REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY

IMPLEMENTING A BLOCKCHAIN PROTOCOL AND CREATING A DIGITAL ASSET TRANSFER ENVIRONMENT

Master's Thesis

SALIH CEMİL ÇETİN

İSTANBUL, 2018

THE REPUBLIC OF TURKEY
BAHCESEHİR UNIVERSITY

THE GRADUATE SCHOOLS OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING

IMPLEMENTING A BLOCKCHAIN PROTOCOL AND CREATING A DIGITAL ASSET TRANSFER ENVIRONMENT

Master's Thesis

SALİH CEMİL ÇETİN

Supervisor: ASSIST. PROF. SERKAN AYVAZ

İSTANBUL, 2018

THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

THE GRADUATE SCHOOLS OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING

Name of the thesis: Implementing a Blockchain Protocol and Creating a Digital Asset Transfer Environment
Name/Last Name of the Student: Salih Cemil ÇETİN
Date of the Defense of Thesis: 28 May 2018

The thesis has been approved by the Graduate School of Natural and Applied Sciences.

Assist. Prof. Yücel Batu SALMAN
Graduate School Director
Signature

I certify that this thesis meets all the requirements as a thesis for the degree of Master of Arts.

Assist. Prof. Tarkan AYDIN
Program Coordinator
Signature

This is to certify that we have read this thesis and we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Arts.

Examining Comittee Members	Signature
Thesis Supervisor Assist. Prof. Serkan AYVAZ	----------
Member Assist. Prof. Yücel Batu SALMAN	----------
Member Assist. Prof. Sabri Serkan GÜLLÜOĞLU	----------

ABSTRACT

IMPLEMENTING A BLOCKCHAIN PROTOCOL AND CREATING A DIGITAL ASSET TRANSFER ENVIRONMENT

Salih Cemil ÇETİN

Computer Engineering

Thesis Supervisor: Assist. Prof. Serkan AYVAZ

May 2018, 44 pages

This study is about investigating blockchain protocol basics, implementing a blockchain protocol and creating an online peer to peer digital asset transfer or payment environment on the protocol for financial purposes. The main purpose is to eliminate any third trusted mediator parties and enabling payment or transfer scenarios directly between electronic peers.

Money is defined by economists as anything commonly accepted by people for the exchange of goods and services. It is obvious that the history of payment is the same as the history of money. From 700 B.C. until today, payment systems' popularity has increased. With the rising of digital communication, geographic distances are getting shorter today. This also influences the payments and money transfer scenarios. In today's world, there are several organizations working as a mediator in payment or money transfer. December 2008 was a milestone for digital money transfer and electronic payment systems from this point with the application of Bitcoin. It is the first digital coin system which does not need any third party organization in money transfer scenario. The application was disrupting all the mediation layer in any digital cash transfer scenario by using its own currency BTC. Although it was declared as untrusted and unsteady in the beginning, lots of initiativies trusted and followed bitcoin and created new alter coins. Today, cryptocurrencies created a huge digital market.

In this thesis, a sample proof-of-work based blockchain protocol was implemented to create a digital asset transfer environment. It provides the main principles of distributed ledger technologies. C#.net was used as the base programming language for the implementation with MongoDB database.

Keywords: Blockchain, Distributed Ledger Technologies, Payments, Money Transfer, Digital Coins

ÖZET

BLOCKCHAIN PROTOKOLÜ GELİŞTİRİLMESİ VE BU PROTOKOL ÜZERİNDE DİJİTAL DEĞER TRANSFERİ ORTAMI OLUŞTURULMASI

Salih Cemil ÇETİN

Bilgisayar Mühendisliği

Tez Danışmanı : Dr. Öğr. Üyesi Serkan AYVAZ

Mayıs 2018, 44 sayfa

Bu çalışma, blok zinziri protokolleri ve temellerinin incelenmesi ve örnek bir blok zinciri protokolü geliştirilerek bu protokol üzerinde dijital değer transferi ve ödeme ortamının oluşturulması üzerinedir. İlgili geliştirmeler sayesinde, geleneksel ödeme ve para transferi senaryolarının aksine, aracı kuruluşların ortadan kaldırılarak doğrudan kişiler arası online ödeme ve transfer senaryolarının gerçeklenmesi amaçlanmaktadır.

Ekonomistler tarafından para, genel kullanıma açık olarak, insanlar tarafından mal veya hizmet karşılığı takas edilebilen değer olarak tanımlanmıştır. Ödemeler dünyasının ortaya çıkışı, paranın bulunması kadar eskiye dayanmaktadır ve popülerliği her geçen gün daha da artmaktadır. Dijital haberleşmenin hızla geliştiği son yıllarda, teknoloji sayesinde coğrafi mesafelerin yakın kılınması, ödemeler ve para transferi dünyasını da hızla değiştirmiş ve geliştirmiştir. Günümüz ödemeler dünyası, sadece bu alanda hizmet veren dev şirketleri barındıracak kadar büyük bir pazar oluşturmaktadır. 2008 yılının kasım ayı, özellikle bu şirketler için yıkıcı bir gelişmeye sahne olmuştur. Yeni bir uygulama olan Bitcoin ile, herhangi bir aracı kuruluşa ihtiyaç duyulmaksızın para transferi ve ödeme işlemleri rahatlıkla gerçekleştirilebilir kılınmıştı. Başlangıçta güvensiz ve stabil olmadığı eleştirilerine maruz kalan Bitcoin'i, aynı prensiplerle çalışan diğer girişimler ve uygulamalar takip etti. Bugün dijital para birimlerinin oluşturduğu pazar, göz ardı edilemeyecek kadar büyük bir kapasiteye ulaşmıştır.

Bu çalışmada, dağıtık defter teknolojilerinin temellerini barındıran bir blok zinciri uygulamasının geliştirilmesi ve incelenmesi ele alınmıştır. Bu doğrultuda, temel geliştirme dili olarak c#.net ve veritabanı teknolojisi olarak MongoDB ürünü kullanılmıştır.

Anahtar Kelimeler: Blok Zinciri, Dağıtık Defter Teknolojisi, Ödemeler, Para Transferi, Dijital Para

CONTENTS

TABLES

FIGURES

ABBREVIATIONS

DLT	:	Distributed Ledger Technology
DSA	:	Digital Signature Algorithm
DSC	:	Digital Supply Chain
ECDSA	:	Elliptic Curve Digital Signature Algorithm
GPU	:	Graphics Processing Unit
ICT	:	Information and Communiation Technologies
IOT	:	Internet of Things
MD4	:	Message-Digest Algorithm 4
MD5	:	Message-Digest Algorithm 5
POW	:	Proof of work
P2P	:	Peer to peer
RIPEMD:		RACE Integrity Primitives Message Digest
RSA	:	Rivest, Shamir ve Adelman
SHA	:	Secure Hash Algorithm
SPoF	:	Single Point of Failure
TCP	:	Transmission Control Protocol
TRX	:	Transaction
UTXO	:	Unspent Transaction Output

1. INTRODUCTION

Money transfer and payment have been a trending topic from the beginning of the trade in the world. And the history of trade is the same as the history of money. It started with simple coins that were used in exchange of services and goods at the beginning. For many years, coins were widely used in trading. After invention of paper and increasing usage of papers, paper money started to replace coins. In last decades, with the development of technology, money transformed into digital form as credit or debit cards, bank accounts, mobile wallets etc. People are having more and more digital and online presence day by day with rapid developments in technology. With this transformation, people need less of cash to cary physically in wallets in their pockets but need more of it in their virtual wallets in mobile devices to pay for daily shopping. Money has always been used as provision of a services and goods throughout its lifetime. The value of things has been estimated with it. Thus, money and payments, moreover money transfer have always been critically important in people's daily lives. Since the issue is critical matter, the rate of the profit is naturally huge. Because of this, there is a big industry formed around payment and money transfer companies that act as mediators in trade scenarios between sides of the trade. Usually these companies are the middleman in the traditional trade scenario as a third(and trusted) party. Both sides of the dealing parties must trust the 3rd party first. The trusted party's accountability is to ensure identification, verification, and realization of the transfer or payment. The middleman gets the transfer fee in this scenario. In basic words, 3rd parties sell people the trust. In today's economic world, 3rd party companies get a significant portion of the cash flow. By-passing these companies in trade results in a large amount of costs remaining in the pocket. In December of 2008, Satoshi published his product's whitepaper named "Bitcoin: Peer-to-Peer Cash Transfer" [1] and claimed the idea of converting each side of the transfer scenarios into a trusted party. The paper was the tipping point of distributed ledger technologies. Bitcoin is not only the milestone for cryptocurrencies and digital assets but also the beginning of the disruption of the 3rd party companies' hegemony. Today, blockchain protocol and moreover distributed ledger systems are very popular in creating trusted parties in untrusted environments in countless industries including financial services, energy trading, forecasting, security, cloud systems, IoT, agriculture etc. Nowadays, new projects on DLT's are the most popular

investment collectors by creating digital assets and cryptocurrencies. There are 3 main critical underlying technologies of blockchain protocol: Cryptography, P2P network, computational logic. Cryptography is used to identify and verify nodes in the network. It also enables to create secured transactions. Peer-to-Peer network based on distribution. To eliminate single points of failure and have a common security of all the system, peer to peer network is used.

Cryptography is the pactice and study to secure communication. It's one of the most vital fields in today's technology and communication world that provides truth and security. The history of cryptography goes back to Caesar. Julius Caesar as a commander, sent his messages to his soldiers with a mathematical based cryptography to get a secure communication. His soldiers had the true key to read the messages [2].

Today, data security is critically important so that it became a science field. Protecting data privacy and integrity in computer communication from unautherized modification, mathematical cryptography is used.

When the communication security is needed, the first things that utilize crytography are computer networks. The main instrument which provides the communication between computers is network. Network security is critical issue for the internet-based security mechanism. There various security application on network layer [3].

P2P network is based on sharing recources between two or more computers without any central point such as a server. Unlike the client-server systems, there is no server or client in this aproach, each user node is a server and a client in the same time. These networks can be a network connected with wires in a small area such as offices and ad-hoc connections or a network of much grander scale. The server-less structure provides to P2P networks the distribution in the same time.

1.1 GOAL OF THESIS

The main goal of this study is to implement a sample blockchain protocol with using blockchain protocol based on DLT basics and main principles, creating a digital asset transfer for an online payment environment between peers without third party organizations. There are several applications today for online payment which do not need any trusted third party organizations using different blockchain or DLT models such as

bitcoin, ethereum, iota etc. Since it is the milestone for all blockchain protocols and DLT solutions today, this study is based on Satoshi's Bitcoin application as the main model, which has started the DLT solutions and been used all over the world since 2009.
Although finance is the first and the largest industry who uses blockchain protocol, it is not the only one. There are several types of research and applications about DLT to solve different fields' problems. Another goal of this thesis is to build a basic blockchain protocol for non-financial purposes.

1.2 OUTLINE OF THESIS

The thesis is organized as follows:
Even this thesis based on building a sample blockchain protocol to provide a digital asset transfer scenario, the protocol is very applicable and useful for various fields. Because of this wide usage of the protocol, we investigate the different industries' approach to DLT and Blockchain protocol to solve different problems in chapter 2.
Chapter 3 can be described the container of the most critical points for the protocol. In the chapter, they are told that the main principles, basics, and philosophy which are the vital importance of a DLT or blockchain application.
In chapter 4, it can be found out the system design of the study. Also, its defined that how the behaviors differ for the different type of applications in the protocol and actions like how to behave if the message includes a new transaction or a new block and how these behaviors change for the application if it is a miner or a regular user. And the last topic of the chapter is the technologies used to implement the sample blockchain protocol, frameworks, encryption and third-party libraries, and database technologies.
And finally, Chapter 5 includes the results of building a blockchain protocol, advantages, and disadvantages of distribution of the all transactions in applications and future works.

2. LITERATURE REVIEW

P2P networks are distibuted systems. Not only distribution but also other technical aspects such as decentralized control, self-organization, adaption and scalibility [4]. Gnutella was one of the first P2P applications. Its main idea was to by-pass the copyright issues with sharing music resources between peers [5] [6]. Of course P2P network applications are not only used for music sharing. Another popular application of P2P networks is bittorent. The application achieves a higher level of robustness and resource utilization [7]. In this application, all upload costs are on the hosting machines. When multiple machines start to download the same file at the same time, they also upload pieces of file to each other. This mechanism provides large scalability to unlimited numbers of downloader machines [8].

There are several researches show that it also can be aplliable for IPTV services, multiplayer online gaming, traffic localization [9] [10] [11].

Although blockchain protocol and DLTs are not very old, they are not new inventions either. The principles and technologies underlying the protocol were already in use for several decades in modern computer technologies. When the founder of bitcoin published his whitepaper, he did not come up with a freshly found a technology but a gathering of currently used technologies with a different and brilliant view. Since the main idea of bitcoin was a peer-to-peer digital cash transfer, it didn't take a long time people to realize that it's not only a solution limited with electronic cash transfer problems but also easily applicable to a wide area to solve many problems.

First of all, it was the financial services who meet the blockchain protocol on DLT. They were the first group digital innovations in finance, also known as fintech, companies cared about the DLT in their business. As a result of bitcoin, it was a natural effect it to apply to payments, clearing, and settlement [12].

It was not only the finance companies that invested in DLT, also other industries joined in the DLT research and put forth their applications such as IBM's hyperledger and hyperledger fabric [13]. As it is told in hyperledger fabric's whitepaper, a blockchain emulates a trusted computing service through a distributed protocol. Their product is an enterprise-grade, open source distributed ledger framework and code base.

The protocol provides not only distributed ledgers, but also high security, scalability and temper-proof transaction history keeping. This is the main reason why other industries such as healthcare, energy trading, supply chain and agriculture, IoT etc also are interested in this technology. Even it was questioned by researchers whether the existence of states or not is still needed when there is a stable blockchain protocol [14] [15]. They refer to not only bitcoin but also other applications in their paper which are based on the creation of decentralized domain name system resistant to top-level domains censorship, decentralized voting systems, decentralized autonomous organization/corporations/societies and other blockchain functionalities. The idea's tipping points are the blockchain protocol's features that started to replace the government services.

The protocol is also very suitable for healthcare services. In 2016, a paper was published about blockchain-based health information exchange network [16]. It's important to keep patient information and share it between institutions securely. The main idea of the work is to describe an approach that can effectively and securely share sensitive healthcare information on a special network for data sharing purposes. It is obvious that one of the strongest sides of a blockchain protocol is secure information storage and transfers. In conclusion, the authors introduced a new consensus algorithm designed to facilitate data interoperability.

Another industry investing in distributed ledger technologies is energy trading market. As already known, energy trading market is one of the biggest trading platforms similar to money transfer and payments industry. Thanks to the same constraints with payments and money transfers, blockchain is very suitable for energy trading market. There is a promising start-up named WePower which aims to enable trading between green energy producers with consumers [17]. The project is based on ethereum blockchain and it tokenizes energy unit.

Similar to healthcare services, the supply chain and agriculture industries also welcome DLT. Digital supply chain integration in recent years is becoming increasingly dynamic. Accessing customer demand, sharing demands effectively, tracking products and service delivery are getting more important every day. Since DLT solutions are getting popular in supply chain industry in creating a successful, efficient and scalable digital supply

chain, some studies in this field discuss and analyze methods that may be used to improve DSCs with DLT solutions [18].

Advancements in Information and Communications Technology have started to become popular in agriculture too. Applications vary from early warning systems to measuring baseline data to planning. Since the blockchain protocol removes the single point failure issues and making critical information sharing more secure, it can also be used to make e-agriculture and farm systems better [19].

Not only ICT but also IOT is getting popular in recent decades. It has started to cover various aspects of our lives. Even it is widely used in today's world, it still suffers from privacy and security vulnerabilities. With bitcoin, blockchain protocol proved that it can solve this problem without any third and trusted party. However, blockchain protocol also has weak sides to be used in IoT applications such as computationally expensiveness, needing high bandwidth and delays. It's clear that IoT topology needs cheaper, faster solutions. There is a research study published in 2016 that gathers these two approaches to create secure and faster IOT applications which provide exact distributed trust [20].

Some other studies show that there are new opportunities and possibilities to apply DLT in other fields such as smart cities, identity management, online public notary etc [21] [22] [23].

Although DLT solutions are applicable in various fields thanks to its strengths, its inception was due to financial services. In this study, we also used it as a financial solution as a peer to peer digital asset transfer environment like bitcoin. However, this study is not only applicable to financial projects, but also any project that fundamentally requires security, anonymity, tamper-proof data keeping, or distribution.

3. PROTOCOL FOUNDATIONS

Blockchain protocol is not a new invention but a gathering of technologies already in use. Since the its main idea is to share and store all the common transaction history of all users in each peer's ledger, it is a difficult task using traditional techniques, and thus we have to change the traditional view of the system. To enable the aimed features of the blockchain protocol, we use cryptography, hash functions, digital signatures, peer-to-peer network structure etc. In this section, this principles and the reasons are discussed and detailed.

3.1. CRYPTOGRAPHY

The basic definition of the cryptography is a process of converting ordinary plain text into unintelligible text or vice-versa. All the used methods of this process is not only to protect the content of the encrypted text but also authenticate users in systems [24]. There are several algorithms that used today in computer science. Table 3.1 shows the security levels of these algorithms.

Table 3.1: Security levels of cryptography algorithms [25]

Security level (power of 2)	Algorithm	Keygen	Sign 59 Bytes (CPU cycles)	Verify (CPU cycles)	Private Key Size (bytes)	Public Key Size (bytes)	Signature Size (bytes)
80	RSA 1024	102,869,553	2,213,112	60,084	1024	128	128
	ECDSA 160	1,201,188	944,364	1,083,060	60	40	40
	MQQSIG160	799,501,482	6,534	92,232	401	137,408	40
	RainbowBinary256181212	30,311,648	38,784	43,800	23,408	30,240	42
96	RSA 1536	322,324,721	5,452,076	87,516	1536	192	192
	ECDSA 192	1,799,284	1,390,560	1,662,664	72	48	48
	MQQSIG 192	800,724,096	7,938	138,972	465	222,360	48
112	RSA 2048	786,466,598	11,020,696	125,776	2048	256	256
	ECDSA 224	2,022,896	1,555,740	1,821,348	84	56	56
	MQQSIG 224	1,107,486,126	9,492	184,392	529	352,828	56
128	RSA 3072	2,719,353,538	31,941,760	230,536	3072	384	384
	ECDSA 256	2,296,976	1,780,524	2,085,588	96	64	64
	MQQSIG 256	1,501,955,022	9,138	218,700	593	526,368	64
	TTS 6440	60,827,704	84,892	76,224	16,608	57,600	43
	3ICP	15,520,100	1,641,032	60,856	12,768	35,712	36

Modern cryptography concerns the following issues:

i. Confidentiality: the encrypted information must be unintelligible to anyone.
Integrity: encrypted information or text cannot be altered.

ii. Non-Repudiation: User who encrypts and sends the message, cannot deny that he or she didn't send it.

iii. Authentication: Both sender and receiver can identify each other by using cryptography.

In general usage, there are 3 types of cryptography techniques:

3.1.1. Symmetric Key Cryptography

Based on sharing the same key between sender and receiver. When sender using the key to encrypt the plaintext, the receiver uses it to decrypt the encrypted text [26] [27].

3.1.2. Asymmetric Key Cryptography

In this type of cryptography, there are 2 different keys to be used, one is the public key which is publicly shareable and the other one is the private key which must be known only by the limited users who is the signers usually. Table 3.2 shows the comparison of symmetric and asymmetric cryptographies.

Table 3.2: Comparison of symmetric and asymmetric encryptions [28]

	Symmetric Encryption	*Asymmetric Encryption*
Functionality	Allows efficient communication between two parties in a closed environment.	Enables security in settings in which symmetric encryption simply does not work or is more difficult to implement.
Computational Effiency	Computes incredibly fast, since the relatively simple operations used are executed very efficiently.	Computes slowly, using computatianally havy and complex operations, based on the difficulty of solving number-theoretic problems.
Key Size	Uses 128-bit eymmetric keys, which are considered very secure.	Employs key sizes of at least 1000 bits to achieve sufficient, lasting security.
Hardware	Performans simple algorithms, requiring inexpensive hardware.	Implements complex and time-consuming algorithms that need more powerful hardware.
Security	No difference. Security is based on the strength of the algorithm and size of the key. Good algorithms exist for both encryption methods and key size effectiveness.	

3.1.3. Public Key Cryptography

It is based on two different but relatively paired to each other keys as public and private keys. The public key can be distributed to anyone, and the private key must be protected by the sender only. Also, public-key cryptography is very widely used technique. The public key is used to encrypt the plaintext and the private key is used for decrypt it. Our

study is also one of the Public-Key Cryptography application. There are different techniques to use public-key cryptography as digital signatures.

3.1.4. RSA Digital Signatures

The cryptographic algorithm was introduced by Ron Rivest, Adi Shamir and Leonard Adleman in 1978. It is one of the most widely used public-key cryptography techniques [29]. It is also used as a digital signature with it's public and private key pairs [30].

3.1.5. ECDSA

Elliptic curves not only promise higher security but also better performance compared to first generation public-key encryption techniques [31]. Elliptic curve digital signatures are easy to apply and comparing to RSA, it has shorter key lengths [32]. With ECDSA, signing and verifying transactions is fast and secure. This is extremely advantageous for DLT applications. There are several signature algorithms in use today. Most common ones are RSA and ECDSA. In table 3.2, there is a comparison between these two algorithms' security strength for their key sizes. In this thesis, we used ECDSA to create digital signatures, to sign transactions and to verify the nodes.

Table 3.3: Security strength by key sizes of RSA and ECDSA

Security Strength	RSA Key Size	ECDSA Key Size
80	1024	160-223
112	2048	224-255
128	3072	256-383
192	7680	384-511
256	15360	512+

Source: https://aws.amazon.com/blogs/networking-and-content-delivery/cloudfront-now-supports-ecdsa-certificates-for-https-connections-to-origins/

3.1.6. Hash Functions

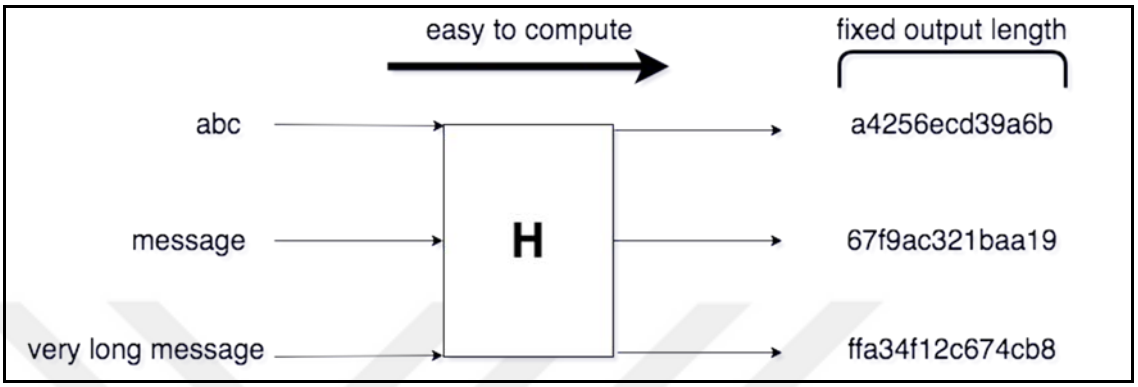
Hash functions are one-way functions which means we cannot know any information about the input if we know only the output [33]. They can be expressed by the formula:

$h(x) = y$ where $x \in Z$ and $y \in Zn$

There are some properties provided as it seen in figure 3.1 about hash functions:

i. Output always has the same length whatever input length is. (fixed-size output result)
ii. Input length is usually longer than output.
iii. To calculate function is an easy and fast operation.

Figure 3.1: Basic Hashing System



One-way functions are “must have” features of cryptography. The main idea of these functions is to protect the data. The output of these functions cannot be understood by the others but the receiver. Moreover, with this idea, we need to transform the message if somebody reads or hears the transmitted message, he or she should not able to understand what the message is. Hash functions are especially used in cryptography for authentication, data integrity, and non-repudiation.

Another main property of hash functions is security. These functions are mathematically strong one-way functions and it is difficult to computer inverse of the output to find the non-transmitted message.

There are several hash functions used in common. These are MD4, MD5, RIPEMD, SHA1, SHA256 etc. In our study, we used SHA256 as hash functions to provide proof-of-work concept.

3.2. PEER TO PEER NETWORK

Unlike the client and server structure, in a peer to peer network, each node work as a client and also server to provide a truly distributed communication model. Basically, peer-to-peer networks are used for sharing data between computers without needing powerful

and expensive servers. This architecture enables to distribute the main cost of sharing huge amounts of data between peers [34].

A P2P network is also good for providing anonymized routing in the traffic of the network, parallel computing issues, storage distribution etc.

Eliminating client and server model, promises various advantages. Firstly, P2P network is robust. In case of any device goes down in the network, the network goes on working (this is the root of a single point of failure issue). It allows sharing data easily both directions, download, and upload. It doesn't only enable sharing files or data but also the devices and sources such as printers, scanners etc.

In conclusion, the major motivations of P2P networks are increasing cost efficiency, decreasing the risk of single point of failure scenarios and getting a better scalability with adding new peers to the system and scale a larger network [35]

3.2.1. Centralized, Decentralized And Distributed Systems

In computer science, centralized solutions are mostly performed around a central server which heads the registered hosts and manages. Clients and hosts connect to servers over TCP by sending requests and receiving responses [36].

Centralized systems are easy to maintain. In this system, it is enough to fix any problem in server side and hosts and clients continue to run. In the same time, this feature brings the single point of failure issues which means when one point gets down all the system is down. Centralized system chaos scenarios usually work on this issue [37]. The system is also not good for scalability. Designers of the system must spend deceitful afford to calculate if the system is scalable to handle the changeable count of clients and hosts.

In contrast to low scalability, it is very fast to create centralized systems. Moreover, these systems use a single framework and hence they don't have diversity.

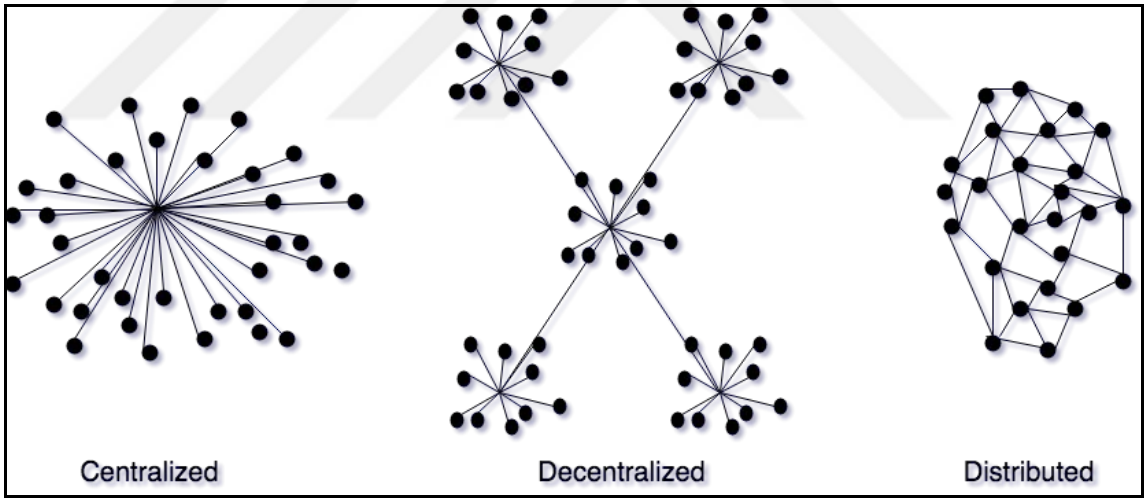
Decentralized systems are a type of form between centralized and distributed systems. These systems can be thought as a combination of multiple centralized systems. Although they are not completely distributed, they are also not distributed.

With the rapid increase of bandwidth usage popularity in computer science and need of resource sharing, took us to distributed network systems and moreover peer to peer solutions. Because of the problems come with centralization, the popularity of peer to peer solutions growths every day.

Unlike centralized solutions, distributed systems don’t have any central servers. In this architecture, each client is a server at the same time. Since every client has its own authority, this system is difficult to maintain. To solve a problem in a client means that just one client has affected from the solution. Since all clients work as a server in distributed systems and not having a central server, distributed systems have no single point of failure. This means, in contrast to centralized systems, when a peer fails, the network goes on working.

Distributed systems are also fault tolerant and scalable systems comparing to centralized systems. In decentralized networks, an infinite number of clients can be added to the network without any scalability problem. On the other hand, because of not having a single framework, distributed systems are difficult to create. Developers must work with lower layer details for communications and resource sharing. Figure 3.2 clearly shows the forms of these 3 systems.

Figure 3.2: Centralized, Decentralized and Distributed System Models

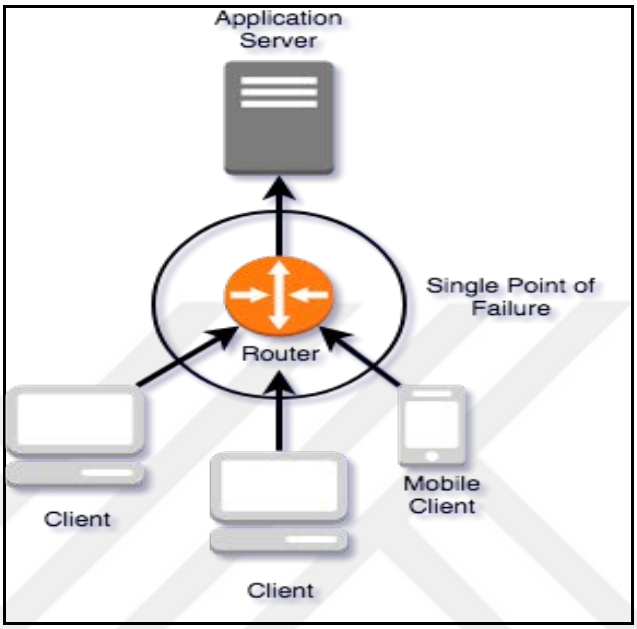


3.2.2. Single Point Of Failure

As IBM described, “A single point of failure is an environment where one failure can result in the simultaneous loss of both the coupling facility list structure for a log stream and the local storage buffer copy of the data on the system making the connection” [38]

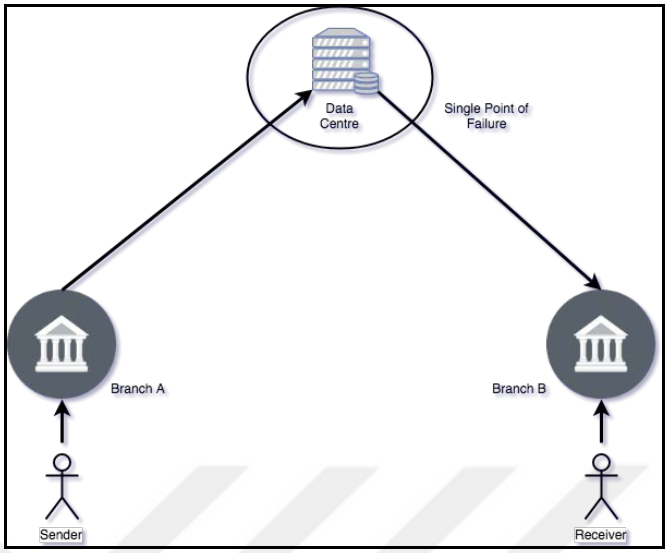
This concept can be seen in various systems in our lives. In a basic network which is shown in figure 3.3, the router may be a single point of failure because of the unique connection point between the application server and clients [39], [40].

Figure 3.3: A SPoF example for a client-server system



With a similar view in a money transfer scenario, a third party organizations' data center may be a single point of failure as it seen in figure 3.4. In this scenario, a sender visits a bank company's branch A to send money to his friend. The transaction record is kept in the company's data center. And receiver visits the branch B and asks for withdrawal. In this scenario, since all the transactions are stored in the ledger of the company in the data center, the data center is a single point of failure. Because of this issue, companies spend huge amounts for data protection and security, and this effects to the sender or receiver as a transaction fee.

Figure 3.4: A SPoF example for traditional money transfer scenario via a bank

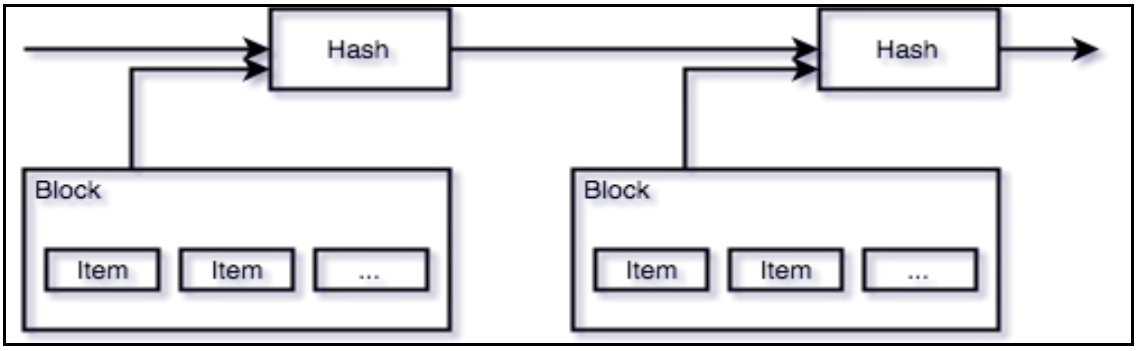


In today's finance industry, money transfer and therefore payment systems have this kind of problems. That's why lot's of people still pay extra fees for mediation like identification, verification, security etc. in money transfer and payment scenarios.

3.2.3. Timestamp Server Approach

This technique is used to prove the existence of a digital document in the time by taking a hash of a block of items to be timestamped and widely publishing the hash as it seen in figure 3.5. Applied methods are closely related with digital signatures and hashes. When signing process is done for a special purpose by the client, in our scenario this is a transaction, the client machine creates a hash of the file to show that it's been just signed. This also helps to solve the double spending problem in financial applications [41].

Figure 3.5: A timestamp server



3.3. COMPUTATIONAL LOGIC

Another essential part of blockchain protocols is computational logic. This study also investigates how addresses are created and work, explanation of proof of work mechanism and reasons why it is used. Furthermore, Satoshi's mint-base transaction structure and simplified verification are discussed. It is also shown how to solve critical issues in traditional money transfer scenarios and how to remove third-party mediators' roles in money transfer scenarios.

3.3.1. Addresses

Bitcoin's blockchain protocol is the first well-known blockchain protocol. In this protocol, the hashes of public keys are simply used as addresses. It has some special rules such as addresses begin with number "1" or "3" and have to be in a fixed length etc.

The addresses in blockchain protocols are public keys. In public key cryptography, we use directly relative public key and private keys together to sign any file. As its name implies, the public key can be shared with everyone publicly. It works just like a postal address to get a mail. And the private key is like the unique key opens the related mailbox. In cryptography, the sender encrypts a file to send and the receiver, who owns the public-private key pair, gets the encrypted file and decrypts it with his own private key. If a file is encrypted with a public key, the unique key to decrypt it is a private key. This is the traditional usage of public-private keys in cryptography.

To create a trusted environment without any central authority in distributed ledger technologies, it is critically important being provable about who the signer is of the transaction. To provide this, we use public key cryptography. In blockchain protocol, every node has its own public and private keys. When the node creates a transaction, it signs the transaction with its own private key and publishes the transaction to the P2P network. After this process, the network has a signed transaction with a public key which means that the owner of the private key used to sign the transaction. If the signature provides that it is signed with the private pair of the public key then the transaction is assumed that the owner of the public key is also the owner of the transaction. This mechanism lets us to use a public key as an address in the protocol.

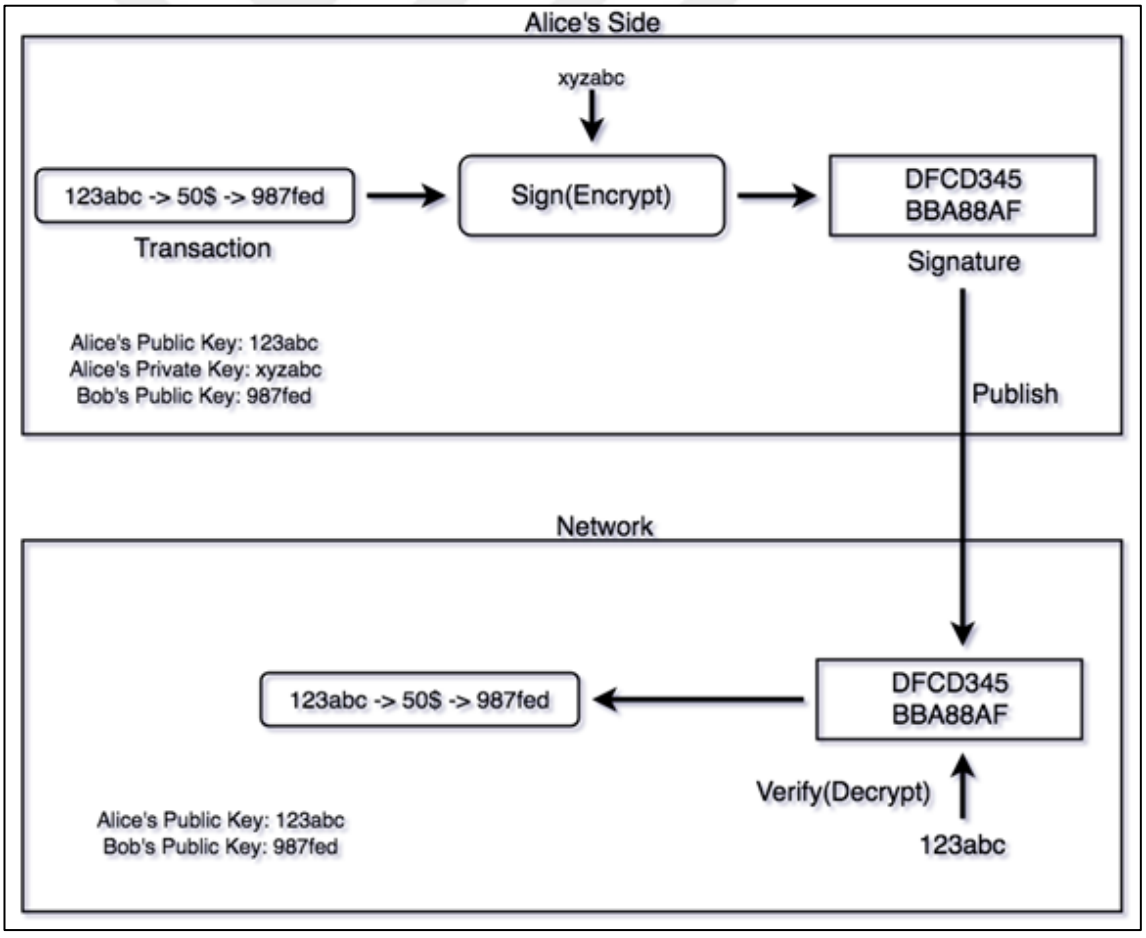
To illustrate the mechanism in a simple example, let us follow a basic scenario to see how a transaction signature algorithm work. Assume that Alice wants to send 50$ to Bob and

he creates a transaction object and signs it with his private key. Then he publishes it to the network. A node who listens to the published transactions in the network catches the transaction. He can easily to understand if it's signed by Alice. The process is visualized in figure 3.6.

In this scenario, we are not hiding transaction information from the network. Although blockchain transactions are transparent, they are extremely temper-proof. The public key cryptography mechanism also provides anonymity. As we use public keys as an address, the network always knows which address is sending how much to which address. But no longer information is shared who is the owner of the address.

This mechanism lets us to solve the identification problem which is solved by the central authority in traditional money transfer scenario.

Figure 3.6: Scheme of how a transaction is published to the network



3.3.2. Proof-Of-Work Concept

In an untrusted environment, we use third-party organizations for mediation. In recent examples, it was a bank company to transfer money between 2 branches. Third party organizations or authorities take transaction fees for identification, verification and transaction security.

Since there is no central authority for mediation in blockchain protocol, it is another critical issue to solve is verification. Money transfer via a bank is simply changing the balance of corresponding accounts in bank's database. The bank keeps all the transaction histories, all the accounts and their owners' information which are assumed as highly sensitive info. It's easy to validate any transaction if you have all the information you need.

Peer to peer money transfer means trying to get rid of any third party company, moreover, it also means removing the need for data centers in the scenario. Blockchain protocol solved this issue by using distributed, and shared ledger approach. In distributed ledger approach, every node can hold the all transaction history. The primary idea is to hold everyone updated and prevent potential frauds in the network. When a transaction published, every node can check if the transaction is valid by this mechanism. Every transaction needs validation to become valid and to be validated, more than %50 of the network should agree it's validation. The basic rule is "democracy works to remove autocracy".

With all this information, in blockchain protocol we need validators. Although everyone can validate transactions, the system doesn't want any cheating validator in it because of the trust and security. To provide trust and validation security, one method is applying cost functions [42]. Cost functions are simply based on difficulty to proof something asked. In bitcoin blockchain protocol, proof of work concept was used as a cost function to proof that the solver of the problem is a validator and spent energy to solve it. In blockchain protocols which uses proof of work concept as the main cost function, the validators are called miners. Miners basically try to solve a mathematical probability problem to create a block.

Blocks can be thought as containers of valid transactions. Miners listen to the network to catch freshly published transactions. They bundle the transactions, validate them, create a block contains valid transaction and some more information and publish the new block

to the network. If nodes validate the block, they add it their ledger and now it is the current block in the chain.

Proof of work concept, as mentioned previously, is based on solving a mathematical probability problem. To understand the problem, let's turn back to hash functions. Hash functions are one-way functions and their output is a unique value for each different input. The output is not interpretable which means it is impossible to understand what the input is if you just know the output. These functions are commonly used to ensure an information is not tempered.

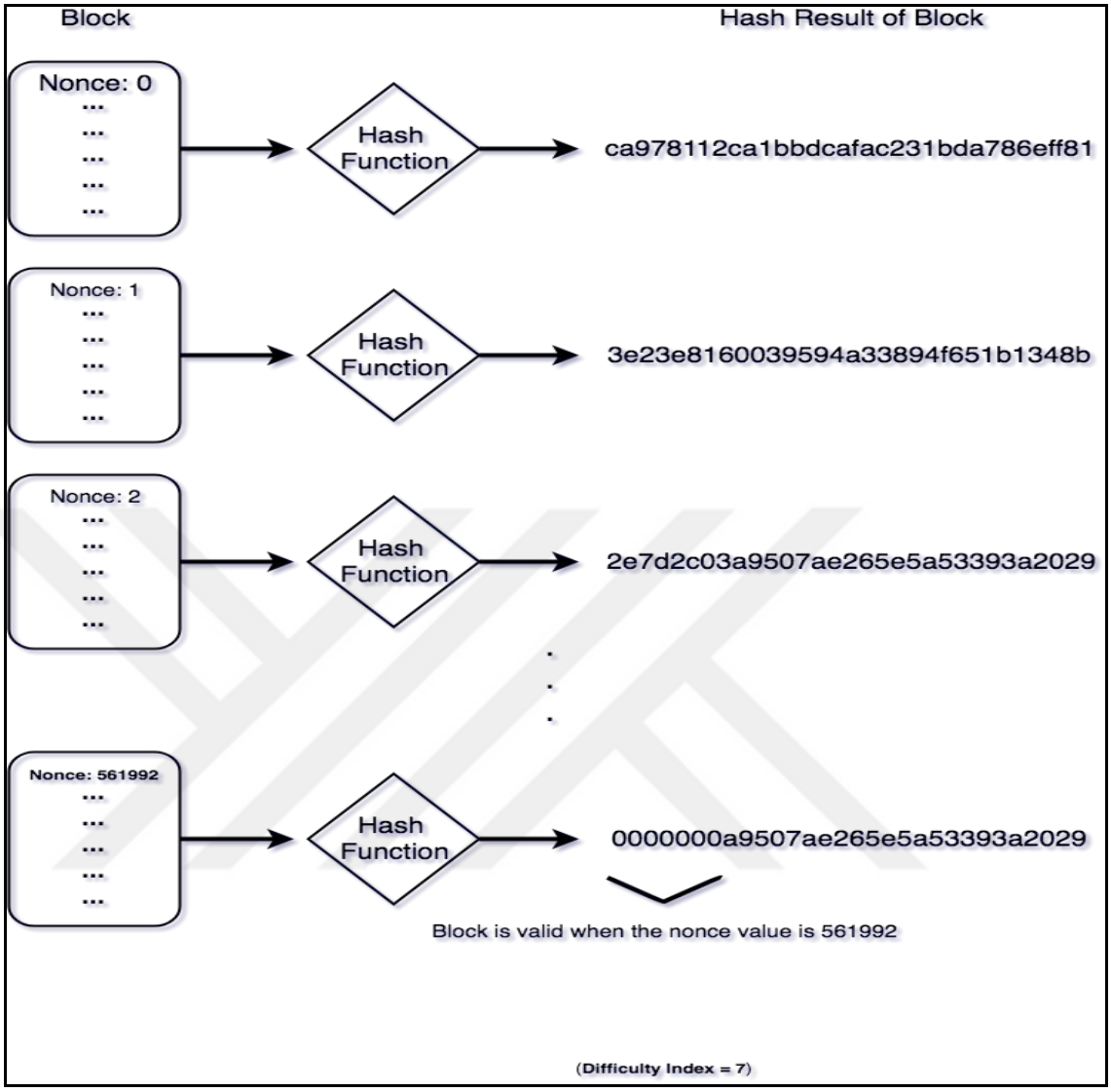
When a miner finishes the validation process of transactions, it gets some other important information together and creates a block. In bitcoin blockchain, there is a property named "Difficulty Index" which represents the beginning zero counts in the hash result of a block. If the difficulty index is 10 currently in the protocol, every block's hash result must start with 10 zeros to show the block is valid. Every block has a key property named "Nonce". The nonce is simply a numeric value which gives us the true hash result of the block if the block's nonce value is correct. A miner's all mining process is trying to find this nonce value to provide the hash result with difficulty index number of starter zeros.

If the nonce value doesn't provide the starting count of zeros in the hash result, miner increments the nonce value and try the hash algorithm again until the true nonce value gets found like it is seen in figure 3.7.

Although to find the true nonce value is very difficult, it is very easy to validate the block if the nonce value is known.

This is proof of work concept. It is simply trying to find a specific number which gives us the desired block hash result [1].

Figure 3.7: Recursion of proof of work process



3.3.3. Simplified Verification

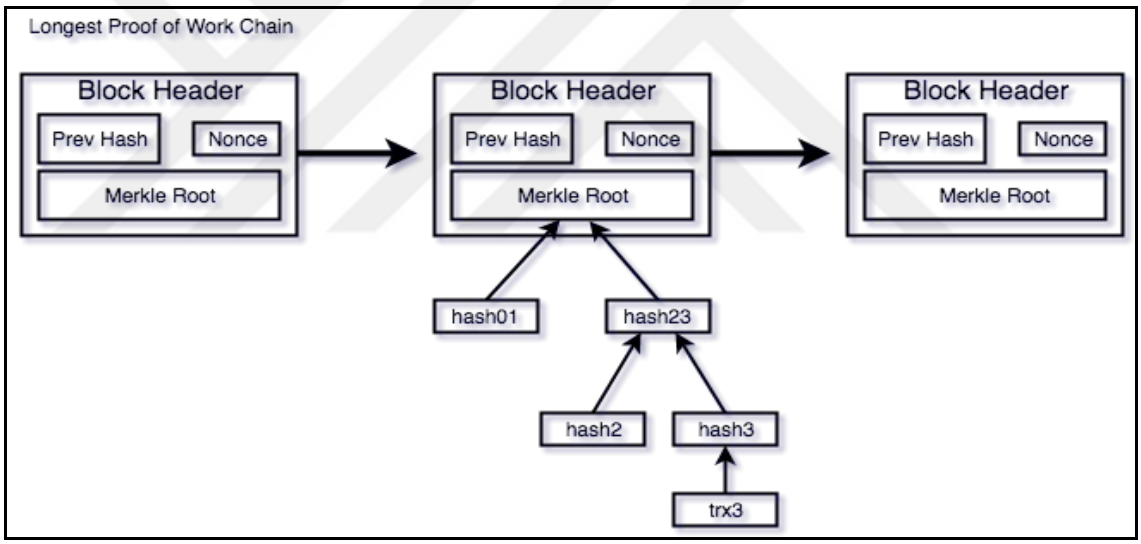
In distributed ledger systems, every single node must have the ledger containing all of the transaction history. The notion of the blockchain democracy requires all nodes' attendance in verification to create a trusted environment. To verify any block, and so transactions in it is possible if only all nodes have the single transaction history in the same and true order. To remove the risk of double-spending, all the transaction histories kept in the network should be the same with transactions order.

According to Satoshi, it's not necessary to run full network node for verifying transactions [1]. This is possible with block header structure which is created smartly.

A block consists of 2 parts: block header and transaction list. Transaction list contains transactions validated by the miner who creates the block. And the block header is a part contains block's characteristic information like a nonce, block id, previous block id, hash merkle root, difficulty index, creation time etc. In figure 3.8, it can be seen that block header keeps the characteristic informations, so it can be thought like a fingerprint of the block.

Hash merkle root is a concept simply keeps the hash of the transaction hashes. It is created in the form of a root, with transaction hashes in the deep. Since header contains the hash merkle root, the transactions cannot be modified in the block even their order cannot be changed. Moreover, if newly created block fits the main chain of the block, this means the new block is validated.

Figure 3.8: How all transaction information fits into the chain



Hash algorithms are simply used for getting abstract of any information regardless of its length. In blockchain protocol, we also use hashes of the blocks to get block information in a short form. Due to blockchain protocol's chain structure, each block has the previous block hash which means having all the information about the previous one. As it mentioned before, by using hash merkle root of transactions list in blocks, we lock and keep every information of the transactions in the block header. And by getting the hash of the block, we also lock and keep all the information of the block in one hash result.

This is how satoshi's simplified payment verification or simplified verification work. If a newly created block has the latest block hash information and also provides the required

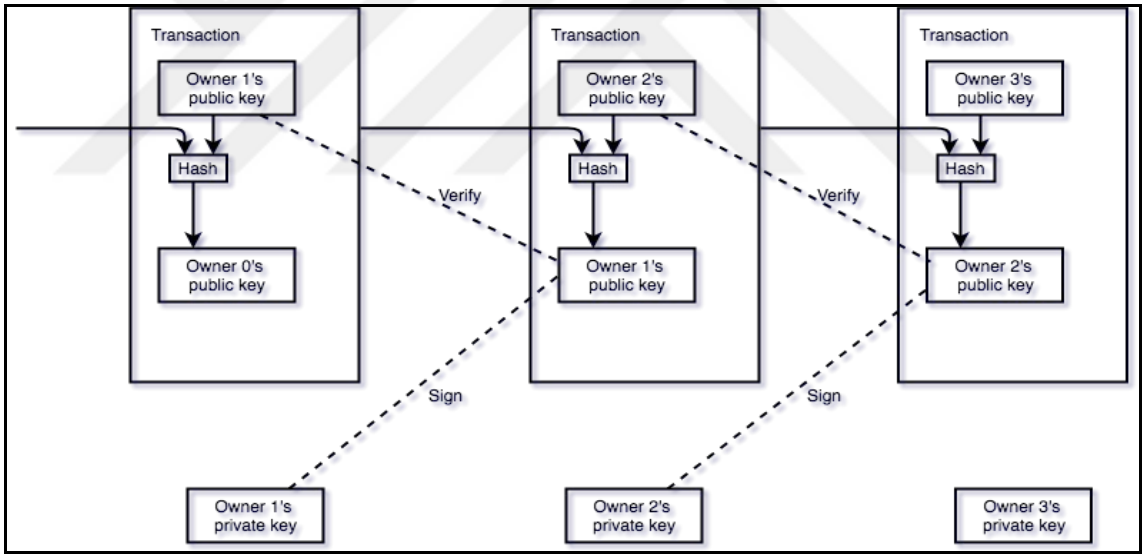
conditions, this means it is valid for every client. A client who gets a new block does not have to run all the node to verify the block and transactions in it, he only needs to control if it fits the main chain [1].

3.3.4. Applying Mint-Base Transaction Model

The key requirement for applying Mint-Base transaction modelis the need for verifying a payment by receivers and making sure that the coin was not spent in previous transactions.

In order to solve this problem, it is needed to create a central authority to check every transaction at each time for double spending. In the mint based model, every transaction goes through to the mint and the mint is aware of all transactions if the coin was used before in another transaction or not.

Figure 3.9: The transaction mechanism in the protocol



In blockchain protocol and another DLT applications, there is no central authority. Since central authorities are avoided, the network should process the transactions as mints. To do this in the protocol, digital signatures are used in each transaction.

As it is shown in figure 3.9, every coin must be assigned to a user's public key or address to get used by him in the network. A coin just can be used by this owner user in a transaction. The only way to enable verifying by the receiver, the system only cares about the earliest transaction that coin was used. The later transactions will be ignored to

eliminate double-spending. The unique way to provide this, announcing transactions publicly to the network and participants will agree on a single history of the order in which they were received.

4. SYSTEM DESIGN

In this chapter, our blockchain protocol model, the foundations and its components are explained in detail. As an application, this is the most critical step to create a project on a blockchain protocol.

4.1. MODELS

This section explains the models that were used in our study while creating a blockchain protocol.

4.1.1. Input

An Input, as the name implies, represents main data to feed transactions. Our study utilizes the mint based transaction mechanism which was also used in bitcoin. In the mint based transaction, every input is basically an output of a past transaction.

In our blockchain protocol, there is an unspent transaction outputs table to use as input which means a spendable coin. UTXO table also represents us the networks total amount.

4.1.2. Output

As stated previously, every input must be an output of a past transaction.

Outputs have an amount in the protocol's base currency and are counted as coins. They also have to be assigned to a user's address spent by the related user.

When an output spent in a transaction as an input, it has to be deleted from UTXO table and inserted into the Inputs table. Inputs table represents the spent outputs which cannot be spent anymore on the network.

Since UTXO and Inputs tables stored on the network, the network does not need to hold total amount of any user. If a user wants to calculate how much coins it has must query the UTXO table and sum the amounts of records which are assigned to his address. This sum result shows user's total spendable amount of coins.

Also in transactions, input and output management is under the responsibility of the sender. While a sender creates a transaction, he has to evaluate and calculate enough inputs and suitable outputs required for the transaction. If the inputs and outputs do not

provide a well-balanced transaction, the transaction will be refused by the miners on the network.

4.1.3. Transaction

Although it became famous with bitcoin, we don't always have to create a financial application on blockchain protocol. The protocol has different strong properties like removing third trusted parties, fast transaction mechanism, keeping data secure and trusted etc. We can easily focus on one or more strong sides of the protocol. Nevertheless, if we aim to build a financial application on blockchain protocol, we have to use transaction models.

Transactions are the main structure which keeps the inputs and outputs to symbolize interactions between peers or users in the network. Since we focused on digital asset transfer on blockchain protocol, we naturally used the transactions as the main object to expose interactions between users.

In this thesis, transaction objects have other main objects such as inputs, outputs, signatures.

Transactions can be thought by two main information parts in general. One is financial information between sender and receiver, and the other one is cryptographic information like signature which shows the transaction is created by the owner.

4.1.4. Signature

While transactions are very secure and essentially tamper-proof in blockchain protocols, they are also very public. This is one of the most important features of blockchain protocols and it is provided by using public key signature algorithms.

In the network, transactions are publicly published. Every user can listen to the transactions. This transparency is not an objectionable situation because of users' anonymization by using their public keys as addresses.

In blockchain protocols, mostly public key signature algorithms are used. With this algorithm, it's very easy to check if the transaction was signed by the real owner -it's assumed that the owner of the transaction is also the unique owner of the private key- just having transaction, signature and the public key trio. If the signature does not provide the transaction information, miners will refuse the transaction.

4.1.5. Block

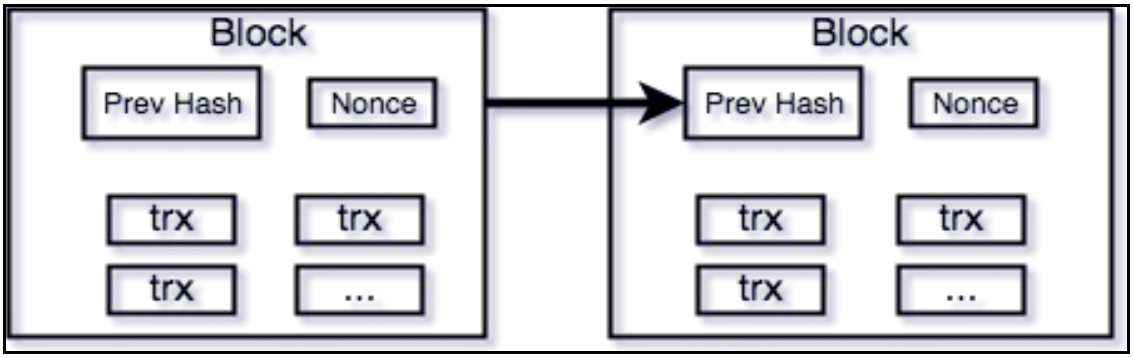
The underlying notion of the blockchain protocol is chaining blocks by storing previous block's hash in every block. Since the protocol is based on this chain of blocks structure, the block model is one of our base models.

While signatures provide transaction security, including hash reference of the previous block in every block, provides the chain and temper-proof history security like it shown in figure 4.1.

Miners in the network use a proof of work mechanism to create a new block. This is also another security application of the protocol. To provide a secure protocol, the block creation process must have some challenges in it. If this challenge becomes an easy process, the chain becomes insecure and opened for forgery attacks by creating retroactive blocks. This means that the single and true history is in danger.

The PoW mechanism makes blocks valid. If the hash of a block provides the desired hash result -in our study, this works with difficulty index and a related count of starter bytes of zeros- it means that the block is valid with transactions and other information in it and fits the longest PoW chain as the latest chain.

Figure 4.1: Referencing mechanism that creates the chain of blocks



A block's hash carries all the block information in it. In case of any information changes, the hash gives us a completely different result which changes the next block's hash result at the same time. Since it is a chain, when any information changes in a block, the later blocks will not provide the valid proof of work result. So this illegal alteration will turn whole the chain into an illegal and no PoW provider chain.

A block consists of two parts: transaction list and block header.

4.1.5.1. Transaction list

Transaction list, as the name implies, a list of all transactions in the block. Especially miners listen to the transactions published into the network. The validation task of the transactions is under the responsibility of miners. Miners must have all node and pick the valid transactions from invalid ones. In order to run this validation algorithm, they have to look to all nodes and history if the transaction is valid or not. When they satisfied with the validation process-miners usually consider the transaction fees to end the calculation up-, they bundle the valid transactions and begin to another one, proof-of-work process.

4.1.5.2. Block header

Block header keeps critical information about current block such as hash merkle root which is a fingerprint of transaction list in the block, previous block hash, proof-of-work result named nonce, exact time information to provide timestamp server concept.

Keeping hash merkle root guarantees that the transactions are not tempered. The concept strictly locks every information in it even the order of transactions. Adding this information to block header enables users to hold only the header's chain in their storage to subsist if they are not a miner. Users if they are not miner have to store at least pruned data. To verify its validity and add a new block to their ledger, they need this pruned ledger. Block header and transaction list structure are shown in figure 4.2.

Figure 4.2: Sample model of a block

4.1.5.2.1 *Hash merkle root*

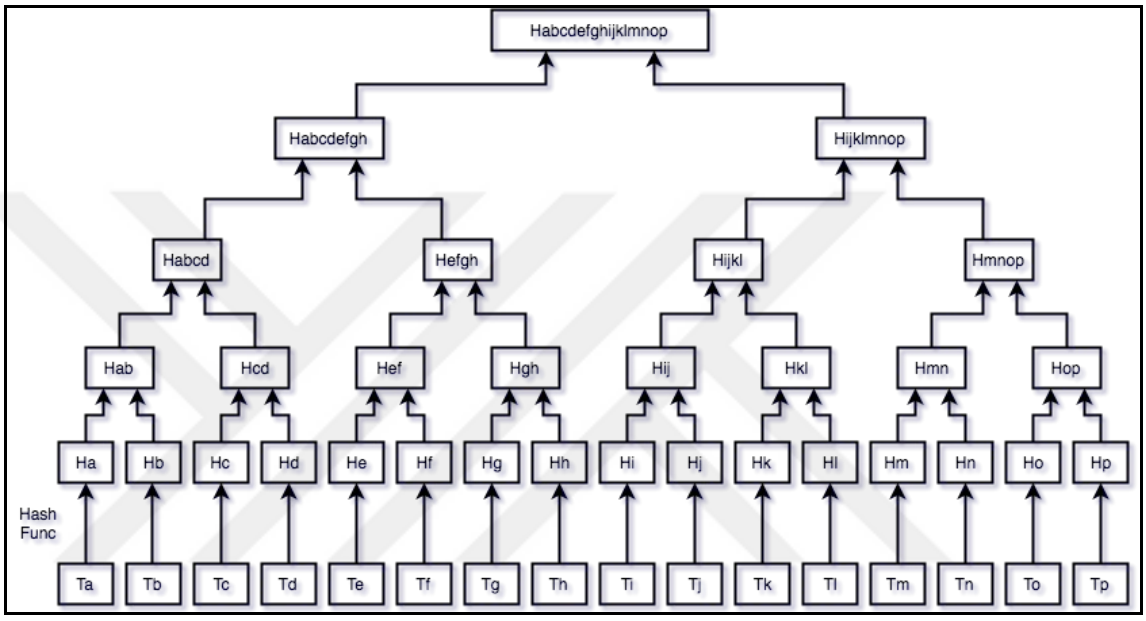
Hash merkle root or hash merkle tree, also known as the hash of hashes like it seen in figure 4.3, is a concept that used to provide data integrity in various fields very often [43]. Data integrity contains -in the meaning- completeness, correctness, and freshness of the data. In lot's of technology or application today, people work with huge sizes of data. This is why data integrity keeps its popularity and moreover getting more and more popular and user population is expanding day by day.

In blockchain protocol, as it mentioned before, it also has a critical importance to have untempered and true transaction history in each client's ledger. In order to get this durable data protection and to have a tamper-proof history record keeping structure, we take advantage of hash merkle root concept.

Hash merkle roots are also another way to represent big size of a list in short format. In blockchain protocol, there are two types of ledgers: a full node which contains all the

transaction history with transaction lists and the other is pruned ledger which does not contain transaction lists but only block headers. By using hash merkle root concept in block header, users who are not miners don't need to keep all the past transaction lists. It is enough to keep only block headers to validate newly created blocks. This also helps to reclaim disk space in user's devices [44].

Figure 4.3: Hash merkle root structure



4.2. DESIGN

It can be found how the system designed by following main principles of a blockchain protocol and DLT in this thesis. Since the main idea is about the distribution of the same transaction history of all, we must be sure that each client has the true ledger.

First of all, we explain our database design. Because of the definition of the distributed ledger, it's important to know what the ledger is. In our study, we used MongoDB as database technology. As it is a collection database, there are 5 collections in a ledger design in the study. These are

i. Blk: The collection keeps the main chain. This collection holds the block header information and the longest PoW chain is consists of the records in this collection.

ii. Blk_trx: The collection keeps the relation between blocks and transactions.

iii. Input: The collection keeps the record of spent transaction outputs as input. When a block published, the outputs used as input in transactions of the block are removed

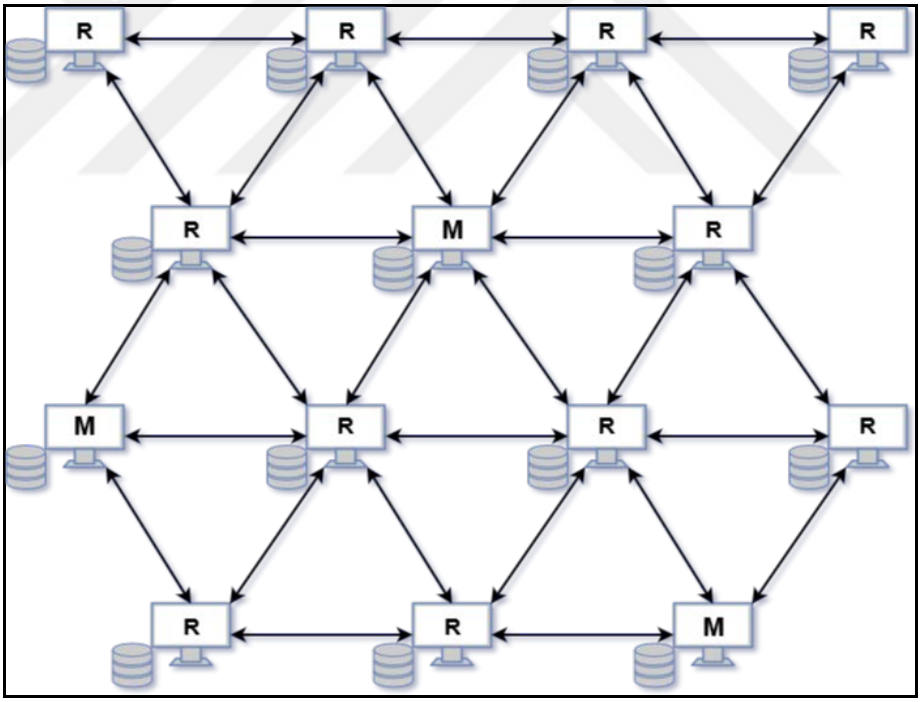
output table and inserted into this collection. The collection can be assumed as dead coins list.

iv. Output: Usually called as UTXO table. For most of the blockchain projects and also for our study, this collection means the expendable coins or cash. If a user creates a transaction and intends to spend his coins, he must refer to the coins assigned to his address in this table.

v. Trx: The collection keeps all the transactions on the network.

Figure 4.1 shows that the protocol consists of a big cluster of the peer machines. There are two kinds of users on the network, miners who take the responsibility of third party organizations over and regular users who do not have any concern but only sending transaction or being a side of any transaction on the network.

Figure 4.4: A distributed ledger structure on a peer to peer network



In figure 4.4, miners are presented with M and regular users presented with R. In PoW based blockchain protocols, the main difference between miners and regular users is the computation power and storage of the machine. Since the PoW concept is based on calculating the intended hash result, miner machines need huge calculation power.

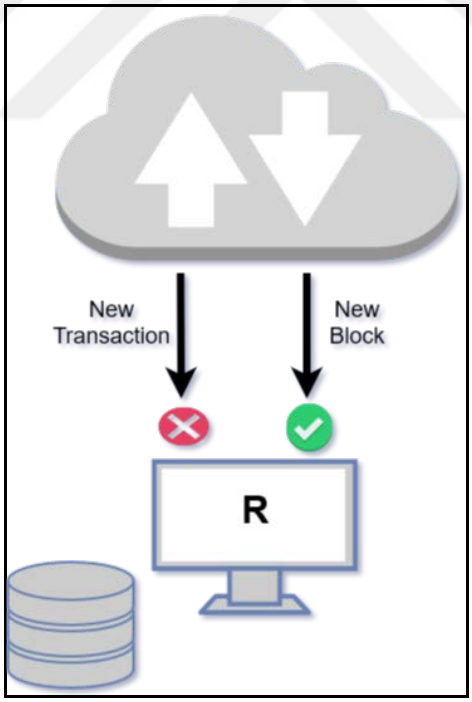
Blockchain protocols are constructed on a very publicly designed network if it's not a private blockchain for a specific organization.

Thanks to the private key encryption and signatures, the transactions are not only anonymous but also very secure and tamper-proof. So that, every client can listen to all the transactions on the network. In our study, regular user application was designed to listen to newly published block messages only with performance concerns.

In the protocol, every client can publish messages to the network freely. When regular user clients able to publish new transaction messages, the miners can publish new transaction and new block message both.

As it seen in figure 4.5, regular users don't care about the transaction messages. In blockchain protocols, every user has to keep its ledger updated always. Because of that, every client is listening new block messages independently if it is a miner or regular user.

Figure 4.5: A regular user's behavior for a different type of incoming messages

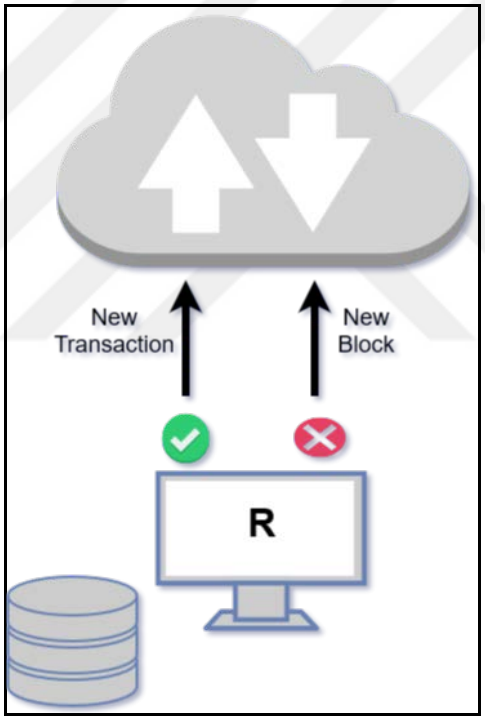


We have a container class in our study which keeps the messages in it. When a new message container arrives at a regular user, the application catches it and checks if it contains a block message or not. When it's a new block message container, the application

gets the block and runs the algorithm which checks if the transactions in the transaction list and the block header is valid to be added to the ledger. If the algorithm returns success, the regular user application adds the block to its own ledger, updates the related tables in its database to keep itself updated.

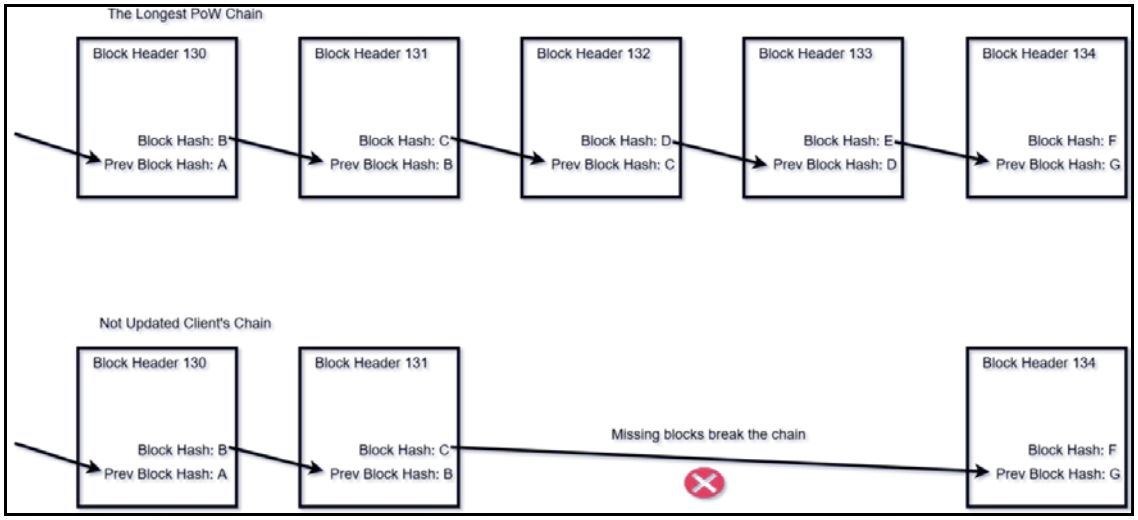
Since a blockchain protocol is closely-coupled to being online, offline clients will not be up to date. Because of this, a client who gets online firstly updates itself by asking the network latest transactions and blocks. Also note that, if a client is not updated, it will suppose every newly published block is invalid.

Figure 4.6: Message types published by a regular user to the network



As is seen in figure 4.6, despite not listening new transaction messages, regular user clients can publish new transaction. On the network, regular users exist only to keep a public and distributed ledger and being a part of transactions. They can be thought as bank account owners in the traditional finance system. Sending and receiving assets on the protocol and keeping the valid chain is their unique roles in the protocol.

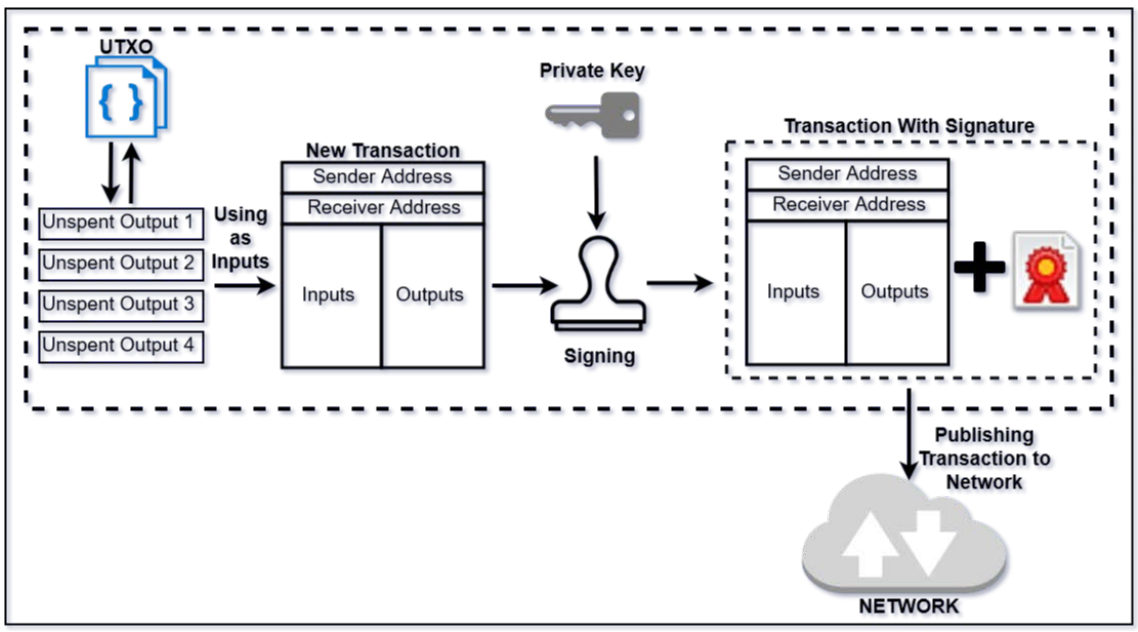
Figure 4.7: How ledgers not up to dated break the chain



In figure 4.7, we can see that how a client who was offline for 2 blocks time and with not updated ledger suppose invalid new valid blocks.

When a regular user application intends to publish a new transaction, first of all, the application checks the UTXO table to find enough amount of output as it seen in figuure 4.8. If there are enough asset assigned to the application address, this means that the user can create this transaction.

Figure 4.8: New transaction creation and publishing process

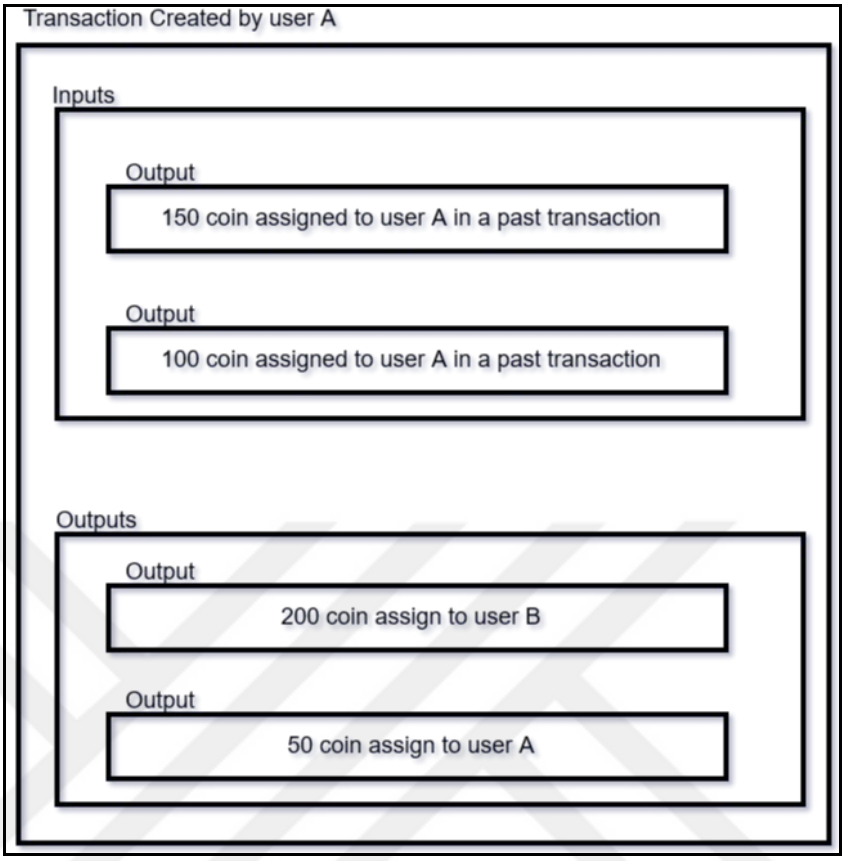


In blockchain protocols, usually mint-based transaction approach is applied. Because of the concerns about the history of the assets or coins which are used as input in a transaction, the system removes the used coins and creates new coins and assigned them to related addresses to use in the future in each transaction. The responsibility of this input and output management is on the transaction owner. As it is shown in figure 4.9, user A is creating a transaction which carries the information of sending 200 coins to user B. There are 150 and 100 coins assigned to address of user A in the UTXO table. The sum of the coins he owned is 250 and he wants to send just 200 of them. So he puts this 2 outputs as input and he asks for creating 2 outputs: 200 coins to assign to user B and 50 coins assign to his own address.

In many protocols, also in our study, a transaction creator usually uses references as input. The application simply refers to a past transaction output like the nth output of mth transaction of the kth block. The miner will find the related block, transaction, and output and check if the referenced output is assigned to the address of user A. If it is, the related validation step will be passed.

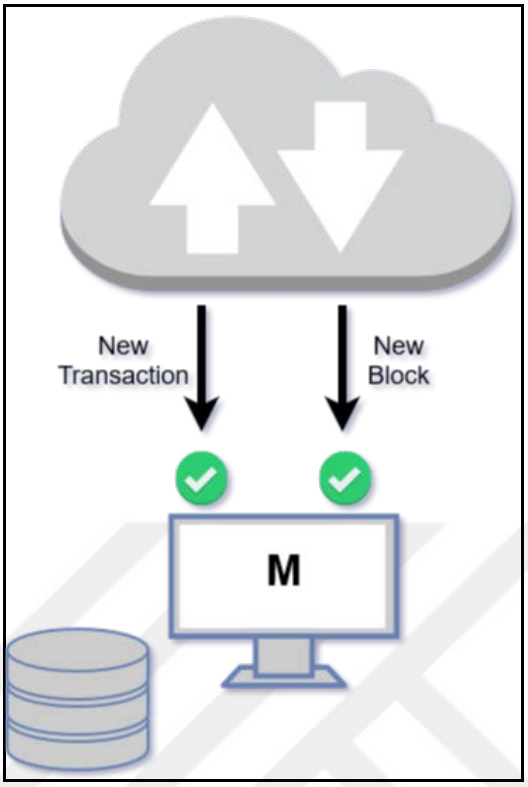
After the input-output management and the creation process finishes in the sender client application, the client has to sign the transaction with his own private key. Transactions without signature will not be processed by the network.

Figure 4.9: A transaction structure with inputs and outputs



Each side of a transaction, have to wait for the related transaction to be a part of a block. If the transaction gets into a block, this means that the transaction was successfully accepted by the network and the cash flow will be ended in another meaning. Also note that, in financial blockchain projects, it is suggested to wait until a count from 3 to 10 blocks on the related transaction's block for security.

Figure 4.10: New incoming messages and a miner

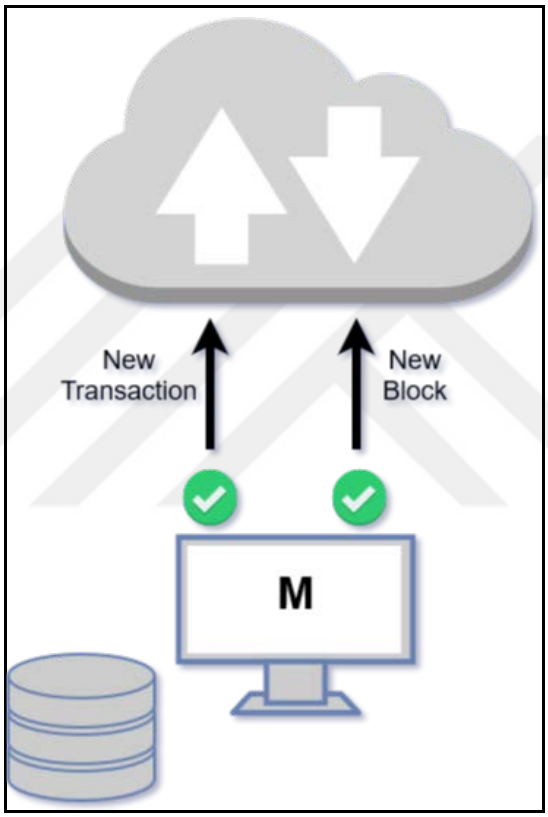


The other type of clients are miners in the network. Miners can be thought as workers in the network who works for the building chain security and transaction validation. This type of clients always needs huge calculation power and storages. Since they are the only validators on the network, they don't have the chance to keep only pruned ledger but all the transaction history.

Miners listen all the new block and new transaction messages on the network like in figure 4.10. When a new transaction message arrives, it adds the message to the list of transactions to be validated. They validate each transaction and adds to the transaction list of a block. When the miner satisfied by the transactions which are planned to be placed in new block, stops the validation and starts the next process of PoW also known as mining. As it seen detailed in figure 4.12, before mining, the inputs of mining process should have been ready. These are valid transactions, prevous block hash which is the hash of the latest block header of the longest PoW chain, current time information and current difficulty index value to use in the mining.

Although they are miners and create blocks, they also can publish transactions like in figure 4.11. Because of the difficulty of the mining process, blockchain protocols were designed on an awarding system. A miner who creates and adds a new block first gets the mining reward in the protocol's currency type. So that, every miner also has to have a wallet and a valid address on the protocol to get the reward to the address. Also when they want to spend their assets or coins, they publish a new transaction to the network.

Figure 4.11: Publishing new message in miner side

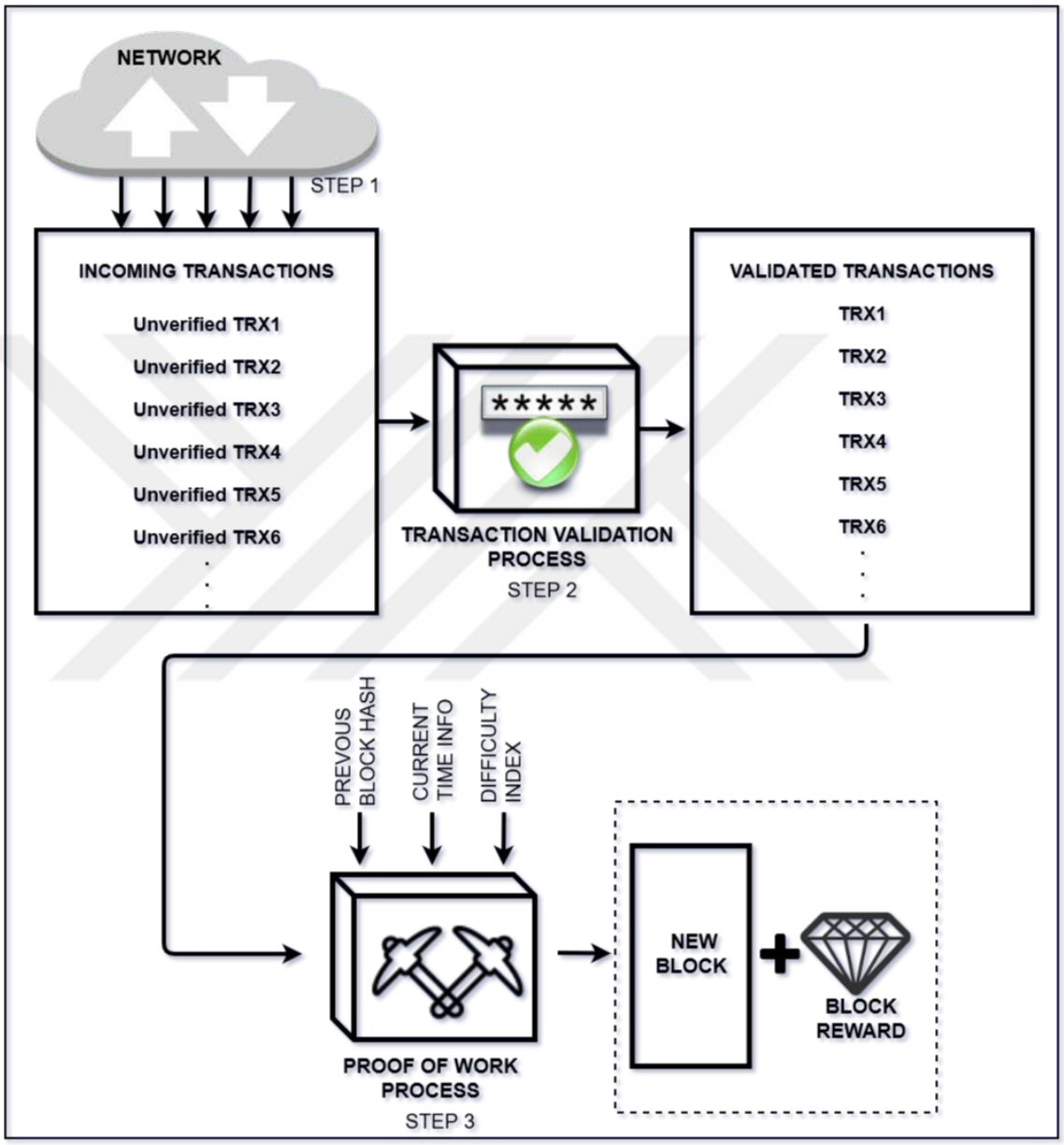


When a miner gets a transaction, firstly checks the signature. The signature algorithm, in our study it is ECDSA, lets users to check signatures if the user has the transaction, senders public key and signature. If the signature is not valid, the transaction will be ignored.

After the signature validation, miner begins the other validation steps such as transaction input and output amount validations, inputs existence validation, transaction owner and assigned input addresses validation etc. In the short expression, if the transaction was signed by the true owner, and the amounts are sufficient to provide transaction, and the

inputs used were assigned to the signature owners' address or public key, and the receiver address is valid, the transaction will be assumed valid.

Figure 4.12: Steps of creation process of a block



When validation step finishes, the mining process starts at the miner application. As it told before, in PoW based blockchain protocols, mining means trying to calculate the targeted hash result. Although calculating hash of an object is not a difficult process, to look for a specific hash result is extremely difficult. Because of this, in bitcoin mining today, miners head to use powerful graphics processor units to calculate the hash result

in parallel processes. In table 4.1, it can be found the GPU performances about Bitcoin mining.

Table 4.1: Different GPU's and their Bitcoin mining performances

BRAND		Mhash/s
NVIDIA	GTX 285	62.1
	GTX 295	99.3
	GTX 460	58.1
	GTX 480	102.6
	GTX 560Ti	67.1
	GTX 580	120.2
	GTX 590	189.4
ATI	HD 4890	105.1
	HD 5750	149.3
	HD 5830	199.2
	HD 5970	514.3
	HD 6850	162.2
	HD 6990	681.9
	HD 5870X2(ARES)	592.9
	HD 5870X2(ARES OC)	823.9

Source: www.pcper.com

In blockchain protocols, miners are in a speed race to get block reward. Since the first publisher of a new block gets the block award, miners have to calculate hashes as fast as it can to provide PoW concept. After the concept has been provided, the miner has to publish the new block as soon as possible.

After creation and publishing of a new block, miner side ends the process. Every client on the network will listen to get new block messages. When any client gets the new block message, checks and validates the transactions in the block, calculates the block header hash with the nonce value which provides the PoW concept. If transactions in the transaction list in the block are valid, and the hash result of the block header with the

nonce value starts with the count of zeros, the clients add this record to their ledger as current block and update the input and output tables in their ledgers by the new blocks transaction list.

In this step, there are two ways to act in client applications. If the client is a regular user application, the application just updates the main chain of blocks, input and output tables. This is pruning process. And if the client is a miner application, besides the update of the chain, input and output tables, it also keeps the transactions. Since miners are main validators on the network, they have to have all the ledger history in their storages.

Note that miner databases are called as a full ledger or full node and regular user databases are called pruned ledger. This is designed by Satoshi to reclaim disk space of the users in the network.

After creating, publishing, mining and adding blocks processes, we can see the system in a general view. Peer-to-peer network provides the all system security against a single point of failure attacks or problems like being offline. The public-private key encryption and signatures provide the transaction security, anonymity, and publicity. The PoW concept and keeping previous block hash result in each block provides the chain security over the transaction security. The combination of these technologies provides the all the protocol security and lets us keeping ledgers distributed and exactly the same with each other.

4.3. USED SOFTWARE TECHNOLOGIES

In this section, the technologies that were used in constructing our basic blockchain protocol are discussed.

4.3.1. C#.Net

In this thesis, we chose .net platform as the main framework partially due to its object-oriented architecture. Also, while deciding on framework, ease of finding needed libraries and third-party tools played big roles.

In the proposed protocol, c# version 4.5 as the main programming language in .net framework was used for the same reasons.

4.3.2. SHA256 Hash Algorithm

Hash algorithms are the most critical issue for the blockchain protocols. In our study, we assumed SHA256 hash algorithm as the main reference point.

.Net framework allows users to use this algorithm easily since it is built in the library mscorlib.dll.

Also, note that bitcoin's blockchain protocol uses the SHA256 hash algorithm as a main hash function but only it applies it as SHA256x2 for the security concerns.

4.3.3. Newtonsoft for .Net

In recent years in the development industry, people started to use a lightweight format for data interchanging named JSON(JavaScript Object Notation). It has various advantages such as ease of use, low message sizes, easily consumed by javascript etc. It is also compatible to .net framework and very easy to use with c#.

In object serialization and communication, also we used JSON format in the study. Also, to provide our proof of work concept, we calculating the hash result of our block object's JSON serialized version.

4.3.4. Babelfor.Net

Another critical part of a blockchain protocol is digital signatures. To provide a public key signature algorithm, we used BabelFor.Net product for free. It enables to use ECDSA to sign and verify our transactions in a very easy and fast way.

4.3.5. MongoDB

As database technology, we decided to choose MongoDB. It is a document database with the high scalability and flexibility to querying and indexing. It also stores JSON-like documents which makes easier to work with JSON objects and compatible with 10+ programming languages including python, java, c++, c# etc. It also enables to run queries very fast on it even it is a large size of the stored data.

4.3.6. SignalR

It's one of the biggest limitation of our study that not having a real P2P network to run our protocol on it. For the communication between our clients, including miners- we used

SignalR real-time messaging library for .net framework. Although, it is basically used for web technologies, we used it for desktop applications with an extension.

It is often used for online real-time messaging. But also easily enables applications to communicate each other in very efficient and easy way. So we simulated our network with SignalR and send and receiver our JSON messages via it.

5. DISCUSSION

In this study, we exemined a financial distributed ledger apllication for payment and money transfer purpose. One of the limitation in this thesis is providing a P2P network. Although it is one of the most critical point in blockchain protocol to provide a true P2P network to remove central authorization and SpoF, there is not new development on P2P network with DLT. Since the essential part of the protocol is storing ledgers in a distributed way, we could achieve this goal by simulating the P2P network connection with signalR communication. Although it is not a real P2P network, it provides a real-time connection between user applications in the network. Each user application stores their own ledger which is the same with the other ones. This approach gives us the main principle of DLT.

Another limitation is concerning the addresses in the protocol. Although it is very difficult to create the same two public keys with the keygen application which based on the algorithm of ECDSA, we didn't implement the control mechanism of the public keys so on the addresses in the protocol. Since addresses are assumed as bank accounts in financial DLT application, it's important the singularity of the addresses.

One of the issues is the main cost function which used to create true blocks and so on to provide the security of the chain. In this study, main cost function is based on the Proof of Work concept which needs high calculation power and energy. Although the biggest DLT application Bitcoin uses this concept, there are still hesitatios and questions about energy consumption. Moreover some assume PoW as waste of energy. To avoid this discussion, there are several alternatives like Proof of Stake, Proof of Burn.

Also another issue is called as "%51 Attack" or "Double Spend Attack". Blockchain protocols are theorically immutable which means the history of records are temper proof. As it's mentioned in its name, if a malicious miner or a group of miners controls more than %50 of the hash power, they can alter incorrect records to the main chain. To solve this problem, the difficulty index of PoW concept must be set correctly. Although this index makes the main chain stronger and trusted, it can also make the transactions slower.

In future work, P2P network limitation can be fixed by creating a real P2P network with TCP/UDP connections between 2 or more machines. And the addresses should be kept in the network to prevent any kind of confusion between addresses in the protocol. Also new

main cost function concepts can be applied to solve energy consumption problems and long transaciton realization time. Another implementation can be difficulty index optimization to improve the chain security. In this study, this index is a constant value. It should be dynamically incremental based on the block creation time.

Also this study includes the pure main fields of a PoW based blockchain protocol. So that, not only projects for financial purposes but also any kind of DLT application can be implemented by refering the protocol studied in this thesis.

6. CONCLUSION

In this study, we investigated the anatomy of a blockchain protocol and implemented a sample blockchain application which provides distributed ledger basics. The steps to track and create a peer to peer payment application on a blockchain protocol are examined. As an application, we proposed an online payment or digital asset transfer environment which does not need to any third-party mediator organization between peers as the other DLT applications. The study is based on the bitcoin's blockchain protocol which keeps the PoW concept in the hearth of the process of creating and inserting blocks to the main chain. Although PoW as the main cost function is based on power consumption, it can easilly be replaced with another concept to save energy and time.

There are several DLT applications developed in different programming languages and database technologies. In this study, we used .net framework with c# for the main implementation and MongoDB for database technology to create our blockchain protocol. From this point of view in this study, it is shown that it is possible to implement a blockchain protocol or a DLT application with local implementation opportunities to compete other DLT applications which are commonly used in today's technology world.

Presented methodologies and design were inspired by bitcoin's blockchain which uses PoW methodology as the main cost function. This study also shows that, it is quite possible that to create a blockchain protocol and implement a DLT application on it is possible by following main DLT principles.

Although our study is a financial application, it also can be helpful for other industries which need to remove third-party organizations, tamper-proof data keeping, data transfer in secure ways or public transactions with anonymity.

REFERENCES

[1] S. Nakamoto, «Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system,» 2008.

[2] R. Churchhouse, «Codes and ciphers,» *New York, New York: Cambridge UP,* 2002.

[3] A. Kahate, Cryptography and network security, Tata McGraw-Hill Education, 2013.

[4] A. Rowstron ve P. Druschel, «Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems,» %1 içinde *IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*, 2001, pp. 329-350.

[5] E. Adar ve B. A. Huberman, «Free riding on Gnutella,» *First monday,* cilt 5, 2000.

[6] M. Ripeanu, «Peer-to-peer architecture case study: Gnutella network,» %1 içinde *Peer-to-Peer Computing, 2001. Proceedings. First International Conference on*, 2001, pp. 99-100.

[7] B. Cohen, «Incentives Build Robustness in BitTorrent,» %1 içinde *Workshop on Economics of Peer-to-Peer systems*, 2003, pp. 68-72.

[8] A. R. Bharambe, C. Herley ve V. N. Padmanabhan, «Analyzing and improving bittorrent performance,» *Microsoft Research, Microsoft Corporation One Microsoft Way Redmond, WA,* cilt 98052, pp. 2005-03, 2005.

[9] Y.-F. Chen, Y. Huang, R. Jana, H. Jiang, M. Rabinovich, B. Wei ve Z. Xiao, «When is P2P technology beneficial for IPTV services,» %1 içinde *Proceedings of the 17th International Workshop on Network and Operating System Support for Digital Audio and Video*, 2007, pp. 04-05.

[10] S. Douglas, E. Tanin, A. Harwood ve S. Karunasekera, «Enabling massively multi-player online gaming applications on a p2p architecture,» %1 içinde *Proceedings of the IEEE international conference on information and automation*, 2005, pp. 7-12.

[11] J. Seedorf, S. Kiesel ve M. Stiemerling, «Traffic localization for P2P-applications: The ALTO approach,» %1 içinde *Peer-to-Peer Computing, 2009. P2P'09. IEEE Ninth International Conference on*, 2009, pp. 171-177.

[12] Mills, D. C. a. Wang, K. a. Malone, B. a. Ravi, A. a. Marquardt, J. C. a. Badev, A. I. a. Brezinski, T. a. Fahy, L. a. Liao, K. a. Kargenian ve V. a. others, «Distributed ledger technology in payments, clearing, and settlement,» 2016.

[13] C. Cachin, Architecture of the Hyperledger blockchain fabric, IBM, 2016.

[14] M. Atzori, «Blockchain technology and decentralized governance: Is the state still necessary?,» 2015.

[15] S. Davidson, P. D. Filippi ve J. Potts, «Disrupting Governance: The New Institutional Economics of Distributed Ledger Technology,» 2016.

[16] K. Peterson, R. Deeduvanu, P. Kanjamala ve K. Boles, «A blockchain-based approach to health information exchange networks,» *Proc. NIST Workshop Blockchain Healthcare,* cilt 1, pp. 1-10, 2016.

[17] WePower, «WePower,» WePower, 2018. [Çevrimiçi]. Available: https://wepower.network/#. [Erişildi: 25 04 2018].

[18] K. Korpela, J. Hallikas ve T. Dahlberg, «Digital supply chain transformation toward blockchain integration,» %1 içinde *50th Hawaii international conference on system sciences*, 2017, 2017.

[19] Y.-P. Lin, J. R. Petway, J. Anthony, H. Mukhtar, S.-W. Liao, C.-F. Chou ve Y.-F. Ho, «Blockchain: The Evolutionary Next Step for ICT E-Agriculture,» *Environments,* cilt 4, p. 50, 2017.

[20] A. Dorri, S. S. Kanhere ve R. Jurdak, «Blockchain in internet of things: challenges and solutions.,» 2016.

[21] J. Sun, J. Yan ve m. a. Z. K. Zhang, «Blockchain-based sharing services: What blockchain technology can contribute to smart cities,» *Financial Innovation,* cilt 2, p. 26, 2016.

[22] O. Jacobovitz, «Blockchain for identity management.,» The Lynne and William Frankel for COmputer Science Department of Computer Science, Ben-Gurion University, Beer Sheva, 2016.

[23] A. Arrendondo, «Arredondo, AbelardoBlockchain and certificate authority cryptography for an asynchronous on-line public notary system,» The University of Texas, 2018.

[24] M. Bellare ve P. Rogaway, «Introduction to modern cryptography,» *Ucsd Cse,* cilt 207, p. 207, 2005.

[25] D. Gligoroski, R. S. Ødegård, R. Jensen, L. Perret, J.-C. Faugère, S. J. Knapskog ve S. Markovski, «MQQ-SIG: an ultra-fast and provably CMA resistant digital signature scheme,» %1 içinde *Lect Notes Comput Sci*, 2011, pp. 184-203.

[26] A. Nath, S. Ghosh ve M. A. Mallick, «Symmetric Key Cryptography Using Random Key Generator. In Security and Management,» %1 içinde *2010 International Conference on Security & Management*, Las Vegas, 2010.

[27] J. Thakur ve N. Kumar, «DES, AES and Blowfish: Symmetric key cryptography algorithms simulation based performance analysis,» *International journal of emerging technology and advanced engineering,* pp. 6-12, 2011.

[28] M. Abomhara, O. O. Khalifa, O. Zakaria, A. Zaidan, B. Zaidan ve H. O. Alanazi, «Suitability of Using Symmetric Key to Secure Multimedia Data: An Overview,» *Journal of Applied Sciences(Faisalabad),* cilt 15, p. 2010, 2010.

[29] M. Bellare ve P. Rogaway, «The exact security of digital signatures-How to sign with RSA and Rabin,» %1 içinde *International Conference on the Theory and Applications of Cryptographic Techniques*, Berlin, 1996.

[30] E. Milanov, «The RSA algorithm,» *RSA Laboratories,* 2009.

[31] M. Adalier, «Efficient and Secure Elliptic Curve Cryptography Implementation of Curve P-256,» %1 içinde *Workshop on Elliptic Curve Cryptography Standards*, 2015.

[32] D. Johnson, A. Menezes ve S. Vanstone, «The elliptic curve digital signature algorithm (ECDSA),» *International journal of information security,* cilt 1, pp. 36-63., 2001.

[33] R. F. Rights, «Global Information Assurance Certification Paper,» 2003.

[34] M. Bawa, B. F. Cooper, A. Crespo, N. Daswani, P. Ganesan, H. Garcia-Molina ve P. Vinograd, «Peer-to-peer research at Stanford,» *ACM SIGMOD Record,* cilt 32, pp. 23-28, 2003.

[35] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek ve H. Balakrishnan, «Chord: a scalable peer-to-peer lookup protocol for internet applications.,» *IEEE/ACM Transactions on Networking,* cilt 11, pp. 17-32, 2003.

[36] M. Paksoy ve J. Prado, «Comparing centralized and decentralized distributed execution systems.,» 2006.

[37] A. Montresor, «Decentralized Network Analysis: a Proposal,» %1 içinde *Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2008. WETICE'08. IEEE 17th*, IEEE, 2008, pp. 111-114.

[38] «www.ibm.com,» ibm, 1 1 2014. [Çevrimiçi]. Available: https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.1.0/com.ibm.zos.v2r1.ieaf100/single.htm. [Erişildi: 21 4 2018].

[39] K. Ranjithprabhu ve D. Sasirega, «Eliminating single point of failure and data loss in cloud computing,» *International Journal of Science and Research,* cilt 3, pp. 2319-7064, 2014.

[40] D. F. Phillips, «Classic Single Point Failures of Redundant DP Systems,» %1 içinde *Dynamic Positioning Conference*, 1998.

[41] A. Bonnecaze, P. Liardet, A. Gabillon ve K. Blibech, «Secure time-stamping schemes: A distributed point of view,» *Annales des télécommunications,* pp. 662-681, 2006.

[42] T. J. Gronberg, D. W. Jansen, L. L. Taylor ve T. K. Booker, «School outcomes and school costs: A technical supplement,» *Texas Joint Select Committee on Public School Finance,* 2005.

[43] M. S. Niaz ve G. Saake, «Merkle Hash Tree based Techniques for Data Integrity of Outsourced Data,» *GvD,* pp. 66-71, 2015.

[44] P.-N. Tan, M. Steinbach ve V. Kumar, «Association analysis: basic concepts and algorithms,» *Introduction to Data mining,* pp. 327-414, 2005.