Ministry of Higher Education
And Scientific Research
University of Diyala
College of Science
Department of
Computer Science

Design of Authorization Technique in Simulation Environment Blockchain

A Thesis

Submitted to College of Science/ University of Diyala in a Partial Fulfillment of the requirements for the Degree of Master in computer Science

By

Wasan Ahmed Ali

Supervised by

Naji M. Sahib
Professor

Dr.Jumana Waleed
Assistant Professor

2020 A.D. **1441 A.H.**

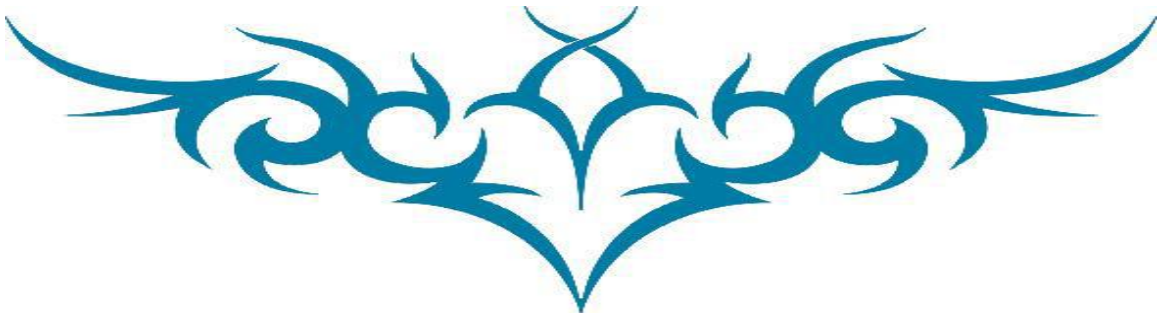
بسم الله الرحمن الرحيم

﴿ وَلِيَعْلَمَ الَّذِينَ أُوتُوا الْعِلْمَ أَنَّهُ الْحَقُّ مِن رَّبِّكَ فَيُؤْمِنُوا بِهِ فَتُخْبِتَ لَهُ قُلُوبُهُمْ وَإِنَّ اللَّهَ لَهَادِ الَّذِينَ آمَنُوا إِلَى صِرَاطٍ مُّسْتَقِيمٍ ﴾

صدق الله العظيم

سورة الحج

اية (54)



Dedication

And he has the first and last credit for lighting my way..... God Almighty

To those who said among them, "And lower them to the wing of humiliation of mercy, and say, My Lord, (24) have mercy on them as my little Lord" (Al-Israa)

".They are buried underneath. "Mom ... and Dad

To those who gave me a lot of giving and motivation ... ... my brothers ... and my sisters.

To those who foad my rhetoric and my life..... "My family".

To the sincere hands that helped me and shared me in fatigue and ignited the light of hope and will ..... my dear teachers.

Wasan Ahmed Ali

Publication Papers

Wasan A.Ali, Naji M. Sahib, Jumana Waleed "The Preservation of authentication and authorization on the blockchain" "2019 2nd international Iraqi conference on engineering technology and its Applications (2nd IICETA)", Al-Najef, Iraq, 2019, pp, 83-88.

Acknowledgments

First and last, I thank God Almighty, who has helped me to finish this study first. And I extend my sincere thanks and gratitude to everyone who helped me and lend a hand to me. In the forefront of whom is the professor, the distinguished professor, "***Naji Matar Sahib"*** and Dr. "***Jumana Waleed*** " Who supervised this research, and their good guidance, valuable comments, and decent treatment have had a great impact on the research reaching this image.....

I also extend my sincere thanks and appreciation to all those who contributed and helped in the success and completion of this study of teachers and professors...

And to the department staff and all of my colleagues who did not strive to help me during the period of completing this study.

Wasan Ahmed. Ali

Linguistic Certification

This is to certify that this research entitled ***"Design of Authorization Technique in Simulation Environment Blockchain"*** *was prepared my linguistic supervision. It was amended to meet the style of English language.*

Signature:

Name: *Asst. Prof. Dr. Salam A. Noman.*

Date: / /2020

Scientific Certification

This is to certify that this research entitled ***"Design of Authorization Technique in Simulation Environment Blockchain"*** *was prepared my scientific supervision. It was amended to meet the style of scientific formula.*

Signature:

Name: *Asst. prof. Dr. Nada H. Mohamed Ali.*

Date: / /2020

Supervisor's Certification

We certify that this research entitled ***"Design of Authorization Technique in Simulation Environment Blockchain"*** was prepared by (***Wasan Ahmed Ali)*** under our supervisions at the University of Diyala collage of Science Department of Computer Science, as a partial fulfillment of the requirements needed to award the degree of Master of Science in Computer Science.

(Supervisor) **(Supervisor)**

Signature: **Signature:**

Name: ***Prof. Naji M. Suhaib***. **Name:** ***Asst. Prof. Dr. Jumana W. Salih***

Date: / /2020 **Date: / /2020**

Approved by University of a Diyala collage of Science Department of Computer Science.

Signature:

Name: *Assist. Prof. Dr. Taha M. Hassan.*

Date: / / 2020.

(Head of Computer Science Department)

Examination Committee Certification

We certify that we have read this research entitled "***Design of Authorization Technique in Simulation Environment Blockchain***", and as an examining committee, examined the student "***Wasan Ahmed Ali"*** in its contents and that in our opinion, it is adequate as fulfill the requirements for the Degree of Master in Computer Science at the Computer Science Department, University of Diyala.

(Chairman)

Signature:

Name: ***Prof. Dr. Ziyad T. Mustafa***

Date: / /2020

(Member)

Signature:

Name: ***Prof Dr. Taha M. Hassan***

Date: / /2020

(Member)

Signature:

Name: ***Asst. Prof. Ahmed Salih Ahmed***

Date: / /2020

(Member/ Supervisor)

Signature:

Name: ***Prof. Naji M. Suhaib*.**

Date: / /2020

(Member / Supervisor)

Signature:

Name: ***Asst. Prof. Dr. Jumana W. Salih***

Date: / /2020

Approved by the Dean of College of Science, University of Diyala.

(The Dean)

Signature:

Name: *Prof. Dr. Tahssen Hussein Mubarak*

Date: / /2020

Abstract

The massive growth of data and all applications used on networks requires great security and safety. Blockchain technology is a static and shared database that is not controlled by any third party. Blockchain technology can be combined with a variety of other technologies as it enters the digital, physical, and biological fields. Also; Authentication is an issue that needs to be thoroughly verified to be authenticated regardless of the traditional authentication methods used to prove that the person is authorized on it. In this thesis design Blockchain system is proposed to simulate each node in the system. The propose system of design Authorization technique in simulation environment blockchain named (ASBchain) consists of six stages to verify the transactions transmitted by user after the registration process, which based on the strong Rivest Shamir Adleman algorithm(RSA) for signature transaction and Secure Hash Algorithm 256. Then verified from any transaction performed based on matching hash function values that sending .Thus, the proposed system can prove that the sender is authorize by authorization process. This is done according to the value of the last hash function for block maintained by this sender based on the time stamp of it. The system was tested in terms of time for each stage and the phases were compared with each other and show that the time spent on Registration ,authentication, and Authorization processes were (00:01:43:0059 s), (00:01:02.0953 s) and (00:01:00.0102 s) respectively for 100 users. The system has proven that all people have equal rights in reliability and use the system. But not every person is authenticate do he is authorized. It is a collaborative environment and the main thing is reliability, safety, decentralization.

List of Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

$\oplus$	XOR Gate.
ASBchain	Authorization Simulation on Blockchain.
Ethash	Etherum Algorithm.
H	Hash value.
HMT	Hash Merkle Tree.
ID	Identifier.
LCG	Linear Congruential Generators.
NewF	New Moment Feature.
NH	New hash
P2P	Peer 2 Peer network.
POS	Proof of Stake.
POW	Proof of Work.
Prev-hash	Previous Hash.
RSA	Rivest, Shamir and Adleman.
SHA256	Secure Hash Algorithm 256.
STR	Sender to Receiver.
T	Transaction.
M	Moment Feature.

Chapter One

Introduction

Chapter One

Introduction

1.1 Overview

The new technologies such as video and voice calls, pictures, emails, and messages permit individuals to communicate directly. These technologies are used to travel directly from the transmitter to the recipient through the internet with keeping the trustworthy between individuals no matter how far apart they are. Nevertheless, if it related to money, individuals should trust a third party to be capable of completing the transaction [1].

So in order to create a digital identity, the users should have to register at the server. Here the users must supply personal sensitive data, user-name, email, phone number, and the details of a credit card. These data are kept on the centralized server across data multicenter. Also; the users must create multiple-identities across multiple-suppliers to access their services. Studies have shown that this procedure of creating multi-identities is cumbersome and inconvenient since the users must be repeated the same process of registration many times and remember the passwords for various services. But these data are vulnerabilities to attacking, and the centralized servers of the suppliers are targets for hackers as primly [2].

Blockchain technology is a relatively new approach to information technology. The first application of blockchain technology is bitcoin which is used in financial exchange [3]. Blockchain technology was adverted in 2008 by Satoshi Nakamoto’s white paper [4]. The work of

Satoshi Nakamto present a solution to the issues which implement and use digital currency, particularly, the double spending issue [3].

Blockchain gives an open decentralized database to any transaction including value like goods, and money. Consequently, the technology of blockchain has been started to slowly invade the internet as a guaranteed substitutional digital model by utilizing cryptography and mathematics [1].

The technology of blockchain has several basic properties; decentralization, transparency, shared ledger based on consensus, immutability, and privacy. Here, it can realize the needed features for authentication and authorization such as secure, decentralized, anonymity [5].

Although the basic features of blockchain that may bring us more reliable, secure, and convenient services, The security problems and challenges of this innovative technique is also a necessary topical that we need to concern it [6].

For a user to join the permission blockchain, there is the need for membership authentication. The researchers have been worked on improving privacy during authentication through depending authentication on attributes, instead of identities. It is confirmation of identity which the entity claims utilizing credentials [7]. Authentication systems are used biometric characteristics that are unique for each entity. The basic feature of the biometrics is that the entity has always with a way to authenticate himself. For example, you can forget a password or may be stolen an access card that you have. But in reality, you cannot forget your fingerprint, your signature, your gait. Biometric is more process as to remember several passwords for the user. So it can be used

to verify the identity of the person because these characteristics are unique for each user. Also, it is difficult to restore their production and it is impossible to exchange it [6]. The process of authentication is used to determine the user's validity who supposed to be. The process of authorization is to determine which resources the user is allowed to access. Authorization determines the permissions of users in concepts of access to and use of resources to create some actions such as append, update, delete, etc. but not every person is right authenticated is authorizer , this via according to some rules based on it [8].

The Technology of Blockchain could be described as a public ledger and every transaction is stored in a block as a list of transactions [9].Figure (1.1) explains the basic block diagram of Blockchain which will explain the components of the block in detail in the next chapter. [10].

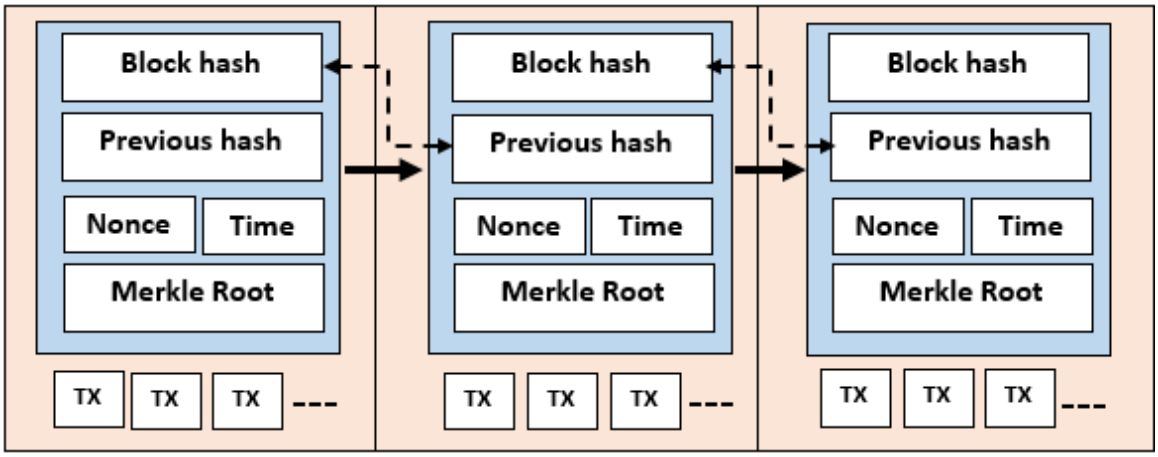


Figure (1.1): A basic block diagram of Blockchain [10].

Therefore, this thesis concentrates on how to prove that a person is authenticated by all of the network and without relying on a central party in a safe and transparent manner, and how to prove that he is authorized

on the network. So here will design and implement system work on blockchain technology called ASBchain system based on secure hash algorithm (SHA256) and strong cryptography RSA Algorithm. It is consisting of three stages Registration, Authentication, Authorization, with each stage have several steps to reach the desired goal. This thesis is dealing with a prototype system. The main objective of the thesis is to understand how blockchain works and how can prove sender is authenticated and authorized consecutively without depending on third party.

1.2 Related Work

Several previous studies suggested by numerous investigators about the Blockchain network. Several studies and researchers that can related their works to the suggested scheme in this thesis:-

- In (2017), Chen, Z., & Zhu, Y. [5] showed how the Personal Archive Service System using Blockchain Technology works and what's different between traditional Third-party verification agencies. Their main contribution is to present a framework of utilizing the technology of Blockchain to exploit its desirable features for building a personal archive that associated certifications. There is no need for Inquisitors. A subject is capable of deciding what to reveal to when and who depending on the request nature.
- In (2017), Grech, A., & Camilleri, A. F. [12]. Proposed a technique that can access to a student's personal information by using blockchain via using biometric identification on a smartphone. Every service from these services would be capable of identifying the student with no requirement for asking for or storing any private data again. Meaning that the right student is the only one who can

hold data. Also, the organization does not need to run complicated systems for accessing rights. It just requires securing the network or device where the verifications initial verification is performed. This would provide considerable resources spent in hardening the network contra data breaches.

- In (2017), Kikitamara, et al [13]. Analyzed the utilization of blockchain in digital identity as one of the steps for building an open model system. The digital identity and Blockchain introduce a system for preserving people's credentials for their public services. The management of digital identity combined with the technology of blockchain delivers decentralized online identities. Whereat Blockchain's implementation in the management of a digital identity leads to different properties (entity, attribute ...) especially needed on the authentication technique. And they used a handshake mechanism that includes procedures involving a public key infrastructure PKI verification mechanism.
- In (2017) Xia, Qi, et al. [14]. They proposed a blockchain-based data-sharing framework that adequately addresses the access control challenges associated with sensitive data stored in the cloud using the static characteristics and independence built into the blockchain. Their system is based on an authorized blockchain that allows access only to invited users, and thus authorized users. As all users are already known and a record of their actions is kept by the blockchain. The system allows users to request data from the shared pool after which their identities and encryption keys are verified.

- In (2017), Moinet, et al. [15] proposed an application for the blockchain as secured decentralized storage for cryptographic keys, in addition to trust information in the independent Wireless Sensor Networks concept. The authors showed how The Blockchain Authentication and Trust module and the human-like knowledge-based trust model demonstrate how to use blockchain persistence to provide solutions to high-level issues in the decentralized ad hoc networking space. More accurate, they showed the capability of building solutions supplying mechanisms of authentication, in addition to trust evaluation in an evaluative and self-organized network.
- In (2017), Hammudoglu, J. S., et al [16]. They have created a biometric mobile authentication system that relies only on local processing, as their open-source Android solution explores the ability of current smartphones to acquire, process and match fingerprints using only their embedded devices. Independently, it does not require any cloud service, server, or authorized access to fingerprint readers. It includes three main stages, obtaining fingerprints, obtaining fine detail features and matching with other fingerprints stored locally, and this made them able to capture and process a fingerprint in a matter of seconds using blockchain technology. This work is specifically designed to be the building block for a self-governing identity solution and integration with the unauthorized blockchain for identity proof and key certification.
- In (2018), Gao, et al.[17].They proposed a system that verifies the original data stored on a blockchain network that reflects the actual reliability of the data, in particular the information provided by the

persons involved in the exchange of the goods. They proposed the BlockID system, which provides a framework that verifies the ID issued by the government institution in a digital certificate, through user authentication based on biometrics, which is also associated with the smartphone. They have analyzed security in their BlockID system and have shown that it meets the purpose of confidentiality and safety but the system cannot authorize someone to access certain network sources to do some operations that a person wants to do.

- In (2018) Yin, Wei, et al. [18] They proposed a new blockchain signature authentication scheme, which differs from the elliptical signature scheme in current blockchain technology, in that it can withstand a quantum algorithm attack in the future. Moreover, their scheme realizes the security that cannot be tampered with under the chosen message attack. Their signature security can reduce a difficult SIS issue on the network. Their work has important theoretical significance and provides new thinking to design and develop counter-blockchain technology in the coming decades, but is authentication sufficient to authorize the user to access network resources? This is what their research lacks
- In (2019), Huh, Jun-Ho, and Kyungryong Seo [11].In their research, they have come up with a fingerprint-based entry pad based on the technology of blockchain. Where they designed and implemented the registration system to enter automatically and securely using smart phones. Their focus is on using the most secure authentication methods - fingerprints that provide safety opportunities and ensure personal information and vital sensitive

data. But their search is void of authorizing the user and authorizing him to access certain sources.

- In (2019), Pawade, Dipti, et al. [19]. In their paper, they designed the system to demonstrate the important advantages of safe storage in blockchain and non-static biometric technology. In their search, new technology was introduced and implemented using blockchain technology to protect biometric data. In this system, to extract features, dynamic data is stored permanently and then obliterated from the system. Additionally, biometric data was kept in vector method characteristics on the blockchain that was fragmented. Hence, will prevent tampering with biometric data, construction the system with protection. Agreeing to the results that are experimental, their accuracy of the system is "82.55%" and the rate of the error is "17.48%".but in their system did not address the process of authorization, but they were satisfied with the status of authentication only on the network of the blockchain.

1.3 Problem Statement

The main problem in this work is the privacy of information from manipulation on networks and because each user has a public and private key and to be able share public keys with other users on the network and prove their authentication without interference from any third party and tampering with data, and how can prevent centralize and self-control. And to protect sensitive user data during Sent across multiple servers without controlling it.

1.4 Aim of the Thesis

The thesis main goal is to design system to users as independent access to network without third-party and to enhancement asymmetric cryptography (signature via RSA) based on big-integer. And design and implementing a way to be more reliable to verify and compare user data based on hash function value, and how reliable and consistent database can be used at a later date. Also implement a way for using on the blockchain to prove authorize the user to access sources to perform some operations for creating and linking blocks on blockchain.

1.5 Thesis Organization

The rest of the thesis chapters are clarified as follow:

Chapter Two: Theoretical Background

This chapter provides a background and overview of blockchain network technology. Architecture and how it works, how the process can be validated, and approach of authentication based on fingerprint and authorization and some of the algorithms used.

Chapter Three: "The Proposed System"

This aim of chapter clarifies and explains the suggested ASBchain System design and its execution.

Chapter Four: Results and Evaluation of the Experimental

This chapter clarifies the outcomes and analysis that have been receiving from the suggested system.

Chapter Five: Conclusions and Suggestions for Future work

This chapter produce work conclusions. Additionally, it produces future work proposals.

Chapter Two

Theoretical Background

Chapter Two

Theoretical Background

This chapter contains an overview of Blockchain technology and describes some basic terminologies used in Blockchain technology. And some algorithms which have been implemented in working.

2.1 Blockchain Technology

The basic ideas behind Blockchain appeared at 1991 when a signed series of data as digitally signing by using as an electronic ledger for files in a way that might simply display none of the documents that are signed in the group had been altered [18]. Blockchain technology was adverted in 2008 by Satoshi Nakamoto's white paper [4].

Blockchain and Bitcoin Technology as explained by Nakamoto solved the most important problems of computer science that represent a barrier to an effective digital pecuniary system for years: the problems of double spending. Double spending problem is that fund must be only once spent, unlike a file, which can be copied several times randomly [20].

The Blockchain is distributed ledger shared via everyone participants based consensus protocol in the network of the Blockchain in which be most of the participants agree on the result [21]. Blockchain keeps a non-stop growing records list, named blocks. Every block includes transactions list and connects to the former generated block, up to the first block, called genesis block. The mining process appends a block and verifies the validity of transactions (avoid double spending) via a Proof of

Stake (POS), or other consensus protocols, like Proof of Work (POW) [2].

New blocks are created via a process called mining via several nodes, called miners. These miners operate anonymously by working jointly and attempting to solve mathematical puzzles, which generates new blocks to the Blockchain. It takes several steps to construct and announce a new block. [22].

The ledger isn't owned by any central servers or central authority. Instead of it is distributed to computers (peers) on the decentralized network [2]. Also; the Blockchain enables each user to be pseudonymous, which means the user is unknown but the user account is not all their transactions are noticeable public [23]. Its basic features are [6]:

1- Decentralized: - The core characteristic of the Blockchain, i.e. Blockchain no needs to rely on a central node at any time, where data can be stored, recorded, and updated by distribution [6].
2- Transparent: - The data is recorded via the system of Blockchain and it is transparent to the nodes, also it is transparent when updating the data, this will have led to the reason of being Blockchain is trusted [6].
3- Open Source: - The most systems of Blockchain are open to each node, the record can be verified publicly and also users can utilize the technologies of Blockchain for creating applications [6].
4- Autonomy: - Each node on the Blockchain system can safely update data or transmit, Because of they are based on consensus, the basic idea is a single person to trust to the entire system, and no one can intervene it [6].

5- Immutable: - Any records will be saved forever, and can't be altered unless certain node can have control more than fifty-one % nodes at the same time [6].

6- Anonymity: - The technologies of Blockchain addressed the issue of trust among the nodes, therefore, transaction or data transfer can be anonymous, only the address of the person on the Blockchain is needed to know [6].

Blockchain is considered as a distributed peer layer to peer net executed on the internet, as illustrated in Figure (2.1) [24].

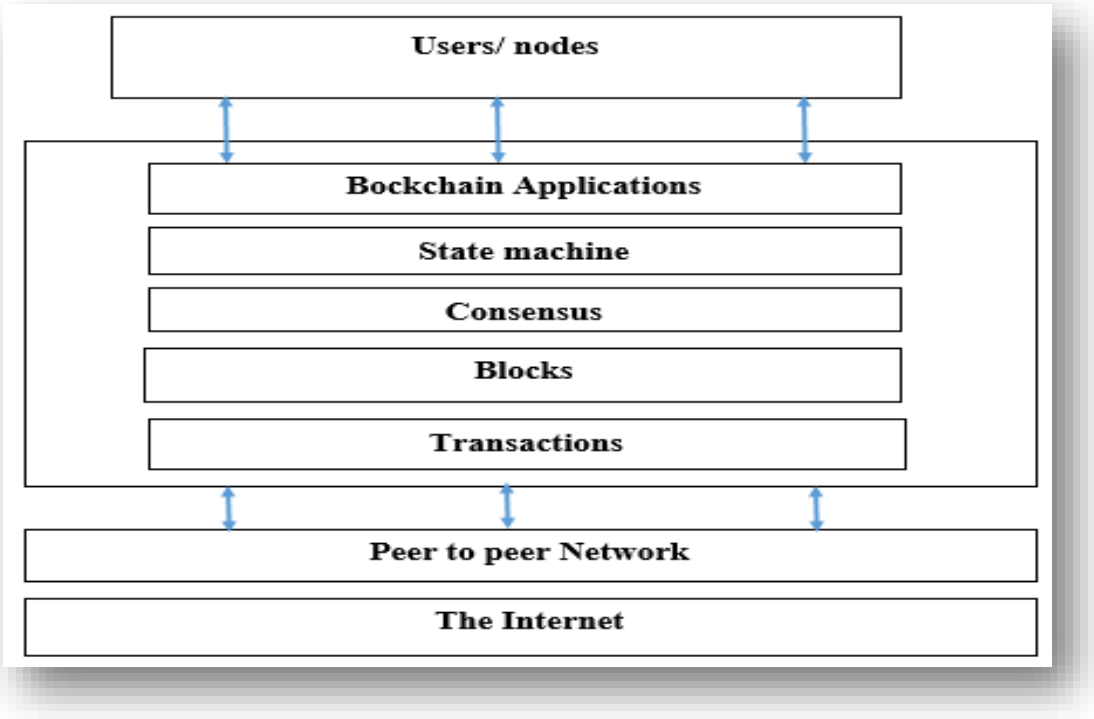


Figure (2.1): Network view of a Blockchain [24].

2.1.1 Peer to Peer (P2P) Network

It is a topology of a network when the whole peers can be connected in addition to transmitting and receiving messages [24]. It is a computer network dependent on nodes, (e.g. computers that are preserving the network worldwide). P2P is a decentralized network where each node shares information with another without anybody controlling the network [22].

The Blockchain relies on more than thousands of nodes in the P2P network, and at each node, the data is updated and replicated. Even in case the nodes become inaccessible or drop it from the network, this network as an entire persist to work, therefore, it becomes highly available [24].

There are two kinds of peers in the P2P network: validator and member peers. The validator peers (represents the special peers) are consuming the services of Blockchain besides validating and verifying the new Blockchain transactions. While member peers are consuming Blockchain services, and every miner holds exactly the same transactions history over the network and comprises a certain responsibility for maintaining and publishing the new transaction blocks to the network [2].

2.1.2 Block

It is a block building unite of Blockchain. It is consisting of set of a transactions with Meta data [2], see figure (2.2) [25]. The block involves two part is the block header and the block body [9].

- Block Header: it includes the following:

  a) Block version: indicates which tuple of block validation rules to be applied.

b) Merkle tree root hash: is the value of hash to the entire block transactions.

c) Timestamp: is the present time at seconds in universal time.

d) N-Bits: target threshold of a valid block hash**.**

e) Nonce: Four-byte field, it usually starts with zeros (0) and increases for each hash calculation.

f) Parent block hash: a 256-bit hash value which indicates to the former block.

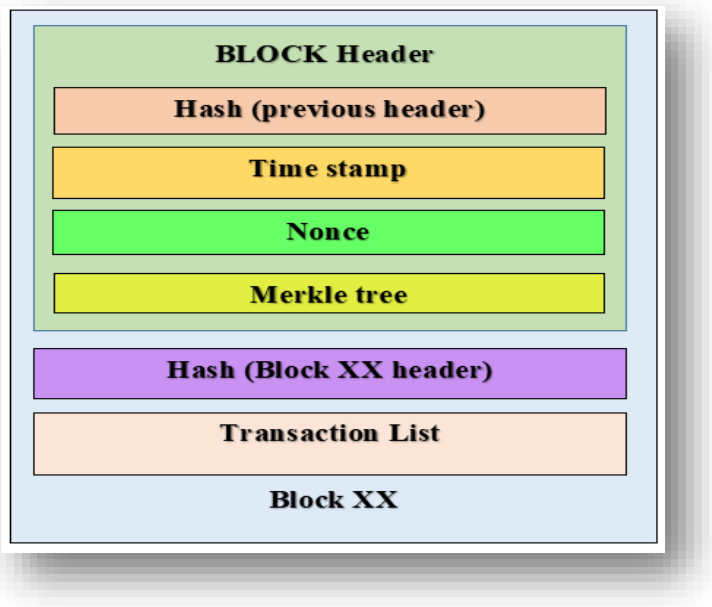


Figure (2.2): Block structure (Generalized) [9].

The block body consists of transactions and a transaction counter. The total number of transactions that can a block contain it relies on the block size and the size of every transaction [9].

2.1.3 Transaction

A transaction refers to the interaction among nodes. With crypto-currencies, e.g., a transaction refers to a movement the crypto-currency

between the peers of the Blockchain network [23]. The transaction contains the sender and the receiver addresses, and other data. Before sending, could be signed the transaction by the private key relevant to the public key of the address [26].

Additionally, the transactions are not arranged based on of a generation because of the propagation delay in the P2P network. Accordingly, the transactions are grouped at a given time to create a block and publish these blocks to the network [2]. Before the broadcast, the transactions to its neighbors, each node that receives the transaction will first verify each transaction with a long checklist of criteria. This assures that only validated transactions are publishing on the network, whilst invalid transactions are rejected by all node that interviews them. Then every node creates a pool of only valid new transactions, nearly in the same order [5].

2.1.4 Ledger

Ledger is the technology upon it the records of transactions are published across multiples sites, companies or institutions, countries and are typically public. Blocks (collection of Transactions) are stored one after another in a continuous ledger, but they can only be inserted when consensus on it [27]. It is immutable in that once data is inserted to the ledger; it cannot be altered [20]. In the distributed ledger each record holds a timestamp and unrepeatable cryptographic signature, thusly making the ledger an auditable date of every transaction in the Blockchain network [28].

2.2 Blockchain Structure

Essentially, the Blockchain is a linked list of the block which utilizes hash pointers rather than usual pointers. Hash pointers are utilized for pointing to the former block [24]. Which consists of timestamp ordered, linked blocks that contain all of the transactions. The blocks are linked such that each block contains the ID of the previous block at the chain [21]. Each block is chained to other blocks via referencing a parent block. If any content in the header is altered, then its child block header will contain invalid hash. Transaction modification is will also detect. A block header also contains the Merkle root of the Merkle tree structure [29]. Consequently, this generates a tamper evident log that impossible to be altered. Furthermore, hash pointer used to trace even the first block named genesis block [2]. This led to the possibility of easily determining and rejecting the changed blocks as in Figure (2.3) [23].

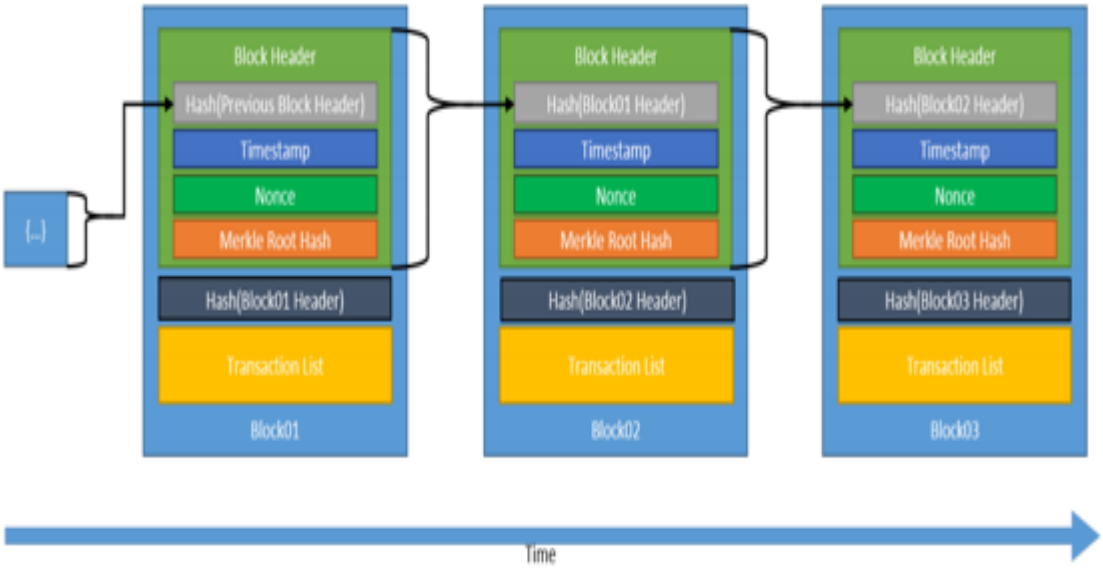


Figure (2.3): Generic chain blocks [23].

2.3 Distributed Blockchain

Distributed Blockchain is block distributed across the nodes of P2P blockchain and the whole nodes hold the same Blockchain copy. The utilization of distributed Blockchain led to the users do not require to equip email, social security number, or telephone number to authority or any central server. The users are capable of generating their digital identity and distributing their public key to the entire distributed network. So, management of distributed decentralized anonymous identity can be provided to the users. Every node in the distributed Blockchain has copies to the whole transactions, which means a node can to monitor the history of whole transactions [2].

2.4 Blockchain security

Blockchain security is an important part of Blockchain technology so it can use asymmetric cryptography. Here, cryptography is often utilized for providing confidentiality service. It cannot be labeled as a perfect solution, however, it represents a decisive constructing block into a big system of security for processing the issue of security. Cryptography supplies different security, like authentication, integrity, confidentiality, authentication of the entity, and authentication of information origin and non-repudiation [2].

2.4.1 Cryptographic Hash Function

One of the essential Blockchain technology content is the utilization of a cryptographic hash function for various processes, like hashing the block content [23]. A cryptographic hash function is a

mathematical model which takes any input of data (string) of any length and results in an alphanumeric string of constant sized. The resulted the string is named digest or hash value or digital fingerprint or checksum. Always, the function obtains the same hash to the same data, although the number of times recalculated.it can be used to validate the integrity of data because the hash cannot be reversed to get the input data and for this reason, it is named a one-way hash function [2]. There are several essential security characteristics in cryptographic hash functions [23]:

1. They are pre-image resistant (for an instant way of one); computationally, it is not useful to calculate the accurate rate of input to produce some value of output (example, a digest is given, compute x, and find "hash(x)" = "digest") [23]. Eq(2.1)

2. The pre-image resistant that consider second, that consider one that does not have the ability to input calculation that hashes to a particular production. Computationally, it is impossible to get a second input that results in an exact output (like, assumed x, determine y in way which hash(x) = hash(y) [23]. Eq. (2.2).

3. 3. They are collision resistant, which meaning one can't get 2-inputs that hash to an exact production. Computationally, it is not useful to get any 2 inputs that yield digest that is exact (like, get an x and y in way which hash(x) =hash(y) [23]. Eq. (2.3).

The technologies of Blockchain take many of transactions and generate a hash fingerprint (the digest) to the list. Any user based on exact transactions list can create the exact digest (fingerprint). When changing a single value in a transaction inside the list, the fingerprint of this block will altered, making it easy to detect cover till minor one-bit

alters [23]. This would easily detectable to the full network because it would be clear that the digital fingerprints have been altered and all transactions would be refused by the nodes, which are responsible for validating transactions and blocks [27].

These hash codes are used in order to interconnect blocks together as in Figure (2.4) [27].

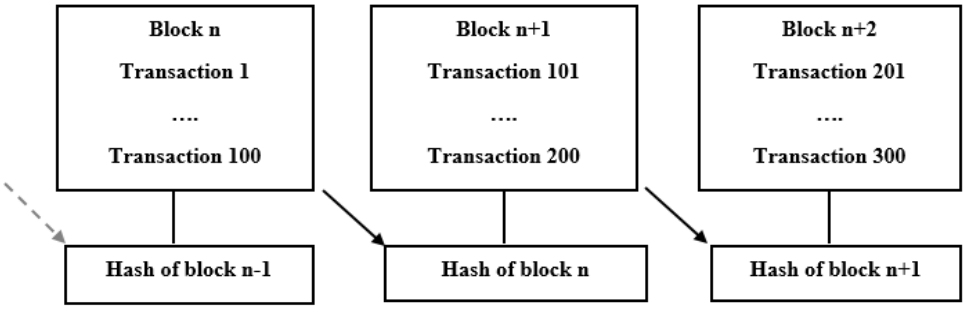


Figure (2.4): The interdependence of blocks [27].

The algorithm (2.1) [**30**] and figure (2.5) explain one round of SHA256 as following [24]:

Algorithm (2.1): SHA256 Algorithm

Input: Message with any length;
Output: Hash Function of 256 bit;
Begin: **Step1**: Append padded bits The message is filled so that its length is congruent to 448, Modulo 512. This padding is single 1 bit added to the end of the message, followed by as many Zeros are required so that the length of bits equals 448 modulo 512. **Step 2:** Append length A 64-bit representation of the message's length is appended to the result. This Step to make the message length an exact multiple of 512 bits in length. **Step 3:** Parsing the message

The padded message is parsed into N 512-bit message blocks,

M (1), M (2)… M (N), by appending 64-bit block.

Step 4: Initialize Hash Value

The initial hash value, H (0) is set, consist of eight 32-bit words, in a Hexadecimal form.

Step 5: Prepare the message schedule

SHA256 uses a message schedule of sixty-four 32-bit words. The words of The message schedule are labeled *W0, W1… W63*.

$$Wt = \begin{cases} M_t^{(t)} & 0 \le t \le 15 \\ \sigma_1^{(256)}(W_{i-2}) + W_{i-7} + \sigma_0^{(256)}(W_{i-15}) + W_{i-16} & 16 \le t \le 63 \end{cases}$$

Where:

$\sigma_1^{(256)}(W_{i-2})$ = $((W_{i-2})\ ROTR\ 17) \oplus ((W_{i-2})\ ROTR\ 19) \oplus ((W_{i-2})\ SHR\ 10)$

$\sigma_0^{(256)}(W_{i-15})$ = $((W_{i-15})\ ROTR\ 7) \oplus ((W_{i-15})\ ROTR\ 18) \oplus ((W_{i-15})\ SHR\ 3)$

Step 6: Initialize the eight working variables, a, b, c, d, e, f, g, and h, with he (i-1)st hash value

For t=0 to 63: {

$T1 = h + \sum_1(e)^{256} + Ch\ (e, f, g) + k_1^{256} + W_t.$

$T2 = \sum_0^{256}(a) + Maj(a,b,c)$

H = G

G = F

F = E

E = d + T1

D = C

C = B

B = A

A = T1+T2 }

Where $\sum_1^{256}(e)$ *= (e ROTR 6) XOR (e ROTR 11) XOR (e ROTR 25).*

$\sum_0^{256}(a)$ *= (e ROTR 2) XOR (e ROTR 13) XOR (e ROTR 22).*

Ch (e, f, g) = (e ∧ f) XOR (~ e ∧ g).

Maj (a, b, c) = (a ∧ b) XOR (a ∧ c) XOR (b ∧ c).

Step 7 : Output

After repeating steps one through four a total of N times, the resulting hash function is

$H_0^N \| H_1^N \| H_2^N \| H_3^N \| H_4^N \| H_5^N \| H_6^N \| H_7^N \|$.

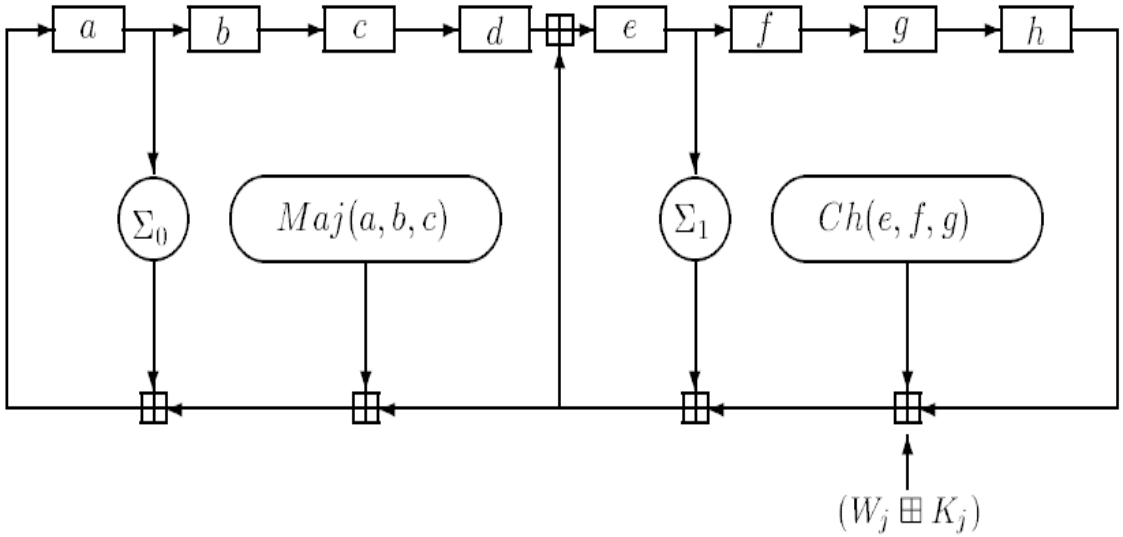


Figure (2.5): A single round of SHA256 function [24].

2.4.2 Digital Signature

Digital signature gives a facility to associate a message with an entity created from this message. It is utilized for providing nonrepudiation and data origin authentication [24]. It is required to authenticate the transaction when creating a transaction on the Blockchain [11]. Additionally, it is generated via utilizing public key cryptography. Public key cryptography utilizes a key that is a collection of private and public keys [2]. Any user has a pair of keys (private and public). The private key is utilized for signing the transactions. The digitally signed transactions are published over the entire network [9].

The public key is broadcasted publicly to determine the digital identity [31]. The typical digital signature is involved with two phases: signing phase and verification phase [9]. The figure (2.6) below explain digital signature scheme [32].

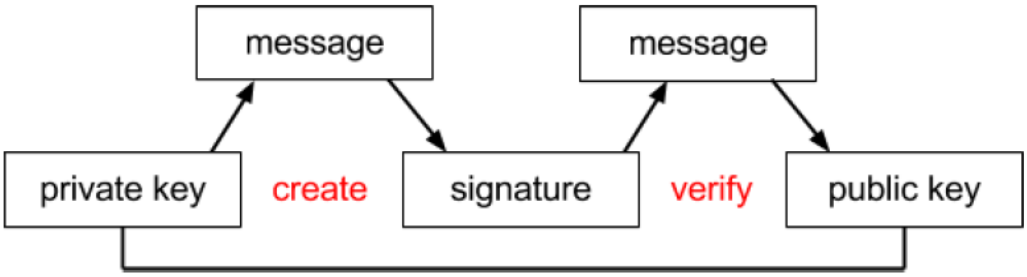


Figure (2.6): Digital Signature Scheme [32].

2.4.3 Merkle Tree

The Blockchain based P2P network in which every peer should have an exact copy of transaction that must be propagated and verified over the P2P network. This is computationally expensive and time consuming. A Merkle tree is utilized to summarize the transactions in each block. This tree is an effective data structure, also named a binary hash tree, which works on summarizing and verifying the large data sets integrity [31]. In this tree, firstly, the inputs are located at the leaves. Secondly, the values of the pair of children nodes are hashed with each other for producing internal node value (parent-node) till a Merkle root (a value that is single-hash-) is obtained [24].

Therefore, The Merkle tree is utilized that rather than sending data only, the data hashed is transmitted and the receiver node matching the hash value against the root [2]. This is accomplished for freeing up the space of storage required to store the Blockchain on the nodes [3]. Figure (2.7) shows an example of a Merkle tree [24].

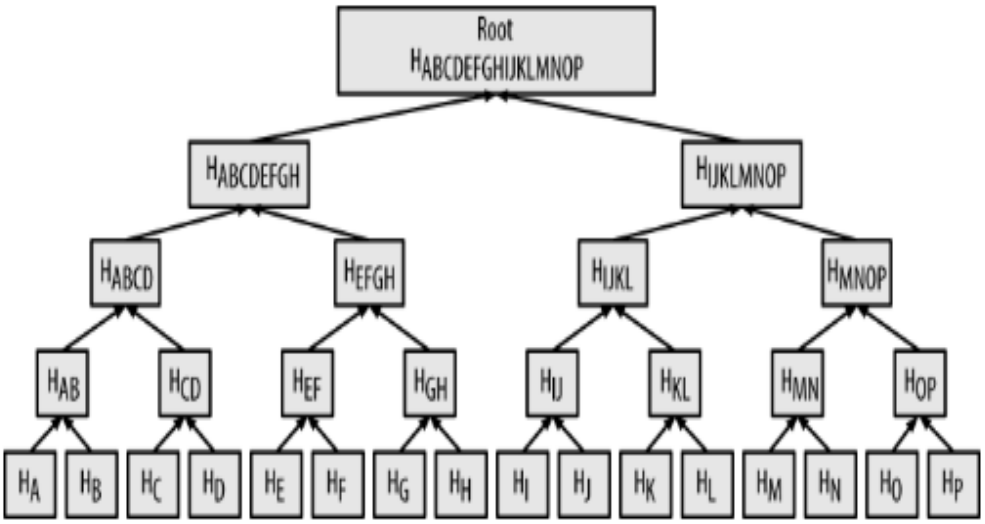


Figure (2.7): An example of a Merkle tree [24].

2.5 Categorization of Blockchain systems

The networks of Blockchain can be classified depending on their model of permission that limits who can maintain them for instance, publish blocks. When anybody is capable of publishing new blocks, it is permission-less. But when only specific users are capable of publishing new blocks, it is permission [23]. There are three kinds type of Blockchain network:

1) Public Blockchain (permission-less): It can be read or write data by any user wish to connect to the network. It is publicly open and anybody is capable of participating as a node in the process of making a decision. These ledgers do not belong to anyone and are open to the public to participate in the permission less or public network [24]. The most common platforms of Blockchain, Ethereum, and Bitcoin [20]. Figure (2.8) shows public Blockchain [6].
2) Consortium Blockchain (permission): In this category, the Blockchain contains two parts one is private and public. The

control of the private one can by a collection of users while the public part consider open to anyone for contribution [24]. Like Hyper-ledger is consortium Blockchain as in Figure (2.9) [6].

3) Private Blockchain (permission): is considered as a centralized network because it is completely controlled via one organization [9]. The node will be restricted, in this Blockchain, not all nodes capable to participate has strict authority management on data access. Private Blockchain is shown in figure (2.10) [6].

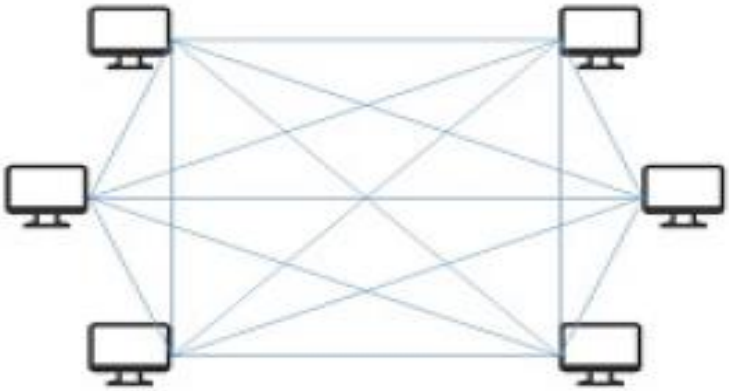


Figure (2.8): Public Blockchain [6].

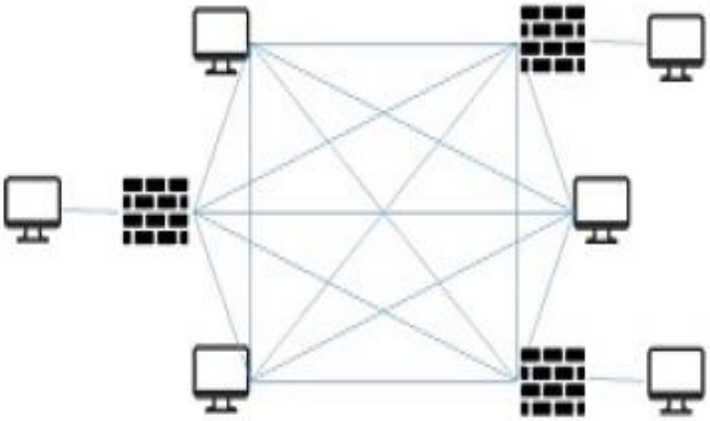


Figure (2.9): Consortium Blockchain [6].

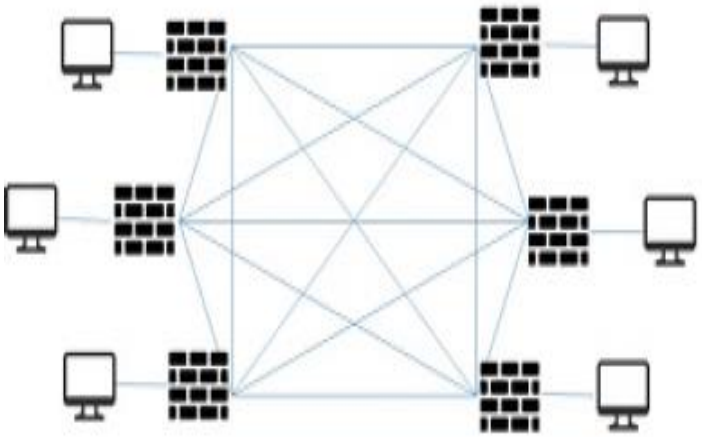


Figure (2.10): Private Blockchain [6].

2.6 Distributed Consensus

Consensus of distributed protocol is the cornerstone of the Blockchain [18]. For reaching an approval on that transactions must be inserted to the distributed ledger, the Blockchain utilizes the protocol of distributed consensus [2]. The next subsections refer to the major consensus models that exist today [27].

2.6.1 Proof-of-work (POW)

Proof-of-work (POW) was proposed by Miguel Castro and Barbara Liskov [2] in 1999 as a solution to the Byzantine general issue [33]. A double-spending problem can be solved in a distributed system by utilizing POW where the P2P distributed timestamp server creates a hash of POW [2].

This kind is utilized in Ethereum and Bitcoin [24]. The mechanism of POW utilizes the solution of the puzzle for proving the data credibility. Generally, the puzzle is hard in computation but an easily verifiable issue. If the node generates a block, then it should solve the puzzle of POW.

After the resolving of the POW puzzle, in order to fulfill the consensus purpose, it will be broadcasted to the other nodes [34].

2.6.2 Proof-of-stake (POS)

POS is a consensus model introduced in 2012 [33]. It is a substitutional algorithm to POW that the capability of verifying and publishing blocks based on the stake (for instance, the native currency amount) already possessed [25]. In a Blockchain based on POS. The mechanism of POS can highly minimize the amount of calculation, which increases the throughput of the whole system of Blockchain [34]. Miners don't require high end computers for participating in the mining. Rather a smaller powerful computer is enough [2].

2.7 Blockchain Work

Blockchain technology, is merely a chain of “blocks”, each containing a unique set of validated transactions that each contain a cryptographic fingerprint called a “hash”. That are grouped together in such a way that the information remains accessible but cannot be tampered with it. Blocks are linked in linear, sequential order by their unique hashes that act as fingerprints—hence the concept of a chain [27]. In figure (2.11), the transaction is composed of the sender, the transaction information, and the receiver, and it is secured by an encryption code and denoted them as (STR). And figure (2.12) shows how it works [1].

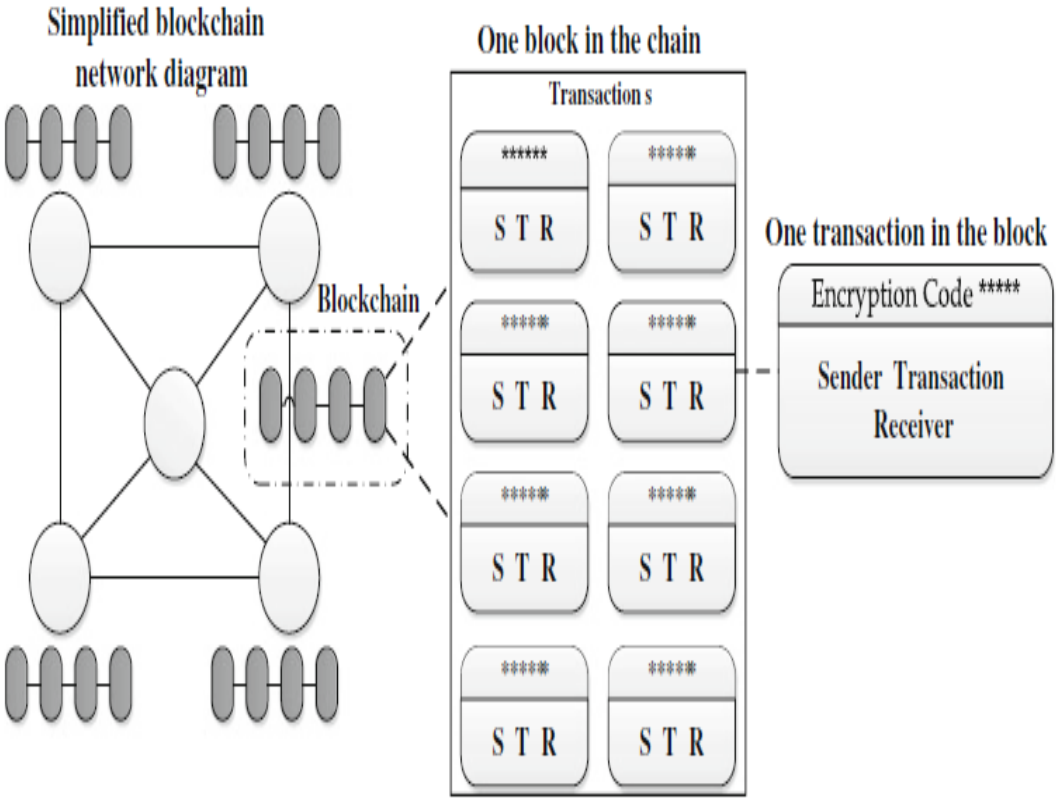


Figure (2.11): Basic components of Blockchain [1]

1- Transaction definition: The sender (One party) generates a transaction and sends it to the network. The message of the transaction contains details of the receiver's public address, the value of the transaction, and a cryptographic digital signature that proves the authenticity of the transaction [1].

2- Authentication of transaction: The users and computers (nodes) of the peer network receive and authenticate the message by deciphering the digital signature. The authenticated transaction is located in the appended transactions pool [1].

3- The generation of block: The appended transactions are placed in a ledger updated version, named a block, via one of the nodes in the network. At a particular timing interval, the node broadcasts the block to the network for validation [1].

4- The validation of block: The validator nodes of the network work on receiving the proposed block and validating it by an iterative process that needs consensus from the majority of the network. Essentially, since the entire parties have an exact set of data, they validate via assuring the information matches their ledgers. Various networks of Blockchain utilize various techniques of validation [1].

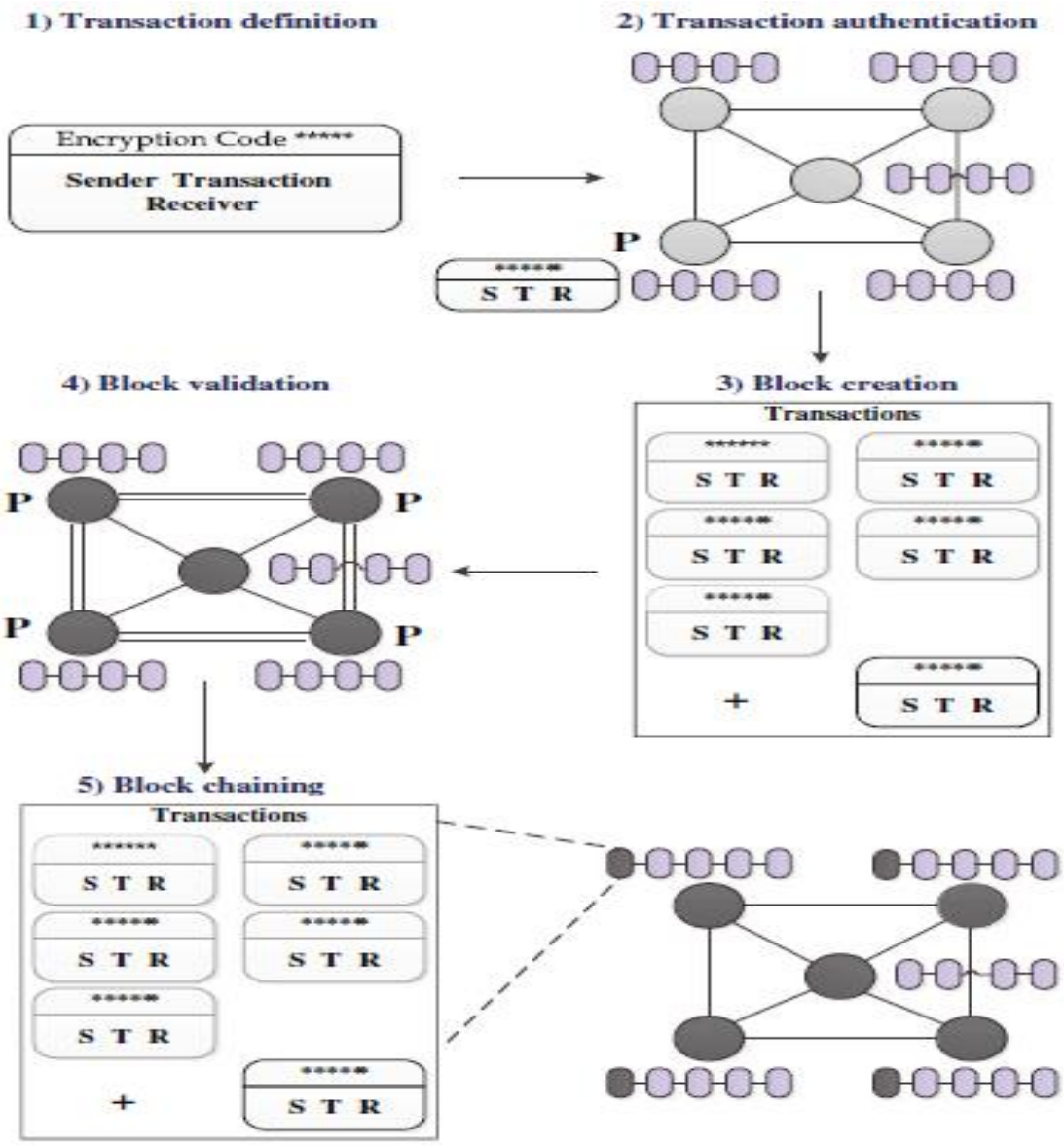


Figure (2.12): How Blockchain Works [1].

2.8 Authentication

Authentication is process of the association of a person with their identity in order to be identify and verify who he is claim to be. Authentication process consider more a matter and critical of urgency than before. Concept of authentication that has growing since of increased criminal's activities. That made biometrics reach to the limelight [35]. People use passwords and different authentication methods to protect a person's data and potentially confidential information. Traditional authentication methods (like personal passwords) are less than secure. Besides requiring the user to remember a different passwords, which can result in an error in user password, which can be stolen and clear password authentication is vulnerable to unauthorized access [35].

Authentication of the user identity can be achieved as follows: 1) something that persons know like a password; 2) something the persons have like a special card, and a key; 3) something the persons are like footprint, and fingerprints. The Biometrical process (biometrics authentication process) assures that the persons are who they claim to be [37].

There are two categories of biometrics technology being used today. One is physiological biometrics which measures characteristics that can be empirically identified such as the (face, fingerprint, hand, and iris). Another category is labeled as behavioral which includes (signature, voice, and keystroke) [38]. These traits or attributes are singular for every individual and consequently, only the owner can prove the authenticity. Furthermore, comparing with other methods of authentication, it is

extremely secure since it is highly tough to crack or steal the characteristics of users [2].

2.8.1 Fingerprint Biometric

Fingerprint biometric is the oldest biometric approach. The fingerprint is classified as being used to identify a person and verify his identity is one of the forms of biometrics. It can prove the identity of the person in a secure and reliable manner compared to the keys, passwords or identity card. There are no two people with the same fingerprint. Even the same person with ten fingers is different [36]. A fingerprint image includes of valleys and interleaving ridges. Often, valleys are the white lines and interleaving ridges are black lines between ridges, see Figure (2.13).

Bifurcations and Ridge terminations are types of minutiae which are trait features of fingerprints. Minutiae fingerprint in fingerprint is illustrated in figure (2.13) [39].

Figure (2.13): Ridges and Valleys [39].

To reduce the original data set is the goal of extracting the feature by measuring specific features or characteristics, which recognize one entry pattern from another as shown in the next section.

One of the traditional tools that can be used to extract features for the fingerprint used is the Hu' seven moments method is briefly reviewed below.

2.8.1.1 Feature Extraction by Invariant Moments

To extract features there is a traditional and frequently used method which is moment-based features. To describe the texture of the area, fixed moments are one of the main methods that can be used in image processing. Invariant to rotation, translation, and change in size are the seven moments that can be used here. This is to deal with different input methods, as well as to significantly reduce the effects of non-linear distortions, in order to better maintain local information. [40].

Hu presented seven moment invariants based on normalized central moments as follows:

$$\emptyset_1 = \eta_{20} + \eta_{02}. \quad \textit{Eq. (2.4)}$$

$$\emptyset_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2. \quad \textit{Eq. (2.5)}$$

$$\emptyset_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \mu_{03})^2. \quad \textit{Eq. (2.6)}$$

$$\emptyset_4 = (\eta_{30} - 3\eta_{12})^2 + (\eta_{21} - \mu_{03})^2. \quad \textit{Eq. (2.7)}$$

$$\emptyset_5 = (\eta_{30} - 3\eta_{12}(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2\text{-}3(\eta_{30} + \eta_{12})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]. \quad \textit{Eq. (2.8)}$$

$$\emptyset_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2[(\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21+}\eta_{03}). \quad \textit{Eq. (2.9)}$$

$$\emptyset_7 = (3\eta_{21} - \eta_{03})(\eta_{03} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]. \quad \textit{Eq. (2.10)}$$

The moment of order (p + q) depends on scaling, translation, rotation and even on gray plane changes given by the above equations [42].Also the

seven moment invariants are useful properties for changing without translating, scaling, and rotating of an image [41].

2.8.2 Linear Congruential Generators (LCG)

A large variety of applications could be used. To generate a series of pseudorandom numbers can be use LCG based on several recurrence congruence. The following equation is the simplest form of the LCG used (12) [43]:

$x_{n+1} = a * x_n + b \ mod \ m.$ *Eq. (2.11)*

For a is named the multiplier, b is the increment, and m the modulus. Assumption: m > 0 and a < m, b < m, x_0 < m. The generated numbers will be in sequence x_0 , $x_{1\ ,}$ $x_{2\ ,}$ ... , for x_0 is initial value named seed. The parameters of an LCG are a; b; m; andx_0 .The selection of its parameters determine the quality of an LCG [43].

2.8.3 Big Integer

It has become possible to deal with mathematical operations very well on modern computers and deal with numbers that do not exceed the length of (32 bits) or (64 bits). So, computer calculations such as subtraction, addition, division, and multiplication with numbers greater than this length became impossible. Since most of today's computer's CPU registers are 32-bit (4 bytes) and 64-bit (8 bytes) wide. It can only deal with numbers that have this length. To solve the problems of arithmetic operations on the big numbers, number of algorithms were developed. They first convert big numbers from base-10 to base-2, then apply a bitwise process on the bit level.

The Java programming language provides BigInteger class in the java.math package. Several programming libraries have been developed. It carries out arithmetic operations on big-integer numbers using bitwise operations. [44].

Java contains a BigIn-teger class. Its supports the following basic operations on the integer number : Subtraction (subtract-BI),Addition (add-BI), multiplication (multiply-BI), division (divide-BI), remainder (remainder-BI), combination of division and remainder (divide-Remainder-BI), modulo (mod-BI) (slightly different from the remainder(BI)), and power (pow (int exponent) [45].

The simplest way to store the big integers, the BigInteger library works to save it as a (long) string.

Whereas this is an overflow in 64-bit unsigned long-long in C/C++ language already. So, BigInteger library uses a class of digit-by-digit processes for processing two Big-integer operands [44].

2.8.3 Rabin-Miller Algorithm:

The simple algorithm used by everyone is designed by Michael Rabin based on some ideas from Gary Miller. Algorithm is used to test large numbers for Primality. The test algorithm (2.2) is as follows [46]:

Algorithm (2.2): Rabin miller Algorithm

Input: *Choose p as random number to test.*
Output: *Prime number or not.*
Begin
Step 1: *Compute b, for b is equal to (p - 1) divided by 2 (i.e., b is the largest Power of 2, such that 2b is a factor of p - 1).*
Step 2: *Compute m, where* $p = 1 + 2^b * m$.
Step 3: *select a random number 'a' where 'a' is smaller than p.*
Step 4: *Fix j = 0, set* $z = a^m \bmod p$.
Step 5: *Check if z = 1 or if z = p - 1, then p exceed the test and may be a prime Number.*
Step 6: *Check if j> 0 and z = 1, then p consider not a prime number.*
Step 7: *Fix j = j + 1.Check if j <b and z ≠ p - 1, fix z = z2 mod p and return to Stage 4. Check if z = p - 1, then p exceed the test and may be prime.*
Step 8: *Check if j = b and z ≠ p - 1, then p consider is not a prime number*
Step 9: *End Algorithm.*

For example let's take the number 221 for testing Primality [47].

Where P=221 then P-1=220=2^2*55, b=2, m=55.

Picking random a1 in range 0< a1<221, a1=174

$a1^{2^0 m}$ Mod p = 174^{55}mod 221 = 471; p – 1.

$a1^{2^1 m}$ Mod p = 174^{110}mod 221 = 220 = p – 1.

Since 220 ≡ -1mod p, either 221 is prime, or 174 is a strong liar for 221.

2.8.4 RSA Algorithm

The RSA algorithm is an algorithm of public-key. It is discovered by both Ron Rivest, and others (RSA) in 1977. It works on both signatures that are digital and encryption. Who uses the RSA algorithm generate and then propagate the product of two-large prime numbers, besides with an auxiliary value, as their public key [48].

In sanctuary records like transport security of data, IP data security, and security of the email, can use RSA. RSA algorithm use different two keys with the relationship of a mathematical with each to each other. The premise of RSA relies on that knowing one of the keys do not help to know the other. By using the RSA algorithm is carefully generate the public key and private keys [49].

Consisting public key from the *n* value that is named as the modulus, whereas the *e* value, that is names as the exponent that is public. Consisting the private key from the *n* modulus and the *d* value, that is named as exponent of the private. The public-key and pair of the private-key of the RSA can be produced as algorithm (2.3) [49]:

Algorithm (2.3): RSA Algorithm

Input: Pair of large random prime number's p and q.
Output: Pairs of keys (public and private).
Begin **Step 1:** Calculate the modulus n such that n = p*q. **Step 2:** Determine an odd public exponent (e) between (3) and (n-1) that is relatively prime to be p-1 and q-1. **Step 3:** Calculate the private exponent d from e, p and q. **Step 4:** Output (n, e) are as the public key and (n, d) are as the private key. **Step 5:** End Algorithm.

Both sender and receiver in the RSA algorithm should know the value of *n*. The value of *e* is known by the sender, and only the receiver knows the value of *d*. therefor; defined the public-key (PU) as {e, n} and defined the private key (PK) as {d, n} [50].

The encryption operation for message M is computed during the equation (2.12):

$$C = m^e \bmod n \qquad \text{Eq. (2.12)}$$

The decryption process for the message encrypted C is calculate through the equation (2.13):

$$M = c^d \bmod n \qquad \text{Eq. (2.13)}$$

The basic advantage of the RSA algorithm is to select d given e and n is infeasible [50].

2.9 Authorization

Authorization is the operation that determines if the user has authority for accessing the requested resources or issue some commands. It is tightly associated with the authentication because the user should be authenticated to become authorized [2]. Reliable authorization and authentication are becoming necessary for many everyday actions or applications and this brings the concern of security. Authorization is granting users' permissions in the concepts of accessing to digital resources and specific actions to given entities and the extent of its usage. Authorization is identical to the successfully authenticate users according to his/her rights information available in the Management System. Authorization determine the responsibilities issue assigned to various nodes included in the maintenance of a Blockchain in the concept of editing, addition, deletion, and uploading of records on the network.

Authentication is less challenging than authorization, essentially, to exceedingly distribute digital content suppliers [8].

In this thesis will depending on authentication as basic to become the person authorized and granted the access right to certain resources.

2.10 Potential Vulnerabilities

Though the technology of Blockchain prohibits many kinds of malicious-attack, it does not get rid of all kinds of attack. In its mechanisms of preventative (such as, cryptography, anonymity, and distributed consensus) which may destroy its strength. These are the following [51]:

a. The Fifty-One Percentage Attack

A fifty-one percentage attack may happen when a single miner node, which dominates the Blockchain content [51].

b. Identity-Theft

Though Blockchain keeps privacy and anonymity, protecting the private key from lost, it will not be able to be recovered. So, all the assets belong to the person owns in the Blockchain will steal [51].

c. Sybil-attack

Because Blockchain doesn't depend on a central authority to manage the identities of the participants. The attacker can generate multi-copies of itself, the attacker can be able to refuse to send blocks and transactions from other nodes. Can solve this attack with POW [2].

d. Illegal activities

The pseudo-anonymity, decentralized property, and immutable transaction for the Blockchain making it difficult to track and monitor transactions on Blockchain [21]. Therefore, the system is capable of misusing money launder, illegal movement of funds [2].

e. System hacking

It is tough to storage and change records stored in a Blockchain, but not systems and the programming codes that implement its technology [51]. It can altered since based on the organization or company, any user can be participate development of these applications by putting the vulnerable code or there are human-mistakes in the code-base may probably end up in the production system that in turn might cause system attack [2].

2.11 Blockchain Applications

Blockchain can be utilized in many application fields financial, non-financial [2].

2.11.1 Financial Applications

Presently, Industries financial could apply Bitcoin Blockchain into their fields to develop their systems [2].

- Bitcoin

Bitcoin is a kind of no regulated digital currency also called crypto-currency which was firstly generated in 2008, via Satoshi Nakamoto. It was initiated with the intent of solving the issues of trust, accountability,

and transparency to exchange money, services, and goods between two parties over the internet and eliminate the intermediaries [52].

The technology of Bitcoin has many useful-nesses. It is secure because the transactions are secured via using public key cryptography and trust is established in a P2P way. It is inexpensive since the transactions are broadcasted immediately over the P2P network and they are propagated quickly to the other nodes. The intermediate cost of handling transactions is lower than the available financial system transaction costs. Bitcoin has several vulnerabilities, it is time-consuming to get the transactions confirmed as new blocks are added every 10 minutes to the Blockchain. Also, the computational power for solving the mathematical issue is exponentially maximizing with the time. [2].

2.11.2 Non-Financial Applications

Many organizations are searching for manners to improve Blockchain and incorporating it to their businesses such as in Bitcoin which used only in financial fields. So to solve this problem can be used Ethereum.

- Ethereum

Ethereum was first introduced by Vitalik Buterin in his paper in early 2014 [29]. It is the second platform's most common Blockchain application. It was adaptive for solving the disadvantages appears in Bitcoin [2]. Therefor; Ethereum supports every type of computations. Ethereum is open source Blockchain platform for executing Smart Contracts, anyone can build various services, contracts, or applications run on this platform [6].Ethereum utilizes a POW algorithm different

from that used in Bitcoin, named Ethash. Ethash is an algorithm of memory intensive and not a computational intensive one [26].

The Ethereum is decentralized and secure, but it has several disadvantages. Its code-base is open source and kept by a tuple of developers. Therefore, when a bug is found in the technical fault or code-base, hackers can easily utilize it [2].

Chapter Three

The Proposed Design of Authorization Technique in Simulation Environment Blockchain System

Chapter Three

The Proposed Design of Authorization Technique in Simulation Environment Blockchain System

3.1 Introduction

This chapter describes the techniques and design details of proposed ASBchain). The proposal system is operating in decentralized, distributed, and reliable blockchain network. Through the proposed ASBchain system the users must be authenticated and enable exchange transactions without sharing their especial data to all servers or nodes for many time on network and without sharing third the party as central.

The design of the proposal ASBchain system with new proposed authentication and authorization methods depending on SHA-256 hashing function and using Blockchain technology fundamentals.

In this chapter, section (3.2) introduces the general design of the proposed system. The design details and techniques of the proposed system are described in (3.3).

3.2 The Block Diagram of the Proposed System

The basic idea of the proposed system is to keep the level of privacy in authentication and authorization in blockchain network througth allow multiple servers to test client's authentication in order to be authorized, to be communicated in a trustworthy way over a decentralized network. The block diagram of the proposed ASBchain system can be shows in figure (3.1) that includes six main phases to perform authentication and authorization task in high level of security.

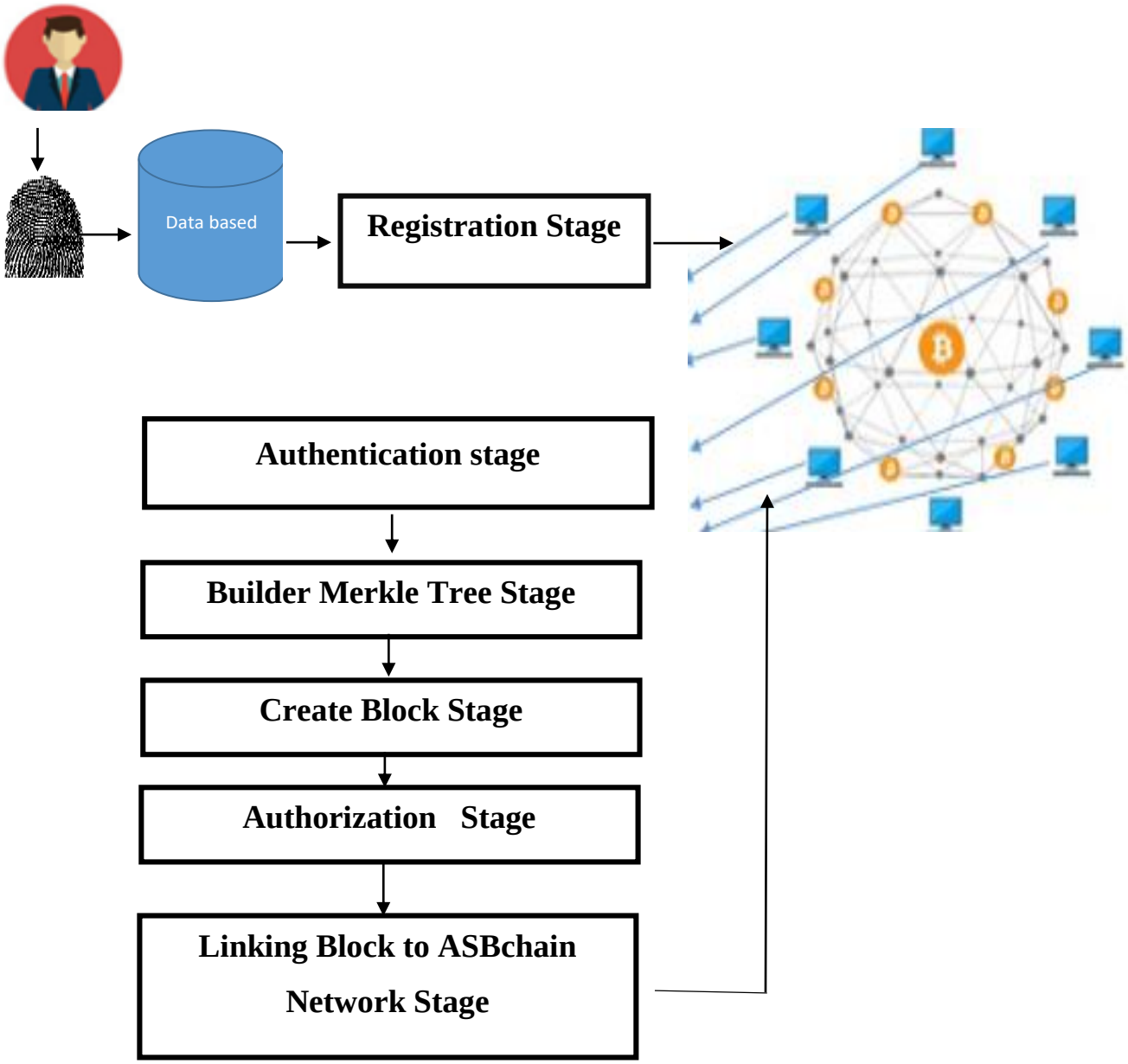


Figure (3.1): Block diagram of the ASBchain Proposed system

3.3 The proposed system

As shown in figure (3.1) the proposed system has many steps must be followed these steps in order will be illustrated in section (3.3.1).

3.3.1 Registration Stage

The first stage in the proposed system is a user registration to create a transaction. This stage is clarify as follows:

3.3.1.1 User Request for Create Transaction Steps

The user participates in the proposed system through a request to create a transaction, at the first step the user is asked to enter a fingerprint of Hu's seven moment's features type that stored as dataset [41].

The second step is generated transaction for the user who is requested to create transaction. A user transaction consists from two field are: SHA-256 of moment feature (after normalizing it) and digital signature. This step has several sub-steps that perform task to achieve their final aim which is generate user's transaction to be able to broadcast to all nodes of ASBchain system. The detailed block diagram of the create transaction or Registration Stage steps in the proposed system is shown in figure (3.2). Where the database store the Hu's 7 moments as features to 100 fingerprints [41].

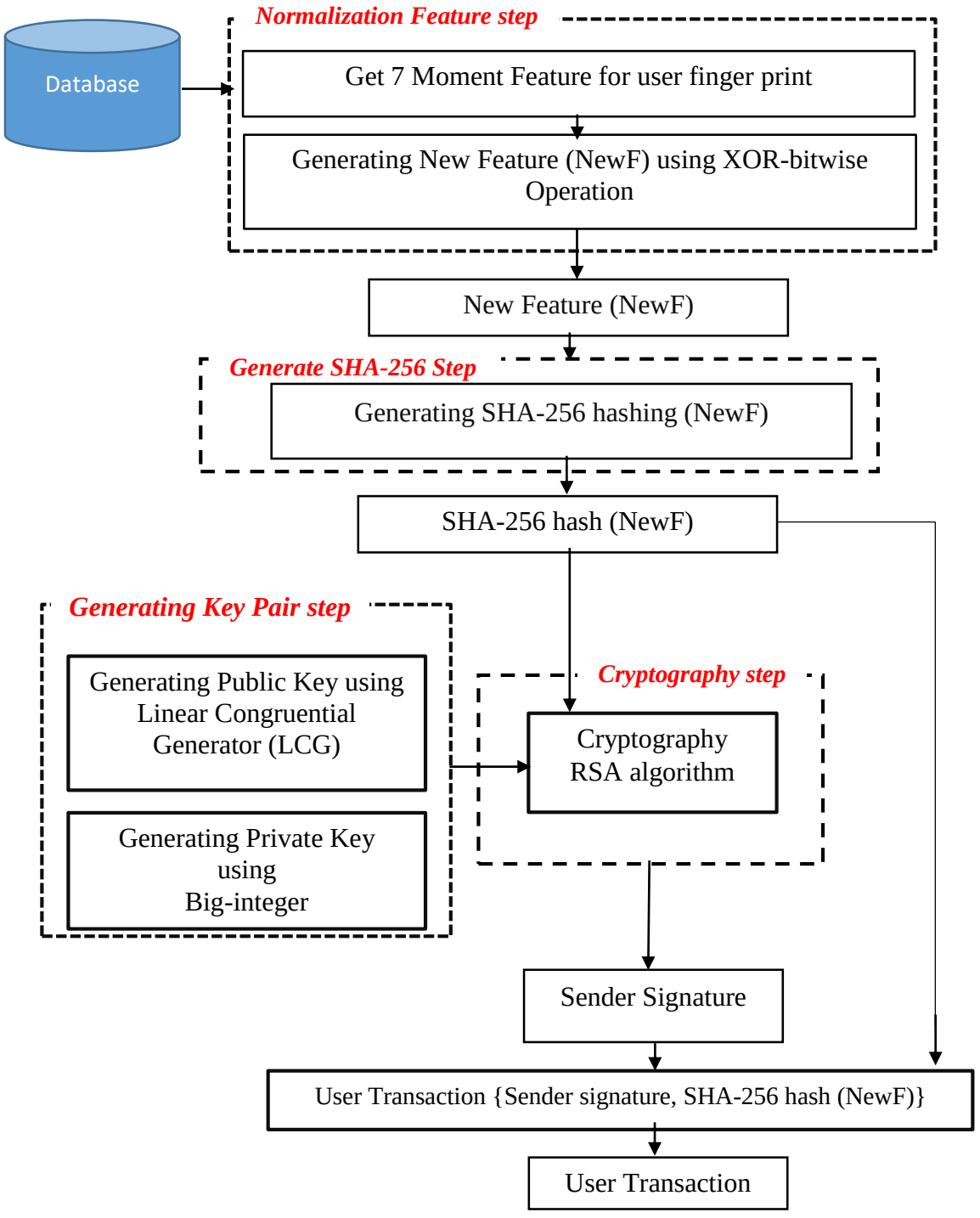


Figure (3.2): Block diagram of Generating Transaction Step.

I) **Normalization Feature Step:** the aim of this step is normalizing seven moments feature of user fingerprint feature to generate one as new feature, this step consists of two sub-stages:

a) ***Load user's 7 moments Feature:*** the first step in generating transaction is loading 7 moments features from the dataset[41] which are represents Hu's seven moment of the user finger print image. Each moments features denoted as (M). Table (3.1) illustrated an example for Hu's seven moments feature for one user.

Table (3.1): Hu's 7 Moments Feature of the One User.

M_i	*Value*
M_1	0.378251222741672
M_2	1.85801715072987E-05
M_3	0.211595782978178
M_4	0.212014750801354
M_5	-0.582177820214538
M_6	-0.000678736409468635
M_7	0.035490288789322

b) ***Generating New Feature based on XOR bitwise operation Stage:*** this stage aims to generate new features by apply Xor bitwise operation between seven moments to reduce data and at the same time maintain data without loss. The result of this step is named as the new feature that denoted by ***NewF***. To generating *NewF* from seven feature moments [$M_1, M_2, M_3, M_4, M_5, M_6, M_7$] must follow two steps are:

1- Since each moments *M* are a float number then apply truncate operation on each them and take only the numbers following the point. Table (3.2) shows example of truncate step for 7 moments feature.

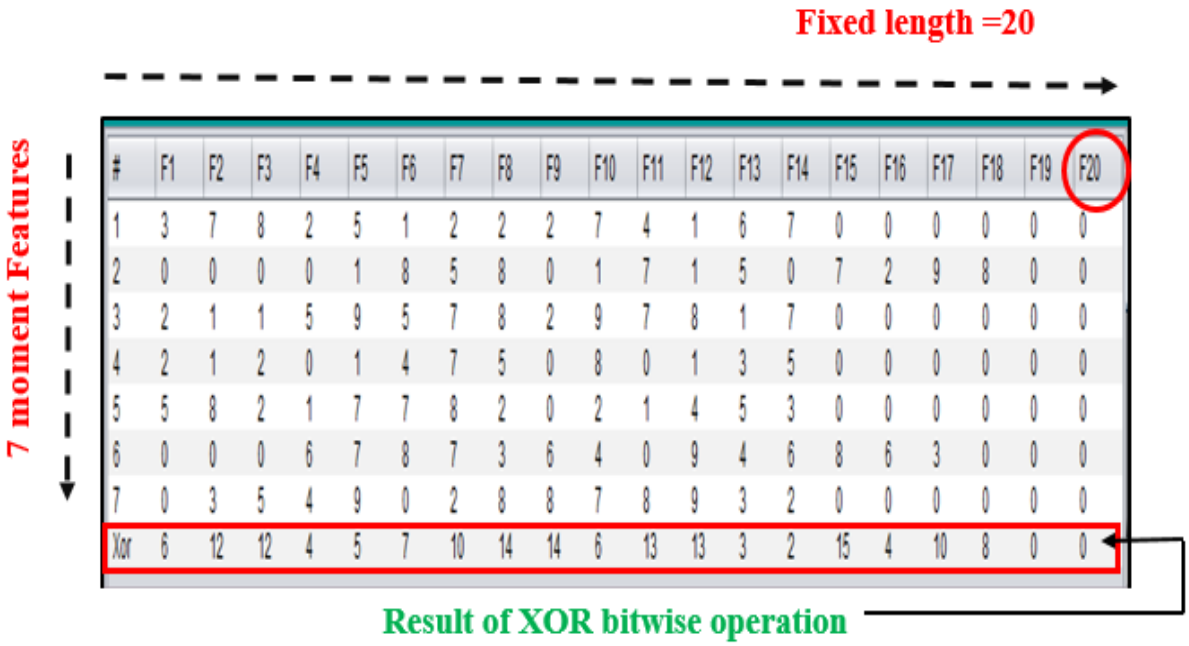
Table (3.2): Example of Truncate step of generate NewF

M_i	**Value of Feature moments**	**Truncate value of Feature moments**
M_1	0.378251222741672	378251222741672
M_2	1.85801715072987E-05	185801715072987
M_3	0.211595782978178	211595782978178
M_4	0.212014750801354	212014750801354
M_5	-0.582177820214538	582177820214538
M_6	-0.000678736409468635	678736409468635
M_7	0.035490288789322	035490288789322

2- The result of the truncate step is 7 integer number for 7 moments features with variant length, to be able to implement the Xor operation between 7 moments features follow in order two step:

2-1 Make all value of 7 moments have the same length by specifying the length of the numbers of each moment feature value that must be equal to a special length and this length is indicated as (fixed length) where each number represents a single feature (M). For example, if Fixed length =20 then all [$M_1, M_2, M_3, M_4, M_5, M_6, M_7$] length must be equal to 20 and reverse it padding moments value with '0' to be its length equal 20.

2-2 After making all length values equal then apply Xor bitwise operation between all features $F_i = [F_1, F_2, .., F_{\text{Fixed Length}}]$ separately and respectively and the result of Xor operation is named (NewF) as clarifying in Figure (3.3).



#	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
1	3	7	8	2	5	1	2	2	2	7	4	1	6	7	0	0	0	0	0	0
2	0	0	0	0	1	8	5	8	0	1	7	1	5	0	7	2	9	8	0	0
3	2	1	1	5	9	5	7	8	2	9	7	8	1	7	0	0	0	0	0	0
4	2	1	2	0	1	4	7	5	0	8	0	1	3	5	0	0	0	0	0	0
5	5	8	2	1	7	7	8	2	0	2	1	4	5	3	0	0	0	0	0	0
6	0	0	0	6	7	8	7	3	6	4	0	9	4	6	8	6	3	0	0	0
7	0	3	5	4	9	0	2	8	8	7	8	9	3	2	0	0	0	0	0	0
Xor	6	12	12	4	5	7	10	14	14	6	13	13	3	2	15	4	10	8	0	0

Figure (3.3): Example of Apply XOR Boolean Operation between 7 Moments Features (F) for User.

As shown in figure (3.3) the final result of Xor operation between 20 features is NewF= 6,12,12,4,5,7,10,14,14,6,13,13,3,2,15,4,10,8,0,0. Then convert it to decimal number.

II) **Apply SHA-256 Hash Algorithm Step:** in each user transactions stored hashing (*NewF*) moments feature that is a unique code that distinguishes them from other transactions. So, this step, produces SHA-256 hash based on the hash algorithm for the new feature (*NewF*) which is represents an addition to secure the transaction step. SHA-256 algorithm generates a fixed size 256-bit (32-byte) hash denoted as (H) that discussed in section (2.4.1.1). Figure (3.4) shows example for generate SHA-

256 hash (*NewF*). Where block (1) is loading as 7 moment features from dataset [41] and block (2) is processing of generate hash value.

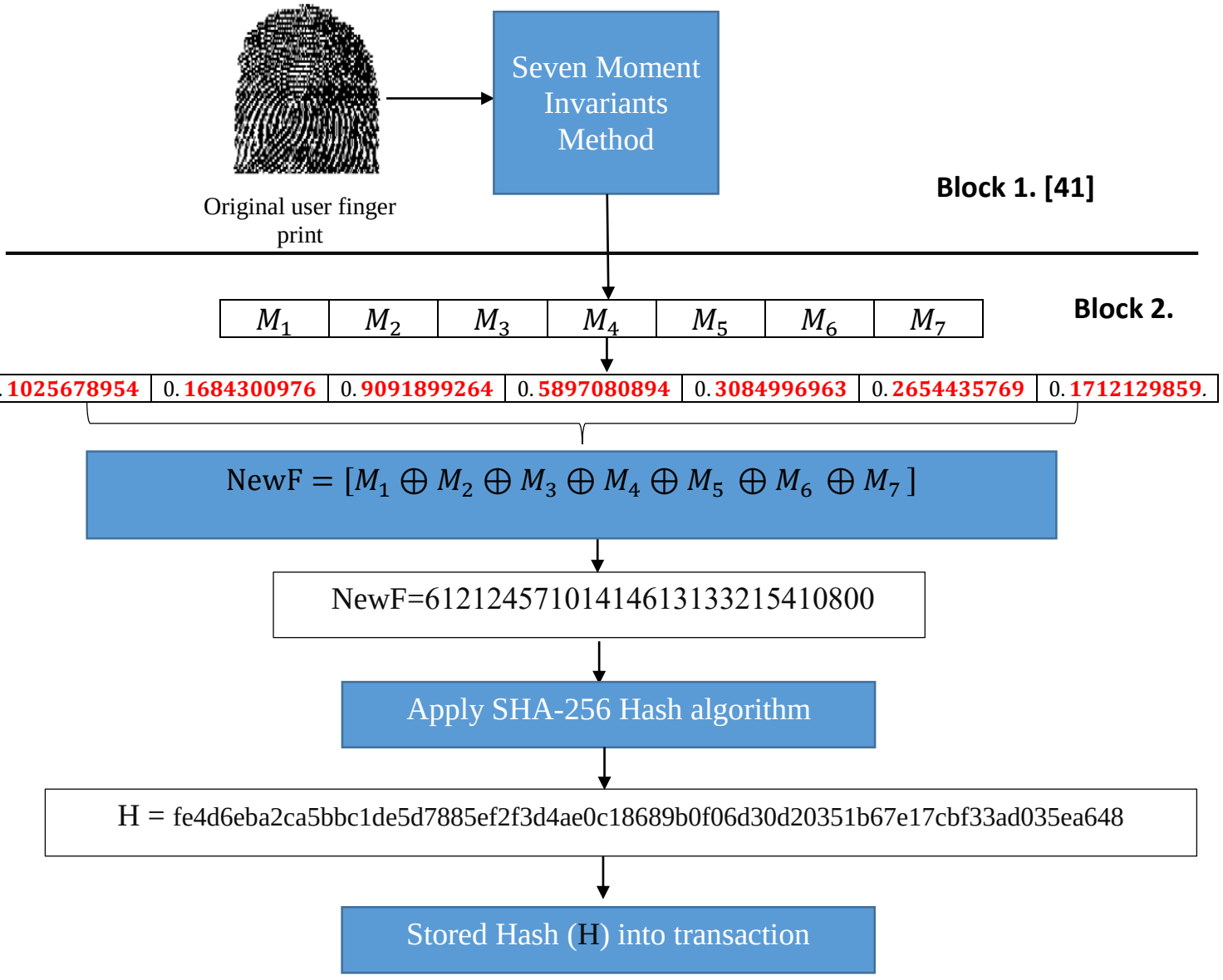


Figure (3.4): Example of the SHA-256 Hash Algorithm Step.

III) **Generating Key Pair (Public Key , Private Key) Step:** the key generation of RSA algorithm is based on the concept of finding two large prime numbers and then consequently finding the public key and private key ,this step aims to generate a large prim number based on linear congruence and big integer to convert the way of finding public and private key to linear methods as clarified in subsection bellow.

a) Generating Public key using Linear Congruential Generator (LCG): this step aims to generate a random number for the dynamic key using LCG, then test if the generated number is prime using the Miller-Rabin algorithm (2.2) in section two**.** To create the public key (PU) based on LCG is shown in algorithm (3.1).

Algorithm (3.1): Generating Public Key (PU) based on LCG method

Input: *X(n)* // a sequence of pseudo random values a // multiplier defined as 0 < a < m b // the increment 0 <= b < m *X* Prime // initial value m // modulo defined as 0 < m n // number of user
Output : *X*public (*n*, X_n)

Begin
Step 1 : Select seed (X_0)
Step 2 : For (n=1 to i)
While (*X* Prim =True)
Begin
X Prim =False

a= newRandom(m)

b= newRandom(m)

Calculate X_{n+1} by equation LCG (m, a, b, X_0) (2.8.2)

No= X_{n+1} % m

Swap (No , X_{n+1})

X Prim = Apply Miller Rabin algorithm(X_{n+1})

algorithm (2.2)

If *X* Prim =True

Cout ("True")

*X*public. add (*n*, X_{n+1})

Else cout ("False")

End while

Return *X*public (*n*, X_{n+1})

End For

Step3 : End Algorithm

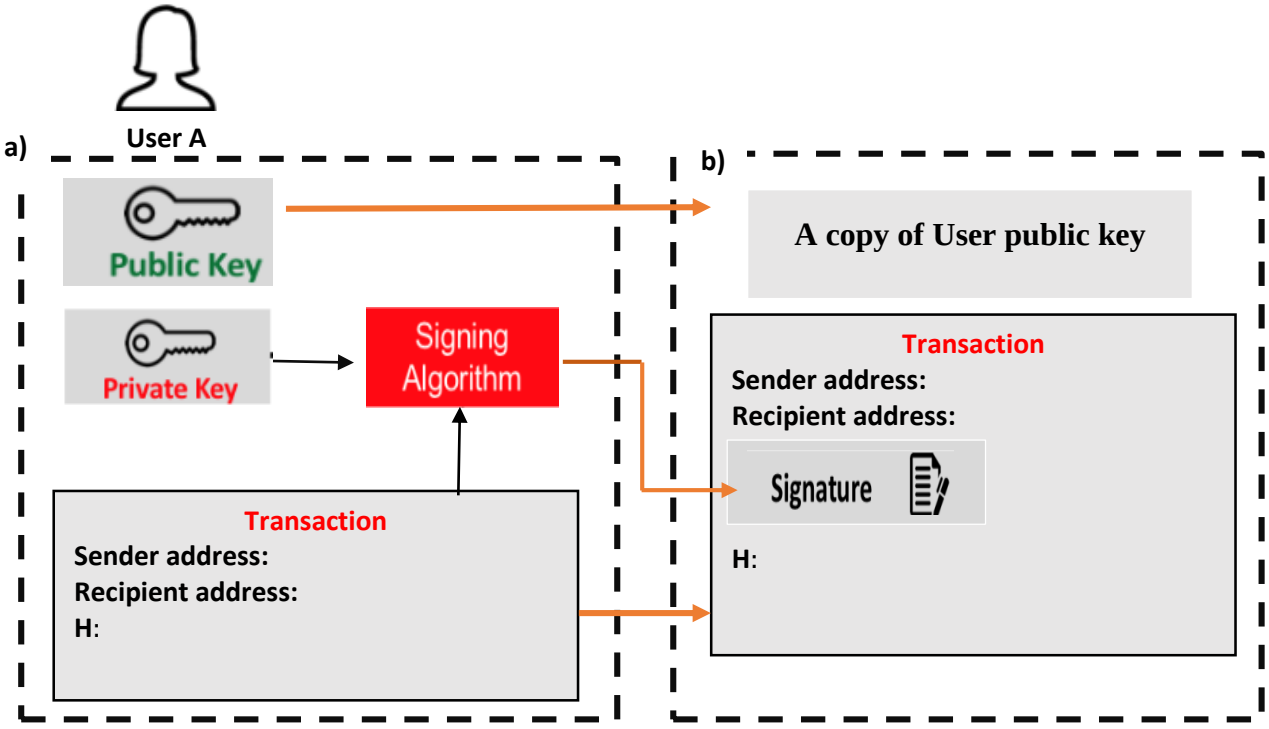
b) Generating Private Key (PK) based on Big integer: the big integer is Mono library for handling very large integers, up to 1024 binary digits, or approximately (safe to use) 3000 decimal digits. In this step, select for each user a private key (P) of big integer type. For example, to select user private key define it as big integer q.

IV) RSA Cryptography step: the fourth stage in generating transaction step of the proposed system is RSA cryptography. In this step, encryption SHA-256 hash (NewF) which denoted as (H) using the RSA algorithm. RSA algorithm is described in subsection (2.8.2), and implemented as illustrated in algorithm

(3.2). Figure (3.5) clarify block diagram of the Implemented RSA Algorithm to signature transaction.

Algorithm (3.2): Implemented RSA Algorithm

Input: Two prime number p, BigInteger q, where p ≠ q, NewF.
Output : Sender Signature
Begin: **Step 1:** Calculate Big integer n = p × q Calculate Big integer ϕ (n) = (p - 1) × (q - 1) Select integer Big integer e GCD (ϕ (n) , e) = 1; 1 < e < ϕ (n) Compute Big integer d ; $d \equiv e^{-1}(mod\ \phi\ (n))$ Public Key PU ={e,n} Private Key PK={d,n} **Step 2 :** //* encrypt plain SHA-256 hash(NewF) BigInteger M = H C = M^d mod n return toHex (C) **Step 3**: End Algorithm.



a) User generate a transaction and signs is with private key.

b) User sends to the ASBchain network the transaction and a copy of user public key.

Figure (3.5): Cryptography Process of Generate Transaction Stage.

Figure (3.5) illustrated the user transaction (T) obtain [{signature of the user using the private key, SHA-256 hash (NewF)}, the public key is declared], then a user broadcasted this transaction T to all nodes of the proposed ASBchain network.

3.3.2 Authentication Stage

The second stage in the proposed ASBchain system is an authentication stage based on SHA-256 algorithm and a strong cryptography RSA algorithm aim to securely verify from authentication of user transactions. After the user transaction is broadcast to all nodes in the ASB chain network, in this step, any node in the ASBchain network can verification from user transaction using user public key and SHA-256 hash

as shown in algorithm (3.3). Figure (3.6) shows flowchart of the authentication stage.

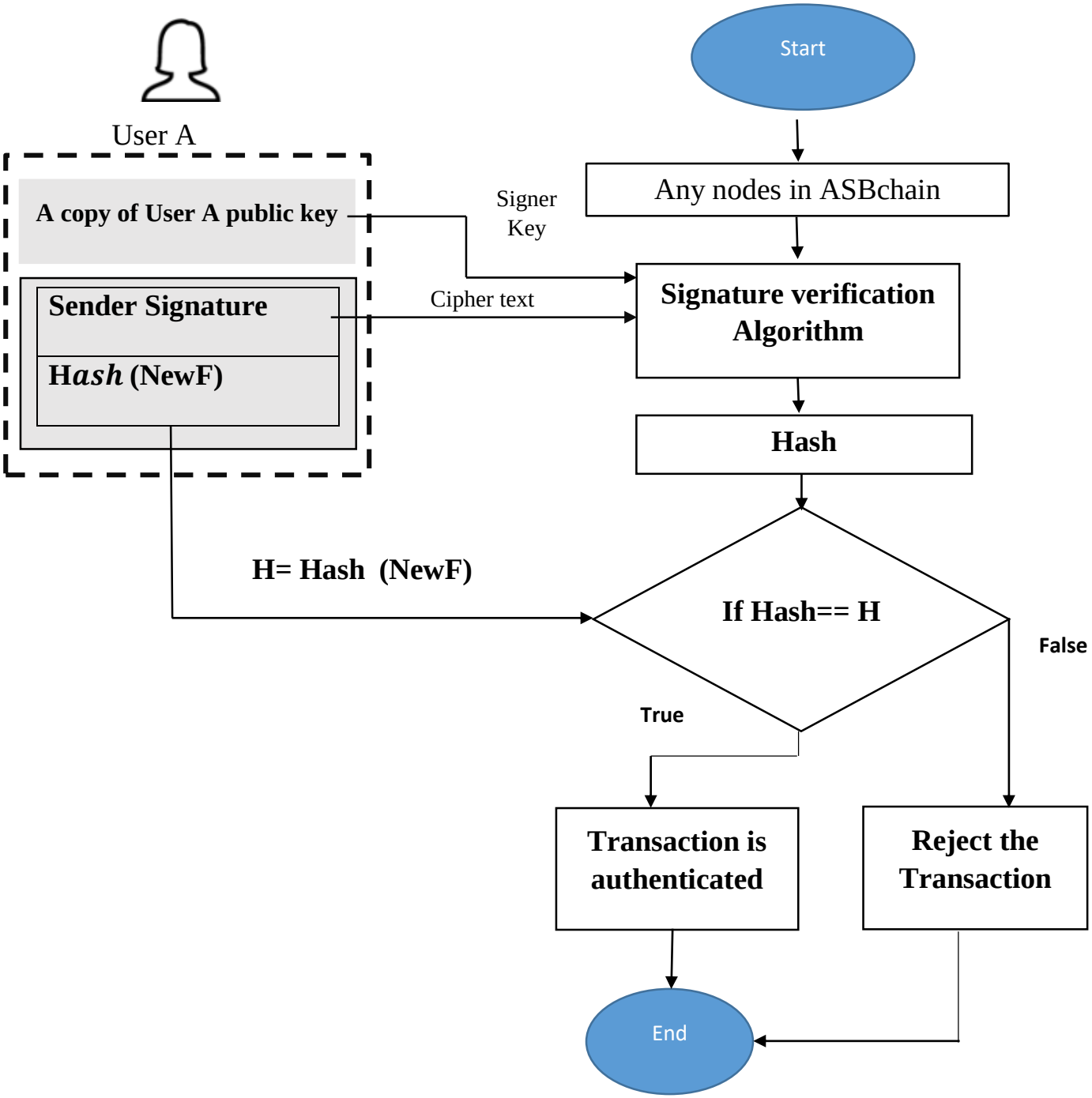


Figure (3.6): Flowchart of the Authentication Transaction

Algorithm (3.3) shows details of the authentication stage of the proposed system.

Algorithm (3.3): Authentication stage based on RSA algorithm

Input : User public key (PU) User transaction {signature, SHA-256 hash (NewF) }
Output: transaction is authenticate or not?

Begin
Step 1: Decryption signature using signer's public key.
NH = C^e mod n
Step 2 : If (SHA-256 hash(NewF) == NH **)**
Then Return " transaction is authencation "
Else Return " transaction is not authencation "
End If
Step 3 : End Algorithm

3.3.3 Builder Merkle Tree Stage

The builder Merkle tree stage in the proposed ASBchain mechanism provides a reliable environment through its summarizing all transactions in a block and storing it in a block header as current hash.

Merkle Root is a complete binary tree where each leaf considers the hashed value of the authentication user transaction T associated with that leaf. The branches are the hash of the series hashes of the two children. The process of re-hashing the concatenation of the child nodes to create the parent node is performed continuously until the top of the tree is reached, called the "Root Hash" as explain in algorithm (3.4).

Algorithm (3.4): Builder Merkle Tree Stage

Input : n //Here the number of transaction is the number of sender T // transaction node Hash // transaction data
Output: HMT // Hash Merkle Tree

Begin
Step 1 : while (level !=last level)
Begin
k=0
If (n mod 2==0) then
For (i=0 ;i<=n ; i+2)
Parent.Hash [k] = T.Hash[i] +T.Hash [i+1]
k++
end for
Else go to step 2;
Step 2 K=0
For (i=0 ;i<= n-1; i+2)
Parent.Hash [k] = T.Hash[i] +T.Hash [i+1]
K++
End for
Duplicated last node T
Parent.Hash [k] = T.Hash[n] +T.Hash [n]
End if
Step 3 If level = last level then
HMT = Parent.Hash [k]
End while
Step 5 : End algorithm

In algorithm (3.4), for each authenticates user transaction T extracted a hash value for this T and stored in array hash[k] , if the number of transaction T is even then the value of hash of the parent node is an equal hash of two T leaf node as shown in figure (3.7) . If the number of transaction T is odd, the proposed system used a new technique to keep the balance of the Merkle tree by duplicated the last leaf node and computed parent hash in same way as show in figure (3.8).

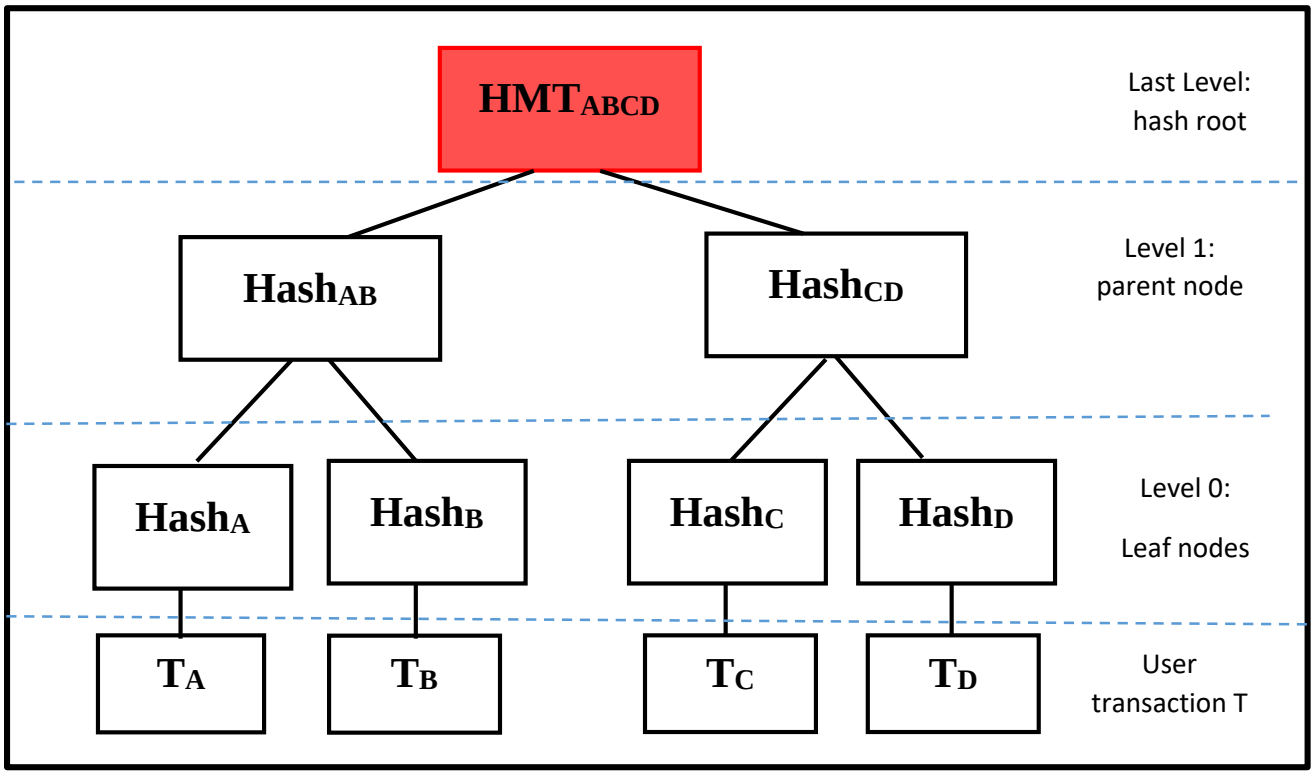


Figure (3.7): Merkle Tree with Even Number of Transaction Case.

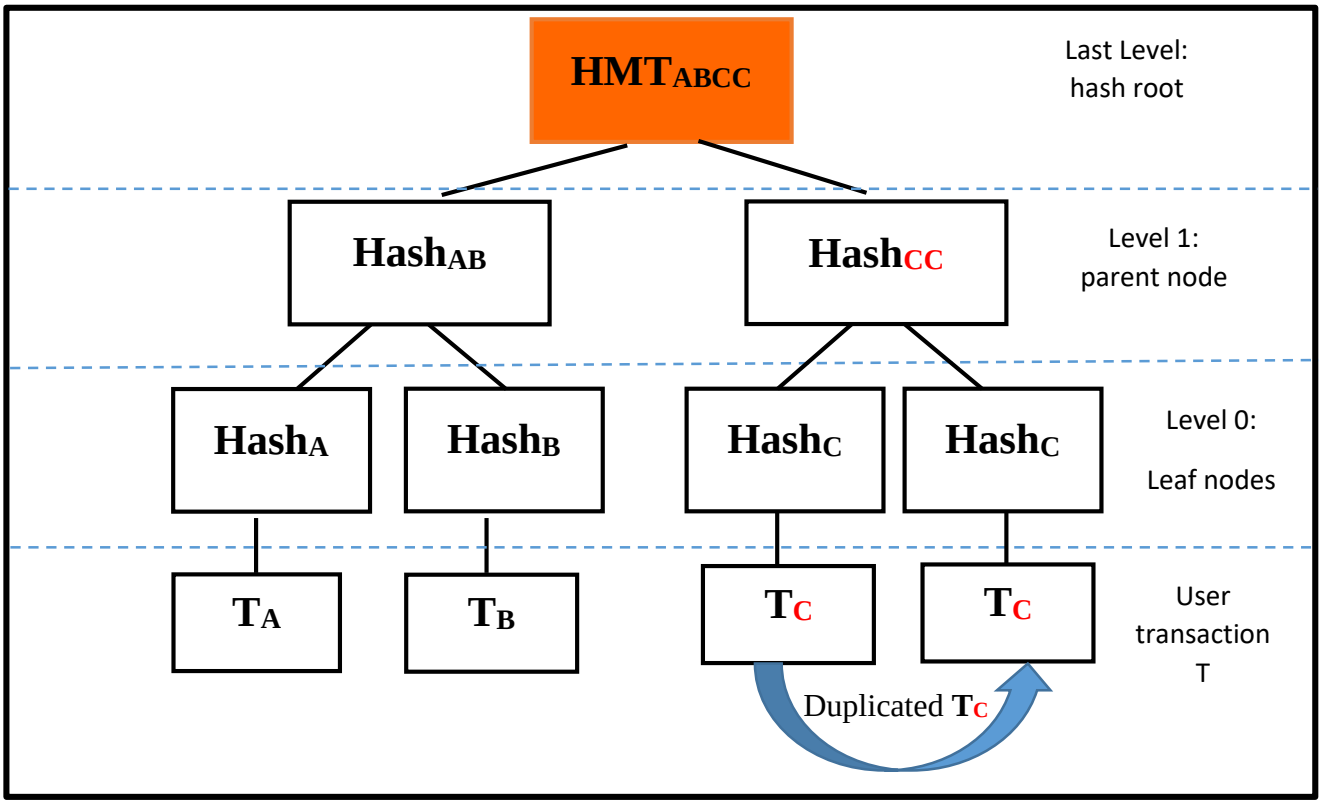


Figure (3.8): Merkle Tree with Odd Number of Transaction Case.

3.3.4 Create Blocks Stage

Each block in the Blockchain network consists of the block header and block body. In particular, the block header includes

I) Hash Merkle Tree (HMT): is a hash value of all transactions in the block that calculated in the previous section.

II) Time-Stamp: represents time and date for the transaction enrollment in the block.

III) Previous- hash: represents cryptography SHA-256 of the previous block. In particular, the first block in the Blockchain does not have value in previous hash, so this value equals "0".

IV) Current -hash: represents hash value for the current block header.

In creating blocks stage, calculated two values which are:

1- Compute the value of Current –hash by copy content of block header (Time-Stamp value, Previous - Hash value, HMT value), then by apply SHA-256 hash algorithm on block header contents to generate one hash value as shown example in figure (3.9).

2- Compute time-stamp value that represents the time to create the current block. Block time-stamp is differing from the time stamp it is located inside the block which is important issues that used in authorization stage.

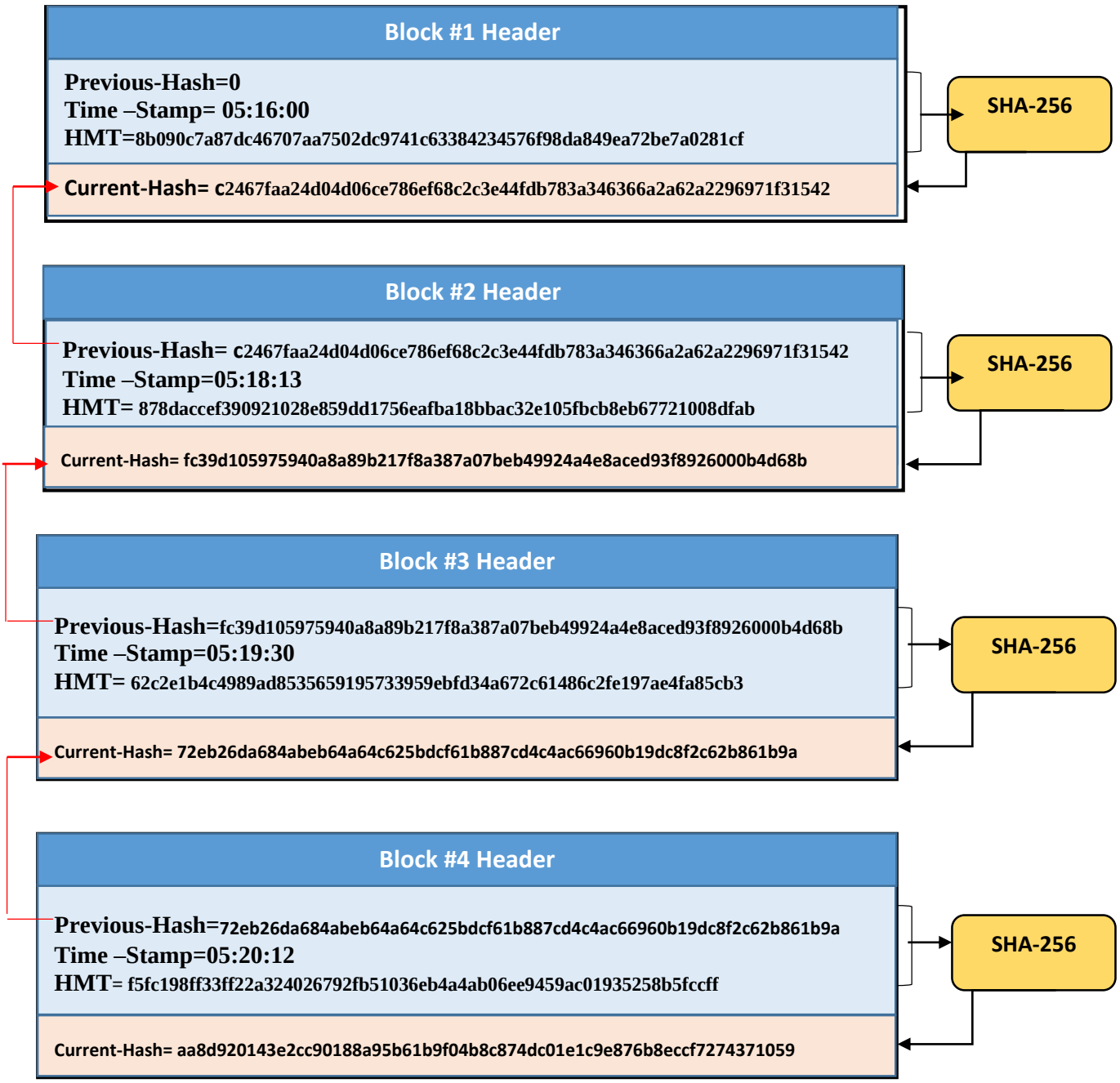


Figure (3.9): Example of the Create Block Stage.

3.3.5 Authorization Stage

This stage aim to determining whether a transaction user has authority to access to specific resources in the proposed ASBchain network. The details of authorization stage illustrated in algorithm (3.5). Figure (3.10) shows flowchart of the authorization stage.

Algorithm (3.5): Authorization Stage

Input: user request [Hash value for last block user transaction have, Time-stamp for block).
Output: User node authorization or not

Begin
Step 1: User sends a request to ASBchain network
Step 2: All node in ASBchain network receive user request and starts to verify from it.
Step 3 : Each node searches to find for Hash value for last block in the ledger (each node has the same ledger exactly as the database) Based on the time stamp of its creation.
Step 4 : If Hash value for last block that sending match with Hash value that stored in leger dataset
Then return " user is authorization "
Else return " user is not authorization "
End if
Step 5: End algorithm

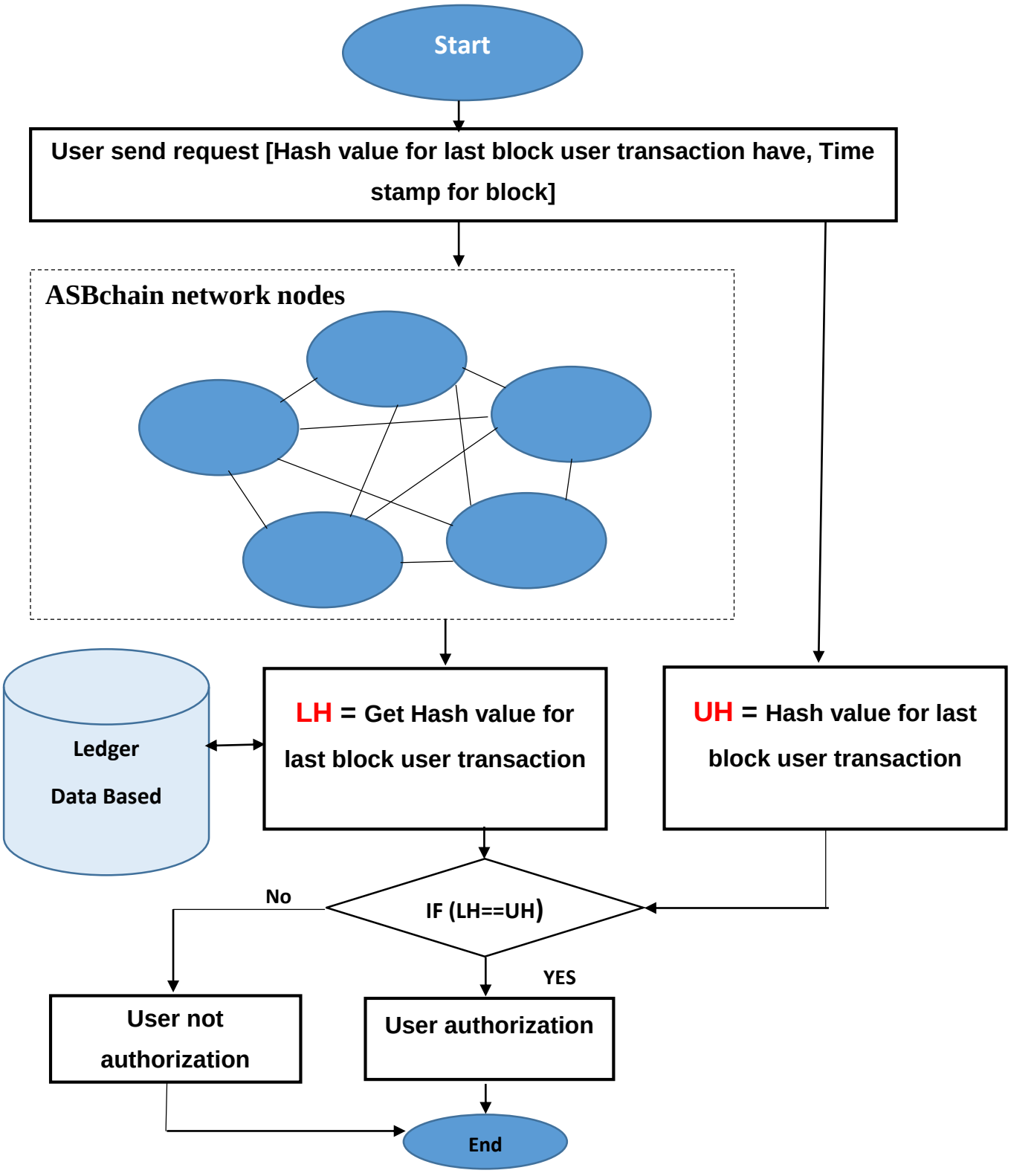


Figure (3.10): Flowchart of the Authorization Stage.

3.3.6 Linking block to ASBchain Stage.

The last stage in the proposed ASBchain network is linking blocks. After the block created in section (3.3.4) and check its authorization as shown in section (3.3.5) then this block distributed to all nodes in ASBchain network. So, this stage aim to connect the new blocks to ASBchain blocks based on the hash function (SHA-256) values that are calculated in equation (3.1):

$$\boldsymbol{New\ hash\ value_i = apply\ SHA256\ ((previous\ block\ hash_i)\ with\ (current\ block\ hash_i)).} \quad \text{Eq (3.1)}$$

New hash value denoted SHA-256 hashing for $block_i$, previous block hash and current block hash value represent SHA-256 hashing for $block_i$, where i=1,2,3…..,n of blocks. Figure (3.11) illustrated example of the linking blocks to ASBchain network stage.

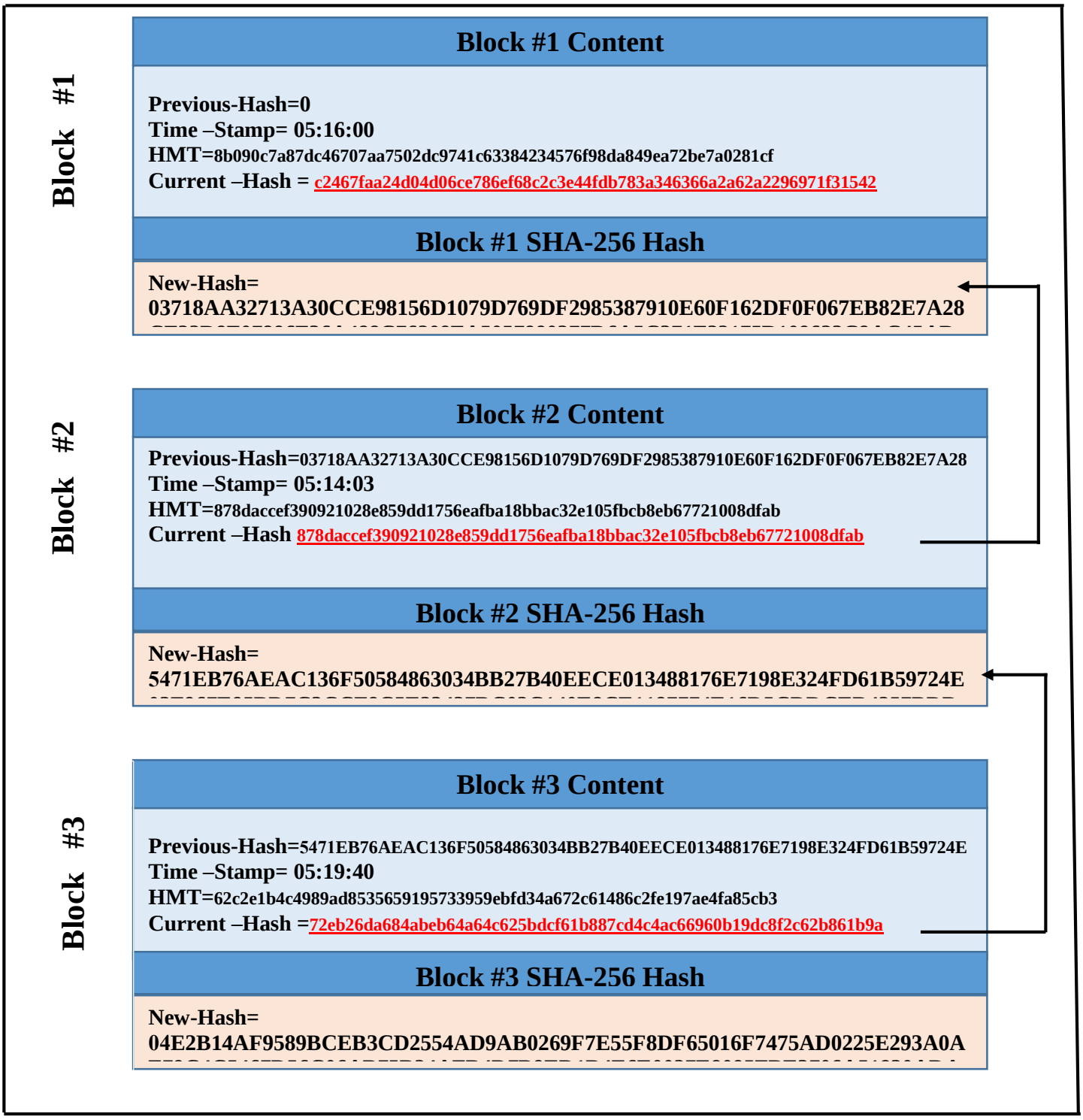


Figure (3.11): Example of the Linking Blocks to ASBchain Network Stage.

Chapter Four

Results and Analysis

Chapter Four

Results and Analysis

4.1 Introduction

The implementation of the proposed system is given in this chapter. Also, this chapter clarifies the results of the proposed Preservation Authentication and Authorization Based on Blockchain System (ASBchain).

Therefore, the initialization of the proposed system presented in section (4.2), implementation of the stages of the proposed system is given in section (4.3). The results of the proposed system are presented in section (4.4) while section (4.5) illustrated a comparison between all stages of the proposed system based on execution time.

4.2 Initialization

- The proposed system is implemented in the Java programming language (Net Beans) IDE 8.2 Ink using laptop computer. The examinations were informed on the processor Intel(R) Core(TM) i7-7700HQ @ 2.80GHz (8 CPUs),~2.8GHz64 bit operating System, and Memory 16 GB RAM.

- The database consists of 100 fingerprint features extraction that stored as features were extracted by using He’s Seven Moments method, which performed by of eight steps to extracted those [**41**].

4.3 Implementation of Proposed System

The proposed system has six main stages executed respectively as described in the following sections:

4.3.1 Implementation of the Registration Stage

The registration stage includes (normalize feature, generate SHA-256 hash, generate pair of keys for each user, and create digital signature), all these components used by the proposed system to create a transaction for each user.

In this stage, the system gets seven-moment feature for each user that request to creates transaction from the database and compute the execution time of creating a transaction for each user as shown in figure (4.1), when the number of user =100, length fixed=20, and key size =1024 bit, in this figure show the execution time measured in second, N is the module, e and d is the public and private keys respectively.

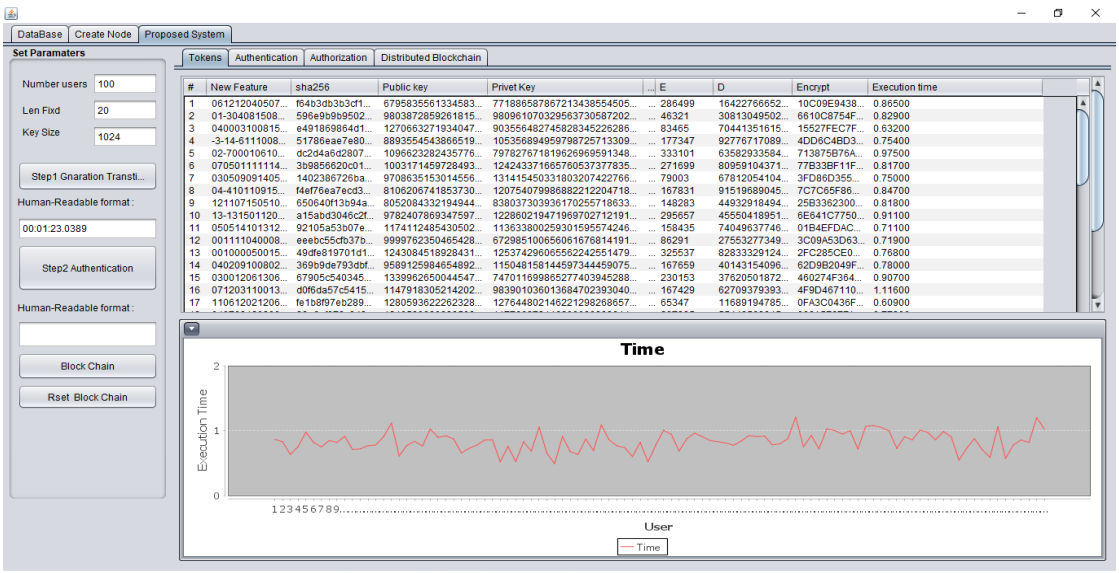


Figure (4.1): Implementation of Registration Stage.

4.3.2 Implementation of the Authentication Stage

The implementation of the authentication phase is shown in figure (4.2), starting with the proposed system interface that requires containing the transactions for each user (public key, digital signature,

and SHA-256 hashing). In this step, check authenticated transaction and compute its execution time for each user.

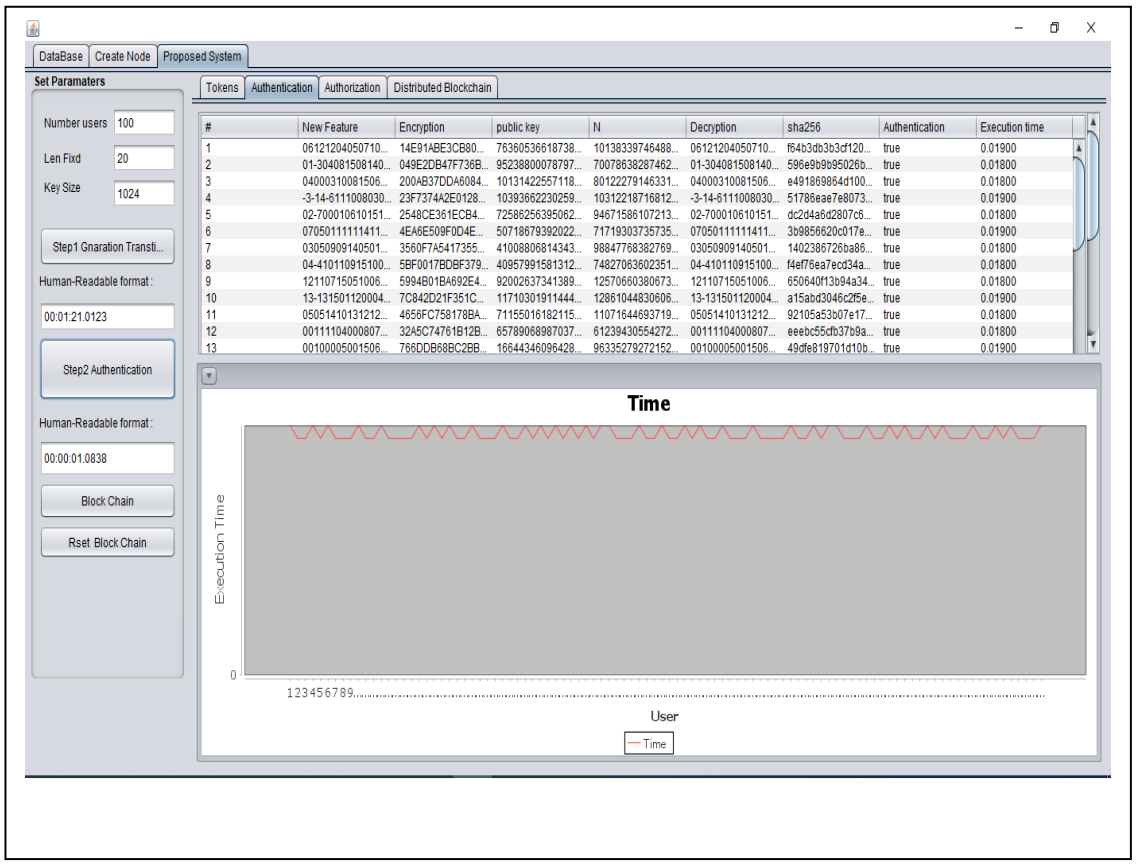


Figure (4.2): Implementation of Authentication Stage.

4.3.3 Implementation of the Builder Merkle Tree Stage

The implementation of the merkle tree phase is shown in figure (4.3), the proposed system interface of builder merkle tree for 100 user requires SHA-256 hash.

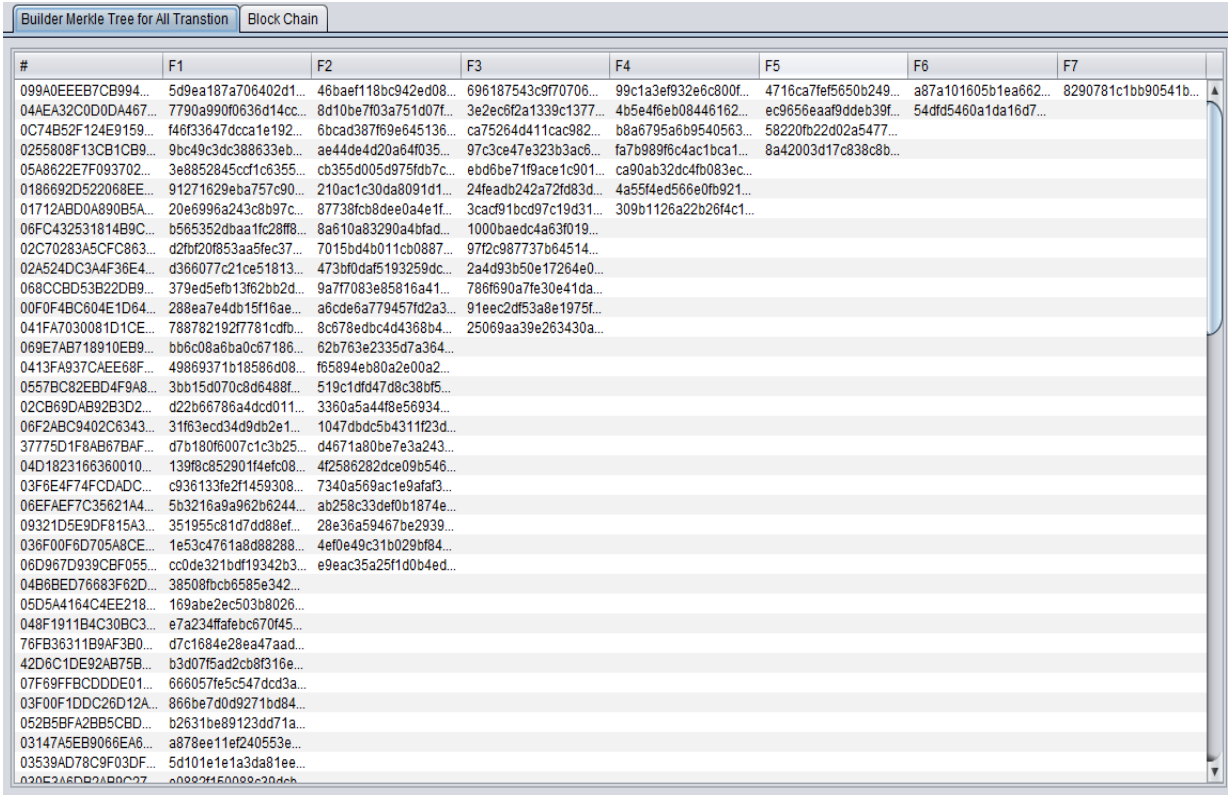


Figure (4.3): Implementation of Builder Merkle Tree Stage.

4.3.4 Implementation of the Create Blocks Stage

Each block in the proposed ASBchain system obtain block header, the block header consists from (Previous –Hash, Time-Stamp, Hash Merkle Tree (HMT), and Current –Hash). Figure (4.4) illustrate implementation of create blocks in the proposed system by computed current-hash value for block and time-stamp (date, time) for each block.

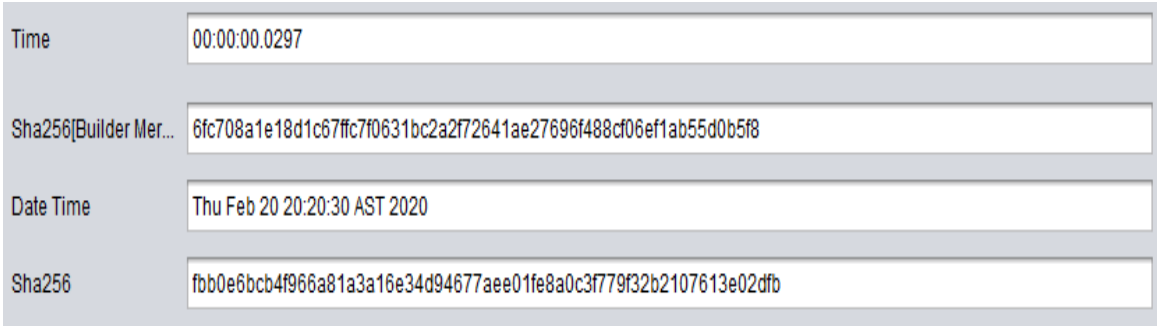


Figure (4.4): Implementation of Create Block Stage.

4.3.5 Implementation of the Authorization Stage

The proposed system node, will receive user requests based on its time-stamp. The authorization implementation includes (get hash value for last block user transaction from ledger databased and compared this value with hash value for last block user transaction that sending) to determined user is authorized to access into ASBchain resource or not as shown in figure (4.5).

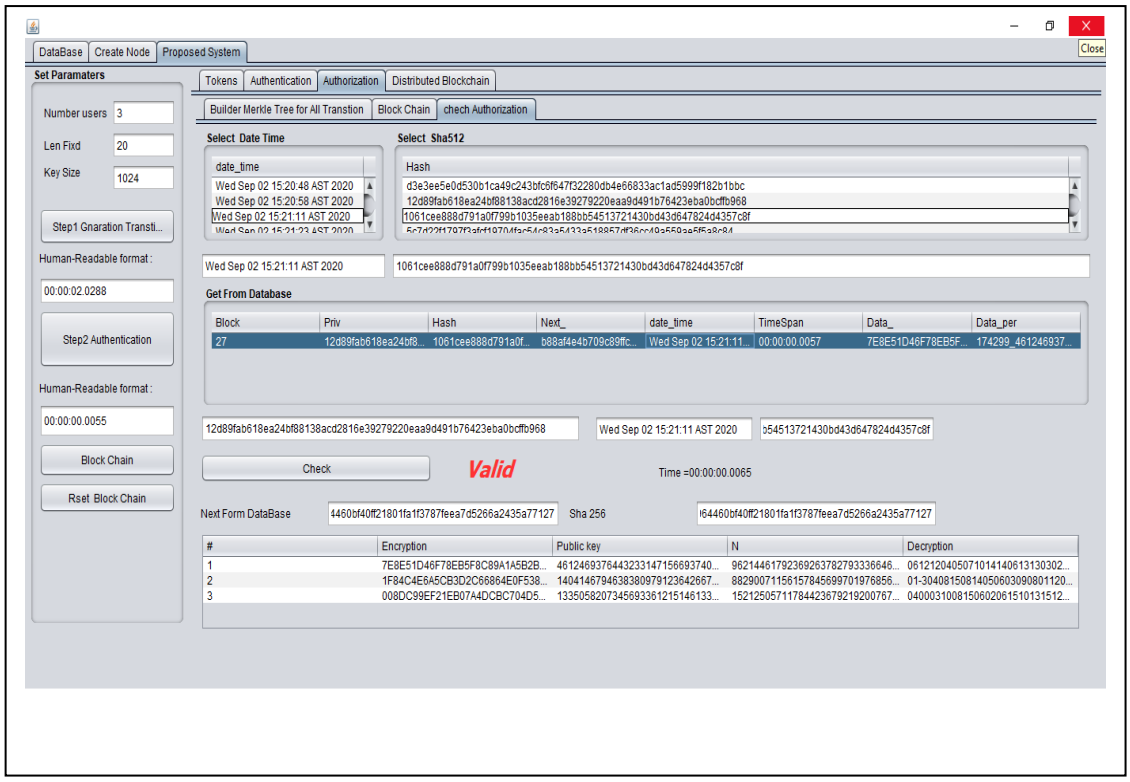


Figure (4.5): Implementation of the Authorization Stage.

4.3.6 Implementation of the Linking Blocks to ASBchain Stage

Figure (4.6) illustrated implementation of linking the blocks to ASBchain system based on SHA-256 hash value as illustrated in section (3.3.6), where this figure include three node A,B, and C, and each node have exactly the same ledger.

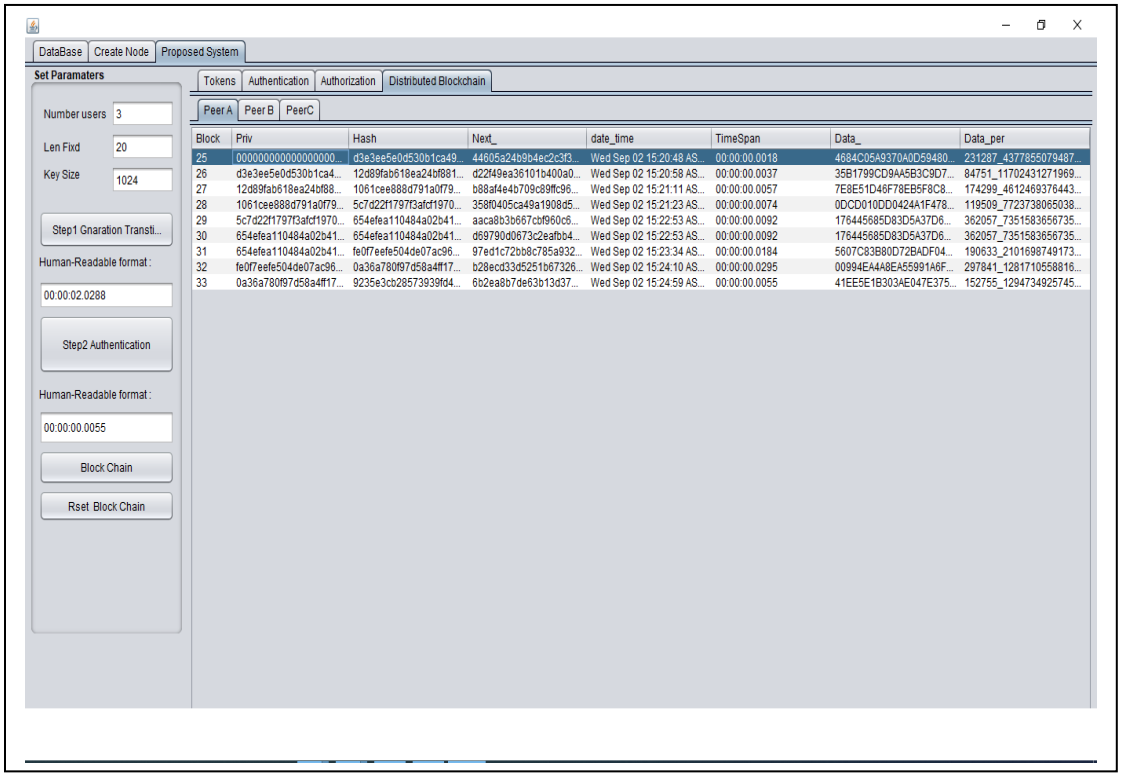


Figure (4.6): Implementation of the linking blocks to ASBchain Stage.

4.4 Results of the Proposed ASBchain Network

The results and execution time of each stage of the proposed system will show sequentially in subsections: (4.4.1, 4.4.2, 4.4.3, 4.4.4, 4.4.5, and 4.4.6).

4.4.1 Results of Registration Stage

The registration phase includes create transaction for each user request by follow sequentially several sub - steps (normalization feature, generate SHA-256, generating key pair (public, private), and RSA cryptography).

A) Results of Normalization Feature

This section present results of the two steps of normalization feature

Step 1: Load Seven Moment Feature of Finger Print Image

In this step, loading seven moment features of each user finger print from dataset [41] as shown in table (4.1).

Table (4.1): Original 7 Moment Feature of Fingerprint Image Dataset [41].

Users ID	M_1	M_2	M_3	M_4	M_5	M_6	M_7
1	0.3782512227 41672	1.85801715072987E-05	0.211595782978178	0.212014750801354	-.582177820214538	-.000678736409468635	0.035490288789322
2	0.536997266543598	0.000406861611365683	4.59682549580711	0.006017693287314 51	-0.00191951897426 209	8.73980186214392E-05	-0.0005886162790138 48
3	0.412386453986438	0.000615736186172554	0.427103053671894	0.30780312049948	-0.701869924722 877	0.00686381355275803	-0.0366165077359323
4	0.405235014879193	0.000998862577490055	2.39251710420598	0.59600226871872	-1.25687296953501	-0.0048216529716662	-0.340355650801672
5	0.333507253729667	0.000521079000075368	1.24460627216642	0.051057856113358 1	0.001534661799559 85	-0.000120222605743 099	-0.0111491827829651
6	0.478957996767475	0.00033409777340468	1.90641922524657	2.19214455366874	4.1414426941323	0.0234891623439423	4.38052237084485
7	0.38712085553853	1.47739030459287E-05	0.089587623526252 1	0.075851011380423 4	-0.0231220657073 342	0.000234092342182281	-0.0016118611006394 5
8	0.547623649554486	5.6246966847981E-05	0.072674651227317	1.43301334350531	-0.3709004921328 47	0.00842656349557505	0.363588884600107
9	0.600068034108333	0.00185905743085733	0.383473320560954	0.907239112668807	-1.85020676301429	0.0329373219563025	-1.04651823394038
10	0.52103011728249	0.00292189548088479	8.26076871047911	0.811497341072719	0.097011912024933 5	0.0419560384805041	0.83930615988946
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
100	0.331590032488931	3.37158439178903E-05	0.572676125236913	0.060875836873436	-0.072654295126136 7	-0.000316532738351697	-0.0013835381981756 4

Step 2: Generate New Moment Feature

The results of generate new moment feature through apply XOR-bitwise operation between 7 moment feature for each user, when length Fixed =20 and number of users =10 is shown in table (4.2). Table (4.3) illustrated generate (NewF), when length Fixed= 30 and number of users=5.

Table (4.2): Result of Generate (NewF) for 10 users.

User ID	Length Fixed =20																				
	No. of Feature	F1	F2	F3	F4	F5	F6	F7	F8	F9	F 10	F 11	F 12	F 13	F 14	F 15	F 16	F 17	F 18	F 19	F 20
1	1	3	7	8	2	5	1	2	2	2	7	4	1	6	7	0	0	0	0	0	0
	2	0	0	0	0	1	8	5	8	0	1	7	1	5	0	7	2	9	8	0	0
	3	2	1	1	5	9	5	7	8	2	9	7	8	1	7	0	0	0	0	0	0
	4	2	1	2	0	1	4	7	5	0	8	0	1	3	5	0	0	0	0	0	0
	5	5	8	2	1	7	7	8	2	0	2	1	4	5	3	0	0	0	0	0	0
	6	0	0	0	6	7	8	7	3	6	4	0	9	4	6	8	6	3	0	0	0
	7	0	3	5	4	9	0	2	8	8	7	8	9	3	2	0	0	0	0	0	0
	Xor	6	12	12	4	5	7	10	14	14	6	13	13	3	2	15	4	10	8	0	0
2	1	3	7	8	2	5	1	2	2	2	7	4	1	6	7	0	0	0	0	0	0
	2	0	0	0	0	1	8	5	8	0	1	7	1	5	0	7	2	9	8	0	0
	3	2	1	1	5	9	5	7	8	2	9	7	8	1	7	0	0	0	0	0	0
	4	2	1	2	0	1	4	7	5	0	8	0	1	3	5	0	0	0	0	0	0
	5	5	8	2	1	7	7	8	2	0	2	1	4	5	3	0	0	0	0	0	0
	6	0	0	0	6	7	8	7	3	6	4	0	9	4	6	8	6	3	0	0	0
	7	0	3	5	4	9	0	2	8	8	7	8	9	3	2	0	0	0	0	0	0
	Xor	6	12	12	4	5	7	10	14	14	6	13	13	3	2	15	4	10	8	0	0
3	1	4	1	2	3	8	6	4	5	3	9	8	6	4	3	0	0	0	0	0	0
	2	0	0	0	6	1	5	7	3	6	1	8	6	1	7	2	5	5	0	0	0
	3	4	2	7	1	0	3	0	5	3	6	7	1	8	9	0	0	0	0	0	0
	4	3	0	7	8	0	3	1	2	0	4	9	9	4	0	0	0	0	0	0	0
	5	7	0	1	8	6	9	9	2	4	7	2	2	8	7	0	0	0	0	0	0
	6	0	0	6	8	6	3	8	1	3	5	5	2	7	5	8	0	0	0	0	0
	7	0	3	6	6	1	6	5	0	7	7	3	5	9	3	2	0	0	0	0	0
	Xor	4	0	3	10	8	15	6	2	6	15	10	13	15	12	8	5	5	0	0	0
4	1	4	0	5	2	3	5	0	1	4	8	7	9	1	9	0	0	0	0	0	0
	2	0	0	0	9	9	8	8	6	2	5	7	7	4	9	0	0	5	0	0	0
	3	2	-2	3	9	2	5	1	7	1	0	4	2	0	5	9	0	0	0	0	0

	4	5	9	6	0	0	2	2	6	8	7	1	8	7	0	0	0	0	0	0	0
	5	-3	1	-2	2	5	6	8	7	2	9	6	9	5	3	5	0	0	0	0	0
	6	0	0	4	8	2	1	6	5	2	9	7	1	6	6	6	0	0	0	0	0
	7	3	4	0	3	5	5	6	5	0	8	0	1	6	7	0	0	0	0	0	0
	Xor	-3	-14	-6	11	10	8	3	1	15	2	4	13	7	7	10	0	5	0	0	0
	1	3	3	3	5	0	7	2	5	3	7	2	9	6	6	0	0	0	0	0	0
	2	0	0	0	5	2	1	0	7	9	0	0	0	0	7	5	3	6	0	0	0
	3	1	-2	2	4	4	6	0	6	2	7	2	1	6	6	4	0	0	0	0	0
	4	0	5	1	0	5	7	8	5	6	1	1	3	3	5	8	0	0	0	0	0
	5	0	0	1	5	3	4	6	6	1	7	9	9	5	5	9	8	0	0	0	0
	6	0	0	0	1	2	0	2	2	2	6	0	5	7	4	3	0	9	0	0	0
	7	0	1	1	1	4	9	1	8	2	7	8	2	9	6	5	0	0	0	0	0
5	Xor	2	-7	0	1	6	10	15	13	15	7	0	5	8	5	6	11	15	0	0	0
	1	4	7	8	9	5	7	9	9	6	7	6	7	4	7	0	0	0	0	0	0
	2	0	0	0	3	3	4	0	9	7	7	7	3	4	0	4	6	0	0	0	0
	3	1	-2	9	0	6	4	1	9	2	2	5	2	4	6	5	0	0	0	0	0
	4	2	-2	1	9	2	1	4	4	5	5	3	6	6	8	7	0	0	0	0	0
	5	4	-2	1	4	1	4	4	2	6	9	4	1	3	2	0	0	0	0	0	0
	6	0	2	3	4	8	9	1	6	2	3	4	3	9	4	2	0	0	0	0	0
	7	4	-2	3	8	0	5	2	2	3	7	0	8	4	4	8	0	0	0	0	0
6	Xor	7	5	1	11	11	14	11	11	1	10	7	10	12	11	12	6	0	0	0	0
	1	3	8	7	1	2	0	8	5	5	5	3	8	5	0	0	0	0	0	0	0
	2	0	0	0	0	1	4	7	7	3	9	0	3	0	4	5	9	2	8	0	0
	3	0	8	9	5	8	7	6	2	3	5	2	6	2	5	2	0	0	0	0	0
	4	0	7	5	8	5	1	0	1	1	3	8	0	4	2	3	0	0	0	0	0
	5	0	2	3	1	2	2	0	6	5	7	0	7	3	3	4	0	0	0	0	0
	6	0	0	0	2	3	4	0	9	2	3	4	2	1	8	2	2	8	0	0	0
	7	0	0	1	6	1	1	8	6	1	1	0	0	6	3	9	4	0	0	0	0
7	Xor	3	5	9	9	14	5	1	8	2	15	13	8	7	11	11	15	10	8	0	0
	1	5	4	7	6	2	3	6	4	9	5	5	4	4	8	0	0	0	0	0	0
	2	0	0	0	0	5	6	2	4	6	9	6	6	8	4	7	9	8	0	0	0
	3	0	7	2	6	7	4	6	5	1	2	2	7	3	1	0	0	0	0	0	0
	4	1	-2	4	3	3	0	1	3	3	4	3	5	0	5	3	0	0	0	0	0
	5	3	7	0	9	0	0	4	9	2	1	3	2	8	4	0	0	0	0	0	0
	6	0	0	8	4	2	6	5	6	3	4	9	5	5	7	5	0	0	0	0	0
	7	3	6	3	5	8	8	8	8	4	6	0	0	1	0	0	0	0	0	0	0
8	Xor	4	-4	10	11	9	15	10	1	8	9	8	7	3	11	1	9	8	0	0	0
	1	6	0	0	0	6	8	0	3	4	1	0	8	3	3	0	0	0	0	0	0
	2	0	0	1	8	5	9	0	5	7	4	3	0	8	5	7	3	0	0	0	0
9	3	3	8	3	4	7	3	3	2	0	5	6	0	9	5	0	0	0	0	0	0

	4	9	0	7	2	3	9	1	1	2	6	6	8	8	0	0	0	0	0	0	0
	5	-3	1	-2	8	5	0	2	0	6	7	6	3	0	1	4	2	0	0	0	0
	6	0	3	2	9	3	7	3	2	1	9	5	6	3	0	2	0	0	0	0	0
	7	-3	1	-2	0	4	6	5	1	8	2	3	3	9	4	0	3	0	0	0	0
	Xor	12	11	7	15	5	10	6	6	14	10	3	6	0	6	1	2	0	0	0	0
	1	5	2	1	0	3	0	1	1	7	2	8	2	4	0	0	0	0	0	0	0
	2	0	0	2	9	2	1	8	9	5	4	8	0	8	8	4	7	0	0	0	0
	3	8	-2	2	6	0	7	6	8	7	1	0	4	7	9	1	0	0	0	0	0
	4	8	1	1	4	9	7	3	4	1	0	7	2	7	1	0	0	0	0	0	0
	5	0	9	7	0	1	1	9	1	2	0	2	4	9	3	3	0	0	0	0	0
	6	0	4	1	9	5	6	0	3	8	4	8	0	5	0	4	0	0	0	0	0
	7	8	3	9	3	0	6	1	5	9	8	8	9	4	0	0	0	0	0	0	0
10	Xor	13	-13	15	1	12	0	4	3	7	11	5	9	4	3	2	7	0	0	0	0

Table (4.3): Result of Generate (NewF) for 10 users.

NO	Length Fixed =30																																		
	no. of feature	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25	F26	F27	F28	F29	F30
1	1	3	7	8	2	5	1	2	2	2	7	4	1	6	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	1	8	5	8	0	1	7	1	5	0	7	2	9	8	0	0	0	0	0	0	0	0	0	0	0	0
	3	2	1	1	5	9	5	7	8	2	9	7	8	1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	2	1	2	0	1	4	7	5	0	8	0	1	3	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	5	8	2	1	7	7	8	2	0	2	1	4	5	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	6	7	8	7	3	6	4	0	9	4	6	8	6	3	0	0	0	0	0	0	0	0	0	0	0	0	0
	7	0	3	5	4	9	0	2	8	8	7	8	9	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Xor	6	12	12	4	5	7	10	14	14	6	13	13	3	2	15	4	10	8	0	0	0	0	0	0	0	0	0	0	0	0
2	1	5	3	6	9	9	7	2	6	6	5	4	3	5	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	4	0	6	8	6	1	6	1	1	3	6	5	6	8	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	4	-2	5	9	6	8	2	5	4	9	5	8	0	7	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	6	0	1	7	6	9	3	2	8	7	3	1	4	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	1	9	1	9	5	1	8	9	7	4	2	6	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	8	7	3	9	8	0	1	8	6	2	1	4	3	9	0	0	0	0	0	0	0	0	0	0	0	0
	7	0	0	0	5	8	8	6	1	6	2	7	9	0	1	3	8	4	0	0	0	0	0	0	0	0	0	0	0	0	0
	Xor	1	-3	4	8	15	8	14	5	6	3	9	8	1	12	0	15	15	9	0	0	0	0	0	0	0	0	0	0	0	0
3	1	4	1	2	3	8	6	4	5	3	9	8	6	4	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	6	1	5	7	3	6	1	8	6	1	7	2	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0

	3	4	2	7	1	0	3	0	5	3	6	7	1	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	3	0	7	8	0	3	1	2	0	4	9	9	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	7	0	1	8	6	9	9	2	4	7	2	2	8	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	6	8	6	3	8	1	3	5	5	2	7	5	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	7	0	3	6	6	1	6	5	0	7	7	3	5	9	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Xor	4	0	3	10	8	15	6	2	6	15	10	13	15	12	8	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0
. . .		. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
10	1	4	0	5	2	3	5	0	1	4	8	7	9	1	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	9	9	8	8	6	2	5	7	7	4	9	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	2	-2	3	9	2	5	1	7	1	0	4	2	0	5	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	5	9	6	0	0	2	2	6	8	7	1	8	7	0	0	0	0	0	00		0	0	0	0	0	0	0	0	0	0
	5	-3	1	-2	2	5	6	8	7	2	9	6	9	5	3	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	4	8	2	1	6	5	2	9	7	1	6	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	7	3	4	0	3	5	5	6	5	0	8	0	1	6	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Xor	-3	-14	-6	11	10	08	03	01	15	02	04	13	07	07	10	00	05	00	00	00	00	00	00	00	00	00	00	00	00	00

A) Results of Generate SHA-256 Hash for New Feature

The results of generate SHA-256 Hash for a new moment feature (NewF) are illustrated in table (4.4), where number of users =10.

Table (4.4): Result of Generate SHA-256 Hashing of New Moment Feature

No. Transaction	NewF	SHA- 256 [NewF]
1	6121245710141461313321541080 0	fe4d6eba2ca5bbc1de5d7885ef2f3d4ae0c18689b0f06d30d20351b67e17cbf3
2	1-348158145639811201515900	75a0bed06fb0868ab81b4a0cab41ebb76f489b1d0eb3256ed8a5379a0a0d4b56
3	4031081562615101315128550 00	bdc50edb0e6cf4258d82429915c5a6aa6015790e49f736b453d29ae6f2c1bcd6
4	-3-14-61110831152413771005000	c5ab1f56b466011ce9ba0f585f7a27b7e0f46ef81d2b28a0a5a9f45c506ca7b0
5	2-70161015131570585611150 00	2f1b2c41b2c9a86076acbc251fdb3dbc761c5e6063122ab4ee94280b9174fc2d
6	7511111141111110710121112600 00	242726c1765be5cb96e4f229a4291e4020ce1b029964d1f6bf96e78d623c5224

7	359914518215138711111510800	56f2efd9021d03b0276a072701b3ad7769bca0970c844feab02f0fdb43d290fb
8	4-410119151018987311198000	262c3f5b42ac11e6ef9c3d0dc6aec0ef3d341e11c0047eee556e8464361cd17a
9	12117155106614103606120000	4cb20e4f2f24a38f40b03876f846906ab3d96b06f13faa9aca30dfbfe73b1644
10	13-13151120437115943270000	36f592113dd9150053760636c65fcbc1cf07ba1b0246426d2a32069ae7e4cc7c

B) Results of Generate Pair of key (Public , Private)

The result of generating prime number using LCG algorithm (2.8.2) with Rabin miller testing to generate public key for each user as illustrated in algorithms (3.1) and (3.2) is shown in table (4.5), Where parameter of LCG (a) and (b) are chosen randomly and x=200, m=1000, and iteration =100 and table (4.6) Where parameter of LCG are x=100, m=100 and iteration =50.

Table (4.5): Results of Generate Prim No. with Miller - Rabin Prime Test**.**

No.	X	a	b	(x*a)+b/m	Result of test
0	200	99	784	20584	FALSE
1	584	375	548	219548	FALSE
2	548	813	678	446202	FALSE
3	202	853	378	172684	FALSE
4	684	448	706	307138	FALSE
5	138	200	6	27606	FALSE
6	606	11	656	7322	FALSE
7	322	661	202	213044	FALSE
8	44	99	722	5078	FALSE
9	78	643	981	51135	FALSE
10	135	763	123	103128	FALSE
11	128	968	641	124545	FALSE
12	545	380	929	208029	FALSE
13	29	599	805	18176	FALSE
14	176	220	937	39657	FALSE
15	657	939	718	617641	FALSE
16	641	334	896	214990	FALSE
17	990	589	440	583550	FALSE
18	550	256	989	141789	FALSE
19	789	624	635	492971	FALSE
20	971	732	231	711003	FALSE
21	3	18	381	435	FALSE

22	435	398	376	173506	FALSE
23	506	849	789	430383	FALSE
24	383	854	985	328067	TRUE
25	67	317	246	21485	FALSE
. . .	. . .	. . .	. . .	. . .	. . .
100	34	83	88	2910	FALSE

Table (4.6): Results of Generate Prim No. with Miller - Rabin Prime Test**.**

No.	X	a	b	(x*a)+b/m	Result of test
0	100	34	49	3449	TRUE
1	49	16	39	823	TRUE
2	23	53	73	1292	FALSE
3	92	39	36	3624	FALSE
4	24	2	87	135	FALSE
5	35	31	16	1101	FALSE
6	1	6	21	27	FALSE
7	27	39	53	1106	FALSE
8	6	5	82	112	FALSE
9	12	30	70	430	FALSE
10	30	53	23	1613	TRUE
11	13	44	24	596	FALSE
12	96	43	24	4152	FALSE
13	52	67	84	3568	FALSE
14	68	2	39	175	FALSE
15	75	40	94	3094	FALSE
16	94	26	81	2525	FALSE
17	25	34	65	915	FALSE
18	15	26	80	470	FALSE
19	70	5	38	388	FALSE
20	88	72	97	6433	FALSE
. . .	….	….	….	….	….
100	20	100	17	2017	TRUE

Table (4.7) and figure (4.7) shows the comparison of the result of Rabin Miller test based on number of (True) and (False) between results of table (4.5), when x=200, m=100, a=50, b=3, and Iteration=100 which is

represent **case 1** and table (4.6), when x=200, m=100, a=100, b=50, and Iteration=100 which is represent **case 2** .

Table (4.7): Comparison of the Result of Rabin Miller Test based on No. (True) and No. (False).

##	No.(True)	No.(False)
Case 1	15	85
Case 2	10	90

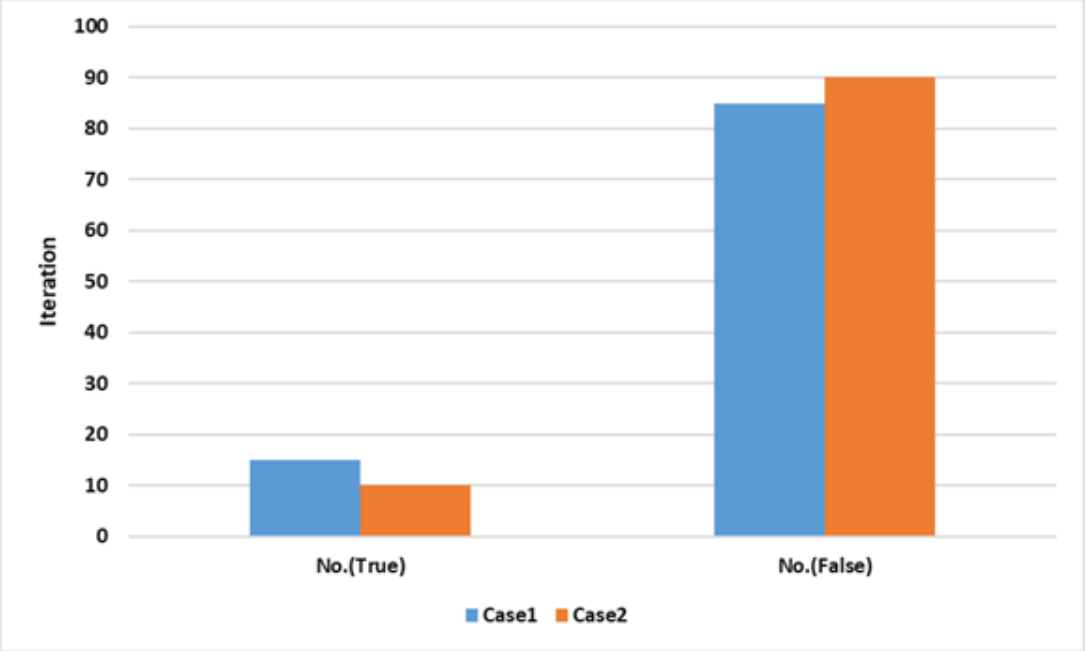


Figure (4.7): Comparison between Case1 &Case2 based on No. (True) & No. (False)

Table (4.8) shows result of generate pair of key (private, public) for 10 users with key size = 1024 and table (4.9) shows result of generate pair of key (private, public) for 5 users with key size = 250, where private key generate using Big-integer method as shown in section (3.3.1.2).

Table (4.8): Result of Create Pair of Key, when user=10 and key Size =1024**.**

User id	User Key	
	Public Key	Private Key
1	11233	1569268038312379044204830867448670793544302250882788852367064993071441618396985834918231691796909270158764340876822233204193006762572885555857669214943472338858867578797190274967305 3
2	113693	7727622594406256206528518465175062713914281190365777252371574306388405862825346085869953608966560113269638299183476212059202317
3	135389	548929657065956482544364290088430656336285935754145808756977306919593571717743833627767456913620533452139488860661412772567070 7
4	299317	1225688238304425504087858918238053887791440478261289770763599933793409979863755322924389268364803364183961933946120531742587849449108634405187456073003008433843122522525497272985620 9
5	277015	8405095801329160759377992493390246976112166220867404998875599680059495104996000291950105099339281053104205227921021896356442841354077949238592907486808868477102042916836185696946087
6	232037	1125547289798620729482773501738200779939934533678490545883915450754199237003237893212932365936766544931833092392869237633689085650410957647465974645616217986382834734624390556612356 7
7	165359	1067791514372159556853244691844301127869557836639328518592425638911177699189810629729971434881207807427922839043566849741862652148567719230413017845159974107713552067034155783995294 3
8	119959	1187812775887069612955453351875325194587227574853109824332807469919935202884426737887450248648141481500060754276837083279586708753124483831340180124339844068008095631262746932447217 7
9	221641	1181633288858649580106067939522868838574025640950269506999175320790290647036923228316044905306067333680152962884444527742821284254373462326684921373528843262107925354344005394493315 9
10	205013	1513678822281224877985933922857340501200927166756816548987803142989226018666454839761683658767691723516839614433044539515030249353869469624310705268332290333786185274952787480720561 7

Table (4.9): Result of Create Pair of Key, when user=5 and key size=250**.**

User id	User Key	
	Public Key	Private Key
1	84257	74876736096800066159811704522977236454135045975826104831611717888986098139 3
2	106553	66861023084806351721908796139579047069785368457417013792649852503108763721 7
3	219915	54021976323081424017667591481151120018847562702709563525956774338748510179 5
4	33895	43871238564991873785509100128324681367295310856557317825115230113256139525 5
5	33895	43871238564991873785509100128324681367295310856557317825115230113256139525 5

C) **Results of RSA Cryptography to Generate User Signature**

Table (4.10) shows the result of ciphering SHA-256 hash value of new feature using RSA algorithm by private key for 10 users. Figure (4.8) illustrated the execution time in second of the 10 user signature that show in table (4.10).

Table (4.10): User signature using RSA algorithm with execution time in (second)

User ID	User signature	Execution Time in (Sec)
.1	0275C1CAAC2FF576D0326D903111E79A22831D4B5B5B3427FF148D20F71394DD2A8C7376084E177 0C1342C65BA97E399FE845EFF40CA1623FEDD5465D3C0BFFED8885E14D54A5D377485E16C03F9 A80CD9DEF11703BBF6CAAB61BB15DAA2CCA58A251F27C4B4A8D6FD53B50F86545EA271DF08A ED22B267405B888BC7279A20F8B016CBB61FEB05423FF3B6CDA1D6CD80490CFAB615813	1.25000 s
.2	065077757465B9F76D3CB71A03940BAE8969BC588A9F479FB4DC49E9B51278DB44B3A5F9B9A1DA	1.04700 s

	C81B47D01CC42096894CBE725D15922CAB51A006DC08E8A37AD52921677D573211848E84CFC08A11AFB8BEB0DF8E0021D213D4E07565D89B42D801F18666F21AD8DCDBFD858A5DB093F3FDB8C50A693FBCDBA3CD3B2172CD71DD5D49524689DBD9BEF489BE0F5E0C9DCD6C6D83EF49D	
.3	**082B70A2CF03CB3B748A7A805094856772A94F37A2EEE0C8A0D673421C280DA88E107B90A81C5C35CA37B583C71B54DB6A2531A307AF0B1309AC60510FD37C9FD9E6AD5AB651192EE4F73463DCD1A2767E4F381FFDC35CCAAB17FE7121D2A43F961AF154FF3FEE148E6454963BC390E29E36FC4AC21DEF110ADF3759CB9C9466E4779AC2BBEE9FB836B2B7231116A48B5BD6D3464C5DE5**	**1.00000 s**
.4	**037B18A8C298AC6C55E8F0B009C93E57B6A3EAE308CE005053B182FCB63C3AD62561F8DEA03FF87C9A0F9A82A17B83ECF13D402235E34B0355F828654308B7822F1238CF19305907A88CDB47FF38017BACCB9B616FD9E77CA1A0375E1E025BDBAD559A30871C9D11F6F19218ED3EA53E7C732976E07A01833D51BF0A11F013F3B91836615B8DC0A8A2DA220AC642A282D55B5B0042E52D**	**2.35900 s**
.5	**018131C737E401162096947800A1F938E98338E2E8FA8A92734F53935F7DD77974AF10BB2F63996EABF95A184F6D22E356EB6E9C723B5C27244B27B6EF8FDA8AC94B2BC64574F6AF6FE27EB57A150FA3E2FB97F26A53C5B883819720B3EC90FE83E30027AE3DC01F684B78BEBCF1C0CA251131B839C25F8A8880C70E4A224F5ED9E52571BAE3AAAE23042C9C6519CC87A3C4393267FEBD**	**0.60900 s**
.6	**018225E9FD326077A79D69352081BB13E100F4FBE20E1BFB27AD4965AC0F7C05048560795587B0D1AFB3BFA00D70DA0F18D9C17F41F2B03D674A95A1821DE2662BAE5720C756E90D1FB095F3121CAF8978EE6CCAF03ED4813DFED0F1C8C3D174DB0004200E000CFDE3B379E9B7D621C0D003911C9023CDADBD85C865293F40B5206B4AD48358758754F6EE64218566A03E401FF1C9D2D4**	**1.18700 s**
.7	**018C856C7146F48152B4099934E88878F6BBA8F9F24C3F927DF941725B0AE851CBB476956EF72D6BD3CD22A0CB4349F4618E2E17900F8C1EB7E5E9694A862565C3B3493C8FC9267EF0080B7C62044B9FA750D57B6D7A49F26BDB183BE379821F14C6820385AA08590D30D007CAA19FCA84AD3A0FDCABE688183ED22404647D89CFC61484F8E34F2C8FDFEE7D8882B568E466BC521C3466**	**1.20300 s**
.8	**06DEFF037DE0A8380FCC570EDE64B4C37A17EC6D47B76F184D999FCCB4C40123560748B13171134918BE394D11582ACA5EDE097C507FAA8523A32A3E096D00BB0A685CC7796636A6A7E29B77DCBCB494DEEA2453A1D3327516FE3FB2CE05A58247D9573B453D6C99013A7C9395BCFF9DAEF35669C1A0E01560EDAD7F4FFE1FC1E2D2F84C6AFA0E42727426BD17A1DD84692A716E6130CB**	**0.81200 s**
.9	**0A2E93EF4E479C4DC16C970F8DBD3D002163B298265E0B18DE3C8A375751095027DA1C24CE2C8CE142DB48F05F4FB1C3E6C037E1646574AC292B61A0CDA203A96FCC3C9C191098722777B4BF5E90C65F5241FF7B01E1D6C5F5DC600539814A10E6C41BB107D071C4F0D44FB3FA9EC1027EEAC89E3756EE4A1FD0454E821801454896D75AE4BACE0C129FFF1C784A10AF331D2D58073EA3**	**1.35900 s**
.10	**071920A3D2CD91AE0CE8DA97265483C87B60A60E5922079D9BCE8E3A7B829D5E79FB3EC9DA6E4**	**1.21900 s**

	694431A4612D208DFE33DA79F97FA32854AC8E83678BC4079BAF8ED35E14CDA78426B8F4F31F63 A8AC363B743D761327F18268811B90571D216C77D38E213DDCF3E4C19549250588EF9359B34BC27B ABFFF3AA45F9510E7DBFE72A2343D5E6D048B79FDCA827707B59C69957A72D964F9	

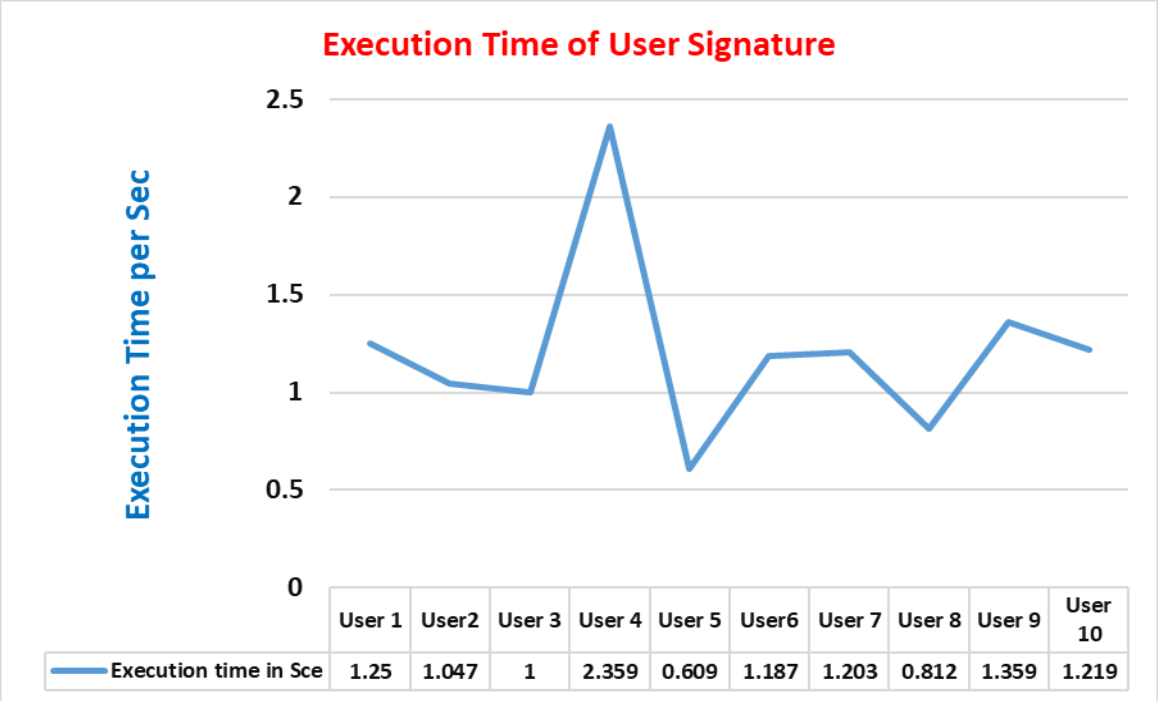


Figure (4.8): Execution Time for each Signature of User Transaction in (Sec).

Table (4.11) shows the results of all steps in the registration phase for 10 users with execution time in second which is the time it takes to create a transaction for each user, and these results are used to create a user transaction. Figure (4.9) illustrated execution time of the create transaction for 100 users. The total execution time in second of create 100 transactions is **00:01:49.0530** sec.

Table (4.11): User Transaction with Execution Time in Second

User ID	SHA 256[New Feature]	public key	privet key	signed data	Execution Time
1	4cdcbf5c3235665899923c7f577c487ab06cc80d240f604311cbbc046cf0f5ef	249839	2010048149323770781406517420898879322012319909910384239482020483431364499114056197087491484967458352636424410402395067049797646544625523618149165487741114848135966480804190019193728165962363459319904638876465846411320583072740824787503630019574889034692682460326403813642993756671460466678487503289359457239 1	2DBE3D3623BC9BBEE2D742EE0871E9CE0EE6320D0B739890F6AA5171FEDCFC73FB3D38A872CA2EE152DED3FDBEA3EAF9247A95690BD13C8B3E814F694AAC4A5533D5754699A5F3A90E687830E3E5CB67768F23C7F7C03AF99220BE5EA344EDEF9403A3E04159CC2E5BF7DB5997F44396559252083D9008126DD9D26861364C57	0.73400 Sec
2	c9eed86fb589959d9b5a4d24d3ab2333d1f6c64e07fa35c8239f8d499e8163fa	205251	2199851528493432810854974139494486667084873222763629429418021665195087037030329587473814090482711129634789082414015173695545289785542995771061242073618756006261300970983817874492632323501849009718357801976260981847013144990845236990743032996725121122130476134689526471539696412107406219485755012456292493793 2	3C1D1A4B56E5FE2BB22FEAA560E9FCCEEFD1CED05166B64E75A092BC9F8FF6E9AE7954DEDFFC31CC54C8F16372ACCCE2A76F088FF6C94CA854484D08C16DAFE5C3D1C1A716AE60B621ECD029239ADCB156FB0A265B3596A8032B25CE0F7C46C983EAE17EF19334F4EF918257A94F12F2FBEF1C23A3005998461EDC734FDD4410	0.64300 Sec
3	726841defc6d61ce3626c6edc9d1b74602921b9b781a2bd5ea0d0dd2492953d9	64493	5849890534088460443442853307705773239842580478873899953908464514558694389480538556171983532949414236858258474552945837731677197614532391336452786230902252357009457129160047105040410865234644915116415780376675543817950265588622892962721547408330122040951823262215881414115887025776396422804429751366725327377 3	6190C38799AF54DA9792B279E2792EA3A9B70C811DE255FF3400F4362B28677E633B65CBFB7F46047F3C6AD0611E311275C1A1239196C319A0122AC154E8A5541EB48500F659911D6EB07D697371FF59BED57798DDB48BAB0F8E515FB680824A021F055818673550A06B4F820E42C55EEBCAFFBB18477D7DC84033D7BF7D9CCB	1.04700 Sec
4	3d2d0fb4861338c3b7359a7ea2a7caad4229f7068e1aa57ae694aa5bb1bb9fd3	190139	1240481824918428902954559480205882097789708984925549066949193677794168435380099036967099386337450362095083503386687808121740994550847442810454891101714533180916568115471870719498301159515361347680563669302648167346570419813182602948174100318696469669869975533195715334489660977952907799427386668445499073637 4	3BDD42B63E78B640948D05D276CCA4991B7CE1094FCAF976B45C983FDA992ADCA6F900449009741E6D9D4BD40BDECDD2E5C7996BD16408C96D7242089BE9B5795A64C75A9F58398C9F9245A365D87EAF1778E8C1ABEB060BDC2EF72EB40163E8C47D152DEFD0AC356D95D5B89EAB7B0A272E9993C40F3F83E5120142B1DCE81E	0.87500 Sec
5	2f1b2c41b2c9a86076acbc251fdb3dbc761c5e6063122ab4ee94280b9174fc2d	33989	2044409290121354945159863054879951906276589983739388687814771876258956437036732707172269917781799400316496387192578921981236920772695176770385994763252569479341189887025653269439278856079132073321625135134154431040140580150669295783225606998913386698215074640051318486937050617201149687913782632617691097195129439348742981373300412517332115649877224069104650786 5	0A0A2B889235AD96705E851FF3A078C2DE184990E91F9CE307A05CC3F76E1D98FFDF4FB09DCBEE2B4A8583F9DE1D905B2E4F82D53EAD14DF14D879A36042603F8AF45737C130693165B4F78A69B3DFC37F9DE1FA84FE3506243DC1FFFC852E95F58454C715E17A3FF1B99E8D9734D89F618A99D042290DDBD8897D6427EE088509DD5466DD27C742FCC4873A4935FEB69B516253C5F329	0.56500Sec
6	242726c1765be5cb96e4f229a4291e4020ce1b029964d1f6bf96e78d623c5224	123221	1564021422340840673808467139071625121209395861043653328175879591973230836439310243236371019756631627779395038897885209559908602790353538015369776814879499754469803540087731385435543171032391385662440367705992771765661329065240195621531971967245 3	03A891493CDE1FC9B248892BE52659FB005C4AB2BB1D098194A6A92F15630A2CD86BF1ABD30724D2E5BAAB0C9E9076695CDC6D379C1E34253709B136E5A9D2A086C7528CDE991B4969022A372D5A4E5C58B9D6D7BB3EC4AC9E7A91B70B818AB6FB94F8829B1AC54559C44	1.71900

			1130904596985127495851931269984195303691338515202658192688547720975278027714030874759836651162139601050616457198665229	FF968B3A1772EA2097CBBD5BA9938E1805975AF9EF26910F424C430F937108CB694F0755510D6A2DCA21B1511	Sec
7	56f2efd9021d03b0276a072701b3ad7769bca0970c844feab02f0fdb43d290fb	208371	1104389067137541918907671502090227329987547257180846982244373690540688002253317907332457740984310413087797633019645129372762629566787818088060732233344791315496341725702627896278172846478190920660413413778442936041109521567684245757034250930041405544288562534524063344385740248735700815394144632901113809074401827393214610136214135981556800044403170867237177480913	029F03166BB82E2BDC17FA4B19FEE0C0FD416E66E7C1357352F40C66512FD1AF3589BF524DAE236ECFA77BA06A38CDA570F8E9DD0E39BD8D611CD780F694FE98AEE851D65CEBE07B97636B05322D163B8CCF8EE8A6D41CFD0BB6D4B758902F340E916E8D016B9E7D7A046196C79887E95849860966315B6AF60F84BF18F656C2CBBD6B1D26EBF11423EEFD052874950AD54202ACB2E26B	0.35800 Sec
8	262c3f5b42ac11e6ef9c3d0dc6aec0ef3d341e11c0047eee556e8464361cd17a	164705	3746039156427070015603977050706172362208706071189826343921262249343974245668387920855971192051510918336703537978176416225488022767686124265028446829487249056259345948988064807691068975808787533867521653270253295647659793856389401764482837110248648029781438019225349521376663349696101169971859240900223180425996787052911293486080365297351346834820874742527658532	0207C21E727EC4EED4EB1C542EBFCC7B37436A0A13624121C082B3D2CBE943681FE677C9A9ED10A3A7E38110C25C0F1BD1444C06F962B913542FEBE589B41B848F7BBA0531AC9EF44C667DA846FD2032E485B61203ACC5F074E2D352CBD182C7624A72706E683778BC5B8601154EEAE399210F29B873B8CCF1988E44930B6CD5C6BF3AE1AFD6C6D75C517F842A0301E27CE56DD5372FD8	1.65600 Sec
9	4cb20e4f2f24a38f40b03876f846906ab3d96b06f13faa9aca30dfbfe73b1644	284849	6258169021464998773564104364157055392955059480777511408546713837269748733743090326503429556861177456274585077821431585280539095667446431949279929147994763324340012083472715523987808960420440231881729306458970916872666947451044974415025072767780413938979299878882436996106093684930009782916230509218716886746536038014488887025261977272203535156108665671563785588	04E101436DCA081D96883D6553B8E1BCD972A3798CCC08687CB74498F03E11C52F7B7613F9F0208902D9776B051E0B9EA4D65CBE5B06A5C282C2440E9F8974DC1A8CFD2063F46F76F7DA9014EE6EB916DC866EC29C126B88960A23FFF44DD456A97BA9B0E745B8407C4789F4D0CDE23A0627014DA25589154F1717A2972524A76F7E6992A9106D676748943940DB47E71E8129B5799065	1.35900 Sec
10	36f592113dd9150053760636c65fcbc1cf07ba1b0246426d2a32069ae7e4cc7c	65913	3415755690550606497956595655848536364109353959975470363174388603662514215399748230206414355743655601542123920225865976143203745754853331999373124688787974744652349500689121053536402443306253846325474365455166748344964413928805925297299907172088683984879624767374630271451191239995088696847147539790662472060400578571139992418107548093125171853191972875498997997	01E08C4041229727455073463B5B12BC752D4A9B7C774CB376AF229A4666589351CB76448C9F2E7CA9F37763F7790911E728B236EB1B7D9EFEF661A5B9150B3E66B2A4190A0A990B1143E6CF91036A8F1917B8AA3565BD49535A2D344ADD2292846424413326B51F223CE68DAF0CD41107009D4D44E3EA45196D5EF18D0AD87401F8EBF525644677B10CBBD67B798C5791553D1CE82A4A	0.48200 Sec

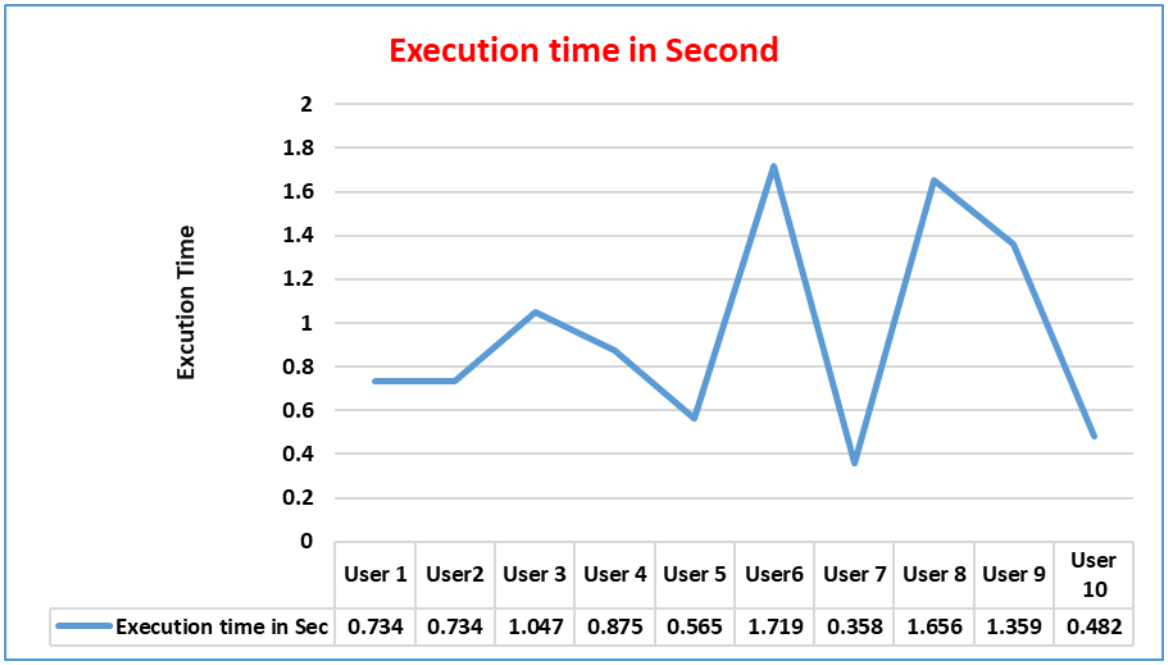


Figure (4.9): Execution Time in Second for Create 10 Transaction.

4.4.2 Results of Authentication Stage

The result of authentication stage represents decryption user signature and compared with incoming SHA-256 hash from user transaction if it's equal then its considered authenticate or (True) transaction otherwise it's not or (False). Table (4.12) shows results of authentication stage where length fixed =20, and key size =1024. The total time = **00:00:02.0972** second. Figure (4.10) clarifies execution time in second of authenticated 10 user transaction that illustrated in table (4.12). Figure (4.11) clarifies execution time in second of authenticated 100 user transaction.

Table (4.12): Result of Authentication Stage.

User ID	New feature	Encryption received	Public key	N	Decryption	SHA 256	Authentication	Execution time
1	6121245710141461313321541080	062813AC170D70389DF4E726E523810AD22045D282921FD69495E23189048C055EFD95BADF82CAF88E06EAFBC2CFB2ADFF98621E049C22CC8BEB9C238334786A7B57AFCFF01B7E89775C5648B8A78C018D9ED4962C240284A30D8782694A21579DC62C30FFE622975198FCDB5A7A676ECA1D120B6700190D9811D4E7D19E164C9F59CE71BBFB9FAD1C7283F1EE23779463C9D4A9C5CAB2	123096984244024585868323510633962054016818319269958857575173111793363312481308056721855412602604772885572195194717573410419271594482453096602158796769964099463911245313819581721813563324451463804010596754034452037268423301973493205453561083753367384059229025754979212674242219970745597295253481115510908412520074231570058059859947051141773882010893466731946395739463C9D4A9C5CAB2	165770574619555785221907452936250880128267719618322554437545172449811433580794734499376633053851026787844431161427896482307061005299214544631112331395967037557695742313394216624333215490203453065108576410177669279868792914159868640072305334623348388537830838229546922920928216744268542076166448228686103165132737047977683550090491171741120469011930695796206672803	fe4d6eba2ca5bbc1de5d7885ef2f3d4ae0c18689b0f06d30d20351b67e17cbf3	fe4d6eba2ca5bbc1de5d7885ef2f3d4ae0c18689b0f06d30d20351b67e17cbf3	true	0.03200 S
2	1-34815814563981120151 5900	02492EB1EF76C5CD3F74B67D54C1C1606567E1FFD9FCE9294D6A5A55C7BD7BB58DABFD2780CC2D8A2FD7154FB42077025527FC460B19B0AC2CFC0C52D9F0E4E282F5A38B101DA598AE8BF33C943A5AFB20F82C2496A2E18450C27E1FD2E8027F95D555CE4EEF5648F4F01B674DAC41EC968898781F1FD80EDDE2FD5D325232DE1B566ACD7813F584529A834C99BDE2ECD373D2627A1327	139037835890558985327134491579734378510316755474477154779324428280788333408209686533028963404459387535144019579743028338188237889008511310281449611687018532968379476510926793348262101095638272409800124356926967370704219069538673996196960414130924148592696065958987888692858915348576384047929876572501183402441095357971703390368100012877821833441631505999207 05777	173518370290544317645614661068097202034037480725485194057677169729488850544092127983969617586729701849081937911195655056247872200209858870940649926421350451316599707351719868375514305609275768833422030426163807984765449720236607214884236143858365076502370907061895913104518431322328466011562568252110066442983285547347099233198032962254818621545196633891610182799	75a0bed06fb0868ab81b4a0cab41ebb76f489b1d0eb3256ed8a5379a0a0d4b56	75a0bed06fb0868ab81b4a0cab41ebb76f489b1d0eb3256ed8a5379a0a0d4b56	true	0.01500 S
3	4031081562615101315128550000	049F151B337E52B3D519BE4E485EA9FEA8714454A68BD69F2F7E67663D2573977CE1D5595C7428E3633FB653094A7C1AA10F0BE81EA2F364F37D1C044CE421154A0EFA19E57E7F2377A39C35E7CC8A255FD19079967F1D72D7598CF9A32842F4A91A9BB08B4F8E110CD792E36ABD58D376225BB360776E	110132407573964982549024798238461234883996459890431993810828176655069577810536703964849248253819620284818437299001112237332010315459526201476337493886342186141020094053057794021849619924799431710759363809806825583341352612280469576078285402447004133543824505256852543756032588870229097938010425	124824004319559391246344532271482984529823720867625033412407222749636814904549907128958598114776062880939776924375724607555320226576803328213080244127494568987522496264406556688788493827098259320908262205772223597275884657074129415558464657344732437494	bdc50edb0e6cf4258d82429915c5a6aa6015790e49f736b453d29ae6f2c1bcd6	bdc50edb0e6cf4258d82429915c5a6aa6015790e49f736b453d29ae6f2c1bcd6	true	0.03100 S

		26C6A7A089014A68664D909F14A1E798074E931E3979A431799CA350C83CF8A1	143397656135363359994738286853416474537371348892345693376316727509575	352521419107484704494821618542665082867147804173698156279567180575374227044115648147636602880519141745143145897			
4	-3-14-611108311524137710050000	0386432A56E8EB0C53A47949250AB5B592D6ACA3BF5C8941D8D536D6542F9800C866D5B1D6856BEF7935115A4F5300EE4E6BFBC5E8B97E4C03713EB4371B14263353D11C07F0164CD6D63C2BBED47CF7D505EA77B4AAB619A241871D104C47B3754115355D4C9EA48390C1F33B77A42FBFEA64E4C3F46A3F51916C2D0D8DEB04B8321AD6D28BC85071E251405597D8E610564879B8897B	832706923420991583898139774240834694701618919535032605202547909221916151197655298596040942746642219890656773939477130501007253776858414797296522549028200219885690255449611095797817631900413039795942083803862857552039740053452623302555405303712904149038008497872146087949195744476674516835847154736010715497870857791583188554465195383520021344307324778678539 62139	177503855967280736624669139990984174318967008708404794853225225799425154802543122467910793255128306266615373091861343573170647931663776550386816099600435941817746480823332409399206067068398312587704356094012473902303081528403469684804512962347034163907679802316164887126613641029023045723 12106464286	c5ab1f56b466011ce9ba0f585f7a27b7e0f46ef81d2b28a0a5a9f45c506ca7b0	c5ab1f56b466011ce9ba0f585f7a27b7e0f46ef81d2b28a0a5a9f45c506ca7b0	true	0.03200 S
5	2-7016101513157058561115000	06A8F57264C4378B30F6916D999A98422B6869F6BB154567456A0362B9841EB6340D03060C35EDC2D8DAA844ABB906F1CDBB3299F1DE4311F330BFA8F3DE216EF0C764C2F682EC4CE20ABCC6E39E4179CC0FEDD54F42275DA62E2D431B0494A81781F111F9DD508EF6E63161833DBC4E1014A67362788A6EB8C892D87EB9037CC26DC5BC5F831736BECC816B2EC3EA79B3E165894C52F3	329185461150505378077866066848804637957933644161957211483826417254229213303731548105550323279852428751429130666328285398829444168891218018864708641846938175217084638036905476265538218212127272402484593878576376211213671573630703387561033831701384330318078789550955303161630542641869893349262068654529621165829549662980122977771678513110777832599024555719112207231	147653758447842454092312719903745523064207506003009740370825701594929861287437481407432995659618845598391095884443744887362389471151981217923874412320434107892997267206623341822733023696952015283252714533364282156036130077771994457635567995989139084024075389869565584782581877199173374036994925227489692472 4222658749381812	2f1b2c41b2c9a86076acbc251fdb3dbc761c5e6063122ab4ee94280b9174fc2d	2f1b2c41b2c9a86076acbc251fdb3dbc761c5e6063122ab4ee94280b9174fc2d	true	0.03100 S
6	7511111141111110710121112260000	0C47F4018F99FCBD3D42CDD9BDCA81792D4FE8786E6B7AE063B2C93E1EE8B4D1773C7AD532C6596DD3D04919438D0A5E92CB24DC6BD109614FA7CEA732FA169B35C82E35351702F32C9C7052968E0E76BCD73B68B304F131DBA68600E65DFAA6C9D3C0C17C84FFA663FB4AE1ABE149B2DBF09D4B0D55CF002372052C6541887EA9EB02AFF20FE961B920E22B8FD30C466C5D5FE52C698C.	191912038305018301745297673839165621554905349511600337938259644083350730903782632771021123625396154094354011391473200677966413167545352052202618611419623303257487470780047269077061210757480479197898125071755383953774953346977318692775807088219114570903265438393090664949303427414152071843122051130054146423828462671866405242084333548704262005407139083148976496019	214855058751825449204878020154387965166748077577619298125543228057244923275423444634921134532791061809035213709454627511808839738436594253056956189302607902160531161223385596526500872972844453336020922194497959346230778109672931551429026440492835201268557105780873987863779422023236491576 70272542327	242726c1765be5cb96e4f229a4291e4020ce1b029964d1f6bf96e78d623c5224	242726c1765be5cb96e4f229a4291e4020ce1b029964d1f6bf96e78d623c5224	True	0.00300 S

7	**359914518215138711111510800**	**0BFB364FC80899DEEB9B54F37D7FB23878845632EA73D6FEF174F1AC9A2ED0D55B32035622B998382DE208ED407F6F225E982E51EA05D3138F61412CED25B348827DD4CA1D37257BDFC7DD2B5F2CEE6477F16E936204E2B63A318874AA9B8D86690B53E8A09DF043897E66CD8F77AA87C2C45F18DF9AB64D809FEC1B6F75D3B742E47FED93B268533EC614805633EC27DBC277024AFABB**	**5460248186319290285979705771190184162229454145411226975549109415243165600872305142079041338430223025718644346254145627692705555137697929493339556790254852446518820161966239823617410188185100644371517217951704794331388371572386382628072358331538379175711233010990095746843687291287752143449002803038815363117383892466907472535604693747614048532863814845093481874**	**2119659786681912035725528722156765649628672310433707515207504872177123454707770196263000078266351585062034653182970275684513563181243641321319662174771729294728130327451631814803588620807253472048029823979568417195544017988633812064378272080048992598169249448433086162762388096446251409842807793344449894956959166458324974**	**56f2efd9021d03b0276a072701b3ad7769bca0970c844feab02f0fdb43d290fb**	**56f2efd9021d03b0276a072701b3ad7769bca0970c844feab02f0fdb43d290fb**	**true**	**0.00300 s**
8	**4-410119151018987311198000**	**0656B05FE6BD0F50CEB2D580183B1BDF605EF592CCAB732A418A5666B57E0A825A0E51E36343EADA6A7B02561B3527B4B1F2EA1F6D2D84403C6118DF57497DC69F358A34F4E7CB18D173684C00CA8D36D8C94958EE4918924EAE09452193FD8D844BD950C87A23DC9551578656BBD8F1693E8E75901D76DD892A5E657F3E89B438FE2790BF0771A2F2220834738150D4D963022E36A72B**	**1332347886173869113554232477674430310183103657351817985050857870310209157514445256819264706309317072572991503118248382355976763279901076643048376304451244447562723789664942844589902450183810531597210866579660501625381642633729273290225794954320388656753075910825272529488406252727829891220166210421241845323717210511116364902092985143107274705728919909287999870** **97**	**2118618320366555136198452609748462374255095083610493892284381746255437190524976915507489539521068735350819810128589714445348001260070831350693495860204145750891346301195825062713392622287102683488130618372362573830587809149032387726662646374988599780657763175885366250801862352204803266212970897612** **5**	**262c3f5b42ac11e6ef9c3d0dc6aec0ef3d341e11c0047eee556e8464361cd17a**	**262c3f5b42ac11e6ef9c3d0dc6aec0ef3d341e11c0047eee556e8464361cd17a**	**true**	**0.00300 s**
9	**12117155106614103606120000**	**083C05F47DF9784576E86333052E10DF6CA1DCD00206561A3A4E0D881E6E211BCBC05233766AE5FF61BDB9A4EFF53E63C300427CB8ADDB32A358C3A420115235722F89972E050C43698B1B99B34EE5E55D1572D0622ADE3CD1C534BEE73B21C8A383518FCBC49C3192E23B56C3C750B961D734157FC7569DA6BE801BB6BBFE748DC37A1FA2FD6630054F556FD754947C5BEDB46FD656C1**	**1871527380518143052851991631493104354164711463031477518072720316595796936096776408008830183509000349458630062345561976995570487060403663095336760286417361641867249385910191640298122720387858009139732190221020986103660259033373052945090868483056394519630125036753741259186839051018698216118938690720007619813203320395348789061849617736907757220641355372890327320** **81**	**2079540503890161518482125530766914614823300349741501736732997621799382198276247672198136999136099081955248694915223371524641459709150431714039946985597021357817509476698535871158541768975782990339512154426568969268934960479980474740126208966256090089810016806271828622974111350245243466764064254368934125469766794294321942**	**4cb20e4f2f24a38f40b03876f846906ab3d96b06f13faa9aca30dfbfe73b1644**	**4cb20e4f2f24a38f40b03876f846906ab3d96b06f13faa9aca30dfbfe73b1644**	**true**	**0.00300 s**
10	**13-13151120437115943270000**	**0522DD16ECF8E1A774C3596EF0540C51034232B078AA31EDCC4110D050E8BCAF850D1C8282F4C4B882E3A117071D9D89B6787279DE3213D2514DBEA00B3C7936961889C**	**6345604033308202964506762566610093602051658280714956147319050613566925624761643615390245359038164048258479441481418818882025457705682636151552498898757959522187473165** **89**	**1004044703191019752514045188738448194861889975167892779315564296178290426216123695837679262254513361298636254984191587937747842128950909250165** **29**	**36f592113dd9150053760636c65fcbc1cf07ba1b0246426d2a32069ae7e4cc7c**	**36f592113dd9150053760636c65fcbc1cf07ba1b0246426d2a32069ae**	**true**	

		CE51269CE10F55788D8B5A88A998C7AF9577D13C8A9567F4BCB469A9046C9754D082EFF977A299B7915E9BF1D0EFA4421E9DC9874D1A27463D433F5B14BBACD877D0DB2AC56BA3B10A593E0BD78B16C18CDF1DC	467885121260206245294533686728700982941808662281152843006587051289729247755466305460462665622881846732964274574776738456700006746929671506791752985143111032224775401888029439521418302041463 10323	785582440170198961054987312277474406407367467602137486037835332228416499264641256138208102160534499349073527473357190495043958045404041174059714675164747068279519 6316001572940078		7e4cc7c		0.00300 S

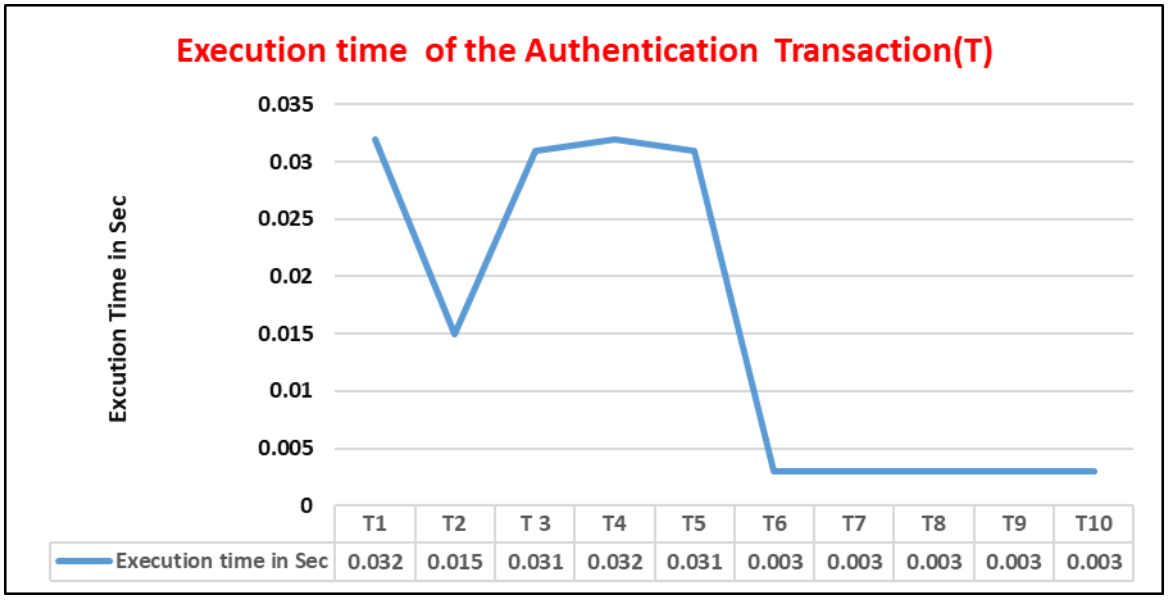


Figure (4.10): Execution Time in Second of the Check Authentication of 10 Transaction.

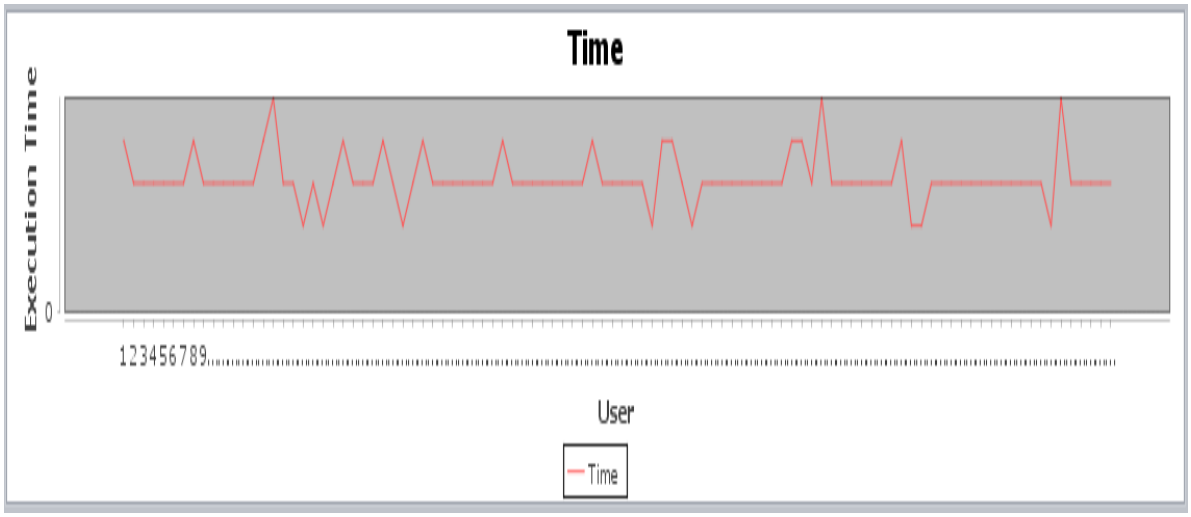


Figure (4.11): Execution Time in Second of the Check Authentication of 100 Transaction.

4.4.3 Results of Builder Merkle Tree Stage

Result of builder merkle tree for all authenticate user transaction based SHA-256 hashing value. Table (4.13) shows builder merkle tree for 8 (even) transactions. Table (4.14) shows builder merkle tree for 7 (odd) transactions.

Table (4.13): Result of Merkle Tree for 8 (even) Transactions.

No. of TX	Hash value of transaction	level1	Level 2	Level3 (Root)
1.	938db8c9f82c8cb58d3f3ef4fd2500036a48d26a712753d2fde5abd03a85cabf4	938db8c9f82c8cb58d3f3ef4fd250036a48d26a712753d2fde5abd03a85cabf4	e0f46e417d0778dca455309e0ee94ee75c4e09d0d201bac7d6879c60f9ced95a	789a758b0ce412b1ded74d9e8482de93c9e1c1e0bd2d8a2a8c83cbc7ec7e0dca
2.	535fa30d7e25dd8a49f1536779734ec8286108d115da5045d77f3b4185d8f790	535fa30d7e25dd8a49f1536779734ec8286108d115da5045d77f3b4185d8f790	370fe02a908f83647df777854a8fb772eb82a8750e05736168e39139fb7958ac	
3.	811786ad1ae74adfdd20dd0372abaaebc6246e343aebd01da0bfc4c02bf0106c	811786ad1ae74adfdd20dd0372abaaebc6246e343aebd01da0bfc4c02bf0106c		
4.	49d180ecf56132819571bf39d9b7b342522a2ac6d23c1418d3338251bfe469c8	49d180ecf56132819571bf39d9b7b342522a2ac6d23c1418d3338251bfe469c8		
5.	cd70bea023f752a0564abb6ed08d42c1440f2e33e29914e55e0be1595e24f45a			
6.	3dd9c0995d54c0abd51a90f1d57b1ce77bc885fc8a7cea52dcad3c2540dda5ee			
7.	de3d43caad2bd3c4f0622fc60deecd06b34a0f25a80e30b81fe051a3c54799bb			
8.	c86f2dfd04b5d52de85408b658cd99e053d9010b38c56da20673c9a891e9746			

Table (4.14): Result of Merkle Tree for 7(odd) Transactions.

No. of TX	Hash value of transaction (level1)	Level 2	Level3	Level4
1	4cdcbf5c3235665899923c7f577c487ab06cc80d240f604311cbbc046cf0f5ef	938db8c9f82c8cb58d3f3ef4fd250036a48d26a712753d2fde5abd03a85cabf4	e0f46e417d0778dca455309e0ee94ee75c4e09d0d201bac7d6879c60f9ced95a	b5d469b97638f9afa2a9a19a0e3dcc630f268d1d38b7cfa5f36c83e1ec859aa5
2	c9eed86fb589959d9b5a4d24d3ab2333d1f6c64e07fa35c8239f8d499e8163fa	535fa30d7e25dd8a49f1536779734ec8286108d115da5045d77f3b4185d8f790	96cd723dcb8df3b5a03957553543d5027e9d645b1ad0d31158de3fc76039e6de	
3	726841defc6d61ce3626c6edc9d1b74602921b9b781a2bd5ea0d0dd2492953d9	811786ad1ae74adfdd20dd0372abaaebc6246e343aebd01da0bfc4c02bf0106c		
4	3d2d0fb4861338c3b7359a7ea2a7caad4229f7068e1aa57ae694aa5bb1bb9fd3	3ada92f28b4ceda38562ebf047c6ff05400d4c572352a1142eedfef67d21e662		
5	36c22960c936fdc781ae7ba0fddd65db41aaa01bfa64f4b8b55a3fc8f06bc8a4			
6	ab73145395db2d15210964fb93da24e4cb6e76c1f8601627096878b67fbd20c8			
7	5ef6b8b1b8a51f1182070ef2288db7653ed16900b7a5fafff363a060c41e38cd			

4.4.4 Results of Create Blocks Stage

Table (4.15) shows results of create blocks by compute values of [previous – hash, Merkle Root Hash (MTH), time-stamp, Current –hash] for each block as illustrated in section (3.3.4).

Table (4.15): Result of create blocks

NO. Block	Previous hash	Merkle Root	Time stamp	Current hash
1.	0000000000	9477cca3d86c652a0303740290212786d0cf149b1d9a9b6e023363dc36ec055a	00:00:02.0937 s	dc2a45fdb4aa489b0de58b71375166988c22164a5e318539d663efd223fe5b99
2.	7bc9a0e9ab59ce2667b9f1436beaba3f759c0420a957dee92e3ab5664f77046b	324c6e7711392127f6378ed3134d8f919b0dff5630c0ad8bced593134174298a	00:00:02.0953 s	6c97594797ee06ad6e78cd2aab71befff3582ccb94854e5146c9bfb6e3738fe1

4.4.5 Results of Authorization Stage

Table (4.16) illustrates results of authorization stage for 10 users. Figure (4.12) shows execution time (per second) to check authorization of 10 user request. Figure (4.13) shows execution time (per second) to check authorization of 100 user request.

Table (4.16): Results of Authorization Stage for 10 user

No. of req.	Value of the hash block(last)	Date-Time	Value hash of the searching block	Result of checking	Time Execution.
1.	ab73145395db2d15210964fb93da24e4cb6e76c1f8601627096878b67fbd20c8	00:00:00.0031 S Fri. Feb 21 12:29:45 AST 2020	ab73145395db2d15210964fb93da24e4cb6e76c1f8601627096878b67fbd20c8	True	0.00200 S
2.	84b8e0c88c80a600a956feb7644185114abfb52524c60192c306892023d2928c	00:00:00.0063 S Ther. Feb 20 10:20:16 AST 2020	84b8e0c88c80a600a956feb7644185114abfb52524c60192c306892023d2928c	True	0.00100 S
3.	6c179f21e6f62b629055d8ab40f454ed02e48b68563913473b857d3638e23b28	00:00:00.0062 S Sund Feb 20 11:36:02 AST 2020	9cbe30d9d4bec188764247cfbabdb1becc1f6baf4d1221f25e6368cd65a2f34e	False	0.00100 S
4.	0d7ac0bb78ce2531c6ff6ef3cf0a19521fa80feda0fd66d2f5b7fd01399e4b98	00:00:00.0125 S Fri Feb 21 12:40:12 AST 2020	0d7ac0bb78ce2531c6ff6ef3cf0a19521fa80feda0fd66d2f5b7fd01399e4b98	True	0.00100 S

5.	126fc22f5a5fa4a5355876fae71f22da ac52445f0da99938851953b84d65650e	00:00:00.0157 S Fri Feb 21 12:41:19 AST 2020	b87ef4142d09b6d318283b66782ab1126d 67621c9b50ac80cd2fd3428abada98	False	0.00200 S
6.	a390105f385d524eb6b2ef0d810c3ea 6b753dc481f5120314c9f0c8284f965ed	00:00:00.0172 S Tus Feb 11 02:12:18 AST 2020	828ee180c971f18826c5047e0745cd44 f8ad14d18ccd9ae5b1d162572653b0e0	False	0.00100 S
7.	b3429882b3adcdeb1e578893f0de72b e6047b5cc7522196376bbd717775c2fa6	00:00:00.0203 S Fri Feb 21 12:46:09 AST 2020	b3429882b3adcdeb1e578893f0de72be 6047b5cc7522196376bbd717775c2fa6	True	0.00600 S
8.	dd126ae28368f51d6a02c2a061e769e8 68ba95e26c9588d46d2d1f0e6615aa10	00:00:00.0234 S Mon Feb 01 09:27:03 AST 2020	dd126ae28368f51d6a02c2a061e769e86 8ba95e26c9588d46d2d1f0e6615aa10	True	0.00100 S
9.	e2eeed14d4bd7c11d6f33ecde1e10b8ca e5a10e0502fdba4fe4f595e3091ab29	00:00:00.0281 S Ther Feb 06 05:14:24 AST 2020	e2eeed14d4bd7c11d6f33ecde1e10b8cae 5a10e0502fdba4fe4f595e3091ab29	True	0.00100 S
10.	863df924c7332a96829ce5be991fe9d65 aad5fdfd34d9bfdb802cef020b2db52	00:00:00.0265 S Fri Feb 21 12:55:53 AST 2020	863df924c7332a96829ce5be991fe9d65a ad5fdfd34d9bfdb802cef020b2db52	True	0.00100 S

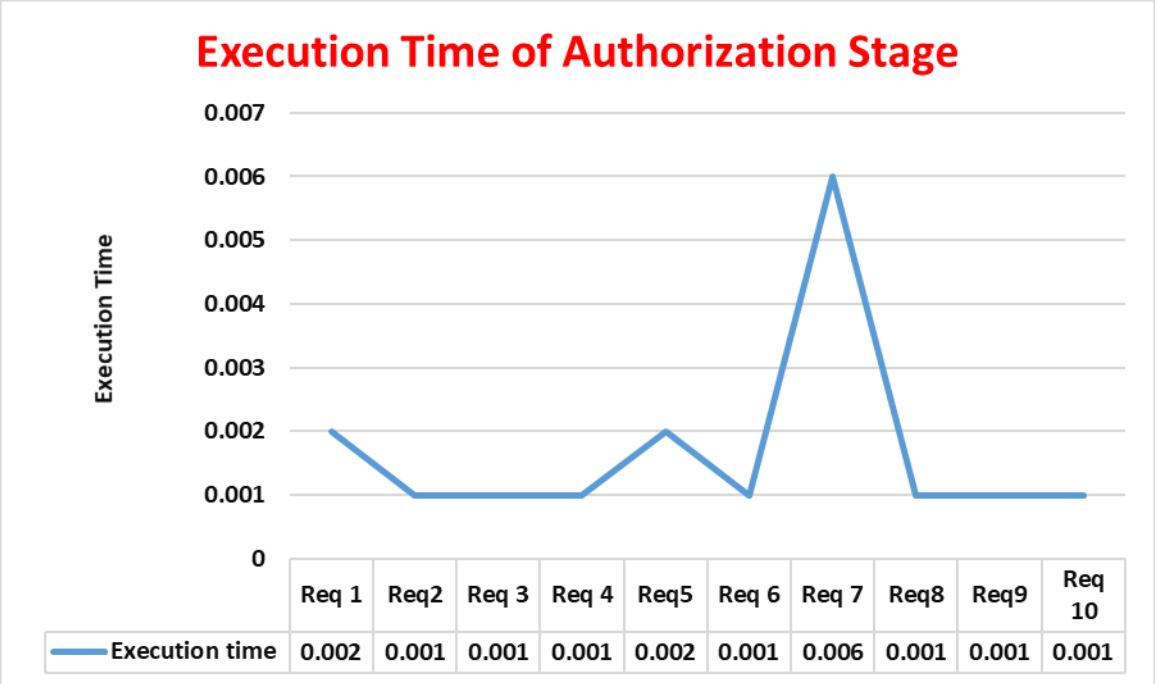


Figure (4.12): Execution time (in second) for check authorization of the 10 user request.

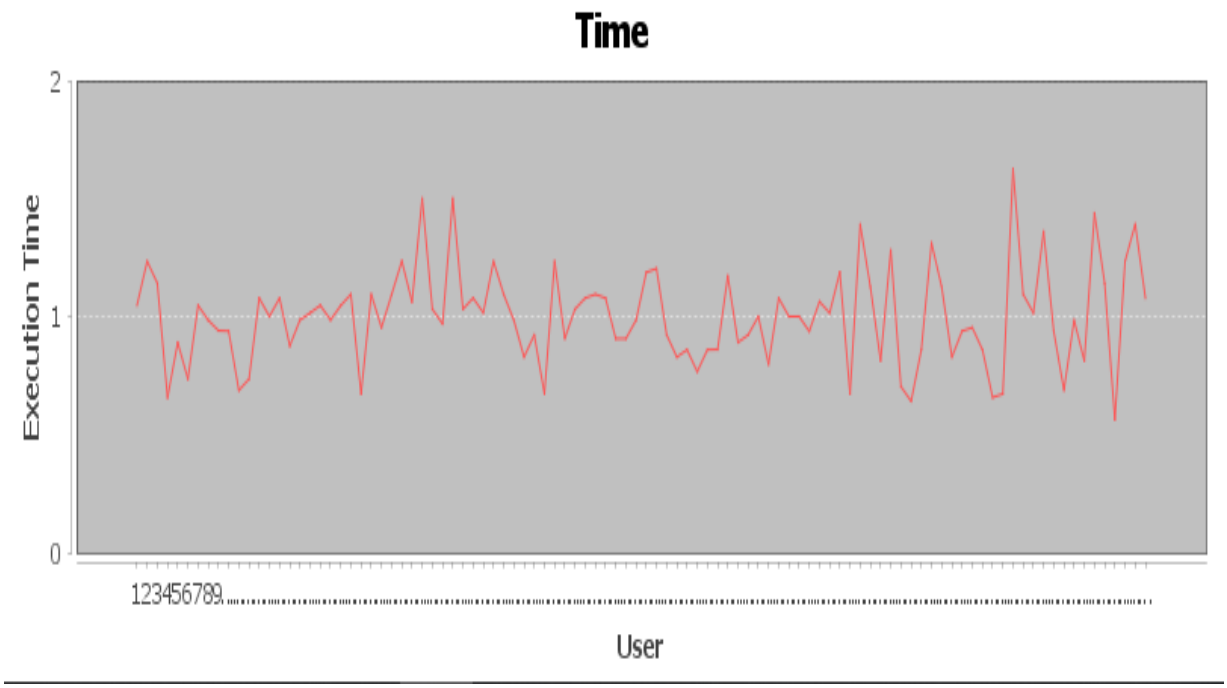


Figure (4.13): Execution time (in second) for check authorization of the 100 user request.

4.4.6 Results of Linking Block to ASBchain Network Stage.

The result of linking block to ASBchain is illustrated in table (4.17).

Table (4.17): Result of Linking Block to ASBchain Network**.**

No.	Previous hash of block	Current hash
1.	0000000000000000000000000000000000000000000000000000000000000000	ba9761d91abf1c0f70e150b0a636ac5f991d72877ce0b57033197f831fa12505
2.	ba9761d91abf1c0f70e150b0a636ac5f991d72877ce0b57033197f831fa12505	55df734366b29d21449143950e18c699ac4d4f36dc25a9e87b76ed0b0aa70af4
3.	55df734366b29d21449143950e18c699ac4d4f36dc25a9e87b76ed0b0aa70af4	2d60252c94e501a0f9d5cdf42f0e5bb4601a639caaff6daa49011871a7401952
4.	2d60252c94e501a0f9d5cdf42f0e5bb4601a639caaff6daa49011871a7401952	37281eb6b2df0de386c7ab45dadd095d941256eb8e38a874e1c25fcc443268ca
5.	37281eb6b2df0de386c7ab45dadd095d941256eb8e38a874e1c25fcc443268ca	157102d90ce6d04d4fbcfc7637ef9310fa9ace5745dd90f2fc6813f646ef26872
6.	157102d90ce6d04d4fbcfc7637ef9310f	a594dfe9d1c2dfcdef0f6dc2400a753f2ea725fa59

	a9ace5745dd90f2fc6813f646ef26	**dfe02bfa283da492a6aa77**
7.	**a594dfe9d1c2dfcdef0f6dc2400a753f2e a725fa59dfe02bfa283da492a6aa77**	**d67f17eefeb9688163b7fa508b20c95afd7f1e43f0 e5474a98d44d6c694c392b**
8.	**d67f17eefeb9688163b7fa508b20c95af d7f1e43f0e5474a98d44d6c694c392b**	**c13523dd8a3f1e618a8730bf5f64a7a10f504b391 00ca71d0d4fcc8cb2bf429b**
9.	**c13523dd8a3f1e618a8730bf5f64a7a10 f504b39100ca71d0d4fcc8cb2bf429b**	**3dbf35e5acb7bf0eb622907277a4f06c8fd8bb2fd 54ba53e75545e9bda5f4e42**
10.	**3dbf35e5acb7bf0eb622907277a4f06c8 fd8bb2fd54ba53e75545e9bda5f4e42**	**f071a72a45991136cd55765221fe618cb9a4d853 32537b1a378440836e2de8b0**

4.5 Comparison between Three stages of the ASBchain System based on Total Execution Time

This section presents comparison between (registration step, authentication step, and authorization step) in a proposed ASBchain system based on totally execution time in second for 100 users as shown in table (4.18) and figure (4.14).

Table (4. 18): Total Execution Time of Registration, authentication, and authorization stages

Time / Stage	**Total Execution Time in Sec**
Registration	00:01:43.0059
Authentication	00.0100.0102
Authorization	00:00:02.0953

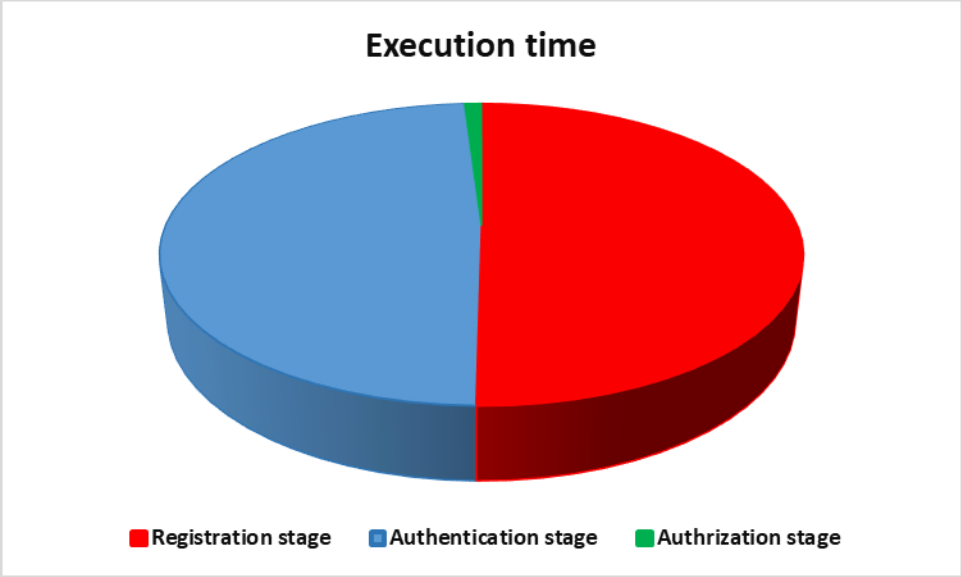


Figure (4.14): Execution time (in second) for three stages: registration, authentication, and authorization.

Chapter Five

Conclusions and Suggestions for Future Work

Chapter Five

Conclusions and Suggestions for Future Work

5.1 Introduction

This chapter finishes this thesis with some conclusions about the implementation and results of the proposed ASBchain system. Section (5.2) present the conclusions, and section (5.3) explaining suggestions for future work.

5.2 Conclusions

Some conclusions can be inferred from the results and tests of this thesis as follows:

1. Section (4.3) demonstrated that the proposed ASBchain system is successfully implemented as blockchain basics through all stages of the proposed ASBchain system. Through create transaction and verifying it then create Merkle tree, block, and then linking it.

2. The result of the registration stage in a proposed ASBchain system attempt to improvement the security level of the proposed system using trusted user data through using strong RSA and using SHA256 keeping the integrity of data from Manipulation during transmission.

3. Tables (4.5) and (4.6) show generate large prim numbers and choose from these numbers in random manner a public key for each user and based on LCG algorithm and Rabin miller testing.

4. Table (4.10) and figure (4.8) shown the proposed system able to integrity data transaction based on digital signature algorithm using user key (public, private) in less time, where the value of the total execution time for signature 100 users is (0.51100 sec).

5. Table (4.11) illustrated the proposed system able to create transaction for 100 users with the total execution time in second of creating 100 transactions is 00:01:49.0530 sec.
6. Table (4.12) and figure (4.10) clarifies the proposed authentication method in proposed ASBchain system has been maintaining authentication of the sender as a guaranteed process of sharing and disseminating information. The total execution time of authenticated 100 user transaction is (00:01:00:0102 sec).

7. Table (4.13) and (4.14) illustrated the proposed system able to build Markel Tree for (even and odd) number of transactions using the proposed Merkle tree method.
8. Table (4.15) shows the ability of proposed system to create blocks based on proposed create block algorithm using SHA256 algorithm for block header.
9. Table (4.16) and figure (4.12) show that not everyone is necessarily certified at the authentication stage it is authorized, depending on the value of the hashtags sent according to the time the block was created. It may be that the value of the block is wrong and the total execution time of 100 user is about (00:00:02:0953 sec).
10. The ASBchain system proved a user with one identity can access multiple nodes without sharing any private user data multiple times.

11. Table (4.17) shows the proposed ASBchain will be created, which will be all transactions are certified from all the network, which cannot be tampered with by linking each block with the previous block by calculating SHA256 Hashing values for them and thus will be a chain linked together with a time stamp.
12. Table (4.18) and figure (4.14) show that the total execution time for three main stages in proposed system (registration ,authentication, and authorization), the proposed system keep the execution time within acceptable limits that reflect on low implementation cost that make the proposed system suitable for different application .

5.3 Suggestions for Future Work

During this work, the possible future works for Blockchain take several directions as follows:

- The ASBchain system is implemented as a simulation environment, so that it can be applying or implementing on virtual or real environment.
- The Registration part could be developed by adding real servers, real nodes, or real applications as client applications such as a wallet or some smart contracts on network which nowadays is developing fast.
- Adapting the ASBchain system for a large network consisting from large numbers of nodes or servers to make the system more reliable.
- The ASBchain system can be adapting to other approaches of authentication or authorization.

Finally, the proposed system can be tested with real type of platforms of blockchain networks such as Bitcoin or Ethereum with actual resources such as (clients, persons, smart contracts, or nodes) and implemented on the main-net of blockchain in order to find the real run time , and real costs .

References

References

1. *Morabito, Vincenzo. "Business innovation through blockchain." Cham: Springer International Publishing (2017).*
2. *Thakur, Mukesh. "Authentication, Authorization and Accounting with Ethereum Blockchain." URL: https://helda. Helsinki. Fi/handle/10138/228842 (visited on 07/03/2018) (2017).*
3. *Vujičić, Dejan, Dijana Jagodić, and Siniša Ranđić. "Blockchain technology, bitcoin, and Ethereum: A brief overview." 2018 17th international symposium infoteh-jahorina (infoteh). IEEE, 2018.*
4. *Nakamoto, Satoshi, and A. Bitcoin. "A peer-to-peer electronic cash system." Bitcoin.–URL: https://bitcoin. Org/bitcoin. Pdf (2008).*
5. *Chen, Zhixong, and Yixuan Zhu. "Personal archive service system using blockchain technology: case study, promising and challenging." 2017 IEEE International Conference on AI & Mobile Services (AIMS). IEEE, 2017.*
6. *Lin, Iuon-Chang, and Tzu-Chun Liao. "A Survey of Blockchain Security Issues and Challenges." IJ Network Security 19.5 (2017): 653-659.*
7. *Trnka, Michal, Tomas Cerny, and Nathaniel Stickney. "Survey of Authentication and Authorization for the Internet of Things." Security and Communication Networks 2018 (2018).*
8. *Shoeb, Md Zahid Hossain, and M. Abdus Sobhan. "Authentication and authorization: security issues for institutional digital repositories." Library Philosophy and Practice (2010): 1.*

9. *Zheng, Zibin, et al. "An overview of blockchain technology: Architecture, consensus, and future trends." 2017 IEEE international congress on big data (BigData congress). IEEE, 2017.*

10. *Mohanta, Bhabendu Kumar, Soumyashree S. Panda, and Debasish Jena. "An overview of smart contract and use cases in blockchain technology." 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2018.*

11. *Grech, Alexander, and Anthony F. Camilleri. "Blockchain in education." (2017).*

12. *Kikitamara, Sesaria, M. C. J. D. van Eekelen, and Dipl Ing Jan-Peter Doomernik. "Digital identity management on blockchain for open model energy system." Unpublished Master's thesis–Information Science (2017).*

13. *Xia, Qi, et al. "BBDS: Blockchain-based data sharing for electronic medical records in cloud environments." Information 8.2 (2017): 44.*

14. *Moinet, Axel, Benoît Darties, and Jean-Luc Baril. "Blockchain based trust & authentication for decentralized sensor networks." arXiv preprint arXiv: 1706.01730 (2017).*

15. *Hammudoglu, J. S., et al. "Portable trust: biometric-based authentication and blockchain storage for self-sovereign identity systems." arXiv preprint arXiv: 1706.03744 (2017).*

16. *GAO, Zhimin, et al. "Blockchain-based identity management with mobile device." Proceedings of the 1st workshop on Cryptocurrencies and blockchains for distributed systems. 2018.*

17. *Yin, Wei, et al. "An anti-quantum transaction authentication approach in blockchain." IEEE Access 6 (2018): 5393-5401.*

18. *Narayanan, Arvind, et al. "Bitcoin and cryptocurrency technologies: a comprehensive introduction". Princeton University Press, 2016.*

19. *Pawade, Dipti, et al. "Implementation of Fingerprint-Based Authentication System Using Blockchain." Soft Computing and Signal Processing. Springer, Singapore, 2019. 233-242.*

20. *Swan, Melanie. "Blockchain for business: next-generation enterprise artificial intelligence systems." Advances in computers. Vol. 111. Elsevier, 2018. 121-162.*

21. *Underwood, Sarah. "Blockchain beyond bitcoin". Communications of the ACM 59.11 (2016): 15-17.*

22. *Jutila, Laura. "The blockchain technology and its applications in the financial sector." (2017).*

23. *Yaga, Dylan, et al. "Blockchain technology overview." Draft NISTIR 8202 (2018).*

24. *Bashir, Imran. "Mastering Blockchain". Packt Publishing Ltd, 2017.*

25. *Sikorski, Janusz J., Joy Haughton, and Markus Kraft. "Blockchain technology in the chemical industry: Machine-to-machine electricity market." Applied Energy 195 (2017): 234-246.*

26. *Costa, Pier Francesco." Ethereum blockchain as a decentralized and autonomous key server: storing and extracting public keys through smart contracts". Diss.*

27. *Berryhill, Jamie, Théo Bourgery, and Angela Hanson. "Blockchains unchained." (2018).*

28. *Meunier, Sebastien. "Blockchain 101: What is blockchain and how does this revolutionary technology work?" Transforming climate finance and green investment with Blockchains. Academic Press, 2018. 23-34.*

29. *Triantafyllidis, Nikolaos Petros, and T. N. O. Oskar van Deventer. "Developing an Ethereum blockchain application". Diss. Ph. D. Thesis, University of Amsterdam, Amsterdam, the Netherlands, 2016.*

30. *Rachmawati, D., J. T. Tarigan, and A. B. C. Ginting. "A comparative study of Message Digest 5 (MD5) and SHA256 algorithm." Journal of Physics: Conference Series. Vol. 978. No. 1. 2018.*

31. *Es-Samaali, Hamza, Aissam Outchakoucht, and Jean Philippe Leroy. "A blockchain-based access control for big data." International Journal of Computer Networks and Communications Security 5.7 (2017): 137.*

32. *Haffke, Florian. "Technical Analysis of Established Blockchain Systems." Master's thesis. Technical University of Munich, SW Engineering for Business Informatics (2017).*

33. *Seibold, Sigrid, and George Samman. "Consensus: Immutable agreement for the Internet of value." KPMG< https://assets. Kpmg. com/content/dam/kpmg/pdf/2016/06/kpmgblockchain-consensus-mechanism. Pdf (2016).*

34. *Li, Xiaoqi, et al. "A survey on the security of blockchain systems." Future Generation Computer Systems (2017).*

35. *Otieno, Brian Ricky. "Biometric Authorization and Authentication."*

36. *Tiwari, Tanuj, Tanya Tiwari, and Sanjay Tiwari. "Biometrics based user authentication." American Journal of Engineering Research 4.10 (2015): 148-159.*

37. *Babich, Aleksandra. "Biometric Authentication. Types of biometric identifiers." (2012).*

38. *Jaiswal, Sushma, Sarita Singh Bhadauria, and Rakesh Singh Jadon. "Biometric: case study." Journal of Global Research in Computer Science 2.10 (2011): 19-48.*

39. *Ain, N. U., et al. "An Efficient Algorithm for Fingerprint Recognition Using Minutiae Extraction." Pakistan Journal of Science 70.2 (2018).*

40. *Yang, Ju Cheng, and Dong Sun Park. "Fingerprint verification based on invariant moment features and nonlinear BPNN." International Journal of Control, Automation, and Systems 6.6 (2008): 800-808.*

41. *Huang, Zhihu, and Jinsong Leng. "Analysis of Hu's moment invariants on image scaling and rotation." 2010 2nd International Conference on Computer Engineering and Technology. Vol. 7. IEEE, 2010.*

42. *Monem S., Esraa Q." Fingerprint Image Features Extraction Using Moment Invariants" Iraqi Journal of Information Technology. Vol.7. No.-4, 2017.*

43. *Li, Chung-Chih, and Bo Sun. "Using Linear Congruential Generators for Cryptographic Purposes." Computers and Their Applications. 2005.*

44. *Bassil, Youssef, and Aziz Barbar. "Sequential and parallel algorithms for the addition of big-integer numbers." arXiv preprint arXiv: 1204.0232 (2012).*

45. *Halim, Steven, and Felix Halim. Competitive Programming: Increasing the Lower Bound of Programming Contests. National University of Singapore, 2010.*

46. *Rahim, Robbi, et al. "Prime number: an experiment Rabin-Miller and fast exponentiation." Journal of Physics: Conference Series. Vol. 930. No. 1. IOP Publishing, 2017.*

47. *Pavuluru, Rajesh. "Miller-Rabin." (2015).*

48. *Chandra, Sayani, et al. "Generate an Encryption Key by using Biometric Cryptosystems to secure transferring of Data over a Network." IOSR Journal of Computer Engineering (IOSR-JCE) 12.1 (2013): 16-22.*

49. *Rashid, Mofeed Turky, and Huda Ameer Zaki. "RSA Cryptographic key generation using fingerprint minutiae." Iraqi Journal for Computers and Informatics ijci 41.1 (2014): 66-69.*

50. *Conti, Vincenzo, Salvatore Vitabile, and Filippo Sorbello. "Fingerprint traits and RSA algorithm fusion technique." 2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems. IEEE, 2012.*

51. *Xu, Jennifer J. "Are blockchains immune to all malicious attacks?" Financial Innovation 2.1 (2016): 1-9.*

52. *Jamsrandorj, Tom. DECENTRALIZED ACCESS CONTROL USING THE BLOCKCHAIN. Diss. University of Saskatchewan, 2017.*

الخلاصة

ان النمو الهائل للبيانات وجميع التطبيقات المستخدمة على الشبكات أمانًا وأمانًا كبيرين. يمكن دمج تقنية Blockchain مع مجموعة متنوعة من التقنيات الأخرى لأنها تدخل في المجالات الرقمية والفيزيائية والبيولوجية. تقنية Blockchain هي قاعدة بيانات مستقرة ومشتركة لا يتحكم فيها أي طرف ثالث. تعد المصادقة أيضًا مشكلة يجب التحقيق فيها بدقة حتى تتم المصادقة عليها بصرف النظر عن طرق المصادقة التقليدية المستخدمة لإثبات أن الشخص مخول. في هذا التصميم ، تم اقتراح نظام Blockchain بطريقة محاكاة لكل عقدة على النظام.حيث تم تصميم تقنية التخويل على شبكة البلوكشين والذي يسمى ASBchain من عدد من العقد للتحقق من المعاملات التي يرسلها المستخدم بعد عملية التسجيل ، والتي تستند إلى خوارزمية SHA256 وخوارزمية RSA القوية. في هذا النظام المقترح ، يتم تسجيل والتحقق من صحة أي معاملة تتم على أساس مطابقة قيم دالة التجزئة بعد التوقيع عليها باستخدام المفتاح الخاص والذي يولد باستخدام Big-integer والمفتاخ العام باستخدام معادلة RSA (LCG). وبالتالي ، يمكن استخدام دالة التجزئة في عملية الترخيص بشرط أن تتم مصادقة الشخص أولاً. يتم ذلك وفقًا لقيمة دالة التجزئة الأخيرة للكتلة التي يحملها هذا المرسل استنادًا إلى طابعه الزمني. تم اختبار النظام من حيث الوقت. تمت مقارنة مراحل النظام المقترحة مع بعضها البعض وتبين أن الوقت الذي يقضيه في عمليات التسجيل والمصادقة والترخيص كان (00:01:43:0059),(00:01:02.0953) و(00:01:00.0102) كجزء من الثانية وعلى التوالي لمئة مستخدم. لقد بين النظام أن جميع الأشخاص لديهم حقوق متساوية في المصادقة والوصول. ولكن ليس كل شخص مصادق عليه يعتبر مخول. ولاتوجد عقدة خاصة لها القدرة المطلقة لإدارة التحكم في blockchain. النظام هو بيئة تعاونية والشيء المهم هو الموثوقية والسرية واللامركزية.

وزارة التعليم العالي والبحث العلمي
جامعة ديالى - كلية العلوم
قسم علوم الحاسوب

تصميم تقنية الترخيص في بيئة المحاكاة بلوكشين

رسالة

مقدمة الى قسم علوم الحاسوب – كلية العلوم – جامعة ديالى كجزء من متطلبات نيل درجة الماجستير في إختصاص علوم الحاسوب

من قِبل

وسن احمد علي

بإشراف

أ. ناجي مطر سحاب

د. جمانة وليد صالح