# FROM BLOCKCHAIN TO INTERNET-BASED VOTING

ELHAM AKBARI

Bachelor of Science in Computer Engineering

Tehran Azad University

January 2013

Submitted in partial fulfillment of requirement for the degree

MASTER OF SCIENCE

at the

CLEVELAND STATE UNIVERSITY

August 2018

We hereby approve this thesis for

ELHAM AKBARI

Candidate for the Master of Science in Software Engineering degree for the Department of Electrical Engineering and Computer Science

and the

CLEVELAND STATE UNIVERSITY'S

College of Graduate Studies by

---

Thesis Committee Chairperson, Wenbing Zhao, Ph.D.

---

Department & Date

---

Thesis Committee Member, Nigamanth Sridhar, Ph.D.

---

Department & Date

---

Thesis Committee Member, Chansu Yu, Ph.D.

---

Department & Date

Student's Date of Defense: August 10, 2018

## ACKNOWLEDGEMENT

I would like to express my deep appreciation to my thesis advisor and mentor, Professor Wenbing Zhao for his support, encouragement, and advice. I would also like to thank the committee members, Professor Chansu Yu and Professor Nigamanth Sridhar for their guidance and interest in my work. Additionally, I wish to thank my parents for their unconditional support and love. They have truly devoted their entire life to the comfort, happiness and success of their children. I would also like to thank my siblings, Shirin, Maryam, and Mohammad Reza, who have been unlimited sources of inspiration and motivation for me. Last but definitely not least, I wish to thank my dear husband, Dr. Emad Mehdizadeh. I owe him for every bit of success I have had from the moment he came into my life. He is the most important motivation for me to move forward and to be a better person each day that goes by.

FROM BLOCKCHAIN TO INTERNET-BASED VOTING

ELHAM AKBARI

## ABSTRACT

Blockchain has been one of the hottest topics among the state-of-the-art technologies. As the enabling technology for Bitcoin, the pioneering cryptocurrency, blockchain is an append-only distributed ledger that is virtually impossible to attack. Hence, blockchain holds great promises as the fundamental technology to enable Internet-based electronic voting. However, Internet-based voting has additional requirements than what monetary transactions such as Bitcoin have to offer. In this thesis, we discuss the key differences of a blockchain-based voting system with digital currencies. In this context we also highlight the requirements, review existing proposed solutions, and outline possible improvements. Specifically, we propose several schemes on how to tackle various issues such as authentication, privacy, transparency, scalability, safety, as well as several other practical aspects of the platform. Most importantly, a blockchain-based voting system needs to ensure that the prospect of tampering with the election result is to a large extent eliminated. At the same time, the voting platform should have proper performance characteristics, *i.e.* sufficient throughput, for a voting of large magnitude such as a presidential election. Being heavily linked together, security and performance should be investigated in a unified framework to capture the interaction effects between the two. To address this concern, for the first time, we will study the performance and security implications of the blockchain voting system in a quantitative manner, using a blockchain simulator developed by researchers at Swiss Federal Institute of Technology, ETH Zurich. In our analysis, we will specifically investigate the stale

block rate and relative mining share of the dishonest network, as the central security measures, as a function of important network parameters that determine the throughput of the network, *i.e.* block size and block interval. Ultimately, we focus on selfish mining and eclipse attacks as the most critical threats to the integrity of the blockchain voting in order to find the optimal network parameters.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

# CHAPTER I

## INTRODUCTION

The voting systems that have been utilized globally to permit people cast their ballots are either paper-based (conventional) or electronic-based. Not only using paper ballots and counting them is prone to errors but also is a time-consuming process. However, the risks of the electronic voting (e-voting) is so substantial that has prevented many governments from implementing it. If any interference with an e-voting system happens, the possible costs are beyond fatal. All in all, the existing voting systems, whether they are electronic or conventional, involve insufficient levels of transparency. In effect, in either case it becomes extremely difficult or unbearable for voters to ensure that their election votes are counted carefully and accurately by the election administrators. As an example, the Virginia's voting machine displayed different security-related problems causing complete discontinuation of it by the Virginia Information Technologies Agency. Moreover, Direct Recording Electronic (DRE) Voting Systems such as those implemented in Brazilian election do not publicly provide any records of the election statistics and results, apart from the counts of the votes. This implies that only government representatives are capable of recounting the votes if required. This by no means can provide the voters with any confidence in the election

results. Although, in some cases the information from a Direct Recording Electronic System can be backtracked to explore association of votes with voters, it would generate serious concerns about the votes confidentiality which is not acceptable in a democratic election.

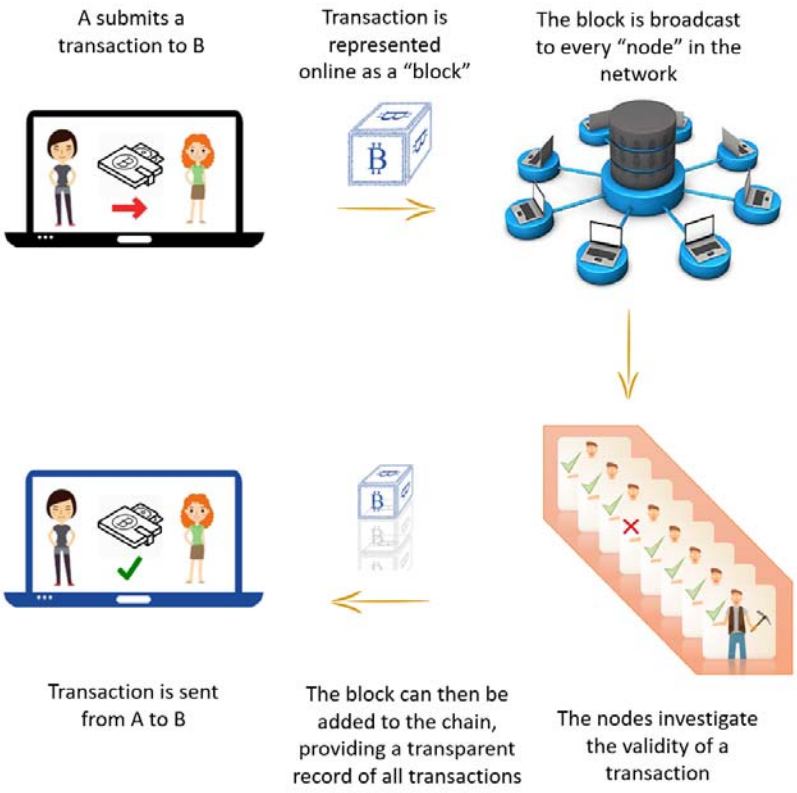



Figure 1-1 The schematic of a blockchain network.

The transparency, assurance, and confidentiality issues of an e-voting system can now be potentially addressed using a new and emerging technology that is called blockchain. With its exclusive features and characteristics, the Blockchain technology possesses very promising potentials. Figure 1, for instance, illustrates the schematic of how a typical blockchain system operates to achieve a safe and secure transaction without the need for any financial institution. When person "A" owes person "B" money for a

service, B requests a transaction from the wallet of A. The transaction is then represented in the network as a “Block”. The block is then broadcast to every “node” in the network. These nodes are normally called “miners”. Once the block is approved by the majority of the miners, the transaction can be finally approved and completed. This provides a transparent and reliable platform for sending and receiving transactions, completely different than conventional methods relying on financial institutions. The same concept can be also extended to other transaction-like notions such as votes.

In 2008, blockchain was first introduced by an unknown person named Satoshi Nakamoto (a pseudonym), who intended to develop a peer-to-peer payment system allowing money transactions through the web without relying on trust or the need for a financial bank. In 2016, an Australian computer scientist and businessman named “Craig Steven Wright” publicly claimed to be the main part of the team that was responsible for inventing the blockchain. A blockchain system is in principle, an open source system, yet resistant to any data modifications.

The unique and secure architecture of a blockchain-based network infers that interfering is fundamentally impossible when properly executed, because a blockchain network is stringently transparent and consensus-based as well as distributed. After the 2016 US presidential elections in which the electronic voting systems were regarded to be interfered with by foreign hackers, the implementation of voting with the aid of blockchain networks has gained increasingly higher attention. For instance, President Obama’s decision to deprive 35 Russian diplomats from the US due to the concerns of Russia’s interference with the 2016 election, deduces the vulnerability of the voting systems to external tampering. A voting system that is equipped with blockchain

technology provides a substantial level of transparency by sustaining an exposed registry of votes, while defending the privacy of the voters. In blockchain technology, consensus from almost all the nodes is mandatory in order for a transaction to gets approved. This makes the voting machine a substantially safer platform. In other words, in order to tinker the election result, an attacker must be able to get access to a significant portion of all the nodes within the network. When a sufficiently large number of nodes are implemented, potential attacks become exponentially more challenging to conduct, if not fully impossible. In the following we aim at reviewing the previous works that have been performed on blockchain-based e-voting systems.

In 2015, Daniel for the first time proposed the use of blockchain technology as a key to secure online voting [1]. In 2016, however, the notion of using blockchain-based e-voting systems gained more traction during the US presidential election. In particular, this followed after September 2016, when FBI Director testified before the House Judiciary Committee that the FBI was investigating Russian hackers attempting to disrupt the 2016 election and that federal investigators had detected hacker-related activities in state voter registration databases, confirming there were multiple attempts to hack voter database registrations [2]. In late 2016, Ryan Osgood [3] discussed the engineering of the blockchain and its benefits as well as the progress and challenges of widespread adoption. In early 2017, Kartik Hegadekatti [4] outlined the procedure underlining voting on the blockchain and reviewed the advantages of such system. He also analyzed the impacts of voting through the Blockchain. In January 2017, Ivo Kubjas [5] described how to make internet voting protocols more secure through the use of blockchain. In May 2017, Ahmed Ben Ayed [6] proposed an electronic voting system design by leveraging

the open source nature of the blockchain technology for making elections secure, reliable, and anonymous, and to help increase the number of voters as well as the trust of people in their governments. In 2017, Moura and Gomes [7] explored the possibility of using blockchain technology to help solve transparency and confidence issues associated with nation-wide elections. They focused on the societal problems and their respective analysis. Finally they analyzed how the adoption of Blockchain into a digital government repertoire can contribute to common e-voting issues and also promote elections transparency, increase auditability, and strengthen democracy. In June 2017, Bartolucci *el al.* [8] discussed possible uses of the blockchain technology for implementing a secure and fair voting system. They introduced a secret share-based voting system on the blockchain, the so-called SHARVOT protocol. The solution they provided used Shamir's Secret Sharing to enable on-chain, i.e. within the transactions script, votes submission and winning candidate determination. Their proposed protocol also used a shuffling technique, *Circle Shuffle*, to de-link voters from their submissions. In 2017, Kaan Koç *et al.* [9] implemented and tested a sample e-voting application as a smart contract for the Ethereum network using the Ethereum wallets and the Solidity language. They considered Android platform to allow voting for people who do not have Ethereum wallets. In their proposed approach, once an election is completed the Ethereum blockchain will maintain the records of ballots and votes. Users can then submit their votes via an Android device or directly from their Ethereum wallets, and these transaction requests are handled with the consensus of all Ethereum node. This can potentially create a transparent environment for e-voting. In 2018, Casado-Vara and Corchado [10] proposed a new model of Blockchain, designated to prevent and minimize the flaws of

the voting system. In the proposed model, the Distributed Ledger (Blockchain) is used to broadcast digital, smart contract voting to a poll station. Then the poll station sends a smart contract to individual voters and registers the vote on a sidechain. At the end of the voting process, the entire sidechain would be committed to the main voting Blockchain. Smart contracts would be used as a platform to vote, with the reason being to prevent malicious activities. At the end of the voting, the poll station applies a multi-signature to the most recent vote of each voter, and smart contract transfers it to the candidate or ballot measure. In 2018, Wang *et al*. [11] proposed an electronic voting scheme based on blockchain based on the homomorphic ElGamal encryption and ring signature. The key properties of such system is reported to be decentralization, self-management, non-interactive, and free-receipt. Moreover, the one-time ring signature ensures the anonymity of the vote trading in the distributed ledger. Furthermore, the public verifiable billboards is claimed to guarantee the voting fairness, while the miner nodes that provide ciphertext ballot counting service make large-scale voting feasible. In 2018, Akbari and Zhao *et al*. [12] analyzed additional requirements of Internet-based voting compared to monetary transactions. They also review existing proposed solutions, and outlined possible improvements. They also proposed to use live biometrics of the voter to perform secure and highly reliable remote authentication. Additionally, a scheme was suggested to protect the secrecy of the ballots while eliminating the influence of the-already-cast votes on the ongoing election process. Finally, they proposed to impose a hierarchy to the voting infrastructure that aligns naturally with traditional voting. This enables parallel processing of multiple blockchains, to overcome the intrinsic scalability limitation of the blockchain technology. Despite numerous publications and studies performed on the

applicability of blockchain for electronic-based elections, the feasibility of such platform especially at large-scale is yet to be fully explored. Furthermore, the security requirements of the blockchain-based voting systems have not received much attention in the literature. In this thesis, for the first time, we propose using Proof of Work (PoW) as the consensus mechanism for internet-based blockchain voting systems to investigate the security guarantees of variant or forked PoW platforms. In particular, we will implement a quantitative framework to analyze the security and performance implications of a PoW-based blockchain-voting system. Based on such framework, we will take into account real-world constraints such as network propagation, block generation intervals, different block sizes, information propagation mechanism, etc. This framework will allow us to quantitatively compare the tradeoffs between the performance and security provisions of various PoW-based deployments. In this regard, we will exploit a blockchain simulator developed at ETH Zurich [13] to evaluate different blockchain-voting instances from a performance standpoint.

# CHAPTER II

# INTRODUCTION TO BLOCKCHAIN TECHNOLOGY

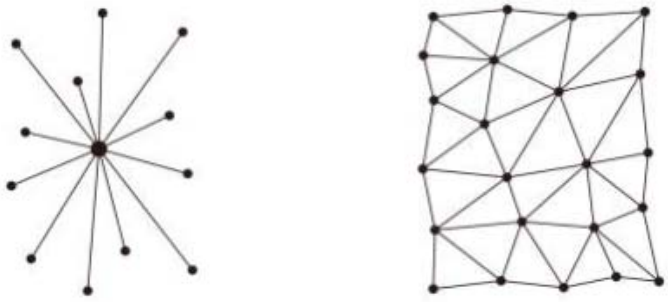
In this chapter we give an overview of how the blockchain works by using Bitcoin as an example. We also describe the most important concepts behind the operation of blockchain systems.

## 2.1 What is Blockchain?

A blockchain, in brief, is a certain type of data structure which controls how data is constructed and stored. Databases, images, CSV and text files are other conventional types of data structures. In other words, a Blockchain or distributed ledger technology (DLT) is a protocol that enables information to be traded between different involved parties within a network without the need for intermediaries. This enables specific events to be agreed upon by parties without the need for a third party. In this context, no single entity owns or controls the data. Moreover, the database should be append-only. i.e. information can only be written to them and old information cannot be modified or deleted unless based on a complete agreement from the entire network of users. In the event that someone decides to rewrite a portion of the ledger, it will take them an enormous amount of time to catch up and overtake the remainder of the network which is

legit for the most part. For this very reason blockchains are known to be extremely difficult to alter.

In a blockchain, the network users anonymously interact with each other via encrypted identities. In effect, each virtual asset is added to an undisputable transaction chain and distributed to all network nodes [14]. This is a unique characteristic of peer-to-peer (P2P) networks. P2P computing or networking is a distributed application architecture that divides responsibilities and functions between peers (Fig. 2.1). In P2P networks, peers are equally privileged to take part in the application. This is unlike server-based networks in which data is completely kept on servers, and one can access the data upon logging in. Majority of the internet is server-based. For instance, websites are held on the server, and clients are those who access it. In server-based networks the clients entrust the data to be definitive. Although, this traditional model is very efficient in computing, it is a centralized network and therefore vulnerable to attacks or failures.



(a) (b)

Figure 2-1. (a) Schematic of a centralized (server-based) network. (b) Schematic of a decentralized (peer-to-peer) network

A P2P network is similar to a gossip network in which every peer has access to almost all the data, and updates are shared between the peers. Since the data in a P2P network is duplicated many times, it is typically considered less efficient than

server-based networks. Even though each duplication or modification of the data introduces too much gossip around the network, each peer is more independent, and can continue operating partially even if they lose connection with the network. Furthermore, since no central server controls the data in P2P networks, they are in general more robust, i.e. attacking peer-to-peer networks is significantly more challenging.

### 2.1.1 History and Applications

Bitcoin is the first widely used application of the blockchain technology. Today, Bitcoin is known worldwide as the first decentralized digital currency, and cryptography-based payment system. It can be simply described as a vast database of transactions that relies on operation of tens of thousands of computing machines around the globe. Bitcoin enables transactions that can take place directly between users. Verification of such transactions is performed through a large network of nodes called miners. These transactions are recorded in a public distributed ledger we know as blockchain. Individual duplicates of such are stored in the network using the universal bitcoin protocol. A series of files known as “blocks” are used to keep the record of every single transaction that has ever occurred in the system. This is in fact called the Bitcoin ledger that has been constantly growing ever since it was introduced in January 2009. The schematic of a typical blockchain system, and Bitcoin for that matter, is shown in Fig. 2.2.

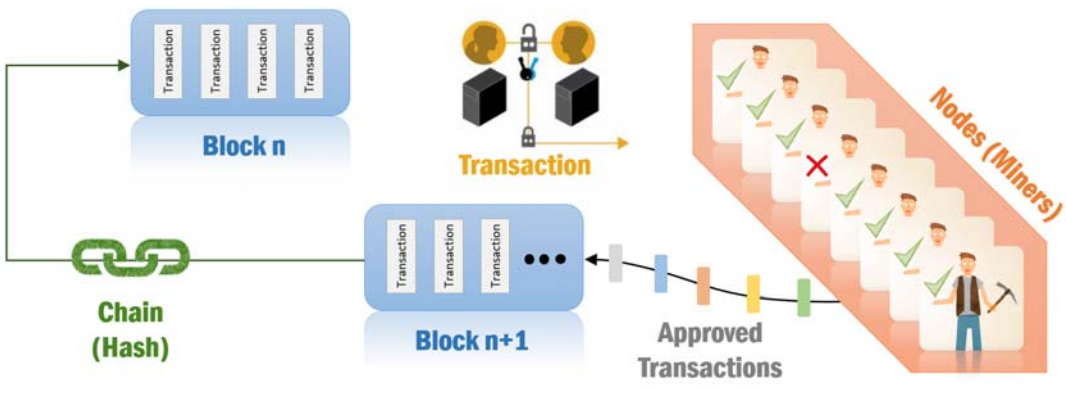



Figure 2-2. The structure of a blockchain system.

The Bitcoin ledger is an open source database in which one does not need permission from anyone in order to write stuff into it. As a result, no logging in or signing up is needed for the users in order to have access to it. The act of appending is done through running an open-source software through which one computer connects to the other computers within the network via web. The software allows one to send or receive transactions, or add data to the ledger by solving a computationally difficult math problem that is yet easily verifiable. This is widely known as “mining”. It is important to note that, the math problems are made challenging using functions called “hash”. In the bitcoin protocol, the hash functions are part of the cryptography algorithm that is used to write new transactions into the system through the mining process. Mining is an vital and essential part of blockchain that guarantees fairness while keeping the network stable and secure. Miners are issued a certain number of bitcoins in exchange for their service. This creates an incentivized platform to attract more people to mine. The higher the number of miners, a larger and more secure network can be created. By reviewing a Bitcoin file, one may easily find out which account has how many Bitcoins and which account is receiving Bitcoin from whom. This level of transparency is what enables the

Bitcoin transactions verifiable by anyone anywhere in the world. As a result, in case someone tries to append a fraudulent transaction the miners would easily know and do not approve it.

To date, thousands of different public and private blockchains are running through the network, many of which are yet to gain substantial traction. They can be divided into four main groups listed in Fig. 2.3. Based on these categories, a blockchain-based e-voting system would fall under “Record-keeping” platforms.

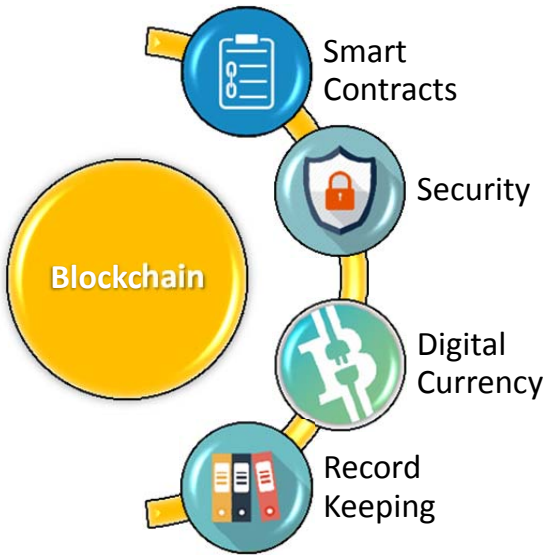



Figure 2-3. Applications of the blockchain technology can be divided into four main categories.

### 2.1.2 Public vs Private Blockchains

Depending on the blockchain technology needed for a certain purpose, we may allow anyone to write data into the ledger or only grant permission to a certain group of trustees, *i.e.* vetted contributors, to do so. A blockchain can be public from two different aspects, giving permission for writing data and for reading data. When it comes to public blockchains, typically the former is intended. On the contrary, in a private blockchain all

the participants are know and trusted. One of the drawbacks of a public blockchain is that it may be vulnerable to potential systematic attacks. When a sufficiently large group of hackers aim at intruding into a network they can outnumber the legitimate miners and compromise the consensus mechanism (described in the following section) in a certain direction. In the case of a digital currency such as Bitcoin, this may not introduce a great deal of concern, but it definitely will for certain record keeping projects such as that of a national election. In case of the latter, a foreign government might have both the power and interest into intervening in the election result in favor of a particular party, as was explained in Chapter 1. This problem can be addressed by a private blockchain in which, a set of trusted entities can have the permission to write, but read-access is granted to everyone. One question arising here is that what is the difference between a private blockchain and conventional third party platforms? Blockchains can eliminate the need for data transfer from organizations to the third party and vice versa. As a substitute, data is transferred between known organizations and a consensus can be made within a small interval. This infers that all parties can operate from a single and known state of events. Encryption is used to maintain privacy of information while digital signatures ensure authenticity and integrity of date. In other words, blockchains can address the problem of demanding trusted third parties [15].

### 2.2 Consensus

"Consensus" is the problem of getting members of a ledger to agree upon some entity. In centralized networks, there exists a control unit that can agree on the correct entities and send them to the entire network. In a distributed ledger, however, nodes in

the network needs to cooperatively come up with an agreement without benefiting a centralized unit. Being further complicated, some nodes within the network may attempt to compromise the integrity of the system by supporting a consensus that favors themselves rather than backing the actual truth. In other words, the consensus mechanism in a blockchain helps line up all the nodes in the system to develop an identical view of any event. Consensus-based blockchains are empowered by the ability to eliminate third parties from the ledger, while still having participants who agree on true and legitimate events. The question arising here is how an agreement can be reached in a general distributed ledger, or how the data to be written on the network can be selected? Moreover, it is imperative to develop a mechanism that can help find a resolution when different nodes claim contradictory things, and no mediation can be sought.

The answer actually lies in the concept of protocols that operate based on pre-settled guidelines for standardizing the consensus mechanism. In a P2P network even if the entire group of peers are trustworthy, the problem of agreement or consensus can still arise. In such scenarios, the network should be able to determine the state of the data, even if peers provide updates at different speeds and/or have somewhat different states. A typical issue known as “fork” occurs in distributed ledgers when several blocks are concurrently added via different miners. This can happen because blocks take time to be shared across the network. This is not unique to monetary blockchains and can also happen to blockchain voting. Hence, it is important to decide which one should count as the legit block. In such cases, a consensus rule called “longest chain” can help distinguish the legitimate block from fraudulent blocks. When a miner elects to acknowledge the legitimacy of an existing path, the path will be extended and it is inferred as a vote

towards consensus on that particular path. This implies that the longer a path is, the more computation has been performed to build it. As an example, let's consider the schematic of Fig. 2.4 and assume that all the miners are synchronized on block 51. Now, if three miners generate three different "Block 52" at almost the same time, the longest chain rule will be used to decide which block is valid. It should be noted that the three different blocks are slightly different because they contain different payment addresses and different set of transactions. In practice, one may assume that the first block 52 they see, in this case block 52 (a), is valid and start making the next block based on that going after a possible block 53 (a). However, in seconds later a block 52 (b) can appear as well, followed by block 53 (b), instead. In such case, based on the longest chain rule, block 52 (a) should be ignored and extension (b) should be regarded as valid.

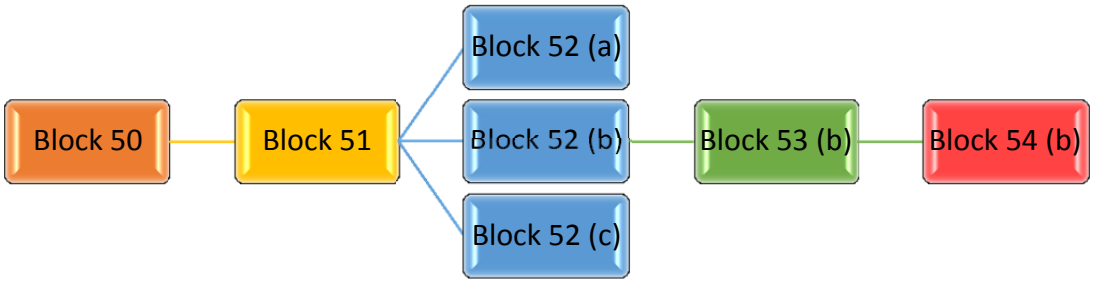



Figure 2-4. The schematic of a blockchain tree; the longest path represents the accepted chain.

### 2.3 Principle of Operation

In this section, Bitcoin is used as an example to describe how blockchain operates. With bitcoin, literally anyone can download the "Bitcoin Core" software and start authenticating transactions as well as generating blocks. The computer that runs the Bitcoin software as a full node can connect to the Bitcoin network, download and store

the ledger, monitor and validate transactions and blocks, create and mine the blocks. With Bitcoin, no sign up or log in is required for joining the network. This is completely unlike a centralized network such as SWIFT in which one must first get granted the permission to download the software and start monitoring the transactions. A potential issue with public blockchain systems is that they can be potentially attacked by anyone. As a result, a certain mechanism is needed is such systems to make the entire network trustworthy even though some of the users do not act that way. Hypothetically, a potential dishonest actor can (a) decline valid transactions that are sent to other players in the network, (b) try generating blocks that are to his/her own interest, (c) conduct double-spending activities, (d) try to generate "longer chains" of select blocks to transform a pre-accepted block into an orphan block. However, they cannot (a) generate Bitcoins out of nowhere, thieve Bitcoins from someone else's account, and (c) make payments on someone else's behalf. Despite all that, the impact a fraudulent node might have is often times very restricted. As long as the majority of the network are trustworthy, they will discard any deceitful transactions originating from dishonest nodes. They will also hear the valid transactions from honest actors, regardless. If a dishonest node has enough block generation power, they can only delay valid transactions by declining to include them into their blocks. Nevertheless, the valid transactions will be still known by the honest miners as "unconfirmed transaction" and therefore will be included in their blocks. In the worst case scenario, if the dishonest node can generate a longer chain of blocks compared to the rest of the nodes to instance the "longest chain rule" and outnumber the honest chains, a transaction can be undone.

In reality however, double-spending can be made extremely difficult to execute in blockchain networks. In Bitcoin, for instance, adding blocks is only possible at the expense of solving extremely difficult computational problems. This would be very expensive for the hacker to conduct because solving such problems requires a large extent of processing power that is only feasible upon buying and maintaining a very large number of computers.

This computational process can be defined as a guessing game in which the miners need to speculate a number that can results in a “hash” when processed alongside the rest of the block data contents. Hash can be described as some sort of password or fingerprint that is associated with the level of difficulty of mining or the total network processing power and needs to be smaller than a certain number. The more computers join the block processing task the more difficult it becomes to solve. This occurs in a self-regulating fashion. Such guessing game is known as “Proof of work” or PoW. Upon publishing a block with a hash smaller than the target value, one proves that they have done sufficient guessing work to satisfy the network at a certain point in time [16].

### 2.4 Proof of Work

A proof of work (PoW) is a piece of data that is expensive and time-consuming to generate but at the same time it is easy for others to verify whether it satisfies predefined requirements. Producing a PoW is normally a random event that has low probability. As a result, a great deal of trial and error will be needed to produce a valid PoW. Bitcoin uses the Hashcash PoW system. In order for a block to be admitted by the network, the miners need to first perform a PoW that encompasses the entire data of the block.

The difficulty of this job can be adjusted in order to restrict the rate at which new blocks can be generated by the network. Because of the small likelihood of successful generation, it is rather impossible to foresee which miner gets to first produce the next block. In order for a block to be valid, the puzzle requires the new hash to be smaller than a predefined number. This infers that for generating each block some work has been performed. Each block contains the hash of the upcoming block, hence each block possesses a chain of blocks which contain a large extent of work. As a result, modifying a block necessitates regeneration of all descendants and reperforming the work they comprehend. This is the main mechanism that protects the blockchain from being tampered. One of the most widely used proof-of-work schemes is based on SHA-2 which will be explained in the following.

The blockchain is to a large extent reproduced at each contributing node. In this context, the system is comprised of two different node types: regular nodes and miners. Regular nodes are those who directly involve in the transactions and can place at either end of a bargain, while miners are those who will receive incentives in return for evaluation of the submitted transactions.

Each user of the blockchain generates a public and private key in which each pair is 256 bit long. Private keys are used by the regular nodes to sign their transactions. A bitcoin user is identified by the 160-bit long hash of their public key encoded using Base58Check. When a set of transactions are grouped together, they are located into a block that can be added to the ledger. In order for a miner to add a new block to the ledger, they have to solve a computationally difficult and therefore expensive problem. The SHA2 secure hash function can for instance be used to hash several pieces of

information, including the transactions, the hash of the previous block, and a nonce (i.e., a random number that is only used once), as shown in Figure 2.5.

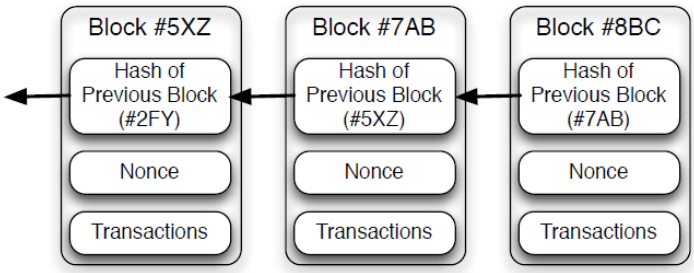



Figure 2-5. The structure of blockchain blocks.

# CHAPTER III

# FROM BLOCKCHAIN TO INTERNET-BASED VOTING

In this chapter, we outline the requirements for a blockchain-based voting system and highlight the key differences with existing blockchain platforms, while proposing new ideas on how such differences can be confronted. Furthermore, we identify the gaps between the proposed solutions and the architecture needed to enable Internet-based voting based on the blockchain technology. More specifically, we introduce a mechanism to authenticate votes based on live biometric characteristics. Additionally, we discuss how the scalability issues can be tackled by enabling parallel blockchains at the precinct-level as well as a hierarchical structure for the vote counting.

## 3.1 Related Work

In principal, blockchains are conceived to be extremely difficult to compromise. There is therefore no exception to a potential blockchain-based voting system. Even though one could try altering some of the ballots in a new block, it could be easily conquered as long as the majority of the miners in the network are honest. This way the new blocks can be consistently generated in a truthful fashion while the faulty blocks will find their ways out the window. To enhance the likelihood of broadcasting every single block to the majority of the miners in Bitcoin blockchain, it is typical for a miner to wait

and not confirm the blocks until six new blocks are fully generated [17]. This implies that every miner should normally have five unconfirmed blocks in their database. That is exactly why there is one hour latency in confirmation of newly generated blocks. Based on this scheme, however, the new blocks would be always vulnerable to attacks until they are confirmed. Once the confirmation goes through, it would be virtually impossible to attack a block simply because a hacker would have to now solve a much more difficult math problem, i.e. to find a nonce with an altered ballot that would lead to exactly the same hash as before. That is why blockchains are referred to as unforgeable public ledgers.

What follows is the big picture of what we propose for our blockchain based e-voting system, which is well in line with that of the existing digital currencies such as Bitcoin:

a. When new ballots are submitted they are broadcasted to all nodes across the network.
b. Upon receiving several new ballots, each node tries collecting them into a block.
c. Each node then works on finding a difficult-to-solve proof-of-work protocol for its block.
d. When a proof-of-work protocol is found, the block is broadcasted to all the nodes.
e. Each node then accepts the block only if all the ballots within it are valid and not previously used elsewhere.
f. Nodes display their approval of the block by trying to create the subsequent or following block in the chain, while using the hash of the accepted block as the preceding hash.

g. It is a common practice for the nodes to consider the longest chain to be the correct one and therefore upcoming blocks will be added in a way to extend it.

Unlike the digital currencies, in which every user, regardless of their geographical location, can take part in transactional activities, voting requires that only legal citizens of a particular nation can vote. This creates a huge difference in the infrastructure of the ledger. Hence, a blockchain in its typical form is far from sufficient from being used as the Internet-based voting. This requirement for voting necessitates the involvement of the government in the process. The government will ensure to put in place extra and exclusive measures to prevent potential frauds. In this context, a separate system must be devised to verify the citizenship and residency status of all the voters before they can be registered in the system. How to set up such a system is beyond the scope of this work. Nevertheless, any developed democratic country is expected to have already had a robust system for voters registration. Even with a robust voters registration system, the internet-based voting would still have to overcome the challenge of authenticating voters remotely. This in principle has nothing to do with the blockchain nature of the platform. Several studies have been already conducted by the blockchain industries who have investigated the use of the blockchain technology for internet-based voting [18], [19], [20]. They all concluded that the security risks at the current moment are greater than the benefits of it citing primarily the concerns on the difficulty of remote authentication of voters. While this may be a valid statement necessitating further studies on the topic, it is imperative to study the security and performance aspects of the blockchain e-voting in a quantitative manner as well. For the first time, herein we will take on this very important aspect of the technology.

Several research groups across the globe have reported works on electronic voting systems based on blockchain [21], [22], [23], [24]. Such researches mostly focus on the conceptual aspects of it and the need for protecting the privacy of the voters, i.e., how the votes can be kept anonymous in the records in the blockchain blocks. In [22], for instance, a trusted third party is proposed to be delegated by the voter registration agency to authenticate each voter based on the hash of the secret message that is exclusively issued for that voter by the agency. The delegation of duties between the voters registration organization and the trusted third party for verification of the entities is reported to ensure some level of privacy. It is also proposed to devote one distributed ledger for each candidate or party. Based on such approach, the winning candidate or party should be determined based upon the ledger with the longest chain. Other than the fact that this may damage the vote secrecy, we will see later in this chapter why this is not a viable option. To deal with the scalability constraint of blockchains, researchers in [24] discuss an algorithm that is proposed based on boardroom voting. A Diffie-Hellman-based algorithm is also utilized to protect the privacy of the voters. Being limited to the boardroom election, in such system every eligible voter must vote in order for the algorithm to work. This is not clearly applicable for general public elections. A real deployment of blockchain voting for general public elections, however, has been recently conducted by a start-up company called Votem [25]. One of the case studies of Votem is related to 2016 general election in the State of Montana in which Votem's Electronic Absentee System (EAS) and the Electronic Ballot Request System (EBRS) was implemented to facilitate absentee ballot delivery serving military and overseas voters in compliance with the Federal Voting Assistance Program and electors with

disabilities in compliance with Americans with Disabilities Act. Votem enables end-to-end verification of the election happens in a rolling and ongoing fashion during the election process. It benefits the exclusive Proof of Vote® Protocol for its end-to-end voter verifiable digital voting system. Despite, Votem has not dealt with the scalability issue of the blockchain voting system at large.

### 3.2 Remote Voter Authentication

We believe that the current technology is mature enough to ensure highly secure and reliable voter authentication. Hence, the need for remote voter authentication cannot be used as a reasonable excuse for not implementing Internet-based voting. While photo IDs are not required for (traditional) voting in the US, the concerns for security risks of remote voter authentication are automatically blown out of proportion [18], [19], [20]. Here, we propose recording the biometric information of every eligible voter at the time of registration. This may include but is not limited to fingerprint, IRIS, and facial characteristics. Such biometric data must be updated periodically prior to each election to ensure each eligible voter is still alive and thereby eliminating the prospect of someone impersonating a dead person. When remotely authenticating the voters, a voter should be also asked to submit the recorded biometric information to the system (e.g. via a smart phone app or full desktop computer application with a webcam). Designated centers may also be considered for people who have access to neither of the above-mentioned options. The camera in the machine can even monitor eye blinking, and possibly monitor body temperatures to ensure an actual live person is authenticating the voting system. Upon implementation of such measures, it would be impossible for a hacker to impersonate

another eligible voter. The nationwide biometrics database would also eliminate the possibility for anyone to vote multiple times while impersonating as different individuals.

### 3.3 Vote Secrecy

Vote secrecy implies that the no one can find out whom a voter has voted for. So long as it is desired to ensure that only eligible voters can vote, absolute vote secrecy cannot be achieved. This is because at some point in the voting process it is inevitable to rely on a centralized voter registration organization who is in charge of registering and authenticating voters. Such organization should be naturally trusted by the general public however and therefore it has the means to trace back the origins of each vote. Similar to [22], here it is assumed that a pseudo-identifier can be created for each voter at the time of authentication in order for them to cast their ballots. More specifically, the voter would use a key generator to produce a pair of public and private keys. The public key would be utilized as the voter ID, while the private key is used to cast the ballot. The voter ID is temporarily linked to the voter portfolio in the ongoing election. This voter ID should be different for different elections to maximize the vote secrecy. The above-mentioned technique would be fully compatible with the requirement of blockchain systems, in which the user's public key is used as the user identifier.

### 3.4 Voting Transparency and Consequences

One of the key and unique features of the blockchain is that it allows anyone to validate the transactions submitted throughout the system. While this feature can help build up the voting transparency, it can at the same time negatively influence the natural flow of an election. In effect, the outcome of the blockchain mining is in real-time

disclosed to the entire network, while the election is still ongoing. As a result, if certain groups of people find out that their candidates have fallen significantly short in votes, they might relinquish and choose not to take part in the election at all. Although this is not a fatal gating factor, in order to resolve this issue, it is proposed here to keep the vote counts secret until the conclusion of the election. This is however contradictory to the notion of using separate ledgers for each candidate in which the length of the chain would automatically reveal the winning party. To address this concern, a single blockchain can be used for an entire election. Even though this approach can cause scalability concerns, it would be described at a later stage how it can be addressed.

Similar to Bitcoin, in which blocks are comprised of several transactions, every block would contain a set of ballots in blockchain-based e-voting. In elections in which several candidates can be voted for by each voter, each ballot can very well contain multiple choices. This could even occur at various levels (e.g. national, state, or local). In such scenario, the ballots should be encrypted using symmetric keys of the voters. Such encryption can be performed in a fashion similar to that of the Pretty Good Privacy (PGP) technique, as shown in Fig. 3.1. In effect, the voter could encrypt his/her key using the public key provided by the election organization. PGP that was introduced in 1991 [26], is an encryption program providing cryptographic privacy and authentication for data communication. PGP has been widely used for signing, encrypting, and decrypting texts, e-mails, files, directories, and whole disk partitions as well as for increasing the security of e-mail communications. This way the general public can only get to find out the number of votes that have been cast, while the winning party is fully disguised until the election is entirely concluded. In such manner, the votes are also further protected by

encryption and only the election organization can decrypt the ballots. Once the election is ended, the election organization would make the encryption keys public so that anyone could verify the ballots. Simple programs can then be employed to automatically count the votes.

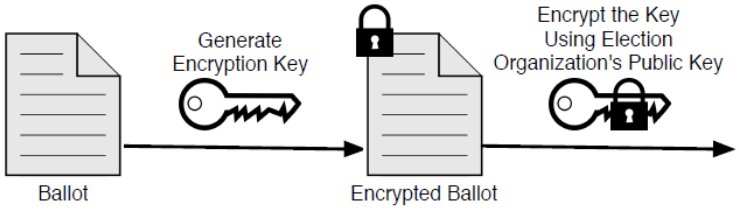



Figure 3-1. Illustration of how ballots can be protected by an encryption scheme similar to that of PGP.

### 3.5 Overcoming the Blockchain Scalability Limitation

The blockchain in its current form of implementation has serious scalability issues in which there are very limited number of new blocks that can be added to the chain in a certain time period. For instance, Bitcoin [27] blockchain only allows for about seven transactions per second to go through at the peak throughput [28]. If we assume that each block contains information about 10 votes, slightly over 250 thousand votes per hour can be added to the blockchain. According to this rate, a national election with 60% participation rate in the US would require about 330 hours, or two weeks for casting all the votes. This is clearly not feasible when a large scale election is to be conducted using the blockchain technology. It is suggested here to address the scalability issue by creating a hierarchical structure in the voting infrastructure. This is well in-line with the traditional form of elections in the United States in which election units can be as large as a county, or as small as a precinct. Such units can be managed by an individual blockchain as opposed to using a single blockchain for the entire nation. This scheme, illustrated in Fig.

3.2, would naturally enable parallel processing to a large extent which is mandatory for a nationwide election.

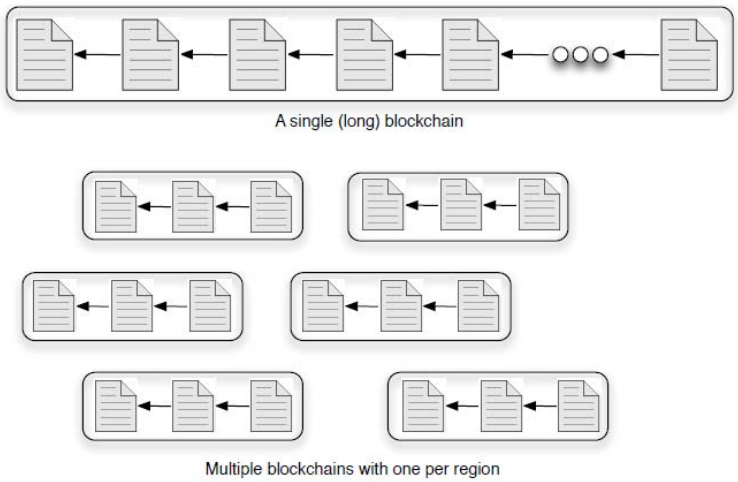



Figure 3-2. By using multiple region-based blockchains the scalability of the e-voting system can be significantly improved.

### 3.6 Other Practical Considerations

Thus far, several practical issues of the Internet-based voting were considered. Here, we discuss a number of other potential challenges that need to be addressed when implementing a blockchain-based voting system in a real-world scenario. First, despite the general wealth in the US, there are still significant number of people who may not have personal computers and/or do not have access to the Internet. Therefore, as mentioned earlier traditional voting centers that are equipped with Internet-connected computers would be still needed to allow any voter to go to such centers for casting their ballots. Moreover, to ensure the security and robustness of the blockchain-based voting infrastructure, a great number of miners would be required. In the case of Bitcoin or other blockchain-based digital currencies, in order to encourage more miners to participate in the mining process, certain incentives are considered to those who successfully solve the

puzzle. Similar incentives would be also necessary to attract miners for the voting infrastructure. After all, it should be kept in mind that voting in any country is an expensive activity. It is conceivable to anticipate tax-payer funds to be allocated for elections. Such funding can be also used to constantly improve the blockchain-related infrastructure and the methodologies that are to be implemented in the upcoming elections. Another viable approach for funding the incentivization process is to convince all the participating parties of the election to contribute to the expenditures. The amount of spending in this process would be actually insignificant as compared to the amount of money that is typically spent by the candidates for campaigning and commercials in each election.

Other potential concern when using internet-based election system is the prospect of letting foreigners to take part in the process of ballot verification. Such imperfection can readily open the door for malicious attacks to take place leading to tampering with the election result. In order to address this issue, one potential solution is to use private blockchains rather than public blockchains. In private blockchains the participants can be easily limited and controlled using traditional methods. A private blockchain can be planned such that only certain entities can append data into the blockchain, without providing external entities with the read or write access. To enable private blockchains, it is typical to use private networks with firewalls and IP whitelisting. Since IPs may be easily faked even by the hackers in other parts of the globe, other practical measure should be also put in place to further strengthen the security of the process. One good example is to consider a registration mechanism for those who are willing to play a role as miners. This could be very similar to that of the voters registration mechanism with the

added measure that the miners should undergo the intense background check. Obviously, only the citizens of that particular nation should be given the ability to register as miners. Being a very sensitive process, however, such technique has to be carefully devised in order not to compromise for the transparency of the voting ledger.

### 3.7 Important Parameters of the Network

In this section we review and introduce some of the more important parameters and performance characteristics of the blockchain that are necessary to consider when quantitively evaluating the performance and security aspects of a blockchain platform. Majority of these parameters will be used in the next chapter when simulating various scenarios of the blockchain voting system.

- **Stale block rate** is the rate at which stale blocks are generated. 'Stale blocks' refer to those blocks that are not included in the longest chain due to, for instance, contradiction or concurrency. Stale blocks are unfavorable to the security and performance of the blockchain as they initiate unwanted chain forks in the system. Chain forks negatively impact the growth of the central chain of the ledger and can cause bandwidth complications in the network. But above all, the stale blocks increase the ability of the dishonest nodes in performing fraudulent activities such as selfish mining, explained in the next chapter.
- **Block interval** is one of the most important parameters of the blockchain systems and is determined by the delay at which data is amended onto the ledger. A smaller block interval suggests a faster ballot approval and a higher likelihood of generating "stale blocks". It should be noted that adjusting the block interval implies changing the difficulty level of the to be solved PoW problem. In other

words, the difficulty of the PoW problem is conversely correlated to the rate at which stale blocks can be generated. This in turn infers that adjusting the difficulty of the problem can directly impacts the capability of the dishonest nodes in attacking the network through tampering with the longest chain of the ledger.

- **Block size** determines the number of ballots that can be collected within each block. Accordingly, the maximum block size regulates the throughput of the blockchain voting system. Larger the size of the block is, the slower the propagation speed and higher the slate block rate will be. Therefore, if one is to improve the throughput of the system, reducing the security of the system will be inevitable.
- **Information propagation mechanism** is the mechanism by which the blocks are broadcast to various network nodes. The broadcast scheme that is determined by the block request management system directly influences the scalability and robustness of the ledger. Most widely used propagation scheme is advertisement-based management system. In such method, as soon as node I receives data from another node, it will advertise the hash of that block to other connections in the network. In case one of the nodes has not already received that particular data, it will request for the content of it.
- **Mining power** is the ratio of the power of the dishonest portion of the network to that of the entire network.

### 3.8 Security Issues of Blockchain Voting System

Three typical ways of attacking a blockchain network are (*i*) double-spending, (*ii*) selfish mining, and (*iii*) eclipse attack. In the following we describe each of the above-mentioned mechanisms.

***Double-spending*** is constituted the most common type of committing scams in blockchains. For digital currencies, double-spending is referred to the case where a certain number of coins are spent in more than one transaction. In case of a blockchain voting system, this may be translated to a ballot that is being used for voting more than once. But as we will see in the following, due to the fundamental nature of blockchain systems, double-spending cannot be considered as a major and rational way of tampering with the election results. Let's first discuss the three different ways that can be used to conduct a double-spending activity in the case of Bitcoin and other digital currencies:

- When two contradictory transactions are submitted to the network in quick succession, a *race attack* is occurred. Obviously, only one of them that involves in the longest chain will eventually go through.
- When one transaction is pre-mined into a new block but it is not released until the very same coins are used for another new transaction. This method that is called *Finney attack* results in invalidation of the first transaction, if successful.
- When more than 51% of the overall computing power in the network is devoted for undoing a transaction and instead putting through a preferred transaction, the activity is called *51% attack*.

It is important to note that in each of the above fraudulent activities, the person who originally submits a transaction is the beneficiary and therefore the actual fraudster.

Having no money involved in the transactions (ballot), a potential dishonest voter would not gain anything by trying to undo his/her initial ballot. Moreover, such activities correspond to the cases in which certain assets, *e.g.* digital coins, are to be spent for contradictory outcomes which is conceptually different than the wills of a potential dishonest voter in blockchain voting. Therefore, double-spending seems to be fundamentally infeasible to accomplish and is of a lower concern in blockchain-based e-voting.

***Selfish mining*** occurs when a team of dishonest miners collude to augment their mining reward revenue. In such scenario miners can potentially earn more reward by concealing the newly produced blocks from the core chain and creating a distinct fork. In order to better understand selfish mining, let's take a deeper look into Bitcoin blockchains. Bitcoin mining operates based on a group of miners who unravel cryptographically complicated problems and get incentivized, *i.e.* receive digital coins, in exchange. Such income depends on a number of factors such as the level of difficulty of the cryptographical problem, mining cost, internet speed, and connection quality. All in all, Bitcoin is arranged in a way to incentivize miners proportional to their mining output. With such strategy in place, even if big groups of miners attempt colluding, they cannot receive more coins combined than what they individually and collectively generated in the public ledger. Nonetheless, if dishonest nodes conceal the new blocks and make them available only in their private network they can rise their share of the network's overall reward. Selfish mining is such an important issue that can even jeopardize the decentralization nature of blockchains causing the centralization of the blockchain

operations. Unlike double-spending, selfish mining should be carefully treated when it comes to blockchain-enabled voting.

Selfish-mining attacks could have profound effects on the integrity of blockchain system. When successful, dishonest adversaries can easily turn into more profitable nodes than the honest nodes. Profits from selfish mining can rise if more computational power is utilized by the adversaries. This can make the attacks exponentially more effective, until to a point where over 50% of the power in the network is held in favor of the attackers. This can ultimately force regular nodes out of the network. In such case, the dishonest portion of the network would be not only able to gather all the block rewards, but also to block any ballot from being counted fairly [29].

***Eclipse attack*** is another deceitful activity in blockchains in which a dishonest node takes control of the victim's inward and outward connections, hence separating the victim from the rest of the nodes in the network. The invader can then block the victim's visibility of the network, and obligate them to spend their computing power on viewing an outdated version of the blockchain network, or even worse divert the power to the advantage of his/her own iniquitous activities. Other than interrupting and damaging the integrity of the blockchain network, eclipse attacks could be the onset of and escalate other potential attacks such as selfish mining.

# CHAPTER IV

# SIMULATIONS: PERFORMANCE VS SECURITY

In this chapter, we leverage a blockchain simulator developed by researchers at ETH to study the main gating factors of blockchain-based voting systems that is the scalability and throughput of the ledgers. Due to the trade-off between the throughput and security of blockchain systems, it is also imperative to study the security concerns of blockchain voting systems. Stale blocks are known to pose serious danger to the integrity and security of distributed ledgers. Accordingly, we use block interval and block size as the inputs of the blockchain simulator to determine the stale block rate in the system. Interestingly, block interval and block size are also key parameters when it comes to calculation of the system throughput. Finally, the output of the simulator is fed into a security model that is developed based on the Markov Decision Processes (MDP) in order to investigate the security implications of the blockchain [13].

## 4.1 Blockchain Simulator

Since real-world implementation with thousands of nodes is extremely challenging in many cases, a powerful simulator can be of vital importance for realistically studying the blockchain performance as a function of network parameters. Recent studies on blockchain systems suggest that there is a trade-off between the

performance and security of PoW-based blockchains. Therefore, it is extremely helpful to have a unified framework that can capture such trade-offs as a function of different network parameters. The novel quantitative framework introduced in [13] can analyze the security and performance implications of various parameters of PoW blockchains. Taking advantage of such framework, not only the security properties of well-known PoW platforms such as Bitcoin, Ethereum, and Litecoin can be examined, but also important blockchain parameters can be adjusted to branch out into other similar platform such as blockchain-based voting. This framework is comprised of two main elements, a blockchain instance and a blockchain security model. A blockchain instance is a proof of work blockchain that is represented by a certain set of network parameters, such as block generation time, block size, network delays, etc. To convincingly analyze any blockchain instance, a simulator can be used to replicate the blockchain network and consensus layers through the implementation of advertisement-based data transmission. One of the key outputs of a blockchain platform is the rate at which the stale blocks are generated. This output can then be used as the input into another model that can give insight to the security behavior of the system. Herein, a security model that is built upon Markov Decision Processes (MDP) [29] is used to simulate the effect of stale block rates and eclipse attacks on selfish mining activities allowing to understand the optimal strategies that may be employed by dishonest adversaries. The security of the system can be regarded as the genesis of any blockchain platform directly determining any go or no-go decision. On the other hand, improving the performance of blockchains is greatly valuable as there are ongoing discussions in the community as to how the scalability

issues of blockchains can be mitigated through maximizing the block size. Please refer to Appendix I for the instruction on the installation of the simulator.

### 4.1.1 Simulator Structure

The list of some of the important blockchain parameters that can be capture by the ETH simulator are summarized in Table 4.1.

Table 4-1. The important network parameters that can be captured in the blockchain simulator used in this work.

| ***Network Parameter*** | **Description** |
|---|---|
| *Block Size* | Fixed block size (bytes) |
| *Number of Blocks* | The number of generated blocks |
| *Number of Nodes* | The total number of nodes in the network |
| *Block Interval* | The average block generation interval (minutes) |
| *PoW Power Distribution* | Mining Power Distribution of the Miners |
| *Number of Connections* | Per Node Within the Network |
| *Block Request Management System* | Protocol Used to Manage Block Requests |
| *Stale Block Rate* | The Ratio of the Stale Blocks to the Entire Blocks |

In the simulator, assigning a new block to a miner is determined based upon the block interval. In compliance with the existing proof of work blockchains, the simulator assumes that typical miners start mining as soon as they receive a block. Also it is assumed that potential forks get automatically resolved based on the longest chain rule. After resolving any fork, the blocks that are not attributed to the main chain of the network are constituted as stale blocks. Since the difficulty variations between various

blocks are not considered in the simulator, the lengths of the chains are just defined and calculated based on the number of blocks forming each chain.

In the communication protocol between the nodes, the channels are formed directly in between every pair of nodes. This way any intermediary machine such as routes are sidestepped completely. Each channel, in this context, would have two main features, bandwidth and latency. In order to credibly take the effect of network latencies in the simulator, the developers have employed the global IP latency statistics from Verizon. Furthermore, in order to accurately capture the bandwidth of the network, testmy.net has been utilized to obtain the bandwidth distributions. However, since the main intent of the simulator is to investigate the effect of the network parameters such as the block size and the block interval, it does not capture transaction propagation which has no correlation with the above-mentioned parameters.

The simulator differentiates miners from the typical nodes of the network. The geographical node distribution of the network is extracted from blockchain.info. Based on this distribution, around 52% of the nodes are located in Europe while North America contributes about 39% of the nodes. The remaining nodes in percentile order are traced to be in Asia Pacific, Australia, Japan, and South America. On the other hand, the distribution of the miners is quite different. Asia pacific possesses a share of about 71%, while North America and Europe only contribute about 24% and 5%, respectively [13]. It should be noted that such distribution which is correlated to the Bitcoin blockchain is merely utilized to replicate a real-world implementation. As discussed in the previous chapter, ideally only miners who live in certain geographical areas and are citizens of a particular country should be authenticated to play a role in voting blockchain system,

either as miners or regular nodes (voters). Despite, the real case geographical distribution employed in the simulator is not going to have a substantial impact on the performance of the network.

### 4.2 Simulations

In this section we first describe the conditions under which the simulation are conducted. Then, the simulations results are presented and discussed. It is noteworthy to mention that the simulator used here has been previously validated through comparison with empirical results of several blockchain platforms such as Bitcoin and Litecoin. The key idea here is to alter the network parameters of interest, such as block interval and block size, to measure other important parameters such as the stale block rates and look into the security aspects of the ledger as a function of those parameters. Finally, a set of parameter values that can result in the best-case performance and security in the network will be proposed in the upcoming sections.

#### 4.2.1 Simulation Conditions

The simulations are conducted based on the assumption that the dishonest nodes cannot potentially utilize more than 30% of the overall mining power [30]. We perform the simulations for block sizes ranging from 1 KB up to 25 MB, given different block intervals ranged between 1 seconds and 30 minutes. The number of generated blocks and the total number of nodes in the network are both considered to be 100 in all the simulations. Based on such combination, each simulation run takes about 70 seconds to complete which is orders of magnitude faster than an actual implementation if we were to really execute the scenario in real-world. As a reference, it should be mentioned that

simulation runs with the number of blocks/nodes of 500/100, and 100/500 would take about 300 seconds and 900 seconds to finish, respectively. A 500/500 combination for number of blocks/nodes also requires longer than an hour to finish for every run. On the other hand, the simulations suggest that they all result in comparable outcome, stale block rate, to that of 100/100 combination (see Fig. 4.1). Since we are to perform over 50 runs, the latter is used to save considerable amount of time.

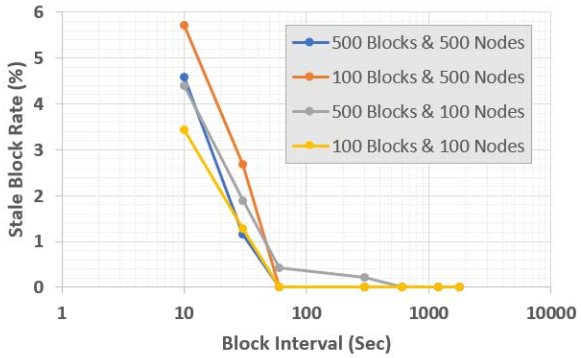



Figure 4-1. Stale block rate as a function of block interval for a block size of 10 KB and for different combinations for the number of blocks and nodes.

### 4.2.2 Simulation Results

Simulations were performed for six different block sizes 1 KB, 10 KB, 100 KB, 1 MB, 10 MB, and 25 MB, while measuring the stale block rate for each of the following block intervals 1, 10, 30, 60, 300, 600, 1200, and 1800 seconds. This accounts for 48 individual scenarios whose results are shown in Fig. 4.2. By conducting these simulations we aim at studying the impact of block interval and block size on the stale block rate. Stale block rate is very pivotal to determining the security of the blockchain network, as it will be discussed in the upcoming sections. For now, it is enough to know that a smaller stale block rate is more desired and typically represents a more secure system.

As a general trend, the simulation results of Fig. 4.2 suggest that the stale block rate is inversely correlated with the block interval. Moreover, at a constant block interval larger block size results in a higher value of stale block rate. Accordingly, it is safe to conclude that increasing the block interval and block size will improve and degrade the security, respectively.

### 4.3 Security Model

In order to investigate the security of voting blockchain system, the security model [13] developed based on Markov Decision Processes (MDP) is used here. In this model, the output of the blockchain simulator, *i.e.* the stale block rate, obtained in the previous section can be used as the input of the MDP model to find the potential relative revenue of the dishonest network for each scenario. The revenue of the dishonest network measured in percentage relative to the total revenue in the network is a key measure of the security of blockchain systems. As discussed in chapter 3, two of the most typical malicious activities that can be performed on a blockchain e-voting are selfish mining and eclipse attacks that are the point of focus in this section.

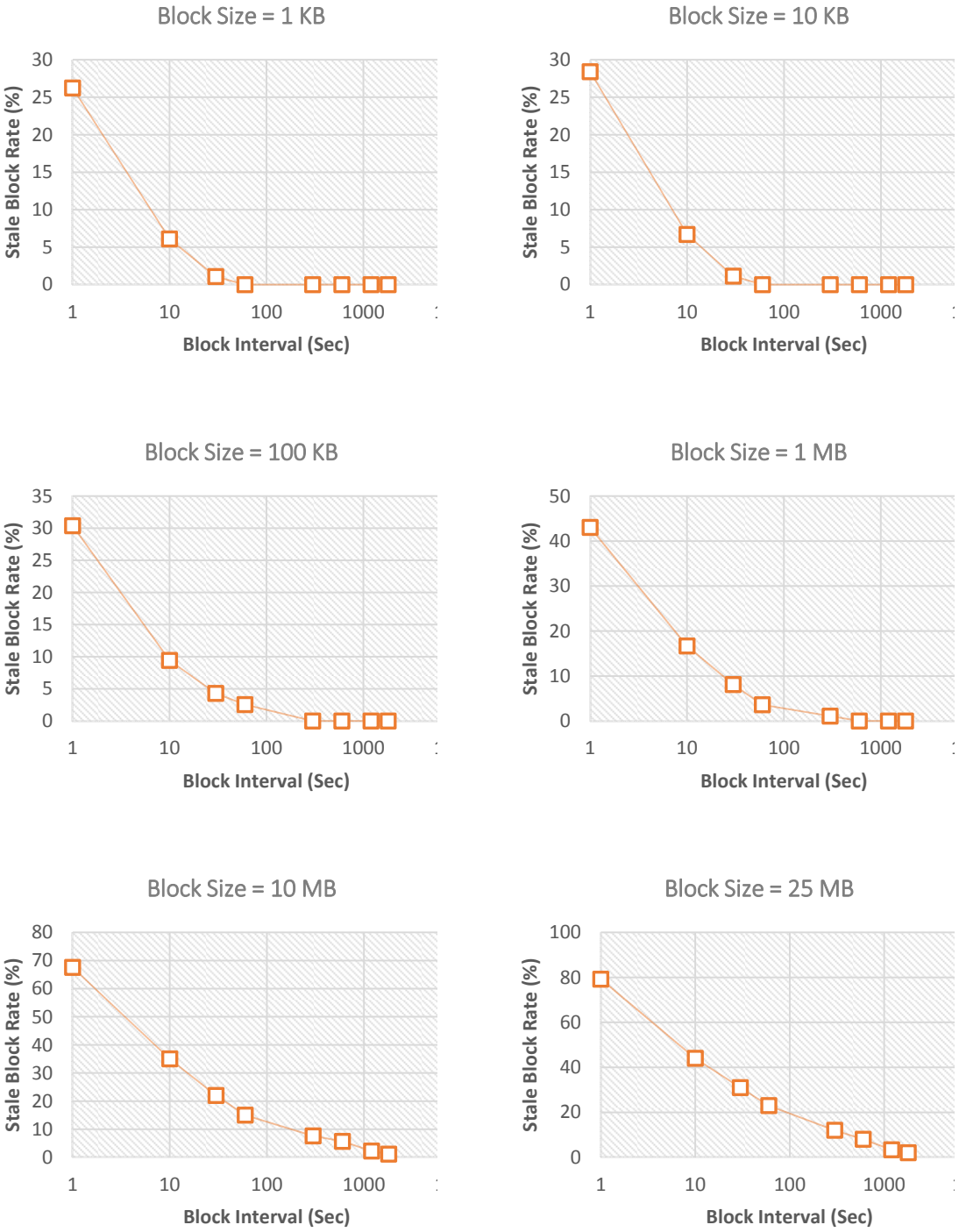



Figure 4-2. Stale block rate as a function of block interval for various block sizes: 1 KB, 10 KB, 100 KB, 1 MB, 10 MB, and 25 MB.

### 4.3.1 Selfish Mining

Based on the standard protocol of blockchains, as soon as a block is found by any nodes, it should be instantly reported to the entire network by the nodes. However, it has been shown that a potential dishonest node can selfishly enhance his/her revenue by deliberately withholding some of the blocks. Understanding such optimal strategies can be used as the means for comparing the security and performance requirements of proof of work blockchains as a function of network parameters. As stated earlier, the goal of the adversaries in selfish mining is not to optimize the overall reward amount, but to enhance the ratio of the adversarial blocks that are accepted to all the blocks in main chain of the network. This malicious activity that can be potentially taken advantage of for tampering with the election results can be modeled by optimizing the relative revenue *rev* defined in equation (1):

$$rev = \lim_{n\to\infty} \frac{\sum_{j=1}^{n} r_{d_j}}{\sum_{j=1}^{n} (r_{h_j} + r_{d_j})} \quad (1)$$

where $r_{h_j}$ and $r_{d_j}$ are the rewards of the honest and dishonest nodes, respectively, in step j. The fact that the objective of dishonest miners in selfish mining is to improve their relative share of the entire reward pool implies that the reward function is not linear, and therefore the problem cannot be simply modeled via typical MDP technique. Instead, the problem should be converted into a family of MDPs, described in [13, 29]. The model used here follows the above-mentioned selfish mining strategy indirectly capturing different network parameters such as block size and block interval. It should be noted that the mining costs will not be considered in this model because the objective of the selfish

mining MDP is to enhance the relative mining share rather than the monetary reward. Figure 4.3 shows the relative mining share of the dishonest network as a function of the stale block rate for two different mining powers of 0.1 and 0.3. This implies that no adversary with a mining power of 0.3, for instance, can harvest more than 30% of the total mining power in the network. Therefore, by using this condition we can ensure to consider the worst case scenario for a malicious activity. In order to stay below a relative mining share of 0.5 in the case of the worst case scenario (mining power of 0.3), we need to ensure not to surpass a stale block rate of ~30%.

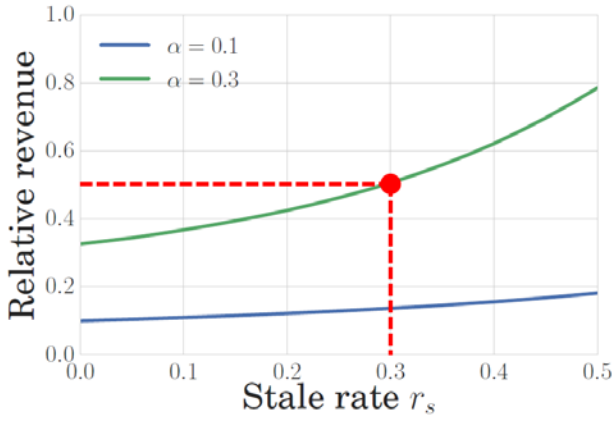



Figure 4-3. Relative mining share of the dishonest nodes as a function of the stale block rate in the network, for mining powers of 0.1 and 0.3 [13].

### 4.3.2 Eclipse Attacks

The MDP model can also take into account the effect of the eclipse attacks in escalating the selfish mining activities. The model assumes that the honest miners are influenced by the stale block rate, while the dishonest portion of the nodes dodge stale blocks and do not mine them. Even though the dishonest nodes are capable of utilizing any mined blocks for their malicious activities, they can easily dodge stale blocks once they go after an honest chain. Practically so, they possess a significantly lower possibility

of mining the stale blocks than the honest portion of the network. The honest nodes get constantly exposed to validation and propagation latencies and naturally they encounter a greater number of stale blocks. Figure 4.4 is the color map of relative mining share showing the impact of eclipse attacks on selfish mining. The graph actually represents the case where the dishonest network misuses the mining power of the honest nodes $\omega$ in order to grow their private chain.

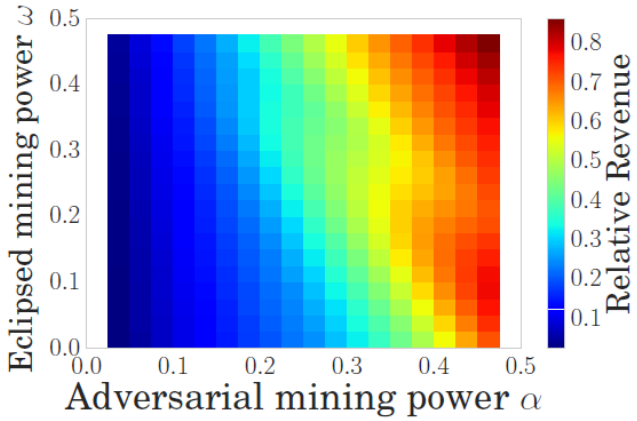



Figure 4-4. The color map of relative mining share (revenue) as a function of adversarial mining power and eclipsed mining power [13].

The graph of Fig. 4.4 can be interpreted in two ways: (*i*) when the eclipsed mining power increases, the maximum tolerable adversarial mining power reduces in order to stay below a relative mining share of 50%. (*ii*) It is also inferred that at a specific adversarial mining power, a more serious eclipse attack, *i.e.* higher value of $\omega$, improves the success rate of the dishonest network in their malicious activities. In other words, a higher eclipse mining power at a particular adversarial mining power results in a higher relative revenue or relative mining share. As a result, when the dishonest network is equipped with eclipse attack a stale block of 30% cannot be tolerated anymore. In order

to also take the effect of eclipse attacks into account we consider a more conservative stale block rate of 10% in our analysis.

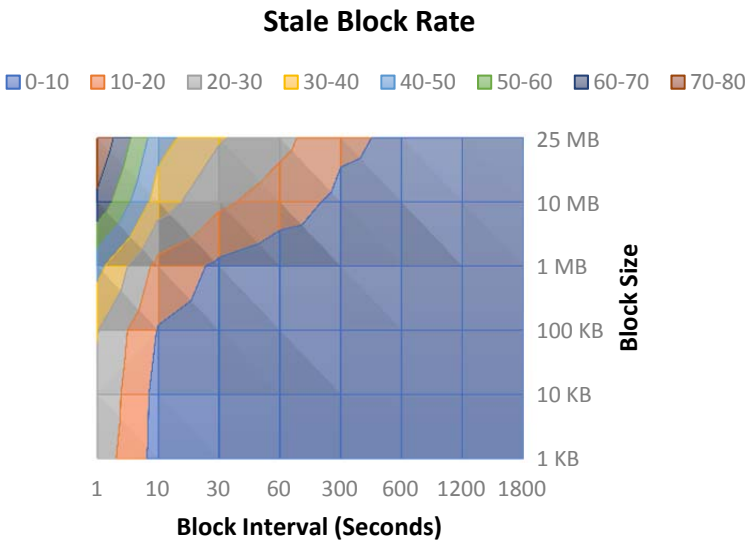



Figure 4-5. The color map of stale block rate as a function of block size and block interval.

In order to consolidate the simulation results of Fig. 4.2, the color map of Fig. 4.5 is generated to provide a more intuitive understanding of various scenarios tested here. Based on this graph, we can easily visualize the allowable combination of block size and block interval values in order to stay within the stale block rate of 10% and lower.

### 4.4 Scalability

In order to investigate the scalability of the blockchain network, we use the notion of throughput defined here as the number of ballots per seconds (bps). To calculate the throughput of the blockchain voting system we should assume a realistic value for the ballot size. Since the size of each virtual transaction is largely determined by the size of the hash and headers, it should be very comparable to that of a potential virtual ballot. In the case of digital currencies, each transaction input requires at least 41 bytes for the

previous transaction reference and other headers while each transaction output requires an additional 9 bytes of headers. Finally, every transaction has a header at least 10 bytes long. Therefore, we estimate the average size of each ballot to be around 100 bytes. The throughput of the voting blockchain for a voting period of 12 hours can be calculated using:

$$N = \left(\frac{S}{B}\right) \times 12 \times 60 \times \left(\frac{60}{t}\right) \quad (2)$$

where *S*, *B*, and *t* are block size in bytes, ballot size in bytes, and block interval in seconds, respectively. Institutively, a higher throughput requires a higher block size and lower block interval. In practice, pushing the limits on the two parameters is not desired as it may jeopardize the security of the ledger. This is because of the higher stale block rate and therefore higher relative revenue (mining share) dishonest nodes can potentially achieve via selfish mining and eclipse attacks. Using equation (2), the throughput of the blockchain voting system for various scenarios of Fig. 4.2 and Fig. 4.5 is calculated and listed in Table 4.2. In this table, the green entries denote allowable stale block rate (10% and below), while the orange entries, correlated to stale block rates of above 10%, represent the block size and interval combinations that pose a danger to the security of the network.

Table 4-2. Throughput of the blockchain voting system for different combinations of block interval and block size.

| Block Interval (Sec) | Block Size | | | | | |
|---|---|---|---|---|---|---|
| | 1 KB | 10 KB | 100 KB | 1 MB | 10 MB | 25 MB |
| 1 | 432,000 | 4,320,000 | 43,200,000 | 432,000,000 | 4,320,000,000 | 10,800,000,000 |
| 10 | 43,200 | 432,000 | 4,320,000 | 43,200,000 | 432,000,000 | 1,080,000,000 |
| 30 | 14,400 | 144,000 | 1,440,000 | 14,400,000 | 144,000,000 | 360,000,000 |
| 60 | 7,200 | 72,000 | 720,000 | 7,200,000 | 72,000,000 | 180,000,000 |
| 300 | 1,440 | 14,400 | 144,000 | 1,440,000 | 14,400,000 | 36,000,000 |
| 600 | 720 | 7,200 | 72,000 | 720,000 | 7,200,000 | 18,000,000 |
| 1200 | 360 | 3,600 | 36,000 | 360,000 | 3,600,000 | 9,000,000 |
| 1800 | 240 | 2,400 | 24,000 | 240,000 | 2,400,000 | 6,000,000 |

It can be observed from the table that a block interval of 10 minutes and block size of 25 MB result in the highest throughput of 18 Millions without compromising the security of the system. Therefore, we introduce the above mentioned combination as the optimum parameters of the network for a blockchain voting application.

Now, let's see if such conditions is feasible for the magnitude of US presidential election. Even though, the number of eligible voters in the US is estimated to be around 250 Millions, in chapter 3 we proposed using precinct-based blockchains in order to address the scalability of blockchain voting. As a result, if we are to assign an individual blockchain to each state, we should decide the feasibility based the most populous state. Historically, California is considered to have the highest population in the United States with the number of eligible voters to be 18.2 Millions. Considering the fact that the

participation rate in the US presidential election is around 60%, the throughput of 18 M obtained with the block size and block interval of 25 MB and 10 minutes, respectively, well satisfies the demands from a blockchain-based voting platform.

# CHAPTER V

## CONCLUDING REMARKS AND FUTURE WORK

In this work, we examined the challenges of using the blockchain technology for building an Internet-based voting system. Several studies have previously argued that the challenges associated with Internet-based voting are too risky to allow a successful implementation. Here, we provide the reasoning and methodologies on how different concerns such as authentication, privacy, transparency, and scalability can be addressed. As for scalability, for instance, which is known to be one of the most gating technical aspects of blockchain-based voting, we introduced a way to mitigate the issue via parallel processing of blockchains based on geographical zones such as states.

Unlike the previous works that only focus on qualitative discussions of blockchain voting and the potential risks associated with it, here for the first time we take on a quantitative study of such platform to understand the performance and security implications of it. To achieve this objective, we took advantage of the blockchain simulator developed at ETH Zurich along with Markov Decision Processes (MDP) model reported in the literature. The simulator along with MDP model enabled us to capture the interaction effects between the performance characteristics of the network, *e.g.*

throughput, block interval and block size, and the security measures of it such as stale block rate and relative mining share of adversarial nodes. Finally, we considered selfish mining and eclipse attacks as the most effective malicious activities that can be done by potential fraudsters, and obtained the optimal and yet very safe block size and block interval of 25 MB and 300 seconds, respectively, resulting in the highest throughput of 18 millions in a 12 hour voting period. Such large throughput along with the parallel processing scheme proposed here very well alleviate the concerns on the scalability of blockchain technology for voting applications.

In this work, the standard block propagation mechanism was utilized throughout the simulations. However, as it is perceived in the literature, the block propagation mechanism can significantly impact the security of the blockchain, since it directly affects the stale block rate. Future works include studying the effect of various block propagation mechanisms and developing an optimal mechanism that can minimize the stale block rate for a similar condition and therefore improve the security of the network. This may also be used indirectly as the weapon to increase the throughput of the blockchain e-voting.

## BIBLIOGRAPHY


[1] M. Daniel, “Blockchain Technology: The Key to Secure Online Voting”, Bitcoin Magazine, Jun. 2015.

[2] U.S. Official: Hackers targeted voter registration systems of 20 states, Associated Press, Sep. 2016.

[3] R. Osgood, “The Future of Democracy: Blockchain Voting,” 2016.

[4] K. Hegadekatti, “Democracy 3.0: Voting Through the Blockchain,” 2017.

[5] I. Kubjas, “Using blockchain for enabling internet voting,” 2017.

[6] A. B. Ayed, “A Conceptual Secure Blockchain-Based Electronic Voting System,” International Journal of Network Security & Its Applications (IJNSA) ,Vol. 9, No. 3, May 2017.

[7] T. Moura and A. Gomes, “Blockchain Voting and its effects on Election Transparency and Voter Confidence,” In Proceedings of the 18th Annual International Conference on Digital Government Research, pp. 574-575. ACM, 2017.

[8] S. Bartolucci, B. Pauline, and J. Daniel, “SHARVOT: secret SHARe-based VOTing on the blockchain,” arXiv preprint:1803.04861, 2018.

[9] A. K. Koç, E. Yavuz, U. C. Çabuk, and G. Dalkılıç, “Towards Secure E-Voting Using Ethereum Blockchain,” 2017.

[10] R. Casado-Vara**1** and J. M. Corchado, “Blockchain for Democratic Voting: How Blockchain Could Cast of Voter Fraud,” 2018.

[11] B. Wang, J. Sun, Y. He, D. Pang, and N. Lu, “Large-scale Election Based On Blockchain,” Procedia Computer Science, 129, pp. 234 – 237, 2018.

[12] E. Akbari, Q. Wu, W. Zhao, M. Yangy, and H. Arabnia, “From Blockchain to Internet-Based Voting,” 2018.

[13] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, “On the security and performance of proof of work blockchains,” In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, pp. 3 – 16, 2016.

[14] J. Seffinga, L. Lyons, and A. Bachmann, “The Blockchain (R)evolution–The Swiss Perspective,” Feb. 2017.

[15] A. Lewis, M. Larsen, and C. Y. Goh, “Understanding Blockchain Technology And What It Means for Your Business,” Asian Insights Office DBS Group Research Google Scholar, 2016.

[16] A. Lewis, “A Gentle Introduction to Blockchain Technology,” 2015. Available: https://bitsonblocks.net/2015/09/09/a-gentle-introductionto-blockchain-technology/

[17] M. Vukoli´c, “The quest for scalable blockchain fabric: Proof-of-work vs. bft replication,” In International Workshop on Open Problems in Network Security, pp. 112–125, 2015.

[18] E. Kaspersky, “Cyber security case study competition-kaspersky,” 2016. http://www.economist.com/whichmba/mba-case-studies/cybersecuritycase-study-competition-2016

[19] Blockchain voting: The end to end process, https://followmyvote.com/blockchain-voting-the-end-to-end-process/

[20] Cutting edge blockchain app development, http://blockchaintechcorp.com/.

[21] M. Kovic, “Blockchain for the people: Blockchain technology as the basis for a secure and reliable e-voting system,” 2017.

[22] K. Lee, J. I. James, T. G. Ejeta, and H. Kim, “Electronic voting service using block-chain,” The Journal of Digital Forensics, Security and Law: JDFSL, Vol. 11, No. 2, 123, 2016.

[23] P. McCorry, S. F. Shahandashti, and F. Hao, “A smart contract for boardroom voting with maximum voter privacy,” IACR Cryptology ePrint Archive, 110, 2017.

[24] T. Moura and A. Gomes, “Blockchain voting and its effects on election transparency and voter confidence,” In Proceedings of the 18th Annual International Conference on Digital Government Research, pp. 574–575. ACM, 2017.

[25] www.votem.com

[26] S. Garfinkel, "PGP: Pretty Good Privacy," O'Reilly and Associates, 1994.

[27] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.

[28] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, *et al*. “On scaling decentralized blockchains. In International Conference on Financial Cryptography and Data Security,” pp. 106–125. Springer, 2016.

[29] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, “Optimal selfish mining strategies in bitcoin,” arXiv preprint arXiv:1507.06183, 2015.

[30] I. Eyal and E. G. Sirer, “Majority is not enough: Bitcoin mining is vulnerable,” In Financial Cryptography and Data Security, pp. 436–454. Springer, 2014.

# APPENDIX

## SIMULATOR INSTALLATION GUIDE

This simulator should be installed in the “Terminal” environment of Ubuntu. One rather simple way to install Ubuntu on a Windows personal computer is to use Oracle VM VirtualBox Manager. Once the Ubuntu is installed on the virtual machine, the following steps will be performed in Terminal:

1- First git should be installed using the following command:

```
sudo apt-get install git -y
```

2- The Simulator should be first cloned using the following command:

```
git clone https://github.com/arthurgervais/Bitcoin-Simulator
```

3- The Simulator is built on ns3. It properly works with versions 3.25 that can be found from the official website: https://www.nsnam.org/ns-3-25/
ns3 should be installed in a new directory as follows:

```
mkdir workspace
cd workspace/
wget https://www.nsnam.org/release/ns-allinone-3.25.tar.bz2
```

The minimal set of packages that are needed to run ns-3 can be installed using:

```
sudo apt-get install gcc g++ python
```

We also need to install open MPI library in order to enable MPI-based simulations and make them more scalable:

```
sudo apt-get install openmpi-bin openmpi-common openmpi-doc libopenmpi-dev
```

4- Untar it with the following command:

```
tar xvfj ns-allinone-3.25.tar.bz2
```

5- For exchanging block messages, rapidjson should be downloaded to the home directory:

```
cd
git clone https://github.com/Tencent/rapidjson
```

6- In this step we need to copy all the files from the simulator and rapidjson to ns3. In order to do so, we first create a new directory named rapidjson under ns-allinone-3.25/ns-3.25. Then we copy the contents of the original rapidjson directory (from the downloaded rapidjson project in step 4) to the newly created rapidjson folder (ns-allinone-3.25/ns-3.25/rapidjson). Then all the files from Bitcoin-Simulation should be copied into the respective folders under ns-allinone-3.xx/ns-3.xx/.

7- Update the following script file ns-allinone-3.25/ns-3.25/src/applications/wscript

- By adding the following lines in module.source:

```
'model/bitcoin.cc',
'model/bitcoin-node.cc',
'model/bitcoin-miner.cc',
'model/bitcoin-simple-attacker.cc',
'model/bitcoin-selfish-miner.cc',
'model/bitcoin-selfish-miner-trials.cc',
'helper/bitcoin-topology-helper.cc',
'helper/bitcoin-node-helper.cc',
'helper/bitcoin-miner-helper.cc',
```

- And by adding the following lines in headers.source:

```
'model/bitcoin.h',
'model/bitcoin-node.h',
'model/bitcoin-miner.h',
'model/bitcoin-simple-attacker.h',
'model/bitcoin-selfish-miner.h',
'model/bitcoin-selfish-miner-trials.h',
'helper/bitcoin-topology-helper.h',
'helper/bitcoin-node-helper.h',
'helper/bitcoin-miner-helper.h',
```

8- Update the following script file: ns-allinone-3.25/ns-3.25/src/internet/wscript.

- By adding the following line in obj.source:

```
'helper/ipv4-address-helper-custom.cc',
```

- And by adding the following line in the beginning of headers.source section:

```
'helper/ipv4-address-helper-custom.h',
```

9- Configure ns3 with the follow command to ensure compatibility and maximum performance (it should be executed under the ns-3.25 directory):

```
CXXFLAGS="-std=c++11" ./waf configure --build-profile=optimized --out=build/optimized --with-pybindgen=/home/bill/Desktop/workspace/ns-allinone-3.24/pybindgen-0.17.0.post41+ngd10fa60 --enable-mpi --enable-static
```

10- As the last step, ns3 should be built using the following command under ns-3.25 directory:

```
./waf
```

11- Eventually, the simulations can be run using the following sample command:

```
./waf --run "bitcoin-test --noBlocks=100 --nodes=6000"
```