QATAR UNIVERSITY

COLLEGE OF ENGINEERING

[Free Chain: Enabling Freedom of Expression through Public Blockchains]

BY

ISRAA ALSARSOUR

A Thesis Submitted to

the Faculty of the College of Engineering

in Partial Fulfillment of the Requirements for the Degree of

Masters of Science in Computing Masters of Science in Computing

June 2021

COMMITTEE PAGE

The members of the Committee approve the Thesis of
Israa Alsarsour defended on 19/04/2021.

Qutaiba M. Malluhi
Thesis/Dissertation Supervisor

Abdelaziz Bouras
Committee Member

Aiman Erbad
Committee Member

Devrim Unal
Committee Member

Approved:

Khalifa Nasser Al-Khalifa, Dean, College of Engineering

ABSTRACT

ALSARSOUR, ISRAA SALEM, Masters : June :2021,

Masters of Science in Computing

Title: Free Chain: Enabling Freedom of Expression through Public Blockchains

Supervisor of Thesis: Qutaibah M, Malluhi.

Everyone should have the right to expression and opinion without interference. Nevertheless, Internet censorship is often misused to block freedom of speech. The distributed ledger technology provides a globally shared database geographically dispersed and cannot be controlled by a central authority. Blockchain is an emerging technology that enabled distributed ledgers and has recently been employed for building various types of applications. This thesis demonstrates a unique application of blockchain technologies to create a platform supporting freedom of expression. The thesis adopts permissionless and public blockchains to leverage their advantages of providing an on-chain immutable and tamper-proof publication medium. The study shows the blockchain potential for delivering a censorship-resistant publication platform. The thesis presents and evaluates possible methods for building such a system using the Bitcoin and Ethereum blockchain networks. Our results demonstrate that the Ethereum blockchain is much more beneficial for our system as it requires a much lower cost than Bitcoin to use as a platform to allow freedom of speech.

DEDICATION

I dedicate my work

To Allah, my creator, my source of inspiration, motivation, and knowledge. To my country, Palestine, and the Arabic and Islamic countries who may benefit from this research. To my family for their encouragement, love, and prayers to finish what I have started and achieved such success. Thank you, my beloved family.

ACKNOWLEDGMENTS

First and foremost, I would like to thank Almighty Allah, who granted me the opportunity to work on this thesis (Alhamdulillah), guided me throughout my life, and blessed me abundantly with notable achievements.

I extend my heartiest gratitude to my parents for their unconditional love, passionate encouragement, and continuous supplication to Allah. It gives me immense pleasure to thank my sister Kata'b for everything she did for me, my brother Salem (and his family), who is my success model. Also, I would like to thank my brother Abd Al-Raheem, who was always available whenever I needed him. Thanks to my brother Khaled who has always been a source of happiness. Moreover, thanks to my best friends Mays Abdul Wahab and Sarah Sheikh, who are my models of determination, and for every help they did to me.

I would like to submit my sincere and deepest gratitude to my supervisor, Dr. Qutaibah M, Malluhi. This thesis would not have accomplished success without his meticulous supervision, admirable patience, and constant support. I would not be whom I am today without having the honor of being mentored by him. I am deeply indebted to him for his highly commendable efforts during my journey in research.

I would also like to thank all my friends and co-workers who have never let me down for their love, support, gifts, and adventures. Special thanks go to Zaraah Alagi, Duaa Almaaitah, Badria Al-Mohhannadi, Maryam Al-Malki, Hemyan Al-Mohannadi, Heba almawajdeh, Taqua Eldiwedah, Nourah Al-Saeed among others.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1. PROBLEM STATEMENT

Blockchain technology is becoming famous for supporting private and secure networks. While the primary application today of public non-permissioned blockchains is to record currency transactions, recently, the thought of using them as a tool for free speech has emerged. Therefore, we aim to use blockchain to enable free speech in an efficient and effective environment in terms of cost, time, and tools to view people's opinions protected from inappropriate censorship by governments and other pressure groups, regardless of the political views expressed. Our goal is to use a blockchain ledger, typically used to store transaction data for storing arbitrary binary data, which we call on-chain storage. The blockchain-published binary can be of any size, but due to the limited block size in blockchain, the data gets divided into chunks that are spread over multiple blocks. This contrasts with other solutions that provide off-chain storage and use blockchain storage to record only the hash of the data located in distinct storage systems such as the InterPlanetary File System (IPFS).

1.2. RESEARCH OBJECTIVE

Freedom of speech became internationally protected by the Universal Declaration of Human Rights (UDHR) [1], announced after World War II in 1948. [2] declared that “Everyone has the right to freedom of opinion and expression; this right includes freedom to hold opinions without interference and to seek, receive and impart information and ideas through any media and regardless of frontiers”. This thesis aims to address the problem of suppression of freedom of speech. Blockchain tackles this problem by using a permanent decentralized ledger that no central entity has control over. This method prevents data from being tampered with because no single dominant authority can alter data [3]. For example, the proposed approach can be utilized in building a social media platform with content that cannot be controlled or blocked by any central authority. While some might argue that our proposed platform users might use it to spread misinformation on the blockchain, the blockchain nature makes publishing misinformation data more costly. Although misinformation is a general problem for any publishing platform, since there is a cost involved in publishing on our platform, this can be a slight deterrent for those who like to spread misinformation. However, this is not a solution to prevent misinformation since if someone is motivated to put this misinformation, money will not be a sufficient deterrent for him. The technical objectives of our proposed research include the following:

- *Identify alternative methods* of adding arbitrary data on public non-permissioned blockchains.
- *Evaluate and compare the alternative methods' performance* and adopt one to use to enable freedom of expression.
- Build an *optimized publishing* tool based on the adopted method.
- Build a *viewing tool* that enables user-friendly exploration of content published on the blockchain.

1.3.MOTIVATION AND SIGNIFICANCE

Relying on social media continues to grow at an exponential rate [4]. Social media allows communication between end-users, which offers them the potential to express their opinions and interact with each other. Recently there have been increased attacks on the freedom of expression on social media, such as YouTube, Facebook, and Twitter. The censorship software monitors people’s online activities and prevents them from accessing unwanted content or deletes what has been posted. According to [5], a total of 64 countries experienced a net decline in diplomatic rights and civil liberties in 2019, with only 37 registering gains.

Furthermore, the existing methods to overcome censorship have failed. An example is domain fronting. Domain fronting makes it possible to hide a real Internet communication and route it throughout a third-party website. Nevertheless, governments have succeeded in stopping this mechanism by blocking all the traffic to the third-party website. The Israeli government presents a clear example of censorship of Facebook posts, Wikipedia content, and any activity/program that uncovers what is happening in Palestine [6]. A program called "Jewish-American on Israel's Fascism: No Hope for Change from Within" was blocked in 28 nations [6] in April 2018. The TV show was a study that shows the aggressive attitude Israelis frequently practice towards Palestinians. Governments are not the only party that block/prevent content on the web. The social medial platform providers contribute to this matter as well. Recently Twitter permanently suspended the account "@realDonaldTrump" of the previous USA president Donald Trump after the misinformation and the risk of incitement of violence he caused on the Internet [7]. YouTube and Facebook followed by doing the same to prevent Donald Trump from posting Facebook posts and YouTube videos that spread hate and violence for his white nationalist supporters by doubting the elective results.

With the growth of social media platforms, it became easy to share information on the Internet. Being able to reach a vast number of people is a positive thing compared to the past. However, the spread of misinformation and fake news also increased. False information could affect one's beliefs and substantially impact political or social impact, affecting political elections results [8]. One example is the misinformation surrounding COVID-19 and its vaccines. Another example of misinformation and its effect was the US elections when Donald Trump tweeted misleading information on Twitter and Facebook. Donald Trump declared early victory before the election result was announced, leading these social media platforms to alter their algorithms to detect and label these posts as fake news. Although social media platforms took actions to limit the spread of misleading information during the election period, Twitter took more than fifteen minutes to label Trump's tweet as obscured [9]. Although blocking content is sometimes needed, there is a fine line between suppressing freedom of speech and protecting society. This distinction is often a grey area, and often the argument of the need to block content is abused for political gains. This research aims at exploring techniques to prevent such abuse.

1.4.THESIS OUTLINE

The thesis consists of 6 main chapters. Chapter 1 states the motivation and significance of the Thesis, in addition to the research objective and problem statement. Chapter 2 reviews the relevant background on the fundamentals of Bitcoin and Ethereum blockchains. Chapter 3 presents our proposed system's methodology, where we discuss the distinctive methods to insert data into two different blockchains, Bitcoin and Ethereum. Chapter 4 lays out the implementation of our proposed system. Chapter 5 exhibits the discussion and evaluation of our proposed approach. Finally, in Chapter 6, we conclude and discuss some challenges and future work.

CHAPTER 2: BACKGROUND AND LITERATURE REVIEW

It is considered necessary that the main problem is clearly understood to determine the solution we are providing. This chapter helps understand how the solution we provide to combat inappropriate censorship would help people express their opinions without the fear of being deleted, blocked, or modified by any third party. It is essential to understand blockchain's technical aspects, emphasizing two leading blockchains, Bitcoin and Ethereum, to understand our solution built on the blockchain.

2.1. CENSORSHIP AND FREEDOM OF EXPRESSION

Freedom of speech is defined to be an individual's right to express or state their opinion without facing any form of inappropriate censorship from any governing body. According to the definition that was given by the International Covenant on Civil and Political Rights, "Everyone shall have the right to freedom of expression" [1]. This freedom is not limited to any form. It consists of the freedom to seek out, receive and impart any information that the individual desires through any medium of their choice. This section provides insight into how inappropriate censorships vary across the world and how people are restricted to access content or only allowed to talk about topics approved by the government. The first section will consist of providing insight into the techniques of censorship that are used. And the second section present example of inappropriate censorship in the world. The third section discusses the conflict between freedom of speech and the need for proper blocking of harmful content.

2.1.1. CENSORSHIP TECHNIQUES

Several techniques are used to block free speech, such as IP blocking, DNS Tampering, keyword filtering, and other methods. These methods track individuals' internet activity and block unwanted content from being accessed [10]. One of the techniques to practice inappropriate censorship on the Internet is IP blocking. IP blocking is used on the Internet to control connections from either a particular IP address or various addresses recognized as questionable or hostile [11]. IP blocking relies on a blacklist of IP addresses of unwanted webpages provided by a third party to the Internet service providers. If a user wants to access a web page, then the web page's IP address is checked within the blacklist. If the IP address is found in the list, then the connection is dropped out.

Domain Name System (DNS) is a service to link between domain names and their corresponding IP addresses. IP addresses are used by web browsers, while humans use domain names to access websites on the Internet [12]. Every device connected to the Internet has a specific IP address. DNS services take the burden from humans of remembering the IP addresses of the websites and other users. DNS Tampering is when the government or any third-party controls the Internet service providers to cancel the DNS registration of the website [11]. If the website is banned from the DNS, the user will get an error stating that the requested website is not found.

The previous two methods work if the third parties have a blacklist list of IP addresses. However, with the massive number of sites on the Internet, not to mention the possibility of having new websites that can be dynamically added, having an updated blacklist is nearly impossible. Besides, third parties will not block all the Internet data that has a taboo subject. Thus, keyword filtering is used to filter the Uniform Resource Locker (URL) of the web pages [11]. In this mechanism, the requested website URL string is checked for any banned keywords, and the connection will be dropped if any keywords are found.

Packet filtering is one of the latest and sophisticated technologies used nowadays to control Internet contents [11]. Any request on the Internet is sent as a packet containing the sender and receiver's IP addresses and contents. These packets are routed from one computer to other using routers to reach the final destination. The previously mentioned methods focused on blacklisting the IP Address and URL string of the banned websites. On the other hand, packet filtering scans the actual content of the webpages for any forbidden content. If a packet contains blacklisted data, the communication will be lost, and a massage unrelated to what happened is displayed to the user. Third parties also use uncommon methods to censor the Internet in addition to the previous techniques inappropriately. Traffic shaping is one of the old methods used to slow the connection to access the targeted website, which indicates that the communication is slow and the Internet is unreliable [11].

2.1.2. COUNTER CENSORSHIP TECHNIQUES

Research has been made to develop strategies to overcome such forms of inappropriate censorship [13]. In this section, we will explore different techniques are used to overcome the block of free speech. However, these strategies have drawbacks and can potentially harm the user, which will be explained later in the section.

In countries where their Internet Service provider only relies on DNS Tampering for inappropriate censorship, changing the DNS provider will restore the blocked site's connection. Changing the Internet setting allows users to change the DNS providers [14]. However, the risk of doing this is that users may connect to a malicious DNS provider. This may result in directing users to poisoned pages to collect user's login information. Another way to bypass the censorship is for website owners to offer their content on multiple websites or domain names.

Using decentralized namespaces is now possible thanks to blockchain technologies (explained in detail in section 2.2), which allow them to exist independently of a single organization's jurisdiction [15]. The BitDNS [16] debate started in 2010, intending to achieve decentralized, stable, and human-readable names.

Like most of the other systems, Blockchain DNS has drawbacks, the most significant of which is that a user cannot easily write an address and receive a response. For such a browser to access blockchain domains, it must first enable specific add-ons.

Another method to escape the unjustifiable blocking of the Internet is using Tor Browser. Tor Browser is open-source software that provides freedom of using the Internet and anonymity if configured correctly [14]. The risk of using Tor Browser is that anyone who can monitor the user's network activities can discover that they rely on counter censorship techniques. Another way to overcome keywords and packet filtering is word conjugations. Internet users would all agree to use common names for a forbidden sense by placing them in a context. This makes it hard for third parties to ban such widely used phrases. Another way to escape censorship is by injecting a link to the restricted websites inside an allowed page; this technique is called link-relaying.

Internet users utilize a Virtual Privet Network (VPN) to bypass any restrictions and access the internet by spoofing the user's location. VPN uses servers to send the user data through the internet. A free or paid VPN service might own the server. If the VPN is configured correctly on the user’s devices, they can use the internet to access any blocked websites. About 416 million users, according to Global Web Index, use VPN to avoid inappropriate censorship or gain more privacy [17]. While VPN overcomes censorship, using free services keeps logs of the user’s activity. It lets third parties such as the government eye-dropping on the web browsing activity, causing substantial privacy risks. A web proxy is similar to VPN, but it is merely a computer that acts as an intermediate between Internet users and the server they want to connect to. Using web proxy and VPN is to hide user's IP addresses. However, VPN also provides more security by encrypting the data users send or receive [18]. SSH tunneling is an SSH tunnel that allows users to send all of their data through an encrypted path, hiding all blocked websites sending requests and responses from censors, who see it as unreadable SSH data [19].

Accessing banned content is a severe offense in most countries, primarily if the content is classified as child pornography, a risk to national security, or hate speech. Some of the mentioned techniques provide circumvention, where the IP address and the computer location are recorded. Thus, users should know what type of bypassing censorship technology is appropriate to use, configure and use correctly. While some of the above techniques' service providers promise to keep their user's logs and information secured, they might be obligated by law to submit them to governments. Thus, immutability (discussed in detail in section 2.2.4) should be essential in any alternative solutions to bypass inappropriate censorship.

2.1.3. CENSORSHIP IN THE WORLD

Freedom of speech suffers immensely in the Arab countries due to the dictatorship regimes that have controlled many countries for the last few decades. In most Arab countries, the press is either under government control or the media is partially owned by the government itself, which allows them to censor what they deem to be inappropriate. The governments in several Arab countries generally impose laws and regulations to ensure that the authorities are given the right to limit media production freedom and give them maximum control over how they can delete content, stop service, or block access. Some Arab government has used censorship strategies such as delaying the Internet connection speed to discourage the distribution of images and videos, track internet usage, and block offensive webpages [20]. While the previous country adopted indirect strategies, another country entirely closed the web for five days in the Arab Spring in January 2011 [21].

Meanwhile, in other countries such as China and North Korea, there are varying censorship levels to ensure that they can restrict people from accessing parts of the internet that could be harmful to the government's image. China has one of the most elaborate internet censorship systems and mainly consists of IP blocking, DNS tampering, keyword filtering, and more to track each individual's internet activity and then block the internet access of those going against their policies and internet regulations [10]. The Chinese government sells all the computers in the country, which frequently allows the government to push an updated blacklist of banned web pages. A clear example was shown when the Chinese government deleted a posted letter on popular platforms Weibo and WeChat [22]. The Letter requested an official investigation for a case of sexual assault of a female student by her university professor, which led her to commit suicide. An anonymous user then published the letter to the Ethereum blockchain, which is now permanently stored in the public domain. Without providing exact parameters and rules for censorship, entities such as social media sites, news sites, and microblogging sites are told to develop strict algorithms designed to filter out the results to ensure that the informational space is even more restricted.

Although the examples display inappropriate censorship over Internet content, the conflict between freedom of speech and proper blocking of harmful content needs to be discussed. Although the First Amendment stated the right to freedom of speech, the Supreme Court later added exceptions to remove the First Amendment's protection on obscenity, child pornography, hate speech, commercial expression, defamation, and potentially offensive speech to minors. Government has the right to block content that contains the previous topics and offer a Severe punishment on who commits such crime [23]. However, censorship practiced beyond these topics considered violating the First Amendment of allowing freedom of exception.

Since inappropriate censorship is displayed to the user as a technological or link issue, users may or may not know they suffer from Internet censorship. Also, even if methods to overcome censorship such as VPN are used, they cannot bypass all the filters. Thus, the leading Internet philosophy is to provide a social media platform with content that cannot be controlled or blocked by any central authority. In this thesis, we investigate two blockchain networks to decide which is more efficient to build a platform to overcome Internet censorship.

2.2. BLOCKCHAIN BACKGROUND

This section gives a detailed insight into how blockchain works and focuses on two leading blockchains: Bitcoin and Ethereum. Understanding blockchain will help in understanding how the proposed solution is effective in promoting freedom of speech in countries where strict censorship rules and policies prevent people from freely expressing and promoting their views [24] [25] [26].

Blockchain is a distributed, immutable, usually public ledger recording transactions and tracking assets. Blockchain peer-to-peer (P2P) architecture allows data to be distributed through the network without being monitored or controlled by any centralized authority [27] and [28]. The assets placed in it can be tangibles, such as currency, resources, intangible products, services, patents, or copyrights. Blockchain initially is the underlying technology behind the Bitcoin [29] cryptocurrency. Blockchain can simply be seen as a data structure comprised of a linked list in which every block has a hash pointer to the previous block. Figure 1 demonstrates a basic form of a blockchain. The hash field in any block is the hash of all the elements of the block. This means that if anything changes in the block, its hash will consequently change. Assuming the data in the second block changes, this means its hash is going to change. Therefore, the prevHash field in the following block will change, which would logically change its hash. Similarly, the change will propagate throughout the whole blockchain. This nature of blockchain data structure allows immutability and data-tampering detection.

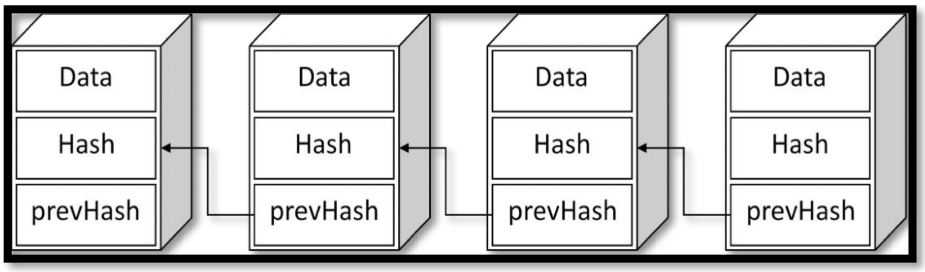


Figure 1: Blockchain as a Data Structure

In centralized systems, the network is controlled by a single authority, which leaves the system exposed to a single point of failure. In contrast, decentralized systems allow multiple parties to have power and decision-making in the network. In a distributed system, its parties are allocated in different geographical locations [30]. Since processing is distributed across several nodes, decision-making may be done centrally or decentrally. Blockchain is considered a decentralized distributed system [31]. This decentralization comes from the fact that all the network nodes participate in the network's decision, i.e., reaching consensus. Furthermore, it is considered distributed as it employs a peer-to-peer network, meaning that anyone can join the network. Figure 2 illustrates the difference between centralized, decentralized, and distributed topologies.

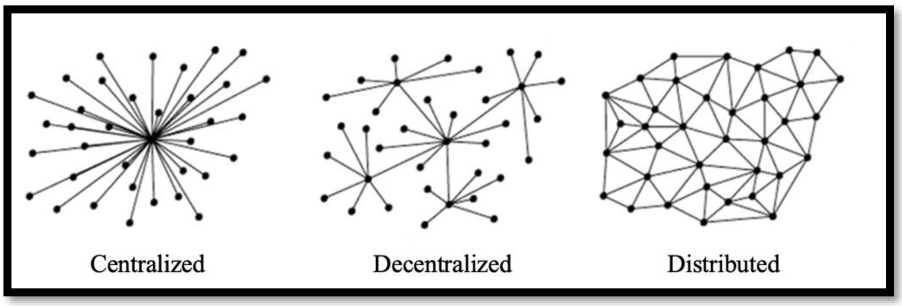


Figure 2: Centralization vs. Decentralization vs. Distributed Networks

One of the essential aspects of blockchain is its miners. Miners are a group of individuals (or nodes) used to validate the network's data by participating in a mining process to create new blocks. Mining ensures that the blockchain remains secured [32]. Suppose Alice wants to send a digital asset to Bob. Alice will make a transaction and broadcasts it to all the nodes in the network. Miners in the network will add the transaction to a block and other transactions once it is validated. The mining process is performed, and the block is added to the blockchain [33].

Consensus, by definition, is the shared view of reality that is agreed upon between the different parts of the system. In centralized systems, the central authority is responsible for deciding the network's behavior and synchronizing its nodes. In contrast, in distributed and decentralized systems, all the nodes in the system determine the network's decision and action. Hence, the node’s synchronization is a challenging task. The blockchain consensus is a way to achieve synchronization in the network and ensure that all the nodes have the correct blockchain version [34]. It can be completed in different ways, including Proof-of-Work and Proof-of-Stake.

To reach consensus, Bitcoin uses Proof-of-Work (PoW) mechanism. In PoW, miners compete to solve a computationally intensive mathematical puzzle. First, they collect the transactions from the transaction pool. Then, they keep trying different random numbers known as the Nonce. As shown in Figure 3, the block's hash is the hash of the nonce, the data, and the previous block [35]. This hash should be less than a specific target which is defined by the difficulty of the blockchain. Once a miner finds this hash, the block is broadcasted to the whole network, where miners validate it. Once the block is validated, it is added to the chain. Through this mechanism, the entire network can agree on one version of the chain [36]. In Bitcoin, the time to solve the puzzle is automatically adjustable every 2016 block (around two weeks) by recalculating the target. The average time between successive blocks is about 10 minutes. In PoW, the incentive behind proposing a valid block is to have the chance of getting a reward that compensates the computing power spent on generating the block. Although PoW proved to be an excellent mechanism to achieve consensus, it has two main disadvantages. It is computationally expensive, and it is exposed to a majority attack. This attack can happen if a node owns more than 51% of the network's total computational power. In such a case, this node can take control over the web and start accepting invalid transactions.

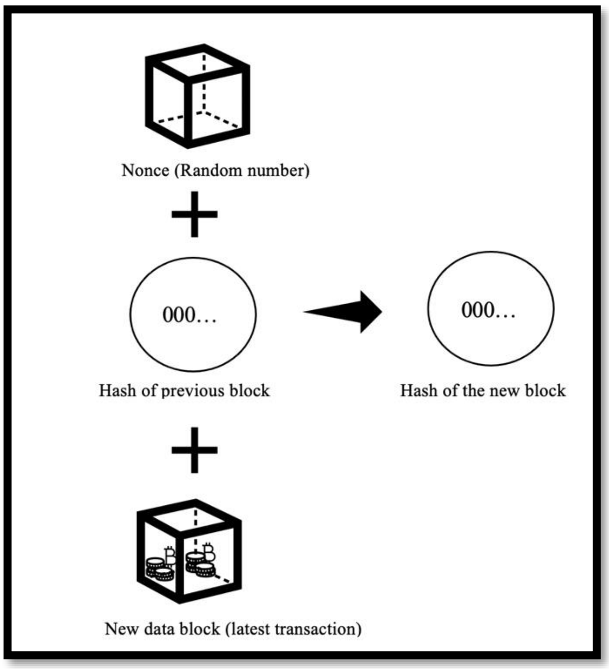


Figure 3: Hash of the Block

Because of the limitations of PoW as it is computationally expensive and is threatened by 51% attack, Vitalik [37] came up with Proof-of-Stake (PoS). Instead of depending on the computational power, PoS is the underlying process that enables validators to obtain enough stake. Users must stake 32 ETH to be a validator on the Ethereum blockchain [38]. Staking is the practice of storing up coins for a set amount of time in exchange for a reward. PoS refers to a method of demonstrating your loyalty to the network by staking, or locking up, a mortgage of ETHER in taking part as a network validator.Validators/ miners are responsible for creating the next block or validate the created one. In PoS, the probability of being selected as the next block producer depends on the miners' number of coins. As the stake increases, the probability increases [39]. Unlike PoW, this approach does not consume energy since they are selected randomly.

Users in PoS are asked to suggest a block when they are chosen as miners or to validate the selected block if they are validators. The validation process is known as attesting. Validators/miners are rewarded for suggesting new blocks or attesting to existing ones. If validators/ miners failed to attest or propose a valid block, they will lose part, if not all, of their stakes. Thus, the stake is used to incentivize the good behavior of the validators/ miners. PoS is not vulnerable to majority attack as the attacker needs to buy the majority of the coins, causing an increase in the coin price, thus, makes the attack very costly [40]. However, PoS has weaknesses and vulnerabilities, such as the *bribe* attack. A bribe attack is made through an attacker where he spends a transaction that he will get later. The attacker starts building secretly another chain connected to the block containing his transaction. The attacker will reveal his malicious chain after his transaction reaches the acceptable confirmation rate and his millicuries chain is longer than the valid chain [41]. Although we explained what PoS is, it is not yet used in the Ethereum mining process. Ethereum still relies on PoW for mining its blocks, and PoS will be used in the upcoming version of Ethereum [42]. The version we have been using support PoW mainnet, while the newer version will be supporting both PoS which Beacon Chain manages, and PoW mainnet. New blocks will be added to the Beacon Chain by stackers. Stakers will not process any of the mainnet transactions. Once the Ethereum mainnet became shared is when Ethereum will fully transition to a proof-of-stake system.

Blockchain offers excellent potential to work in different sectors due to its combination of characteristics. These include transparency, decentralization, and immutability, drawing many researchers' attention over the years. Recently, several applications have been proposed in the literature that deploys blockchain architecture for a wide range of problems, including journalism, censorship-resistant social media platforms, e-voting, etc.

The two most popular blockchain technologies that are available right now are Bitcoin and Ethereum. In the following sections, we will explore the differences between these two technologies, their history, and the benefits that each provides.

2.2.1. BLOCKCHAIN LANGUAGES

Blockchain technology has gained interest among developers around the world. In 2018, the number of blockchain developers reached around 105 thousand out of eighteen million worldwide [43]. In 2020, LinkedIn reported [44] that the most in-demand hard-skill yet challenging to find is blockchain. There are several languages blockchain developers need to know. This section mentions some of them.

There are blockchain languages specific to protocol development. These languages are used to build the blockchain protocols, the consensus mechanism, and the network architecture. C++ is one of the most common languages used in core development due to the capacity to keep memory and Processor use under close control and its ability to verify and propagate blocks quickly. Bitcoin and Ethereum core implementation are built using C++. Golang (Go) [45] is another language used to implement the blockchain core. Go was designed by Google to be a language that combines the productivity and stability of a statically typed, compiled language with the quality of programming of an encoded, dynamically typed language. Go is fast to build languages where users can create an extensive program in few seconds, so developers use it to build core implementation blockchain. Ethereum core implementation was built using Go in addition to C++. Java language was also used to construct the bitcoinj library of Bitcoin protocol [43]. It allows users to maintain their wallets and transactions without the need to have a local copy of Bitcoin Core. Java was also used to build the Hyperledger Besu, an Ethereum client, for its excellent characteristics such as the enormous community, speed, maintains and extensibility.

There is another set of languages used in blockchain for building Smart Contracts where smart contracts are a piece of code that execute automatically if a condition is met. Solidity is an object-oriented programing language inspired by C++, Java, and Python. It is a Turing Complete language created to build smart contracts in Ethereum. Remix a browser that was built to allow developers to write, compile and deploy smart contracts without installing any programs. Although there are other languages to create smart contracts for Ethereum like Serpent, still solidity considers the most popular language used by Ethereum programmers. JavaScript [45] is the most popular language for web development among developers worldwide. Recently it became famous for using as a language to build smart contracts. The fact that JavaScript is object-oriented, prototype-based, and promotes functional programming makes it suitable for blockchain programming. For smart contract creation, Nebulas, and Neo a shared blockchains, support JavaScript and TypeScript.

JavaScript is a common high-level programming language, as previously mentioned. Its widespread use is also chosen as one of the first programming languages for Software Development Kit (SDK) development by blockchain ventures [43]. JavaScript SDK is considering to have the most robust API documentation. Steller and Raiden blockchains have JavaScript SDK that performs the expected functionality of managing a wallet. Rust is another language used to build SDK that can be compatible with iOS and Android platforms. Rust is a lightweight and scalable language; having these features made it famous, considering it was a new language first intruded in 2010 [43]. In 2020, CasperLabs launched a Rust Contract SDK with full functionality to build blockchain applications based on their SDK.

After getting some idea of what languages were used in blockchain, In the following sections, we will tackle the two most popular blockchain technologies available right now: Bitcoin and Ethereum. We will explore the differences between these two technologies, their history, and the benefits that each provides.

2.2.2. BITCOIN

Bitcoin, the world’s first cryptocurrency, was established in 2008 with the publication of "Bitcoin: A Peer-to-Peer Electronic Cash System" by Satoshi Nakamoto [29]. Bitcoin builds on previous innovations like Hash Cash [46] and B-money [47] designed to provide a decentralized electronic currency that is not monitored or governed by a central authority to issue the currency or validate any kind of transaction between two individuals [48]. Unlike traditional currencies with a physical form or shape, Bitcoin is entirely virtual as no physical coin exists anywhere in the world [48]. The virtual coins are used during the transaction to send the value between the sender and the receiver. Users of Bitcoin own either a public key or a private key, which they use to prove the ownership of Bitcoins in the network itself. These keys allow them to sign transactions to unlock their value and spend it by transferring the network's money to its new owners. Possessing a key that can enable them to sign the transaction is the only prerequisite of the expenditure Bitcoin, which essentially puts the control of the money entirely on the user's hands. Bitcoin was made to solve double-spending, which means that there is a risk that digital currencies could be spent more than one time by creating a replica of it, which is a problem prevalent in the digital currency space [49]. Unlike traditional cash, digital currency or a token contains a digital file that could be duplicated by any tech-savvy individual who has a general idea of how it was created [49]. Bitcoin solved this problem by proposing a P2P distributed timestamp server that would be used to generate proof of the transaction and align them in chronological order to prevent duplication of its tokens [29]. The protocols are set to reduce the rate of the reward given to miners every four years. These reduce the rate by half and also limit the number of Bitcoins that could ever be produced to 21M BTC, which randomly reach zero by the year 2140 [48].

Bitcoin is now one of the significant cryptographic currencies on the market. The programming language of Bitcoin is called Bitcoin script. It is a stack-based language since it uses a stack data structure. A stack is a fundamental data structure that can be visualized as a card stack. Two operations are allowed by a stack: push and pop on/from the top of the stack [50]. Push adds while Pop removes the top object of the stack. What is unique about Script is that it supports cryptographic operations for calculating the hash and verifying signatures. Another important characteristic of Bitcoin Script is Turing incomplete, meaning it is not a general-purpose programming language. It does not support loops and recursion. Although this feature limits Bitcoin's abilities, it is considered a security feature as the execution time, and memory usage is predictable. It protects the transactions from logic bombs that might cause a Denial-of-Service attack (DoS). In the Bitcoin Core source code, the maximum allowed script size is 10,000 bytes. However, there is no limit on the number and size of output per transaction. In the Bitcoin Core source code, the maximum allowed script size per block is 10,000 bytes. Theoretically, one can put as much as 750KB of data in one transaction using one or more outputs.

Transaction validation in Bitcoin is not an automatic process. It is achieved by executing Locking and Unlocking scripts placed in output and input scripts, respectively see Figure 4. Both scripts should be combined. The result of both script's execution should be true. Else, the transaction would be invalid to validate that a transaction redeems the output of another transaction correctly.

Consider the example in Figure 5, where it illustrates validating locking and unlocking scripts.

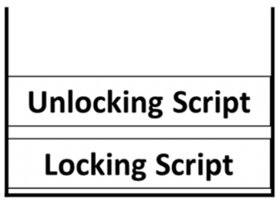


Figure 4: Concatenation of Unlocking and Locking Scripts

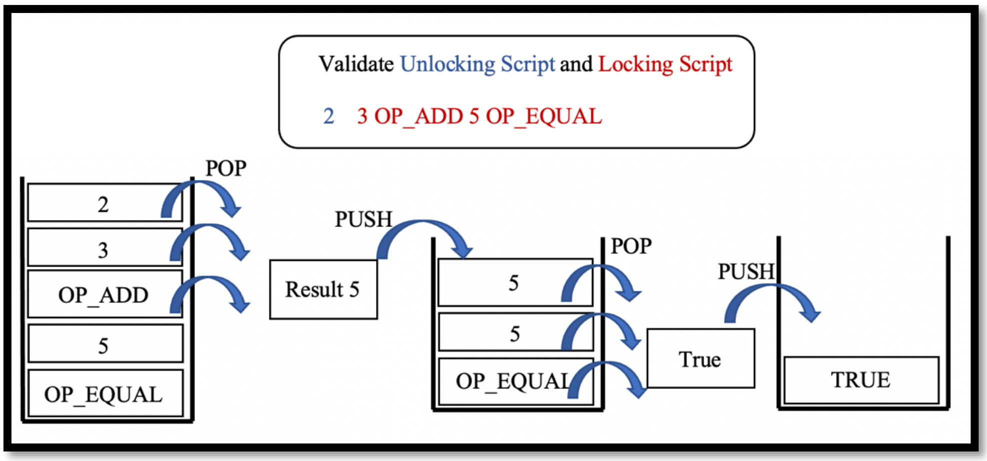


Figure 5: Script Stack-Base Example

A user is identified by a public key (pubkey) of 26-35 alphanumeric characters presenting a Bitcoin address [8]. A random 256-bit number is a private key, and an elliptic curve digital signature algorithm (ECDSA) produces the corresponding pubkey. A transaction in the format of "Alice pays x BTC to Bob" is presented by inputs and outputs illustrated in Figure 6. In our example, an input is where Alice should provide proof of her ownership of x BTC received in previous transactions by unlocking the x BTC calling a script scriptSig. An output contains the locking script called scriptPubKey holds the x BTC and Bob pubkey address, and only the receiver" Bob" will be able to unlock it using his signature and pubkey. Thus, a transaction is achieved by Alice signing the message "reference number, Bob's public key, BTC amount x." The reference number refers to a block w_i in the current BTC chain $w_0, w_1, ..., w_n$, where

Alice received a minimum of x BTCs in a transaction with the provided reference number included in w_i. For instance, the block w_i consists of a transaction with these reference numbers, which shows that Alice received a particular quantity of BTCs.

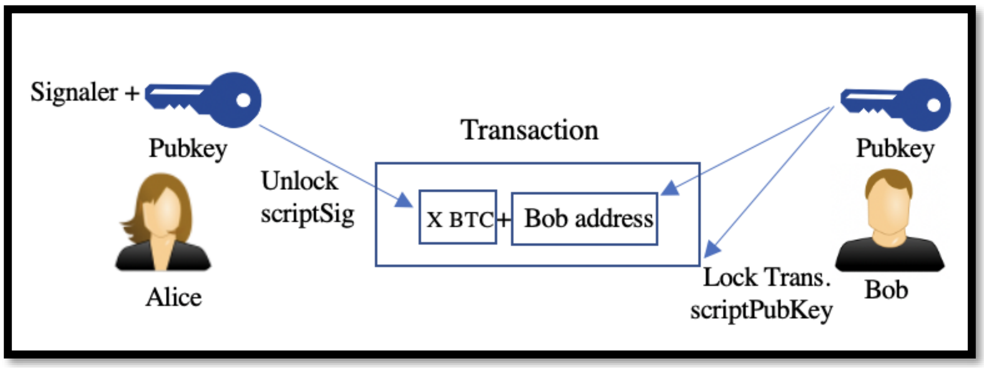


Figure 6: Bitcoin Transaction

It is essential to understand the Standard Transactions of Bitcoin since we will be using them in our proposed solution. A transaction is considered to be standard if it succeeds in passing Bitcoin Core’s IsStandard() and IsStandardTx() tests [11]. Bitcoin support multiple standard transactions:

a) PAY-TO-PUBLIC-KEY-HASH (P2PKH): P2PKH is the default transaction in Bitcoin Blockchain. The locking script known as ScriptPubKey is consists of the public key hash of the recipient. A script of the receiver pubkey and a valid digital signature should be provided, known as ScriptSig, to unlock (spent) the transaction.

b) PAY-TO-PUBLIC-KEY (P2PK): P2PK is a simpler version of P2PKH. The difference is that the locking script is now the pubkey instead of the pubkey hash, and the unlocking script is the only signature.

c) PAY-TO-MULTISIG (Multisig): Multisig is a script where it has many **n** pubkeys as a locking script and requires some or all **m** signatures to those pubkeys to unlock the script.

d) PAY-TO-SCRIPT-HASH (P2SH): P2SH is a more advanced script used mostly for Multisig transactions. The locking script contains the custom redeem script hash. And to unlock it, the redeem script and the data needed to unlock it should be provided. It allows users to create their own redeem scripts and share them easily with others making the burden on the sender's sender to give the redeem script and data needed to unlock (spend) it.

e) OP_RETURN: The opcode OP_RETURN was introduced to Bitcoin blockchain due to the growing number of users trying to save arbitrary data in the blockchain abusing other standard scripting opcodes such as P2PKH. Opcode OP_RETURN can store 80 bytes of random data in each transaction. A transaction can have many outputs, but only one OP_RETURN output transaction is allowed for the transaction to be considered standard [51]. OP_RETURN script always assesses to false, thus creating unspendable UTXO. Miners can safely remove the output transaction of OP_RETURN from the UTXO set and do not need to keep track of them.

f) COINBASE: An input script used only by the miners is referred to as the "Coinbase". This data provides up to 100 bytes of arbitrary. For holding ASCII encoded string, for instance, names of their mining pools, miners are at liberty to manipulate these bytes of the Coinbase data. For example, the following text was added by Satoshi Nakamoto using the Coinbase data: "The Times 03/Jan/2009 Chancellor on the brink of the second bailout for banks" [52], when Nakamoto firstly used the Bitcoin blockchain. Whereas this field is an approach to store arbitrary data in the Bitcoin Blockchain ledger, only miners can access it, and general Bitcoin users have no access to it. Thus, this section contains it for attention only, and the same will not be mentioned for a second time.

2.2.3. ETHEREUM

As Bitcoin became more popular, researchers spent their time determining uses of blockchain other than in cryptocurrencies. In 2013, Vitalik Buterin published a white paper [37] to propose a new and functional use of blockchain that could allow one to utilize it in different applications. This proposal was for the Ethereum blockchain, and, after getting interested and attracting financial and technical support, the Ethereum Foundation was founded [53].

Bitcoin scripts are a beneficial way of creating and designing different types of transactions. However, they have their limitations. Ethereum blockchain system includes a Turing-complete programming language that allows creating different types of smart contracts. A smart contract is an automated computer program that can automatically transfer digital assets within the same blockchain. A smart contract by nature is immutable since its code is on the blockchain, and it can host transactions without any human interaction [54]. Since 2016, Ethereum became the second most used blockchain-based cryptocurrency. Decentralized applications (Dapp) are software applications or services that operate and run on a blockchain or P2P network of devices rather than a single device and are uncontrolled by a single authority. Dapp can easily be supported by Ethereum smart contracts, allowing users to program the blockchain and build blockchain applications [55]. Each smart contract has a unique address, value, state variables, and functions. The smart contract is executed when transactions are sent to it, which results in changing the state variables in the contract, and it might cause sending new transactions to other addresses. The transaction's sender has to pay for each step of the 'program' they activate, including computation and memory storage costs. The user is then able to create a new contract by deploying code to the blockchain.

The run-time environment for Ethereum smart contracts is based on the Ethereum Virtual Machine (EVM). Solidity, a Turing-complete Ethereum scripting language, can create an operation that the EVM can run. All nodes run EVM in a P2P network where all nodes perform the same smart contract computations. EVM can read and write to the blockchain both executable code and data. It also can verify digital signatures, and it will not execute any code unless a digital signature verifies the message.

Moreover, the Ethereum blockchain supports two types of accounts. The first is known as a contract account and involves establishing new accounts with relevant code. The other type is Ethereum Externally Owned Accounts (EOAs). These accounts support transactions with various fields in common, including Nonce, gasPrice, gasLimit, Data, and others. Additionally, a contract creation has a unit field that holds the EVM code required to initialize the account [56]. Ethereum also supports contracts that can generate logs through events, which are informative messages are stored in each Ethereum blocks transaction log; each event is associated with the address of the contract that triggered it.

The transaction fees in Ethereum are called Gas, where it helps avoid unnecessary or infinite loops since the language is Turing complete. The gas of transactions charges to match the amount of work they do. The sender of a transaction specifies the gas limit where it indicates how much gas he is willing to pay for and the gas price per gas unit. The gas paid goes to miners, and they can refuse a transaction with a low gas price. Ether is the native cryptocurrency used and supported by the Ethereum blockchain. Like Bitcoin, the network in Ethereum is a public permissionless blockchain, whereby anyone can access the ledger. The block gas limit in Ethereum is about 15M gas [57], which was 10M gas when we started the research. It fluctuates depending on the miners in the network, where they bid to raise or reduce the block size of the Ethereum block.

Ethereum has three Merkle Patricia trees; State, Transaction, and Receipt trees, in which each stores some type of data see Figure 7 for more clarification. The Receipt tree saves the information resulting from emitting an event log, such as block number, block hash, transaction hash, gas used by current transaction, logs created by the transaction, and more. The Transaction tree contains parameters such as nonce, gas price, gas limit, recipient, transfer value, transaction signature values, and account initialization for contract creation and transaction data for message call [56]. A transaction that deploys a contract is stored in a State tree, such as the post-transaction state, the accumulative gas used, logs created from executing the transaction, and the logs' information. Our goal is to take advantage of these Trees to impend arbitrary data and evaluate the cost and speed they need.

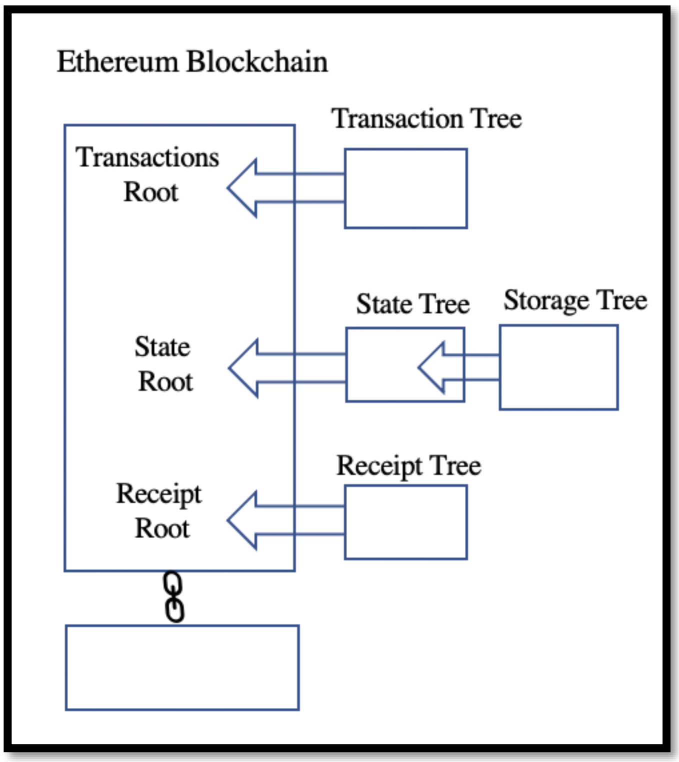


Figure 7: Transaction, State, and Receipts Trees in Ethereum

2.2.4. BLOCKCHAIN APPLICATIONS

The use of blockchain for social media platforms is substantially studied in the literature. Multiple works explore such solutions. They focus on replacing the current social media platforms, i.e., Facebook, Twitter, YouTube, etc., with blockchain-based alternatives. Moving from the legacy platforms that are centralized to blockchain-based solutions will protect the user's data from a central authority and third parties.

DLive [58] is a social media platform that replaces streaming platforms like YouTube and Twitch. Such platforms have grown into billion-dollar companies as they profit from the content creators and viewers. In DLive, the user is the platform's owner as it is based on the Lino blockchain. Unlike legacy platforms that take cut from the content creator reward, DLive rewards the video creator and even the viewers without taking any platform fees. Lino-based applications have three main principles: no platform cuts, rewarding incentives, and decentralized ownership. The first two principles help maintaining the long-term growth of the network and the decentralized right ensures that a single authority does not control the network. LINO points are the fundamental value units in the Lino blockchain. These points are distributed among the system contributors classified into content creators, viewers, Lino App developers, infrastructure providers, and validators [59]. Currently, DLive main focus is user growth and developing a healthy community. However, App developers can later earn some profit by following the following methods: Lino App Reward, Ads. or Lino Stake Rewards (LS Rewards) [58].

Like the Twitter microblogging platform, Freitas proposed Twister [60] blockchain-based microblogging framework. The platform has features similar to that of Twitter which makes it adoptable by the users. These features include posting, direct messaging, hashtag, and mentioning. Twister's main design concepts include scalability, independence, and security, and it uses encrypted communication to provide integrity; most importantly, it gives privacy where the IPs are hashed. It consists of three independent network layers; the first layer utilizes Bitcoin protocol to provide user registration and authentication. The second layer is Distributed Hash Table (DHT) that includes key/value storage. The third layer is based on BitTorrent protocol to provide near-instant notification delivery to the users. The platform uses the same mechanism used by Bitcoin to void double-spending to guarantee the users' uniqueness. Thus, when a new user registers, access is not granted directly. The block should be endorsed first. In the second layer, DHT routing is utilized to request data from other users. Another usage of DHT is the direct sending of notifications between the users.

Several works have introduced conceptual models for implementing blockchain-based social media platforms. One example is Ushare [61]. It is a user-centric blockchain-enabled social media network that enables the users to control and prove ownership over their content by using a scriptable relationship system. Ushare comprises four components: the blockchain, a relationship system, a hash table, and a PCA. The blockchain is intuitively used to record the data and ownership of the system's content. As for the relationship system, it is a Turing-complete programmable unit that is used to ensure traceability and to maintain ownership. The hash table, on the other hand, holds the encryption of the content on the system. Finally, the Personal Certificate Authority (PCA) manages a client's circle and shares content within a circle securely. Additionally, it ensures that data is encrypted and stored in the hash table before it is broadcasted. The PCA is built on top of existing bitcoin wallets. Ushare and such applications can be either be made independently or on top of the existing blockchain. This includes the possibility of building such an application on top of the Ethereum blockchain. Another platform is Enigma [62], a decentralized blockchain platform by MIT. Other platforms include, but are not limited to, BlockStack, DECENT, MultiChain, and Hyperledger [61].

Also, Block Notary Stampery and Verify can be used for copyright protection and for protecting an individual's intellectual property, i.e., their content that they would share with people. These utilize IPFS to provide content-addressable decentralized options to ensure that the content is verified and associated with the user that created them. This can be used to ensure that no individual could claim the content that the individual posted or claim its rights. It can be useful to ensure that no one can restrict the freedom of expression an individual has when using the application. If this is not taken care of, people could claim the content for various reasons, including stealing the content or falsely removing the content based on differing reasons.

2.2.5. PRIVACY, ANONYMITY, AND IMMUTABILITY

This thesis aims to overcome inappropriate Internet censorship caused by third parties, including blocking, deleting, and tampering with user data. Our solution must contain essential properties such as privacy, anonymity, and immutability to protect users and ensure their published data integrity. This section will discuss what each terminology means and how they overcome blocking freedom of speech.

Blockchain has become a recent breakthrough towards enabling secure computing to have a centralized authority over the networked public system. Blockchain is considered a distributed ledger from the data management side, where it organizes transactions by setting them in a chain of blocks. From the security point of view, since blockchain is built and sustained by a P2P network, security is obtained through smart and decentralized cryptography exploitation with crowdsourcing.

One of the main aspects of privacy in blockchains ledger is the utilization of private and public keys [63]. These use asymmetric cryptographic algorithms that are used to secure transactions between users that are in the network. The public and private keys are assigned to each person. Such keys are connected cryptographically, and they are arbitrary strings of numbers. It is proven mathematically that no one can guess the private key knowing the public key [63]. Thus, sharing public keys in the network gives no personal information. This adds a layer of security and safeguards the blockchain's data from hackers. Each blockchain user has an address retrieved from his public key. The use of these addresses is to send and receive transactions on the blockchain. Due to the transparency characteristic of blockchain, any user can view previous transactions [64]. However, Public addresses in the block cannot disclose the identities or private data of the users. Instead, the address behaves like pseudonymous identities used during the transaction to keep the parties' identities secured [65]. Nakamoto advises blockchain users not to use their public addresses more than once. By doing that, one can avoid the probability of an adversary trying to delouse all his previous transactions trying to reveal some private data. A digital signature provides a level of security relaying on private keys. Users use their private keys to access their assets and wallets on the blockchain, giving identity authentication. Suppose a user wants to send currency to another user. He must supply a digital signature driven from his private key. This mechanism prevents hackers from theft or fraud of user data.

It is desirable to offer anonymity, to create a freedom of expression supporting platform. Anonymity can be achieved by achieving two concepts: pseudonymity and unlinkability. A pseudonym is a name (holder) that does not disclose one's real name or personal information. In Bitcoins blockchain, this is achieved by using the public keys. Unlinkability generally means that different user interactions with the system should not be linkable to each other. In the context of a social media platform, the same user's posts or content must not be linked together to their pseudonym or real identity.

Immutability is considered to be a fundamental and essential component of blockchain. Because the transactions made in the network cannot be deleted or edited once that is successfully verified and added into it, each block added to the web contains the hash of the parent block or the block before it. The hash is generally calculated using a Merkle Tree, which generates a cryptographic hash of the block itself and is placed in all transactions [66]. Merkle tree is a data structure constructed by hashing pairs of transactions recursively until a root node is created. These are used in Bitcoin to summarize the transactions and produce an overall digital fingerprint of it. The hash algorithm that is used in the Merkle Tree consists of double-SHA256. In Bitcoin, a simplified payment verification based on Merkle Tree reduces the computation effort and the size low [66], [29]. However, in Ethereum, a variation of the Merkle Tree is used called the Patricia Merkle Tree [64].

2.3. BLOCKCHAIN-BASED DATA STORAGE

This section will tackle two types of blockchain-based data storage, namely, off-chain and on-chain storage. What we mean by these and the advantages and disadvantages are discussed in the following subsections.

2.3.1. OFF-CHAIN STORAGE

The browser uses URLs to get access to web pages requested from centralized servers. InterPlanetary File System (IPFS) was introduced to the world in 2015 [67]. IPFS and Swarm are a distributed storage platform that works on a P2P network where each user-node in the network saves part of website site data that the user requested, and the loaded data is collected from all the nearby nodes. Users might use a "public proxy" to reach IPFS/Swarm data when they do not intend to behave as a local node. IPFS and Swarm introduce the idea of saving data in a distributed manner. One of the applications is off-chain storage. Off-chain storage does not necessarily mean not using blockchain to store user’s data. It indicates that the data is not publicly accessible. An application to use off-chain storage is to keep data using distributed file systems such as IPFS and Swarm or any centralized cloud services. Users save data on one of the places mentions earlier off-chain, while a pointer to where the data stored is kept on the blockchain [68].

The off-chain transactions are faster than on-chain transactions since the network is smaller. Moreover, off-chain is cheaper since it is mostly free. However, the transactions are not publicly available on the public ledger of the blockchain. For instance, if Alice wants to send X BTC to Bob using an off-chain transaction, they have to provide the hash of the off-chain transaction on the blockchain. A proof of Alice giving something to Bob is there on the ledger. However, it did not provide how many bitcoins were sent and received [69]. Decentralized data storage relies on blockchain to sustain its strategies. Blockchain and decentralized storage, on the other hand, are not associated, and one can function without the other.

Filecoin [70] is a P2P network infrastructure that provides storage facilities using blockchain and native cryptocurrency. The cryptocurrency FIL supports all transfers. For users to store their data, they have to pay FIL to data nodes for doing so. The blockchain ledger keeps track of transactions and proves that miners are correctly saving files. IPFS is the foundation of Filecoin. A particular company does not set the price of Filecoin; instead, it is determined by a free market in which everyone may participate. While this provides more freedom, it can be challenging for organizations operating under strict storage limits to quantify "gas fees," which are expenses of storing messages. Filecoin is only in a testing phase, and many of its features are still being developed. Filecoin framework is only compatible with Linux and macOS computers, not Windows.

Like Filecoin, Sia [71] is a distributed data storage system that encodes and distributes files throughout the P2P network using blockchain. Sia's server divides every file into 30 units and passes them to various hosts. It is also compatible with Windows, Linux, and macOS. Sia employs Reed-Solomon erasure coding to guarantee duplication, which allows a file to be recovered using ten parts only. Sia also applies the Threefish algorithm to encode the data before sending it to the network. Besides, Sia provides an API for developers can use to create apps. To store 1 TB of data, the user needs to pay around 2$ monthly. However, the users will have to pay additional fees for contract formation fees and bandwidth fees for accessing the data.

Storj [72] is a significant rival to Sia, providing related services such as blockchain storage. Similar to Sia, the Storj server divides a file into 80 units. The stored units are saved encrypted across the network. Nodes need 30 segments to reassemble the file. While Storj convinces users to store their data in its service, Tardigrade is the actual service doing the storing. There is a lot of debate whether they belong to the same company or not. Either way, together, they provide a perfect place for users to store their data in a distributed manner. The cost they provide for saving 1 TB of data is 55$ monthly, and they only support widows, and they are working to support other operating systems.

Storing data on distributed file systems has many challenges. One of them is encouraging users to participate in the network by using their personal storage space to save other’s data. To solve this problem, the file owner has to pay the other nodes to store his data. The previous challenge leads to another challenge which is having a malicious node. Suppose a malicious node implants a program that generates a hypothetical large file. The nodes can free their spaces where it appears to other nodes. They are storing data. KopperCoin [73] suffers from this challenge since a user has to destroy many coins if he is willing to keep a file for a fixed time. And the users are going to gain some coins from the transactions containing the file. The system did not mention any relation between the coin gained and the destroyed coin, leading to the malicious node attack. Recovery is another challenge these distributed systems face. Even though the idea of inventing them was to overcome the immutability issue, it is not guaranteed to recover or retrieve files saved on the distributed system if the nodes misbehave or disappear from the network. This may occur if users use Permacoin [74]. Another challenge facing these distributed systems is that any third party can block them. Therefore, the idea of storing data itself on blockchain due to all the security and immutability it provides has emerged between researchers introducing on-chain storage. On-chain storage provides immutability to the data since it has impregnability against being destroyed, which helps achieve freedom of speech.

2.3.2. ON-CHAIN STORAGE

With the popularity of Bitcoin, Ethereum, and other cryptocurrencies such as ZCash and Monero, it can be seen that researchers have paid more attention toward establishing a trust-based model that is decentralized in nature. This would ensure that the data remain protected from any outsider that may want to delete or modify it. Along with that, blockchain provides insight on how advancements are being made so that it can be used to host the data inside it while keeping them secured.

However, the previous section's applications use blockchain to store the hash of the data or use the blockchain for registering the users in the system. These users save the data in decentralized systems such as IPFS, Swarm and store data location on blockchain. This provides proof of existence but does not assure that someone might change/delete the external storage data. However, these applications offer many advantages to the world, the lack of the immutability factor, and other challenges mentioned in the previous section. Thus, researchers such as [75] have determined and surveyed methods and strategies to insert arbitrary data into a blockchain called on-chain storage. These consist of methods that have been defined in the past and lesser-known methods optimized for efficiency. The methods are compared based on how efficient these are, the cost of the function, the convenience it provides for reconstruction of the data, and the negative impact on the blockchain itself. Some of the methods that are discussed include P2SH (Pay-to-Secure-Hash), P2FKH (Pay-to-Fake-Key-Hash), P2FSH (Pay-to-Fake-Secure-Hash), OP_RETURN algorithms [75].

Based on the assessment carried out, it was found that, when comparing the efficiency and the costs of the methods, it can be seen that the P2FKH and the P2FSH are considered wasteful and not scalable, as the data can only provide 20 bytes of data per UTXO, which is not spendable. Meanwhile, the likes of P2FMS are considered to be cost-effective as it can store about 195 bytes per UTXO, which is unspendable, by utilizing three uncompressed keys [75]. OP_RETURN was also considered to be cost-effective as it was seen that it was prunable and has a significantly low cost, which can be ideal for storing small amounts of data up to 80 bytes [75]. For more extensive data, however, it can be seen that Data Drop without the use of the signature method would be cost-efficient and would be able to provide the least amount of data overhead [75]. With all these experiments that explain how much storing data on the blockchain might cost, it needs to be extended to other blockchain platforms like Ethereum.

Besides, some services exist to help people publish their opinion on the Bitcoin blockchains, such as Apertus Service [76], Satoshi Uploader [77], P2SH Injectors Service [78], and Crypto Graffiti Service [79]**.** The disadvantages of these services are that they have to pay an extra fee to publish content. And it will not allow anyone to express their thoughts and opinions without being restricted to the platform's rules and regulations. The following explains what each service uses to allow users to publish their opinion on the Bitcoin blockchain.

- According to the creator of *Crypto Graffiti Service* [79], this is a web-oriented service that enables the user to read messages and files from Bitcoin's blockchain and also allows the user to insert dating to into the blockchain. The main features include the possibility of inserting data through multiple P2PKH output scripts in one transaction. Moreover, it allows a maximum data storage of 60 KB. The Crypto Graffiti search transactions with a minimum of 90%

printable characters or an image file if they wish to view the data inserted before. 10% service charges are charged along with the message cost, charged depending on the message size.

- *Satoshi Uploader Service* [77] allows data insertion in one transaction using multiple P2X outputs. The service offers data storage features and a length field, and a CRC32 checksum code that detects errors in data for convenient decoding.
- *P2SH Injectors Service* offers various services [78] that allow data insertion through multiple P2SH input scripts. This service allows the storage of data chunks in P2SH input scripts. This service enables users to produce their P2SH redeem scripts containing hash values of data chunks and their consequent verification to maintain the file's integrity and protect it from unauthorized changes.
- *Apertus Service* [76] breaks data into fragments to facilitate its insertion in multiple transactions through a random number of P2PKH output scripts. This is followed by referencing these fragments in an archive given in the blockchain. These data fragments can be recovered and re-arranged. Data may be accompanied by a comment, file name, or digital signature, depending on the selected encoding option.

CHAPTER 3: METHODOLOGY

With Ethereum and Bitcoin blockchain’s help, our proposal's key goal is to help demonstrate and accomplish the freedom of speech. As discussed in Chapter 2, blockchain's primary application is to record crypto-currency transactions on the chain distributed ledger. The research aims to figure out the techniques for employing blockchain as a tool for free speech. In this chapter, we identify and review different methods to insert data in Bitcoin and Ethereum blockchains. These methods can, later on, be used by the developer to build a standalone social media platform where people can publish critical posts that are permanently stored in the ledger.

3.1. ALTERNATIVE DATA INSERTION METHODS IN BITCOIN

Insertion of arbitrary data in the ledger through various methods is enabled through the Bitcoin blockchain. Examples of arbitrary data can be files and short messages. As we have discussed in chapter two, Bitcoin transactions are created using Forth-like scripts [80]. A script describes how the Bitcoin receiver can spend the received Bitcoins. For uncommon Bitcoin transfers, the script specifies that for Bob, the recipient, to spend the received Bitcoins, Bob has to present his public key (pubkey) and digital signature where his pubkey hash equals his ID embedded in the script. The digital signature will prove the proprietorship of the private key, which corresponds to the provided pubkey. A transaction is valid if the combined script does not trigger failure, and the top stack item is non-zero when the script exits. The stacks hold byte vectors of at most 520 bytes long. The script words are called opcodes (also known as commands or functions).

Our proposed solution aims to include a maximum number of non-transaction-related bytes into the script without triggering a failure. Miners will consist of the transaction in the block. Furthermore, miners will gain the transaction fees users put to insert the desired data in the Bitcoin ledger in addition to their mining reword. There are several ways for arbitrary data such as short messages and files to be injected into the Bitcoin ledger. We will illustrate how to embed data using the five standard transactions in bitcoin in the following subsections.

3.1.1. OP_RETURN

The opcode OP_RETURN is the first standard method. There would be an opcode followed by one push data op, and in this arrangement, 80 bytes of data that is not related to the transaction itself (e.g., human-readable messages) per transaction can be stored [81]. Users can use OP_RETURN only once, which is one requirement for a transaction to be a standard transaction. If more than one OP_RETURN needs to be used, they will require multiple transactions. Nonetheless, users cannot control the sequence of these transactions to be mined by the miners. Insertion of small amounts of data (or transaction metadata) can easily be managed by this method. Figure 8 illustrates a graphical presentation of inserting data using this method.

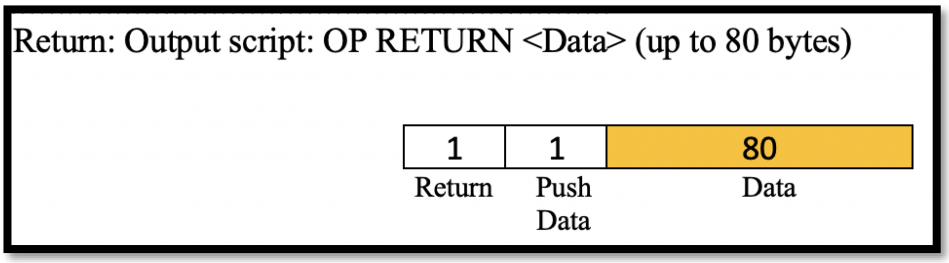


Figure 8: Return Transaction

3.1.2. PAY-TO-FAKE-KEY-HASH

The standard Pay-to-Public-Key-Hash script is a well-known and debatable data insertion technique, wherein the data in the <PubKeyHash> field of the output script is stored by a non-dust amount of Bitcoin. This is known as Pay-to- Fake-Key-Hash (P2FKH). There is no public key with the user, which can be hashed on the data being stored. Due to this, users can never spend these transaction outputs. Figure 9 shows a graphical illustration of this method. Nevertheless, the miners are unaware of the hash denotes a real public key possessed by someone. As valid Unspent Transaction Outputs UTXOs, these UTXOs must be recorded (forever) by the miners. The P2FKH method provides the storage of 20 bytes per output; however, a single transaction can entail several outputs. In Bitcoin's blockchain, the images, text, and mp3 files have been stored through this method, and now it is the method utilized by tools such as Apertus.io [48].

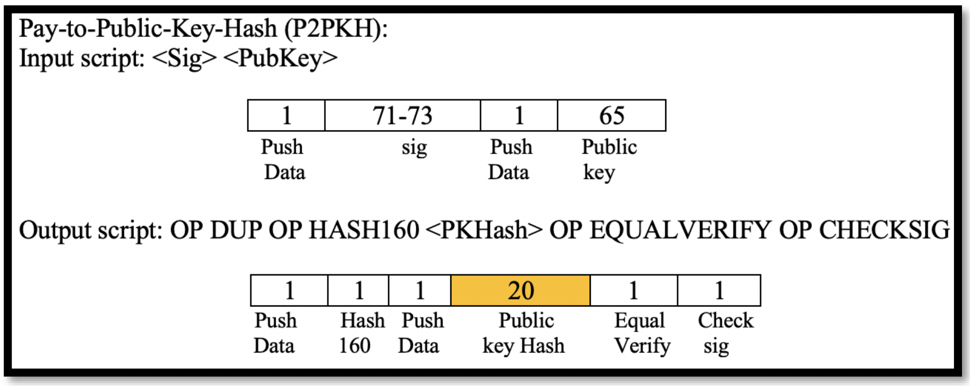

Pay-to-Public-Key-Hash (P2PKH):
Input script: <Sig> <PubKey>
1
71-73
1
65
Push Data
sig
Push Data
Public key
Output script: OP DUP OP HASH160 <PKHash> OP EQUALVERIFY OP CHECKSIG
1
1
1
20
1
1
Push Data
Hash 160
Push Data
Public key Hash
Equal Verify
Check sig

Figure 9: Pay-to-Public-Key-Hash Transaction

3.1.3. PAY-TO-FAKE-KEY

Rather than a fake public key hash, you can store the data as a fake public key (P2FK). There are 65 bytes in an uncompressed public key, and it has fewer OP codes that make up the overall script, due to which it is turned out as an efficient method for data storage compared to the P2FKH; see Figure 10 for a graphical illustration. Nonetheless, the community doesn't use it as a widespread method to store the data. A likely reason for this is since the nodes can easily identify fake (uncompressed) public keys, and the miners could shut down this approach in the future. In this regard, the data can be saved with a fake compressed public key (33 bytes) besides realizing data efficiency compared to P2FKH. Nevertheless, the issue of creating un-spendable UTXOs is also endured by this method.

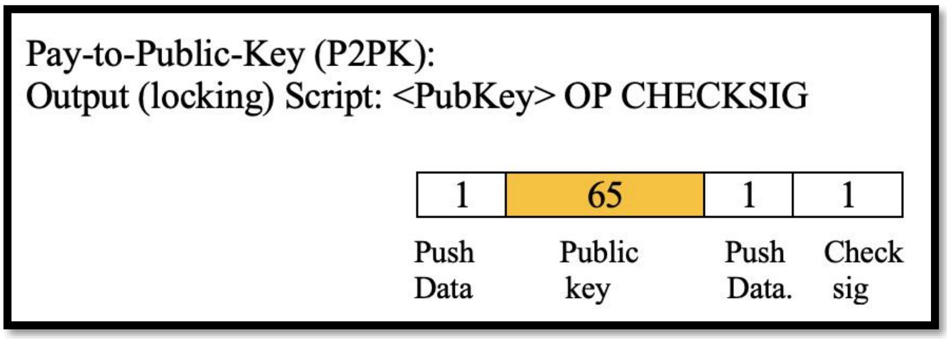

Pay-to-Public-Key (P2PK):
Output (locking) Script: <PubKey> OP CHECKSIG
1
65
1
1
Push Data
Public key
Push Data.
Check sig

Figure 10: Pay-to-Public-Key Transaction

3.1.4. PAY-TO-MULTISIG

A one-of-two or one-of-three multisig script containing a real public key and one or two fake keys having arbitrary data is another method to insert data (Pay-to-Fake-Multisig). It is generally seen in the Blockchain technology Creation of UTXO bloat can be prevented since these transactions are spendable. As far as the lowest overhead cost is concerned, a (real) compressed public key would be used, and the data will be stored via two fake uncompressed public keys (65 bytes each). With this method's help, the UTXO set will maintain the data till the spending of these outputs (using the one real key) by the user. Within a single transaction, one can consistently store multiple P2FMS outputs in all of them using the same public key. Data reconstruction becomes easy to see in Figure 11 for a graphical illustration. Despite that, there should be more than 400 bytes in the transactions that contain a single OP_CHECKMULTISIG. Twenty bytes per sigop is the default requirement, and 20 sigops are considered one instance of OP_CHECKMULTISIG. The redemption of these UTXOs becomes unproductive because of these shortcomings. Additionally, compared to the min non-dust values, the cost for spending these UTXOs will be higher.

Thus, to store arbitrary data with a burn amount, users can use all three pubkey fields with no respect for the UTXO bloat.

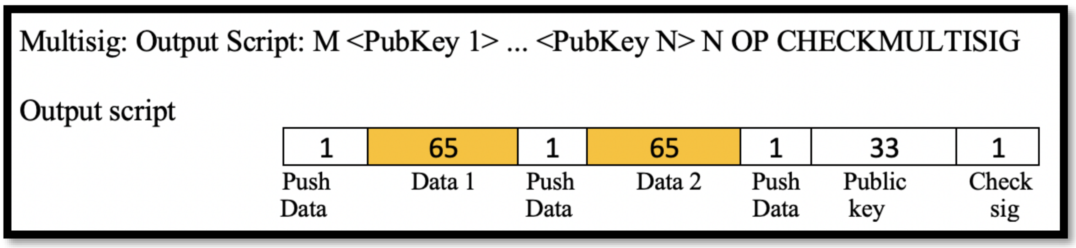


Figure 11: Multisig Transaction

3.1.5. PAY-TO-FAKE-SCRIPT-HASH

Like the P2FKH, data storage as a fake hash is done using the Pay-to-Fake-Script-Hash (P2FSH) method. As compared to P2FKH, two fewer OP codes are required by the P2FSH (making it more effective); however, an un-spendable UTXO is still developed. The method stores data in the input script, where a P2SH output is spent. Creating the UTXO and spending the UTXO are the two stages of the P2SH. A Redeem Script is first developed, and the HASH160 algorithm is subsequently applied to this script to create the P2SH UTXO. Accordingly, the output script is given below:

OP_HASH160 <RedeemScriptHash> OP_EQUAL

For consuming this UTXO, an input script is created containing the Redeem Script (as a solo stack element, hence confined to 520 bytes), led by a series of Script operations. The Redeem Script will conclude in only true upon execution. The data insertion has two approaches: arbitrary data may be stored in the Redeem Script, or/and may be stored in the bit of the input script leading the Redeem Script. E.g., a user can create a Redeem Script with an OP_PUSHDATA2 (three bytes) followed by a 517-bytes data element. Apart from OP_0, any stack element is considered as “true”, the UTXO will successfully be redeemed by this script. Nevertheless, owing to the 520-

byte Redeem Script limit, large amounts of data can be efficiently stored in the input script's segment-leading the Redeem Script (see Figure 12 for a graphical illustration). Since June 2014, experts have been using the variations of the following P2SH-based methods for data storage in the Blockchain.

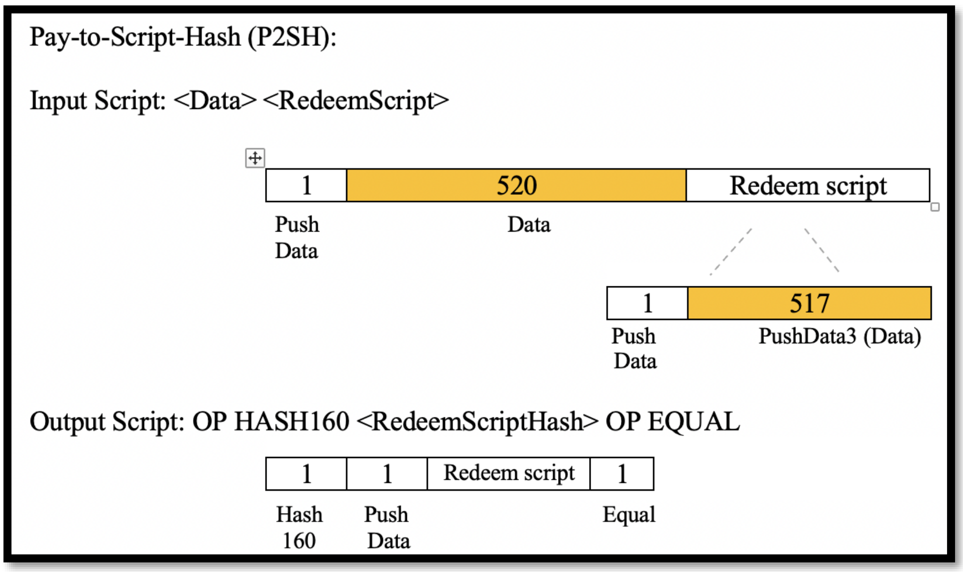

Pay-to-Script-Hash (P2SH):
Input Script: <Data> <RedeemScript>
1
520
Redeem script
Push Data
Data
1
517
Push Data
PushData3 (Data)
Output Script: OP HASH160 <RedeemScriptHash> OP EQUAL
1
1
Redeem script
1
Hash 160
Push Data
Equal

Figure 12: Pay-to-Script-Hash Transaction

3.2. ALTERNATIVE DATA INSERTION METHODS IN ETHEREUM

Ethereum Blockchains offers many data insertion methods. Random data such as files and short messages can be inserted into the ledger. This section describes the data insertion methods for Ethereum blockchains.

The Ethereum Virtual Machine (EVM) is responsible for the runtime environment of smart contracts and can run any operations created by the user using Solidity. An Ethereum account is a 20 bytes string with Ether balance, nonce, contract code, and storage. Additionally, the Ethereum blockchain supports two types of transactions: message call transactions and contract creation transactions. Both types of transactions share the common fields: nonce, gasPrice, gasLimit, to, value, v, r, and s.

Moreover, a contract creation transaction carries a boundless array of bytes, specifying the EVM code for the account initialization process. Conversely, a message call transaction features a byte array data field of an unlimited size that defines the message call's input data.

The indicated solution's objective is to insert non-transaction-related data into the transaction data field or unit, depending on whether the transaction is for a message call or contract creation. We conducted experiments using Ethereum to embed data to Ethereum Blockchain using the following methods.

3.2.1. DATA FIELD

A message call transaction can either be a call to a smart contract or be a simple transfer of ether to a non-contract "externally owned account". Whether the message call is a call to a smart contract or a simple transfer of ether to a non-contract "externally owned account", the data fields are available for the senders to embed information of any size depending on their gasLimit. The data field is mainly concerned with contract creation transactions, such as creating and executing contracts used to specify the called function parameters. However, if the transaction is a simple transfer of ether, the data field is unused and can be used for inserting random data in the ledger. This data can be retrieved in the Ethereum explorer.

3.2.2. CONTRACT CREATION

This type of transaction is a contract creation transaction that involves establishing new accounts with relevant code. A smart contract is published to the Ethereum blockchain when creating the account. The smart contract may be called through a message call transaction by specifying the specific function in the data field. The sender should send a payment for each step of the "program" which they activate. It includes both computation and memory storage costs, and users can then create new contracts by deploying code to the blockchain. An init field is part of the contract creation transaction, and it is a limitless byte containing the EVM code required for the initialization of the account. When an account is created, the init is executed a single time, and then it is instantly discarded. However, the EVM-code will be stored in the state Tree, which gives the user the chance to embed data in the EVM-code using a parameterized constructor.

3.2.3. EVENT CREATION

As we have discussed in chapter two, events are messages stored in the Ethereum blocks transaction log. Event logs are supported by the EVM logging functionality and are part of the transaction receipts Tree (one of the three Merkle Patricia trees stored in Ethereum). This means Merkle proofs can be requested for log information because the receipt root is stored in the block, which gives the user the chance to embed data in the EVM-code using a parameterized constructor.

Since the data and init fields are specified as unlimited size byte arrays, theoretically, one can embed as much information as she wants in Ethereum transactions if she has sufficient ether (more than gasLimit).

3.3. DATA RECONSTRUCTION

3.3.1. LINKING DATA TOGETHER

Our solution aims are to include arbitrary data in both blockchains. However, a limited maximum size can be stored through a transaction on both Bitcoin and Ethereum. Thus, we need to split the desired data into multiple chunks (i.e., transactions), requiring reconstructing the user’s data to see their data again. There are many ways to link the divided data together, affecting rebuilding the original message by putting together these data pieces. In the following paragraphs, we discuss possible methods for creating and storing these data chunks, and we highlight the method we used for our implementation.

One can add the transaction address of the previous transaction to the end of the following transaction that contains the next chunk of data. This will help link the data together in the reconstruction phase easily. However, this method has some disadvantages, like the compressed transaction address (33 bytes), which will be part of the inserted data and added to its cost. Additionally, this method is not beneficial since the user will have to wait for the transaction to be published in the ledger to include its address in the next transaction with the next chunk of data. In Bitcoin, the mentioned way will take about 10 minutes without waiting for confirmation and about 1 hour if we wait for six confirmations between two transactions (chunks) occupying two consecutive blocks. However, In Ethereum, this method will take about 10 seconds between two transactions (chunks) in two consecutive blocks.

Another way to link data together is to indicate unique codes added to the end of each chunk, and the last fragment will have another unique code to mark the end of the data. This method has many advantages, like making the chunks independent of each other, making it faster to publish through the blockchain without waiting for the transaction ID like the first method. Besides, these unique codes are of small bytes so including them within the data is not expensive. However, there is no control over whether the miners will put the transactions within one block or multiple blocks. Hence, it is hard to reconstruct them; one can quickly solve this problem by adding chunks numbers in the first transaction. This will not be complete, so adding to each unique code the number of its data chunk will also help, so it is easy to search for how many fragments they are and reorder them in the reconstruction phase.

To conclude this subsection and by exploring both method’s advantages and disadvantages, we decided to use the second method in both blockchains. Because it leads to faster publication time, and it is excellent that the preceding publishing chunk is independent of the subsequence chunk. More analysis of the chosen method is shown in chapter 5, which covers the results and discussions.

3.3.2. IDENTIFYING PUBLISHED DATA

Bitcoin Insertion data size differs depending on the transaction type, as mentioned in section 3.1. In general, small data can fit in one single transaction using several outputs of the same or different script type. For example, to insert 100 KB data using the P2FKH method, we need to use five output scripts within one single transaction. Thus, looking at how data is inserted in Bitcoin, the suitable way to find the Bitcoin ledger's published data is to exhaustively search the ledger for the inserted data based on the transaction script type. This method was detailed in [75]. Instead of a full search, we can start searching the ledger from the block number we find the transaction by providing the transaction ID. And since the data in Bitcoin can be put in one transaction, it is easy to reconstruct the data going through the outputs of the identified transaction, assuming the data was stored in the ledger in the same order it was broadcasted single transaction. Nevertheless, suppose the data is split through multiple transactions. In that case, we can continue the search for the inserted data within the same block we find the transaction id in or the following block until the unique code of the last data chunk is found.

Furthermore, using the transaction Id in the Ethereum blockchain is also convenient. In section 3.2, we mentioned different insertion methods for using Ethereum, and all of these methods are somehow similar in retrieving the published data. Like the Bitcoin reconstruction method, the first chunk can be retrieved by providing the transaction Id of the first transaction and then going through the transactions in the same block or the following blocks. Therefore, using the unique codes, we can reconstruct the whole data again.

CHAPTER 4: SYSTEM IMPLEMENTATION

This study builds a system that allows the user to publish text data and small-size pictures into the Bitcoin and Ethereum blockchains. The user can choose between the different methods that we discussed in the preceding chapters. Additionally, the user can see the published data from other methods throughout the system and the blockchain ledger. Finally, our system's additional advantage is that it is a standalone application, ensuring that it’s not in any third party’s hands. This assures the data's integrity and eliminates the need for extra service fees to use the system.

We tested out the system on a machine with a 2.8 GHz Intel Core i7 processor running macOS. The device has 16 GB memory and 50 MBps connectivity to the Internet. We used Kovan TestNet for the Ethereum blockchain. Additionally, we used TestNet for the Bitcoin blockchain. To evaluate our system, we ran experiments on different days during August 2020. For implementation and evaluation on the Ethereum platform, we used the Ether.js 4.0.0 API with Webstorm version 2019.3.4. We used Bitcoin- 0.11.3.jar and Eclipse photon 2018 version 4.8.0 to implement and evaluate the Bitcoin network.

4.1. ETHEREUM IMPLEMENTATION

As discussed in the previous sections, an Ethereum blockchain has three methods to insert data into the ledger. For the data field, using Ether.js API, we could insert any data size as long as we had sufficient gas. We inserted data by making message calls to a new address each time. The value field was set to zero, but we specified the gas price based on the network's reasonable current gas price. Even though we ran our experiments on Kovan, our system users can choose which network they want to publish (see Figure 13).

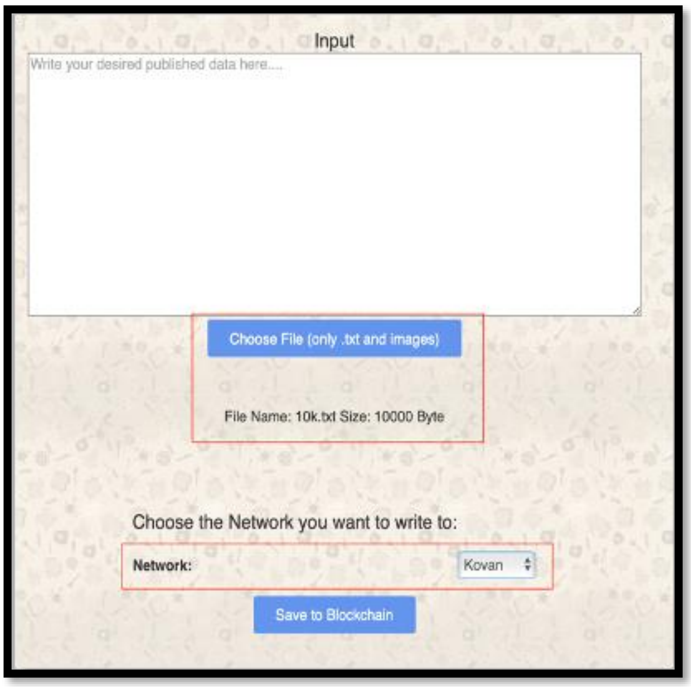
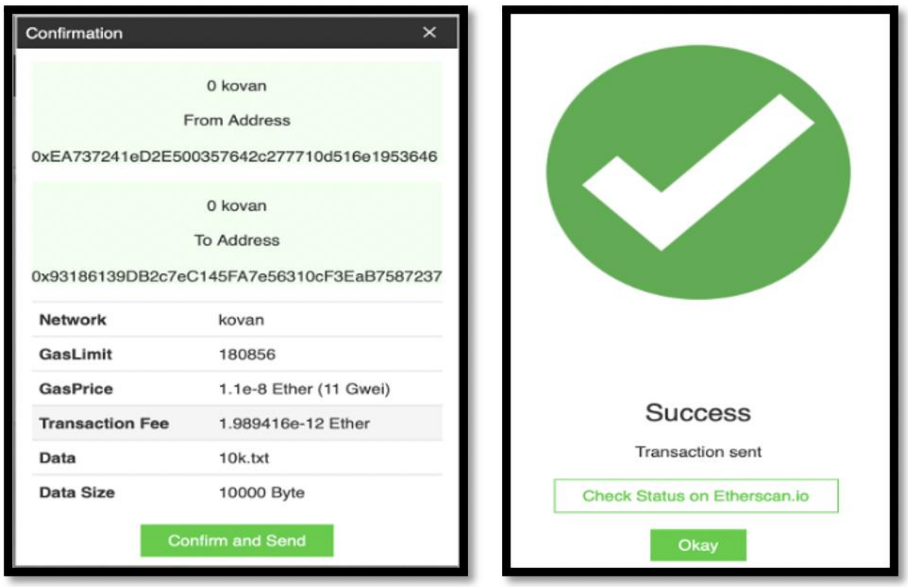

Input
Write your desired published data here....
Choose File (only .txt and images)
File Name: 10k.txt Size: 10000 Byte
Choose the Network you want to write to:
Network:
Kovan
Save to Blockchain

Figure 13: Writing Data Front-End

An estimated transaction fee pops up to the user before sending the transaction for him/her to confirm the transaction, as shown in Figure 14. After confirmation, a link is provided to where the transaction is in the ledger (transaction hash).



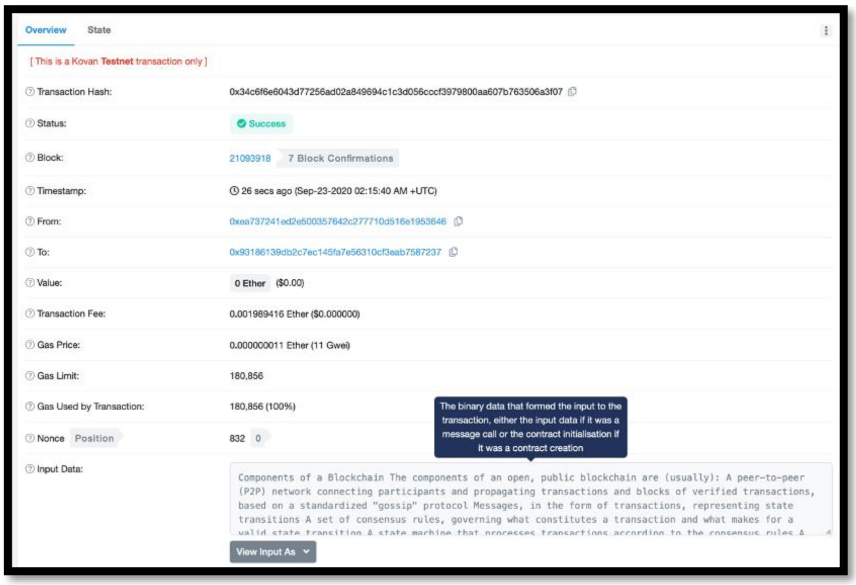
0 kovan
From Address
0xEA737241eD2E500357642c277710d516e1953646

0 kovan
To Address
0x93186139DB2c7eC145FA7e56310cF3EaB7587237

Network	kovan
GasLimit	180856
GasPrice	1.1e-8 Ether (11 Gwei)
Transaction Fee	1.989416e-12 Ether
Data	10k.txt
Data Size	10000 Byte

Figure 14: Confirmation Page and Success Page

The users can explore their previously published data using the provided transaction hash from the Ethereum explorer (see Figure 15) or using our system front-end (see Figure 16). Our front-end explorer helps display a large, published data item by searching the ledger for all the transactions, reconstructing its data, and displaying it in an integrated easy-to-view way.

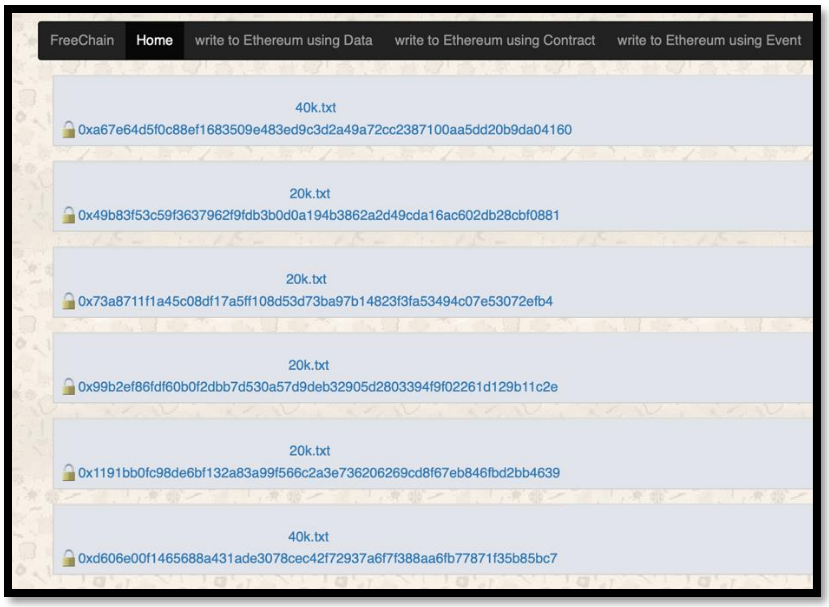


[This is a Kovan **Testnet** transaction only]

Transaction Hash:	0x34c6f6e6043d77256ad02a849694c1c3d056cccf3979800aa607b763506a3f07
Status:	Success
Block:	21093918 7 Block Confirmations
Timestamp:	26 secs ago (Sep-23-2020 02:15:40 AM +UTC)
From:	0xea737241ed2e500357642c277710d516e1953646
To:	0x93186139db2c7ec145fa7e56310cf3eab7587237
Value:	0 Ether ($0.00)
Transaction Fee:	0.001989416 Ether ($0.000000)
Gas Price:	0.000000011 Ether (11 Gwei)
Gas Limit:	180,856
Gas Used by Transaction:	180,856 (100%)
Nonce Position	832 0
Input Data:	Components of a Blockchain The components of an open, public blockchain are (usually): A peer-to-peer (P2P) network connecting participants and propagating transactions and blocks of verified transactions, based on a standardized "gossip" protocol Messages, in the form of transactions, representing state transitions A set of consensus rules, governing what constitutes a transaction and what makes for a valid state transition A state machine that processes transactions according to the consensus rules A

The binary data that formed the input to the transaction, either the input data if it was a message call or the contract initialisation if it was a contract creation

Figure 15: Ethereum Explorer


FreeChain
Home
write to Ethereum using Data
write to Ethereum using Contract
write to Ethereum using Event

40k.txt
0xa67e64d5f0c88ef1683509e483ed9c3d2a49a72cc2387100aa5dd20b9da04160

20k.txt
0x49b83f53c59f3637962f9fdb3b0d0a194b3862a2d49cda16ac602db28cbf0881

20k.txt
0x73a8711f1a45c08df17a5ff108d53d73ba97b14823f3fa53494c07e53072efb4

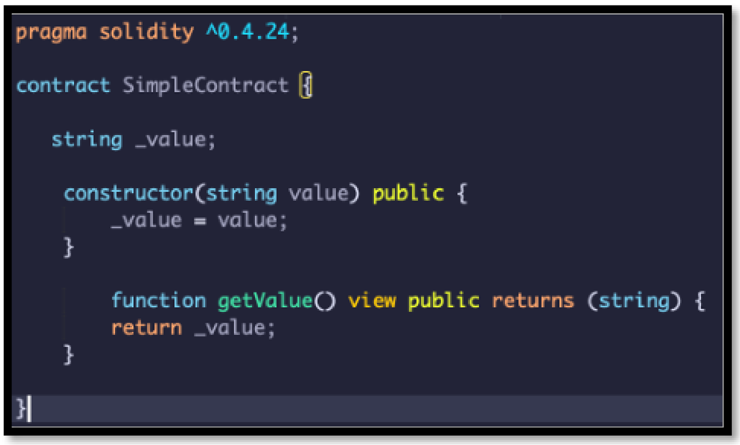
20k.txt
0x99b2ef86fdf60b0f2dbb7d530a57d9deb32905d2803394f9f02261d129b11c2e

20k.txt
0x1191bb0fc98de6bf132a83a99f566c2a3e736206269cd8f67eb846fbd2bb4639

40k.txt
0xd606e00f1465688a431ade3078cec42f72937a6f7f388aa6fb77871f35b85bc7

Figure 16: Front-End Data Explorer

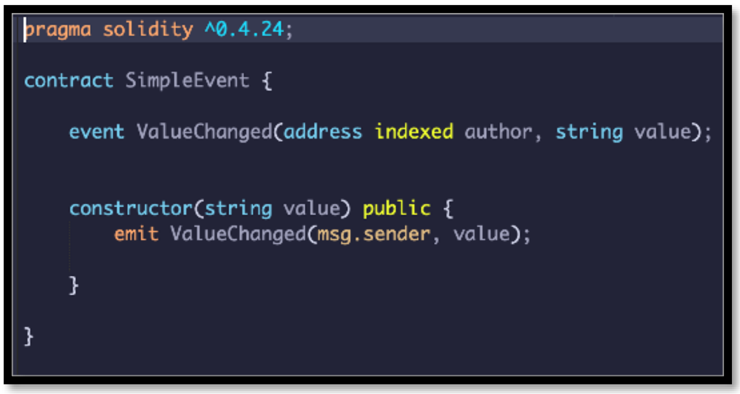
We created a contract with a string field to hold data for inserting data using the Contract Creation method. It is updated using a parameterized constructor using Solidity (see Figure 17). Thus, when we deploy the contract in our system, we include the data in the constructor, which allows data to be included in the contract EVM-code and the state Tree. Similarly, the users have to provide the transaction hash to explore their previously published data in our platform.



```
pragma solidity ^0.4.24;

contract SimpleContract {

   string _value;

    constructor(string value) public {
        _value = value;
    }

        function getValue() view public returns (string) {
        return _value;
    }

}
```

Figure 17: Solidity Code of Contract Creation

The last method we used is event creation. We created an event using Solidity that saves data in an event by emitting it using the parameterized constructor, which kept the contract EVM-code and the transaction receipt Tree (see Figure 18). One can declare an event with a keyword event followed by event name and parameters. *event InsertData(indexed address author, string value)*



```
pragma solidity ^0.4.24;

contract SimpleEvent {

    event ValueChanged(address indexed author, string value);


    constructor(string value) public {
        emit ValueChanged(msg.sender, value);

    }

}
```

Figure 18: Solidity Code of Event Creation

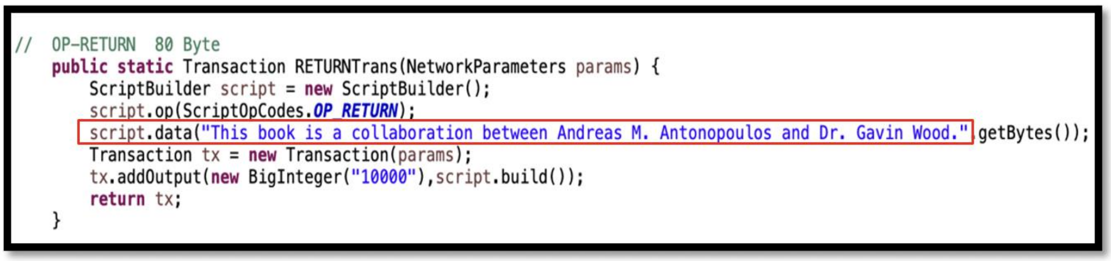
Additionally, one can go back in history and search for these logs later on. Events can inform external users that something happened on the blockchain through returned values. Thus, programs can subscribe and pay attention to those events through the interface of an Ethereum client. However, we create a new event for each message following Satoshi Nakamoto's advice to achieve anonymity in our design. His advice was to use a new IP address for each transaction to prevent others from identifying/linking user's transactions together.

4.2. BITCOIN IMPLEMENTATION

In the Bitcoin Core implementation, there is no size limit on Bitcoin transactions. Thus, the transaction size is essentially limited by the block size. However, the Bitcoin documentation [80] mentions that "transactions that pay a fee of at least 0.00001 BTC/kb are added to the block, highest-fee-per-kilobyte transactions first, until the block is not more than 750,000 bytes big". In the Bitcoin Core source code, the maximum allowed script size is 10,000 bytes. However, there is no limit on the number and size of output per transaction. Theoretically, one can put as much as 750KB of data in one transaction using one or more outputs.

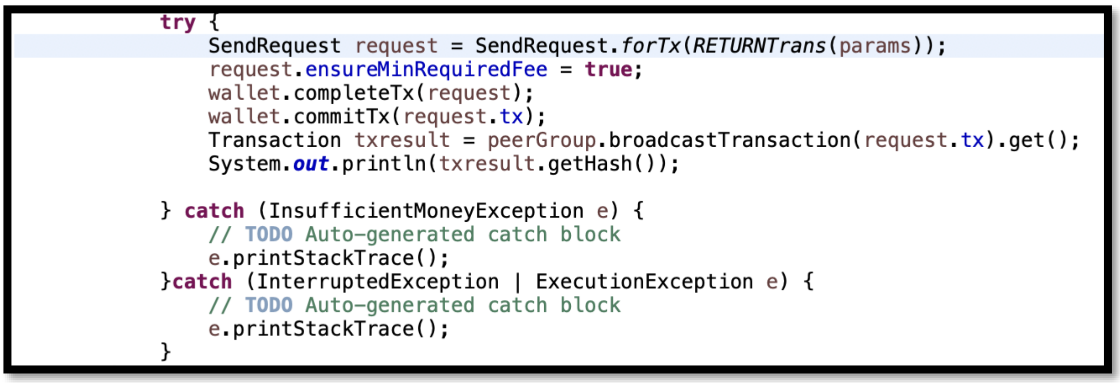
We ran our experiments using the Bitcoin.jar API. As we have mentioned in the preceding sections, Bitcoin has five different methods to insert data in the ledger, as demonstrated in Chapter 3. One can use multiple methods within one script to restrict that each method cannot exceed the data permitted limit. For the chosen data sizes, when we want to use the P2FK method, we have to divide the data into 65 bytes chucks and use multiple P2FK in one transaction using various outputs. This is done to all other transaction methods except OP RETURN since one can only use one OP RETURN per transaction.

We published the data using Bitcoin TestNet. Due to storage limitations, we used the API to connect and download the Bitcoin ledger each time we run the code. After downloading the blockchain, we added a listener to the wallet for any new transaction to display the transaction results. For creating the transactions, we used script builder, which allowed us to build the scripts ourselves. An example of the script code for the OP RETURN transaction is shown in Figure 19. After creating the transactions, a request will be made, and the transaction is added to it. The request is added to the wallet to complete the transaction, then after committing the transactions, it is broadcasted through the network's peers, as illustrated in Figure 20.



```
//  OP-RETURN  80 Byte
    public static Transaction RETURNTrans(NetworkParameters params) {
        ScriptBuilder script = new ScriptBuilder();
        script.op(ScriptOpCodes.OP_RETURN);
        script.data("This book is a collaboration between Andreas M. Antonopoulos and Dr. Gavin Wood.".getBytes());
        Transaction tx = new Transaction(params);
        tx.addOutput(new BigInteger("10000"),script.build());
        return tx;
    }
```

Figure 19: OP RETURN Transaction Code

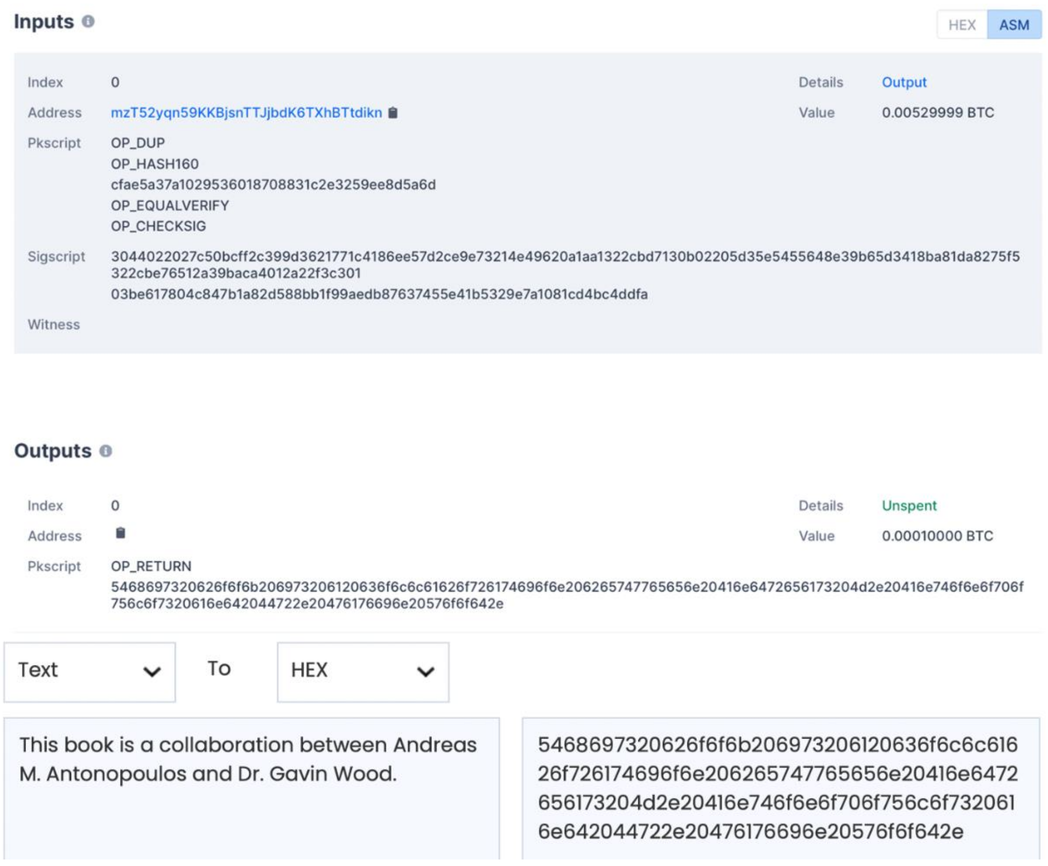


```
        try {
            SendRequest request = SendRequest.forTx(RETURNTrans(params));
            request.ensureMinRequiredFee = true;
            wallet.completeTx(request);
            wallet.commitTx(request.tx);
            Transaction txresult = peerGroup.broadcastTransaction(request.tx).get();
            System.out.println(txresult.getHash());

        } catch (InsufficientMoneyException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }catch (InterruptedException | ExecutionException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
```

Figure 20: Publishing the Transaction

Since the Bitcoin implementation was only used for evaluating the approach, we did not build a GUI for publishing and displaying the data. As for seeing the transactions data, we use the transaction Id and the Bitcoin Explorer to see the published data.

Figure 21 illustrates an OP RETURN transaction that was published on the 4th of February 2021.



```
Index      0                                          Details   Output
Address    mzT52yqn59KKBjsnTTJjbdK6TXhBTtdikn         Value     0.00529999 BTC
Pkscript   OP_DUP
           OP_HASH160
           cfae5a37a1029536018708831c2e3259ee8d5a6d
           OP_EQUALVERIFY
           OP_CHECKSIG
Sigscript  3044022027c50bcff2c399d3621771c4186ee57d2ce9e73214e49620a1aa1322cbd7130b02205d35e5455648e39b65d3418ba81da8275f5
           322cbe76512a39baca4012a22f3c301
           03be617804c847b1a82d588bb1f99aedb87637455e41b5329e7a1081cd4bc4ddfa
Witness
```

```
Index      0                                          Details   Unspent
Address                                               Value     0.00010000 BTC
Pkscript   OP_RETURN
           5468697320626f6f6b206973206120636f6c6c61626f726174696f6e206265747765656e20416e6472656173204d2e20416e746f6e6f706f
           756c6f7320616e642044722e20476176696e20576f6f642e
```

This book is a collaboration between Andreas M. Antonopoulos and Dr. Gavin Wood.

```
5468697320626f6f6b206973206120636f6c6c616
26f726174696f6e206265747765656e20416e6472
656173204d2e20416e746f6e6f706f756c6f732061
6e642044722e20476176696e20576f6f642e
```

Figure 21: OP RETURN Transaction on Blockchain Explorer

CHAPTER 5: RESULTS AND DISCUSSION

An evaluation has been conducted to compare Bitcoin and Ethereum blockchains publishing cost and speed to evaluate which blockchain method is better used in publishing arbitrary data. The price depends on the transaction fees multiplied by the number of transactions needed to be published in the ledger. The block adding rate is almost constant. Thus, the speed is determined by the transaction numbers required to insert data into the ledger and the blocks that will include these transactions. For example, if we have a message that we need to publish, it requires **n** transactions with no control of how they will spread over blocks, requiring **m** blocks.

$$Speed \ = \frac{1}{No.Transactions\ (n) * No.Block\ (m)} \qquad (1)$$

Because a block is added to the Bitcoin blockchain around every 10 minutes [82] and about every 13 seconds in Ethereum [83], the time needed to publish a specific size message will be analyzed. Finally, the ease of browsing is also evaluated. This is affected by the way we fragment the data and recollect it again to present it in a graphical user interface that users can use to publish and browse their data.

5.1. ETHEREUM EVALUATION

The block gas limit in Ethereum is about 10M gas, leading to typical block sizes of 20-30KB. Therefore, we tried 1KB, 10KB, 20KB, 30KB, and 40KB data sizes to be inserted into the ledger. We experimented using the platform we built to measure the cost, speed, and ease of browsing. We ran the experiment multiple times using the data sizes mentioned earlier, and each data point of the experiment is an average of 5 readings. In each run, we set the gas price differently, which does not affect the gas used. We calculated the transaction fee for each run, which varies as shown in Equation 2.

$$transaction\ fee \ = \ gas\ price \ * \ gas\ used \qquad (2)$$

Figure 22 illustrates the transaction fee of publishing data of varying sizes using the different methods at a gas price of 0.000000011 Ether (11 Gwei).

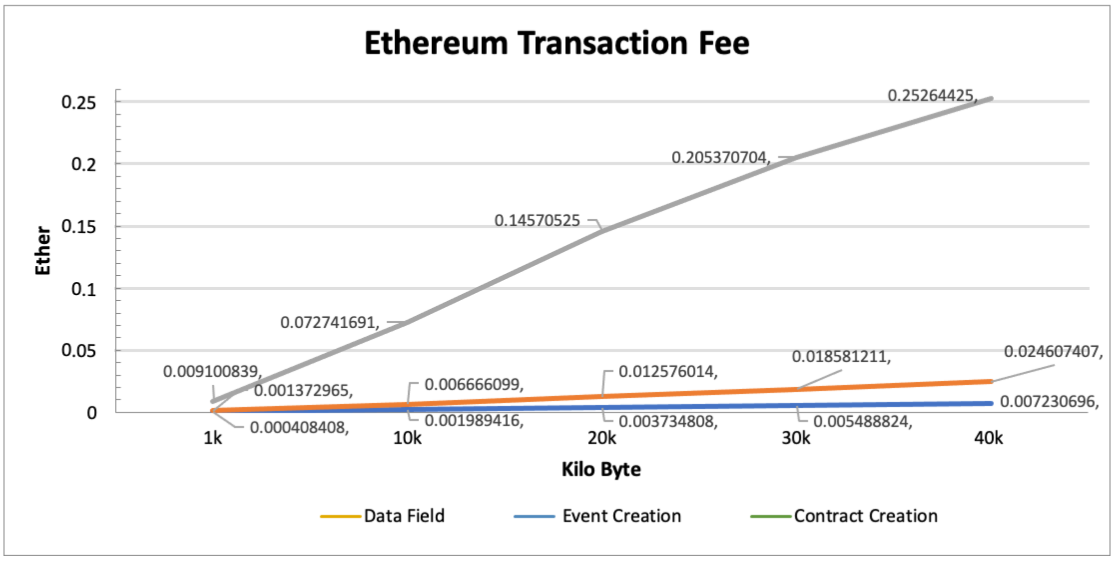


Figure 22: Ethereum Transaction Fee in Ether

We used the Ethereum market price on the 25th of September 2020 to calculate the USD transaction fee when we conducted our experiment. However, over a couple of months, the Ethereum market price changed dramatically. Due to that, we conducted another experiment to show the differences. Table 1 and Table 2 illustrate the fluctuation of the Ethereum publishing prices of different data sizes at various times over the past and current year.

Table 1: Sep 25, 2020, Ethereum Transaction Fee in USD

Data Size/Method	1KB	10KB	20KB	30KB	40KB
Data Field	0.14	0.70	1.32	1.93	2.55
Event Creation	0.48	2.35	4.43	6.55	8.67
Contract Creation	3.21	25.64	51.37	72.40	89.06

Table 2: April 19, 2020, Ethereum Transaction Fee in USD

Data Size/Method	1KB	10KB	20KB	30KB	40KB
Data Field	0.9	4.22	7.91	11.63	15.32
Event Creation	2.9	14.12	26.65	39.37	52.14
Contract Creation	19.3	154.13	308.72	435.14	535.31

We found that the data field approach is the least expensive method to insert data into the ledger, as shown in the cost figures reported in Table 1 versus the statistics shown in Table 2. Additionally, the Event Creation method's cost is higher than the data field method since the data is stored in the receipt trees as logs and in the init data field and the EVM-code in the transaction trees. The cost of the event creation method is illustrated in Table 1 and Table 2. The contract creation method is the most expensive one, as displayed in Table 1 and Table 2. This is because the data is stored in the state and transaction trees, which is significantly costlier than storing data in other trees. Therefore, if a user wants to insert data into the Ethereum blockchain, it is recommended to use the data field method.

We also evaluated the ease of browsing. As mentioned earlier, ease of browsing is affected by the way we fragment the data and recollect it again. We could insert data in the ledger in one transaction for 1KB, 10KB, and 20KB during the experiment. However, we had to split data into two transactions for 30KB data size and three transactions for 40KB data size for all methods. The number of fragments was small, which made reconstruct the data faster to display to the users.

Another factor taken into consideration to evaluate the Ethereum blockchain is the speed of publishing data. A new block is added to the Ethereum blockchain every

13 seconds. According to the Ethereum yellow paper [56], seven confirmation is needed to confirm a transaction, which takes about two minutes. Equation 1 describes the parameters the speed depends on the number of blocks and the number of transactions. As mentioned in the previous paragraph, the data was only split for 30KB, 40KB, and the rest of the data sizes were sent by one transaction. In our experiment, we found that the publishing time needed to differ due to the data size. Table 3 illustrates how the publishing time is increasing with the increase of the data sizes. The Event creation method is considered the fastest method compared to the other methods.

Table 3: Ethereum Data Insertion Time in Seconds

Data Size/Method	1KB	10KB	20KB	30KB	40KB
Data Field	11.46	16.23	19.86	23.50	32.50
Event Creation	10.23	12.56	14.73	19.52	20.00
Contract Creation	11.09	16.53	20.39	25.50	29.90

5.2. BITCOIN EVALUATION

Bitcoin hard forks have split Bitcoin into the following cryptocurrencies: Bitcoin Core (BTC, the original version by Satoshi Nakamoto), Bitcoin Cash (BCH), Bitcoin Private (BTCP), and Bitcoin Gold (BTG). Currently, Bitcoin Core has a block size limit of 1MB. In our thesis, we only consider Bitcoin Core. We considered similar data sizes of 1KB, 10KB, 20KB, 30KB, and 40KB to insert into the Bitcoin ledger to compare both blockchains. We carried an experiment using Bitcoin-0.11.3.jar using the Eclipse program to measure the cost, speed, and ease of browsing. We run the experiment multiple times using the data sizes mentioned earlier, and each data point of the experiment is an average of 5 readings. In each run, we calculated the transaction fee for each run. Figure 23 illustrates the transaction fees in Satoshi of each method used to insert the data. We used the Bitcoin market price on the 25th of September 2020 to calculate the USD transaction fee. However, as we mentioned in the previous section, the market price has drastically changed over a short time. Therefore, Table 4 and Table 5 illustrate the prices during different times over the previous and current year.

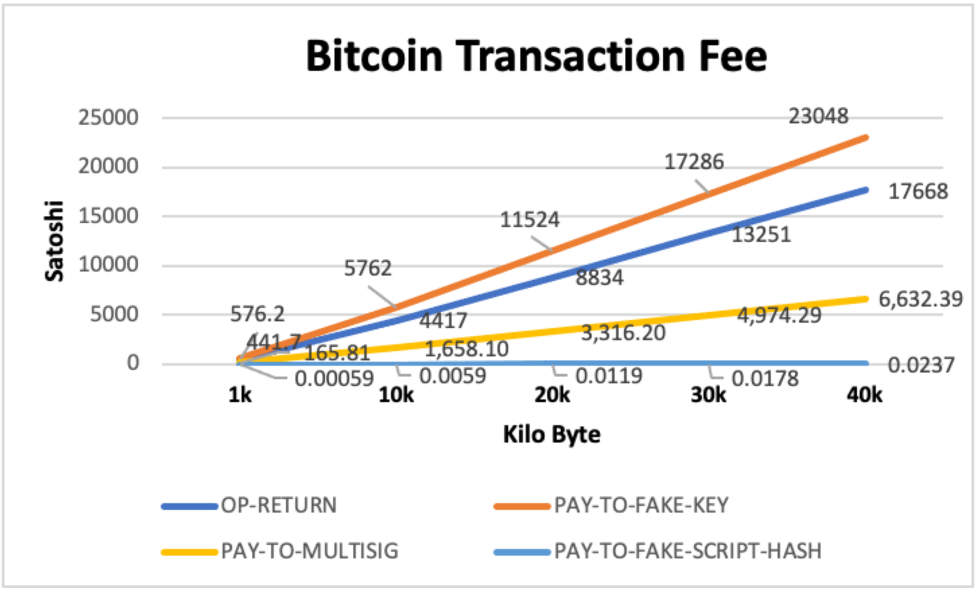


Figure 23: Bitcoin Transaction Fee in Satoshi (1 Satoshi = 0.00000001 BTC)

Table 4: Sep 25, 2020, Bitcoin Transaction Fee in USD

Data Size/Method	1KB	10KB	20KB	30KB	40KB
OP RETURN	134.75	1347.5	2695	4042.5	5390
P2FK	92.15	921.54	1843.08	2764.62	3686.15
P2FKH	3234	32340	64680	97020	129360
Multisig.	66.13	661.35	1,322.70	1,984.05	2,645.40
P2FSH	6.09	60.89	121.78	182.68	243.57

Table 5: April 19, 2021, Bitcoin Transaction Fee in USD

Data Size/Method	1KB	10KB	20KB	30KB	40KB
OP RETURN	643	6,430	12,860	19,290	25,720
P2FK	791.38	79,13.85	15,827.69	23,741.54	31,655.38
Data Size/Method	1KB	10KB	20KB	30KB	40KB
P2FKH	14,288.50	142,885	285,770	428,655	571,540
Multisig.	315.58	3,155.83	6,311.66	9,467.48	12,623.31
P2FSH	36.61	366.10	732.20	1,098.31	1,464.41

As we can see, comparing the transaction fee, the most expensive method is P2FKH which is not shown in Figure 23 due to the vast difference in costs with a transaction fee of 2222.2 Satoshi to insert 1KB of data. However, the tables show how much costs are needed to insert different data sizes using it. OP RETURN and P2FK are also considered expensive. P2FSH and P2FMS are the least costly methods. Thus, if the user wants to insert data into the Bitcoin ledger, they recommend using the PAY-TO-FAKE-SCRIPT-HASH method.

Another aspect of evaluating the methods is the utilization of fake keys to insert data. We want to analyze which of the methods have the minimum effect on the UTXO set. Using fake keys will make transactions unspendable UTXO which will lead to bloating the UTXO set. UTXO set is not only used to verify blocks. It is also continuously verifying new transactions. A vast UTXO set will require allocating more hardware with lower power, making retrieving data expensive and decreasing the number of full nodes in the network. Looking at the proposed methods, P2FKH and P2FSH are considered disadvantageous since they bloat the ledger with unspendable

UTXO just to insert 20 bytes of data. That other unfavorable method is P2FK that provides only 65 bytes per one unspendable UTXO.

On the other hand, P2FMS inserts 195 bytes using all three fake keys, resulting in unspendable UTXO. Suppose one used two fake keys to insert 130 bytes and one real key, making the UTXO spendable. Lastly, OP PRTURN provides 80 bytes with unspendable UTXO without bloating the set since the users can discard these UTXO from their sets.

The speed of publishing data is measured to evaluate the Bitcoin blockchain methods. A new block is added to the Bitcoin blockchain every 10 minutes. According to [84], most users wait for six confirmations to confirm a transaction that takes about 60 minutes—Bitcoin blockchain unconfirmed transactions set in the mining pool. To speed up the confirmation, one needs to pay a high transaction fee, so miners are encouraged to include the transaction in the next block. Equation 1 describes the parameters the speed depends on the number of blocks and the number of transactions. To insert data in the Bitcoin ledger, we need to split the data into multiple outputs and try to fit as many as possible. In our experiment, we found that the time required differ due to the data size. Table 6 illustrates how the time is increasing with the increase of the data sizes. As shown in Table 6, the P2FSH method is considered the fastest than the other methods.

Table 6: Bitcoin Time Insertion of Data in Seconds

Data Size/Method	1KB	10KB	20KB	30KB	40KB
OP RETURN	26	260	2600	26000	260000
P2FK	30	35	45	55	66
P2FKH	40	54	64	76	87
Multisig.	30	35	45	55	66
P2FSH	15	23	36	47	60

As for ease of browsing, we mentioned in section 3.3 that the data that exceed the block size limit has to be split. Thus, we need to find a way to reconstruct the data in a fast way. Therefore, instead of relying on the transaction id of the previous data chunk to augment data, we added a unique code to the end of each chunk which made the reconstructing phase faster. Another aspect to look at is the number of transactions needed to insert the data. The Bitcoin ledger's data was not fragmented, making reconstructing the data faster to display to the users. We were able to fit all the data size within one of the multiple outputs for all methods except for the OP RETURN method due to the Bitcoin protocol allows for a transaction to only hold one OP RETURN.

On the other hand, in the Ethereum ledger, the 30 KB and 40 KB data sizes were split into chunks and used 2 and 3 transactions, respectively, in both the Contract Creation and the Event Creation methods. Thus, the reconstructing phase was slower than the Data Field method by seconds. Overall, Bitcoin method P2FSH is the fastest in reconstructing the data, and the Data Field method in Ethereum is considered the fastest. Table 7 illustrates the time needed to reconstruct the data in both Bitcoin and Ethereum blockchains.

Table 7: Bitcoin and Ethereum Reconstructing Time in Seconds

Data Size/Method	1KB	10KB	20KB	30KB	40KB
OP RETURN	28	280	2800	28000	280000
P2FK	32	40	54	66	87
P2FKH	43	65	75	86	97
Multisig.	32	40	54	66	87
P2FSH	20	32	46	57	71
Data Field	15	25	38	54	63
Event Creation	30	43	55	67	83
Contract Creation	30	43	55	67	83

To conclude, after looking at both Bitcoin and Ethereum blockchains results, we can conclude that the methods provided by the Ethereum blockchain are much more beneficial to use to build the future social media platform for allowing freedom of speech. Users of this social media platform are willing to pay money for it and tolerate the delay due to the blockchain transactions nature if they need their data to be permanently stored in the ledger.

CHAPTER 6: CONCLUSION, CHALLENGES, AND FUTURE WORK

6.1. CONCLUSION

Recently, social media proved to have a vital role in spreading and revealing the truths in many real-life scenarios. However, many countries deploy powerful mechanisms to block content from the external world and impose censorship on Internet media for inappropriate political reasons.

For proof-of-work-based blockchains, the content in the blocks is non-removable and non-modifiable. Furthermore, these permissionless networks offer participants pseudonymity and, therefore, provide some level of identity protection. Thus, blockchains may give a solution to facilitate freedom of speech. On the other hand, criminals may use blockchains to distribute illegal contents (e.g., child pornography) or establish black markets. It is worth noting that Bitcoins did not receive sufficient attention until it was used as the payment method on Silkroad for illegal contents. Though a blockchain has limited bandwidth for content distribution, it is enough to carry out these illegal activities. When blockchains are used for publishing content, there is a chance that authoritarian regimes will try to obstruct such content. However, massive stake and investment have been put in these blockchains, making it much harder for them to be blocked. In this thesis, we have proposed and successfully demonstrated a solution that can facilitate the ability to achieve freedom of speech by using Ethereum and Bitcoin blockchains. This is mainly based on blockchain's immutable ledger nature that prevents any changes in the data. A straightforward application of the proposed methodology would be building a genuinely free social media platform that a central authority cannot manipulate. This thesis proposed different methods to achieve this goal using the Bitcoin and Ethereum blockchains and presented these other possible approaches.

The thesis investigated several methods for publishing on-chain arbitrary content using the Bitcoin and Ethereum networks. To summarize the results, the Bitcoin ledger OP RETURN method only provides 80 Bytes, which is suitable for small data sizes and is not practical to be used for extensive data. If one uses it for large data items, the drawback will be how much time is needed to publish and reconstruct it. The most efficient Bitcoin method to insert large data is the Pay-To-Fake-Script-Hash method, as demonstrated in our analysis. This method is the cheapest and fastest method among the other Bitcoin methods. As for Ethereum results, the Data Field method is the least expensive, although the Event Creation method is the fastest method to insert the data into the ledger. As for ease of browsing, both blockchain methods allowed the data to be inserted within one or two transactions, reducing the overhead of dividing and reconstructing the data.

6.2.CHALLENGES

Although blockchain technology has the potential to be used as a tool to achieve "freedom of speech" as demonstrated in this thesis, there are some challenges that are worth pointing out:

6.2.1. ANONYMITY*:* Recently, the state-of-art became active regarding the deanonymization of blockchain users. Moreover, there are some reported incidents of identifying some users in the blockchain. Thus, mixers can be used to maximize the users' privacy in our platform. However, this will increase publishing data costs due to paying mixers fees and transaction fees. However, some researchers are also trying to find de-mixing techniques [85] for the mixing services, which raises the challenge of relying on other methods that anonymize users.

6.2.2. RISKS OF ARBITRARY BLOCKCHAIN CONTENT: Since the proposed solution introduces the idea of inserting arbitrary data into the blockchain, there is no guarantee that users will not misuse these channels for inappropriate content insertion. One solution might be by creating a blacklist of words that are agreed and shared in the blockchain community to prevent misbehaving people from publishing inappropriate content. Additionally, we can look at the cost as a key feature in our solution to deter inserting improper data since it is not a free platform as the other available social media platforms. Thus, the cost of publishing data might work to our advantage. However, the cost may not be an issue for some users who are motivated to publish inappropriate content like child pornography. In [86], the authors proposed an algorithm to detect transactions containing illicit content.

6.2.3. SECURITY: Other risks were discussed in detail in [87], where the risks of inserting malware will harm the network users. However, this risk is not what might the future platform face since the platform only allows the insertion of binary data, not malicious smart contracts. However, these binary data may correspond to a malicious executable code that already exists on the blockchain. There are a lot of researchers who are trying to find a solution to such risk. Bitcoin provides template scripts for the nodes to discard non-compliant scripts, which will prevent malicious scripts from being executed.

6.2.4. EFFICIENCY: When storing user's sensitive data on the blockchain, the data will be duplicated on all nodes. Although this is the nature of blockchain, and that's why the data is immutable, it is considered a drawback for replicating these data and is consuming the node’s storage. However, the replications are what allowed users to achieve their free speech. Nevertheless, our proposal is not intended to publish all data (e.g., all postings in a social networking scenario). It is only designed to publish critical and sensitive data, which is, in practice, expected to be very limited and will only constitute a tiny fraction of the entire data set. Therefore, replicating such a small critical fraction of the data may not be a serious limitation.

6.2.5. USABILITY: Users who rely on blockchain to store their data permanently have to understand the nature of blockchain transactions in terms of cost and delay. They have to sacrifice by paying miners’ transaction fees which fluctuate based on the market price of used blockchain. In addition, they need to tolerate the delay of their data to be confirmed. As we illustrate in the results and discussion chapter, users will see their transactions within 30 seconds, and the time increases when the data increases. The actual delay that they have to tolerate confirms the transactions since each confirmation needs 10 minutes in Bitcoin and 13 seconds in Ethereum. In practice, it is customary to wait for six confirmations in Bitcoin and seven confirmations in Ethereum.

6.2.6. BLOCKING*:* There is a debate that although we can use Bitcoin and Ethereum blockchains to achieve our goal, blockchain transactions can be blocked using the censorship techniques we mentioned in section 2.1.1, for example, packet filtering. Our proposed solution relies on Bitcoin, and Ethereum blockchain, in which people have put an enormous investment. Blocking them is not that simple and easy. Using techniques such as packet filtering to block access to the Bitcoin and Ethereum networks will have vast economic consequences since a huge stake has already been put in these platforms. Thus, blocking these blockchains is much more difficult than blocking a webpage.

6.3.FUTURE WORK

Our thesis used smart contracts and wallets to test the ability to insert data and measure the time and costs needed. In certain scenarios, we are convinced that users of a future social media platform would be willing to tolerate the delay and the cost for their data to be permanently stored on the blockchain. Therefore, a social media platform can potentially be built to take advantage of these methods. Such a platform would use the blockchain only to publish sensitive and critical data. Other data will be stored using conventional methods. The users who may be interested in such a platform may be journalists and human rights activists. Thus, our future work would be building a standalone social media platform on top of Ethereum and Bitcoin ledgers to express their opinions without any censorship freely. However, there is also a need to address the challenges that were mentioned in the previous section. Each challenge is by itself a research problem that could be tackled in the future

REFERENCES

[1] U. G. Assembly, "Universal declaration of human rights," *UN General Assembly,* vol. 302, no. 2, 1948.

[2] "Universal Declaration of Human Rights," United Nations, [Online]. Available: https://www.un.org/en/universal-declaration-human-rights/. [Accessed 3 August 2020].

[3] R. Hanifatunnisa and B. Rahardjo, "Blockchain based e-voting recording system design," in *2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)*, 2017.

[4] G. Matthew and A. Kohirkar, Social media analytics: Techniques and insights for extracting business value out of social media., IBM Press, 2015.

[5] "freedom in the world 2019," 2019. [Online]. Available: https://freedomhouse.org/report/freedom-world/freedom-world-2019. [Accessed 23 February 2019].

[6] R. FANTINA, "Social Media and Israel: Censorship of the Truth," counterpunch, 6 July 2018. [Online]. Available: https://www.counterpunch.org/2018/07/06/social-media-and-israel-censorship-of-the-truth/. [Accessed 3 May 2019].

[7] B. Fung, "Twitter bans President Trump permanently," CNN, 9 January 2021. [Online]. Available: https://edition.cnn.com/2021/01/08/tech/trump-twitter-ban/index.html. [Accessed 23 January 2021].

[8] P. Nakov and G. Da San Martino, "Fact-Checking, Fake News, Propaganda, and Media Bias: Truth Seeking in the Post-Truth Era," in *Proceedings of the*

2020 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts, 2020.

[9] AP, "US Election 2020: Did social media actually counter misinformation?," The Indian Express, 5 November 2020. [Online]. Available: https://indianexpress.com/article/explained/us-elections-2020-social-media-misinformation-donald-trump-6950915/. [Accessed 28 January 2021].

[10] N. Spring and G. F. Riley, "Passive and Active Measurement.," *LNCS*, vol. 6579, 2011.

[11] R. Terman, "Internet Censorship (Part 2): The Technology of Information Control," Townsend Center for the Humanities, [Online]. Available: https://townsendcenter.berkeley.edu/blog/internet-censorship-part-2-technology-information-control. [Accessed 3 Feb 2021].

[12] "What Is DNS? | How DNS Works," Cloudflare, [Online]. Available: https://www.cloudflare.com/learning/dns/what-is-dns/. [Accessed 6 Jan 2021].

[13] A. Zarras, "Leveraging Internet services to evade censorship," *International Conference on Information Security, Springer*, 2016.

[14] "Understanding and Circumventing Network Censorship," SURVEILLANCE SELF-DEFENSE, 25 April 2020. [Online]. Available: https://ssd.eff.org/en/module/understanding-and-circumventing-network-censorship. [Accessed 2 Feb 2021].

[15] H. Kalodner, M. Carlsten, P. Ellenbogen, J. Bonneau and A. Narayanan, "An empirical study of Namecoin and lessons for decentralized namespace design," in *The Workshop on the Economics of Information Security (WEIS)*, Netherlands, 2015.

[16] "BitDNS and Generalizing Bitcoin," Satoshi Nakamoto Institute, 15 November 2010. [Online]. Available: https://satoshi.nakamotoinstitute.org/posts/bitcointalk/threads/244/. [Accessed 20 February 2021].

[17] M. Mari, "How Facebook's Tor service could encourage a more open web," Wayback Machine. The Guardian, 10 June 2016 . [Online]. Available: https://www.theguardian.com/technology/2014/dec/05/how-faceboook-tor-service-encourage-open-web. [Accessed 20 February 2021].

[18] D. Rafte, "Proxy vs. VPN: 4 differences you should know," NortonLifeLock, [Online]. Available: https://us.norton.com/internetsecurity-privacy-proxy-vs-vpn.html. [Accessed 20 February 2021].

[19] C. HOFFMAN, "How to Use SSH Tunneling to Access Restricted Servers and Browse Securely," How To Geek, 12 July 2017. [Online]. Available: https://www.howtogeek.com/168145/how-to-use-ssh-tunneling/. [Accessed 20 February 2021].

[20] "Beset by online surveillance and content filtering, netizens fight on," REPORTS, 25 January 2016. [Online]. Available: https://rsf.org/en/reports/beset-online-surveillance-and-content-filtering-netizens-fight. [Accessed 3 Jun 2020].

[21] "How governments handle the news," The Economist, 9 Feb 2008. [Online]. Available: https://www.economist.com/middle-east-and-africa/2008/02/07/how-governments-handle-the-news. [Accessed 3 Jun 2020].

[22] E. Muzzy, "How the Ethereum Blockchain Became a Tool in the Fight for China’s #MeToo Movement," 16 May 2018. [Online]. Available:

https://medium.com/@everett.muzzy/how-the-ethereum-blockchain-became-a-tool-in-the-fight-for-chinas-metoo-movement-e4017b1acddd. [Accessed 21 Feb 2020].

[23] K. A. Ruane, Freedom of Speech and Press: Exceptions to the First Amendment, Washington: DC: Congressional Research Service, 2014.

[24] M. E. Erendor and G. Tamer, "The New Face of the War: Cyber Warfare," *Cyberpolitik Journal,* vol. 2, no. 3, pp. 53-70, 2017.

[25] B. Liang and H. Lu, " Internet development, censorship, and cyber crimes in China," *Journal of Contemporary Criminal Justice,* vol. 26, no. 1, pp. 103-120, 2010.

[26] J. L. Qiu, "Virtual Censorship in China: Keeping the ate between the cyberspaces," *International Journal of Communications Law and Policy,* vol. 4, no. 1, p. 25, 1999.

[27] S. S. Gupta, Blockchain, John Wiley & Sons, Inc., 2017.

[28] H.-J. Yoon, "Blockchain technology and healthcare," *Healthcare informatics research,* vol. 25, no. 2, 2019.

[29] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009. [Online]. Available: https://bitcoin. org/bitcoin.

[30] "Distributed VS Decentralized Blockchain Systems," 1 December 2020. [Online]. Available: https://zipmex.com/learn/distributed-vs-decentralized/. [Accessed 20 February 2021].

[31] "What is Decentralization? Basics Explained," Lisk Academy, [Online]. Available: Available: https://lisk.io/academy/blockchain-basics/benefits-of-blockchain/what-is-decentralization. [Accessed 13 Jan 2020].

[32] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys & Tutorials,* vol. 18, no. 3, pp. 2084-2123, 2016.

[33] "Pulling the Blockchain apart.. The Transaction Request.," [Online]. Available: https://medium.com/ignation/pulling-the-blockchain- apart-the-transaction-request-8d6c2b48090f. [Accessed 13 Jan 2020].

[34] "Consensus Protocols," Lisk, [Online]. Available: https://lisk.io/academy/blockchain-basics/how-does-blockchain-work/consensus-protocols. [Accessed 13 Jan 2020].

[35] M. E. Peck, "Blockchains: How they work and why they'll change the world," *IEEE spectrum,* vol. 54, no. 10, pp. 26-35, 2107.

[36] A. Extance, "The future of cryptocurrencies:Bitcoin and beyond," *Nature News,* vol. 526, no. 7571, p. 21, 2015.

[37] V. Buterin, "Ethereum white paper," in *GitHub repository 1*, 2013.

[38] L. Ross, "How to Stake Ethereum," benzinga, 16 March 2021. [Online]. Available: https://www.benzinga.com/money/how-to-stake-ethereum/. [Accessed 19 March 2021].

[39] "PROOF-OF-STAKE (POS)," Ethereum, 23 December 2020 . [Online]. Available: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/. [Accessed 20 February 2021].

[40] "What is a Blockchain?," Peercoin University, [Online]. Available: https://university.peercoin.net. [Accessed 13 Jan 2020].

[41] R. G. IV, "Vulnerability: Proof of Work vs. Proof of Stake," Robert Greenfield IV, 24 Aug 2017. [Online]. Available:

https://robertgreenfieldiv.medium.com/vulnerability-proof-of-work-vs-proof-of-stake-f0c44807d18c. [Accessed 20 February 2021].

[42] "Stake your ETH to become an Ethereum validator," ethereum.org, 6 May 2021. [Online]. Available: https://ethereum.org/en/eth2/staking/#:~:text=Ethereum%20will%20have%20a%20proof-of-stake%20Beacon%20Chain%20and,fully%20transition%20to%20a%20proof-of-stake%20system%20once%20. [Accessed 8 May 2021].

[43] D. Rhodes, "Blockchain Programming Languages: An Introductory Guide," Komodo, [Online]. Available: https://blog.komodoplatform.com/en/blockchain-programming-languages/. [Accessed 20 January 2021].

[44] B. Anderson, "The Most In-Demand Hard and Soft Skills of 2020," LinkedIn, 9 January 2020. [Online]. Available: https://business.linkedin.com/talent-solutions/blog/trends-and-research/2020/most-in-demand-hard-and-soft-skills. [Accessed 20 January 2021].

[45] M. Draper, "The most popular programming languages used in blockchain development," 18 JANUARY 2019 . [Online]. Available: https://www.freecodecamp.org/news/the-most-popular-programming-languages-used-in-blockchain-development-5133a0a207dc/. [Accessed 20 JANUARY 2020].

[46] "HashCash, the origin of the blockchain revolution?," Davies, [Online]. Available: https://www.daviescoin.io/blog/hashcash-the-origin-of-the-blockchain-revolution. [Accessed 13 Feb 2021].

[47] A. Lielacher, "The History Of Bitcoin Part 3: What Is B-Money," btcmanager, 4 April 2018. [Online]. Available: https://btcmanager.com/the-history-of-bitcoin-part-3-what-is-b-money/. [Accessed 21 Feb 2021].

[48] A. M. Antonopoulos, Mastering Bitcoin: Programming the open blockchain, vol. 2, O'Reilly Media, Inc., 2018.

[49] U. W. Chohan, "The Double Spending Problem and Cryptocurrencies," *SSRN Electronic Journal*, 2017.

[50] A. Narayanan, J. Bonneau, E. Felten, A. Miller and S. Goldfeder, "Bitcoin and cryptocurrency technologies: a comprehensive introduction," *Princeton University Press*, 2016.

[51] S. Bistarelli, I. Mercanti and F. Santini, "An analysis of non-standard bitcoin transactions," in *In 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), IEEE*, 2018.

[52] J. Thomason, S. Bernhardt, T. Kansara and N. Cooper, "Blockchain Introduced," *Research Anthology on Blockchain Technology in Business, Healthcare, Education, and Government. IGI Global*, pp. 40-64, 2020.

[53] T. Laurence, Blockchain for dummies, John Wiley & Sons, 2019.

[54] J. Fairfield, "Smart contracts, Bitcoin bots, and consumer protection," *Wash. & Lee L. Rev.*, vol. 71, no. 35, 2014.

[55] E. Advisors, D. Sornette, U. Advisors and V. Lange, "ETHEREUM ANALYTICS," Master Thesis, 2019.

[56] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper* , vol. 151, no. 2014, pp. 1-32, 2014.

[57] C. Harper and c. Kim, "Ethereum Gas Limit Hits 15M as ETH Price Soars," coindesk, 22 April 2021. [Online]. Available: https://www.coindesk.com/ethereum-gas-limit-eth-price-soars. [Accessed 7 May 2021].

[58] "Welcome to DLive," [Online]. Available: https://community.dlive.tv/about/welcome-letter/. [Accessed 29 April 2020].

[59] "Lino Blockchain Overview," [Online]. Available: https://docsend.com/view/y9qtwb6. [Accessed 29 April 2020].

[60] M. Freitas, "twister-a P2P microblogging platform.," *arXiv preprint arXiv,* vol. 1312, no. 7152, 2013.

[61] A. Chakravorty and C. Ron, "Ushare: User Controlled Social Media Based on Blockchain," in *Proceedings of the 11th international conference on ubiquitous information management and communication*, 2017.

[62] H. Shrobe, D. L. Shrier and A. Pentland, "Enigma: Decentralized computation platform with guaranteed privacy," *The MIT Press,* pp. 425-454, 2018.

[63] A. P. Joshi, M. Han and Y. Wang, "A survey on security and privacy issues of blockchain technology," *Mathematical foundations of computing,* vol. 1, no. 2, p. 121, 2018.

[64] "Ethereum Daily Transactions Chart," [Online]. Available: https://etherscan.io/chart/tx. [Accessed 30 September 2019].

[65] L. Arnold, M. Brennecke, P. Camus, G. Fridgen, T. Guggenberger, S. Radszuwill, A. Rieger, A. Schweizer and N. Urbach, "Blockchain and Initial Coin Offerings: Blockchain’s Implications for Crowdfunding.," *Business transformation through blockchain, Palgrave Macmillan,* pp. 233-272, 2019.

[66] E. Politou, F. Casino, E. Alepis and C. Patsakis, "Blockchain Mutability: Challenges and Proposed Solutions.," *IEEE Transactions on Emerging Topics in Computing.,* 2019.

[67] B. Hesse, "What Is IPFS and Why Does It Matter in Brave's Web Browser?," lifehacker, 21 Jan 2021 . [Online]. Available: https://lifehacker.com/what-is-ipfs-and-why-does-it-matter-in-braves-web-brows-1846103479. [Accessed 12 Feb 2021].

[68] S. S. Arslan, "Compress-Store on Blockchain: A Decentralized Data Processing and Immutable Storage for Multimedia Streaming," *arXiv preprint arXiv,* vol. 1905, no. 10458, 2019.

[69] R. Pinto, "On-Chain Versus Off-Chain: The Perpetual Blockchain Governance Debate," forbes, 6 Sep 2019. [Online]. Available: https://www.forbes.com/sites/forbestechcouncil/2019/09/06/on-chain-versus-off-chain-the-perpetual-blockchain-governance-debate/?sh=7e3caa991f5e. [Accessed 29 April 2021].

[70] R. Sheldon, "Comparing 4 decentralized data storage offerings," Search Storage, 30 Nov 2020. [Online]. Available: https://searchstorage.techtarget.com/tip/Comparing-4-decentralized-data-storage-offerings. [Accessed 1 March 2021].

[71] "Hosting on the Sia Network," Sia Setup, 4 March 2021. [Online]. Available: https://siasetup.info/learn/hosting. [Accessed 1 March 2021].

[72] Y. Sverdlik, "How Storj Is Building a Storage Cloud Without Owning a Single Disk," Data Center Knowledge, 22 Aug 2019. [Online]. Available:

https://www.datacenterknowledge.com/startups/how-storj-building-storage-cloud-without-owning-single-disk. [Accessed 20 Febreuary 2021].

[73] H. B. ,. C. Kopp and F. Kargl, "Koppercoin - A distributed file storage with financial incentives," in *International Conference on Information Security Practice and Experience*, Springer, Cham, 2016.

[74] A. Miller, A. Juels, E. Shi, B. Parno and J. Katz, "Permacoin: Repurposing bitcoin work for data preservation," *IEEE Symposium on Security and Privacy,* pp. 475-490, 2014.

[75] A. Sward, I. Vecna and F. Stonedahl, "Data insertion in bitcoin's blockchain," *Ledger,* vol. 3, 2018.

[76] HugPuddle, "Apertus – Archive data on your favorite blockchains," 2013. [Online]. Available: http://apertus.io . [Accessed 20 March 2020].

[77] K. Shirri, "Hidden surprises in the Bitcoin blockchain and how they are stored: Nelson Mandela, Wikileaks, photos, and Python software," 2014. [Online]. Available: http://www.righto.com/2014/02/ascii-bernanke-wikileaks-photographs.html . [Accessed 20 March 2020].

[78] A. Le Calvez, "Non-standard P2SH scripts," 2015. [Online]. Available: https://medium.com/@alcio/nonstandard-p2sh-scripts-508fa6292df5 . [Accessed 20 March 2020].

[79] Hyena, "Cryptograti.info," 2016. [Online]. Available: https://cryptograffiti.info. [Accessed 23 February 2019].

[80] "Miner fees," 2017. [Online]. Available: https://en.bitcoin.it/wiki/Miner_fees. [Accessed 30 September 2019].

[81] R. Matzutt, J. Hiller, M. Henze, J. H. Ziegeldorf, D. Müllmann, O. Hohlfeld and K. Wehrle, "A quantitative analysis of the impact of arbitrary blockchain content on bitcoin," in *In International Conference on Financial Cryptography and Data Security*, Springer, Berlin, Heidelberg, 2018.

[82] "Developer Guides," [Online]. Available: https://bitcoin.org/en/developer-guide#block-chain. [Accessed 3 August 2020].

[83] "Ethereum Average Block Time," ycharts, [Online]. Available: https://ycharts.com/indicators/ethereum_average_block_time. [Accessed 28 April 2021].

[84] C. COMBEN, "What Are Blockchain Confirmations and Why Do They Matter?," 10 OCTOBER 2018. [Online]. Available: https://coincentral.com/blockchain-confirmations/. [Accessed 6 Feb 2021].

[85] H. K. S. L. J. H. Younggee Hong, "Poster: De-mixing Bitcoin Mixing Services," 2018.

[86] K. Cremona, D. Tabone and C. De Raffaele, "Cybersecurity and the blockchain: preventing the insertion of child pornography images," *nternational Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC). IEEE,* pp. 197-204, 2019.

[87] R. Matzutt, J. Hiller, M. Henze, J. Henrik Ziegeldorf, D. Müllmann, O. Hohlfeld and K. Wehrle, "A quantitative analysis of the impact of arbitrary blockchain content on bitcoin," *In International Conference on Financial Cryptography and Data Security,* pp. 420-438, 2018.

[88] G. Zyskind and O. Nathan, "Decentralizing Privacy: Using Blockchain to Protect Personal Data," *IEEE Security and Privacy Workshops*, pp. 180-184, 2015.

[89] S. Ball, "Foucault and education: Disciplines and knowledge," *Routledge*, 2013.

[90] V. Goel, "Facebook tinkers with users' emotions in news feed experiment, stirring outcry," *The New York Times*, vol. 29, 2014.