

A NOVEL PENALTY SYSTEM TO LIMIT PROFITABILITY OF SELFISH MINING

---

A Thesis

presented to

the Faculty of Natural and Applied Sciences

at Notre Dame University-Louaize

---

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

---

by

GEORGIO SEMAAN

MAY 2021

© COPYRIGHT

By

Georgio Semaan

2021


All Rights Reserved


Notre Dame University - Louaize  
Faculty of Natural and Applied Sciences  
Department of Computer Science

We hereby approve the thesis of

Georgio Semaan

Candidate for the degree of Master of Science in Computer Science

  
Dr. Hoda Maalouf Supervisor, Chair

  
Dr. Nazir Hawi Committee Member

## **Declaration**

I, hereby declare that this submission is entirely my own work, in my own words, and that all sources used in researching it are fully acknowledged and all quotations properly identified. It has not been submitted, in whole or in part, by me or another person, for the purpose of obtaining any other credit. I understand the ethical implications of my research, and this work meets the requirements of the Faculty of Natural and Applied Sciences Committee at Notre Dame University-Louaize.

**Student Name: Georgio Semaan**

**Student ID: 20060609**

## Acknowledgments

I would like to express my sincere gratitude to Dr. Hoda Maalouf who helped me in every milestone through the development of this thesis. I want to recognize her guidance, constant encouragement and will in providing clarifications through all the challenges faced during the phases of this project.

When it comes to my precious parents, I would like to thank them for their presence and motivation throughout my entire academic life.

I would like to express my special gratitude to Dr. Jean Noel Semaan and Ms. Stephanie Matta, who offered invaluable support and humor over the years.

Finally, I would like to expand my special note of thanks to everyone who supported me during the years spent at Notre Dame University - Louaize, particularly my teachers and my colleagues especially Dr. Amine Moussa and Ms. Rita Mehanna.

It was a privilege and special honor to be one of Notre Dame University - Louaize family.

**Still We Rise**

## Table of Contents

<b>Acknowledgments .....</b>	<b>v</b>
<b>Table of Contents .....</b>	<b>vi</b>
<b>List of Figures .....</b>	<b>viii</b>
<b>List of Tables .....</b>	<b>x</b>
<b>List of Abbreviations .....</b>	<b>xi</b>
<b>Abstract.....</b>	<b>xii</b>
<b>Chapter 1: Introduction and Problem Definition .....</b>	<b>1</b>
1.1 Introduction to the General Problem .....	1
1.2 Problem Definition .....	2
1.3 Research Objectives .....	3
1.4 Approach and Main Results .....	3
1.5 Thesis Organization.....	4
<b>Chapter 2: Understanding Blockchain Technology .....</b>	<b>5</b>
2.1 Overview of Blockchain.....	5
2.2 Understanding how Blockchain technology functions.....	9
2.3 Blockchain Fundamentals .....	13
2.4 Requirements to be part of a Blockchain system .....	13
2.5 Structure of Blockchain.....	15
2.6 Types of Blockchain.....	21
2.6.1 Public Blockchains .....	22
2.6.2 Private Blockchains .....	23
2.6.3 Permissioned Blockchains.....	25
2.7 Blockchain uses.....	25
2.7.1 Smart contracts. ....	26
2.7.2 Cryptocurrency and Bitcoin .....	32
2.7.3 Other uses of Blockchain.....	35
2.8 Blockchain challenges and risks .....	37
2.8.1 Initial costs.....	37
2.8.2 Integration with Legacy Systems .....	38
2.8.3 Energy Consumption .....	38
2.8.4 Public Perception.....	39
2.8.5 Privacy and Security .....	40

2.8.6 Risk of losing keys .....	40
2.8.7 Mining centralization.....	41
2.9 Blockchain mining process .....	42
2.9.1 Consensus Algorithm .....	43
2.9.2 Mining requirements .....	58
2.9.3 Mining profitability .....	61
2.9.4 Bitcoin Cloud Mining.....	63
2.9.5 Bitcoin mining example .....	64
2.9.6 Mining problems. ....	67
2.9.7 Solution to the selfish Mining Problem.....	72
<b>Chapter 3: Optimum Penalty system against selfish mining .....</b>	<b>88</b>
3.1 Introduction .....	88
3.2 Original Work .....	91
<b>Chapter 4: Conclusion .....</b>	<b>107</b>
4.1 Summary of the Main Results.....	107
4.2 Main Contributions of the Thesis.....	107
4.3 Possible Extensions and Future Work.....	108
<b>Bibliography .....</b>	<b>109</b>

## List of Figures

Figure 1: How does a Blockchain works (Wild et al., 2015).....	10
Figure 2: Chronologically ordered blocks (Joshi et al., 2018).....	16
Figure 3: Cryptographic hash of a cat picture (Rosenbaum, 2019).....	18
Figure 4: Hashing the modified cat picture (Rosenbaum, 2019).....	19
Figure 5: Simple hashing example (Rosic, 2019).....	19
Figure 6: Example of Hash function used in Bitcoin (Brennan et al., 2016).....	20
Figure 7: Types of Blockchains (Gupta, 2017).....	22
Figure 8: Traditional Contracts (Voshmgir & Kalinov, 2017).....	27
Figure 9: Smart Contracts (Voshmgir & Kalinov, 2017).....	28
Figure 10: "Trust" in Smart Contracts (Voshmgir & Kalinov, 2017).....	30
Figure 11: Smart contracts Different levels ( <i>Smart Contracts - Simple to Complex - BlockchainHub</i> , n.d.).....	31
Figure 12: How does Bitcoin Work? (Rosic, n.d.-c).....	33
Figure 13: Block generation and mining process (Kim & Jo, 2018).....	43
Figure 14: BTC mining pools market share ( <i>Blockchain Charts</i> , n.d.).....	50
Figure 15: PoW versus PoS (Rosic, n.d.-b).....	55
Figure 16: Simple diagram showing Bitcoin mining (Kufeoglu & Ozkuran, 2019).....	65
Figure 17: 51% Attack (Katrenko & Sotnichek, 2020).....	68
Figure 18: Selfish Mining Fork (Katrenko & Sotnichek, 2020).....	70
Figure 19: State machine with transition frequencies (Eyal & Sirer, 2018).....	73
Figure 20: Pool Revenue (Eyal & Sirer, 2018).....	76
Figure 21: Relation between $\alpha$ and $\gamma$ (Eyal & Sirer, 2018).....	77
Figure 22: ZeroBlock generation to prevent block withholding (Solat & Potop-Butucaru, 2016a).....	86
Figure 23: HardFork in Selfish Mining (Saad et al., 2020).....	92
Figure 24: Graph of the Percentage Lost in function of Reduction Step (S1).....	98
Figure 25: Graph of Number of private Blocks lost in function of the Reduction Step (S1).....	99
Figure 26: Graph of the Percentage Lost in function of Reduction Step (S2).....	103



Figure 27: Graph of Number of private Blocks lost in function of the Reduction Step (S2)  
..... 103

## List of Tables

Table 1: Applications of blockchain (Nofer et al., 2017) .....	37
Table 2: Selfish Miners Behavior Comparision. ....	93
Table 3: Block rewards distribution according to 5% deduction in S1 (Fast SM) .....	95
Table 4: Relationship between reduction step and percentage lost (Hasty SM). ....	96
Table 5: Needed reduction step for selfish mining cases in S1 .....	100
Table 6: Block rewards distribution according to 5% deduction in S2 (Fast SM) .....	101
Table 7: S1 vs. S2 (5% reduction for Fast SM) .....	101
Table 8: Needed reduction step for selfish mining cases in S2 .....	104
Table 9: S1 vs. S2 in a simple case .....	105

## List of Abbreviations

IoT: Internet of Things.

HFT: High Frequency Trading.

PoW: Proof of Work.

PoS: Proof of Stake.

SHA-256: Secure Hash Algorithm 256.

CPU: Central Processing Unit.

GPU: Graphics Processing Unit.

FPGA: Field-programmable Gate Array.

ASIC: Application-specific Integrated Circuit.

CLI: Command-line Interface.

FaW: Fork-after-withhold.

BWH: Block-withholding.

BTC: Bitcoin.

ETH: Ethereum.

FP: Freshness Preferred.

SM: Selfish Miner.

## **Abstract**

In recent times, a new technology – Blockchain was brought forward to the global society. The world is using the term ‘Blockchain technology’ to signify diverse things such as the Bitcoin Blockchain, virtual currencies like Cardano and XRP, and smart contracts. Blockchain-based applications are springing up, covering several fields comprising financial services, reputation system and Internet of Things (IoT), and so on. Generally, Blockchain is implicit to be distributed ledgers that is a list of transactions verified and stored into blocks and shared among a number of computer nodes in a decentralized manner. Similar to any new introduced technology, Blockchain after its implementation by several communities is facing several problems. Among these complications, the famous block-withholding problem known also as selfish mining can arise anytime in any Blockchain system. This research describes in detail the Blockchain and focuses on solving the problem arising from selfish mining.

In Particular, a penalty system is proposed to defend against selfish mining. The approach is based on deducting percentages of the rewards acquired by any selfish miner after solving the PoW, transmitting the block for validation and later addition to the chain.

The goal of this research is to specifically find the optimum penalty system to guarantee stop selfish mining and to give an equal opportunity for all miners to get the proper rewards for their contributed work.

As a final point, by incorporating this approach, the decentralized nature of Blockchain is preserved, taking the fact that no selfish miner can take control over the system. The

involved Blockchain miners are guaranteed to get the proper reward distribution every time they participate in finding a block. Further modifications can be introduced to this approach by modifying the block size in a way to improve any Blockchain network.

Keywords:

Blockchain, Bitcoin, selfish mining, block-withholding.

# **Chapter 1: Introduction and Problem Definition**

This chapter introduces the general problem, research objectives, main result, and the thesis organization.

## **1.1 Introduction to the General Problem**

Over the past few years, the booming Blockchain technology has led to the development of numerous decentralized applications. The extensive use of Blockchain systems is leading to the creation of new chains regularly. These chains are consisted of consecutive related blocks that are added by nodes known as miners. Mining is the process by which nodes in the Blockchain's network validate and confirm transactions. The corresponding winning miner uses computational power (mining power) and energy to solve a difficult mathematical problem in order to add a new block to the chain. Hence, gain a reward as a reimbursement for his hard work and contribution. Mining is having huge attention these days, as miners all over the world are fighting each other to win the block races and get rewards. Some of the miners are tweaking the standard mining process towards achieving more rewards. Selfish mining is one of the methods, which can be used in this favor. This method will grant selfish miners a reasonable advantage over a regular miner on the Blockchain since their rewards are comparatively greater due to less wastage (Eyal & Sirer, 2018). In addition, it will lead to the collapse of the

decentralized nature of the Blockchain, as a selfish pool manager will control the system.

If not tackled properly, this loophole in mining process will increase the difficulty of Blockchain development and acceptance in the industry. Hence, Blockchain developers are facing many challenges of selecting the proper solution to this problem.

## **1.2 Problem Definition**

Selfish mining permits an individual miner or group of miners (pool) to obtain a revenue larger than its share of mining power. The key objective behind the selfish mining method is to force honest miners into performing wasted computations on the public branch of the chain. Explicitly, this method drives honest miners to spend all their mining power and hard work on blocks that are destined to be discarded later on. Selfish miners accomplish this goal by selectively publishing their mined blocks to quash the honest miners' work. In short, the selfish miner keeps its mined block private, creating a private branch and forking the Blockchain in secret. In the interim, honest miners continue mining on the shorter public branch. Since, selfish miners own a small portion of the total system mining power; their corresponding private chain will not stay ahead of the public chain for a long time. Therefore, the selfish miners reveal their private chain at specific opportune moments, such that the honest miners will be obliged to switch to this chain, abandoning the shorter public branch. This allows the selfish miners to gain higher

revenues, and refrains honest miners from the rewards after spending wasted efforts for their previous work.

This technique is harmful to the honest miners as well as to the system. Researches and solutions have been conducted and presented to solve this issue beforehand. Nevertheless, there is also a gap where new techniques can be found and implemented to this problem as no definite solution or framework can be used ultimately.

### **1.3 Research Objectives**

The main purpose of this research is to find a solution to selfish mining. This study aims to find the optimum penalty system to punish selfish miners based on the frequency of their work. As well as finding the correct percentage window to reduce gradually the profits of selfish miners to a point where this act becomes non-profitable and more of a total loss.

### **1.4 Approach and Main Results**

A new concept of penalty system is introduced and established to stop selfish mining in a totally different approach than the previous work done. This penalty system, if properly used is very simple and robust in preventing any selfish mining frequency. The calculations used in this approach are done according to two different scenarios. The outcome of both scenarios proved that they are mutually efficient and capable to end selfish mining.



## **1.5 Thesis Organization**

Chapter 2 presents the basic concepts of Blockchains by introducing the Blockchain technology, architecture, uses, challenges, and the mining process and problems. Then, previously proposed solutions to selfish mining problem are heavily presented after identifying selfish mining problem as one of the major threat to Blockchain. In Chapter 3, the original work is presented in detail alongside a calculation done that evaluates the proposed approach. The study is concluded in Chapter 4 by listing the main contribution and possible future work

## **Chapter 2: Understanding Blockchain Technology**

Blockchain is a system in which a record of transactions made in Bitcoin or another cryptocurrency are maintained across several computers that are linked in a peer-to-peer network (*BLOCKCHAIN / Definition of BLOCKCHAIN by Oxford Dictionary on Lexico.Com Also Meaning of BLOCKCHAIN, n.d.*).

Blockchain, the groundbreaking technology and the distributed ledgers are attracting considerable attention lately and initiating several projects in diverse industries (Nofer et al., 2017). The Blockchain concept is mainly used in the financial industry and more precisely in the well-known cryptocurrency application Bitcoin.

Blockchain is a novel solution that provides trustless trust. It is a shared, trusted, public, distributed ledger of transactions. That is created and used to revolutionize the way corporations and other firms do business.

### **2.1 Overview of Blockchain**

Blockchain technology since its introduction has been emerging in the past decade. It started in 2008 the minute a paper was published by Satoshi Nakamoto. It introduced Bitcoin as a digital asset that can be transferred directly between peers

in a trustless environment while maintaining a public history of all transactions. Bitcoin was built upon a technology called at some later time Blockchain.

Blockchain is a distributed, expandable computing architecture on the internet, where records of resources or exchange are stored. Blockchain allows peer-to-peer networking and public-key cryptography. It was mainly introduced as the core technology of the cryptocurrency Bitcoin, first implemented in 2009, serving as the public ledger for all the transactions.

Blockchain, the foundation of Bitcoin, has received extensive attentions recently, and is serving as an immutable ledger that allows transactions to take place in a decentralized manner. Blockchain-based applications are springing up, covering numerous fields including financial services such as online payment and digital assets, reputation system and Internet of Things (IoT), et cetera. Blockchain is also used in other fields including smart contracts, public services, and security services (Zheng et al., 2017).

However, even after a decade of introduction, this technology is still in its early stages of usage and development. Its potential is recently disputed; some people say it is destined for becoming the next milestone after the Semantic Web, while others grant it a less significant role. Nevertheless, enterprises, government agencies, and non-profit organizations are currently showing big interests in this new paradigm, which allows for transfer of any type of value. The original script is open source and grants access to everyone, a project aimed to foster democracy (Hermann, 2018).

Blockchain is considered to be like a public ledger and all committed transactions are stored in a list of succeeding blocks. This chain grows recurrently, as new blocks are regularly being added in a continuous manner. To achieve user security and ledger consistency, asymmetric cryptography and distributed consensus algorithms have been implemented strongly in every system relying on Blockchain. In general, Blockchain technology has several key features of decentralization, persistency, anonymity and auditability (Bansod & Ragha, 2020). With these characteristics, it can significantly save the cost and improve the efficiency of any application.

The use of Blockchain is favored in multiple ways. First, Blockchain is immutable which means that every transaction cannot be altered or tampered with at any given time once it is validated. This favors the main goal and requirement for high reliability and honest businesses in any industry towards attracting more customers. In addition, Blockchain is distributed, its use can avoid the single point of failure situation for any system. Furthermore, Blockchain technology when used in the smart contracts applications has a big advantage because, miners (transactions verifiers) can execute these contracts automatically once a contract has been deployed on the Blockchain (Chatterjee et al., 2021).

Even though the Blockchain technology has a great potential for future systems, it is still facing a great number of technical challenges. The major drawback encountered is scalability. For instance, the Bitcoin block size is limited to 1 MB for every block mined every ten minutes (Zheng et al., 2017). In due course, the Bitcoin network is limited to a rate of seven transactions per second, this

consequently leads to its incapability in dealing with high frequency trading (HFT) in financial markets. Nevertheless, larger chains generate problems in storage that is becoming large over time, and in propagation, which is making the network slow. This will lead to less users willing to maintain such large chains and to gradual centralization. Consequently, the tradeoff between block size and security has been a tough challenge (Zheng et al., 2017). Furthermore, previous research has as well proved that some miners could achieve larger revenue than their fair share through selfish mining strategy compared to honest mining (Eyal & Sirer, 2018). If this strategy works, a single entity of selfish miners will take control, and this will subsequently promote to the collapse of the decentralized nature of the system. In this case, these selfish miners hide their mined blocks or in other terms make these blocks private, while other public miners are mining their own blocks at the same time. While temporary hiding these blocks and releasing them at proper moments, selfish miners can make more revenue in the future. In this way, this group of selfish miners will control and delay the development of the chain. Hence, solutions need to be found and implemented in order to stop and fix the problem when occurred.

In addition, the same study done by Zheng et al. showed that privacy leakage could also happen in Blockchain even if the users only make transactions with their public key and private key (Zheng et al., 2017).

Moreover, currently used consensus algorithms like proof of work (PoW) and proof of stake (PoS) in Blockchain technology are facing some thoughtful problems.

Proof of work alone wastes a huge amount of electric energy, while proof of stake consensus lead to the phenomenon of the rich get richer.

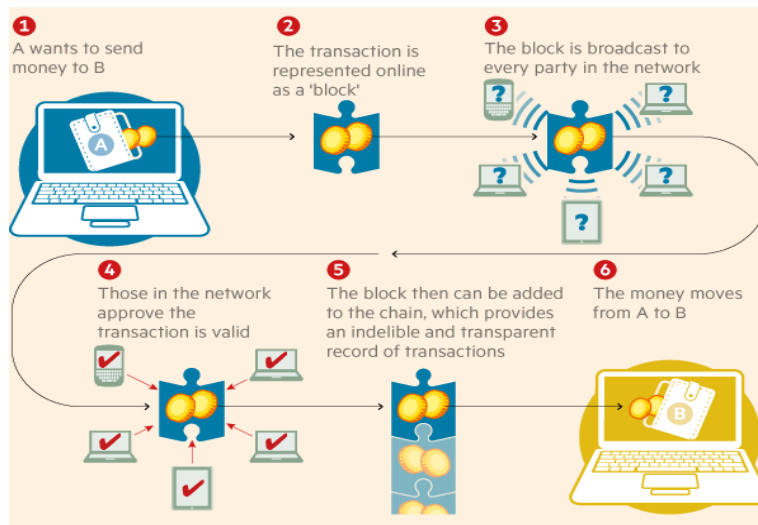
There is a lot of literature on Blockchain from various sources, such as academic papers, blogs, wikis, forum, research, conference proceedings and journal articles. However, because it is a new technology, more work and research are being conducted to the extent of taking full advantage of its numerous potential applications in the future.

## **2.2 Understanding how Blockchain technology functions**

Every block in a Blockchain is a computer code that contains a form of information, such as a contract, a certificate of ownership, a statement of authenticity, or a proof of bank's financial transaction. Nevertheless, each block of information is securely connected or chained to the other block through a digital signature. As new information is regularly added, the complexity and length of the Blockchain increases and the database consequently gets bigger with more and more people or nodes taking part of it. However, after adding the information to the chain, if a party or node tries to make any unauthorized change, every other participant in the chain can see where the change is probably going to happen and whether agree or disagree if that change should be valid or not. In most cases, this change will be invalid conforming to the Blockchain standards.

The following example will illustrate more the functioning of the Blockchain. Two parties A and B are involved in some sort of business, for example A wants to pay

1500\$ to B for a house rental. They can register that onto a Blockchain. They will record a contract (number of transactions) onto the Blockchain that will indicate that one of the parties has agreed to add 1500\$, so that another could rent the house. This will give them a transparent public ledger so that anyone in the chain can see that the latter has agreed to send the money and pay the rent. Thus, the parties involved would not be able to go back later and change the amount, or transfer the ownership through the title deeds because everything is publicly registered in the Blockchain as shown in Figure 1. For this reason, this distributed ledger allows the production of a tamper proof record of transactions. It is very similar to how an accounting ledger works. Nonetheless, essentially, the idea behind a distributed ledger is that we get rid of the intermediary (middleman) or the third party.



**Figure 1: How does a Blockchain works** (Wild et al., 2015)

Blockchain utilizes also cryptography to allow each member on the network to operate the ledger in a safe way without the need for a central authority.

Originally, Blockchain was just the computer science term for how to structure and share data. At the present time, Blockchain is hailed to be the “fifth evolution” of

computing, the so-called absent trust layer for the Internet (Laurence, 2017). However, it will eventually be able to generate and achieve trust in digital data, for the reason that once the information is added to the Blockchain database, it will be virtually impossible to change, tamper with or delete it. In other words, once a block is recorded on the ledger, it is problematic to alter or remove it. Such capability makes this technology special, and ultimately differs it from other systems. To explain more, as soon as a node wants to add any type of information to the chain, other participants in the network will run algorithms to assess and confirm the corresponding transaction and decide whether to validate it or not. If a majority of nodes decide that the transaction looks valid (that is, the information should match the Blockchain's history), then this new transaction will be approved with along other transactions and a new block is added to the chain (Lansiti & Lakhani, 2007).

Yet again, Blockchain technology is relatively new, and there are many ways to comprehend it more in the course of time. This innovation comes from incorporating old technology in new ways, and brings a new approach to distributed databases. It can be thought of these chains as distributed databases that store and share information controlled by a group of individuals. Thus, in its simplest form, a Blockchain acts like a shared, replicated, append-only database where write access is shared among participants, but validation can be performed by all participants in a public domain (Popper, 2018).

In addition, the blocks in a Blockchain can be considered as book pages if Blockchain is to be compared to a book, which is in this case a sequence or a chain



of pages. Each page in a book contains text, and information related to book title, chapter number, page number, author details, ISBN, etc (*Crypto Mining Glossary: Blocks in Blockchain, Hash, Reward - MineBest*, 2021). This information about data is called “metadata”. Metadata is a set data that describes and gives information about other data. Thus, a block contains data and information about the data (Kranz, 2021). In particular, every block’s header contains some technical information about the block, along with some other things.

The header consists of six important components:

- The technical data including an ID, a version number (related to the set of protocol rules this block conforms to), and the size of the block.
- The Merkle root (it gathers all the transactions in the block into a single hash).
- The Difficulty target (related to mining and the level to successfully mine the block).
- The previous block hash.
- The timestamp.
- The nonce (a random number that can be changed to create different hashes while searching for a suitable one during mining) (Frankenfield et al., 2021).

Thus, in order to be used in an effective way, a Blockchain has its own particular ways of storing and organizing data. On the other hand, nearly every system have always been using conventional Databases to store organized information. Consequently, the challenging Blockchain technology is simultaneously analogous to the concept of a database.

Nevertheless, for the big question asked on how a Blockchain system differs from a normal database. The simple answer to the above question will be that a Blockchain system is a package, which comprises of a standard database plus some software that adds new rows, validates that these new rows conform to a certain consensus, and broadcasts them to its peers (nodes) across a network, ensuring that all nodes have the same data in their respective databases.

## **2.3 Blockchain Fundamentals**

The common fundamentals to most Blockchain systems are:

- A novel solution that provides trustless trust.
- A de-centralized system.
- A data store that contains any type of data.
- A shared, trusted, public, distributed ledger of transactions.
- Continuously growing list of transactional records.
- Real-time data replication across a number of systems or nodes.
- Peer-to-peer distributed database.
- Usage of cryptography and digital signatures to secure data from tampering and revision (Lewis Antony, 2015).

## **2.4 Requirements to be part of a Blockchain system**

To be a part of a Blockchain system, every prospective participant needs to install and run required software that connect its computer or server to other participants in the network. By running these software, the participants act as individual validators known as “network nodes”. When a new node connects to the network

for the first time, it will download a full copy of the Blockchain database onto its computer or server memory.

Depending on the use of Blockchain, various commercial products help in the implementation procedure. A sample list of products available in the market are Ethereum (a decentralized platform that runs smart contracts on a custom-built Blockchain), Eris (a Blockchain with smart contract functionality), Hyperledger (developed by IBM), etc.

Depending on the intended application, the node must satisfy certain requirements. As stated in Bitcoin's website, on every computer or server wishing to become a Bitcoin full node, the corresponding user must install the suitable Bitcoin node software and follow its instructions properly (*Running A Full Node - Bitcoin*, n.d.). The website also states that any participant who meets the below requirements, will have an easy-to-use node:

- Computer hardware running recent versions of Windows, Mac OS X, or Linux.
- 200 gigabytes of free disk space, accessible at a minimum read/write speed of 100 MB/s.
- 2 gigabytes of memory (RAM).
- A broadband Internet connection with upload speeds of at least 400 kilobits (50 kilobytes) per second.
- An unmetered connection, with high upload limits, or a regularly monitored connection ensuring that its upload limits are not exceeded. On high-speed connections, it is common for full nodes to use 200 gigabytes upload or more a month. Download usage is around 20 gigabytes a month, and around an additional 195 gigabytes the first time, the user starts the node.

- A full node can be left running six hours a day. (Other things can be done on the computer while running a full node.) More running hours would be better, and best case scenario would be if the node would be constantly turned on and continuously working.

Otherwise, a participant might try running a node on weak hardware. This will possibly work but with a big probability that he will likely spend more time dealing with issues and not achieve the required goal and profit.

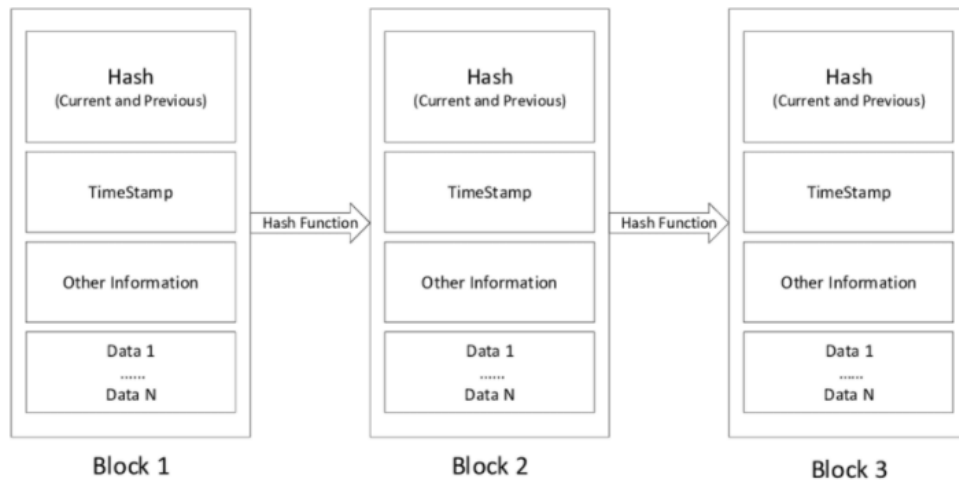
## 2.5 Structure of Blockchain

Blockchain architecture is composed of the following core components:

**Block:** As its name indicates, Blockchain is a chain of blocks. Each and every “block” is a list of transactions documented into a ledger over a given time interval. The chain is a continuously growing list of blocks or records linked together. Cryptography is used to secure all the blocks in any chain. The size, period, and triggering event for a block are different for every Blockchain depending on its use. Thus, the Blockchain is exactly a chain of blocks linked to together through the Linked list data structure. However, an important thing to note in this list is that instead of holding a traditional pointer to refer to the previous block, it uses the hash of the previous block to refer to it as an alternative (Laurence, 2017).

Similar transactions are grouped together to form a block. These formed blocks are then added in a chronological order to be later stored on the nodes local databases whether on their computers or servers.

It is important to mention that not all Blockchains are recording and securing a record of the movement of their cryptocurrency as their key objective. However, all Blockchain do record the movement of their tokens in any application where they are implemented. In other words, the transaction is simply a recording of the data and not the data itself. Moreover, when assigning a value to it, in the case of monetary or financial transactions, then it will be used to interpret what that data in particular means.



**Figure 2: Chronologically ordered blocks** (Joshi et al., 2018)

The first block in any Blockchain is called the genesis block, and it is the original message in any chain.

When updating the chain or adding a new message to it, the corresponding update added to the Blockchain is a new message connected to the one before. In other words, all the blocks are chained together and every new block will be based and directly related to the previous block and so on.

**Chain:** the “chain” known also as “hash” is what links one block to another in the Blockchain, which is mathematically “chaining” them together (S. Shetty et al.,

2019). In other terms, the chain is also the responsible for the binding of Blockchains together and it allows them to create mathematical trust. The hash in Blockchain is shaped from the data that is found in the prior block. It is the fingerprint of this data, and what locks blocks in order and time.

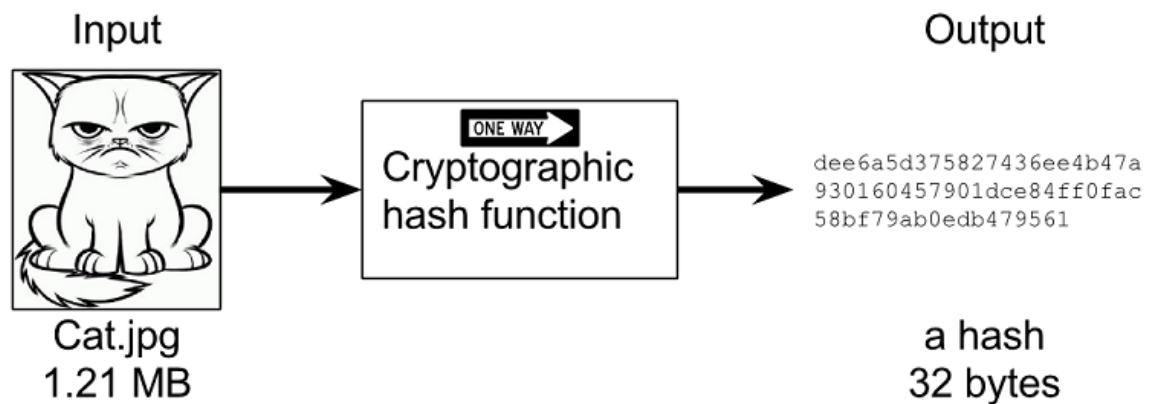
**Network:** is the third part of Blockchain, and is composed of “full nodes”. Full nodes are computers that run algorithms responsible of the network security. Each node comprises a complete record of all the transactions that were ever recorded in that chain. The nodes are positioned all over the world and can be functioned by anyone. In fact, it is challenging, expensive, and time-consuming to operate a full node, so people do not do it free. Instead, they are motivated to operate a node to get fees, rewards or incentives. The fundamental Blockchain algorithm recompenses them for their service usually by a token (the reward) or cryptocurrency (6.25BTC) in the case of Bitcoin.

Blockchain is somewhat an innovation compared to hashing, which was identified previously in the chain section. Many applications have been using the concept of “hashing” since it was invented, because of its strong point in generating a one-way function that cannot be decrypted.

To grasp the idea behind this state of the art concept, a detailed discussion on hashing is presented in the next paragraph in order to emphasize on its importance in any Blockchain system. It is the most complex model to understand in Blockchain systems because it will connect one block to another and allows them to generate trust.

In simple terms, hashing focuses on taking an input string of any length, and generating an output of a fixed length or prearranged size. The bit string output is typically 32 characters long, which subsequently represents the hashed data. To create a cryptographic hash of a file, the required file will be sent to a computer program specialized in performing hashing.

Figure 3 below shows a hash example by taking a random picture and creating its specific cryptographic hash.



**Figure 3: Cryptographic hash of a cat picture** (Rosenbaum, 2019)

In the Figure 3 above, the cat picture is taken, then processed by a certain mathematical function, and finally transformed to a big fixed length number. The obtained output number is not related by any means to the original picture. In addition, the original cat picture cannot be reconstructed from the obtained hash by whatever way. This is why hashing is acknowledged to be a *one-way function*.

Moreover, if any modification is done to the original cat picture and the same hash algorithm is applied to the modified picture. A different output will therefore be the result of the hash function as shown in Figure 4 below.

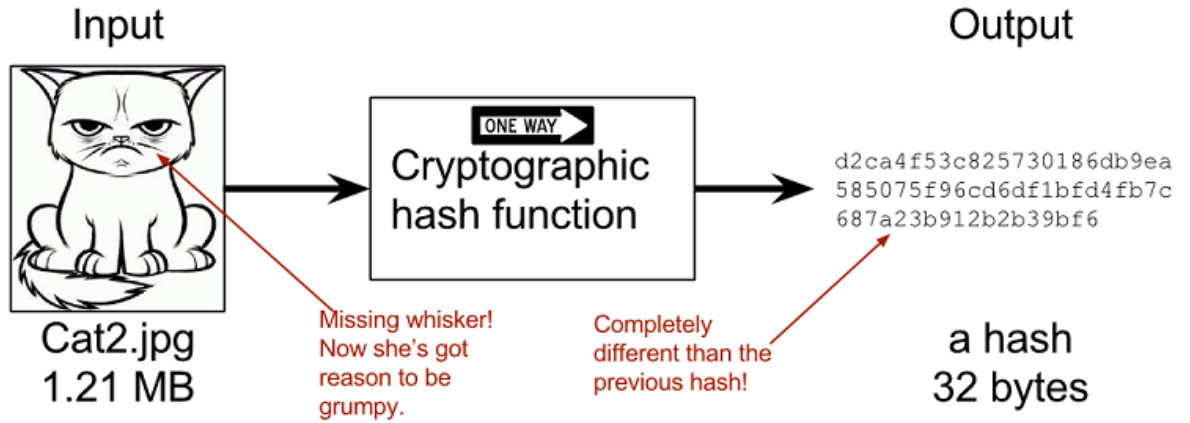


Figure 4: Hashing the modified cat picture (Rosenbaum, 2019)

Another example is illustrated in Figure 5 below. This simple case shows how hashing of strings is done. For any string of any length given as an input, an output of fixed length is generated.

INPUT	HASH
Hi	639EPCDo8ABB273B1619E82E78C29A7DF02C1051B1820E99FC395DCAA3326B8
Welcome	53A53FC9E2A03F9B6E66D84BA701574CD9CF5F01FB498C41731881BCDC68A7C8

Figure 5: Simple hashing example (Rosic, 2019)

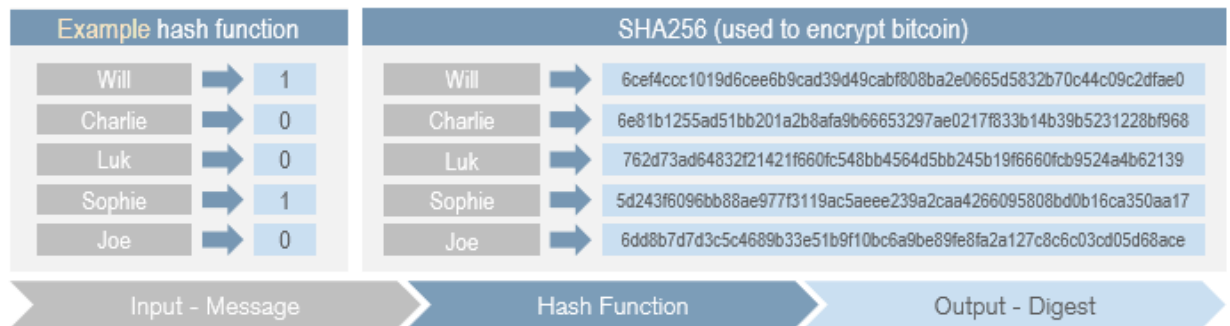
Furthermore, these hash methods are applied in Blockchain extensively. One of many cryptographic hash functions used by Blockchains is the Secure Hash Algorithm 256 known as SHA-256. A common algorithm generates an almost-unique, fixed-size 256-bit hash. Thus, every time this algorithm is used, and no matter how big or small the input is, the output will always have a fixed length of 256-bits (32-byte). Therefore, instead of remembering the input data, which sometimes could be varied and huge, the user can just remember the hash and keep track of it (Rosic, 2019). This becomes critical when dealing with huge amount of



data and transactions. As working and identifying the output hashes is a lot easier than working with than the diverse inputs given.

In addition, hash functions are a single cryptography that is the essence and the core of the Blockchain technology. In the cryptocurrency context, the Bitcoin protocol relies immensely on cryptography and hash functions, which in any instance transforms the corresponding input data (transactions) of variable size, to a proper output data of a fixed size.

The input message can be of any sort of data (text, character strings, binary etc.) and of any length. It will follow a precise mathematical transformation or a set of transformations to become a fixed length output.



**Figure 6: Example of Hash function used in Bitcoin** (Brennan et al., 2016)

A complex example is given in Figure 6, to compare and to demonstrate how hash functions transform messages into digests. This particular example shows actually the specific encryption SHA-256 used by Bitcoin. The first Hash function  $H$  is borrowed from the SANS institute 2003 (Silva, 2021). It is a very simple method, where it accepts messages of any length, and outputs a fixed length digest of one-bit. As shown in the Figure above, the first function  $H$  returns 0 as the message

digest if the input has an odd number of characters, and returns 1 if the output has an even number of characters. However, the second function is a more precise and protected hash method because for every input, a unique fixed length output is given. When applying it, the output obtained is more robust compared to the first function. SHA-256 is a part of the SHA functions family, the 256 represented in its name indicates that the output is 256 bits in size (Brennan et al., 2016).

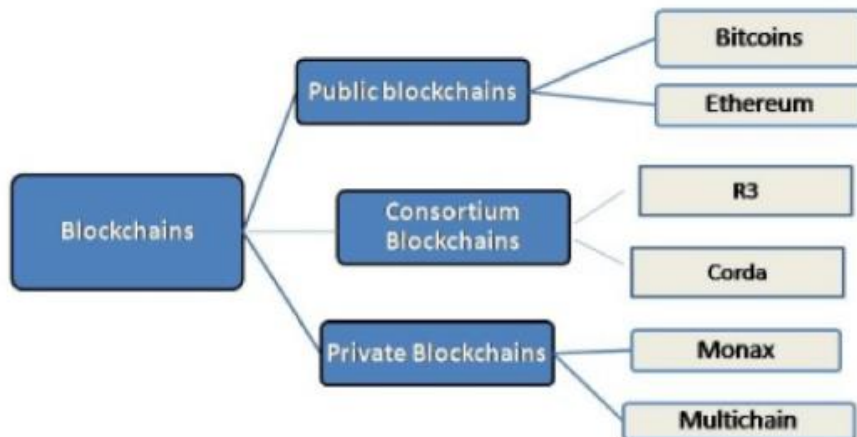
Moreover, the cryptographic power of any complex hash function is proportional to its output. In other words, it is hard even impossible to reverse the function to determine the corresponding input, taking a single output of any hash function.

Hash functions are deterministic models; they always give a consistent output from a given input. Whereas, reversing the hashing and trying to generate a message from any digest is implausible and considered as a mathematical trapdoor.

## **2.6 Types of Blockchain**

The main types of Blockchains as shown in Figure 7 are presented in this section:

1. Public Blockchains like Bitcoins and Ethereum.
2. Private Blockchains Monax and Multichain.
3. Permissioned (consortium) Blockchains like R3 and Corda.



**Figure 7: Types of Blockchains** (Gupta, 2017)

### 2.6.1 Public Blockchains

A public Blockchain is presented to be a permissionless Blockchain. This Blockchain network is open to anyone who wishes to participate in the network. Anyone can join the network as a node and can conduct transactions. Thus, a public Blockchain is a platform where anyone can read or write in the system, provided that they could show a proof of work. In addition, this type of Blockchain is a fully decentralized network where members can download the software on their personal computers or servers and track the records. Participants will be sharing the data among them, and each one of them can see and take part in the ongoing transactions. Not a single node will be able to maintain ledgers in these types of Blockchains because it is entirely open to the public to join and participate. Thus, all the users are allowed to download a copy of the ledger on their local machine, and can analyze the ongoing transactions on the network. However, the final decision will be reached according to the distributed consensus mechanism maintained by the corresponding miners.

Some examples of this private type are:

- ***Ethereum**, a provider of a decentralized platform and programming language that helps running smart contracts and allow developers to publish distributed applications.*
- ***Factom**, a provider of records management, records business processes for business and governments.*
- ***Blockstream**, a provider of sidechain technology, focused on extending capabilities of Bitcoin. The company has started experimenting on providing accounting with the use of public Blockchain technology (Adarsh, 2017).*

### **2.6.2 Private Blockchains**

A private Blockchain is open to a certain group of people that has agreed to share the ledger with each other. On the other hand, this type of chains is more secure compared to public Blockchain network. Particularly, only groups of people who have agreed to share the ledger take part in the network, and have mutually approved to trust an organization known as the owner. This organization has given the authority to allow participants to read a particular transaction, as not all transactions are necessary available to be read by participants. In other words, private Blockchains allow only the owner to gain the rights to make any changes that have to be done in any transactions. This could be similar to the existing centralized authority, where the owner has all the power to change the rules, revert transactions, etc., based on the need and requirements. Moreover, private network gives more privacy to the members. Without proper permission, a node is not

authorized to see any transaction. The information sent over the network is completely secured, hard to break and remains totally unfolded.

Another benefit of this type is that the information under such a network cannot be hacked because it is sent in the form of cryptography, which is very secure and very difficult to be tampered with. Thus, partaking in the private Blockchain network is trusted and secure, as nobody else will get to know about the transactions.

Banks and financial institutions use private Blockchains instead of the public ones in order to make banking transactions safer. These financial institutions could find and work on specific use cases to build proprietary systems and reduce the costs, while at the same time increase their efficiency.

Some of the examples could be:

- Eris Industries, aims to be the provider of shared software database using Blockchain technology.
- Blockstack, aims to provide financial institutions back office operations, including clearing & settlement on a private Blockchain.
- Multichain, provider is an open source distributed database for financial transactions.
- Chain Inc., a provider of Blockchain API's. Chain collaborated with NASDAQ OMX Group Inc., to provide a platform that enables trading private company shares with the Blockchain (Adarsh, 2017).

### **2.6.3 Permissioned Blockchains**

Permissioned Blockchains also called Consortium Blockchains fall between the public and private chains. They are also called: partial public and private Blockchain. In this type of chains, only several selected members are allowed to share the ledger leaving other members aside. These members are given full control to share ledgers and oversee the transactions. Not one authority will have control in the consortium Blockchain, because control is divided between more than one participants. In other words, it is one where instead of allowing any person with an internet connection to participate in the verification of transaction process or allowing a single company to have full control; a few selected nodes are predetermined (Varshney, 2017). Analogous to private Blockchains, user and data privacy is maintained in this type also.

## **2.7 Blockchain uses**

Blockchain technology is being addressed as one of the most groundbreaking and revolutionary technological advances at present. This technology revolutionized the perception of all monetary related dealings by introducing the Bitcoin. Nevertheless, what gives the Blockchain an immense potential for the near future, is its ability to be implemented in new applications beyond processing Bitcoin transactions. Moreover, Blockchain technology is targeting currently a range of industries such as healthcare, trade, finance, et cetera.

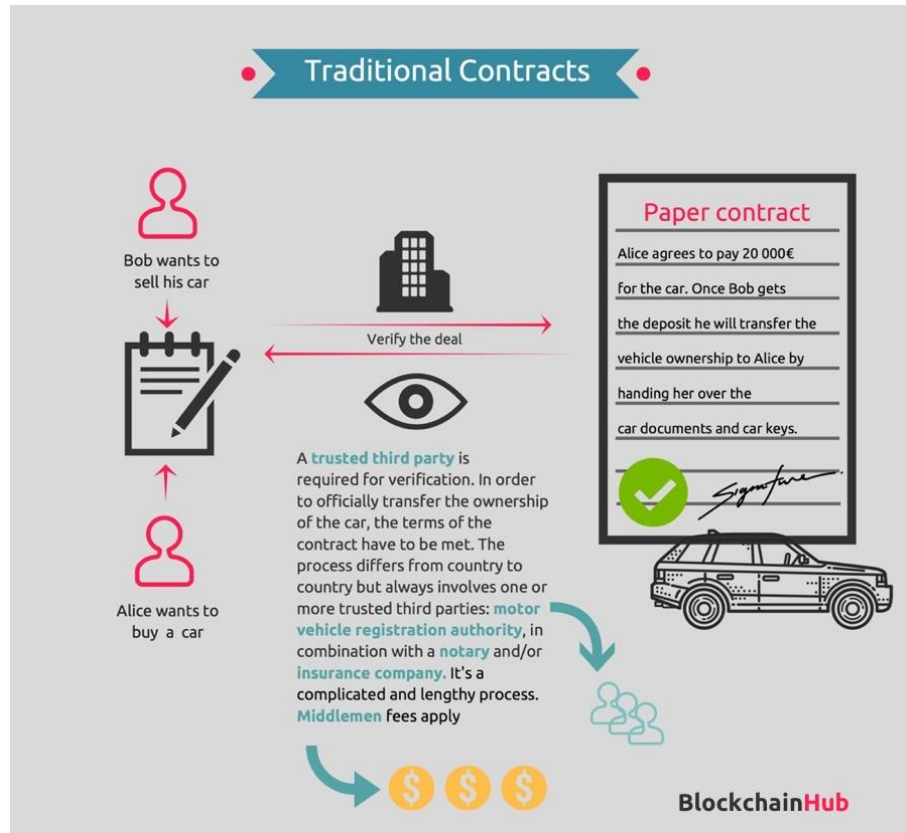
These applications are introduced in the following sections.

### **2.7.1 Smart contracts.**

“A smart contract is a self-enforcing piece of software that is managed by a P2P network of computers. Smart contracts are efficient rights management tools that provide a coordination and enforcement framework for agreements between network participants, without the need of traditional legal contracts. They can be used to formalize simple agreements between two parties, the bylaws of an organization, or to create tokens”.(*What Is a Smart Contract? Auto Enforceable Code - Blockchain*, n.d.)

In other words, smart contracts are considered as secured pieces of code or procedures. Written and stored on a Blockchain, they are executed when predetermined terms and conditions are met. The main benefit of smart contracts is that they are used in most business collaborations. They are utilized to enforce some type of agreement allowing participants to be certain of the outcome, with eliminating the need for any intermediary or trusted third parties involvement (*What Are Smart Contracts on Blockchain? | IBM*, n.d.).

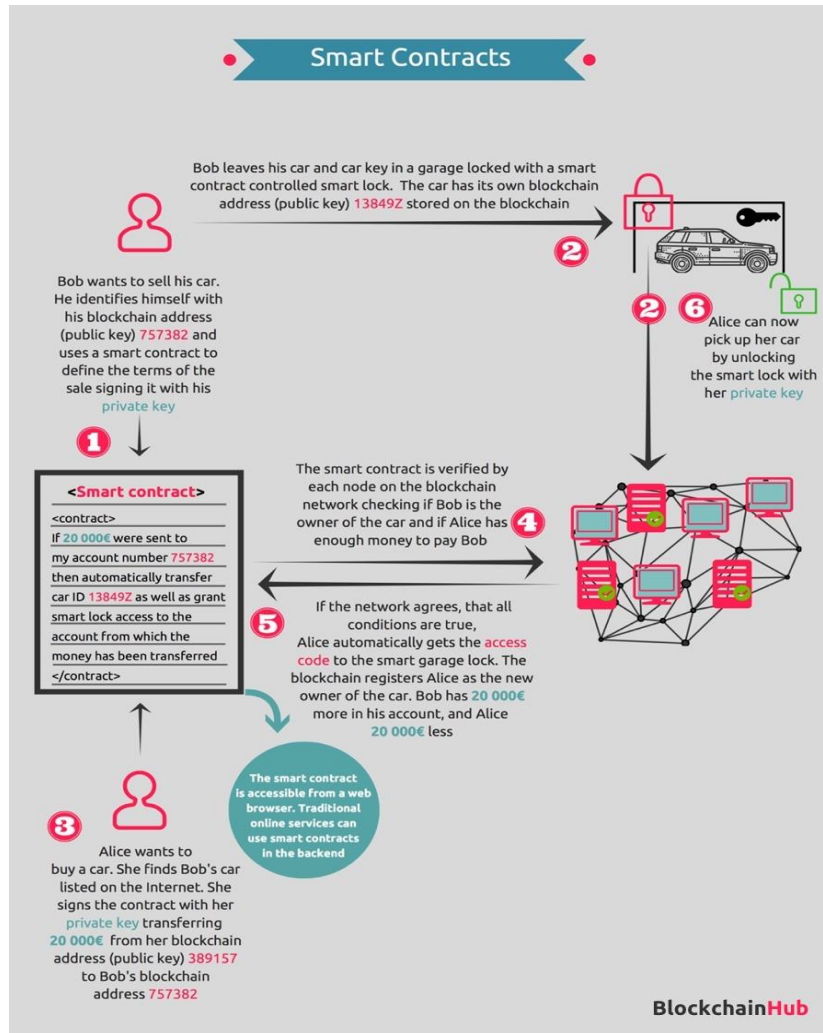
The best way to explain the concept of smart contract is through a car sale example.



**Figure 8: Traditional Contracts** (Voshmgir & Kalinov, 2017)

Bob wants to sell his car, and Alice is a prospective buyer. Ideally, in the traditional scenario, a trusted third party is required to verify every step in this procedure. In order to transfer officially the ownership of the car, the terms and conditions of the contract have to be met by the two participants. This process is usually frustrating, complicated and involves one or more trusted third parties like the motor vehicle registration authority, insurance company, notary, etc... To compensate the work of these intermediaries, various commissions and fees will be consequently added to the base price of the car.





**Figure 9: Smart Contracts** (Voshmgir & Kalinov, 2017)

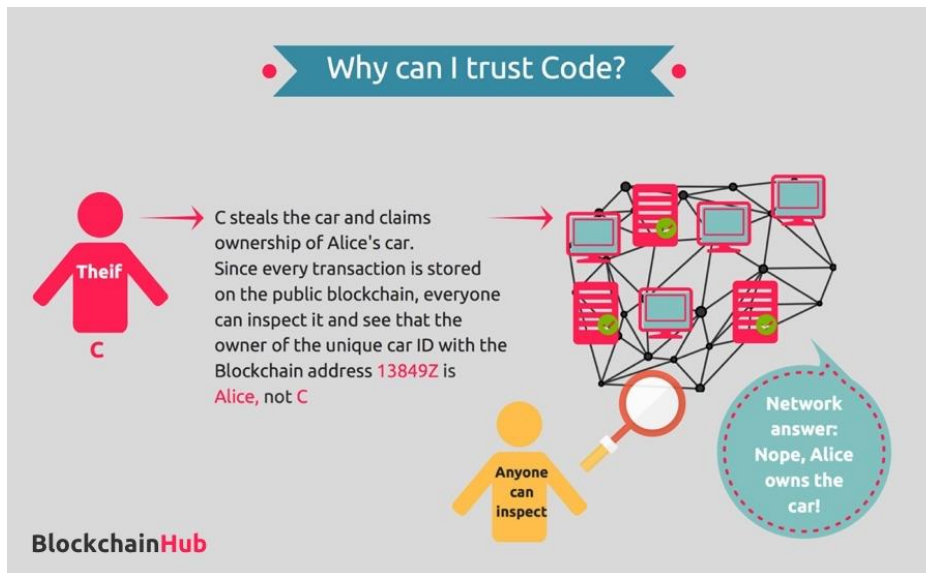
The main purpose of a smart contract on Blockchain is to restructure the complex traditional contracts process in a way to eliminate the involvement of the several third parties associated to the lack of trust between the participants.

In this case, the seller Bob identifies himself with his public key also known as Blockchain address 757382 and uses a smart contract to define the terms and conditions of the sale signing it with his private key. For example, the computer code of the smart contract is written and states that if 20 000 € were sent to the account number 757382 then automatically transfer car ID 13849Z as well as grant

lock access to the account from which the money has been transferred. Second, Bob leaves his car and keys in a garage with smart contract controlled smart lock. The car also has its own public key 13849Z stored on the Blockchain. Third, Alice the potential buyer finds Bob's car listed, she will then proceed to sign the contract with her private key transferring 20 000 € from the Blockchain address or public key 389157 to Bob's Blockchain address 757382. Fourth, the smart contract is verified by every node on the Blockchain network checking if Bob is the owner and if Alice has the needed amount to pay Bob. Fifth, if the network agrees that all the conditions are met, automatically Alice will then get the smart garage lock. The transfer of ownership would be automatic as the transaction gets registered to the Blockchain. Bob has now 20 000 € more in his account and Alice has 20 000 € less, plus the ownership of the car. No intermediary were required in the whole process. Finally, Alice can now unlock the smart lock with her private key and pick her car.

The main feature here is that every transaction is transparent on the chain. This means that every computer running the Blockchain protocol could check at any time whether a certain person is the rightful owner of the car or not.

Stealing cars would be very difficult, particularly once a person has the verified Blockchain smart keys granting access control, to unlock their prospective cars. As the owner of the car, Bob for instance can authorize others to drive his car as long as he states the public key of the specific individual. In these cases only opening the car would be possible with a smart key on the Blockchain.



**Figure 10: "Trust" in Smart Contracts** (Voshmgir & Kalinov, 2017)

### 2.7.1.1 Types of smart contracts:

Blockchain and smart contracts have a great potential in many industries. Smart contracts can be found in banking, insurance, energy sector, e-government, telecommunication, music & film industry, art world, mobility, education and many more fields. Smart contract use cases range from simple to complex.

Land titles, birth certificates, birth certificates, school and university degrees are examples for simple technological use cases. Decentralized autonomous organization on the other hand, are the most complex form of a smart contract. Figure 11 shows the different levels of smart contract complexity and the possible real world applications.



**Figure 11: Smart contracts Different levels** (*Smart Contracts - Simple to Complex - BlockchainHub, n.d.*)

Taking into account the fact that Blockchain is still a new technology, some industries could adopt smart contracts later than others, especially if they are subject to heavy rules or their applications require high network properties.

**2.7.1.2 Benefits of smart contracts:**

Smart contracts have many benefits that go together with Blockchain technology. The benefits are as follows:

- **Savings:** Smart contracts remove the need for trusted third parties since participants can trust the visible data, and the technology to properly execute the transaction. There is no need for an intermediary to validate and verify the terms and conditions of a contract as it is coded and stored in the Blockchain.
- **Security:** Transaction records are encrypted in the Blockchain, this will make them very hard to hack. Each individual record is connected to previous and subsequent records on a distributed ledger, the whole chain would need to be

altered to change a single record (*What Are Smart Contracts on Blockchain? / IBM, n.d.*).

- **Speed and accuracy:** Smart contracts are automated. No processing paperwork time is needed or other complications that may occur often in the required documents that have been filled mostly manually. Computer code is also more exact and accurate than the traditional contracts.
- **Trust:** Transactions in Smart contracts are automatically executed following predetermined rules. The encrypted records of those transactions are shared across participants. Thus, the transactions can't be altered or changed plus they are secured from tampering and revision.

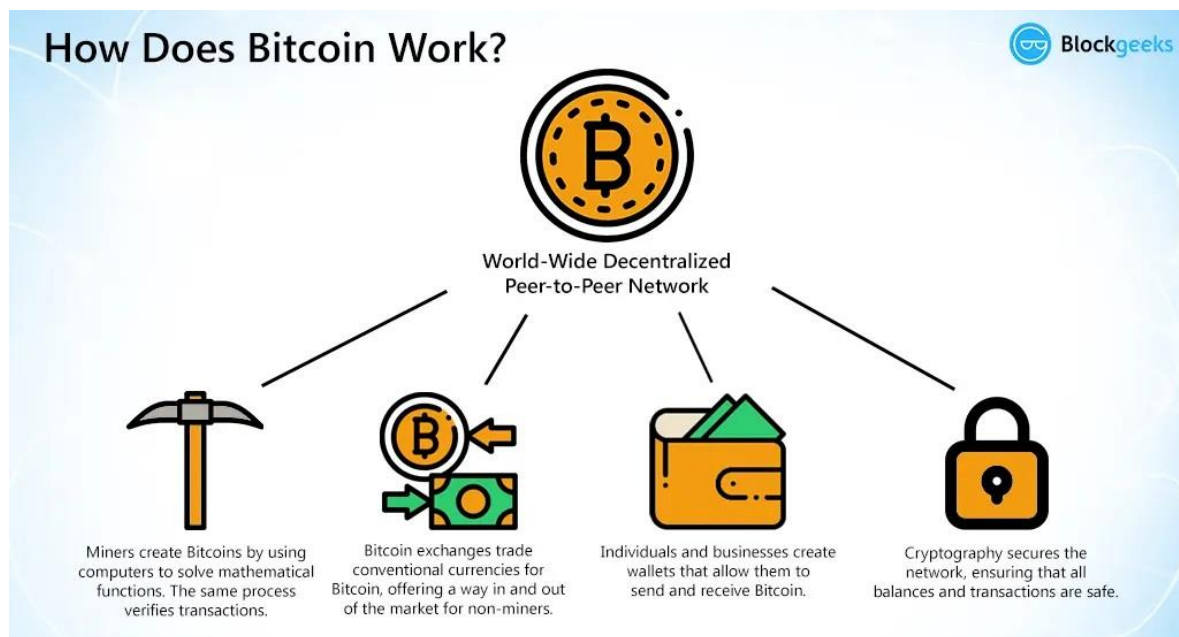
### **2.7.2 Cryptocurrency and Bitcoin**

Bitcoin, a type of cryptocurrency, is a digital currency that allows a protected and decentralized payment system without the need of a central bank or single administrator. It can be used between users on the peer-to-peer Bitcoin network without the need for a third party.

Bitcoin works on the Blockchain technology (a vast public ledger), where all confirmed transactions are included in the blocks. Each block is broadcasted to the users in the network for validation. This way, all the nodes are aware of each transaction, which subsequently prevents stealing and double spending.

Bitcoin inventors created this decentralized system, in a way that each user controls his own funds and knows what is going on in the system. Specifically, they wanted to put the seller in charge by eliminating the middle man, and making all the transactions transparent.

The usage of Bitcoin has evolved rapidly in a very short time. Its currency is accepted worldwide, from jewelry chains (REEDS Jewelers) to hospitals and companies. Businesses such as Expedia, Dell, PayPal, Tesla and Microsoft use it also. Websites and Bitcoin Magazine publish its news, forums discuss cryptocurrency and trade its coins. Bitcoin has also its own application programming interface (API), price index, and exchange rate.



**Figure 12: How does Bitcoin Work?** (Rosic, n.d.-c)

To start using Bitcoins, individuals and businesses need to set up and create their own virtual wallet. This wallet is a free open-source software that can be acquired by anyone from bitcoin.org; it acts like a private finance software to keep track of the Bitcoin balance and transactions. The wallet moreover holds all the individual's Bitcoin funds, transactions and security keys.

There exist two ways to earn Bitcoins. The first way is by purchasing BTC using real money at online companies or exchanges. These Bitcoin exchanges trade

conventional currencies for Bitcoin, offering a way in and out of the market for non-miners. The second way to earn them is by mining them. Miners create Bitcoins by solving complex mathematical functions. Once the mathematical problem is solved and all the other nodes agree, the new Bitcoin (BTC) is added to the public ledger; the successful miner will then be awarded a bit of the new Bitcoin for his effort. At any point in time the created or bought Bitcoins can be used in specific transactions to specific applications.

To understand more how a Bitcoin transaction works, a small example will clarify how these coins are used between individuals.

Barbara, a buyer, wants to use her Bitcoin to buy a merchandise from John, an online merchant. She will send him her private key which contains the amount, the address she is transferring Bitcoins from, and John's wallet address. John will then receive the key in order to decode it. At the same time, the transaction is broadcasted to all the other nodes on the network (Barbara's ledger) for verification. This verification process is done by miners. Their computers bundle the transactions of the past 10 minutes into a new transaction block. Once confirmed, this mining process gives John the green light to proceed with Barbara's transaction. Consequently, the winning miner will have its share and get paid - in this case, Bob. A new address is created in Bob's wallet with a balance of newly issued Bitcoins (Rosic, n.d.-c).

### **2.7.3 Other uses of Blockchain**

Apart from its big potential in the financial and business fields. Blockchain can be used across many more fields due to the fact that it is time saving, secure and it improves trust.

The insurance industry can make use of Blockchain because insurance providers require a well-organized method to process claims and provide customers with objective and timely payouts. For example, Everledger is a company that produces ledger of diamond certification used by insurance companies to create/read/update claims.

Blockchain can be used as well in the public sector. Through its use, recordings of various governmental interactions can be transformed into a more effective and transparent way of doing things. Thus, Blockchain could be used in the voting process in order to increase transparency and improve voter confidence. For example, Ballotchain solution is used to match a Bitcoin transaction to a vote cast by an elector in support of the respective candidate selected. Basically, voters cast their vote by giving a Ballotcoin (a little sum of cryptocurrency as wanted) to the wallet of their candidate. Each vote hence benefits from the characteristics of a Blockchain transaction, to be specific: It is non-modifiable; it is non-repudiable; it cannot be enrolled in numerous ways; all nodes have a substantial duplicate.

In addition, Blockchain in healthcare is seen as a state of the art model for health data exchanges. It is an effective and safe model for managing medical records for pre-authorizing payments, and for recording all related transactions. For example,



Factom, a provider of Blockchain technology, announced its partnership with medical records and services solutions provider HealthNautica. This partnership strives to provide a permanent record-keeping system, to secure medical records and to audit trails using Blockchain technology.

Lastly, application of Blockchain resides in the music industry. Blockchain can secure and maintain music rights ownership info in a public ledger in addition to the distribution of payments in this type of business.

Table 1 shows the different applications of Blockchain.

Type	Application	Description	Examples
Financial applications	Crypto-currencies	Networks and mediums of exchange using cryptography to secure transactions	Bitcoin Litecoin Ripple Monero
	Securities issuance, trading and settlement	Companies going public issue shares directly and without a bank syndicate. Private, less liquid shares can be traded in a blockchain-based secondary market. First projects try to tackle securities settlement	NASDAQ private Equity, Medici, Blockstream, Coinsetter
	Insurance	Properties (e.g., real estate, automobiles, etc.) might be registered using the blockchain technology. Insurers can check the transaction history	Everledger
Non-financial applications	Notary public	Central authorization by notary is not necessary anymore	Stampery Viacoin Ascribe
	Music industry	Determining music royalties and managing music rights ownership	Imogen heap
	Decentralized proof of existence of documents	Storing and validating the signature and timestamp of a document using Blockchain	
	Decentralized storage	Sharing documents without the need of a third party by using a Storj peer-to-peer distributed cloud storage platform	

Decentralized internet of things	The blockchain reliably stores the communication of smart devices within the internet of things	Filament ADEPT (developed by IBM And Samsung)
Anti-counterfeit solutions	Authenticity of products is verified by the blockchain network consisting of all market participants in electronic commerce (producers, merchants, marketplaces)	Blockverify
Internet applications	Instead of governments and corporations, Domain Name Servers (DNS) are controlled by every user in a decentralized way	Namecoin

**Table 1: Applications of blockchain** (Nofer et al., 2017)

## 2.8 Blockchain challenges and risks

Blockchain technology may be seen very promising to be used in the future. Despite its major benefits, there are also many challenges that are still preventing its widespread adoption. At present, most of the noticeable risks may consist of Blockchain-based crypto-currencies, but that may change as new applications arise from its use.

### 2.8.1 Initial costs

The software implemented to run Blockchain technology is usually developed according to the needs of each application. Therefore, in most cases it will be very expensive to initially implement the required software, or to purchase it, or to develop in-house. Moreover, firms need to acquire specialized hardware in accordance with the software in order to run this technology smoothly.

In addition, qualified personnel must be hired and paid large amount of money to be able to do the right job. Thus, the shift to a complete or partial Blockchain system is limited to high-sized business due to the inevitable high setup costs

involved (*Five Challenges Blockchain Technology Must Overcome Before Mainstream Adoption* / Nasdaq, n.d.).

### **2.8.2 Integration with Legacy Systems**

In order to integrate a Blockchain system, firms must either modify completely their previous used system or come up with a decent resolution to fit their existing system with the new Blockchain-based solution. Significant set of relocation tasks need to be executed to transfer business documents and frameworks to the new system to completely migrate from the old legacy system. Therefore, large changes must be done to the existing systems in order to facilitate a smooth transition. However, sometimes it may be impossible for Blockchain solutions to handle all the needed functions. Hence, in these particular cases a new system that is compatible with Blockchain will be deployed by the organization to reconcile effectively the two systems.

This procedure may take an important amount of time, assets and funds. Thus, many organizations are unwilling to make the move to Blockchain solutions (*Five Challenges Blockchain Technology Must Overcome Before Mainstream Adoption* / Nasdaq, n.d.).

### **2.8.3 Energy Consumption**

In order to validate transactions made on the Blockchains, both Ethereum and Bitcoin networks use and rely on the proof of work (PoW) consensus mechanism. This validation process depends on the computation of a very complex mathematical problems. The calculations require large amounts of energy to power

the computers solving the problems, and to cool down the hardware used. According to The economist magazine mining new virtual coinage in Bitcoin needs a big amount of electricity.

“Alex de Vries, a bitcoin specialist at PwC, estimates that the current global power consumption for the servers that run bitcoin’s software is a minimum of 2.55 gigawatts (GW), which amounts to energy consumption of 22 terawatt-hours (TWh) per year—almost the same as Ireland. Google, by comparison, used 5.7 TWh worldwide in 2015.” (*Why Bitcoin Uses so Much Energy | The Economist*, n.d.)

This energy consumption is growing erratically with time as the number of Bitcoin miners is increasing. Besides, they are consuming each year approximately five times more energy than the year before, and there is no signs of slowdown. Companies using Blockchain are running to find more sustainable methods to address energy issues with climate change being a major concern.

#### **2.8.4 Public Perception**

While Blockchain technology is reforming many different industries, knowledge of the benefits of this distributed technology is still narrow and limited to those who are involved in its space, and those whose industries are implementing its solutions. Currently, Blockchain technology is widely used in Bitcoin.

Prior to achieving majority acceptance, members of the public must learn the difference between Bitcoin, other cryptocurrencies and the Blockchain. This will allow the technology to stand on its own, which will lead to a growth in readiness

to exploit the technology (*Five Challenges Blockchain Technology Must Overcome Before Mainstream Adoption* | Nasdaq, n.d.).

### **2.8.5 Privacy and Security**

Blockchains are designed to be publicly visible. This feature is creating a number of concerns since all the data belong on a public ledger, and is accessible for the world to see and use. For example, the Bitcoin Blockchain, is designed to be accessible to all those who have made a transaction on this particular network. However, governments and some corporations require protection, control, and access restriction to their data for a countless of reasons. Thus, Blockchain will not serve its purpose in these particular cases.

Still, the use of private Blockchains may settle this matter for some as it can be customized to meet the needs and specifications of the task at hand. Although implementing such chains takes a big amount of resources and expertise. This will consequently lower the firms and governments urge to adopt and implement this technology.

### **2.8.6 Risk of losing keys**

A private-public key pair is being used in Blockchains to encrypt and decrypt the data in blocks. They are fundamental to each and every transaction recorded on any chain.

As stated in Investopedia, a private key is defined as an advanced form of cryptography that allows a user to gain access to his or her cryptocurrency. It is

represented as a series of alphanumeric characters, 51 in total preventing it from being hacked or deciphered.

The Storage and preservation of private keys is mandatory and essential to the involved chain participants. However, if a member loses his private key, he can no longer have access to the wallet, preventing him to manipulate coins. The safest solution to this problem is to always store the private key in any possible way, and in a very secure location (Frankenfield, 2020).

### **2.8.7 Mining centralization**

Mining is the process by which transactions are verified, validated, and added to the Blockchain. It is the procedure of solving complex mathematical cryptographic problems using high end computing hardware. Miners usually get incentives for every transaction validated or for every Bitcoin created (Kano & Nakajima, 2018). This will encourage people to take part of these mining networks and earn rewards according to their ability and assets to solve cryptographic problems.

According to Forbes, “Someone who controls 51% of the computing power pointed at the Bitcoin network is able to choose which transactions can be processed. Someone who controls the majority of the network hashrate can also reorganize the history of network transactions in a malicious attempt to spend the same money twice.” (Torpey, 2019)

Gaining such a large percentage is not easy, but if it is reached in any way, it can cause Bitcoin to collapse and lose its purpose. Thus, it might become for instance

more like a traditional payment system. And eventually, currencies it was meant to improve and replace in some cases will have the same use.

## **2.9 Blockchain mining process**

In every Blockchain, all the made transactions are grouped into blocks according to a predetermined block size. Every newly created block is then joined to the previous blocks in a continuous manner. This will create the chain of blocks, hence the name Blockchain is used (Battista et al., 2015).

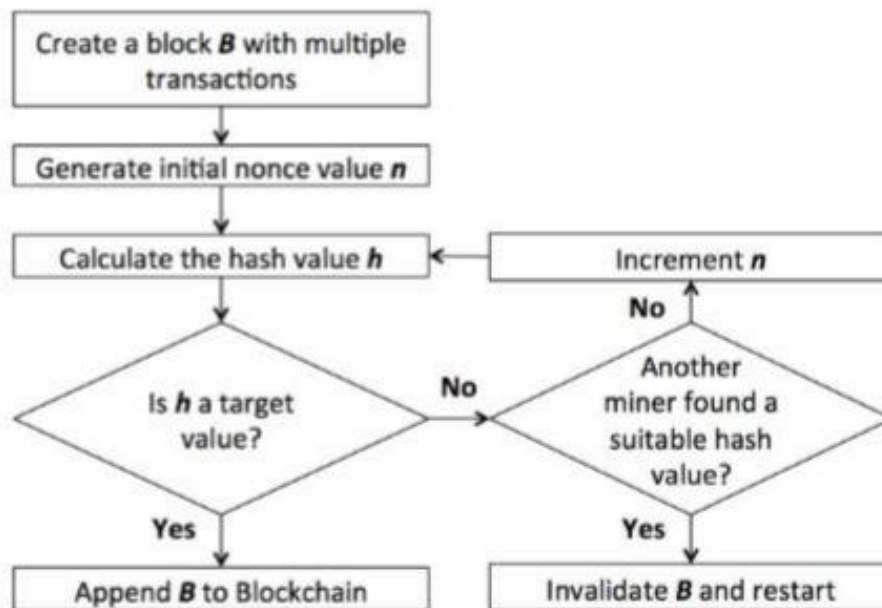
The structure of a typical block is shown in Figure 13 below. To prevent any arbitrary addition of a block into the Blockchain, there is an crucial process applied called mining (Kim & Jo, 2018).

Prior to adding a block to an existing Blockchain, a signature value must be discovered that produces a specific hash value. In detail, it is a hash operation that takes the summary hash of all the transactions within a block; in this case the new block to be added; and the previous block's hash value as an input.

Additionally, a specific value called nonce is added and a new hash value is calculated. This procedure continues with a new nonce continuously generated until a special nonce is created that produces a hash value starting with a predetermined number of 0's.

The effort to find the signature value can be done by any individual or organization with suitable computing resources, called miner or mining pool. For example, in Bitcoin and Ethereum there are thousands of miners in the mining network that compete to find the suitable signature value. In any case, when a miner successfully

finds the signature value, he/she can attach the block to the Blockchain and get rewarded with new cryptocurrency (Kim & Jo, 2018).



**Figure 13: Block generation and mining process** (Kim & Jo, 2018)

From a technical point of view, the mining process is the operation of inverse hashing: it determines a nonce, so that the cryptographic hash algorithm of block data results in less than a given threshold.

### 2.9.1 Consensus Algorithm

Blockchain by its nature has no central authority due to the fact that it is a decentralized peer-to-peer system. This will consequently produce a corruption free system form. Nevertheless, it still creates a major problem concerning decision making when compared to a typical centralized organization where all the decisions are taken by a leader or a board of people. This task is impossible in Blockchains



because every Blockchain has no leader. Hence, the decision making in these systems must follow certain consensus mechanisms.

“Consensus decision-making is a group decision-making process in which group members develop, and agree to support a decision in the best interest of the whole. Consensus may be defined professionally as an acceptable resolution, one that can be supported, even if not the “favorite” of each individual. Consensus is defined by Merriam-Webster as, first, general agreement, and second, group solidarity of belief or sentiment.” (Rosic, n.d.-a)

In simpler words, the consensus is used to reach agreement between parties in a certain approach that could benefit the entire group as a whole. Unlike voting, where a majority will rule without taking care of the well-being of the minority. Ideally, consensus can be used by a group of people distributed around the world to create a more equal and unbiased decision making.

The method by which consensus decision-making is reached is called the consensus mechanism. The objectives of this mechanism are agreement seeking, collaboration, cooperation, egalitarianism, inclusion, and participation.

Before Bitcoin, many systems failed because they were unable to solve a certain consensus problem called “Byzantine Generals Problem”.

The Byzantine Generals Problem is a term used in computer science in definite situations where involved parties must agree on a single approach in order to avoid complete failure. However, some of the involved parties are corrupt, untrustworthy and broadcast incorrect information.

To understand more this problem, let there be  $n$  generals  $\{G_0, G_1, G_2, \dots, G_{n-1}\}$ , where  $G_0$  is the commanding general. All these generals can communicate with each other by sending one to one messages. Supposing that the commander  $G_0$  holds a certain information bit  $b_0$  initially. The main goal is to design a system responsible for sharing the same bit  $b_0$  between all the generals. Nevertheless, some of the generals may be traitors, and in some cases the commander is one of them as well. Their main goal is to prevent the loyal ones from reaching a definite agreement. It is supposed that a trustworthy general will always execute a protocol authentically, while a traitor may do anything as it desires.

The accurate protocol should satisfy the following conditions:

1. All dedicated generals will reach an agreement at the end and hold the same bit  $b_0$ .
2. If  $G_0$  is reliable, every loyal general will hold  $b_0$  at the end.

Let  $t$  be the number of traitors among the  $n$  generals. A protocol was given for the case  $n > 3t$  and it was shown that  $n > 3t$  is necessary for a solution to exist (Wang, 2014).

A simpler way to explain this problem is by considering that there is a group of generals wanting to attack a city. These generals will face two distinct problems:

- The generals and their respective armies are very far apart which makes coordination very tough due to the lack of the presence of a centralized authority.

- The city to be attacked has a huge army and the only way to win is that the attack should be done by all generals at once.

In order to be successful, generals must flawlessly coordinate and be in sync all the time. The armies on the left of the city must send a messenger to the armies on the right of the city with a message containing all the attack strategies and timing. However, suppose that the armies on the right are not ready for the attack and send back the messenger through the city with another attack timing to the armies on the left. This is where the problem arises taking into consideration a number of things that can also happen to the messenger as well. For instance he could get captured, killed or replaced by another fake messenger by the city. This would definitely lead to a tempered information received by the armies which by its turn results in an uncoordinated attack and total defeat.

This problem if occurred will have a clear impact on Blockchain knowing that the chain is a huge network. Considering the case where a node A is sending 1 Bitcoin from its wallet to another node B. What will guarantee that another node C in this network will not temper with the value and change it to 10 Bitcoins?

In order to solve this issue, what these generals need in this situation, is a robust consensus mechanism that needs to be implemented to insure the win for their army as a unit despite all the possible setbacks in their way.

Hence, to solve the Byzantine Generals problem, the list of the used consensus mechanisms will be presented in the following sections. (Rosic, n.d.-a)

### **2.9.1.1 Proof of Work (PoW)**

Satoshi Nakamoto, Bitcoin's creator, was able to bypass the Byzantine Generals problem by applying the Proof of Work protocol (Krawisz, n.d.).

Proof of work or PoW is a consensus algorithm used in most Blockchain network. PoW is put into operation in order to confirm transactions and to append new blocks to the chain. With PoW, all the miners finalize transactions by competing against each other, and get compensated after finding and proving the solution to a very complicated mathematical puzzle.

In other terms, PoW protocol requires a node to try and solve a hard computational problem in order to validate a batch of transactions, and add them as a new block to the Blockchain (Porat et al., 2017).

The complex problem is a mathematical puzzle that requires a tremendous computational power to be solved. The complexity of this task is very sensitive and evolving, as the network is constantly growing and the algorithms used need more power.

The solution to this problem or mathematical equation is called hash: finding the input given the output (Jakobsson & Juels, 1999).

This algorithm is used in Blockchain when miners solve the puzzle or the complex problem, form the new block and confirm transactions. In addition, PoW allows also to change the complexity of the puzzle based on the power of the network.

This consensus is famously implemented in Bitcoin where the puzzle is a hashcash and the average time for block formation is around 10 minutes.

Another project that uses this algorithm is Ethereum, thus generally speaking, one can say that the majority of Blockchain applications rely on the Proof of Work consensus model.

To have a better understanding of the PoW, this study will show how it works in the context of the Byzantine Generals problem. Back to the above example, where the left side army want to send a message to the right army affirming that the attack will take place on Monday. First, they will attach the nonce to the original text in the message. Then, this nonce appended text will be hashed according to some sort of criteria or condition. If the condition is met, the message will be given to the messenger to be delivered to the other side. If not, the value of the nonce will be randomly changed until getting the desired result. Even if this process is needed to insure that the message will be bullet proof. Its major drawback is time consumption and massive amount of computing power.

In the case where the messenger get caught and the message is altered. According to the hash functions properties, the hash itself will get changed also. When the right side generals get the message and see that it is not starting with the correct amount of 0s (nonce) they can consequently know that something is wrong and call off the attack.

The possible gap here is that no hash function is 100% flawless. There is a case where the message can be tampered with, and the nonce is changed in a way resulting with the required number of 0s. This case is plausible and feasible but extremely time consuming. And, the generals are going to rely on strength in numbers to counter this issue.

Supposing that there are three generals on the left instead of just one general who has to send a message to the ones on the right. For this to be done, they need to create their own cumulative message, hash it and then append the desired nonce (six 0s in this case) to the resulting hash. Noticeably, this is enormously time consuming, but in this case, if the messenger does get caught, the needed amount of time to tamper the cumulative message and find the correct nonce for the hash will be tremendous (a factor of years). Plus, instead of sending one messenger, the generals send multiple messengers, and by the time the city is going through the computation process it will be attacked and destroyed.

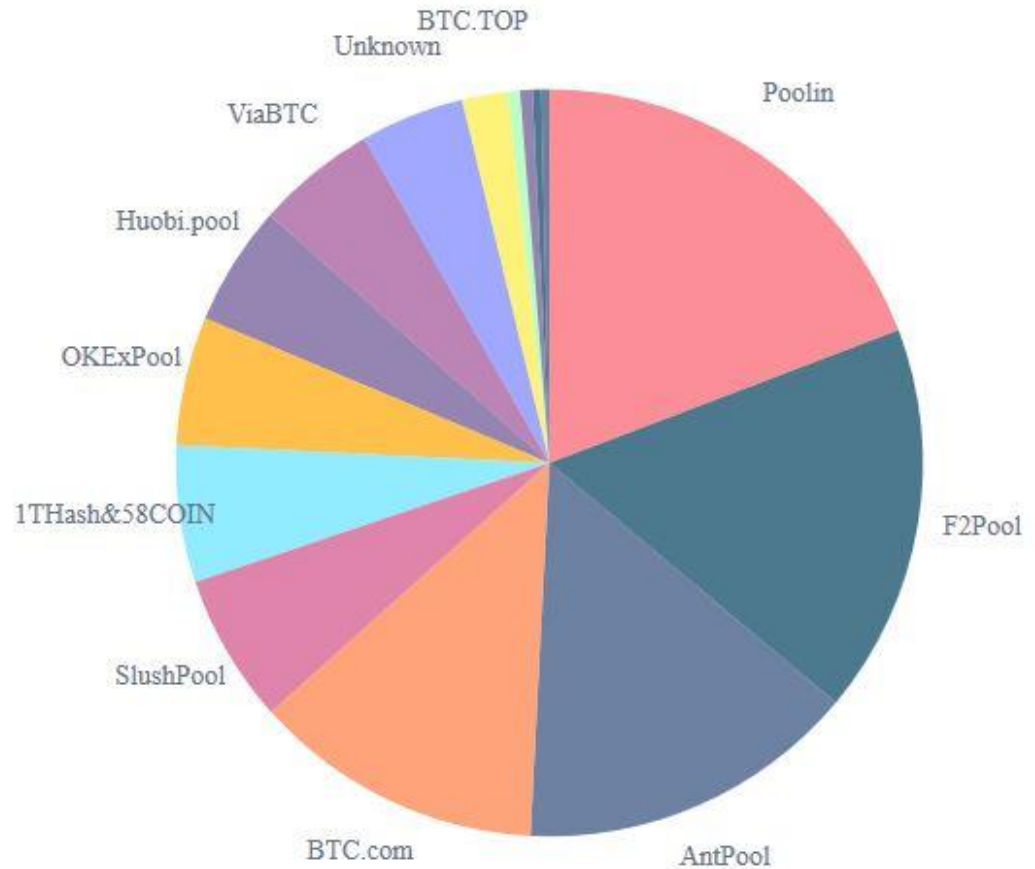
For the generals on the right, the task is very easy. What needs to be done, is appending the messages with the correct nonce already given, hash them, and check whether the hash matches or not. In conclusion, hashing is the core of the process behind PoW. The procedure for finding the nonce for the proper target hash is extremely difficult and time consuming. Though, checking the results should be straightforward and simple (Rosic, n.d.-a).

In the Blockchain case, miners need to solve cryptographic puzzles to mine a block so that it can be added later to the chain. This technique involves a great amount of energy, time and computational power, taking into account that the puzzles are designed in a way to be challenging and difficult for any system. After solving the difficult puzzles, the miner present their block to the network for verification. The verification process in which a decision will be made regarding if the block belongs to the chain is very simple compared to solving the problem.

Unfortunately, even if the Proof of Work consensus mechanism provides a solution to the Byzantine Generals Problem, there are some issues that come with it. PoW is an inefficient process due to the fact that it requires great amounts of energy and power. Plus, miners that can acquire more powerful and faster machines usually have better chance of mining than others.

As a result, Bitcoin for example isn't completely decentralized as it intended to be.

Figure 14 below shows the market share of the most popular Bitcoin mining pools.



**Figure 14: BTC mining pools market share** (*Blockchain Charts, n.d.*)

It can be seen according to the graph that ~65% of the hashrate is divided among 4 major mining pools. And hypothetically, these major mining pools can simply team up and launch a 51% attack (will be explained later) on the Bitcoin network.

### **2.9.1.2 Proof of Stake (PoS)**

In the Proof of Stake based Blockchain networks, a pre-selected group of miners is set to compete to solve the crypto-puzzles with a successful mining probability proportional to the amount of their stakes (Li et al., 2017). After solving the puzzles, miners are encouraged to take part in the verification process in order to achieve and win certain transaction rewards. The newly mined block should be then propagated over the network, and be verified the quickest way possible to decrease consensus propagation delay.

Particularly, PoS is a popular consensus algorithm because it requires mild cost and computing power. The probability of winning a mining competition is determined by a miner's stake, since the difficulty level of the crypto-puzzle for each miner is adjusted according to the amount of their stakes. However, users might be penalized for faulty behaviors (Porat et al., 2017).

The leader in PoS is elected with a probability proportional to the amount of stake it owns in the system. In the simplest case, stake is the amount of currency, but it can also be (for example) the age of the coin that a miner holds (Siim, 2017).

PoS based consortium Blockchain networks rely on two major steps in consensus management, the mining step and mined block propagation for verification step. The designated miners record the new transactions of the user into a block, before competing to solve the puzzle according to their stakes in the mining step. The



fastest miner finding the valid nonce that meets the requirements of the crypto-puzzle will propagate its mined block to the other miners for verification. Once the block is verified and added, the concerned miner will receive the reward for its effort.

On the other hand, since the number of preselected miners is limited, the miners need to propagate the mined block to more validators (Li et al., 2017). Increasing the number of these validators can lead to the elimination of the centralized block verification, and to decrease the impacts of compromised verifiers resulting in a more reliable and secure Blockchain network. In addition, the verifiers can form diverse sets to finish the verification process in a more efficient way. Each miner have to recruit its own verifiers to verify the mined blocks. When the mined block is verified and validated, miners will consequently have to share the transaction fee with their respective verification contributors. If the offered transaction fee by the Blockchain network user is high, the transaction records in the mined blocks can be verified by more verifiers. Still, the more verifiers the more costly and time-consuming process. In most cases Blockchain users should strategically set transaction fees in a way to incentivize miners and save costs.

Ethereum for instance, is planning to a future move from PoW to PoS, because as mentioned earlier the entire mining process in PoS will rely heavily on the validators leading to a more efficient Blockchain network.

In addition, Miners require a lot of energy when using Proof of Work. For instance, a study done in 2005 estimated that one Bitcoin transaction require the same amount of electricity as powering 1.57 American households for one day

(Frankenfield, 2021). These energy costs are paid with Fiat currencies (currencies that are not backed by gold or silver, but rather by the governments responsible for issuing it), leading to continuous pressure on the digital currency value.

Developers are concerned about this problem, and the Ethereum community wants to exploit the Proof of Stake method for a greener and cheaper form of consensus.

The PoS general process will work according to the following steps:

- Each validator will have to lock up some of its coins as stake.
- In block validation, the validators or verifiers will start to discover blocks suitable to be added to the chain. These blocks are validated by placing bets on them.
- In the case where a block is appended, the involved verifiers will get a reward correspondingly to their bets (Rosic, n.d.-a).

Nevertheless, similar to other protocols PoS is suffering from flaws and facing big roadblocks ahead. To be more specific, take the situation where there is a main branch, and another chain which forks from the main one. The problem here is to be able to stop the miner from mining on the bifurcated chain and hence force a hardfork. In this case, any validator can simply put the Blockchain user money in both chains without any fear of any impact at all. And no matter the situation, the user will always win despite if the his/her actions maybe malicious.

### **2.9.1.3 Comparison between PoW and PoS**

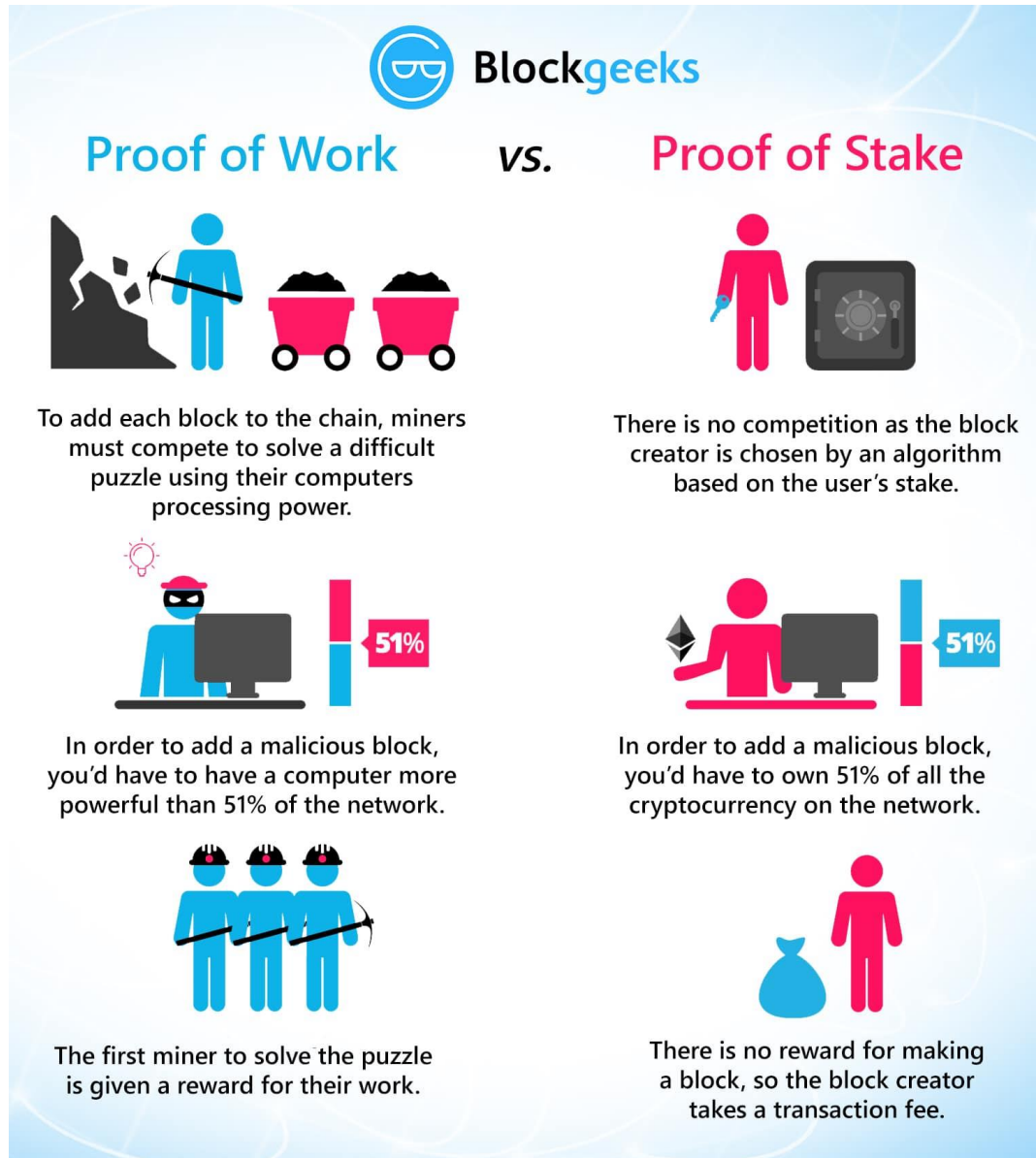
PoS and PoW have the same purpose, but the process using them is quite different.

Figure 15 below, shows a comparison between the two protocols.

- To add a block to the chain: PoW miners must compete to solve the difficult crypto-puzzle using their computers processing power. In contrast, there is

no competition in PoS as the block creator is chosen by the algorithm based on the user's stake.

- In PoW, a node or a mining pool must have a computer more powerful than 51% of all other nodes in order to take control over the network. Comparing to PoS where the node must own 51% of the supply of the cryptocurrency on the chain (this can be used sometimes to add malicious blocks).
- In PoW, the first miner to solve the puzzle will get the reward for its work. Unlike in PoS, there is no reward, and the block creator takes a transaction fee.



**Figure 15: PoW versus PoS** (Rosic, n.d.-b)

While the overall PoS process goal remains the same as PoW, the method of reaching the objective is totally different. In PoW, the miners need to solve hard cryptographically puzzles by using their computational resources. Whereas in POS, instead of miners, there are verifiers, which lock up some of their stake (Ether for example) in the system. Next, the verifiers will bet on the blocks that they are more

likely to be added next to the chain. The time when the block gets added, the verifiers get a reward in proportion to their stake (Rosic, n.d.-b).

To conclude, verifiers in PoS do not rely on their computing power because the only factors that have impact on this protocol are the complexity of the network and the verifier stakes (total number of own coins). Thus, the possible switch from PoW to PoS may provide the energy savings benefit, and a safer network as attacks become very expensive. In addition, PoS algorithm must be as bulletproof as possible because, without penalties, a proof of stake-based network could be cheaper to attack (Rosic, n.d.-b).

#### **2.9.1.4 CASPER protocol: a safer system**

To solve the issues resulting from using PoS, Vitalik Buterin a co-founder of Ethereum created the CASPER protocol. He designed an algorithm that can use the set of some circumstances under which a bad verifier might lose its deposit (Buterin et al., 2019).

CASPER will be a security deposit protocol that depends on an economic consensus system. Verifiers or validator nodes must pay a deposit as a security in order to take part of the consensus and be able to create new blocks. This protocol will determine the exact sum of the rewards to be given to the validators due to its control over the security deposits.

The validator will be accountable for his actions. For example, if he creates an invalid block, the security deposit will be lost, as well as his privilege to be part of the network consensus.

In other words, the security system in this protocol is based on bets. These bets are transactions that will reward their validator with the prize money according to the consensus rules. So, CASPER protocol is based on the concept in which validators will gamble on the basis of the others' bets and leave positive feedbacks that are able to come to a consensus in a timely manner.

How is CASPER different from other Proof of Stake protocols?

The main difference compared to other protocols is that CASPER is designed to work in a trustless system and be more Byzantine Fault Tolerant. Malicious elements will have their stake slashed off. And, any validator who acts in a malicious or Byzantine manner will get directly penalized by taking their deposit away.

Hence, perfectly executing CASPER protocol and PoS will be critical if Ethereum plans to scale up.

### **2.9.1.5 Blockchain Consensus Conclusion**

Without consensus mechanisms, the Byzantine Fault Tolerant decentralized peer-to-peer system would exist.

While, PoW and PoS are the most popular protocols, continuous work is being conducted to come up with newer mechanisms. No consensus mechanism is perfect. Nevertheless, due to the fact that new cryptocurrencies are constantly appearing with their respective protocols, it will always be promising to use the most suitable one.

## 2.9.2 Mining requirements

### 2.9.2.1 Hardware

In order to achieve best results, mining hardware is continuously evolving with time. At first, miners relied on their central processing unit (CPU) to mine. However, they realized after a short time that this was not fast enough and it slowed the systems of the host computer. Thus, they rapidly proceeded to use the graphics processing unit (GPU) in the graphic cards knowing that this process will consume less power and it is approximately 100 times faster in hashing. Later, miners shifted to using a new type of equipment that pushed the performance even higher. Field-programmable gate array (FPGA) processors were used and attached to computers using simple USB connection. These miners used much less power than GPU's and created mining farms as an achievement (*Everything You Need to Know about Bitcoin Mining*, n.d.).

Currently, groups of miners are using application-specific integral circuit (ASIC), and took over the industry completely. The modern ASIC machines mine at exceptional speeds while consuming less energy than all the other technologies.

In Particular, Bitcoin mining is increasing in popularity so does the Bitcoin price. This leads also to a rise in the value of the ASIC Bitcoin mining hardware. Bitcoin mining with anything less than ASIC will consume more power in electricity than what a miner is more likely to earn. It is always required to mine with the best hardware all time.

Equally, the more mining hardware is deployed, the more the difficulty in solving the hash is raised. This makes it impossible most of the time for a miner to compete

nowadays without using an ASIC system. Besides, this technology is constantly evolving, getting faster, more efficient and more productive. Therefore, it keeps pushing the limits of finding what makes the best mining hardware.

Antminer S9 for example has a 0.098 W/Gh power efficiency and an approximate BTC earning per month around 0.3603

### **2.9.2.2 Software**

Special programs or software are needed and used in the mining process. In the case of Bitcoin, the most commonly used mining software are CGMiner, or BFGMiner, which are command line interfaces (CLI) that process commands to a computer program in the form of lines of text (Batabyal, 2020) (Szmigielski, 2016). On the other hand, in the case of Ethereum, in order to connect the required mining hardware to an Ethereum mining pool and to the Ethereum network, miners can use on of the available software like Ethminer, Hive OS, Dual Miner, etc.

### **2.9.2.3 Mining Pools**

Mining pools are groups or coalitions of miners working together as one entity. They combine their hashing power to solve a block, and share all the rewards according to the hashing power of each involved miner. In Bitcoin, without the presence of these pools, mining can take up a couple of years without any benefit in the form of earned rewards. Hence, it is more suitable to share the work and split the reward between miners in a pool. Also, miners can at any time redirect their hashing power to any mining pool that is more suitable for them. Top Bitcoin mining companies are located in China. F2Pool, AntPool, BTCC are among these



pools. It is estimated that these pools own approximately 60% of Bitcoin hash power. Therefore, they are responsible of mining about 60% of all new Bitcoins. Georgia has its share also, BitFury one of the largest producers of Bitcoin mining hardware currently mines about 15% of all Bitcoins (Tuwiner, 2021).

In Ethereum, Ethermine and f2pool sites are the largest Ethereum mining pools at the moment (May, n.d.).

#### **2.9.2.4 Setting up the wallet**

The virtual wallet is a software that anyone can acquire to be used in keeping track of balance and transactions. The wallet holds all the individual's funds, transactions performed and security keys of users.

The Blockchain wallet interface for example shows the current balance for both Bitcoin or Ether tokens, and displays the most recent transactions.

Wallets come in different forms. The most capable one is called full client. It can perform Bitcoin transactions and act as an access to the Bitcoin network. Full clients also store a copy of the Bitcoin Blockchain locally. An example of a full client is the Bitcoin Core software application (Szmigielski, 2016).

Bitcoin Core is an open source project which maintains and releases Bitcoin software called "Bitcoin Core". It consists of both "full-node" software for fully validating the Blockchain as well as a Bitcoin wallet (*Bitcoin Core :: About*, n.d.).

Another good Bitcoin wallet is Copay which functions on different operating systems. In addition, Bitcoins are stored in the wallets by using a unique address belonging to its owner.

Furthermore, wallets allow also Ethereum users to interact with smart contracts on the Ethereum network in addition to storing Ether. For example, MetaMask is a browser extension and mobile wallet for iOS and Android used in Ethereum Blockchains (*Ethereum Wallets / Ethereum.Org*, n.d.).

Moreover, wallets should be at all-time secured from potential threats by enabling for example a multi-factor authentication, or keeping it and backing it up on an offline device.

After following the proper steps in setting up the environment, a miner can start the mining process. Miners must always be up to date with all news and updates related to the system they are involved in.

### **2.9.3 Mining profitability**

The mining process appears to be an easy task. A node sets up a computer to help solve complex math puzzles, and is rewarded for the respective work offered. In reality, it is a complex exercise that needs to be solved flawlessly by any node wishing to mine on the Blockchain.

For instance, the reward for successfully completing a block is 6.25 BTC in mining Bitcoins. In March of 2020, the price of Bitcoin was about \$9,160 per Bitcoin, which means you'd earn \$57,250 ( $6.25 \times 9,160$ ) for completing a block as of the time of writing. This amount can be acquired by any winning miner who successfully solved the complex hash problem.

The rewards for Bitcoin mining are halved every four years. In 2009, when Bitcoin was first mined, mining one block would earn the winning miner 50 BTC. In 2012,

this was halved to 25 BTC. By 2016, this was halved once more to 12.5 BTC. In May 2020, the latest halving occurred and the reward is now 6.25 BTC per block (Derks et al., 2018). On the other hand, in Ethereum, for every successfully mined block, 2ETH are given as a reward.

Moreover, the winning miners will also obtain along with the reward, fees associated with every transaction. These fees serve as another incentive for the miners to do their job. Hence, miners will surely prioritize the transactions with higher fees. The obtained reward is then transferred instantaneously to the Ethereum or Bitcoin wallet linked with the miner or the miner's pool. There are several Ethereum profitability calculators available online in the market. They are provided by services such as CryptoCompare, CoinWarz, WhatToMine and MyCryptoBuddy.

The approximate income or reward can be calculated based on the miner hash rate/power and the electricity consumption. Keeping in mind the costs of the chosen hardware and the possible upgrades on the network bandwidth. Hence, no matter what the system a miner choose, He/She has to take account for the setup, including in some cases , GPUs for instance that can cost up to 700\$ a piece. It is possible to put together a basic rig for some not so popular crypto for around 3000\$. Nevertheless, some miners spend more than 10000\$ on their rigs (Marquit, 2020). On top of building the rig, a miner also needs to take into account that he is going to use quite a lot of power. For example, the electricity fee required in mining one BTC is more than 3000\$ in several states (Marquit, 2020). Hence, any miner needs

to do proper budgeting before setting up the system in order to recoup the original investment and gain profits.

#### **2.9.4 Bitcoin Cloud Mining**

Bitcoin cloud mining enables people to earn Bitcoins without buying mining hardware, Bitcoin mining software, electricity, bandwidth or other offline issues (*Best Bitcoin Cloud Mining Contract Reviews and Comparisons*, n.d.).

Bitcoin cloud mining known also as cloud hashing enables miners to buy the output mining power from Bitcoin mining hardware located in remote data centers. This type of mining is done remotely in the cloud, and removes the hustle encountered by the miners when dealing with installation, hosting, power, upkeep trouble, etc. It will attract a far wider audience including the people who lack the technical background and knowledge required to get into mining.

Cloud mining is supported by companies that set up mining rigs at their own premises. A cloud miner needs only to register and purchase shares (a proportion of the Bitcoin miners hash power) from these companies. Basically, if the miner purchases a higher hash rate, he is expected to receive more coins for what he pays, but this will cost him more (Marquit, 2020). Hence, the mining firms will do all the mining work and give the cloud miner revenues in a consistent way.

However, this type of mining have some major concerns. Fraud is the major one. In addition to lower profit compared to regular mining. Even more, some mining companies halt their operations if Bitcoin's price fall below certain levels.

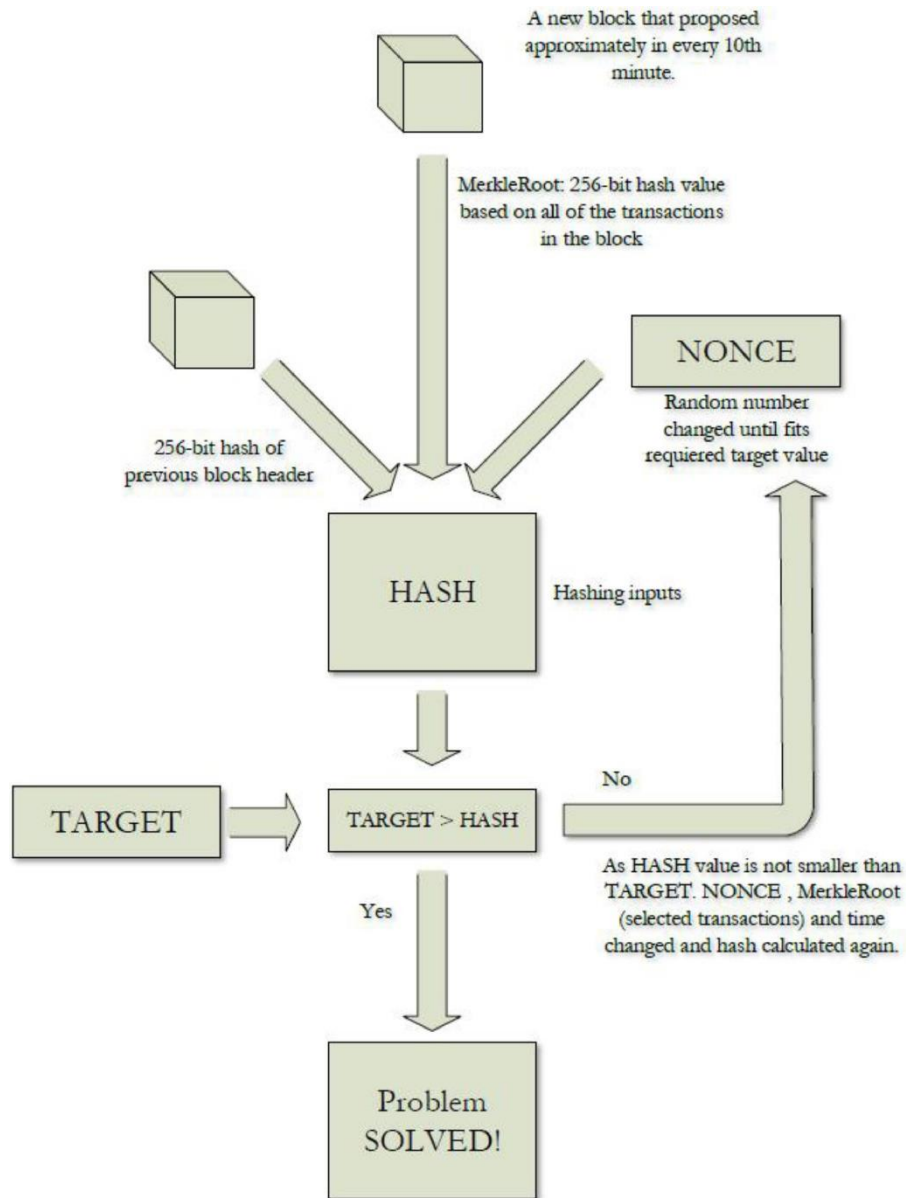
### **2.9.5 Bitcoin mining example**

Bitcoin mining is a decentralized computational process, where transactions are verified and added to the public ledger, known as the Blockchain (Nakamoto, 2009).

Figure 16 below, shows the Bitcoin mining process. The miner calculates the hash of a block of transactions and the summary information of the previous block. Each corresponding block has a nonce value and the miner needs to select randomly a nonce value in order to obtain a hash of the block smaller than a target value, which is periodically recalculated by the network. The arbitrary tries for nonce values to find the valid hash is the known PoW. Once the miner finds the hash that satisfies the required number of zero bits, it diffuses the block to the rest of the network. The other nodes then validate the block and start to create the next block for the Blockchain using the hash of the accepted block.

The winning miner is compensated for its effort with a special transaction. In this case, the miner is awarded BTC whenever a new block is added. The amount awarded with each mined block is called the block reward. The block reward is halved every 210,000 blocks or approximately every 4 years. In 2009, it was 50. In 2013, it was 25, in 2018 it was 12.5, and sometime in the middle of 2020, it will halve to 6.25.

This reward offers an incentive for the miners to participate in this type of network. In addition, the network self-adjusts the difficulty of hash calculations to keep the flow of rewards stable, as a result new blocks are only created once every 10 min on average (Küfeoğlu & Özkuran, 2019).



**Figure 16: Simple diagram showing Bitcoin mining** (Kufeoglu & Ozkuran, 2019)

In a simpler way, this process can be best described by the following steps (Mason, n.d.):

1. Verify if transactions are valid.
2. Transactions are bundled into a block

3. The header of the most recent block is selected and entered into the new block as a hash.
4. Proof of work is completed.
  1. A new block is proposed.
  2. A header of the most recent block and nonce are combined and a hash is created.
  3. A Hash number is generated.
  4. If the Hash is less than the Target Value the PoW has been solved.
  5. The miner receives the reward in Bitcoins and transaction fees.
  6. If the Hash is not less than the Target Value, the calculation is repeated and that takes the process of mining difficulty (explained later).
5. A new block is added to the Blockchain and added to the peer-to-peer network.

The Mining Difficulty Steps are:

1. More miners join the peer-to-peer network.
2. The rate of block creation increases.
3. Average mining times reduce.
4. Mining difficulty increases.
5. The rate of block creation declines.
6. Average mining time returns to the ideal average mining time of 10 minutes.
7. The cycle continues to repeat at an average 2-week cycle.

## **2.9.6 Mining problems.**

### **2.9.6.1 51% Attack.**

In PoW, the miner uses all of its assets to find the nonce in order to generate a predefined pattern of hash. The first miner to find the suitable nonce between its network peers, will add the block to the chain and claim his reward. The system will reward the miner for its effort by generating a specific amount of coins and offering it to the winning miner as prizes. The more computation power used by the miner is high, the more chance of getting rewards is high. Therefore, if a miner or a group of miners possess a large amount of computation power (more than 50%), he/she can control all the activities in the Blockchain for his/her own benefits, whether beneficial or harmful. In this case, the miner can mine a longer chain in faster time than others and as per longest chain rule, this generated chain will have a high probability to be accepted by the network.

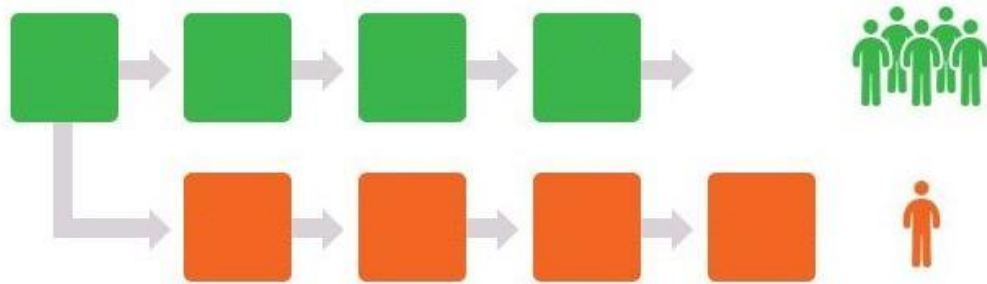
“The 51% attack is a technique which intends to fork a Blockchain in order to conduct double-spending. Adversaries controlling more than half of the total hashing power of a network can perform this attack” (Sayeed & Marco-Gisbert, 2019).

Corrupt miners can use this exploit, initiate the so called 51% attack, and control the Blockchain.



By definition and according to Investopedia a 51% attack refers to an attack on a Blockchain—most commonly Bitcoins, for which such an attack is still hypothetical—by a group of miners controlling more than 50% of the network's mining hash rate or computing power (Frankenfield, 2019).

Thus, to initiate this attack, the unethical miner or the mining pools will continue to mine blocks by selecting transactions from the memory pool. The resulting mined blocks however are not broadcasted to the network. In the meantime, other miners continue to mine simultaneously to add their corresponding blocks on the main chain. On the main chain a corrupt miner can make any type of transaction with other nodes, for example balance transfer to another user to close some sort of a deal, or money conversion and exchange into fiat money. These transactions will be confirmed in the Blockchain and broadcasted to all the nodes in the main chain. The fraudulent miner will by his turn broadcast the other longer chain secretly mined. Consequently, the network will receive this valid longer chain that doesn't contain the transactions the corrupt user did with other nodes and exchanges. The network will accept this longer chain and discard the shorter one as shown in Figure 17 below.



**Figure 17: 51% Attack** (Katrenko & Sotnichek, 2020)

As a result all the transactions made by the false miner will take place. All the transactions done by other miners will be reversed and discarded along with the shorter chain. Thus, the malicious user can double spend his money by making fake transactions around the clock and scamming different targeted individuals in exchange for his profit.

The other network users will be unaware of this attacker activity, given the fact that all transactions are confirmed in the usual manner. Only the transactions made by the corrupt user will be applied knowing that he intentionally mined the second longer chain.

In conclusion, if his attack is made, it will be able to cause numerous damages to the newer and small network. However, older and larger networks are at less risk, due to the fact that the attacker will have to acquire enormous amount of money and mining power to manage the attack.

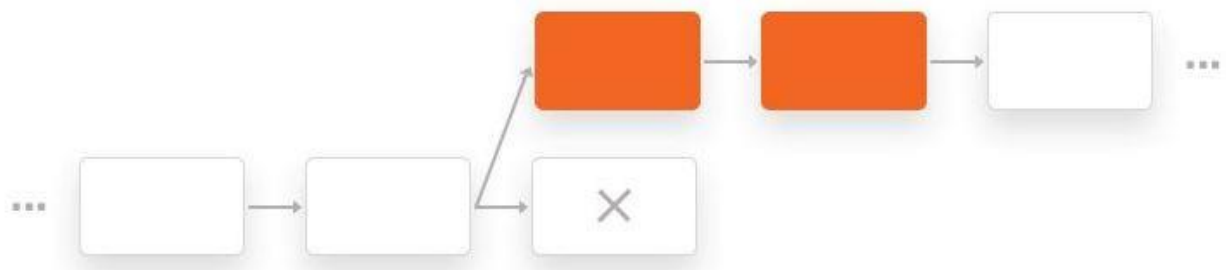
#### **2.9.6.2 Selfish Mining.**

Selfish mining is an attack done by a miner alone or by a mining pool towards other individual miners or mining pools. Selfish miners play by their own rules. These rules are tailored to help them put honest miner's effort to waste, and consequently, increase their own relative reward.

According to Investopedia, selfish mining is a strategy for mining Bitcoin in which groups of miners collude to increase their revenue. Bitcoin was invented to decentralize production and distribution of money. However, selfish mining if used can result in centralizing Bitcoin mining operations. This technique was first

proposed by Cornell researchers Emin Gün Sirer and Ittay Eyal in a 2013 paper. They proved that miners can earn more Bitcoins by withholding newly-generated blocks from the main Blockchain and by creating separate forks when releasing them (Eyal & Sirer, 2018) (Frankenfield & Rasure, 2021).

In brief, after a selfish miner/pool finds a block, he will not propagate it directly as per the standard rules. Instead, he will keep it hidden, so that other miners cannot mine on top of his block. Then he will add or publish the block or his private branch when needed. This is done to invalidate or cancel honest miner's block and gain the rewards as shown in Figure 18 below.



**Figure 18: Selfish Mining Fork** (Katrenko & Sotnichek, 2020)

Selfish miners follow their own set of rules. This set of rules is divided into two subsets according to the miner who find a block. The first subset is when the selfish miner finds a block, while the second subset is when the honest miner finds a block.

The rules followed to perform selfish mining are:

- 1) The selfish miner finds a block:
  - a) If this miner is now ahead of the other honest miners (there was a tie before this time), he will then publish his entire private branch and he will consequently gain the compensation.

- b) Else, in the case where this miner is way ahead of the others, he will continue mining on his own private branch.
- 2) The honest miner or any other miner finds a block:
- a) If this miner is ahead of the selfish miner, and by this, is going to win the reward. The selfish miner will switch to this winning branch directly.
  - b) Else if the results are the same (it's a tie), the selfish miner will reveal his private branch immediately, expecting a win.
  - c) Else if the selfish miner is ahead, he will reveal his branch and win the reward.
  - d) Else if the selfish miner is way ahead of everyone else, he will reveal in this case his first unpublished block and will continue to mine on his private branch.

By applying these rules, the attacker or selfish miner will waste huge amount of resources from honest ones (Azimy & Ghorbani, 2019).

### **2.9.6.3 Fork-after-Withhold (FaW)**

Fork after withholding (FaW) attack combines the selfish mining or Block-withholding attack (BWH) with intentional forks. Similar to the selfish mining, FaW attack is always profitable regardless of an attacker's computational power or network connection state. It also provides superior rewards compared to the BWH attack (Kwon et al., 2017).

FaW is a variation of selfish mining that appears to be more recompensing for the attackers. During a FaW attack, the malicious miner hides a winning block, and

depending on the circumstances, it will either discard it or release it later to create a fork. By delaying the withheld block, it will result an intended fork (a block submission collision) with a third-party miner when it is triggered to submit it (Chang & Park, 2019).

In addition, FaW increases the attacker reward, since he will compromise the victim's pool while having a separate reward channel in a main pool. Hence, the attacker in the main pool will not share the reward with others, and is consequently motivated to behave according to the consensus protocol. Particularly, he behaves like an honest miner who does not withhold blocks and submits them on time according to the standards. On the other hand, in the victim pool, the attacker will share the rewards with other miners while withholding the block (no contribution to the pool). In other words, he will pretend to contribute and share the block reward if the other miners within the pool find a block.

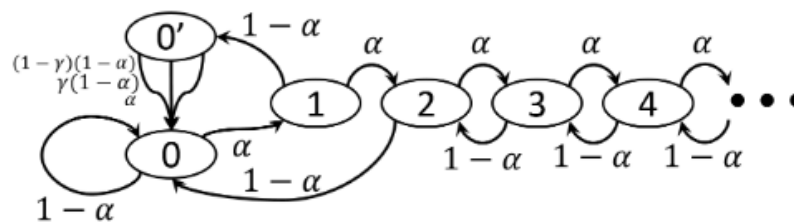
Unlike selfish mining, the FaW attack is always profitable. Detecting it is harder than detecting selfish mining attackers, even though the FaW attack does utilize intentional forks as well (Kwon et al., 2017).

### **2.9.7 Solution to the selfish Mining Problem.**

This research will focus mainly on the selfish mining problem. Previously proposed solutions will be presented in the following sections, precluding to suggesting a robust solution presented in the following chapter that will help in putting an end to this behavior.

### 2.9.7.1 Majority is not enough: Bitcoin Mining is Vulnerable.

This solution suggested a convenient modification to the Bitcoin protocol that will protect Bitcoin in the general case. If used properly, it will forbid the use of selfish mining by pools that own less than 0.25 of the resources (mining power) as shown later. This threshold is lower than the wrongly assumed 0.5 bound, but certainly better than the current situation where any group of any size can compromise the whole system. In this suggested solution, researchers analyzed the expected rewards for a system where the selfish pool has a mining power of  $\alpha$ , and the honest ones of  $(1-\alpha)$ . Figure 19 clarifies the growth of the system as a state machine. The states of this system symbolize the lead of the selfish pool; in other words it is the difference between the number of blocks in the selfish pool's private branch and the length of the public branch. Zero lead is disjointed to states 0 and 0'.



**Figure 19:** State machine with transition frequencies (Eyal & Sirer, 2018)

State zero is where there are no branches or fork; i.e. there is only one single, global, public longest chain. State zero prime is the state where there exist two public branches of length one: the main public branch, and the branch that was once private to the selfish miners, which is published at some point to match the length of the main branch.

The transitions in the Figure above relate to mining events done by the selfish pool or by any other miner. Keeping in mind that these events occur at exponential intervals with an average frequency of  $\alpha$  and  $(1-\alpha)$  correspondingly.

The expected rewards from selfish mining are analyzed by taking into account the frequencies associated with each state transition. Several cases can be considered when describing the associated events that trigger state transitions.

When the selfish pool has a private branch of length one and the other miners mine one block, this pool will publish its branch straightaway, which results in two public branches of length one. All Miners belonging to the selfish pool will immediately mine on the pool's branch, in order to realize block discovery on this branch, and to achieve a reward for the pool. Following the standard Bitcoin protocol, honest miners by their turn will mine on the branch they heard of first. The ratio of honest miners that choose to mine on the selfish pool's block is designated by  $\gamma$ , and the other  $(1-\gamma)$  for the honest miners on the other honest branch.

For state  $s = 0, 1, 2 \dots$  with frequency  $\alpha$ , the selfish pool mines a block and increases its lead by one to  $s+1$ . For state  $s = 3, 4 \dots$  with frequency  $(1-\alpha)$ , the honest miners mine a block and the lead is decreased by one to  $s-1$ . If the honest miners mine a block when the lead is two, the selfish pool publishes its private branch, and the system drops to a lead of zero. In the other situation, when the lead is one, if the honest miners mine a block, the state zero prime will be achieved. From this state, there are three possibilities that all lead to state 0 with total frequency one.

The list of possibility is:

- The pool mines a block on its private branch (frequency  $\alpha$ )
- The honest miners mine a block on the private branch (frequency  $\gamma(1-\alpha)$ )
- The honest miners mine a block on the public branch (frequency  $(1-\gamma)(1-\alpha)$ ). (Eyal & Sirer, 2018)

In addition, a probability distribution over the state space provides the basis for evaluating the revenue obtained by the selfish pool as well as by the honest miners. Hence, the revenue for finding a block is gained by its miner if and only if this block ends up in the main chain.

After certain investigation, the planned branching brought on by the selfish pool leads the honest miners to waste their resources by working on blocks that end up outside the Blockchain. These dropped blocks are known as orphaned blocks. Thus, this will lead consequently to a drop in the total block generation rate. The mining difficulty is adapted in the protocol such that the mining rate becomes one block per 10 minutes on average at the main chain. Therefore, the revenue rate of each miner is directly related to the ratio of its blocks out of the blocks in the main chain.

In this experiment, the model was done using a simulator to simulate 1000 miners working at identical rates. A subset of  $1000\alpha$  miners running the selfish mining algorithm, while the others follow the standard Bitcoin protocol. In addition, block propagation time is also assumed to be negligible compared to the mining time. In the case of two same length branches, honest miners are divided in a way such that a ratio of  $\gamma$  of them mine on the selfish pool's branch while the rest mine on the other branch.

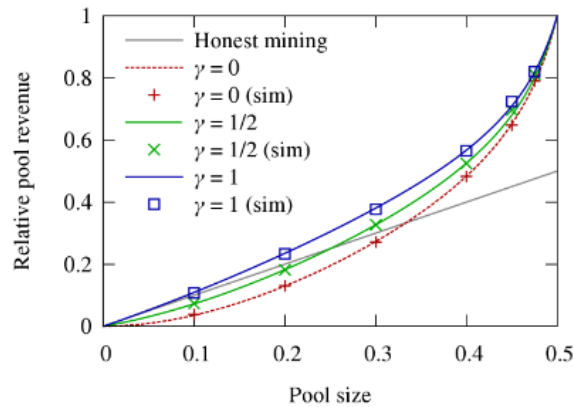


When the selfish pool's revenue is larger than  $\alpha$ , this pool will earn more than its relative size. As a result, its corresponding miners will earn more than their relative mining power.

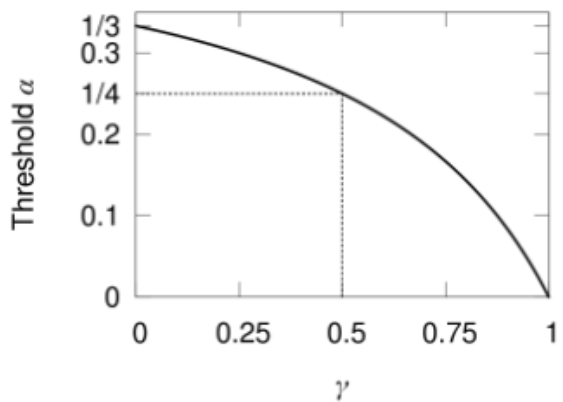
The below observation is that “a pool of size  $\alpha$  obtains a revenue larger than its relative size for  $\alpha$  in the following range”:

$$\frac{1 - \gamma}{3 - 2\gamma} < \alpha < \frac{1}{2} \quad (\text{Eyal \& Sirer, 2018}).$$

This observation prove that if  $\gamma$  increases, the mining resource threshold for selfish mining decreases, but if  $\gamma$  decreases, the threshold increases, raising the difficulty of selfish mining.



**Figure 20: Pool Revenue** (Eyal & Sirer, 2018)



**Figure 21: Relation between  $\alpha$  and  $\gamma$**  (Eyal & Sirer, 2018)

In Figure 20, the pool’s revenue is illustrated for different  $\gamma$  values with pool size ranging from 0 to 0.5. Noting that the selfish pool is only at risk when it holds secretly just one block, and the honest miners might publish a block that would compete with it.

For  $\gamma= 1$ , the selfish pool can quickly propagate its block if the others have already found their own branch, so that all honest miners would still mine on the selfish pool’s block. This is the case where the pool will take zero risk when following selfish mining algorithm, and its revenue is always better than the honest algorithm. Hence, when the threshold is zero, a pool of any size will certainly benefit by doing selfish mining.

For  $\gamma= 0$ , the honest miners will always publish and propagate their block first, and  $\alpha$  is one third according to Figure 21 which shows the threshold as a function of  $\gamma$ . With  $\gamma= 1/2$  the threshold is at  $1/4$  (Eyal & Sirer, 2018).

It is also noted in Figure 20 that the slope of the pool revenue as a function of the pool size is greater than one above the threshold. This indicates that for a pool

running the selfish mining procedure, the profits of each pool member rise with pool size, taking into consideration that pools are larger than the threshold.

At this stage, honest or rational miners will favor joining the selfish pool in order to increase their revenues. Besides, by accepting new members, the selfish pool's members would increase their own revenue. Therefore, the selfish pool will increase in size, and become a majority. Once this majority is reached, this pool will control the Blockchain. Therefore, selfish mining becomes unnecessary at this point, since the selfish pool is faster than the others and is collecting all the system's revenue by switching to this modified Bitcoin protocol (where blocks generated outside this pool are ignored). Thus, the Bitcoin will not be decentralized as it was originally intended to be.

The simple solution proposed is a backwards-compatible change to the Bitcoin protocol to address the problem and raise the threshold. Explicitly, when competing branches are learned by a miner, the miner by its turn should propagate all of them, and choose randomly one to mine on. For the case of two branches of equal length one, it would result in half of the nodes mining on the private selfish branch and the other half mining on the public branch. This yields to using  $\gamma = 1/2$ , which gives in turn a threshold of  $\alpha = 1/4$ .

Each miner applying this modification decreases the selfish pool's ability to increase its power and profits. This enhancement will not introduce new weaknesses to the protocol. At present, the choice for each miner is arbitrary, and determined by the network when there is a fork of equal length. The change

presented explicitly randomizes this arbitrary choice, and therefore does not introduce new vulnerabilities (Eyal & Sirer, 2018).

### **2.9.7.2 Fresh Bitcoins, a Solution for the Honest Miner.**

As introduced and stated in the first solution, the success of selfish mining depends on two parameters:  $\alpha$ , which represents the mining power of the selfish mining cooperation and  $\gamma$ , the ratio of the honest mining power used to mine on the block released by the selfish miners group during an occurring block race. Knowing that the mining power is the percentage of computational power that a particular miner or mining pool controls out of the total computational power of all the miners (Heilman, 2014). The block race will happen after the announcement of two blocks with same parent at approximately the same time. Miners will be obliged to choose the firstly received block, and not retransmit the second block in compliance with the Bitcoin protocol. Ever since these two blocks were announced at roughly the same time. Some miners will see one block and other miners will see the other broadcasted block first. This will cause a fork into two branches in the Blockchain. It can be viewed that the minimum value of  $\alpha$  is sufficient such that selfish mining is successful as the security threshold for a particular  $\gamma$ . These two variables are inversely proportional, in other words when the value of  $\gamma$  increases,  $\alpha$  will decrease and vice versa.

Accordingly, the previous study that was done by Eyal and Sirer, if  $\gamma = 0$ , then selfish mining is gainful at  $\alpha \geq 0.33$  or 33%, else if  $\gamma = 0.99$  then selfish mining is profitable at  $\alpha \geq 0.009$ . This former study suggested also a defensive solution

against selfish mining by fixing  $\gamma$  to 0.5. Thus, this raised the threshold for a selfish union to be profitable to at a minimum of 25% or  $\alpha \geq 0.25$ .

In this solution, Freshness Preferred (FP) is introduced as a new mining strategy, which is tweaked and designed to defend against selfish mining aiming to decrease the profitability of selfish miners by using unforgeable timestamps to penalize the selfish miners that withhold blocks. By using FP, the threshold of the minimum share of mining power necessary to profitably selfishly mine will be raised from 25% to 30% (Heilman, 2014).

Freshness Preferred has some rules that need to be followed by a miner in order to succeed and gain the particular profit.

At the event when an FP miner receives two blocks within fraction of seconds between each other noted w:

- If the two blocks are from branches of equivalent length, the miner will accept the block with the most recent valid timestamp and certainly rejects the other block.
- Else if the two blocks have equal timestamps, the miner will definitely choose the block received first.
- Otherwise, the miner will accept the block from the branch of greater length.

In all other events, the miner will behave according to the standard Bitcoin protocol.

In FP, this protocol is tweaked in a way that, the FP miner prefers the block having the most recent timestamp, rather than accepting the block which arrives the earliest

to the miner. The main goal in this solution is to ensure that when a block race happens, it will be won by the nodes that have the most recently created blocks. Thus, by withholding blocks, the selfish miners will have a reduced percentage compared to the other honest miners mining on their withheld blocks. This percentage of honest miners that mine on a selfish block is denoted by  $\gamma$ . And according to the study made by Eayl and Sierer reducing  $\gamma$  will increase the threshold for selfish mining to be effective.

A block race is modeled between a selfish and honest block, taking into account that all the honest miners have conformed to the FP strategy. The selfish block referred as  $B_s$  is discovered at time  $D_s$ , and at some later time  $D_h$ , any honest miner discovers a block  $B_h$  (Heilman, 2014). The selfish pool will always react to the publication of  $B_h$  and will release directly a  $B_s$  in response. This block race is evaluated from the perspective of the FP miner who learned about  $B_s$  at a time noted as  $L_s$  and  $B_h$  at a time noted as  $L_h$ .

The heuristic rule used in this solution is to “overestimate the attacker and underestimate the defender”, it is assumed also that there is no propagation delay for the selfish pool, so that this pool will instantly learn about the honest block at the discovery time  $D_h$ , compared to the honest miners that have a lengthy propagation delay of time  $pd_h$ . In addition, the selfish pool in this solution is allowed to win all timestamp ties.

Under these rules the FP-miner learns about  $B_s$  directly when  $B_h$  is discovered, but doesn't learn about  $B_h$  until sometime later due to the propagation delay  $pd_h$ .

The block race occurs when both the blocks are released in the same window of time. This block race time window is denoted by the parameter  $w$ . If the two blocks are released with time interval more than  $w$  seconds apart, the first block released will always win the race regardless of the timestamp.

It is also assumed in this solution that the block race window  $w$  is larger than the propagation delay  $p_{dh}$ . The difference between the time a miner learns about  $B_h$  and  $B_s$  will always be smaller than  $w$ . Therefore, it will always be the case within the block race window, regardless of when  $B_h$  happens. Thus, the FP miner will accept the proper competing block based on its respective timestamps. If  $T_s$  is older than  $T_h$ , then the FP miner will choose the honest block ( $T_s$  and  $T_h$  are the timestamps for selfish and honest blocks). Contrarily, when  $T_s = T_h$ , then the FP miner will prefer the selfish block since the selfish block is allowed to win at all time. Since, by definition the selfish block can never have a more recent timestamp than the honest block, as it was discovered earlier than the honest block.

Moreover, the probabilities of accepting a selfish block by an FP miner and discovering  $B_h$  by an honest miner within the same timestamp as  $B_s$  are equal and depend on two factors: 1) the increment of the timestamp and 2) the per second rate at which the honest miners discover new blocks (Heilman, 2014).

In the case where the unforgeable timestamp is compromised and the selfish pool is capable of forging timestamps for future use. It is shown that FP mining strategy is still strong and defensive against selfish mining because the selfish miners must still commit to a timestamp within the block they discover. In this case of the block race, where the selfish miners instantly react to the publication of the honest block,

these miners can choose a certain timestamp such as  $T_s \geq D_s$ . Thus, allowing  $T_s > T_h$ . In addition, it is assumed that the selfish pool can predict the propagation delay of the honest miners, and choose  $T_s$  in a way to maximize their chance of winning the block race. In other words, the pool will choose a timestamp of  $T_s = D_s + w + p_{dh}$  to guarantee that any honest block that reaches the FP-miner prior to  $T_s$  will be older than  $T_s$  but still within the window  $w$  chosen to be equal to 120 seconds (Heilman, 2014).

By applying this new defense mechanism, this solution showed that the minimum share of mining power necessary to profitably selfish mine is raised from 25% to 32% and to 30% if the selfish pool has gained the ability to forge the correct timestamps (Heilman, 2014).

### **2.9.7.3 ZeroBlock: Preventing Selfish Mining in Bitcoin.**

A third approach to defend against the selfish mining attack or block withholding strategy is to apply the ZeroBlock solution. When applying this solution, a selfish miner or selfish mining pool cannot achieve more than the expected reward. And only in mediocre cases, a selfish miner can still intentionally create unprofitable forks. These forks are unbeneficial, because they do not lead to any extra reward for the selfish mining pools. but also reduce the likelihood to earn unexpected rewards regardless of the pools mining power.

The key idea in this solution is by the time any selfish miner holds a block privately for a certain amount of time greater than a fixed interval (expected time calculated by honest miners). This withheld block will be rejected by all the honest miners



accordingly. Since, each block must be generated and received by the network within a maximum acceptable given time.

Each miner will compute locally this time interval based on the expected delay for a block mining (depends on the difficulty of proof-of-work), and on the information propagation time (60 seconds) in the Bitcoin network (Solat & Potop-Butucaru, 2016a). The expected time is calculated according to the following formula:

$$\text{Expected Time} = \text{BGT} + \text{BT}.$$

Where BGT is the block generation time, and BT is the block propagation time (Solat & Potop-Butucaru, 2016b).

In addition, the honest nodes will entirely perform their calculations in rounds with the length equal to the expected time (Solat & Potop-Butucaru, 2016b). Besides, unlike the process used in Bitcoin where the entire chain is broadcasted, nodes only broadcast the new block also known by the head of the chain. Therefore, the honest miner will either receive a block, or broadcast a block within this expected amount of time. Otherwise, a dummy block called Zeroblock will be generated by this honest node, and then added consequently to the local chain.

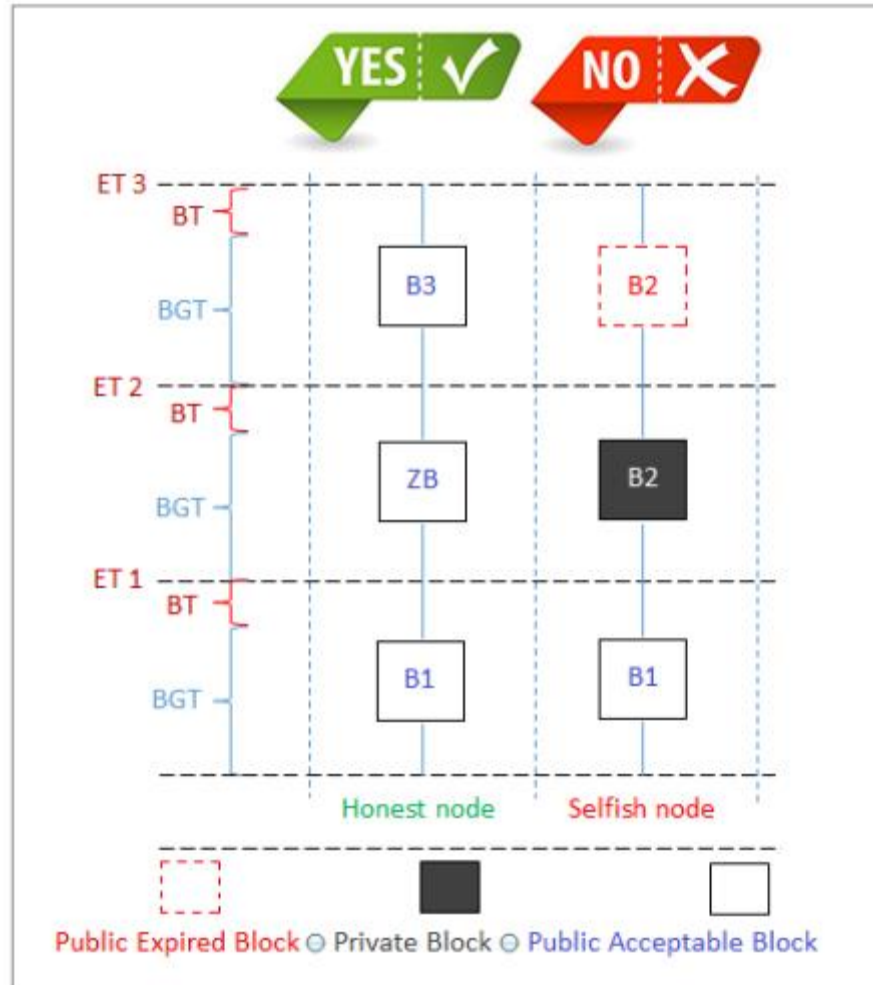
In detail, in the initialization phase of this algorithm, all the nodes participating in the network agree on the first block known as the genesis block. Each node has a local chain which initially is equal to the genesis block (Solat & Potop-Butucaru, 2016b). All the nodes have in common an access to a shared clock, and every single node maintains locally a definite variable LTime, which is a seconds counter (starting at 0) that is updated at each time second. In addition, each node keeps track locally of the variable ET. ET is initially equal to zero, and will be increased

with the value of the Expected Time at each loop. Furthermore, all the nodes keep a Boolean parameter in order to know if a new block is being generated or no. The value of this FlagNewBlock will be changed to True as soon as a new block is generated.

After this phase, ZeroBlock algorithm starts an infinite loop, where the ET value will be updated by using a refresh() function as follows:

$$ET = ET + (BGT + BT)$$

While the LTime counter is less than or equal to ET, an honest miner checks if there is a new block. In this case, the node investigates if its PoW has been computed properly. After verifying that the new block head includes the proper hash of the current local chain, this newly created block becomes the head of the local chain. Otherwise, if during the Expected Time, there is no new block, the honest miner generates a ZeroBlock that comprises a fixed value hash and adds this new block to its local chain (Solat & Potop-Butucaru, 2016b). This generated block will aid in preventing block withholding, knowing that any next block will automatically include the hash of the local chain which includes also the hash of this ZeroBlock. As a result, if a selfish miner or a selfish mining pool chooses not to reveal a block during the expected time corresponding to the generation of this ZeroBlock, it will not be able to use this expired block by consequence. Besides, in the case when any honest node receives more than one new block during the ET interval, it will certainly accept the first one.



**Figure 22: ZeroBlock generation to prevent block withholding** (Solat & Potop-Butucaru, 2016a)

Figure 22 shows a simple representation of ZeroBlock usage. Block B2 is generated by a selfish miner and kept private until ET2 hence it has expired. Accordingly, PoW of B2 at ET3 includes only the hash of B1 and consequently will be rejected by the honest nodes. ZB the ZeroBlock is generated by the honest node that didn't receive any new block until ET2. The PoW of B3 at ET3 includes the hash of B1 and the hash of ZB. Then, the system will accept this block according to the proposed algorithm.

To conclude, this solution clearly shows that when using Zeroblock, none of the honest miners or nodes will accept blocks on the selfish miners' chain. Thus, selfish miners will not be able to impose on honest miners to work on their private chain later to become the public chain. Therefore, this approach will prevent block withholding or selfish mining, resulting in an impossibility to fork the chain intentionally in any Blockchain application. Furthermore, freshly joined nodes in any system using ZeroBlock will always be able to retrieve the correct chain provided that the majority of nodes are honest (Solat & Potop-Butucaru, 2016b).

All the above solutions presented work flawlessly in defending against selfish mining. However, this research will present a completely different approach that will hopefully stop this behavior by imposing penalties on selfish miners according to the frequency of their acts.

## **Chapter 3: Optimum Penalty system against selfish mining**

This research will introduce a new technique to defend against selfish mining. This approach will be based on deducting the rewards that a miner acquire after solving the PoW, transmitting the block to be validated by the system, and adding the block to the chain. The current reward obtained for Bitcoin mining is 6.25 BTC per block. This value is halved approximately every four years. When Bitcoin was first mined in 2009, mining one block would earn the winning node 50 BTC. Back to 6.25 BTC value, in November 2020, the price of Bitcoin was about 17900\$, which implies that any winning miner or mining pool will earn 111.875\$ ( $6.25 * 17900$ ) for completing a block. Thus, this is not a bad incentive to solve the complex hash problem and adding blocks to the Bitcoin Blockchain.

### **3.1 Introduction**

Incentives has led some selfish miners to work in an unorthodox mode in a way to try to fool the other nodes and get all the rewards that can be collected. These selfish miners will withhold mined blocks privately and reveal them at opportune moments. They will intentionally fork the chain, then add their privately withheld mined blocks, and by consequence win the rewards causing other honest miners to lose their time, energy, and computation power. Moreover, this losing block mined by the honest miners will become an orphan block and will not be added to the Blockchain.

The aim of this research is to find the optimum penalty system to guarantee stop selfish mining and to give an equal opportunity for all miners to get the proper rewards for their work. In other terms, the total reward should be uniformly distributed to every participating miner. And, selfish miners will be penalized according to their actions whether they selfish mine recurrently or occasionally.

The main purpose is to find the proper calculation to reprimand selfish miners and to find the proper solution to defend against selfish mining or block withholding.

Selfish mining can occur if any node abides by a specific set of rules.

These rules are:

- 1) The selfish miner finds a block:
  - a) If this miner is now ahead of the other honest miners (there was a tie before this time), he will then publish his entire private branch and he will win consequently gain the compensation.
  - b) Else, in the case where this miner is way ahead of the others, he will continue mining on his own private branch.
- 2) The honest miner or any other miner finds a block:
  - a) If this miner is ahead of the selfish miner, and by this, is going to win the reward. The selfish miner will switch to this winning branch directly.
  - b) Else if the results are the same (it's a tie), the selfish miner will reveal his private branch immediately, expecting a win.
  - c) Else if the selfish miner is ahead, he will reveal his branch and win the reward.

- d) Else if the selfish miner is way ahead of everyone else, he will reveal in this case his first unpublished block and will continue to mine on his private branch.

In case of generation of a new block by the honest miners, (1) if the size of honest branch is longer than the selfish branch, then the selfish cartel tries to set its private branch equal to the public branch. (2) If the selfish branch is one block more than the public branch, then selfish miners publish their private chain completely (3). Else if the selfish branch is more than one block longer than the public branch, then the selfish miners publish only the head of their private branch.

In case of generation of a new block by selfish miners, they keep this new block private and in case of a competition with the honest miners, they publish their private branch to win the competition.

In general any case can happen, this study will not tackle the case where the two chains are of equal length which is one honest block and one selfish block. Knowing that, in this tie any miner can win in this block race. This situation can happen at any time where the honest miner finds a block and the selfish miner immediately reveal his private chain consisting of one block. However this research will concentrate only on finding the optimal solution to the event where the selfish branch is only one block ahead of the public branch. For example, if any honest miner publishes a block, any random selfish miner will publish his private chain consisting of two blocks.

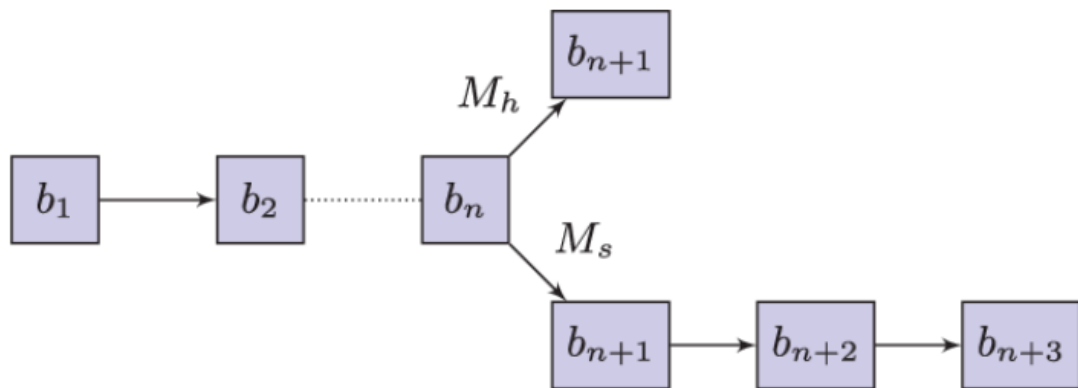
### 3.2 Original Work

Selfish mining can occur any time a selfish miner decides to sabotage the system and create a fork in the main long chain. This research will only focus on the case where the intentionally created fork in the main chain produces two separate branches. An honest branch consisting of one block, and another selfish branch comprising of two blocks. The proposed resolution is to create a solid penalty system that penalizes selfish miners and generalizes the outcome of this practice. This penalty system fines a random selfish miner no matter the frequency of his actions during a fixed timestamp. A hasty selfish miner will be penalized like any other selfish miner that exercises selfish mining occasionally. Nevertheless, the hasty selfish miner will consequently lose more than a slow selfish miner, due to the fact that he already invested in more power and energy to generate blocks.

At any moment, selfish mining can be approached by any dishonest miner with varying the speed in his method. Selfish mining can vary from one time up to six times in an hour timeframe. Hence, this penalty system aims to find a suitable percentage of the block reward that will stop selfish mining in all the cases whether rapid or slow. By finding this fixed constant percentage value, the block reward will be gradually reduced to zero. Thus, this ultimate solution will be finding a proper percentage value applied on individual block rewards during every fork so that the total reward for selfish miners will be zero for a certain amount of time. Therefore, the act of selfish mining will become unnecessary, time consuming, and unprofitable. Knowing the fact that any random selfish miner will be eventually wasting all his hashing power, money, and energy without getting any reward. In



addition, hasty selfish miners will consequently loose more than slow selfish miners because they invested more hash power to create more forks and blocks. As a result, no selfish miner will be granted reward when implementing this penalty system. However, on a long run a slow selfish miner will lose less than a speedy selfish miner in terms of total loss. Knowing that, a hasty selfish miner is intentionally forking the system six times. And, the slow selfish miner two times. Like mentioned before, each fork is dividing the main chain into an honest branch consisting of block, and a selfish branch consisting of two blocks. Figure 23 depicts an example of this type of fork used for the computations.



**Figure 23: HardFork in Selfish Mining** (Saad et al., 2020)

To obtain the result, this solution took a special case where selfish mining can occur during a 60 minutes time frame. In the course of this interval, any random selfish miner can do this technique in a: fast, moderate, average, slow, or one time shot.

Thus, every ten minutes, a hasty or fast selfish miner will create a hardfork containing two private blocks represented in red color as illustrated in Figure 23

above. Leading to the creation of six forks in total during the course of one hour operation. Twelve private selfish blocks in total will be published compared to six honest blocks for the normal case. Whereas, another arbitrary miner, working in a moderate pace, will create the fork every 15 minutes causing a total of four forks in 60 minutes time. In contrast, an average selfish miner will create a fork every 20 minutes, resulting in a maximum number of three forks in an hour. Only two forks for the slow miner, compared to just one fork for the one shot selfish miner. Table 2 below will represent and clarify all these cases.

Type of Selfish Miner	Number of forks per hour	Rate
Fast	6 forks	1 fork per 10 minutes
Moderate	4 forks	1 fork per 15 minutes
Average	3 forks	1 fork per 20 minutes
Slow	2 forks	1 fork per 30 minutes
One Time	1 fork	1 fork per 60 minutes

**Table 2: Selfish Miners Behavior Comparison.**

In this solution, the block reward is fixed to 6.25 BTC since Bitcoin mining is taken into consideration. In Bitcoin, as of May 2020, the reward given for mining a block is 6.25 BTC per block. In particular, a block reward is made up of newly created Bitcoins (6.25 BTC) plus transactions fees. Although, in the calculation done, transaction fees are eliminated from the equations, and only 6.25 BTC are used as block reward for every block validated and added to the chain. Any other

cryptocurrency mining like Ethereum (ETH), Cardano (ADA) can be used in the calculation instead. However, this study focused on Bitcoin in specific knowing the fact that Bitcoin is the most popular and still on the rise.

According to the general Bitcoin protocol rules and in absence of this proposed penalty system, when a fork is imposed, the longest branch consisting of 2 blocks will always be admitted by the system, whereas the shortest branch composed of 1 block will be discarded permanently. Thus, the classic reward for this case will be 12.5 BTC ( $6.25 * 2 = 12.5$ ) typically waged to the selfish miner. Besides, any other competing honest miner on the other hand will waste his hash power, money and energy on a rejected short branch containing a block fated to be abandoned (orphan block).

In this study two different scenarios have been taken into consideration, in order to come up with an optimal solution to oppose against selfish mining.

In the first scenario S1, a specific percentage is continuously deducted from every block reward for every fork imposed. A deduction percentage is constantly distributed and applied to each and every block every time a fork occurs. Thus, the block reward is decreased by this constant specific step every time a fork and two blocks are introduced. In other terms, for a step of 5%, after the first fork, the first block reward will be 95% of 6.25 BTC, and the second block reward will be 90% of 6.25 BTC. For the same miner, after the second fork, the first block reward will be 85% of 6.25 BTC, the second block reward will be 80% of 6.25 BTC accordingly. In addition according to the selfish mining frequency used, the reward

will decrease every time a fork is presented for the same 5% step in this case as shown in Table 3, and other reduction steps in other cases.

Reduction step: 5%												
Forks	1 <sup>st</sup> Fork		2 <sup>nd</sup> Fork		3 <sup>rd</sup> Fork		4 <sup>th</sup> Fork		5 <sup>th</sup> Fork		6 <sup>th</sup> Fork	
Blocks	B1	B2	B1	B2	B1	B2	B1	B2	B1	B2	B1	B2
Reward	95%	90%	85%	80%	75%	70%	65%	60%	55%	50%	45%	40%
BTC	5.9375	5.625	5.3125	5	4.6875	4.375	4.0625	3.75	3.4375	3.125	2.8125	2.5

**Table 3: Block rewards distribution according to 5% deduction in S1 (Fast SM)**

According to last row in Table 3, the sum of block rewards for the six forks is 50.625 BTC in an hour compared to the sum without penalty equal to 75 BTC. When comparing these two values, it is found that a 32.5 percent loss is applied in this case. Hence, when applying a 5% reduction on every block reward, a hasty selfish miner will lose 32.5% of the normal block reward. Thus, in the 60 minutes timeframe, this hasty miner lost approximately 4 of his 12 blocks added.

Furthermore, the Reduction step was steadily increased each time by 5% from 5% to 100% in order to find the right percentage window to stop every selfish mining activity depending on its speed and frequency. Hence, the main objective of this exercise is to find and apply a precise step that will guarantee the total loss of a selfish miner. To elaborate more, for the same hasty miner case, this step was augmented gradually, from 5% to 100%. And, it was found that a reduction of 50% was sufficient and promises to stop selfish mining in this case as shown in Table 4 below.

Reduction	Percentage lost	Blocks lost out of 12
5	32,5	3.9
10	62.5	7.5
15	76,25	9.15
20	83,33333333	10
25	87.5	10.5
30	90	10.8
35	92,08333333	11,05
40	93,33333333	11.2
45	94,58333333	11.35
50	95,83333333	11.5
55	96.25	11.55
60	96,66666667	11.6
65	97,08333333	11.65
70	97.5	11.7
75	97,91666667	11.75
80	98,33333333	11.8
85	98.75	11.85
90	99,16666667	11.9
95	99,58333333	11.95
100	100	12

**Table 4: Relationship between reduction step and percentage lost (Hasty SM).**

Table 4 shows the relationship between the reduction step and the percentage lost for a hasty selfish miner. It can be shown that when the reduction step reaches 50%,

the percentage lost was 95.83333333% or 11.5 blocks out of 12 lost (roughly 12 blocks). Thus, by using this 50% reduction step in this case, and by tweaking or modifying the Bitcoin algorithm to accommodate this penalty system value, it will be irrelevant to practice fast selfish mining. Hence, the efforts done by the selfish miner went to waste, and his threat to the system will be completely eliminated. Not only, he didn't acquire the potential reward but he wasted all his systems resources mining on these becoming non-profitable blocks instead.

As a result, these blocks will be validated and added to the Blockchain main chain as per general rules, but without any remuneration handed over to the miner involved.

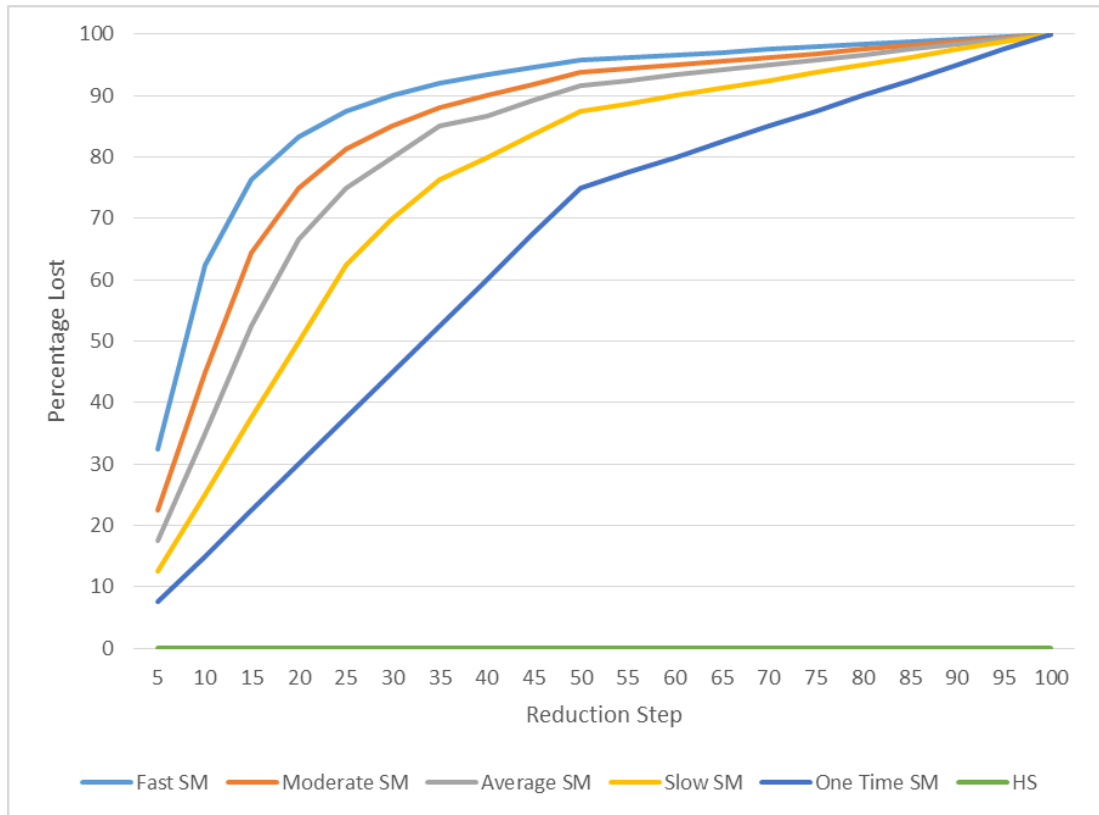
The same calculation is repeatedly done for every selfish mining frequency (five times in total), taking into account the increasing reduction step each time. Any random selfish miner can choose at any time to vary its speed from slow to rapid.

According to Table 3, the five frequencies for selfish mining are:

- Fast selfish mining resulting in 6 forks/hour.
- Moderate selfish mining resulting in 4 forks/hour.
- Average selfish mining resulting in 3 forks/hour.
- Slow selfish mining resulting in 2 forks/hour.
- One time selfish mining resulting in 1 fork/hour.

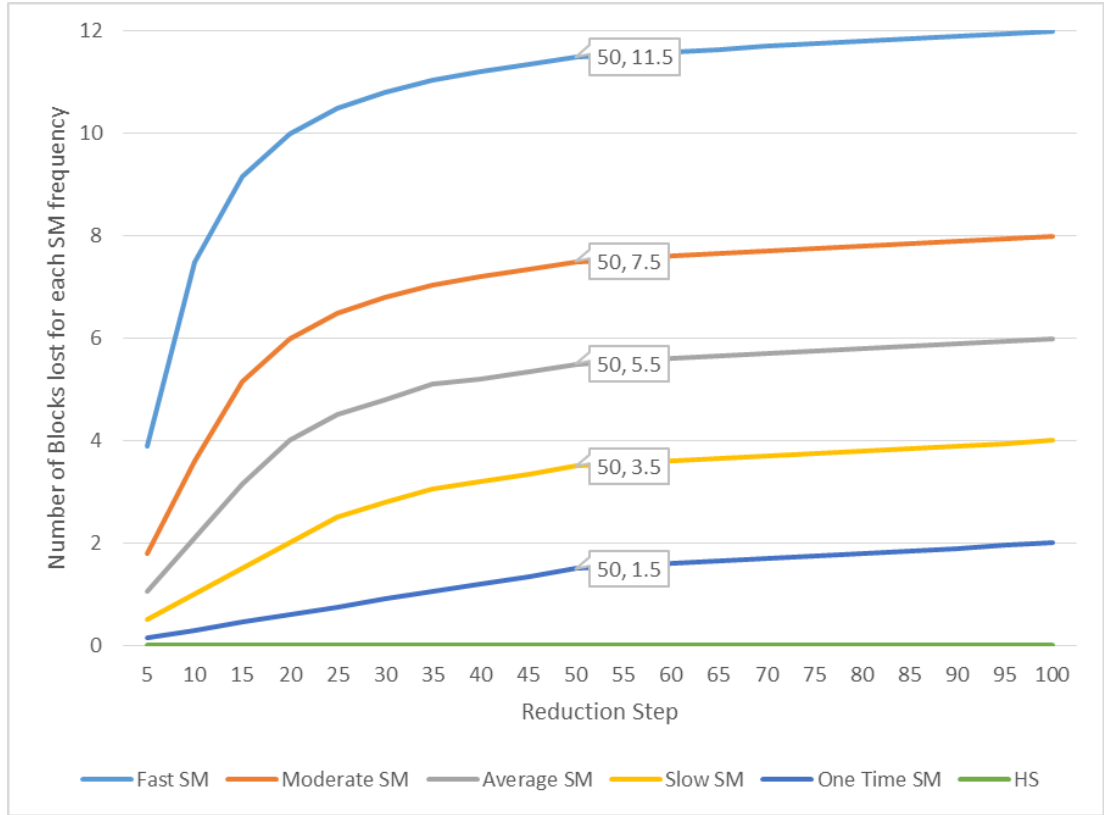
Each computation done resulted in finding the specific reduction step that guarantees countering selfish mining for each frequency. And for each case, according to this precise reduction step, the block percentage lost was calculated

along with the number of blocks lost. Two graphs were illustrated and illustrated to find the specific reduction step for S1.



**Figure 24: Graph of the Percentage Lost in function of Reduction Step (S1)**

Figure 24 depicts the graph of the Percentage Lost for all the selfish miners in function of the reduction step. According to the graph data, each colored curve represents a selfish mining case. All the curves are increasing with the increase in the reduction step, and ideally reach a point where selfish mining becomes unprofitable. This data showed that 50% reduction rate is required for all the cases. Subsequently, in order for S1 to work properly and evenly for every selfish mining frequency faced, the reduction rate is fixed to 50% as the general rule for this solution in this particular scenario.



**Figure 25: Graph of Number of private Blocks lost in function of the Reduction Step (S1)**

Figure 25 shows the graph of the number of blocks lost for each selfish mining frequency in function of the reduction step. This second graph also indicates that 50% is sufficient to stop selfish mining for all the cases varying from one time to fast. For example, for the Moderate SM, when the reduction is 50%, the number of blocks lost is 7.5 out of 8. According to math.com, when applying rounding to the nearest ones rule (*Numbers - Estimating and Rounding - In Depth*, n.d.), 7.5 is rounded to 8, hence 8 blocks reward is lost. Hence, if this miner successfully forked the system four times, his 8 blocks will be accepted and added to the chain but without gaining any revenue from this process.

Table 5 shows the needed reduction step for every selfish mining case, along with a summary of the outputs of the above two graphs.



Selfish miner activity	Reduction	Percentage Lost	Blocks reward lost
Fast	50%	95,83333333%	11.5 out of 12
Moderate	50%	93,75%	7.5 out of 8
Average	50%	91,66666667%	5.5 out of 6
Slow	50%	87,5%	3.5 out of 4
One Time	50%	75%	1.5 out of 2

**Table 5: Needed reduction step for selfish mining cases in S1**

As a result, in this first scenario S1, it is shown that 50% is sufficient to counter the Fast, Moderate, Average, Slow, and One Time selfish mining cases.

In conclusion, 50% is the step needed in this scenario S1 to be used in order to guarantee the defeat of any selfish miner no matter the frequency he is mining on.

In the second scenario S2, a specific percentage is deducted from every block reward for every fork imposed. A deduction percentage is constantly distributed and applied to each and every block every time a fork occurs with constantly using the same deduction percentage for the two blocks in every fork, and increasing this percentage for the next fork, and so on. Thus, the block rewards is the same for every fork, and is decreased by the constant specific step every time a fork is introduced. In other terms, for a step of 5%, after the first fork, the first and the second blocks reward will be 95% of 6.25 BTC. For the same miner, after the second fork, the two blocks reward will be 90% of 6.25 BTC accordingly. In addition, the reward will decrease every time a fork is presented for the same 5% step in this case as shown in Table 6, and other reduction steps in other cases.

Reduction step: 5%

Forks	1 <sup>st</sup> Fork		2 <sup>nd</sup> Fork		3 <sup>rd</sup> Fork		4 <sup>th</sup> Fork		5 <sup>th</sup> Fork		6 <sup>th</sup> Fork	
Blocks	B1	B2	B1	B2	B1	B2	B1	B2	B1	B2	B1	B2
Reward	95%	95%	90%	90%	85%	85%	80%	80%	75%	75%	70%	70%
BTC	5.9375	5.9375	5.625	5.625	5.3125	5.3125	5	5	4.6875	4.6875	4.375	4.375

**Table 6: Block rewards distribution according to 5% deduction in S2 (Fast SM)**

According to last row in Table 6, the sum of block rewards for the six forks is 61.875 BTC in an hour compared to the sum without penalty equal to 75 BTC. When comparing these two values, it is found that a 17.5 percent loss is applied in this case. Hence, when applying a 5% reduction on every block reward, a hasty selfish miner will lose 17.5% of the normal block reward. Thus, in the 60 minutes timeframe, this hasty miner lost 2.1 blocks approximately 2 of his 12 blocks added.

Selfish Mining frequency	Fast Selfish Mining	
Scenario	S1	S2
Reduction Step	5%	5%
Loss percentage	32.5%	17.5%
Lost Block reward (Blocks added to the chain but without rewards).	3.9 Blocks	2.1 Blocks

**Table 7: S1 vs. S2 (5% reduction for Fast SM)**

Table 7 clarifies the difference between the 2 scenarios for a fast selfish miner for a reduction step of 5%. In the 2 scenarios, the selfish miner is losing, however S1 is more punitive than S2. In both cases, after six forks in 60 minutes, private blocks

are added to the chain as per standard Blockchain rules, but without full reward. After an hour timeframe, in both S1 and S2, all the 12 blocks (fast selfish mining: 6 forks, 12 blocks in 60 minutes) are added to the chain, but with different rewards in the 5% reduction. For example, the total reward in S1 is 50.625BTC  $((12 - 3.9) * 6.25\text{BTC})$  compared to 61.875BTC  $((12 - 2.1) * 6.25\text{BTC})$  in S2 according to the data from Table 7. In other deduction steps, some blocks are added with zero reward.

Likewise, the Reduction step in S2 was steadily increased from 5% to 100% at a rate of 5% each time. This variation will give a clear observation in finding the right percentage window to stop every selfish mining activity depending on its speed and frequency. Therefore like in S1, the main objective in S2 is to find and apply the precise step that will guarantee stopping selfish mining.

After performing the necessary work and the repeated calculations for the five selfish mining frequencies. By continuously increasing the reduction step from 5 to 100%, it was found that for S2 the optimal reduction step needed is 75%.

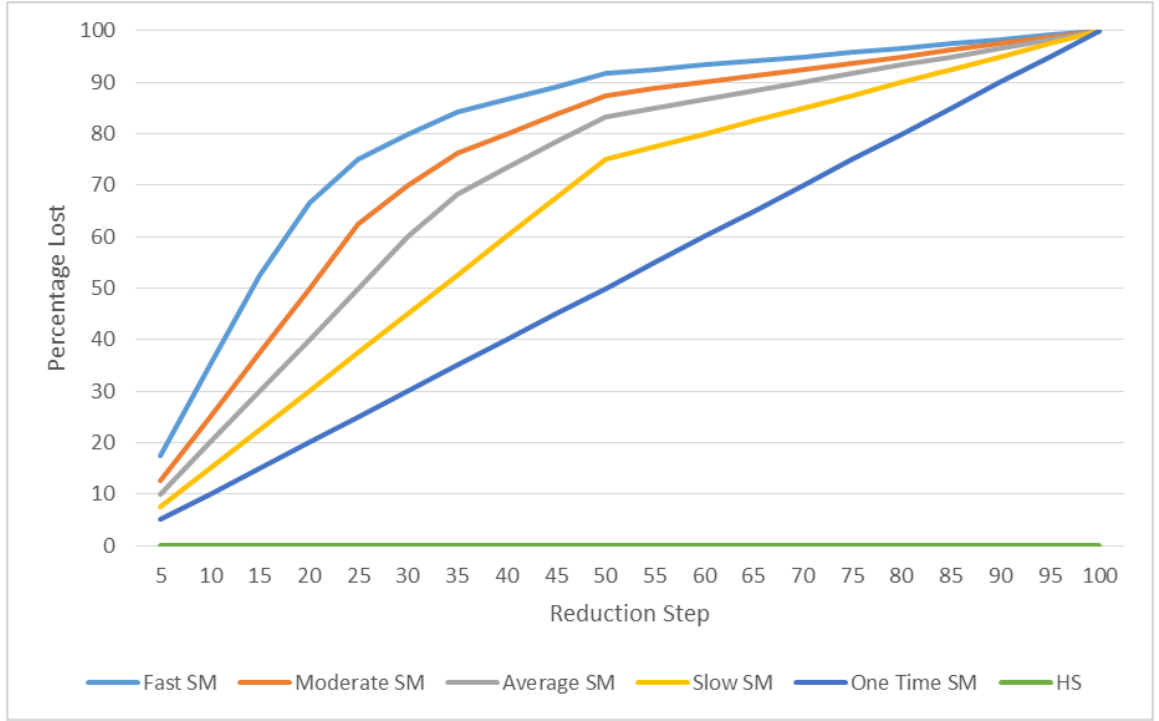


Figure 26: Graph of the Percentage Lost in function of Reduction Step (S2)

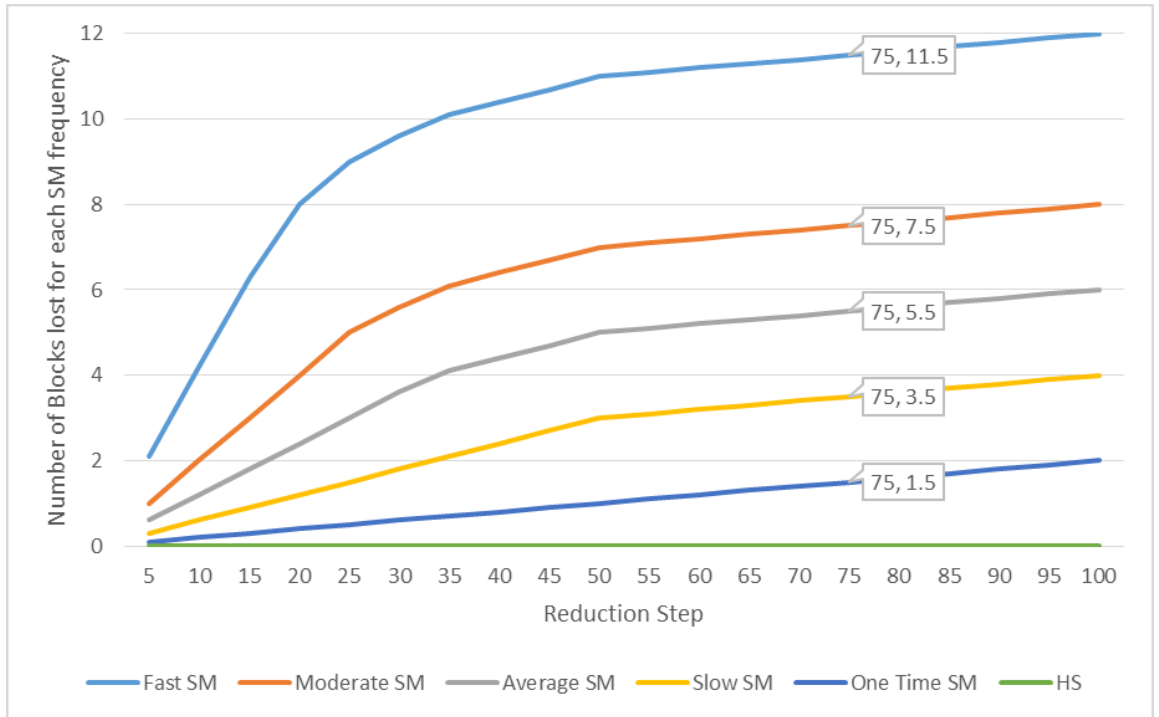


Figure 27: Graph of Number of private Blocks lost in function of the Reduction Step (S2)

Figure 26 depicts the graph of the Percentage Lost for all the selfish miners in function of the reduction step in S2. According to the graph data, each colored curve represents a selfish mining case. All the curves are increasing with the increase in the reduction step, and ideally reach a point where selfish mining becomes unprofitable. This data showed that in order for S2 to be guaranteed for stopping every selfish mining frequency faced, the reduction rate must be fixed to 75% as the general rule for this solution in this specific scenario.

Figure 27 shows the graph of the number of blocks lost for each selfish mining frequency in function of the reduction step. This second graph also specifies that 75% is the exact reduction step needed to stop selfish mining for all the cases varying from one time to fast. For example, for the Average SM, when the reduction is 75%, the number of blocks lost is 5.5 out of 6. According to math.com, when applying rounding to the nearest ones rule (*Numbers - Estimating and Rounding - In Depth*, n.d.), 5.5 is rounded to 6, hence 6 blocks reward is lost. Hence, if this miner successfully forked the system three times, his 6 blocks in the 3 forks will be validated and added to the chain but without gaining any profits from this process.

Table 8 shows the needed reduction step for every selfish mining case in S2, along with a summary of the outputs of the above 2 graphs represented in Figures 26 - 27.

Selfish miner activity	Reduction	Percentage Lost	Blocks reward lost
Fast	75%	95,83333333%	11.5 out of 12
Moderate	75%	93,75%	7.5 out of 8
Average	75%	91,66666667%	5.5 out of 6
Slow	75%	87,5%	3.5 out of 4
One Time	75%	75%	1.5 out of 2

**Table 8: Needed reduction step for selfish mining cases in S2**

Therefore, in this second scenario S2, it is shown that 75% is sufficient to counter the Fast, Moderate, Average, Slow, and One Time selfish mining cases.

In conclusion, 75% is the step needed in this scenario S2 to be used in order to guarantee the defeat of any selfish miner no matter the frequency he is mining on.

Both scenarios assure stopping selfish miners from getting rewards by finding the required constant reduction step needed to stop the selfish act. A simple comparison of the two scenarios is shown in the following table.

<b>Scenario S1</b>				<b>Scenario S2</b>			
5% Reduction Step				5% Reduction Step			
Discounted block reward percentage according to the reduction step							
Fork 1		Fork 2		Fork 1		Fork 2	
B1	B2	B1	B2	B1	B2	B1	B2
95%	90%	85%	80%	95%	95%	90%	90%
30% Reduction Step				30% Reduction Step			
Discounted block reward percentage according to the reduction step							
Fork 1		Fork 2		Fork 1		Fork 2	
B1	B2	B1	B2	B1	B2	B1	B2
70%	40%	10%	0%	70%	70%	40%	40%

**Table 9: S1 vs. S2 in a simple case**

Table 9 above, shows the different approach taken in the two scenarios when the calculation was done according to two reduction cases 5 and 30%. As stated before, S1 is harsher than S2 as the percentage is constantly decreasing in every block in every fork, whereas in S2 this percentage is the same for every two blocks in the same fork. In both scenario, this percentage is increased persistently for every fork imposed to the system.

According to the data presented in the above table, the two different approached represented in each scenario can be chosen to fight selfish mining efficiently. Thus, the next needed step is to tweak or modify any Blockchain systems algorithm (Bitcoin in this research) to accommodate this solution, consequently stopping selfish mining profits completely. If S1 is chosen the required reduction step needed in the modified algorithm is 50% whereas if S2 is chosen the required deduction needed is 75%.

## **Chapter 4: Conclusion**

This chapter sum up the core results, contributions, and possible future extensions to the work yet to come.

### **4.1 Summary of the Main Results**

A new penalty system was shaped with its proper calculations to solve the selfish mining problem. The calculations aim to find a specific percentage value used to limit and stop selfish block reward. This solution also complies with every selfish mining frequency ranging from one time up to six times in a sixty minutes timeframe. Two scenarios were taken into account when finding this proper reduction step. S1 yielded in a 50% value, whereas S2 75%. Whether relying on S1 or S2, selfish mining is no longer beneficial for fraudulent nodes. Private blocks resulting from imposed forks are added to the main chain of the Blockchain system as per standards. However, by applying and implementing this solution, block rewards are completely eliminated, consequently the losses will undeniably vary differently from selfish miner to another depending on the frequency used.

### **4.2 Main Contributions of the Thesis**

Using the penalty system proposed in this thesis grant several advantages. Blockchain users can incorporate this approach in their system to guarantee proper reward distribution for every participating miner in the system. New miners are incentivized to join these secured Blockchains, as they will be rewarded evenly and



according to their respective work. In addition, the decentralized nature of Blockchain is preserved, taking the fact that no selfish miner or selfish mining pool can control the system in this case. The selfish pool will no longer increase in size towards a majority to control the Blockchain. Knowing that the penalty is every time imposed when a selfish miner is present depending on his rate of mining. Any random selfish miner will without doubt lose the block reward, in addition to the loss of energy, money and power used to mine. This optimum penalty system can be used in any Blockchain network where mining process is heavily used.

### **4.3 Possible Extensions and Future Work**

The system cannot be used only to penalize selfish miners but it can be extended to reward honest miners. In every fork, the remaining block reward from every discount can be distributed to honest miners so that they get a share for their hard work. Furthermore, block size can be varied each time a mining occurs, in a manner that if a selfish miner withhold a block for a period of time, the block can be discarded when published. Taking into account that its size in this mining time doesn't comply with the agreed block size approved on in the first place for this mining phase.

## Bibliography

- Adarsh, S. (2017). Decentralized Computing Using Blockchain Technologies and Smart Contracts: Emerging Research and Opportunities. In *ProtoView* (Vol. 2017, Issue 10). Ringgold Inc. <https://search.proquest.com/other-sources/decentralized-computing-using-blockchain/docview/1878301958/se-2?accountid=28281>
- Azimy, H., & Ghorbani, A. (2019). Competitive Selfish Mining. *2019 17th International Conference on Privacy, Security and Trust (PST)*, 1–8. <https://doi.org/10.1109/PST47121.2019.8949043>
- Bansod, S., & Ragha, L. (2020). Blockchain Technology: Applications and Research Challenges. *2020 International Conference for Emerging Technology (INCET)*, 1–6. <https://doi.org/10.1109/INCET49848.2020.9154065>
- Batabyal, A. (2020, April 30). *Bitcoin Mining Software - 5 Best Bitcoin Mining Software in 2020*. <https://coinswitch.co/news/top-10-best-bitcoin-mining-softwares-2020-latest-bitcoin-mining-software-review>
- Battista, G. Di, Donato, V. Di, Patrignani, M., Pizzonia, M., Roselli, V., & Tamassia, R. (2015). Bitconeview: visualization of flows in the bitcoin transaction graph. *2015 IEEE Symposium on Visualization for Cyber Security (VizSec)*, 1–8. <https://doi.org/10.1109/VIZSEC.2015.7312773>
- Best Bitcoin Cloud Mining Contract Reviews and Comparisons*. (n.d.). Retrieved April 26, 2021, from <https://www.bitcoinmining.com/best-bitcoin-cloud-mining-contract-reviews/>
- Bitcoin Core :: About*. (n.d.). Retrieved April 26, 2021, from <https://bitcoincore.org/en/about/>
- BLOCKCHAIN | Definition of BLOCKCHAIN by Oxford Dictionary on Lexico.com also meaning of BLOCKCHAIN*. (n.d.). Retrieved May 1, 2021, from <https://www.lexico.com/definition/blockchain>

- Blockchain Charts*. (n.d.). Retrieved May 2, 2021, from <https://www.blockchain.com/charts/pools>
- Brennan, C., Lunn, W., & Suisse, C. (2016). Blockchain: The Trust Disrupter. *Equity Research, Europe/United Kingdom Equity Research Technology Research*, 1–339. <https://www.finextra.com/finextra-downloads/newsdocs/document-1063851711.pdf>
- Buterin, V., Reijnders, D., Leonardos, S., & Piliouras, G. (2019). *Incentives in Ethereum's Hybrid Casper Protocol*. <https://doi.org/10.1109/BLOC.2019.8751241>
- Chang, S., & Park, Y. (2019). Silent Timestamping for Blockchain Mining Pool Security. *2019 International Conference on Computing, Networking and Communications (ICNC)*, 1–5. <https://doi.org/10.1109/ICCNC.2019.8685563>
- Chatterjee, A., Shahaab, A., Gerdes, M. W., Martinez, S., & Khatiwada, P. (2021). Chapter 22 - Leveraging technology for healthcare and retaining access to personal health data to enhance personal health and well-being. In S. Bhattacharyya, P. Dutta, D. Samanta, A. Mukherjee, & I. Pan (Eds.), *Recent Trends in Computational Intelligence Enabled Research* (pp. 367–376). Academic Press. <https://doi.org/https://doi.org/10.1016/B978-0-12-822844-9.00044-X>
- Crypto mining glossary: blocks in blockchain, hash, reward - MineBest*. (2021, May 25). <https://minebest.com/blog/crypto-mining-blockchain-block-hash-reward>
- Derks, J., Gordijn, J., & Siegmans, A. (2018). From chaining blocks to breaking even: A study on the profitability of bitcoin mining from 2012 to 2016. *Electronic Markets*, 28(3), 321–338. <https://doi.org/10.1007/s12525-018-0308-3>
- Ethereum wallets | ethereum.org*. (n.d.). Retrieved April 26, 2021, from <https://ethereum.org/en/wallets/>
- Everything you need to know about Bitcoin mining*. (n.d.). Retrieved April 26, 2021, from <https://www.bitcoinmining.com/>
- Eyal, I., & Sirer, E. G. (2018). Majority is Not Enough: Bitcoin Mining is Vulnerable. *Commun. ACM*, 61(7), 95–102. <https://doi.org/10.1145/3212998>
- Five Challenges Blockchain Technology Must Overcome Before Mainstream Adoption |*

- Nasdaq*. (n.d.). Retrieved April 22, 2021, from <https://www.nasdaq.com/articles/five-challenges-blockchain-technology-must-overcome-before-mainstream-adoption-2018-01-03>
- Frankenfield, J. (2019, May 6). *51% Attack Definition*.  
<https://www.investopedia.com/terms/1/51-attack.asp>
- Frankenfield, J. (2020, January 29). *Private Key*.  
<https://www.investopedia.com/terms/p/private-key.asp>
- Frankenfield, J. (2021, April 21). *Proof of Stake (PoS) Definition*.  
<https://www.investopedia.com/terms/p/proof-stake-pos.asp>
- Frankenfield, J., & Rasure, E. (2021, April 1). *Selfish Mining Definition*.  
<https://www.investopedia.com/terms/s/selfish-mining.asp>
- Frankenfield, J., Rasure, E., & Reeves Marcus. (2021, September 22). *Block Header (Cryptocurrency) Definition*. <https://www.investopedia.com/terms/b/block-header-cryptocurrency.asp>
- Gupta, A. (2017, May 23). *Blockchain Technology- Types and Components*.  
<https://www.linkedin.com/pulse/blockchain-technology-types-components-dr-anita-gupta>
- Heilman, E. (2014). *One Weird Trick to Stop Selfish Miners: Fresh Bitcoins, A Solution for the Honest Miner (Poster Abstract) BT - Financial Cryptography and Data Security* (R. Böhme, M. Brenner, T. Moore, & M. Smith (Eds.); pp. 161–162). Springer Berlin Heidelberg.
- Hermann, V. (2018). *The SAGE Encyclopedia of the Internet*. SAGE Publications, Inc.  
<https://doi.org/10.4135/9781473960367> NV - 3
- Jakobsson, M., & Juels, A. (1999). *Proofs of Work and Bread Pudding Protocols(Extended Abstract) BT - Secure Information Networks: Communications and Multimedia Security IFIP TC6/TC11 Joint Working Conference on Communications and Multimedia Security (CMS'99) September 20–21, 1999, Leuven* (B. Preneel (Ed.); pp. 258–272). Springer US. [https://doi.org/10.1007/978-0-387-35568-9\\_18](https://doi.org/10.1007/978-0-387-35568-9_18)

- Joshi, A., Han, M., & Wang, Y. (2018). A survey on security and privacy issues of blockchain technology. *Mathematical Foundations of Computing, 1*, 121–147. <https://doi.org/10.3934/mfc.2018007>
- Kano, Y., & Nakajima, T. (2018). A novel approach to solve a mining work centralization problem in blockchain technologies. *International Journal of Pervasive Computing and Communications, 14*(1), 15–32. <https://doi.org/10.1108/IJPC-C-D-18-00005>
- Katrenko, A., & Sotnichek, M. (2020, October 8). *Blockchain Attack Vectors: Vulnerabilities of the Most Secure Technology | Apriorit*. <https://www.apriorit.com/dev-blog/578-blockchain-attack-vectors>
- Kim, Y., & Jo, J. Y. (2018). Dynamically adjusting the mining capacity in cryptocurrency with binary blockchain. *International Journal of Networked and Distributed Computing, 6*(1), 43–52. <https://doi.org/10.2991/ijndc.2018.6.1.5>
- Kranz, G. (2021, July). *What is metadata and how does it work?* <https://whatis.techtarget.com/definition/metadata>
- Krawisz, D. (n.d.). *The Proof-of-Work Concept | Satoshi Nakamoto Institute*. Retrieved April 26, 2021, from <https://nakamotoinstitute.org/mempool/the-proof-of-work-concept/>
- Kufeoglu, S., & Ozkuran, M. (2019). *Energy Consumption of Bitcoin Mining*.
- Küfeoğlu, S., & Özkuran, M. (2019). Bitcoin mining: A global review of energy and power demand. *Energy Research & Social Science, 58*, 101273. <https://doi.org/https://doi.org/10.1016/j.erss.2019.101273>
- Kwon, Y., Kim, D., Son, Y., Vasserman, E., & Kim, Y. (2017). *Be Selfish and Avoid Dilemmas: Fork After Withholding (FAW) Attacks on Bitcoin*. <https://doi.org/10.1145/3133956.3134019>
- Lansiti, M., & Lakhani, K. (2007). *The Truth About Blockchain*. <https://hbr.org/2017/01/the-truth-about-blockchain>
- Laurence, T. (2017). *Blockchain For Dummies* (1st ed.). For Dummies.
- Lewis Antony. (2015, September 9). *A Gentle Introduction to Blockchain Technology* –

- Bits on Blocks*. <https://bitsonblocks.net/2015/09/09/gentle-introduction-blockchain-technology/#incentives>
- Li, W., Andreina, S., Bohli, J.-M., & Karame, G. (2017). *Securing Proof-of-Stake Blockchain Protocols*. [https://doi.org/10.1007/978-3-319-67816-0\\_17](https://doi.org/10.1007/978-3-319-67816-0_17)
- Marquit, M. (2020, April 27). *Is Bitcoin Mining Profitable?*  
<https://www.thebalance.com/can-bitcoin-mining-make-a-profit-4157922#citation-8>
- Mason, B. (n.d.). *Bitcoin Mining for Dummies - Step-by-step guide to mine bitcoin*. Retrieved April 26, 2021, from <https://www.fxempire.com/education/article/bitcoin-mining-for-dummies-427762>
- May, A. (n.d.). *Ethereum Mining Pools: The Best Mining Pool for ETH Listed*. Retrieved April 26, 2021, from <https://miningpools.com/ethereum/>
- Nakamoto, S. (2009). *Bitcoin: A Peer-to-Peer Electronic Cash System*.  
<https://bitcoin.org/bitcoin.pdf>
- Nofer, M., Gomber, P., Hinz, O., & Schiereck, D. (2017). Blockchain. *Business & Information Systems Engineering*, 59(3), 183–187. <https://doi.org/10.1007/s12599-017-0467-3>
- Numbers - Estimating and Rounding - In Depth*. (n.d.). Retrieved May 7, 2021, from <http://www.math.com/school/subject1/lessons/S1U1L3DP.html>
- Popper, N. (2018, June 27). *What is the Blockchain? Explaining the Tech Behind Cryptocurrencies - The New York Times*.  
<https://www.nytimes.com/2018/06/27/business/dealbook/blockchains-guide-information.html>
- Porat, A., Pratap, A., Shah, P., & Adkar, V. (2017). *Blockchain Consensus : An analysis of Proof-of-Work and its applications*.
- Rosenbaum, K. (2019). *Grokking Bitcoin*. Manning.  
<https://books.google.com.lb/books?id=VTgzEAAAQBAJ>
- Rosic, A. (n.d.-a). *Blockchain Consensus: A Simple Explanation Anyone Can Understand*. Retrieved April 26, 2021, from <https://blockgeeks.com/guides/blockchain-consensus/>

- Rosic, A. (n.d.-b). *Proof of Work vs Proof of Stake: Basic Mining Guide - Blockgeeks*. Retrieved April 26, 2021, from <https://blockgeeks.com/guides/proof-of-work-vs-proof-of-stake/>
- Rosic, A. (n.d.-c). *What is Bitcoin? [The Most Comprehensive Step-by-Step Guide] Updated!* Retrieved April 22, 2021, from <https://blockgeeks.com/guides/what-is-bitcoin/>
- Rosic, A. (2019, October 17). *What Is Hashing? [Step-by-Step Guide-Under Hood Of Blockchain]*. <https://blockgeeks.com/guides/what-is-hashing/>
- Running A Full Node - Bitcoin*. (n.d.). Retrieved April 22, 2021, from <https://bitcoin.org/en/full-node#what-is-a-full-node>
- S. Shetty, S., Njilla, L., & Kamhoua, C. A. (2019). Introduction. In *Blockchain for Distributed Systems Security* (pp. 1–24). <https://doi.org/https://doi.org/10.1002/9781119519621.ch1>
- Saad, M., Spaulding, J., Njilla, L., Kamhoua, C., Shetty, S., Nyang, D., & Mohaisen, D. (2020). Exploring the Attack Surface of Blockchain: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, *PP*, 1. <https://doi.org/10.1109/COMST.2020.2975999>
- Sayeed, S., & Marco-Gisbert, H. (2019). Assessing Blockchain Consensus and Security Mechanisms against the 51% Attack. *Applied Sciences*, *9*, 1788. <https://doi.org/10.3390/app9091788>
- Siim, J. (2017). *Proof-of-Stake. Research Seminar in Cryptography*. <https://github.com/mit-dci/tangled-curl/blob/master/>
- Silva, J. (2021). *SANS Institute Information Security Reading Room An Overview of Cryptographic Hash Functions and Their Uses*.
- Smart contracts - simple to complex - BlockchainHub*. (n.d.). Retrieved May 2, 2021, from <http://blockchainhub.net/blog/infographics/smart-contracts-simple-complex/>
- Solat, S., & Potop-Butucaru, M. (2016a). *ZeroBlock: Preventing Selfish Mining in Bitcoin*. <https://hal.archives-ouvertes.fr/hal-01310088>

- Solat, S., & Potop-Butucaru, M. (2016b). ZeroBlock: Timestamp-Free Prevention of Block-Withholding Attack in Bitcoin. *ArXiv E-Prints*, arXiv:1605.02435.
- Szmigielski, A. (2016). *Bitcoin essentials : gain insights into Bitcoin, a cryptocurrency and a powerful technology, to optimize your Bitcoin mining techniques*. Birmingham, UK: Packt Publishing.
- Torpey, K. (2019, July 28). *Bitcoin Mining Centralization Is 'Quite Alarming', But A Solution Is In The Works*. <https://www.forbes.com/sites/ktorpey/2019/07/28/bitcoin-mining-centralization-is-quite-alarming-but-a-solution-is-in-the-works/?sh=3824c170530b>
- Tuwiner, J. (2021, April 21). *9 Best Bitcoin Mining Pools: Legit Sites (2021 Companies)*. <https://www.buybitcoinworldwide.com/mining/pools/>
- Varshney, A. (2017, April 3). *Here's all you need to know about Blockchain and its different types - The Financial Express*. <https://www.financialexpress.com/industry/technology/heres-all-you-need-to-know-about-Blockchain-and-its-different-types/612970/>
- Voshmgir, S., & Kalinov, V. (2017, February 9). *Smart Contracts Explained - Infographic - BlockchainHub*. <https://blockchainhub.net/blog/infographics/smart-contracts-explained/>
- Wang, J. (2014). A simple Byzantine Generals protocol. *Journal of Combinatorial Optimization*, 27(3), 541–544. <https://doi.org/10.1007/s10878-012-9534-3>
- What are smart contracts on blockchain? | IBM*. (n.d.). Retrieved April 22, 2021, from <https://www.ibm.com/topics/smart-contracts>
- What is a Smart Contract? Auto enforceable Code - Blockchain*. (n.d.). Retrieved April 22, 2021, from <https://blockchainhub.net/smart-contracts/>
- Why bitcoin uses so much energy | The Economist*. (n.d.). Retrieved April 22, 2021, from <https://www.economist.com/the-economist-explains/2018/07/09/why-bitcoin-uses-so-much-energy>
- Wild, J., Arnold, M., & Stafford, P. (2015, November 1). *Technology: Banks seek the key*



*to blockchain* / *Financial Times*. <https://www.ft.com/content/eb1f8256-7b4b-11e5-a1fe-567b37f80b64>

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. *2017 IEEE International Congress on Big Data (BigData Congress)*, 557–564.  
<https://doi.org/10.1109/BigDataCongress.2017.85>

