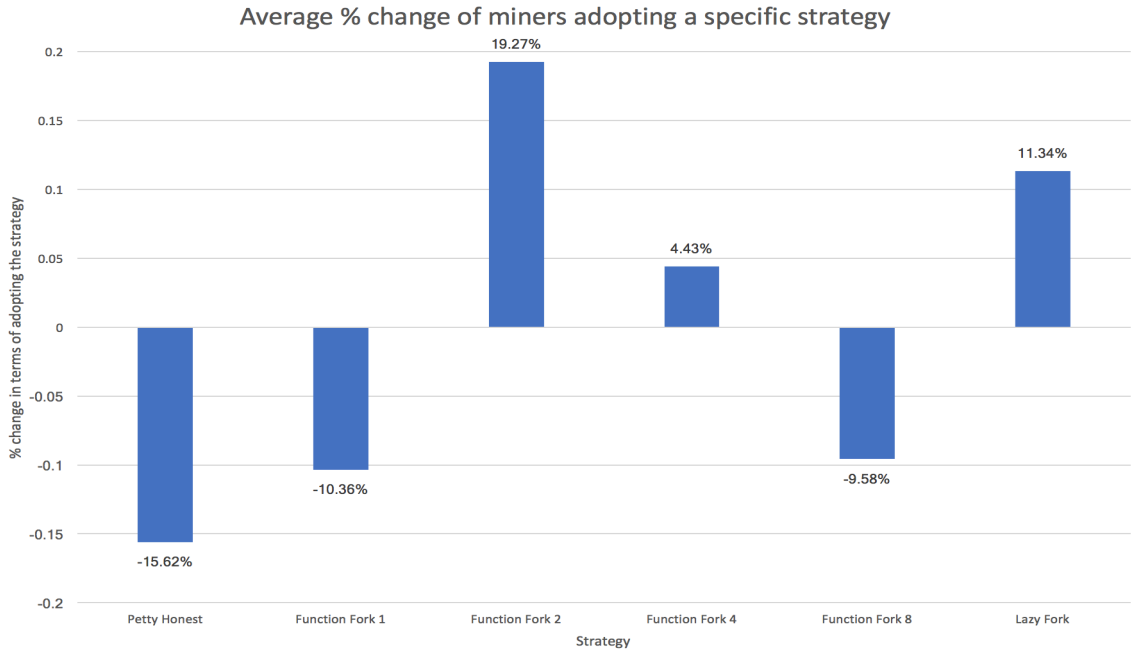

Figure 14 : Average adoption of strategies across all block rewards

What this demonstrates is undercutting is a preferable way for miners to maximise their profits but equally, leaving 50% of the transaction fees present in the block as an incentive for others not to undercut your block is popular too. This questions the motives of the miner in terms of acting honestly. If the miner can cause forking in the blockchain where the number of orphaned blocks is increased, this paves the way for an increase of double spend attacks and eventually could destabilise the Bitcoin currency. The results of the simulation demonstrate that undercutting is popular and this is bad news for

Bitcoin as if the miners were to act rationally and weigh up their options before selecting the strategy that would reap them the greatest revenue, this would lead to an increase in orphaned blocks due to multiple undercutting taken place which in turn could have an impact on communication and confirmations between nodes.

6.3 Finding 3: Selfish Mining has the capability to cause damage

As stated in [1] and [7], mining and the revenue a miner can receive for blocks mined are proportional to the amount of computational power that miner has. Often, miners form pools in which they can all baluster the computational power of the pool and seek to improve their chances of finding the block that makes its way onto the public blockchain. The results of the simulation question this proportionality. The simulation shows that as the block reward will decrease, the potential to earn disproportionate rewards based on your computational power remains consistent. The results of the simulation show that the disproportionality starts when we have 0.3 or 30% of the computational power of the network. Interestingly, as shown in Figure 15, the current largest mining pool controls 17.2% of the Bitcoin network but the top two mining pools control over 30% of the total network.

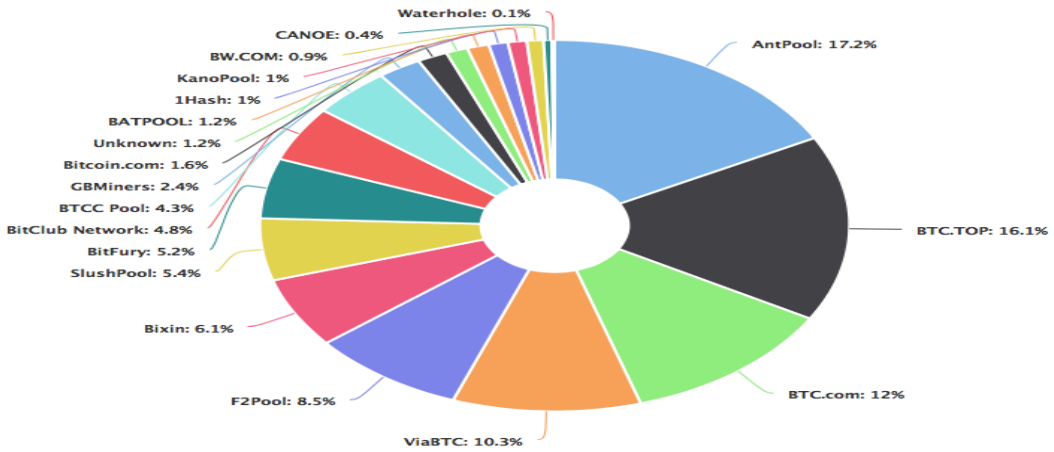

Figure 15 : Distribution of computational power and mining pools [28]

Selfish mining strategies show, as is confirmed in the simulation, that with a 0.4 or 40% of the computational power of the network, the miner or pools of miners can achieve 50% of the revenue of the blocks mined. Selfish mining strategies demonstrate that a marginal pool can achieve more revenue than its computational power dictates. The fact that for lower computational power, the miner could achieve higher profits incentivises rational miners to jump from the honest strategy to the selfish strategy. As Figure 16 demonstrates, regardless of the block reward, after the computational power of the miner exceeds 0.3, the rewards begin to get disproportionate.

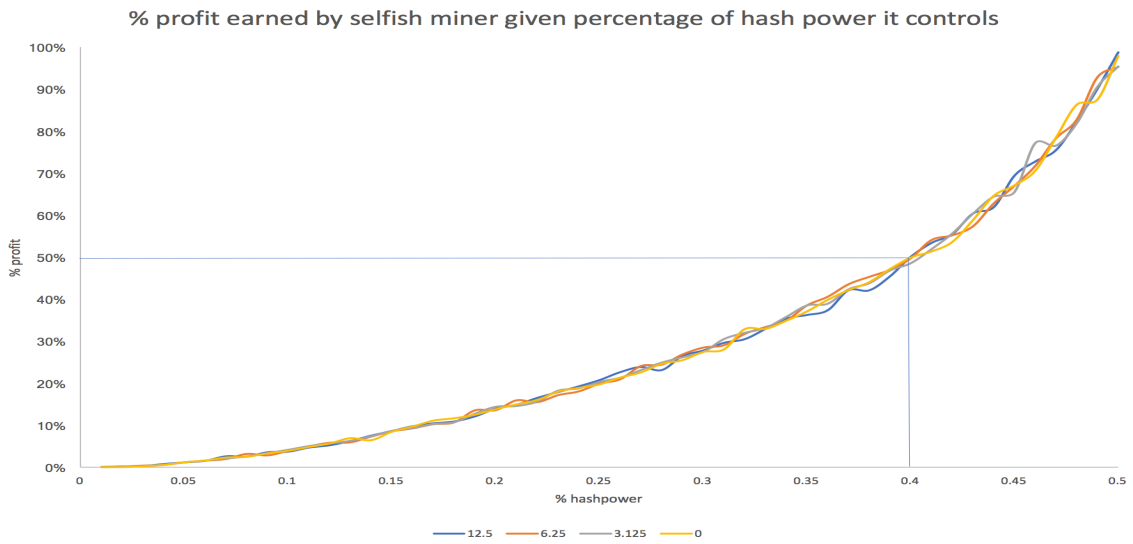

Figure 16 : Selfish Mining profits based on computational power

The implications of this, as suggested in [1], is that as the selfish mining cohort grow in size, the honest miners will act rationally and prefer to join the selfish cohort to get the higher rewards. As [1] suggests, this will mean that the marginal pool will grow into a majority pool, they then have the power to modify protocols to allow for attacks such as block withholding attacks or even simpler, they could ignore blocks within the network that were not created by one of their miners and gain all the mining revenue by being the only ones whom create blocks and confirm blocks. This also has implications for the decentralised nature of the currency as it is then no longer decentralised as one person or group have control of the Bitcoin network.

6.4 Finding 4: Honest mining still has a place for low computational power and γ values

Despite disproportionate rewards being gained by other strategies, the simulations have demonstrated that honest mining still very much has a space within the strategy space of a rational miner. According to the simulation findings, for low values of Ω (the relative fraction of computational power controlled by the miner, honest or dishonest) where Ω is 0.3 or less and when γ (the relative fraction of the honest mining fraternity that would back the miner, honest or dishonest in a race condition) is 0.2 or less, honest mining very much is the best strategy with an in terms of relative gains against other strategies. Relative gains are calculated with the percentage gain of one strategy subtracted from the gain of the comparing strategy over the Ω value as in [2]. The Ω value is used to represent the dishonest miners gain under the utilisation of an honest strategy. This means that the core foundations of the Bitcoin currency are very much strong and accurate in terms of the assumption of computational power and revenue. This means that

the miner could act honestly when they are "*computationally weak*" or have little influence within the network but as their influence increases or they become part of a pool or their computational power increases, they could then start to adopt alternative strategies.

Average relative gain of Honest Strategy compared to other strategies				
	γ			
	0	0.1	0.2	Average
Selfish Mining	35.92%	30.55%	25.17%	30.54%
Lead Stubborn	63.26%	51.07%	40.46%	51.59%
Trail Stubborn	35.81%	30.96%	26.03%	30.93%
Hybrid Stubborn	100.00%	82.32%	67.30%	83.20%

Figure 17 : Average relative gains (in terms of percentage) of honest strategies over other strategies

Figure 17 demonstrates the relative gains for honest mining against the other strategies for the Ω and γ values discussed above. This shows that as the miners' computational power grows, other strategies become more fruitful in terms of revenues to explore because for the same computational power, they being to provide disproportionate revenue. For honest mining, when the miner or pool of miners has a high fraction looking to back them in the event of a race condition, honest mining is not preferred. The implications of this are that if the miner has high influence on the network, they can turn honest miners into rational miners by helping them maximise their revenue disproportionately to their computational power. This yet again questions the core principle of the security of Bitcoin relying on miners being honest.

6.5 Finding 5: Lead stubborn strategies dominate the middle ground in terms of Ω and γ

As the γ fraction increases past 0.3, the lead stubborn strategy begins to become dominant where Ω is between 0.3 - 0.4. In other words, if control the top 2 mining pools in the Bitcoin network were possible and the miners within the network were rational rather than honest, they would leave the honest strategy for a lead stubborn strategy. However, honest mining still dominates in a small space particularly when the Ω value is 0.1. The lead stubborn strategy looks to the lead stubborn miner astutely revealing blocks on their private chain to encourage a race condition between the dishonest miner or pool of miners and the honest miners. In this middle ground, lead stubborn strategies can perform up to 54% better than honest mining and 12% better than selfish mining. It is important to quantify this relative gain as shown in [1]. Based on [28], as of August 2017, all bitcoin miners on average earn just under 9 million dollars per day. In an example where the Ω value is 0.4 and the γ value is 0.6, lead stubborn strategies, in terms of relative gain, are

41% better than honest mining and 7% better than selfish mining. Using the relative gain calculation from [2], this 7% difference between the stubborn strategies and the selfish strategy would equate to:

$$9000000 \text{ x } 0.07 \text{ x } 0.4 = \$252{,}000 = £194{,}573$$

This demonstrates that a 7% increase in computational power is the equivalent of an extra £194,000 worth of profit based on the current Bitcoin exchange at the time of writing.

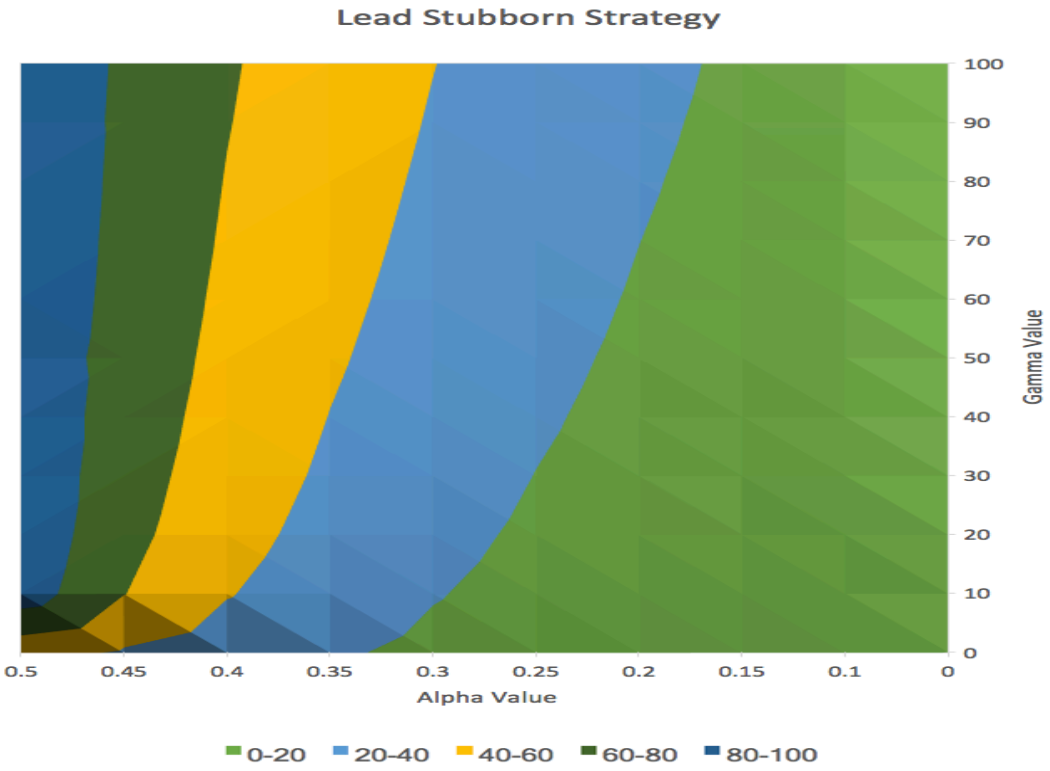

Figure 18 : Percentage gains in terms of relative gains and profitability when employing a lead stubborn strategy

As Figure 18 shows, disproportionate profits are possible with computational power being 0.3 when a γ value is zero and as the γ fraction increases, the Ω power needed to gain disproportionate profits also decreases although is it a marginal disproportionality for an. However, as soon as the Ω value becomes 0.4 and the γ reaches 0.4, there is a substantial shift in terms of disproportionately where this combination of values gain 50% or higher gains and this continues throughout the simulations where ultimately, with a high gamma fraction, the lead stubborn miner can get disproportionate gains to the tune of 60% when the γ value is 0.9 and Ω is 0.4. This yet again demonstrates that employing strategies other than the honest strategy is the best course of action to maximise profitability and gains.

6.6 Finding 6: Hybrid Stubborn has a place in the strategy space

The author's hybrid stubborn strategy is one which employs lead stubborn and trail stubborn elements within the behaviour of the miner dependent on different circumstances. The hybrid strategy starts to outperform other strategies when the γ value is 0.7 or greater for some values of Ω. This leads up to the hybrid stubborn strategy being dominant when the γ value is 0.9 or higher against other strategies. In comparison to honest mining, the hybrid stubborn strategy outperforms honest mining in terms of relative gains for certain values of Ω when γ is 0.5 or higher as shown in Figure 19.

Average relative gain of hybrid stubborn strategies compared to other strategies					
	γ				
	0.7	0.8	0.9	1.0	Average
Honest Mining	21.41%	29.91%	38.78%	47.48%	34.39%
Selfish Mining	-3.29%	1.54%	6.73%	11.75%	4.18%
Lead Stubborn	-5.57%	-2.15%	2.19%	6.28%	0.19%
Trail Stubborn	-1.26%	2.97%	8.27%	12.73%	5.68%

Figure 19 : Average relative gains (in terms of percentage) of hybrid stubborn strategies over other strategies

The hybrid stubborn strategy is a preferred strategy for when a high fraction of miners are acting rationally and want to maximise their profits. If a high ratio of these miners will support the hybrid stubborn miner to maximise their relative gains, the hybrid stubborn strategy outperforms honest mining and other popular mining strategies. In the simulations, as shown in Figure 20, for Ω values higher than 0.45, as γ increases, disproportionate profits begin to be reaped by hybrid stubborn miners as the γ increases and the Ω value decreases. This seems to suggest that if two relative stubborn strategies are employed as a behaviour for a miner to then switch between during the simulation, the miner will act rationally and play the

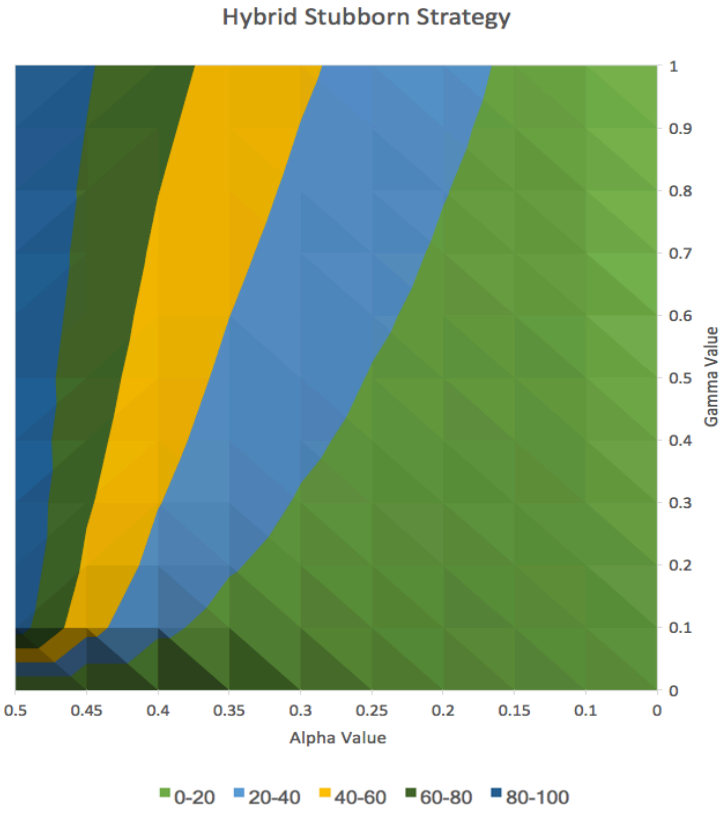

Figure 20 : Percentage gains in terms of relative gains and profitability when employing a hybrid stubborn strategy

strategy which will gain them a disproportionate reward in comparison to the amount of computational power it controls.

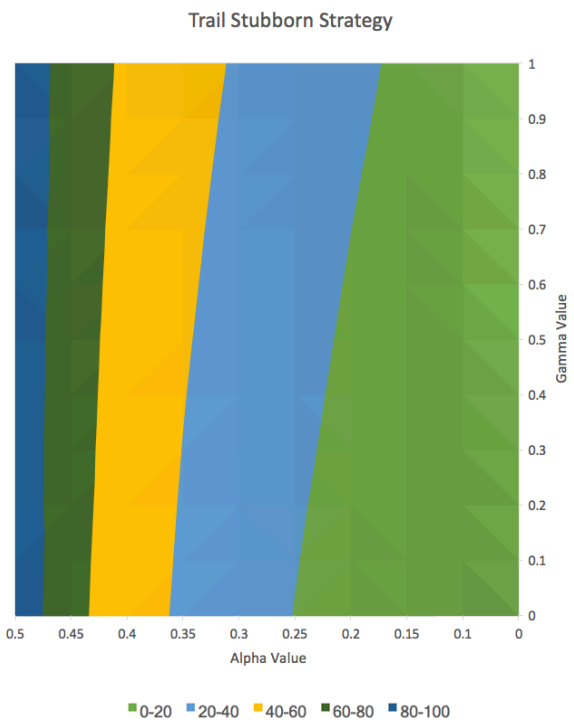

Figure 21 : Percentage gains in terms of relative gains and profitability when employing a selfish strategy

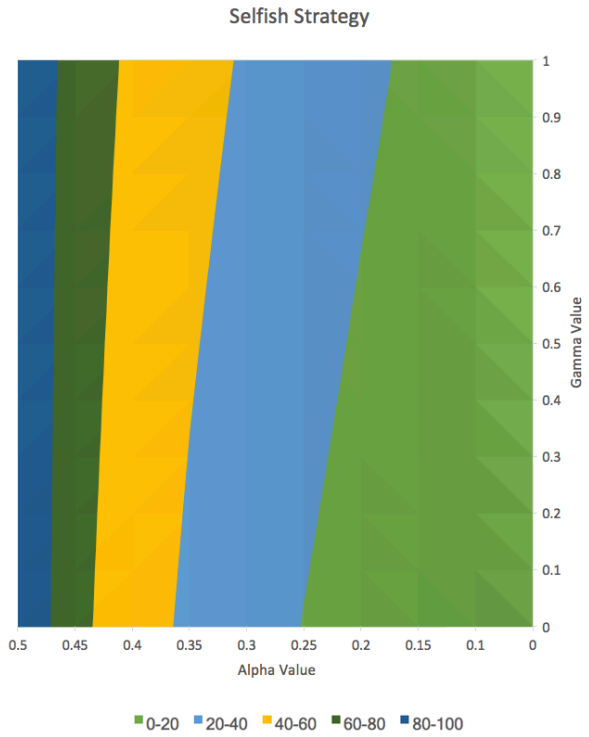

Figure 22 : Percentage gains in terms of relative gains and profitability when employing a trail stubborn strategy

Although the hybrid strategy has a place in the strategy place, it is worthwhile noting that the strategy space in which hybrid stubborn is successful is small in comparison to other strategies such as the lead stubborn strategy or the trail stubborn strategy which occupy a great dominance of the strategy space. The findings also show that whereas selfish mining has had much prominence in terms of research, it occupies a small strategy space as well in comparison to other stubborn strategies. As Figure 21 and Figure 22 show, both trail and selfish mining strategies share an almost identical distribution of profitability. This demonstrates that enforcing a cap in terms of the number of blocks the trail stubborn miner has before they adopt the public blockchain can be as profitable as employing the selfish strategy in terms of concealing blocks from the network to get them to waste computational power on blocks that are not destined for the public blockchain.

7 Implications for Bitcoin

In terms of detecting these strategies, there would be some tell-tale signs as suggested in [1] and [2] that an attack using these strategies had taken place. The first of these would be the number of orphaned blocks there would be. An orphan block is named as such it is a block which is ostracised following a race condition and does not make it onto the blockchain. [1] suggests that such blocks would be difficult to account for as the network itself carries out maintenance to prune blocks so a measurement of the number of blocks attributed to mining attacks would be difficult. Another detection method, suggested by [1] would be the timing between blocks on the blockchain. If a miner decides to hide blocks instead of publishing this, this would lead to disproportionate gaps between blocks being added to the blockchain. This could lead to the discovery of mining strategies being adopted, and as suggested previously, timestamps or methods such as GHOST or Freshness Preferred could be the answer to this problem. The GHOST protocol as suggested by [29] is an alternative to the longest chain rule where network security loss is reduced by including orphaned blocks in calculating which blockchain is the longest as well as providing block rewards for orphaned blocks at a reduced rate. The Freshness Preferred protocol [30], uses unforgeable timestamps which seeks to solve issues with block withholding as in the event of a tie or race condition, a newer block will always win. This suggests changing the protocol to add timestamps to each block which are unforgeable and preferring blocks which are more recent as opposed to older blocks. This decreases the profitability as in a race condition, the more recent blocks will always win. [30] provides this method which makes selfish mining less profitable than honest mining for Ω values less than 0.32. Another proposed solution to selfish mining strategies is proposed in [31] where Zeroblock prevent selfish mining as if a selfish miner keeps a block private for longer than a fixed period of time, the block is rejected by the honest miners.
Currently, these are some of the methods which could be used to detect an attack has happened but as of the time of writing, there is no mechanism in the Bitcoin network to stop the mining strategies from taking place. Albeit, no such activity has ever been reported on the network, it is, nonetheless possible. To counteract mining strategies, Bitcoin Improvement Protocols (BIPS) have been suggested to lower the likelihood of a selfish or stubborn attack taking place and one such resolution is to randomly assign miners to branches in the event of a fork or a threshold for mining pools.

The main implication based on the findings is that a minority pool, who are receiving disproportionate rewards, would grow into a majority pool. This majority pool could then cause problems for the decentralised nature of Bitcoin by possibly creating their own protocols and possibly reject transactions or blocks from those nodes who do not side with them. A simple method that the Bitcoin community could adopt to overcome this is by allowing a certain threshold size where miners or pools of miners must have a computational power below or equal to the threshold size. In terms of threshold size, in

terms of default protocol, a hard fork would be needed which could destabilise the currency based on recent events. Another such implication would be the aggressiveness of the strategies. If, like in [3], miners were to aggressively fork each other's blocks then the number of orphaned blocks would increase and, as [26] highlights, the difficulty to find a block would drop as the longest blockchain would be missing a proportion of blocks. This could also lead to other attacks such as a 51% attack being easier to carry out as suggested in [26]. In terms of overcoming these strategies, the whole network would need to ensure consensus in terms of how to deal with these problems. It is envisioned that if such an attack were to be carried out, then the network would retaliate and change their approach. However, achieving this, in practice would be difficult in terms of the sheer number of nodes involved. In terms of the hybrid stubborn attack, [2] suggest a stubborn attack must be sustained for a least two weeks to gain extra revenue so such activity could be prevalent if mining pools are monitored. Due to the decentralised nature of the cryptocurrency, this would be important to reach a consensus in terms of the threshold size and the methods of overcoming the strategies if they are ever discovered in the network, which may be difficult as [11] highlights as the way Bitcoin is governed is uncertain.

8 Conclusion

This thesis has demonstrated that mining strategies are an interesting weakness of the Bitcoin network. Such strategies are vast as suggested in [1] and [2], the strategy space is vast and cannot be explored in a single thesis. The author has demonstrated that a hybrid strategy does exist which can provide disproportionate revenue for a fraction of the computational power that is used. The author has also demonstrated that the hybrid stubborn strategy can outperform its competitor strategies in certain circumstances. Through looking at past literature and the author's use of simulators, the thesis has shown that the strategy space for mining strategies is vast and that strategies exist and thrive where miners act rationally instead of honestly. The thesis has also shown that Carlesten et al [3] were correct in the use of deviant mining strategies but the thesis proves that you would not have to wait till the block reward is zero to experience such deviant behaviour. The thesis does result in some interesting development points. In terms of future development, the author can see merit in exploring a stubborn selfish hybrid strategy where, dependent on outcomes and probabilistic actions, the miner takes certain actions in a stubborn way and then completes other actions using a selfish strategy. Similar work has been carried out by [21] but it would be interesting to see if this strategy would overcome the strategy mentioned in [21]. Secondly, it is important to carry out more risk analysis involving the simulations, looking at the risks and expected payoff involved with running certain strategies, considering confidence intervals within the calculations. An interesting future development, would be to explore this tradeoff in terms of the expected payoff for the miners who employ this strategy and carry out risk

analysis when it comes to adopting the selfish mining strategy. Also, exploring whether latency in which the nodes hear about blocks being found could have a detrimental effect on revenue, could provide further insight. It could mean that the higher the relative latency, the quicker the relative speed the dishonest miner announces their block after hearing about the honest miner's block. Finally, in terms of the strategies, it is important to see whether if methods to counteract the strategies do work so exploring the Freshness Preferred as an answer to these attacks would be necessary work in the future.

9 References

[1] I. Eyal and E. G. Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography*, 2014.

[2] K.Nayak, S. Kumar, A. Miller, and E. Shi. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrucken, Germany, March 21-24, 2016*, pages 305–320, 2016.

[3] M. Carlsten, H. A. Kalodner, S. M. Weinberg, and A. Narayanan. On the instability of bitcoin without the block reward. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016,* pages 154–167, 2016.

[4] M. Carlsten. The Impact of Transaction Fees on Bitcoin Mining Strategies. PhD diss., Princeton University, 2016.

[5] L. Qi. *Blockchain and Bitcoin (1).* 2017.

[6] E. Chatellier. "Bitcoin Transactions", *Ceric35.net*, 2017. [Online]. Available: http://ceric35.net/bitcoin/img/Bitcoin_Transaction_Visual.png. [Accessed: 01- Aug- 2017].

[7] S. Nakamoto. "Bitcoin: a peer-to-peer electronic cash system," *http://www.bitcoin.org/bitcoin.pdf* , 2015.

[8] A. Narayanan, J. Bonneau, E. Felten, A. Miller and S. Goldfeder. *Bitcoin and cryptocurrency technologies*, 1st ed. Princeton: Princeton University Press, 2016, pp 51-76

[9] K. Shirriff. "Bitcoin mining the hard way: the algorithms, protocols, and bytes", *Righto.com*, 2014. [Online]. Available: http://www.righto.com/2014/02/bitcoin-mining-hard-way-algorithms.html. [Accessed: 04- Aug- 2017].

[10] "Mining - Bitcoin Wiki". *En.bitcoin.it*, 2017. [Online]. Available: https://en.bitcoin.it/wiki/Mining. [Accessed: 01- Jun- 2017].

[11] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten. SoK: Research perspectives and challenges for Bitcoin and cryptocurrencies. In *2015 IEEE Symposium on Security and Privacy, SP 2015*, pages 104–121, 2015.

[12] N. T. Courtois and L. Bahack. On subversive miner strategies and block withholding attack in bitcoin digital currency. *arXiv preprint arXiv:1402.1718*, 2014.

[13] M. Vasek, M. Thornton, and T. Moore. Empirical analysis of denial-of-service attacks in the Bitcoin ecosystem. In *Workshop on Bitcoin Research*, 2014.

[14] J. Garay, A. Kiayias, and N. Leonardos. The Bitcoin Backbone Protocol: Analysis and Applications. *Cryptology ePrint Archive, Report 2014/765*, 2014.

[15] L. Bahack. Theoretical bitcoin attacks with less than half of the computational power (draft). *arXiv preprint arXiv:1312.7013,* 2013.

[16] M. Conti, C. Lal, S. Ruj, et al. A Survey on Security and Privacy Issues of Bitcoin. *arXiv preprint arXiv:1706.00916*, 2017.

[17] C.A. Vyas and M. Lunagaria, Security Concerns and Issues for Bitcoin. In *the proceedings of National Conference cum Workshop on Bioinformatics and Computational Biology*, NCWBCB-2014, 2014.
[18] I. Eyal and E. Gün Sirer. Bitcoin Is Broken, *Hacking Distributed*, 2013. [Online]. Available: http://hackingdistributed.com/2013/11/04/bitcoin-is-broken/. [Accessed: 01-Sep- 2017].
[19] C. Decker. On the scalability and security of bitcoin. PhD. diss. 2016
[20] A.Sapirshtein, Y.Sompolinsky, and A.Zohar. Optimal selfish mining strategies in bitcoin. *arXiv preprint arXiv:1507.06183*, 2015.
[21] S.Pachal. Rational Mining on Bitcoin. MSc. thesis. 2017
[22] J. Göbel, H. Keeler, A. Krzesinski, and P. Taylor. "Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay," *Performance Evaluation*, July 2016.
[23] J. A. Kroll, I. C. Davey, and E. W. Felten. The economics of Bitcoin mining, or Bitcoin in the presence of adversaries. In *WEIS*, 2013.
[24] A. Kiayias, E. Koutsoupias, M. Kyropoulou, and Y. Tselekounis. Blockchain mining games. In *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16*, pages 365–382, 2016.
[25] S.Barber, X. Boyen, E. Shi, E. Uzun. Bitter to better - how to make bitcoin a better currency. In: *Financial Cryptography and Data Security*. Springer (2012)
399–414
[26] M. Kaiser. Simulation and analysis of the selfish mining attack on BSc. thesis. 2016
[27] Y. Sompolinsky and A. Zohar. Bitcoin's security model revisited. *arXiv preprint arXiv:1605.09193*, 2016.
[28] "Hashrate Distribution". Blockchain.info, 2017. [Online]. Available: http://blockchain.info/pools. [Accessed: 01- Jul- 2017].
[29] Y. Sompolinsky and A. Zohar. Secure high-rate transaction processing in bitcoin. In *Financial Cryptography and Data Security*, 2015.
[30] E.Heilman. One weird trick to stop selfish miners: Fresh bitcoins, a solution
for the honest miner. In *IACR Cryptology ePrint Archive*, 2014:7, 2014.
[31] S. Solat and M. Potop-Butucaru. "ZeroBlock: Timestamp-Free Prevention of Block-Withholding Attack in Bitcoin." *arXiv preprint arXiv:1605.02435* , 2016
[32] G. Karame and E. Androulaki. *Bitcoin and blockchain security*, 1st ed. Norwood, Massachusetts: Artech House, 2016, pp. 34-54.

10 Code

Although, the author has not created a software solution, the files associated with the simulator and findings have been uploaded to the Git repository. The address of my Git project repository is:

https://git-teaching.cs.bham.ac.uk/mod-msc-proj-2016/gxs612.git

The contents of Git project repository are shown below:

File	Information
Data Collection.xlsx	This is the file which contains the data collected as part of the thesis as an Excel file
SelfishSim.py	This is a Selfish Mining simulator provided by HurlSly@reddit which shows how a selfish mining strategy can be adopted
StubbornSim13.py	This is the incomplete attempt by the author regarding mining strategies. It contains both lead and trail stubborn strategies along with the hybrid stubborn strategy. As explained in the thesis, upon confirming data against [2], a simulator was provided by [2] to confirm results (Simulator2.zip) which was then used instead. This is incomplete in the sense that the results may not be accurate as dictated by Simulator2.zip from [2]
Simulator.1.zip	This is the simulator provided by [3] which can be used within Xcode. To use the simulator, the Readme.md file from the developer has been provided, which explains how the simulator operates
Simulator2.zip	This is the simulator provided by [2] which can be used as a Java application. To use the simulator, the ReadMe.md file within the Simulator shows how to utilise the simulator from the command line

11 Bibliography

[32] M. Möser and R. Bohme. Trends, tips, tolls: A longitudinal study of bitcoin transaction fees. In *Workshop on Bitcoin Research*, pages 19–33, 2015.
[33] E. Heilman. One weird trick to stop selfish miners: Fresh bitcoins, a solution for the honest miner. *IACR Cryptology ePrint Archive, 2014:7*, 2014.
[34] A. Gervais, G. O. Karame, K. Wust, V. Glykantzis, H. Ritzdorf, and S. Capkun. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 3–16. ACM, 2016.
[35] I. Eyal. The Miner's Dilemma. In *IEEE Symposium on Security and Privacy*, 2015.
[36] S.Bullock. M2006 Lecture 21: Bitcoin and Blockchain, 2016
[37] C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In *IEEE P2P*, 2013.
[38] M. Kiran and M. Stannett. "Bitcoin risk analysis," Available: http://www.nemode.ac.uk/wp-content/uploads/2015/02/2015-Bit-Coin-risk-analysis.p Dec. 2014.
[39] K.Spotz. Bitcoin's "Political Neutrality is a Myth" - Amir Taaki Interview. [online] *Cointelegraph*. Available at: https://cointelegraph.com/news/bitcoins-political-neutrality-is-a-myth-amir-taaki-interview [Accessed 8 Jul. 2017]. 2014
[40] K. Shirriff. "Bitcoin mining the hard way: the algorithms, protocols, and bytes", *Righto.com*, 2014. [Online]. Available: http://www.righto.com/2014/02/bitcoin-mining-hard-way-algorithms.html. [Accessed: 04- Aug- 2017].

12 Appendices

12.1 Appendix 1:

```
#define ATOMIC false     //not realistic, but do not force miners to mine on top of their own blocks
#define NOISE_IN_TRANSACTIONS false //miners don't put the max value they can into a block (simulate tx latency)

#define NETWORK_DELAY BlockTime(0)         //network delay in seconds for network to hear about your block
#define EXPECTED_NUMBER_OF_BLOCKS BlockCount(10000)

#define LAMBERT_COEFF 0.13533528323661//coeff for lambert func equil  must be in [0,.2]
//0.13533528323661 = 1/(e^2)

#define SEC_PER_BLOCK BlockRate(600)     //mean time in seconds to find a block

#define B BlockValue(0 * SATOSHI_PER_BITCOIN) // Block reward
#define A BlockValue(50 * SATOSHI_PER_BITCOIN)/SEC_PER_BLOCK  //rate transactions come in


struct RunSettings {

    unsigned int numberOfGames;
    MinerCount totalMiners;
    MinerCount fixedDefault;
    GameSettings gameSettings;
    std::string folderPrefix;

};

void runStratGame(RunSettings settings, std::vector<std::unique_ptr<LearningStrategy>> &learningStrategies, std::unique_ptr<Strategy>
    defaultStrategy);

void runSingleStratGame(RunSettings settings);

void runStratGame(RunSettings settings, std::vector<std::unique_ptr<LearningStrategy>> &learningStrategies, std::unique_ptr<Strategy>
    defaultStrategy) {

    //start running games
    //initally the total blocks mined are zero as is the number of blocks in the longest chain
    BlockCount totalBlocksMined(0);
    BlockCount blocksInLongestChain(0);

    //this is the ending of the result folder
    std::string resultFolder = "";

    //prefix an end to the file
    if (settings.folderPrefix.length() > 0) {
        resultFolder += settings.folderPrefix + "-";
    }

    resultFolder += std::to_string(rawCount(settings.fixedDefault));

    //create a vector for miners and learning miners
    std::vector<std::unique_ptr<Miner>> miners;
    std::vector<Miner *> learningMiners;

   //the hash rate is 1.0 divied by the number of total miners
   HashRate hashRate(1.0/rawCount(settings.totalMiners));
   //the number of miners is the total miners subtracted by the fixed default
   MinerCount numberRandomMiners(settings.totalMiners - settings.fixedDefault);

   //for each miner
   for (MinerCount i(0); i < settings.totalMiners; i++) {
       //give it a number
       auto minerName = std::to_string(rawCount(i));
       //each miner has a number, its name, its hash rate, the network latency and the cost
       MinerParameters parameters {rawCount(i), minerName, hashRate, NETWORK_DELAY, COST_PER_SEC_TO_MINE};

       //add to the miners vector each miner with their parameters and the default strategy
       miners.push_back(std::make_unique<Miner>(parameters, *defaultStrategy));

       //if the number of miners is less than the number of random miners
       if (i < numberRandomMiners) {

           //add the last element of the miners vector to the end of the learning miners
           //vector
           learningMiners.push_back(miners.back().get());
       }
   }

//    LearningModel *learningModel = new MultiplicativeWeightsLearningModel(learningStrategies, learningMiners.size(), resultFolder);

   //needed to run the learning models
   LearningModel *learningModel = new Exp3LearningModel(learningStrategies, learningMiners.size(), resultFolder);

   MinerGroup minerGroup(std::move(miners));

//    double phi = std::sqrt(strategies.size() * std::log(strategies.size())) / std::sqrt(settings.numberOfGames);
//    double phi = std::sqrt(strategies.size() * std::log(strategies.size())) / std::sqrt(settings.numberOfGames / 100);

   double phi = .01;

   //create the blockchain
   auto blockchain = std::make_unique<Blockchain>(settings.gameSettings.blockchainSettings);

   //for each round of the game
   for (unsigned int gameNum = 0; gameNum < settings.numberOfGames; gameNum++) {
//        double n = gameNum;
//        double nMax = settings.numberOfGames;
//        double phi = std::pow(.9, (n / nMax) * 30.0);

        //reset the settings
        blockchain->reset(settings.gameSettings.blockchainSettings);
        learningModel->writeWeights(gameNum);
        minerGroup.reset(*blockchain);
        learningModel->pickNewStrategies(phi, learningMiners, *blockchain);
        minerGroup.resetOrder();

        GAMEINFO("\n\nGame#: " << gameNum << " The board is set, the pieces are in motion..." << std::endl);
```

```
        //the result is from the runGame
        auto result = runGame(minerGroup, *blockchain, settings.gameSettings);

        GAMEINFO("The game is complete. Calculate the scores:" << std::endl);

        //the maximum profit is the transaction rate multiplied by the expected number of blocks
        // mulitplied by the seconds in which each block is discovered subtracted from the money left in the end which is then dividied by the
            number of miners
        Value maxProfit = (A * (EXPECTED_NUMBER_OF_BLOCKS * settings.gameSettings.blockchainSettings.secondsPerBlock) - result.moneyLeftAtEnd) /
            Value(rawCount(settings.totalMiners) / 4);

        // Steps 3, 4, 5
        learningModel->updateWeights(result, maxProfit, phi);

        totalBlocksMined += result.totalBlocksMined;
        blocksInLongestChain += result.blocksInLongestChain;

//        BlockCount staleBlocks(result.totalBlocksMined - result.blocksInLongestChain);
//        std::cout << result.moneyInLongestChain << " in chain and " <<
//        result.moneyLeftAtEnd << " left with " << 100 * double(rawCount(staleBlocks)) / double(rawCount(result.totalBlocksMined)) << "% orphan
    rate" <<  std::endl;

//      std::cout << totalBlocksMined << " total blocks mined" << std::endl;
    }
    learningModel->writeWeights(settings.numberOfGames);

    delete learningModel;
//    std::cout << 100 * double(rawCount(blocksInLongestChain)) / double(rawCount(totalBlocksMined)) << "% in final chain" << std::endl;

    GAMEINFOBLOCK(
                GAMEINFO("Games over. Final strategy weights:\n");
                learningModel->printWeights();
                )
}

void runSingleStratGame(RunSettings settings) {
    using std::placeholders::_1;
    using std::placeholders::_2;

    //create a vector dynamic array for storing the learning strategies
    std::vector<std::unique_ptr<LearningStrategy>> learningStrategies;

    //deafult weighting
    StratWeight defaultWeight(1);

    //create a default strategy which takes in arguments of whether the miner is atomic or
    //non atomic and the latency
    std::unique_ptr<Strategy> defaultStrategy(createDefaultStrategy(ATOMIC, NOISE_IN_TRANSACTIONS));

    learningStrategies.push_back(std::make_unique<LearningStrategy>(createDefaultStrategy(ATOMIC, NOISE_IN_TRANSACTIONS), defaultWeight));

    //add to the end of the learning strategies vector, a learning strategy called Petty strategy with
    //atomic and latency and default weight being the parameters
     learningStrategies.push_back(std::make_unique<LearningStrategy>(createPettyStrategy(ATOMIC, NOISE_IN_TRANSACTIONS), defaultWeight));

     //create the function strategies with differing values and add them to the end of the learning
     //strategies vector
     for (int i = -1; i < 3; i++) {
         int funcCoeff = static_cast<int>(pow(2, (i + 1)));
         std::function<Value(const Blockchain &, Value)> forkFunc(std::bind(functionForkPercentage, _1, _2, funcCoeff));
         learningStrategies.push_back(std::make_unique<LearningStrategy>(createFunctionForkStrategy(ATOMIC, forkFunc, std::to_string(funcCoeff)),
             defaultWeight));
     }

//  strategies.push_back(createFunctionForkStrategy(NO_SELF_MINING, std::bind(functionForkLambert, _1, _2, LAMBERT_COEFF), "lambert"));

    //create a strategy where the strategy is a Lazy fork strategy which takes in an argument of whether
    //the miners are atomic or not
    learningStrategies.push_back(std::make_unique<LearningStrategy>(createLazyForkStrategy(ATOMIC), defaultWeight));

    //run the game with settings, learning strategies and the default strategy
    runStratGame(settings, learningStrategies, std::move(defaultStrategy));
}

int main(int, const char * []) {

    //create a blockchain with a set amount of block reward, a set amount of transaction fee, the max number of blocks and the discovery rate of
        blocks
    BlockchainSettings blockchainSettings = {SEC_PER_BLOCK, A, B, EXPECTED_NUMBER_OF_BLOCKS};

    //create the simulation using the settings for the block chain above
    GameSettings gameSettings = {blockchainSettings};

//
//    for (MinerCount i(0); i < MinerCount(71); i += MinerCount(6)) {
//        RunSettings runSettings = {300000, MinerCount(100), i, gameSettings, "mult"};
//        runSingleStratGame(runSettings);
//    }

    //to run the simulation, decide the number of games, the number of miners, the settings of the simulation and where to output to
    RunSettings runSettings = {1000, MinerCount(200), MinerCount(0), gameSettings, "Simulation1"};

    runSingleStratGame(runSettings);

}
```

Amendments made to simulator from Simulator1.zip to make the simulator more accessible. This took more time than expected due to the nature of the simulator being in C++ and customisation being problematic.

12.2 Appendix 2:

```
//Hybrid Stubborn Strategy
public static Gain newLT(double a, double g){
        AbstractZeroHandler ltZeroHandler = new AbstractZeroHandler() {

                @Override
                public void zeroHandler(State init) {
                        double rand = StubbornMining.getRand();
                        if (rand < init.a) {
                                init.state = 1;
                        } else if (rand < init.a + init.g * (1 - init.a)) {
                                init.blocksGained += init.pending;
                                init.publicGained += 1;
                                init.pending = 0;
                                init.state = 0;
                                init.primeState = false;
                        } else {
                                init.state = -1;
                                init.primeState = false;
                        }
                }
        };

        Gain newLt = baseCode(a, g, ltZeroHandler, false);
        return newLt;
}
```

Hybrid stubborn strategy used in Simulator2.zip

12.3 Appendix 3

For these simulations, 10000000 simulations were run with the average taken at the end of the strategy completion. Different values of alpha, gamma were attempted . Values for alpha up to 0.5 are considered as over this, the fundamental security properties of the currency are broken as the attacker can guarantee she earns all the mining reward

Strategy									Relative Gain of Strategy X to Strategy Y = Gain of X - Gain of Y / alpha								
Alpha	Gamma	Honest Mining	Selfish Mining	Lead Stubborn	Equal Fork Stubborn	Trail Stubborn	Hybrid Stubborn	Winner	Hybrid compared to Honest	Selfish compared to Honest	Hybrid compared to Selfish	Lead Stubborn compared to Trail	Trail Stubborn compared to Lead	Lead Stubborn compared to Honest	Trail Stubborn compared to Honest	Lead Stubborn compared to selfish	Hybrid to Lead Stubborn	Hybrid to Trail Stubborn
0.1	0	10.00%	3.56%	1.91%	2.20%	3.54%	0.00%	Honest Mining	-100.00%	-64.40%	-35.60%	-16.30%	16.30%	-80.90%	-64.60%	-16.50%	-19.10%	-35.40%
0.15	0	15.00%	7.63%	4.21%	5.17%	7.57%	0.00%	Honest Mining	-100.00%	-49.13%	-50.87%	-22.40%	22.40%	-71.93%	-49.53%	-22.80%	-28.07%	-50.47%
0.2	0	20.00%	12.97%	7.38%	9.59%	12.92%	0.00%	Honest Mining	-100.00%	-35.15%	-64.85%	-27.70%	27.70%	-63.10%	-35.40%	-27.95%	-36.90%	-64.60%
0.25	0	25.00%	19.51%	11.38%	15.59%	19.62%	0.00%	Honest Mining	-100.00%	-21.96%	-78.04%	-32.96%	32.96%	-54.48%	-21.52%	-32.52%	-45.52%	-78.48%
0.3	0	30.00%	27.31%	16.24%	23.48%	27.60%	0.00%	Honest Mining	-100.00%	-8.97%	-91.03%	-37.87%	37.87%	-45.87%	-8.00%	-36.90%	-54.13%	-92.00%
0.35	0	35.00%	36.65%	22.12%	33.44%	37.11%	0.00%	Trail Stubborn	-100.00%	4.71%	-104.71%	-42.83%	42.83%	-36.80%	6.03%	-41.51%	-63.20%	-106.03%
0.4	0	40.00%	48.37%	29.01%	46.70%	48.81%	0.00%	Trail Stubborn	-100.00%	20.93%	-120.93%	-49.50%	49.50%	-27.48%	22.03%	-48.40%	-72.53%	-122.03%
0.45	0	45.00%	65.18%	38.00%	65.56%	65.24%	0.00%	Equal Fork Stubborn	-100.00%	44.84%	-144.84%	-60.53%	60.53%	-15.56%	44.98%	-60.40%	-84.44%	-144.98%
0.5	0	50.00%	100.00%	47.06%	93.25%	94.57%	0.00%	Selfish Mining	-100.00%	100.00%	-200.00%	-95.02%	95.02%	-5.88%	89.14%	-105.88%	-94.12%	-189.14%
		Honest Mining	Selfish Mining	Lead Stubborn	Equal Fork Stubborn	Trail Stubborn	Hybrid Stubborn	Winner									
0.1	0.1	10.00%	4.30%	2.92%	3.08%	4.21%	1.18%	Honest Mining	-88.20%	-57.00%	-31.20%	-12.90%	12.90%	-70.80%	-57.90%	-13.80%	-17.40%	-30.30%
0.15	0.1	15.00%	8.56%	5.81%	6.37%	8.45%	2.02%	Honest Mining	-86.53%	-42.93%	-43.60%	-17.60%	17.60%	-61.27%	-43.67%	-18.33%	-25.27%	-42.87%
0.2	0.1	20.00%	14.02%	9.69%	11.02%	13.91%	3.23%	Honest Mining	-83.85%	-29.90%	-53.95%	-21.10%	21.10%	-51.55%	-30.45%	-21.65%	-32.30%	-53.40%
0.25	0.1	25.00%	20.61%	14.62%	17.19%	20.55%	5.02%	Honest Mining	-79.92%	-17.56%	-62.36%	-23.72%	23.72%	-41.52%	-17.80%	-23.96%	-38.40%	-62.12%
0.3	0.1	30.00%	28.39%	20.93%	25.10%	28.51%	8.07%	Honest Mining	-73.10%	-5.37%	-67.73%	-25.27%	25.27%	-30.23%	-4.97%	-24.87%	-42.87%	-68.13%
0.35	0.1	35.00%	37.64%	29.25%	35.02%	37.96%	13.40%	Trail Stubborn	-61.71%	7.54%	-69.26%	-24.89%	24.89%	-16.43%	8.46%	-23.97%	-45.29%	-70.17%
0.4	0.1	40.00%	49.21%	41.10%	47.94%	49.49%	23.96%	Trail Stubborn	-40.10%	23.03%	-63.13%	-20.98%	20.98%	2.75%	23.73%	-20.28%	-42.85%	-63.83%
0.45	0.1	45.00%	65.75%	60.53%	66.28%	65.79%	46.68%	Equal Fork Stubborn	3.73%	46.11%	-42.38%	-11.69%	11.69%	34.51%	46.20%	-11.60%	-30.78%	-42.47%
0.5	0.1	50.00%	100.00%	91.09%	95.61%	95.25%	89.05%	Selfish Mining	78.10%	100.00%	-21.90%	-8.32%	8.32%	82.18%	90.50%	-17.82%	-4.08%	-12.40%
		Honest Mining	Selfish Mining	Lead Stubborn	Equal Fork Stubborn	Trail Stubborn	Hybrid Stubborn	Winner									
0.1	0.2	10.00%	5.04%	3.92%	3.95%	4.90%	2.32%	Honest Mining	-76.80%	-49.60%	-27.20%	-9.80%	9.80%	-60.80%	-51.00%	-11.20%	-16.00%	-25.80%
0.15	0.2	15.00%	9.50%	7.31%	7.58%	9.32%	3.95%	Honest Mining	-73.67%	-36.67%	-37.00%	-13.40%	13.40%	-51.27%	-37.87%	-14.60%	-22.40%	-35.80%
0.2	0.2	20.00%	15.08%	11.74%	12.46%	14.88%	6.14%	Honest Mining	-69.30%	-24.60%	-44.70%	-15.70%	15.70%	-41.30%	-25.60%	-16.70%	-28.00%	-43.70%
0.25	0.2	25.00%	21.71%	17.38%	18.77%	21.50%	9.22%	Honest Mining	-63.12%	-13.16%	-49.96%	-16.48%	16.48%	-30.48%	-14.00%	-17.32%	-32.64%	-49.12%
0.3	0.2	30.00%	29.46%	24.46%	26.64%	29.49%	13.92%	Honest Mining	-53.60%	-1.80%	-51.80%	-16.77%	16.77%	-18.47%	-1.70%	-16.67%	-35.13%	-51.90%
0.35	0.2	35.00%	38.63%	33.70%	36.52%	38.78%	21.44%	Trail Stubborn	-38.74%	10.37%	-49.11%	-14.51%	14.51%	-3.71%	10.80%	-14.09%	-35.03%	-49.54%
0.4	0.2	40.00%	50.05%	46.42%	49.23%	50.30%	33.98%	Trail Stubborn	-15.05%	25.13%	-40.18%	-9.70%	9.70%	16.05%	25.75%	-9.07%	-31.10%	-40.80%
0.45	0.2	45.00%	66.33%	65.83%	67.05%	66.55%	56.72%	Equal Fork Stubborn	26.04%	47.40%	-21.36%	-1.60%	1.60%	46.29%	47.89%	-1.11%	-20.24%	-21.84%
0.5	0.2	50.00%	100.00%	94.83%	94.39%	94.02%	94.51%	Selfish Mining	89.02%	100.00%	-10.98%	1.62%	-1.62%	89.66%	88.04%	-10.34%	-0.64%	0.98%
		Honest Mining	Selfish Mining	Lead Stubborn	Equal Fork Stubborn	Trail Stubborn	Hybrid Stubborn	Winner									
0.1	0.3	10.00%	5.77%	4.88%	4.84%	5.62%	3.47%	Honest Mining	-65.30%	-42.30%	-23.00%	-7.40%	7.40%	-51.20%	-43.80%	-8.90%	-14.10%	-21.50%
0.15	0.3	15.00%	10.44%	8.76%	8.81%	10.20%	5.78%	Honest Mining	-61.47%	-30.40%	-31.07%	-9.60%	9.60%	-41.60%	-32.00%	-11.20%	-19.87%	-29.47%
0.2	0.3	20.00%	16.13%	13.65%	13.93%	15.89%	8.78%	Honest Mining	-56.10%	-19.35%	-36.75%	-11.20%	11.20%	-31.75%	-20.55%	-12.40%	-24.35%	-35.55%
0.25	0.3	25.00%	22.80%	19.75%	20.34%	22.59%	12.88%	Honest Mining	-48.48%	-8.80%	-39.68%	-11.36%	11.36%	-21.00%	-9.64%	-12.20%	-27.48%	-38.84%
0.3	0.3	30.00%	30.54%	27.38%	28.27%	30.45%	18.73%	Selfish Mining	-37.57%	1.80%	-39.37%	-10.23%	10.23%	-8.73%	1.50%	-10.53%	-28.83%	-39.07%
0.35	0.3	35.00%	39.62%	37.10%	38.07%	39.70%	27.37%	Trail Stubborn	-21.80%	13.20%	-35.00%	-7.43%	7.43%	6.00%	13.43%	-7.20%	-27.80%	-35.23%
0.4	0.3	40.00%	50.88%	49.88%	50.46%	51.07%	40.69%	Trail Stubborn	1.72%	27.20%	-25.48%	-2.98%	2.98%	24.70%	27.68%	-2.50%	-22.98%	-25.95%
0.45	0.3	45.00%	66.91%	68.58%	67.98%	66.93%	62.21%	Lead Stubborn	38.24%	48.69%	-10.44%	3.67%	-3.67%	52.40%	48.73%	3.71%	-14.16%	-10.49%
0.5	0.3	50.00%	100.00%	95.45%	96.09%	93.86%	95.28%	Selfish Mining	90.56%	100.00%	-9.44%	3.18%	-3.18%	90.90%	87.72%	-9.10%	-0.34%	2.84%
		Honest Mining	Selfish Mining	Lead Stubborn	Equal Fork Stubborn	Trail Stubborn	Hybrid Stubborn	Winner									
0.1	0.4	10.00%	6.51%	5.85%	5.73%	6.34%	4.58%	Honest Mining	-54.20%	-34.90%	-19.30%	-4.90%	4.90%	-41.50%	-36.60%	-6.60%	-12.70%	-17.60%
0.15	0.4	15.00%	11.38%	10.13%	9.99%	11.12%	7.58%	Honest Mining	-49.47%	-24.13%	-25.33%	-6.60%	6.60%	-32.47%	-25.87%	-8.33%	-17.00%	-23.60%
0.2	0.4	20.00%	17.19%	15.42%	15.37%	16.86%	11.30%	Honest Mining	-43.50%	-14.05%	-29.45%	-7.20%	7.20%	-22.90%	-15.70%	-8.85%	-20.60%	-27.80%
0.25	0.4	25.00%	23.90%	21.94%	21.93%	23.64%	16.23%	Honest Mining	-35.08%	-4.40%	-30.68%	-6.80%	6.80%	-12.24%	-5.44%	-7.84%	-22.84%	-29.64%
0.3	0.4	30.00%	31.61%	29.87%	29.85%	31.48%	22.84%	Selfish Mining	-23.87%	5.37%	-29.23%	-5.37%	5.37%	-0.43%	4.93%	-5.80%	-23.43%	-28.80%
0.35	0.4	35.00%	40.61%	39.69%	39.55%	40.57%	32.08%	Selfish Mining	-8.34%	16.03%	-24.37%	-2.51%	2.51%	13.40%	15.91%	-2.63%	-21.74%	-24.26%
0.4	0.4	40.00%	51.72%	52.56%	51.78%	51.82%	45.65%	Lead Stubborn	14.13%	29.30%	-15.18%	1.85%	-1.85%	31.40%	29.55%	2.10%	-17.28%	-15.43%
0.45	0.4	45.00%	67.48%	70.59%	68.73%	67.50%	66.28%	Lead Stubborn	47.29%	49.96%	-2.67%	6.87%	-6.87%	56.87%	50.00%	6.91%	-9.58%	-2.71%
0.5	0.4	50.00%	100.00%	95.39%	95.02%	94.28%	94.25%	Selfish Mining	88.50%	100.00%	-11.50%	2.22%	-2.22%	90.78%	88.56%	-9.22%	-2.28%	-0.06%
									-7.17%									
		Honest Mining	Selfish Mining	Lead Stubborn	Equal Fork Stubborn	Trail Stubborn	Hybrid Stubborn	Winner									
0.1	0.5	10.00%	7.24%	6.76%	6.60%	7.09%	5.69%	Honest Mining	-43.10%	-27.60%	-15.50%	-3.30%	3.30%	-32.40%	-29.10%	-4.80%	-10.70%	-14.00%
0.15	0.5	15.00%	12.32%	11.44%	11.25%	12.06%	9.29%	Honest Mining	-38.07%	-17.87%	-20.20%	-4.13%	4.13%	-23.73%	-19.60%	-5.87%	-14.33%	-18.47%
0.2	0.5	20.00%	18.24%	17.12%	16.83%	17.92%	13.70%	Honest Mining	-31.50%	-8.80%	-22.70%	-4.00%	4.00%	-14.40%	-10.40%	-5.60%	-17.10%	-21.10%
0.25	0.5	25.00%	25.00%	23.87%	23.53%	24.68%	19.27%	Honest Mining	-22.92%	0.00%	-22.92%	-3.24%	3.24%	-4.52%	-1.28%	-4.52%	-18.40%	-21.64%
0.3	0.5	30.00%	32.69%	31.97%	31.53%	32.46%	26.63%	Selfish Mining	-11.23%	8.97%	-20.20%	-1.63%	1.63%	6.57%	8.20%	-2.40%	-17.80%	-19.43%
0.35	0.5	35.00%	41.60%	42.00%	41.05%	41.53%	36.26%	Lead Stubborn	3.60%	18.86%	-15.26%	1.34%	-1.34%	20.00%	18.66%	1.14%	-16.40%	-15.06%
0.4	0.5	40.00%	52.56%	54.72%	53.06%	52.50%	49.80%	Lead Stubborn	24.50%	31.40%	-6.90%	5.55%	-5.55%	36.80%	31.25%	5.40%	-12.30%	-6.75%
0.45	0.5	45.00%	68.06%	72.17%	69.59%	68.00%	69.67%	Lead Stubborn	54.82%	51.24%	3.58%	9.27%	-9.27%	60.38%	51.11%	9.13%	-5.56%	3.71%
0.5	0.5	50.00%	100.00%	93.84%	94.19%	93.92%	93.52%	Selfish Mining	87.04%	100.00%	-12.96%	-0.16%	0.16%	87.68%	87.84%	-12.32%	-0.64%	-0.80%
									2.57%		-14.78%							
		Honest Mining	Selfish Mining	Lead Stubborn	Equal Fork Stubborn	Trail Stubborn	Hybrid Stubborn	Winner									
0.1	0.6	10.00%	7.98%	7.64%	7.48%	7.83%	6.77%	Honest Mining	-32.30%	-20.20%	-12.10%	-1.90%	1.90%	-23.60%	-21.70%	-3.40%	-8.70%	-10.60%
0.15	0.6	15.00%	13.26%	12.72%	12.43%	13.00%	10.98%	Honest Mining	-26.80%	-11.60%	-15.20%	-1.87%	1.87%	-15.20%	-13.33%	-3.60%	-11.60%	-13.47%
0.2	0.6	20.00%	19.30%	18.70%	18.26%	18.97%	16.02%	Honest Mining	-19.90%	-3.50%	-16.40%	-1.35%	1.35%	-6.50%	-5.15%	-3.00%	-13.40%	-14.75%
0.25	0.6	25.00%	26.10%	25.71%	25.10%	25.78%	22.27%	Selfish Mining	-10.92%	4.40%	-15.32%	-0.28%	0.28%	2.84%	3.12%	-1.56%	-13.76%	-14.04%
0.3	0.6	30.00%	33.76%	34.00%	33.11%	33.58%	29.98%	Lead Stubborn	-0.07%	12.53%	-12.60%	1.40%	-1.40%	13.33%	11.93%	0.80%	-13.40%	-12.00%
0.35	0.6	35.00%	42.59%	43.96%	42.53%	42.49%	40.14%	Lead Stubborn	14.69%	21.69%	-7.00%	4.20%	-4.20%	25.60%	21.40%	3.91%	-10.91%	-6.71%
0.4	0.6	40.00%	53.40%	56.50%	54.39%	53.39%	53.70%	Lead Stubborn	34.25%	33.50%	0.75%	7.77%	-7.77%	41.25%	33.48%	7.75%	-7.00%	0.77%
0.45	0.6	45.00%	68.64%	73.51%	70.53%	68.60%	72.44%	Lead Stubborn	60.98%	52.53%	8.44%	10.91%	-10.91%	63.36%	52.44%	10.82%	-2.38%	8.53%
0.5	0.6	50.00%	100.00%	96.35%	95.36%	95.08%	96.05%	Selfish Mining	92.10%	100.00%	-7.90%	2.54%	-2.54%	92.70%	90.16%	-7.30%	-0.60%	1.94%
									12.45%		-8.59%							
		Honest Mining	Selfish Mining	Lead Stubborn	Equal Fork Stubborn	Trail Stubborn	Hybrid Stubborn	Winner									
0.1	0.7	10.00%	8.71%	8.52%	8.37%	8.55%	7.86%	Honest Mining	-21.40%	-12.90%	-8.50%	-0.30%	0.30%	-14.80%	-14.50%	-1.90%	-6.60%	-6.90%
0.15	0.7	15.00%	14.20%	13.91%	13.62%	13.98%	12.63%	Honest Mining	-15.80%	-5.33%	-10.47%	-0.47%	0.47%	-7.27%	-6.80%	-1.93%	-8.53%	-9.00%
0.2	0.7	20.00%	20.35%	20.13%	19.72%	20.07%	18.31%	Selfish Mining	-8.45%	1.75%	-10.20%	0.30%	-0.30%	0.65%	0.35%	-1.10%	-9.10%	-8.80%
0.25	0.7	25.00%	27.20%	27.40%	26.69%	26.92%	25.06%	Lead Stubborn	0.24%	8.80%	-8.56%	1.92%	-1.92%	9.60%	7.68%	0.80%	-9.36%	-7.44%
0.3	0.7	30.00%	34.84%	35.74%	34.66%	34.60%	33.33%	Lead Stubborn	11.10%	16.13%	-5.03%	3.80%	-3.80%	19.13%	15.33%	3.00%	-8.03%	-4.23%
0.35	0.7	35.00%	43.58%	45.76%	44.10%	43.44%	43.65%	Lead Stubborn	24.71%	24.51%	0.20%	6.63%	-6.63%	30.74%	24.11%	6.23%	-6.03%	0.60%
0.4	0.7	40.00%	54.23%	58.08%	55.62%	54.21%	57.12%	Lead Stubborn	42.80%	35.58%	7.23%	9.67%	-9.67%	45.20%	35.53%	9.62%	-2.40%	7.28%
0.45	0.7	45.00%	69.21%	74.61%	71.37%	69.09%	74.96%	Hybrid Stubborn	66.58%	53.80%	12.78%	12.27%	-12.27%	65.80%	53.53%	12.00%	0.78%	13.04%
0.5	0.7	50.00%	100.00%	96.90%	95.00%	94.40%	96.46%	Selfish Mining	92.92%	100.00%	-7.08%	5.00%	-5.00%	93.80%	88.80%	-6.20%	-0.88%	4.12%
									21.41%		-3.29%						-5.57%	-1.26%
		Honest Mining	Selfish Mining	Lead Stubborn	Equal Fork Stubborn	Trail Stubborn	Hybrid Stubborn	Winner									
0.1	0.8	10.00%	9.45%	9.36%	9.25%	9.36%	8.94%	Honest Mining	-10.60%	-5.50%	-5.10%	0.00%	0.00%	-6.40%	-6.40%	-0.90%	-4.20%	-4.20%
0.15	0.8	15.00%	15.13%	15.07%	14.83%	14.94%	14.31%	Selfish Mining	-4.60%	0.87%	-5.47%	0.87%	-0.87%	0.47%	-0.40%	-0.40%	-5.07%	-4.20%
0.2	0.8	20.00%	21.41%	21.55%	21.17%	21.18%	20.55%	Lead Stubborn	2.75%	7.05%	-4.30%	1.85%	-1.85%	7.75%	5.90%	0.70%	-5.00%	-3.15%
0.25	0.8	25.00%	28.29%	28.96%	28.24%	28.07%	27.80%	Lead Stubborn	11.20%	13.16%	-1.96%	3.56%	-3.56%	15.84%	12.28%	2.68%	-4.64%	-1.08%
0.3	0.8	30.00%	35.91%	37.36%	36.30%	35.72%	36.56%	Lead Stubborn	21.87%	19.70%	2.17%	5.47%	-5.47%	24.53%	19.07%	4.83%	-2.67%	2.80%
0.35	0.8	35.00%	44.57%	47.33%	45.61%	44.48%	47.14%	Lead Stubborn	34.69%	27.34%	7.34%	8.14%	-8.14%	35.23%	27.09%	7.89%	-0.54%	7.60%
0.4	0.8	40.00%	55.07%	59.38%	56.94%	55.09%	60.35%	Hybrid Stubborn	50.88%	37.68%	13.20%	10.73%	-10.73%	48.45%	37.73%	10.78%	2.43%	13.15%
0.45	0.8	45.00%	69.79%	75.65%	72.20%	69.89%	77.38%	Hybrid Stubborn	71.96%	55.09%	16.87%	12.60%	-12.60%	67.91%	55.31%	12.82%	4.04%	16.64%
0.5	0.8	50.00%	100.00%	97.40%	94.95%	95.97%	95.54%	Selfish Mining	91.08%	100.00%	-8.92%	2.86%	-2.86%	94.80%	91.94%	-5.20%	-3.72%	-0.86%
									29.91%		1.54%						-2.15%	2.97%
		Honest Mining	Selfish Mining	Lead Stubborn	Equal Fork Stubborn	Trail Stubborn	Hybrid Stubborn	Winner									
0.1	0.9	10.00%	10.18%	10.18%	10.12%	10.13%	10.03%	Selfish Mining	0.30%	1.80%	-1.50%	0.50%	-0.50%	1.80%	1.30%	0.00%	-1.50%	-1.00%
0.15	0.9	15.00%	16.07%	16.23%	16.06%	15.95%	16.00%	Lead Stubborn	6.67%	7.13%	-0.47%	1.87%	-1.87%	8.20%	6.33%	1.07%	-1.53%	0.33%
0.2	0.9	20.00%	22.46%	22.91%	22.59%	22.36%	22.78%	Lead Stubborn	13.90%	12.30%	1.60%	2.75%	-2.75%	14.55%	11.80%	2.25%	-0.65%	2.10%
0.25	0.9	25.00%	29.39%	30.40%	29.85%	29.28%	30.57%	Hybrid Stubborn	22.28%	17.56%	4.72%	4.48%	-4.48%	21.60%	17.12%	4.04%	0.68%	5.16%
0.3	0.9	30.00%	36.99%	38.93%	37.87%	36.84%	39.65%	Hybrid Stubborn	32.17%	23.30%	8.87%	6.97%	-6.97%	29.77%	22.80%	6.47%	2.40%	9.37%
0.35	0.9	35.00%	45.57%	48.73%	47.06%	45.53%	50.53%	Hybrid Stubborn	44.37%	30.20%	14.17%	9.14%	-9.14%	39.23%	30.09%	9.03%	5.14%	14.29%
0.4	0.9	40.00%	55.91%	60.68%	58.17%	55.86%	63.53%	Hybrid Stubborn	58.83%	39.78%	19.05%	12.05%	-12.05%	51.70%	39.65%	11.93%	7.12%	19.18%
0.45	0.9	45.00%	70.37%	76.36%	73.13%	70.40%	79.70%	Hybrid Stubborn	77.11%	56.38%	20.73%	13.24%	-13.24%	69.69%	56.44%	13.31%	7.42%	20.67%
0.5	0.9	50.00%	100.00%	96.38%	95.86%	94.55%	96.70%	Selfish Mining	93.40%	100.00%	-6.60%	3.66%	-3.66%	92.76%	89.10%	-7.24%	0.64%	4.30%
									38.78%		6.73%						2.19%	8.27%
		Honest Mining	Selfish Mining	Lead Stubborn	Equal Fork Stubborn	Trail Stubborn	Hybrid Stubborn	Winner									
0.1	1	10.00%	10.92%	11.02%	11.00%	10.92%	11.12%	Hybrid Stubborn	11.20%	9.20%	2.00%	1.00%	-1.00%	10.20%	9.20%	1.00%	1.00%	2.00%
0.15	1	15.00%	17.01%	17.31%	17.29%	16.99%	17.64%	Hybrid Stubborn	17.60%	13.40%	4.20%	2.13%	-2.13%	15.40%	13.27%	2.00%	2.20%	4.33%
0.2	1	20.00%	23.52%	24.20%	24.08%	23.53%	24.98%	Hybrid Stubborn	24.90%	17.60%	7.30%	3.35%	-3.35%	21.00%	17.65%	3.40%	3.90%	7.25%
0.25	1	25.00%	30.49%	31.83%	31.43%	30.49%	33.34%	Hybrid Stubborn	33.36%	21.96%	11.40%	5.36%	-5.36%	27.32%	21.96%	5.36%	6.04%	11.40%
0.3	1	30.00%	38.06%	40.33%	39.48%	38.04%	42.89%	Hybrid Stubborn	42.97%	26.87%	16.10%	7.63%	-7.63%	34.43%	26.80%	7.57%	8.53%	16.17%
0.35	1	35.00%	46.56%	50.06%	48.66%	46.62%	53.82%	Hybrid Stubborn	53.77%	33.03%	20.74%	9.83%	-9.83%	43.03%	33.20%	10.00%	10.74%	20.57%
0.4	1	40.00%	56.74%	61.77%	59.47%	56.77%	66.72%	Hybrid Stubborn	66.80%	41.85%	24.95%	12.50%	-12.50%	54.43%	41.93%	12.58%	12.38%	24.88%
0.45	1	45.00%	70.94%	77.09%	74.02%	70.92%	81.82%	Hybrid Stubborn	81.82%	57.64%	24.18%	13.71%	-13.71%	71.31%	57.60%	13.67%	10.51%	24.22%
0.5	1	50.00%	100.00%	96.86%	94.87%	95.59%	97.46%	Selfish Mining	94.92%	100.00%	-5.08%	2.54%	-2.54%	93.72%	91.18%	-6.28%	1.20%	3.74%
									47.48%		11.75%						6.28%	12.73%

A selection of findings from using Simulator2.zip. The colour coded calculations relate to relative gains against different strategies.

For these simulations, 10000000 simulations were run with the average taken at the end of the strategy completion. A blockchain limit was imposed of 100 to ensure that the simulation does not get out of hand. Different values of alpha, gamma were attempted

Hybrid Stubborn

alpha / gamma	0	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
0	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
0.1	0.00%	1.18%	2.02%	3.23%	5.02%	8.07%	13.40%	23.96%	46.68%	89.05%
0.2	0.00%	2.32%	3.95%	6.14%	9.22%	13.92%	21.44%	33.98%	56.72%	94.51%
0.3	0.00%	3.47%	5.78%	8.78%	12.88%	18.73%	27.37%	40.69%	62.21%	95.28%
0.4	0.00%	4.58%	7.58%	11.30%	16.23%	22.84%	32.08%	45.65%	66.28%	94.25%
0.5	0.00%	5.69%	9.29%	13.70%	19.27%	26.63%	36.26%	49.80%	69.67%	93.52%
0.6	0.00%	6.77%	10.98%	16.02%	22.27%	29.98%	40.14%	53.70%	72.44%	96.05%
0.7	0.00%	7.86%	12.63%	18.31%	25.06%	33.33%	43.65%	57.12%	74.96%	96.46%
0.8	0.00%	8.94%	14.31%	20.55%	27.80%	36.56%	47.14%	60.35%	77.38%	95.54%
0.9	0.00%	10.03%	16.00%	22.78%	30.57%	39.65%	50.53%	63.53%	79.70%	96.70%
1	0.00%	11.12%	17.64%	24.98%	33.34%	42.89%	53.82%	66.72%	81.82%	97.46%
1st	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Last	0.00%	11.12%	17.64%	24.98%	33.34%	42.89%	53.82%	66.72%	81.82%	97.46%
% Change	0.00%	11.12%	17.64%	24.98%	33.34%	42.89%	53.82%	66.72%	81.82%	97.46%

Lead Stubborn

alpha / gamma	0	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
0	0.00%	1.91%	4.21%	7.38%	11.38%	16.24%	22.12%	29.01%	38.00%	47.06%
0.1	0.00%	2.92%	5.81%	9.69%	14.62%	20.93%	29.25%	41.10%	60.53%	91.09%
0.2	0.00%	3.92%	7.31%	11.74%	17.38%	24.46%	33.70%	46.42%	65.83%	94.83%
0.3	0.00%	4.88%	8.76%	13.65%	19.75%	27.38%	37.10%	49.88%	68.58%	95.45%
0.4	0.00%	5.85%	10.13%	15.42%	21.94%	29.87%	39.69%	52.56%	70.59%	95.39%
0.5	0.00%	6.76%	11.44%	17.12%	23.87%	31.97%	42.00%	54.72%	72.17%	93.84%
0.6	0.00%	7.64%	12.72%	18.70%	25.71%	34.00%	43.96%	56.50%	73.51%	96.35%
0.7	0.00%	8.52%	13.91%	20.13%	27.40%	35.74%	45.76%	58.08%	74.61%	96.90%
0.8	0.00%	9.36%	15.07%	21.55%	28.96%	37.36%	47.33%	59.38%	75.56%	97.40%
0.9	0.00%	10.18%	16.23%	22.91%	30.40%	38.93%	48.73%	60.68%	76.36%	96.38%
1	0.00%	11.02%	17.31%	24.20%	31.83%	40.33%	50.06%	61.77%	77.09%	96.86%
1st	0.00%	1.91%	4.21%	7.38%	11.38%	16.24%	22.12%	29.01%	38.00%	47.06%
Last	0.00%	11.02%	17.31%	24.20%	31.83%	40.33%	50.06%	61.77%	77.09%	96.86%
% Change	0.00%	9.11%	13.10%	16.82%	20.45%	24.09%	27.94%	32.76%	39.09%	49.80%

Trail Stubborn

alpha / gamma	0	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
0	0.00%	3.54%	7.57%	12.92%	19.62%	27.60%	37.11%	48.81%	65.24%	94.57%
0.1	0.00%	4.21%	8.45%	13.91%	20.55%	28.51%	37.96%	49.49%	65.79%	95.25%
0.2	0.00%	4.90%	9.32%	14.88%	21.50%	29.49%	38.78%	50.30%	66.55%	94.02%
0.3	0.00%	5.62%	10.20%	15.89%	22.59%	30.45%	39.70%	51.07%	66.93%	93.86%
0.4	0.00%	6.34%	11.12%	16.86%	23.64%	31.48%	40.57%	51.82%	67.50%	94.28%
0.5	0.00%	7.09%	12.06%	17.92%	24.68%	32.46%	41.53%	52.50%	68.00%	93.92%
0.6	0.00%	7.83%	13.00%	18.97%	25.78%	33.58%	42.49%	53.39%	68.60%	95.08%
0.7	0.00%	8.55%	13.98%	20.07%	26.92%	34.60%	43.44%	54.21%	69.09%	94.40%
0.8	0.00%	9.36%	14.94%	21.18%	28.07%	35.72%	44.48%	55.09%	69.89%	95.97%
0.9	0.00%	10.13%	15.95%	22.36%	29.28%	36.84%	45.53%	55.86%	70.40%	94.55%
1	0.00%	10.92%	16.99%	23.53%	30.49%	38.04%	46.62%	56.77%	70.92%	95.59%
1st	0.00%	3.54%	7.57%	12.92%	19.62%	27.60%	37.11%	48.81%	65.24%	94.57%
Last	0.00%	10.92%	16.99%	23.53%	30.49%	38.04%	46.62%	56.77%	70.92%	95.59%
% Change	0.00%	7.38%	9.42%	10.61%	10.87%	10.44%	9.51%	7.96%	5.68%	1.02%

Equal Fork Stubborn

alpha / gamma	0	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
0	0.00%	3.54%	7.57%	12.92%	19.62%	27.60%	37.11%	48.81%	65.24%	94.57%
0.1	0.00%	4.21%	8.45%	13.91%	20.55%	28.51%	37.96%	49.49%	65.79%	95.25%
0.2	0.00%	4.90%	9.32%	14.88%	21.50%	29.49%	38.78%	50.30%	66.55%	94.02%
0.3	0.00%	5.62%	10.20%	15.89%	22.59%	30.45%	39.70%	51.07%	66.93%	93.86%
0.4	0.00%	6.34%	11.12%	16.86%	23.64%	31.48%	40.57%	51.82%	67.50%	94.28%
0.5	0.00%	7.09%	12.06%	17.92%	24.68%	32.46%	41.53%	52.50%	68.00%	93.92%
0.6	0.00%	7.83%	13.00%	18.97%	25.78%	33.58%	42.49%	53.39%	68.60%	95.08%
0.7	0.00%	8.55%	13.98%	20.07%	26.92%	34.60%	43.44%	54.21%	69.09%	94.40%
0.8	0.00%	9.36%	14.94%	21.18%	28.07%	35.72%	44.48%	55.09%	69.89%	95.97%
0.9	0.00%	10.13%	15.95%	22.36%	29.28%	36.84%	45.53%	55.86%	70.40%	94.55%
1	0.00%	10.92%	16.99%	23.53%	30.49%	38.04%	46.62%	56.77%	70.92%	95.59%
1st	0.00%	3.54%	7.57%	12.92%	19.62%	27.60%	37.11%	48.81%	65.24%	94.57%
Last	0.00%	10.92%	16.99%	23.53%	30.49%	38.04%	46.62%	56.77%	70.92%	95.59%
% Change	0.00%	7.38%	9.42%	10.61%	10.87%	10.44%	9.51%	7.96%	5.68%	1.02%

Selfish Mining

alpha / gamma	0	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
0	0.00%	3.56%	7.63%	12.97%	19.51%	27.31%	36.65%	48.37%	65.18%	100.00%
0.1	0.00%	4.30%	8.56%	14.02%	20.61%	28.39%	37.64%	49.21%	65.75%	100.00%
0.2	0.00%	5.04%	9.50%	15.08%	21.71%	29.46%	38.63%	50.05%	66.33%	100.00%
0.3	0.00%	5.77%	10.44%	16.13%	22.80%	30.54%	39.62%	50.88%	66.91%	100.00%
0.4	0.00%	6.51%	11.38%	17.19%	23.90%	31.61%	40.61%	51.72%	67.48%	100.00%
0.5	0.00%	7.24%	12.32%	18.24%	25.00%	32.69%	41.60%	52.56%	68.06%	100.00%
0.6	0.00%	7.98%	13.26%	19.30%	26.10%	33.76%	42.59%	53.40%	68.64%	100.00%
0.7	0.00%	8.71%	14.20%	20.35%	27.20%	34.84%	43.58%	54.23%	69.21%	100.00%
0.8	0.00%	9.45%	15.13%	21.41%	28.29%	35.91%	44.57%	55.07%	69.79%	100.00%
0.9	0.00%	10.18%	16.07%	22.46%	29.39%	36.99%	45.57%	55.91%	70.37%	100.00%
1	0.00%	10.92%	17.01%	23.52%	30.49%	38.06%	46.56%	56.74%	70.94%	100.00%
1st	0.00%	3.56%	7.63%	12.97%	19.51%	27.31%	36.65%	48.37%	65.18%	100.00%
Last	0.00%	10.92%	17.01%	23.52%	30.49%	38.06%	46.56%	56.74%	70.94%	100.00%
% Change	0.00%	7.36%	9.38%	10.55%	10.98%	10.75%	9.91%	8.37%	5.76%	0.00%

Default Mining

alpha	0	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Outcome	0.00%	10.00%	15.00%	20.00%	25.00%	30.00%	35.00%	40.00%	45.00%	50.00%

A selection of findings from using Simulator2.zip. Layout adopted to make comparison easier.